



ΕΛΛΗΝΙΚΟ ΜΕΣΟΓΕΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ

Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Πτυχιακή εργασία

Τίτλος: Πληροφοριακό σύστημα για αεροπορική εταιρεία

Ζαχαρούλα Ζαχαρίου (ΑΜ: 3655)

Επιβλέπων εκπαιδευτικός: Νικόλαος Παπαδάκης

ΙΑΝΟΥΑΡΙΟΣ 2021

Ευχαριστίες

Θα ήθελα με τη παρούσα πτυχιακή εργασία, να ευχαριστήσω θερμά τα εξής σημαντικά πρόσωπα για εμένα. Την μητέρα μου, πού με στήριξε ψυχολογικά να τελειοποιήσω τις σπουδές μου, αλλά και να αντιμετωπίζω τις όποιες δυσκολίες της ζωής. Τον πατέρα μου, πού μπορεί να μην είναι πλέον εν ζωή, αλλά ήταν και θα είναι πάντα μαζί μου στα εύκολα και στα δύσκολα. Την αδερφή μου, πού με έμαθε να επιμένω και να μην παρατάω την κάθε μου προσπάθεια σε ό,τι κι αν κάνω στην ζωή μου. Επίσης, θα ήθελα να ευχαριστήσω τη γιαγιά μου, η οποία με το χιούμορ και την αισιοδοξία της με έμαθε να βλέπω πάντα την θετική όψη των πραγμάτων. Τέλος, θα ήθελα να ευχαριστήσω εξίσου θερμά τον επιβλέπων καθηγητή Δρ. Νικόλαο Παπαδάκη, πού κατά τη διάρκεια των σπουδών μου, με βοήθησε να κυνηγάω τους στόχους μου και να διευρύνω τους ορίζοντές μου.

Abstract

The purpose of this thesis is to study, design and develop an information system for booking and issuing tickets for an airline company.

More specifically, the Python programming language was used for backend development, as well as the HTML, JavaScript, CSS languages for frontend development. The Web Framework used is Python Django. JavaScript jQuery library was also used. Regarding the database management system, Microsoft SQL Server v18.7.1 was used. Finally, AJAX technology was used to exchange data between the client and the server, after the page was loaded without the need to reload.

The system is developed, with the central theme of managing the booking of an airline company, allows a user to register in the information system as a customer, as a travel agent, as well as an administrator. The user-customer can make reservations through the system, issue tickets and see various information related to his account. The user-travel agent can make reservations and issue tickets on behalf of a customer, as well as enter new customer details. The user-administrator can manage pilots, flight attendants, airplanes, travel agents, other administrators, users, airline routes and transports. It can also see aggregate reports, related to the fullness of the itineraries.

Σύνοψη

Σκοπός της παρούσας πτυχιακής εργασίας είναι η μελέτη, σχεδίαση και ανάπτυξη ενός πληροφοριακού συστήματος διαχείρισης κρατήσεων και έκδοσης εισιτηρίων για μια αεροπορική εταιρεία.

Πιο συγκεκριμένα, χρησιμοποιήθηκε η γλώσσα προγραμματισμού Python για backend development, καθώς και οι γλώσσες HTML, JavaScript, CSS για frontend development. Το Web Framework, το οποίο χρησιμοποιήθηκε, είναι το Django της Python. Επίσης, έγινε χρήση της βιβλιοθήκης jQuery της JavaScript. Όσο αφορά το σύστημα διαχείρισης βάσεων δεδομένων, χρησιμοποιήθηκε ο Microsoft SQL Server v18.7.1. Τέλος, έγινε χρήση της τεχνολογίας AJAX για ανταλλαγή δεδομένων μεταξύ του πελάτη (client) και εξυπηρετητή (server), αφού φορτωθεί η σελίδα χωρίς την ανάγκη επαναφόρτωσης (reloading) της σελίδας.

Το σύστημα που αναπτύχθηκε, με κεντρικό θέμα τη διαχείριση των κρατήσεων μιας αεροπορικής εταιρείας, δίνει τη δυνατότητα σε έναν χρήστη να εγγραφεί στο πληροφοριακό σύστημα ως πελάτης, ως ταξιδιωτικός πράκτορας, καθώς και ως διαχειριστής. Ο χρήστης-πελάτης, μπορεί μέσα από το σύστημα να κάνει κρατήσεις, να εκδώσει εισιτήρια και να δει διάφορες πληροφορίες, σχετικές με το λογαριασμό του. Ο χρήστης-ταξιδιωτικός πράκτορας, μπορεί να κάνει κρατήσεις και να εκδώσει εισιτήρια για λογαριασμό ενός πελάτη, καθώς και να καταχωρεί στοιχεία νέων πελατών. Ο χρήστης-διαχειριστής, μπορεί να διαχειριστεί τα στοιχεία των πιλότων, των αεροσυνοδών, των αεροπλάνων, των ταξιδιωτικών πρακτόρων, των διοικητικών υπαλλήλων, των χρηστών, των δρομολογίων και των μεταφορών. Επίσης, μπορεί να δει συγκεντρωτικές αναφορές, σχετικές με την πληρότητα των δρομολογίων.

Πίνακας Περιεχομένων

1	Εισαγωγή.....	1
1.1	Περίληψη.....	1
1.2	Σκοπός και στόχοι της εργασίας.....	1
1.3	Δομή εργασίας.....	1
2	Μεθοδολογία Υλοποίησης.....	2
2.1	Μέθοδος Ανάλυσης & Ανάπτυξης.....	2
2.2	Βασικά Προγράμματα Δημιουργίας Βάσης Δεδομένων & Ιστοσελίδων....	2
2.2.1	Εισαγωγή στο Microsoft SQL Server Management Studio.....	2
2.2.2	Εισαγωγή στο Visual Studio Code.....	3
2.3	Ροή εργασίας.....	4
2.3.1	Περιγραφή ροής εργασίας.....	5
3	Μοντελοποίηση της εφαρμογής.....	18
3.1	Ανάλυση εφαρμογής.....	18
3.2	Σχεδιασμός Υλοποίησης.....	18
3.2.1	Δημιουργία πινάκων στη βάση δεδομένων.....	18
3.2.2	Εισαγωγή εικόνων στον editor.....	22
3.2.3	Δημιουργία σελίδων HTML.....	24
3.2.4	Δημιουργία σελίδων JavaScript.....	26
3.2.5	Δημιουργία σελίδων CSS.....	27
3.2.6	Δημιουργία models.....	29
3.2.7	Δημιουργία URLs.....	31
3.2.8	Δημιουργία stored procedures στη βάση δεδομένων.....	32
3.2.9	Δημιουργία views.....	33
3.3	Υλοποίηση.....	35

3.3.1	<i>Microsoft SQL Server Management Studio</i>	35
3.3.2	<i>Visual Studio Code</i>	75
4	Αποτελέσματα	224
4.1	<i>Εκτέλεση της εφαρμογής</i>	224
4.2	<i>Χρήση της εφαρμογής</i>	227
4.3	<i>Συμπεράσματα</i>	278
	<i>Βιβλιογραφία</i>	279
	<i>Παραπομπές</i>	280

Λίστα Εικόνων

Εικόνα 1.	Εικονικό περιβάλλον του Microsoft SQL Server Management Studio.....	3
Εικόνα 2.	Εικονικό περιβάλλον του Visual Studio Code.....	4
Εικόνα 3.	Διάγραμμα οντοτήτων-συσχετίσεων (E-R)	5
Εικόνα 4.	Εγκατάσταση Visual Studio Code Βήμα 1 ^ο	6
Εικόνα 5.	Εγκατάσταση Visual Studio Code Βήμα 2 ^ο	7
Εικόνα 6.	Εγκατάσταση Visual Studio Code Βήμα 3 ^ο	8
Εικόνα 7.	Εγκατάσταση Visual Studio Code Βήμα 4 ^ο	9
Εικόνα 8.	Εγκατάσταση Python Βήμα 1 ^ο	10
Εικόνα 9.	Εγκατάσταση Python Βήμα 2 ^ο	10
Εικόνα 10.	Εγκατάσταση Python Βήμα 3 ^ο	11
Εικόνα 11.	Εγκατάσταση Python στο Visual Studio Code Βήμα 1 ^ο	11
Εικόνα 12.	Εγκατάσταση Python στο Visual Studio Code Βήμα 2ο.....	12
Εικόνα 13.	Εγκατάσταση Python στο Visual Studio Code Βήμα 3 ^ο	12
Εικόνα 14.	Εγκατάσταση Microsoft SQL Server Management Studio Βήμα 1ο.....	13
Εικόνα 15.	Εγκατάσταση Microsoft SQL Server Management Studio Βήμα 2ο.....	14
Εικόνα 16.	Εγκατάσταση Microsoft SQL Server Management Studio Βήμα 3 ^ο	14
Εικόνα 17.	Εγκατάσταση Microsoft SQL Server Management Studio Βήμα 4ο.....	15
Εικόνα 18.	Εγκατάσταση Microsoft SQL Server Management Studio Βήμα 5ο.....	16
Εικόνα 19.	Εγκατάσταση Βάσης Δεδομένων Βήμα 1ο	16
Εικόνα 20.	Εγκατάσταση Βάσης Δεδομένων Βήμα 2ο	17
Εικόνα 21.	Βάση Δεδομένων του Microsoft SQL Server Management Studio	17
Εικόνα 22.	Δημιουργία πινάκων στη βάση δεδομένων με άμεσες ενέργειες Βήμα 1ο	18
Εικόνα 23.	Δημιουργία πινάκων στη βάση δεδομένων με άμεσες ενέργειες Βήμα 2ο	19
Εικόνα 24.	Δημιουργία πινάκων στη βάση δεδομένων με άμεσες ενέργειες Βήμα 3 ^ο	20
Εικόνα 25.	Ανανέωση λίστας πινάκων	20
Εικόνα 26.	Δημιουργία πινάκων στη βάση δεδομένων με άμεσες ενέργειες Βήμα 4 ^ο	21
Εικόνα 27.	Δημιουργία πινάκων στη βάση δεδομένων με χρήση κώδικα SQL.....	22
Εικόνα 28.	Εισαγωγή εικόνων στον editor (1)	23
Εικόνα 29.	Εισαγωγή εικόνων στον editor (2)	24
Εικόνα 30.	Δημιουργία σελίδων HTML	24
Εικόνα 31.	Βασική σύνταξη σελίδας HTML	25
Εικόνα 32.	Δημιουργία σελίδων JavaScript.....	26
Εικόνα 33.	Βασική σύνταξη σελίδας JavaScript	27
Εικόνα 34.	Δημιουργία σελίδων CSS	28

Εικόνα 35. Βασική σύνταξη σελίδας CSS	29
Εικόνα 36. Σελίδα models	30
Εικόνα 37. Βασική σύνταξη σελίδας models	30
Εικόνα 38. Σελίδα URL	31
Εικόνα 39. Βασική σύνταξη σελίδας urls	32
Εικόνα 40. Δημιουργία stored procedure Βήμα 1ο.....	32
Εικόνα 41. Δημιουργία stored procedure Βήμα 2°	33
Εικόνα 42. Σελίδα views	34
Εικόνα 43. Δημιουργία view	35
Εικόνα 44. Δημιουργία πίνακα ταξιδιωτικών πρακτόρων	35
Εικόνα 45. Δημιουργία πίνακα αεροσυνοδών.....	36
Εικόνα 46. Δημιουργία πίνακα αεροπλάνων.....	36
Εικόνα 47. Δημιουργία πίνακα πόλεων	36
Εικόνα 48. Δημιουργία πίνακα πελατών.....	36
Εικόνα 49 Δημιουργία πίνακα μερών	36
Εικόνα 50. Δημιουργία πίνακα υπαλλήλων	37
Εικόνα 51. Δημιουργία πίνακα διοικητικών υπαλλήλων	37
Εικόνα 52. Δημιουργία πίνακα πιλότων	37
Εικόνα 53. Δημιουργία πίνακα κρατήσεων.....	37
Εικόνα 54. Δημιουργία πίνακα κρατήσεων σε μεταφορές.....	37
Εικόνα 55. Δημιουργία πίνακα δρομολογίων.....	38
Εικόνα 56. Δημιουργία πίνακα εισιτηρίων	38
Εικόνα 57. Δημιουργία πίνακα εισιτηρίων σε μεταφορές.....	38
Εικόνα 58 Δημιουργία πίνακα μεταφορών	38
Εικόνα 59. Δημιουργία πίνακα χρηστών	39
Εικόνα 60. Αποθηκευμένη διαδικασία διαγραφής ταξιδιωτικού πράκτορα	39
Εικόνα 61. Αποθηκευμένη διαδικασία εισαγωγής ταξιδιωτικού πράκτορα.....	39
Εικόνα 62. Αποθηκευμένη διαδικασία αναζήτησης ταξιδιωτικού πράκτορα	39
Εικόνα 63. Αποθηκευμένη διαδικασία επιλογής ταξιδιωτικού πράκτορα	40
Εικόνα 64. Αποθηκευμένη διαδικασία ανανέωσης ταξιδιωτικού πράκτορα.....	40
Εικόνα 65. Αποθηκευμένη διαδικασία διαγραφής αεροσυνοδού.....	40
Εικόνα 66. Αποθηκευμένη διαδικασία εισαγωγής αεροσυνοδού	40
Εικόνα 67. Αποθηκευμένη διαδικασία αναζήτησης αεροσυνοδού.....	41
Εικόνα 68. Αποθηκευμένη διαδικασία επιλογής όλων των αεροσυνοδών	41
Εικόνα 69. Αποθηκευμένη διαδικασία επιλογής αεροσυνοδού	41
Εικόνα 70. Αποθηκευμένη διαδικασία ανανέωσης αεροσυνοδού	42
Εικόνα 71. Αποθηκευμένη διαδικασία διαγραφής αεροπλάνου.....	42
Εικόνα 72. Αποθηκευμένη διαδικασία εισαγωγής αεροπλάνου.....	42
Εικόνα 73. Αποθηκευμένη διαδικασία αναζήτησης αεροπλάνου	42
Εικόνα 74. Αποθηκευμένη διαδικασία επιλογής όλων των αεροπλάνων	43
Εικόνα 75. Αποθηκευμένη διαδικασία επιλογής αεροπλάνου	43
Εικόνα 76. Αποθηκευμένη διαδικασία ανανέωσης αεροπλάνου.....	43
Εικόνα 77. Αποθηκευμένη διαδικασία ακύρωσης κρατήσεων.....	44
Εικόνα 78. Αποθηκευμένη διαδικασία αλλαγής κωδικού.....	44
Εικόνα 79. Αποθηκευμένη διαδικασία επιλογής όλων των πόλεων	45
Εικόνα 80. Αποθηκευμένη διαδικασία διαγραφής πελάτη.....	45
Εικόνα 81. Αποθηκευμένη διαδικασία εισαγωγής πελάτη.....	45
Εικόνα 82. Αποθηκευμένη διαδικασία αναζήτησης πελάτη	45
Εικόνα 83. Αποθηκευμένη διαδικασία επιλογής όλων των πελατών	46
Εικόνα 84. Αποθηκευμένη διαδικασία επιλογής πελάτη	46
Εικόνα 85. Αποθηκευμένη διαδικασία ανανέωσης πελάτη.....	46
Εικόνα 86. Αποθηκευμένη διαδικασία διαγραφής μέρας	46
Εικόνα 87. Αποθηκευμένη διαδικασία εισαγωγής μέρας	47
Εικόνα 88. Αποθηκευμένη διαδικασία επιλογής όλων των ημερών.....	47
Εικόνα 89. Αποθηκευμένη διαδικασία ανανέωσης μέρας	47

Εικόνα 90. Αποθηκευμένη διαδικασία διαγραφής υπαλλήλου	47
Εικόνα 91. Αποθηκευμένη διαδικασία εισαγωγής υπαλλήλου	48
Εικόνα 92. Αποθηκευμένη διαδικασία επιλογής υπαλλήλου	48
Εικόνα 93. Αποθηκευμένη διαδικασία ανανέωσης υπαλλήλου	48
Εικόνα 94. Αποθηκευμένη διαδικασία επιλογής αριθμού θέσεων αεροπλάνου.....	49
Εικόνα 95. Αποθηκευμένη διαδικασία επιλογής στοιχείων πελάτη	49
Εικόνα 96. Αποθηκευμένη διαδικασία επιλογής μεταφορών με ενδιάμεσες στάσεις	50
Εικόνα 97. Αποθηκευμένη διαδικασία επιλογής άμεσων μεταφορών.....	50
Εικόνα 98. Αποθηκευμένη διαδικασία επιλογής τελευταίας μεταφοράς.....	51
Εικόνα 99. Αποθηκευμένη διαδικασία αναφοράς πληρότητας πτήσεων.....	51
Εικόνα 100. Αποθηκευμένη διαδικασία διαγραφής διοικητικού υπαλλήλου	51
Εικόνα 101. Αποθηκευμένη διαδικασία εισαγωγής διοικητικού υπαλλήλου.....	52
Εικόνα 102. Αποθηκευμένη διαδικασία αναζήτησης διοικητικού υπαλλήλου	52
Εικόνα 103. Αποθηκευμένη διαδικασία επιλογής όλων των διοικητικών υπαλλήλων.....	52
Εικόνα 104. Αποθηκευμένη διαδικασία επιλογής διοικητικού υπαλλήλου	53
Εικόνα 105. Αποθηκευμένη διαδικασία ανανέωσης διοικητικού υπαλλήλου	53
Εικόνα 106. Αποθηκευμένη διαδικασία διαγραφής πιλότου.....	53
Εικόνα 107. Αποθηκευμένη διαδικασία εισαγωγής πιλότου.....	54
Εικόνα 108. Αποθηκευμένη διαδικασία αναζήτησης πιλότου	54
Εικόνα 109. Αποθηκευμένη διαδικασία επιλογής όλων των πιλότων	54
Εικόνα 110. Αποθηκευμένη διαδικασία επιλογής πιλότου	55
Εικόνα 111. Αποθηκευμένη διαδικασία ανανέωσης πιλότου.....	55
Εικόνα 112. Αποθηκευμένη διαδικασία διαγραφής κράτησης.....	56
Εικόνα 113. Αποθηκευμένη διαδικασία δημιουργίας κράτησης.....	57
Εικόνα 114. Αποθηκευμένη διαδικασία αναζήτησης κράτησης	58
Εικόνα 115. Αποθηκευμένη διαδικασία επιλογής κράτησης	58
Εικόνα 116. Αποθηκευμένη διαδικασία ανανέωσης κράτησης.....	59
Εικόνα 117. Αποθηκευμένη διαδικασία ελέγχου εγκυρότητας κράτησης	60
Εικόνα 118. Αποθηκευμένη διαδικασία διαγραφής δρομολογίου.....	60
Εικόνα 119. Αποθηκευμένη διαδικασία εισαγωγής δρομολογίου.....	60
Εικόνα 120. Αποθηκευμένη διαδικασία αναζήτησης δρομολογίου	60
Εικόνα 121. Αποθηκευμένη διαδικασία επιλογής όλων των δρομολογίων	61
Εικόνα 122. Αποθηκευμένη διαδικασία επιλογής δρομολογίου	61
Εικόνα 123. Αποθηκευμένη διαδικασία ανανέωσης δρομολογίου.....	61
Εικόνα 124. Αποθηκευμένη διαδικασία εισόδου χρήστη	62
Εικόνα 125. Αποθηκευμένη διαδικασία δημιουργίας εισιτηρίου (1).....	63
Εικόνα 126. Αποθηκευμένη διαδικασία δημιουργίας εισιτηρίου (2).....	64
Εικόνα 127. Αποθηκευμένη διαδικασία αναζήτησης εισιτηρίου	65
Εικόνα 128. Αποθηκευμένη διαδικασία επιλογής εισιτηρίου	65
Εικόνα 129. Αποθηκευμένη διαδικασία διαγραφής εισιτηρίου.....	65
Εικόνα 130. Αποθηκευμένη διαδικασία αναζήτησης διαθέσιμων μεταφορών δρομολογίου	66
Εικόνα 131. Αποθηκευμένη διαδικασία εισαγωγής μεταφοράς.....	66
Εικόνα 132. Αποθηκευμένη διαδικασία αναζήτησης μεταφοράς.....	67
Εικόνα 133. Αποθηκευμένη διαδικασία επιλογής μεταφοράς	67
Εικόνα 134. Αποθηκευμένη διαδικασία ανανέωσης μεταφοράς.....	68
Εικόνα 135. Αποθηκευμένη διαδικασία ελέγχου εγκυρότητας μεταφοράς	69
Εικόνα 136. Αποθηκευμένη διαδικασία εισαγωγής χρήστη-διαχειριστή.....	70
Εικόνα 137. Αποθηκευμένη διαδικασία αναζήτησης χρήστη-διαχειριστή	70
Εικόνα 138. Αποθηκευμένη διαδικασία ανανέωσης χρήστη-διαχειριστή	70
Εικόνα 139. Αποθηκευμένη διαδικασία εισαγωγής χρήστη-ταξιδιωτικού πράκτορα	71
Εικόνα 140. Αποθηκευμένη διαδικασία αναζήτησης χρήστη-ταξιδιωτικού πράκτορα.....	71
Εικόνα 141. Αποθηκευμένη διαδικασία ανανέωσης χρήστη-ταξιδιωτικού πράκτορα	71
Εικόνα 142. Αποθηκευμένη διαδικασία εισαγωγής χρήστη-πελάτη	72
Εικόνα 143. Αποθηκευμένη διαδικασία αναζήτησης χρήστη-πελάτη.....	72
Εικόνα 144. Αποθηκευμένη διαδικασία ανανέωσης χρήστη-πελάτη	72

Εικόνα 145. Αποθηκευμένη διαδικασία διαγραφής χρήστη	73
Εικόνα 146. Αποθηκευμένη διαδικασία εισαγωγής χρήστη	73
Εικόνα 147. Αποθηκευμένη διαδικασία επιλογής χρήστη	74
Εικόνα 148. Αποθηκευμένη διαδικασία ανανέωσης χρήστη	74
Εικόνα 149. Εικόνες προς χρήση στην εφαρμογή	75
Εικόνα 150. Σελίδα html επεξεργασίας ταξιδιωτικού πράκτορα.....	75
Εικόνα 151. Σελίδα html αναζήτησης ταξιδιωτικού πράκτορα.....	76
Εικόνα 152. Σελίδα html επεξεργασίας αεροσυνοδού	77
Εικόνα 153. Σελίδα html αναζήτησης αεροσυνοδού	78
Εικόνα 154. Σελίδα html επεξεργασίας αεροπλάνου	79
Εικόνα 155. Σελίδα html αναζήτησης αεροπλάνου	80
Εικόνα 156. Σελίδα html ταμπλό πελάτη.....	81
Εικόνα 157. Σελίδα html επεξεργασίας πελάτη.....	82
Εικόνα 158. Σελίδα html αναζήτησης πελάτη.....	83
Εικόνα 159. Σελίδα html επεξεργασίας κράτησης πελάτη.....	84
Εικόνα 160. Σελίδα html αναζήτησης κράτησης πελάτη.....	85
Εικόνα 161. Σελίδα html προεπισκόπησης εισιτηρίου στην οθόνη του πελάτη.....	86
Εικόνα 162. Σελίδα html προεπισκόπησης εισιτηρίου στην οθόνη του ταξιδιωτικού πράκτορα.....	87
Εικόνα 163. Σελίδα html αρχικής σελίδας.....	88
Εικόνα 164. Σελίδα html εισόδου χρήστη.....	89
Εικόνα 165. Σελίδα html επεξεργασίας διοικητικού υπαλλήλου	90
Εικόνα 166. Σελίδα html αναζήτησης διοικητικού υπαλλήλου.....	91
Εικόνα 167. Σελίδα html επεξεργασίας πιλότου	92
Εικόνα 168. Σελίδα html αναζήτησης πιλότου.....	93
Εικόνα 169. Σελίδα html συγκεντρωτικής αναφοράς πτήσεων.....	94
Εικόνα 170. Σελίδα html επεξεργασίας κράτησης ταξιδιωτικού πράκτορα.....	95
Εικόνα 171. Σελίδα html αναζήτησης κράτησης ταξιδιωτικού πράκτορα.....	96
Εικόνα 172. Σελίδα html επεξεργασίας δρομολογίου.....	97
Εικόνα 173. Σελίδα html αναζήτησης δρομολογίου	98
Εικόνα 174. Σελίδα html επεξεργασίας μεταφοράς (1)	99
Εικόνα 175. Σελίδα html επεξεργασίας μεταφοράς (2)	100
Εικόνα 176. Σελίδα html αναζήτησης μεταφοράς.....	101
Εικόνα 177. Σελίδα html επεξεργασίας χρήστη-διαχειριστή (1).....	102
Εικόνα 178. Σελίδα html επεξεργασίας χρήστη-διαχειριστή (2).....	103
Εικόνα 179. Σελίδα html αναζήτησης χρήστη-διαχειριστή.....	104
Εικόνα 180. Σελίδα html επεξεργασίας χρήστη-ταξιδιωτικού πράκτορα (1).....	105
Εικόνα 181. Σελίδα html επεξεργασίας χρήστη-ταξιδιωτικού πράκτορα (2).....	106
Εικόνα 182. Σελίδα html αναζήτησης χρήστη-ταξιδιωτικού πράκτορα	107
Εικόνα 183. Σελίδα html επεξεργασίας χρήστη-πελάτη (1).....	108
Εικόνα 184. Σελίδα html επεξεργασίας χρήστη-πελάτη (2).....	109
Εικόνα 185. Σελίδα html αναζήτησης χρήστη-πελάτη	110
Εικόνα 186. Σελίδα html χρηστών.....	111
Εικόνα 187. Σελίδα html καλωσορίσματος χρήστη-διαχειριστή	111
Εικόνα 188. Σελίδα html καλωσορίσματος χρήστη-ταξιδιωτικού πράκτορα	112
Εικόνα 189. Σελίδα html καλωσορίσματος χρήστη-πελάτη	112
Εικόνα 190. Σελίδα javascript επεξεργασίας ταξιδιωτικού πράκτορα (1)	113
Εικόνα 191. Σελίδα javascript επεξεργασίας ταξιδιωτικού πράκτορα (2)	114
Εικόνα 192. Σελίδα javascript αναζήτησης ταξιδιωτικού πράκτορα.....	115
Εικόνα 193. Σελίδα javascript επεξεργασίας αεροσυνοδού (1).....	116
Εικόνα 194. Σελίδα javascript επεξεργασίας αεροσυνοδού (2).....	117
Εικόνα 195. Σελίδα javascript αναζήτησης αεροσυνοδού	118
Εικόνα 196. Σελίδα javascript επεξεργασίας αεροπλάνου (1)	119
Εικόνα 197. Σελίδα javascript επεξεργασίας αεροπλάνου (2)	120
Εικόνα 198. Σελίδα javascript αναζήτησης αεροπλάνου	121
Εικόνα 199. Σελίδα javascript επεξεργασίας πελάτη (1)	122

Εικόνα 200. Σελίδα javascript επεξεργασίας πελάτη (2)	123
Εικόνα 201. Σελίδα javascript αναζήτησης πελάτη	124
Εικόνα 202. Σελίδα javascript επεξεργασίας κράτησης πελάτη (1)	125
Εικόνα 203. Σελίδα javascript επεξεργασίας κράτησης πελάτη (2)	126
Εικόνα 204. Σελίδα javascript επεξεργασίας κράτησης πελάτη (3)	127
Εικόνα 205. Σελίδα javascript αναζήτησης κράτησης πελάτη	128
Εικόνα 206. Σελίδα javascript αρχικής σελίδας (1)	129
Εικόνα 207. Σελίδα javascript αρχικής σελίδας (2)	130
Εικόνα 208. Σελίδα javascript εισόδου χρήστη	131
Εικόνα 209. Σελίδα javascript επεξεργασίας διοικητικού υπαλλήλου (1)	132
Εικόνα 210. Σελίδα javascript επεξεργασίας διοικητικού υπαλλήλου (2)	133
Εικόνα 211. Σελίδα javascript αναζήτησης διοικητικού υπαλλήλου	134
Εικόνα 212. Σελίδα javascript επεξεργασίας πιλότου (1)	135
Εικόνα 213. Σελίδα javascript επεξεργασίας πιλότου (2)	136
Εικόνα 214. Σελίδα javascript αναζήτησης πιλότου	137
Εικόνα 215. Σελίδα javascript συγκεντρωτικής αναφοράς πτήσεων	138
Εικόνα 216. Σελίδα javascript επεξεργασίας κράτησης ταξιδιωτικού πράκτορα (1)	139
Εικόνα 217. Σελίδα javascript επεξεργασίας κράτησης ταξιδιωτικού πράκτορα (2)	140
Εικόνα 218. Σελίδα javascript επεξεργασίας κράτησης ταξιδιωτικού πράκτορα (3)	141
Εικόνα 219. Σελίδα javascript αναζήτησης κράτησης ταξιδιωτικού πράκτορα	142
Εικόνα 220. Σελίδα javascript επεξεργασίας δρομολογίου (1)	143
Εικόνα 221. Σελίδα javascript επεξεργασίας δρομολογίου (2)	144
Εικόνα 222. Σελίδα javascript αναζήτησης δρομολογίου	145
Εικόνα 223. Σελίδα javascript προεπισκόπησης εισιτηρίου	146
Εικόνα 224. Σελίδα javascript επεξεργασίας μεταφοράς (1)	147
Εικόνα 225. Σελίδα javascript επεξεργασίας μεταφοράς (2)	148
Εικόνα 226. Σελίδα javascript επεξεργασίας μεταφοράς (3)	149
Εικόνα 227. Σελίδα javascript αναζήτησης μεταφοράς	150
Εικόνα 228. Σελίδα javascript επεξεργασίας χρήστη-διαχειριστή (1)	151
Εικόνα 229. Σελίδα javascript επεξεργασίας χρήστη-διαχειριστή (2)	152
Εικόνα 230. Σελίδα javascript επεξεργασίας χρήστη-διαχειριστή (3)	153
Εικόνα 231. Σελίδα javascript αναζήτησης χρήστη-διαχειριστή	154
Εικόνα 232. Σελίδα javascript επεξεργασίας χρήστη-ταξιδιωτικού πράκτορα (1)	155
Εικόνα 233. Σελίδα javascript επεξεργασίας χρήστη-ταξιδιωτικού πράκτορα (2)	156
Εικόνα 234. Σελίδα javascript επεξεργασίας χρήστη-ταξιδιωτικού πράκτορα (3)	157
Εικόνα 235. Σελίδα javascript αναζήτησης χρήστη-ταξιδιωτικού πράκτορα	158
Εικόνα 236. Σελίδα javascript επεξεργασίας χρήστη-πελάτη (1)	159
Εικόνα 237. Σελίδα javascript επεξεργασίας χρήστη-πελάτη (2)	160
Εικόνα 238. Σελίδα javascript επεξεργασίας χρήστη-πελάτη (3)	161
Εικόνα 239. Σελίδα javascript αναζήτησης χρήστη-πελάτη	162
Εικόνα 240. Σελίδα javascript χρηστών	163
Εικόνα 241. Σελίδα css επεξεργασίας ταξιδιωτικού πράκτορα	163
Εικόνα 242. Σελίδα css αναζήτησης ταξιδιωτικού πράκτορα	164
Εικόνα 243. Σελίδα css επεξεργασίας αεροσυνοδού	164
Εικόνα 244. Σελίδα css αναζήτησης αεροσυνοδού	165
Εικόνα 245. Σελίδα css επεξεργασίας αεροπλάνου	165
Εικόνα 246. Σελίδα css αναζήτησης αεροπλάνου	166
Εικόνα 247. Σελίδα css ταμπλό πελάτη	166
Εικόνα 248. Σελίδα css επεξεργασίας πελάτη	167
Εικόνα 249. Σελίδα css αναζήτησης πελάτη	167
Εικόνα 250. Σελίδα css επεξεργασίας κράτησης πελάτη	168
Εικόνα 251. Σελίδα css αναζήτησης κράτησης πελάτη	169
Εικόνα 252. Σελίδα css αρχικής σελίδας	169
Εικόνα 253. Σελίδα css εισόδου χρήστη	170
Εικόνα 254. Σελίδα css επεξεργασίας διοικητικού υπαλλήλου	170

Εικόνα 255. Σελίδα css αναζήτησης διοικητικού υπαλλήλου.....	171
Εικόνα 256. Σελίδα css επεξεργασίας πιλότου.....	171
Εικόνα 257. Σελίδα css αναζήτησης πιλότου.....	172
Εικόνα 258. Σελίδα css συγκεντρωτικής αναφοράς πτήσεων.....	172
Εικόνα 259. Σελίδα css επεξεργασίας κράτησης ταξιδιωτικού πράκτορα.....	173
Εικόνα 260. Σελίδα css αναζήτησης κράτησης ταξιδιωτικού πράκτορα.....	174
Εικόνα 261. Σελίδα css επεξεργασίας δρομολογίου.....	174
Εικόνα 262. Σελίδα css αναζήτησης δρομολογίου.....	175
Εικόνα 263. Σελίδα css προεπισκόπησης εισιτηρίου.....	175
Εικόνα 264. Σελίδα css πάνω πανό στοιχείων χρήστη & εξόδου του.....	176
Εικόνα 265. Σελίδα css επεξεργασίας μεταφοράς.....	176
Εικόνα 266. Σελίδα css αναζήτησης μεταφοράς.....	177
Εικόνα 267. Σελίδα css επεξεργασίας χρήστη-διαχειριστή.....	177
Εικόνα 268. Σελίδα css αναζήτησης χρήστη-διαχειριστή.....	178
Εικόνα 269. Σελίδα css επεξεργασίας χρήστη-ταξιδιωτικού πράκτορα.....	178
Εικόνα 270. Σελίδα css αναζήτησης χρήστη-ταξιδιωτικού πράκτορα.....	179
Εικόνα 271. Σελίδα css επεξεργασίας χρήστη-πελάτη.....	179
Εικόνα 272. Σελίδα css αναζήτησης χρήστη-πελάτη.....	180
Εικόνα 273. Σελίδα css χρηστών.....	180
Εικόνα 274. Σελίδα css καλωσορίσματος χρήστη.....	181
Εικόνα 275. Μοντέλο ταξιδιωτικού πράκτορα.....	181
Εικόνα 276. Μοντέλο αεροπλάνου.....	181
Εικόνα 277. Μοντέλο πελάτη.....	182
Εικόνα 278. Μοντέλο υπαλλήλου.....	182
Εικόνα 279. Μοντέλο πιλότου.....	182
Εικόνα 280. Μοντέλο αεροσυνοδού.....	182
Εικόνα 281. Μοντέλο διοικητικού υπαλλήλου.....	183
Εικόνα 282. Μοντέλο χρήστη-ταξιδιωτικού πράκτορα.....	183
Εικόνα 283. Μοντέλο χρήστη-πελάτη.....	183
Εικόνα 284. Μοντέλο χρήστη-διαχειριστή.....	183
Εικόνα 285. Μοντέλο χρηστών.....	183
Εικόνα 286. Μοντέλο δρομολογίου.....	184
Εικόνα 287. Μοντέλο γραμμής δρομολογίου.....	184
Εικόνα 288. Μοντέλο πόλης.....	184
Εικόνα 289. Μοντέλο μέρας.....	185
Εικόνα 290. Μοντέλο μεταφοράς.....	185
Εικόνα 291. Μοντέλο αντικειμένου δρομολογίου.....	185
Εικόνα 292. Μοντέλο αντικειμένου πιλότου.....	186
Εικόνα 293. Μοντέλο αντικειμένου αεροσυνοδού.....	186
Εικόνα 294. Μοντέλο αντικειμένου διοικητικού υπαλλήλου.....	186
Εικόνα 295. Μοντέλο αντικειμένου αεροπλάνου.....	187
Εικόνα 296. Μοντέλο γραμμής μεταφοράς.....	187
Εικόνα 297. Μοντέλο κράτησης.....	187
Εικόνα 298. Μοντέλο αντικειμένου πελάτη.....	188
Εικόνα 299. Μοντέλο αντικειμένου μεταφοράς.....	188
Εικόνα 300. Μοντέλο γραμμής κράτησης.....	188
Εικόνα 301. Μοντέλο εισιτηρίου.....	188
Εικόνα 302. Μοντέλο πληροφοριών πελάτη.....	189
Εικόνα 303. Μοντέλο πληρότητας πτήσεων.....	189
Εικόνα 304. URLs εισόδου διαχειριστή & επεξεργασίας ταξιδιωτικού πράκτορα.....	189
Εικόνα 305. URLs επεξεργασίας αεροπλάνου.....	189
Εικόνα 306. URLs επεξεργασίας πελάτη.....	190
Εικόνα 307. URLs επεξεργασίας πιλότου.....	190
Εικόνα 308. URLs επεξεργασίας αεροσυνοδού.....	190
Εικόνα 309. URLs επεξεργασίας διοικητικού υπαλλήλου.....	190

Εικόνα 310. URLs επεξεργασίας δρομολογίου	190
Εικόνα 311. URLs επεξεργασίας χρήστη-πελάτη	191
Εικόνα 312. URLs επεξεργασίας χρήστη-ταξιδιωτικού πράκτορα	191
Εικόνα 313. URLs επεξεργασίας χρήστη-διαχειριστή.....	191
Εικόνα 314. URLs επεξεργασίας μεταφοράς.....	191
Εικόνα 315. URLs επεξεργασίας κράτησης ταξιδιωτικού πράκτορα	191
Εικόνα 316. URLs επεξεργασίας κράτησης πελάτη	191
Εικόνα 317. URLs επεξεργασίας & εισόδου χρήστη.....	192
Εικόνα 318. URLs αναζήτησης των εγγραφών	192
Εικόνα 319. URLs εισιτηρίου	192
Εικόνα 320. URL ταμπλό πελάτη.....	193
Εικόνα 321. URLs συγκεντρωτικής αναφοράς πτήσεων	193
Εικόνα 322. Εισαγωγή βιβλιοθηκών της Django, JSON, Email & μοντέλων εγγραφών	193
Εικόνα 323. View αρχικής σελίδας εισόδου	194
Εικόνα 324. View κρυπτογράφηση κωδικού	194
Εικόνα 325. View ελέγχου εγκυρότητας κωδικού.....	194
Εικόνα 326. View ελέγχου εγκυρότητας στοιχείων χρήστη	195
Εικόνα 327. View αποσύνδεσης χρήστη.....	195
Εικόνα 328. View αρχικής σελίδας εφαρμογής.....	195
Εικόνα 329. View αλλαγής κωδικού χρήστη	196
Εικόνα 330. View επεξεργασίας ταξιδιωτικού πράκτορα.....	196
Εικόνα 331. View αναζήτησης ταξιδιωτικού πράκτορα.....	196
Εικόνα 332. View αποθήκευσης ταξιδιωτικού πράκτορα.....	197
Εικόνα 333. View διαγραφής ταξιδιωτικού πράκτορα	197
Εικόνα 334. View ελέγχου εγκυρότητας ταξιδιωτικού πράκτορα.....	198
Εικόνα 335. View επεξεργασίας αεροπλάνου.....	198
Εικόνα 336. View αναζήτησης αεροπλάνου	199
Εικόνα 337. View αποθήκευσης αεροπλάνου	199
Εικόνα 338. View διαγραφής αεροπλάνου	199
Εικόνα 339. View ελέγχου εγκυρότητας αεροπλάνου	200
Εικόνα 340. View επεξεργασίας πελάτη.....	200
Εικόνα 341. View αναζήτησης πελάτη.....	201
Εικόνα 342. View αποθήκευσης πελάτη.....	201
Εικόνα 343. View διαγραφής πελάτη	201
Εικόνα 344. View ελέγχου εγκυρότητας πελάτη.....	202
Εικόνα 345. View επεξεργασίας πιλότου.....	202
Εικόνα 346. View αναζήτησης πιλότου.....	202
Εικόνα 347. View αποθήκευσης πιλότου.....	203
Εικόνα 348. View διαγραφής πιλότου	203
Εικόνα 349. View ελέγχου εγκυρότητας πιλότου.....	203
Εικόνα 350. View επεξεργασίας αεροσυνοδού	204
Εικόνα 351. View αναζήτησης αεροσυνοδού	204
Εικόνα 352. View αποθήκευσης αεροσυνοδού	204
Εικόνα 353. View διαγραφής αεροσυνοδού.....	205
Εικόνα 354. View ελέγχου εγκυρότητας αεροσυνοδού	205
Εικόνα 355. View επεξεργασίας διοικητικού υπαλλήλου.....	206
Εικόνα 356. View αναζήτησης διοικητικού υπαλλήλου.....	206
Εικόνα 357. View αποθήκευσης διοικητικού υπαλλήλου	206
Εικόνα 358. View διαγραφής διοικητικού υπαλλήλου	207
Εικόνα 359. View ελέγχου εγκυρότητας διοικητικού υπαλλήλου	207
Εικόνα 360. View σελίδας χρηστών	207
Εικόνα 361. View διαγραφής χρήστη	208
Εικόνα 362. View ελέγχου εγκυρότητας χρήστη	208
Εικόνα 363. View επεξεργασίας χρήστη-πελάτη	208
Εικόνα 364. View αποθήκευσης χρήστη-πελάτη	209

Εικόνα 365. View αναζήτησης χρήστη-πελάτη	209
Εικόνα 366. View ελέγχου εγκυρότητας χρήστη-πελάτη	210
Εικόνα 367. View επεξεργασίας χρήστη-ταξιδιωτικού πράκτορα	210
Εικόνα 368. View αποθήκευσης χρήστη-ταξιδιωτικού πράκτορα	211
Εικόνα 369. View αναζήτησης χρήστη-ταξιδιωτικού πράκτορα	211
Εικόνα 370. View ελέγχου εγκυρότητας χρήστη-ταξιδιωτικού πράκτορα	212
Εικόνα 371. View επεξεργασίας χρήστη-διαχειριστή	212
Εικόνα 372. View αποθήκευσης χρήστη-διαχειριστή	213
Εικόνα 373. View αναζήτησης χρήστη-διαχειριστή.....	213
Εικόνα 374. View ελέγχου εγκυρότητας χρήστη-διαχειριστή	213
Εικόνα 375. View επεξεργασίας δρομολογίου.....	214
Εικόνα 376. View αποθήκευσης δρομολογίου.....	214
Εικόνα 377. View αναζήτησης δρομολογίου	214
Εικόνα 378. View διαγραφής δρομολογίου	215
Εικόνα 379. View ελέγχου εγκυρότητας δρομολογίου.....	215
Εικόνα 380. View επεξεργασίας μεταφοράς.....	216
Εικόνα 381. View αποθήκευσης μεταφοράς.....	216
Εικόνα 382. View ελέγχου εγκυρότητας μεταφοράς.....	217
Εικόνα 383. View αναζήτησης μεταφοράς.....	217
Εικόνα 384. View διαγραφής μεταφοράς	217
Εικόνα 385. View εισαγωγής τύπου θέσεων αεροπλάνου στη μεταφορά	217
Εικόνα 386. View επεξεργασίας κράτησης χρήστη-ταξιδιωτικού πράκτορα	218
Εικόνα 387. View εισαγωγής μεταφορών στη κράτηση του χρήστη-ταξιδιωτικού πράκτορα	218
Εικόνα 388. View αποθήκευσης κράτησης χρήστη-ταξιδιωτικού πράκτορα	219
Εικόνα 389. View αναζήτησης κράτησης χρήστη-ταξιδιωτικού πράκτορα.....	219
Εικόνα 390. View διαγραφής κράτησης χρήστη-ταξιδιωτικού πράκτορα.....	219
Εικόνα 391. View ελέγχου εγκυρότητας κράτησης χρήστη-ταξιδιωτικού πράκτορα	219
Εικόνα 392. View επεξεργασίας κράτησης χρήστη-πελάτη	220
Εικόνα 393. View αποθήκευσης κράτησης χρήστη-πελάτη	220
Εικόνα 394. View αναζήτησης κράτησης χρήστη-πελάτη.....	220
Εικόνα 395. View διαμόρφωσης μηνύματος email του εισιτηρίου	221
Εικόνα 396. View μετασχηματισμού κράτησης σε εισιτήριο	221
Εικόνα 397. View έκδοσης εισιτηρίου από τον ταξιδιωτικό πράκτορα	221
Εικόνα 398. View έκδοσης εισιτηρίου από τον πελάτη.....	222
Εικόνα 399. View ταμπλό πελάτη	222
Εικόνα 400. View σελίδας συγκεντρωτικής αναφοράς πτήσεων	222
Εικόνα 401. View εισαγωγής συγκεντρωτικής αναφοράς πτήσεων.....	223
Εικόνα 402. View σελίδας καλωσορίσματος χρήστη	223
Εικόνα 403. Εκτέλεση εφαρμογής Βήμα 1ο & 2ο.....	224
Εικόνα 404. Εκτέλεση εφαρμογής Βήμα 3ο	225
Εικόνα 405. Εκτέλεση εφαρμογής Βήμα 4ο	225
Εικόνα 406. Εκτέλεση εφαρμογής Βήμα 5ο	226
Εικόνα 407. Εκτέλεση εφαρμογής Βήμα 6ο	227
Εικόνα 408. Είσοδος χρήστη.....	228
Εικόνα 409. Έλεγχος κενών στοιχείων εισόδου χρήστη.....	228
Εικόνα 410. Έλεγχος εγκυρότητας στοιχείων εισόδου χρήστη.....	228
Εικόνα 411. Εγγραφή χρήστη-πελάτη	229
Εικόνα 412. Αρχική σελίδα διαχειριστή & λογότυπο-κουμπί μετάβασης στην αρχική σελίδα	229
Εικόνα 413. Λίστα πιλότων.....	230
Εικόνα 414. Προβολή στοιχείων πιλότου	230
Εικόνα 415. Επεξεργασία στοιχείων πιλότου.....	231
Εικόνα 416. Επιβεβαίωση διαγραφής πιλότου	231
Εικόνα 417. Λίστα αεροσυνοδών	231
Εικόνα 418. Προβολή στοιχείων αεροσυνοδού.....	232
Εικόνα 419. Επεξεργασία στοιχείων αεροσυνοδού	232

Εικόνα 420. Επιβεβαίωση διαγραφής αεροσυνοδού	232
Εικόνα 421. Λίστα αεροπλάνων	233
Εικόνα 422. Προβολή στοιχείων αεροπλάνου	233
Εικόνα 423. Επεξεργασία στοιχείων αεροπλάνου	233
Εικόνα 424. Επιβεβαίωση διαγραφής αεροπλάνου	234
Εικόνα 425. Λίστα ταξιδιωτικών πρακτόρων	234
Εικόνα 426. Προβολή στοιχείων ταξιδιωτικού πράκτορα	234
Εικόνα 427. Επεξεργασία στοιχείων ταξιδιωτικού πράκτορα.....	235
Εικόνα 428. Επιβεβαίωση διαγραφής ταξιδιωτικού πράκτορα	235
Εικόνα 429. Λίστα διοικητικών υπαλλήλων	235
Εικόνα 430. Προβολή στοιχείων διοικητικού υπαλλήλου	236
Εικόνα 431. Επεξεργασία στοιχείων διοικητικού υπαλλήλου	236
Εικόνα 432. Επιβεβαίωση διαγραφής διοικητικού υπαλλήλου	236
Εικόνα 433. Λίστα χρηστών-πελατών	237
Εικόνα 434. Προβολή στοιχείων χρήστη-πελάτη.....	237
Εικόνα 435. Επεξεργασία στοιχείων χρήστη-πελάτη	238
Εικόνα 436. Επιβεβαίωση διαγραφής χρήστη-πελάτη.....	238
Εικόνα 437. Λίστα χρηστών-ταξιδιωτικών πρακτόρων.....	238
Εικόνα 438. Προβολή στοιχείων χρήστη-ταξιδιωτικού πράκτορα.....	239
Εικόνα 439. Επεξεργασία στοιχείων χρήστη-ταξιδιωτικού πράκτορα	239
Εικόνα 440. Επιβεβαίωση διαγραφής χρήστη-ταξιδιωτικού πράκτορα.....	239
Εικόνα 441. Λίστα χρηστών-διαχειριστών	240
Εικόνα 442. Προβολή στοιχείων χρήστη-διαχειριστή	240
Εικόνα 443. Επεξεργασία στοιχείων χρήστη-διαχειριστή	241
Εικόνα 444. Επιβεβαίωση διαγραφής χρήστη-διαχειριστή	241
Εικόνα 445. Λίστα δρομολογίων.....	241
Εικόνα 446. Προβολή στοιχείων δρομολογίου	242
Εικόνα 447. Επεξεργασία στοιχείων δρομολογίου.....	242
Εικόνα 448. Επιβεβαίωση διαγραφής δρομολογίου	242
Εικόνα 449. Λίστα μεταφορών.....	243
Εικόνα 450. Προβολή στοιχείων μεταφοράς	243
Εικόνα 451. Επεξεργασία στοιχείων μεταφοράς.....	243
Εικόνα 452. Επιβεβαίωση διαγραφής μεταφοράς	243
Εικόνα 453. Λίστα κρατήσεων	244
Εικόνα 454. Προβολή στοιχείων κράτησης	244
Εικόνα 455. Επεξεργασία στοιχείων κράτησης.....	244
Εικόνα 456. Επιβεβαίωση διαγραφής κράτησης	244
Εικόνα 457. Χρονοπρογραμματισμένη διεργασία (scheduled task).....	245
Εικόνα 458. Λίστα πελατών	245
Εικόνα 459. Προβολή στοιχείων πελάτη	246
Εικόνα 460. Επεξεργασία στοιχείων πελάτη.....	246
Εικόνα 461. Επιβεβαίωση διαγραφής πελάτη	246
Εικόνα 462. Δημιουργία νέου πιλότου	247
Εικόνα 463. Έλεγχος εγκυρότητας στοιχείων πιλότου	247
Εικόνα 464. Έλεγχος ήδη καταχωρημένου πιλότου	248
Εικόνα 465. Επιβεβαίωση διαγραφής πιλότου	248
Εικόνα 466. Συσχέτιση πιλότου με άλλες εγγραφές.....	249
Εικόνα 467. Δημιουργία νέου αεροσυνοδού.....	250
Εικόνα 468. Έλεγχος εγκυρότητας στοιχείων αεροσυνοδού	250
Εικόνα 469. Έλεγχος ήδη καταχωρημένου αεροσυνοδού	250
Εικόνα 470. Επιβεβαίωση διαγραφής αεροσυνοδού	251
Εικόνα 471. Συσχέτιση αεροσυνοδού με άλλες εγγραφές	251
Εικόνα 472. Δημιουργία νέου αεροπλάνου.....	252
Εικόνα 473. Έλεγχος εγκυρότητας στοιχείων αεροπλάνου	252
Εικόνα 474. Έλεγχος ήδη καταχωρημένου αεροπλάνου	252

Εικόνα 475. Επιβεβαίωση διαγραφής αεροπλάνου	253
Εικόνα 476. Συσχέτιση αεροπλάνου με άλλες εγγραφές	253
Εικόνα 477. Δημιουργία νέου ταξιδιωτικού πράκτορα	254
Εικόνα 478. Έλεγχος εγκυρότητας στοιχείων ταξιδιωτικού πράκτορα	254
Εικόνα 479. Έλεγχος ήδη καταχωρημένου ταξιδιωτικού πράκτορα	254
Εικόνα 480. Επιβεβαίωση διαγραφής ταξιδιωτικού πράκτορα	255
Εικόνα 481. Δημιουργία νέου διοικητικού υπαλλήλου	256
Εικόνα 482. Έλεγχος εγκυρότητας στοιχείων διοικητικού υπαλλήλου	256
Εικόνα 483. Έλεγχος ήδη καταχωρημένου διοικητικού υπαλλήλου	256
Εικόνα 484. Επιβεβαίωση διαγραφής διοικητικού υπαλλήλου	257
Εικόνα 485. Συσχέτιση διοικητικού υπαλλήλου με άλλες εγγραφές.....	257
Εικόνα 486. Δημιουργία νέου χρήστη-πελάτη	258
Εικόνα 487. Έλεγχος εγκυρότητας στοιχείων χρήστη-πελάτη.....	258
Εικόνα 488. Έλεγχος ήδη καταχωρημένου χρήστη-πελάτη.....	259
Εικόνα 489. Έλεγχος εγκυρότητας κωδικού πρόσβασης.....	259
Εικόνα 490. Επιβεβαίωση διαγραφής χρήστη-πελάτη.....	259
Εικόνα 491. Δημιουργία νέου χρήστη-ταξιδιωτικού πράκτορα.....	260
Εικόνα 492. Έλεγχος εγκυρότητας στοιχείων χρήστη-ταξιδιωτικού πράκτορα.....	260
Εικόνα 493. Έλεγχος ήδη καταχωρημένου χρήστη-ταξιδιωτικού πράκτορα.....	261
Εικόνα 494. Έλεγχος εγκυρότητας κωδικού πρόσβασης.....	261
Εικόνα 495. Επιβεβαίωση διαγραφής χρήστη-ταξιδιωτικού πράκτορα.....	261
Εικόνα 496. Δημιουργία νέου χρήστη-διαχειριστή	262
Εικόνα 497. Έλεγχος εγκυρότητας στοιχείων χρήστη-διαχειριστή	263
Εικόνα 498. Έλεγχος ήδη καταχωρημένου χρήστη-διαχειριστή	263
Εικόνα 499. Έλεγχος εγκυρότητας κωδικού πρόσβασης.....	263
Εικόνα 500. Επιβεβαίωση διαγραφής χρήστη-διαχειριστή	264
Εικόνα 501. Δημιουργία νέου δρομολογίου.....	265
Εικόνα 502. Έλεγχος εγκυρότητας στοιχείων δρομολογίου	265
Εικόνα 503. Έλεγχος ήδη καταχωρημένου δρομολογίου	265
Εικόνα 504. Έλεγχος ελάχιστης διαφοράς χρόνου πραγματοποίησης δρομολογίου.....	266
Εικόνα 505. Έλεγχος μεταγενέστερης ώρας άφιξης δρομολογίου	266
Εικόνα 506. Επιβεβαίωση διαγραφής δρομολογίου	266
Εικόνα 507. Συσχέτιση δρομολογίου με άλλες εγγραφές	267
Εικόνα 508. Δημιουργία νέας μεταφοράς.....	267
Εικόνα 509. Έλεγχος εγκυρότητας στοιχείων μεταφοράς	268
Εικόνα 510. Έλεγχος χρονικής επικάλυψης μεταφοράς με άλλες.....	268
Εικόνα 511. Επιβεβαίωση διαγραφής μεταφοράς	268
Εικόνα 512. Δημιουργία νέας κράτησης από τον ταξιδιωτικό πράκτορα.....	269
Εικόνα 513. Δημιουργία νέας κράτησης από τον πελάτη	269
Εικόνα 514. Έλεγχος εγκυρότητας στοιχείων κράτησης από τον ταξιδιωτικό πράκτορα	270
Εικόνα 515. Έλεγχος εγκυρότητας στοιχείων κράτησης από τον πελάτη	270
Εικόνα 516. Έλεγχος εγκυρότητας ημερομηνίας κράτησης	270
Εικόνα 517. Μήνυμα μη διαθέσιμων μεταφορών	271
Εικόνα 518. Μήνυμα διαθέσιμων μεταφορών	271
Εικόνα 519. Έλεγχος επιλογής μεταφοράς	272
Εικόνα 520. Έλεγχος ήδη καταχωρημένης κράτησης	272
Εικόνα 521. Επιβεβαίωση διαγραφής κράτησης.....	272
Εικόνα 522. Δημιουργία νέου πελάτη.....	273
Εικόνα 523. Έλεγχος εγκυρότητας στοιχείων πελάτη	273
Εικόνα 524. Έλεγχος ήδη καταχωρημένου πελάτη	273
Εικόνα 525. Επιβεβαίωση διαγραφής πελάτη	274
Εικόνα 526. Μήνυμα επιτυχούς έκδοσης εισιτηρίου.....	274
Εικόνα 527. Προεπισκόπηση εκδιδόμενου εισιτηρίου	275
Εικόνα 528. E-mail επιβεβαίωσης έκδοσης εισιτηρίου	275
Εικόνα 529. Ταμπλό πελάτη.....	276

Εικόνα 530. Άμεση επεξεργασία στοιχείων χρήστη-διαχειριστή.....	276
Εικόνα 531. Άμεση επεξεργασία στοιχείων χρήστη-ταξιδιωτικού πράκτορα	276
Εικόνα 532. Άμεση επεξεργασία στοιχείων χρήστη-πελάτη.....	277
Εικόνα 533. Γραμμή φίλτρων συγκεντρωτικής αναφοράς.....	277
Εικόνα 534. Έλεγχος εγκυρότητας ημερομηνιών.....	277
Εικόνα 535. Λίστα πληροτήτων διαθέσιμων δρομολογίων	278

Λίστα Πινάκων

Πίνακας 1. Στάδια δημιουργίας της εφαρμογής.....	18
--	----

1 Εισαγωγή

1.1 Περίληψη

Στη παρούσα πτυχιακή εργασία, έχουμε μια εφαρμογή μπαίνοντας σε συγκεκριμένο URL της ιστοσελίδας (εδώ πρόκειται για το URL: <http://127.0.0.1:8000>). Η εφαρμογή μας επιτρέπει να δημιουργούμε κρατήσεις και εκδόσεις εισιτηρίων της αεροπορικής εταιρείας “Rodos Flights” ως πελάτες ή ως ταξιδιωτικοί πράκτορες για λογαριασμό των πελατών. Επίσης μας επιτρέπει να δημιουργούμε, τροποποιούμε και διαγράφουμε λίστες με τα στοιχεία όλων των εργαζομένων, των ταξιδιωτικών πρακτόρων, των αεροπλάνων, των μεταφορών και των δρομολογίων της εταιρείας ως διαχειριστές.

1.2 Σκοπός και στόχοι της εργασίας

Σκοπός της παρούσας πτυχιακής εργασίας, είναι να εξοικειωθούμε με το εικονικό περιβάλλον των προγραμμάτων Visual Studio Code, καθώς και του Microsoft SQL Server Management Studio, για τη δημιουργία της εφαρμογής χρησιμοποιώντας ιστοσελίδες και βάση δεδομένων με τα αντίστοιχα προγράμματα.

Στόχος της εργασίας, είναι να διευκολύνουμε τους χρήστες της εφαρμογής, να κλείνουν τα αεροπορικά ταξίδια τους και να τροποποιούν τα δρομολόγια των αεροπλάνων με μεγάλη ευκολία, ταχύτητα και ευελιξία.

1.3 Δομή εργασίας

Η δομή της εργασίας, αποτελείται από τη βασική μεθοδολογία υλοποίησης, με τη χρήση κώδικα των γλωσσών προγραμματισμού SQL, Python και Javascript. Καθώς και με τη διεπαφή χρήστη (server) και πελάτη (client) μέσω της Django Framework.

2 Μεθοδολογία Υλοποίησης

Σε αυτή τη παράγραφο, περιγράφεται συνοπτικά η μεθοδολογία υλοποίησης της παρούσας πτυχιακής. Δηλαδή ποια προγράμματα χρησιμοποιήθηκαν για την υλοποίηση των ιστοσελίδων και της βάσης δεδομένων, καθώς και οι γλώσσες προγραμματισμού για τη χρήση κατάλληλων αλγορίθμων μεταφοράς δεδομένων και επικοινωνίας μεταξύ πελάτη (client) και χρήστη (server).

2.1 Μέθοδος Ανάλυσης & Ανάπτυξης

Η υλοποίηση ολόκληρης της εργασίας, βασίζεται σε δύο βασικά προγράμματα. Πρόκειται για το Microsoft SQL Server Management Studio 18 και το Visual Studio Code.

Αρχικά, χρησιμοποιούμε το πρόγραμμα Microsoft SQL Server Management Studio 18, για να δημιουργήσουμε τη βάση δεδομένων με τις εγγραφές. Καθώς και τις αποθηκευμένες διαδικασίες (stored procedures), που περιέχουν κώδικα γραμμένο σε γλώσσα SQL, με τις απαραίτητες εντολές για την εκτέλεση συγκεκριμένων ενεργειών (για παράδειγμα τη ανανέωση των στοιχείων ενός πελάτη).

Έπειτα, χρησιμοποιούμε το πρόγραμμα Visual Studio Code, για να δημιουργήσουμε τις ιστοσελίδες της εφαρμογής χρησιμοποιώντας τη γλώσσα προγραμματισμού HTML. Επειδή χρησιμοποιεί και τη γλώσσα προγραμματισμού Python, είναι απαραίτητο να έχουμε σ' αυτό το πρόγραμμα, εγκατεστημένο το εργαλείο Python. Επίσης, δημιουργούμε μέσω του Visual Studio Code, τις κατάλληλες συναρτήσεις χρησιμοποιώντας τη γλώσσα προγραμματισμού JavaScript και Django, για τη γραφική διεπαφή μεταξύ χρήστη (server) και πελάτη (client).

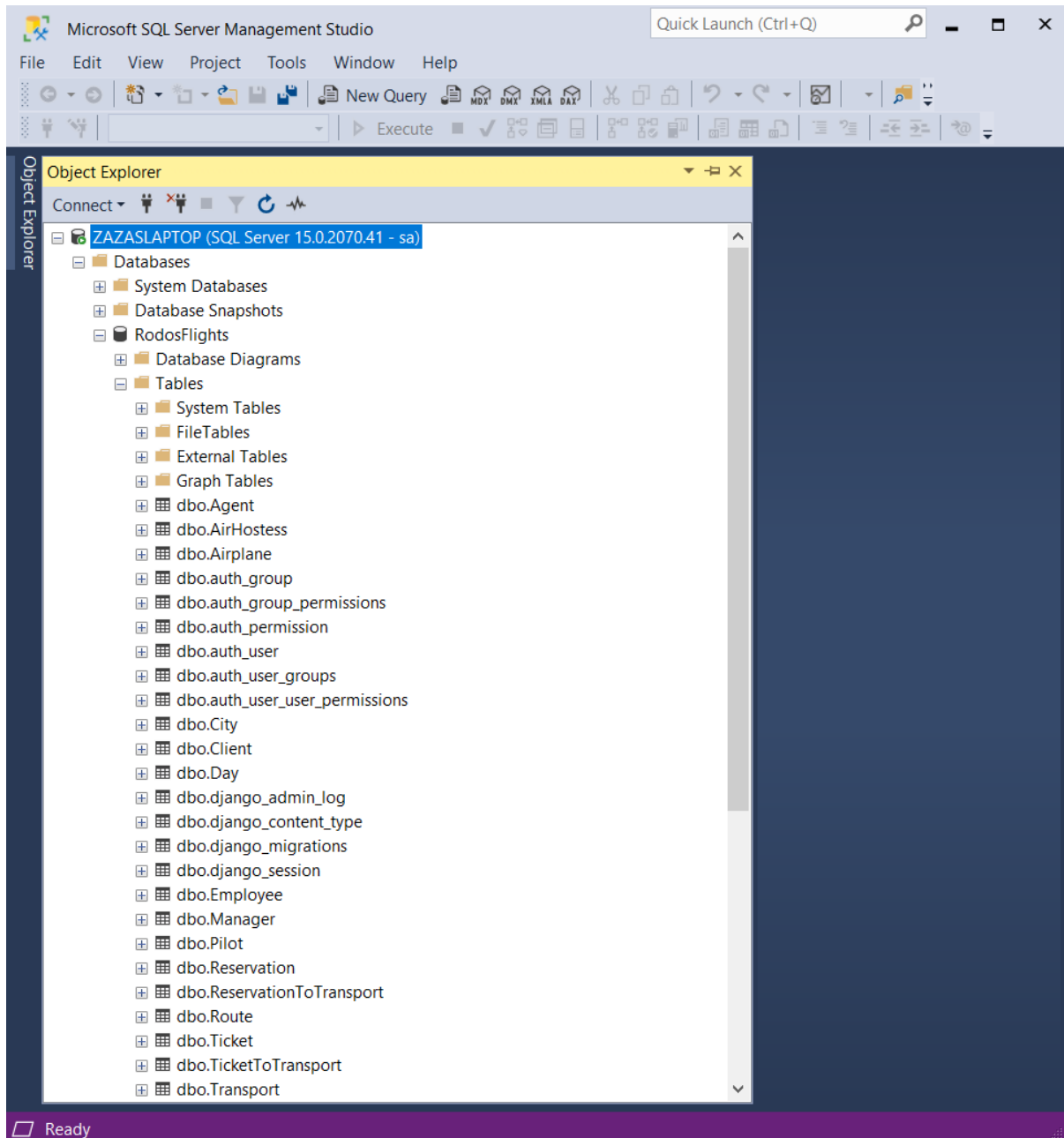
2.2 Βασικά Προγράμματα Δημιουργίας Βάσης Δεδομένων & Ιστοσελίδων

2.2.1 Εισαγωγή στο Microsoft SQL Server Management Studio

Το SQL Server Management Studio (SSMS), είναι μια εφαρμογή λογισμικού που ξεκίνησε για πρώτη φορά με τον Microsoft SQL Server 2005[[Παραπομπές](#)] και χρησιμοποιείται για τη διαμόρφωση, διαχείριση και διαχείριση όλων των στοιχείων του Microsoft SQL Server. Είναι ο διάδοχος του Enterprise Manager στο SQL 2000 ή πριν. Το εργαλείο περιλαμβάνει τόσο προγράμματα επεξεργασίας σεναρίων όσο και εργαλεία γραφικών που λειτουργούν με αντικείμενα και δυνατότητες του διακομιστή.

Ένα κεντρικό χαρακτηριστικό του SSMS είναι η εξερεύνηση αντικειμένων, η οποία επιτρέπει στο χρήστη να περιηγηθεί, να επιλέξει και να ενεργήσει σε οποιοδήποτε από τα αντικείμενα του διακομιστή. Η Microsoft ενσωμάτωσε τη συμβατότητα προς τα πίσω για παλαιότερες εκδόσεις του SQL Server, επιτρέποντας έτσι σε μια νεότερη έκδοση του SSMS να συνδεθεί με παλαιότερες εκδόσεις παρουσιών του SQL Server.

Ξεκινώντας από την έκδοση 11, η εφαρμογή βασίστηκε στο κέλυφος του Visual Studio 2010, χρησιμοποιώντας Windows Presentation Foundation (WPF)[[Παραπομπές](#)] για τη διεπαφή χρήστη. Οι εκδόσεις 18 και μετά βασίζονται στο απομονωμένο κέλυφος του Visual Studio 2017.



Εικόνα 1. Εικονικό περιβάλλον του Microsoft SQL Server Management Studio

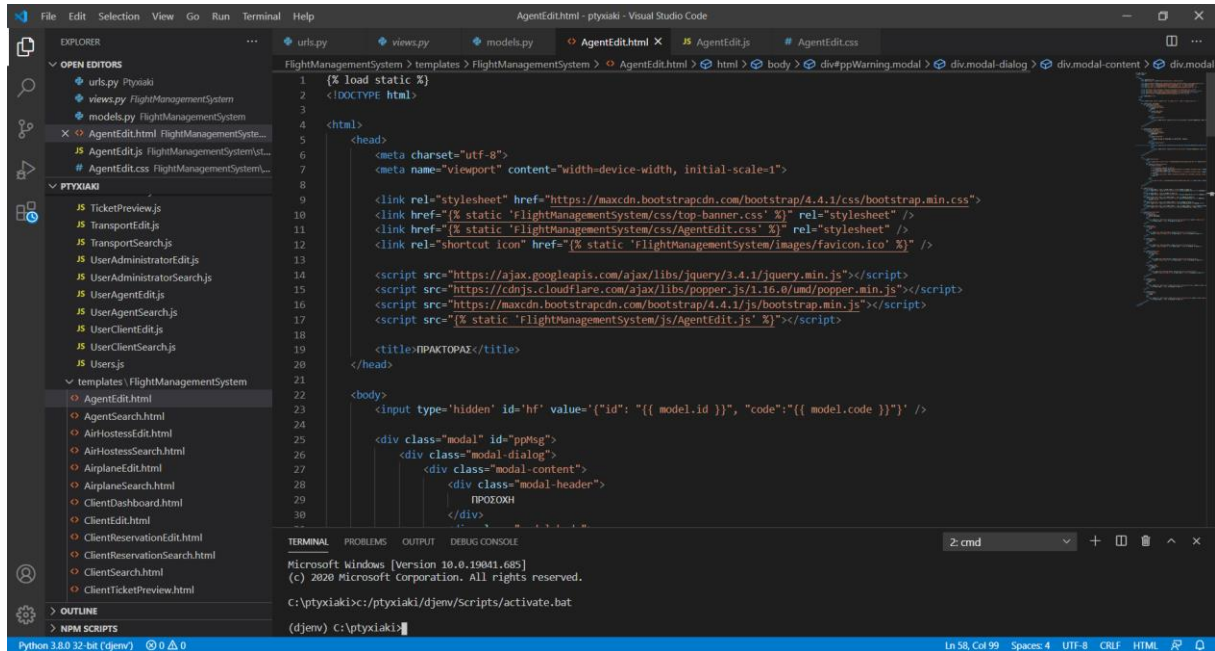
2.2.2 Εισαγωγή στο Visual Studio Code

Το Visual Studio Code, είναι ένας δωρεάν επεξεργαστής πηγαίου κώδικα, πού δημιουργήθηκε από τη Microsoft για Windows, Linux και macOS. Οι λειτουργίες περιλαμβάνουν υποστήριξη για εντοπισμό σφαλμάτων, επισήμανση σύνταξης, έξυπνη ολοκλήρωση κώδικα, αποσπάσματα, αναδιαμόρφωση κωδικών και ενσωματωμένο Git[[Παραπομπές](#)]. Οι χρήστες μπορούν να αλλάξουν το θέμα, τις συντομεύσεις πληκτρολογίου, τις προτιμήσεις και να εγκαταστήσουν επεκτάσεις που προσθέτουν επιπλέον λειτουργικότητα.

Η Microsoft έχει κυκλοφορήσει τον πηγαίο κώδικα του Visual Studio Code στο αποθετήριο VSCoDe του GitHub.com[[Παραπομπές](#)], υπό τη επιτρεπτή άδεια του Ινστιτούτου Τεχνολογίας της Μασαχουσέτης (MIT), ενώ οι μεταγλωττισμένες εκδόσεις είναι

δωρεάν λογισμικό.

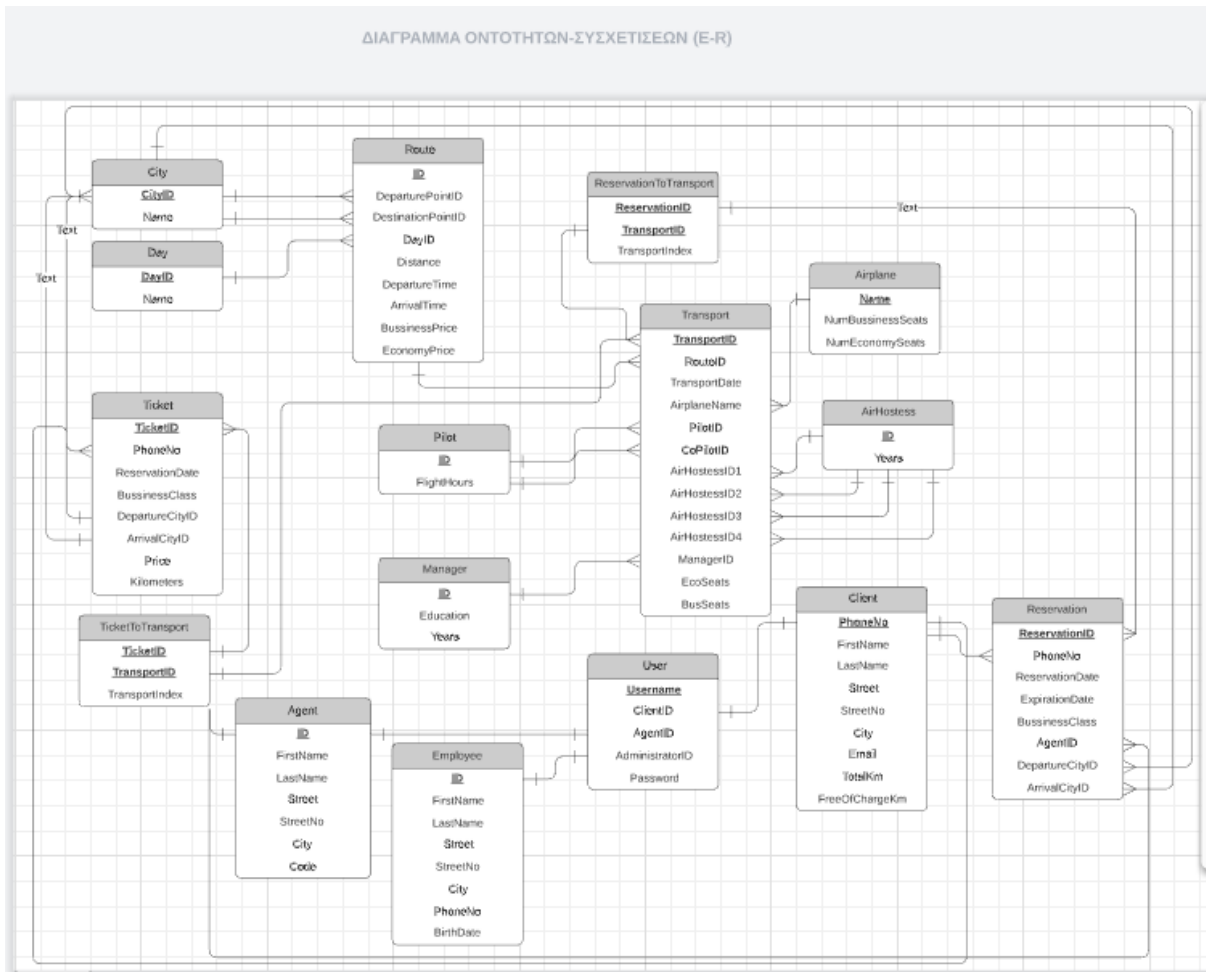
Σύμφωνα με την έρευνα προγραμματιστών του Stack Overflow 2019[[Παραπομπές](#)], το Visual Studio Code κατατάχθηκε ως το πιο δημοφιλές εργαλείο περιβάλλοντος προγραμματιστή, με το 50,7% των 87.317 ερωτηθέντων να αναφέρουν ότι το χρησιμοποιούν.



Εικόνα 2. Εικονικό περιβάλλον του Visual Studio Code

2.3 Ροή εργασίας

Η παρακάτω εικόνα, μας παρουσιάζει ένα διάγραμμα, το οποίο περιγράφει τη συσχέτιση μεταξύ των οντοτήτων της βάσης δεδομένων της εργασίας:

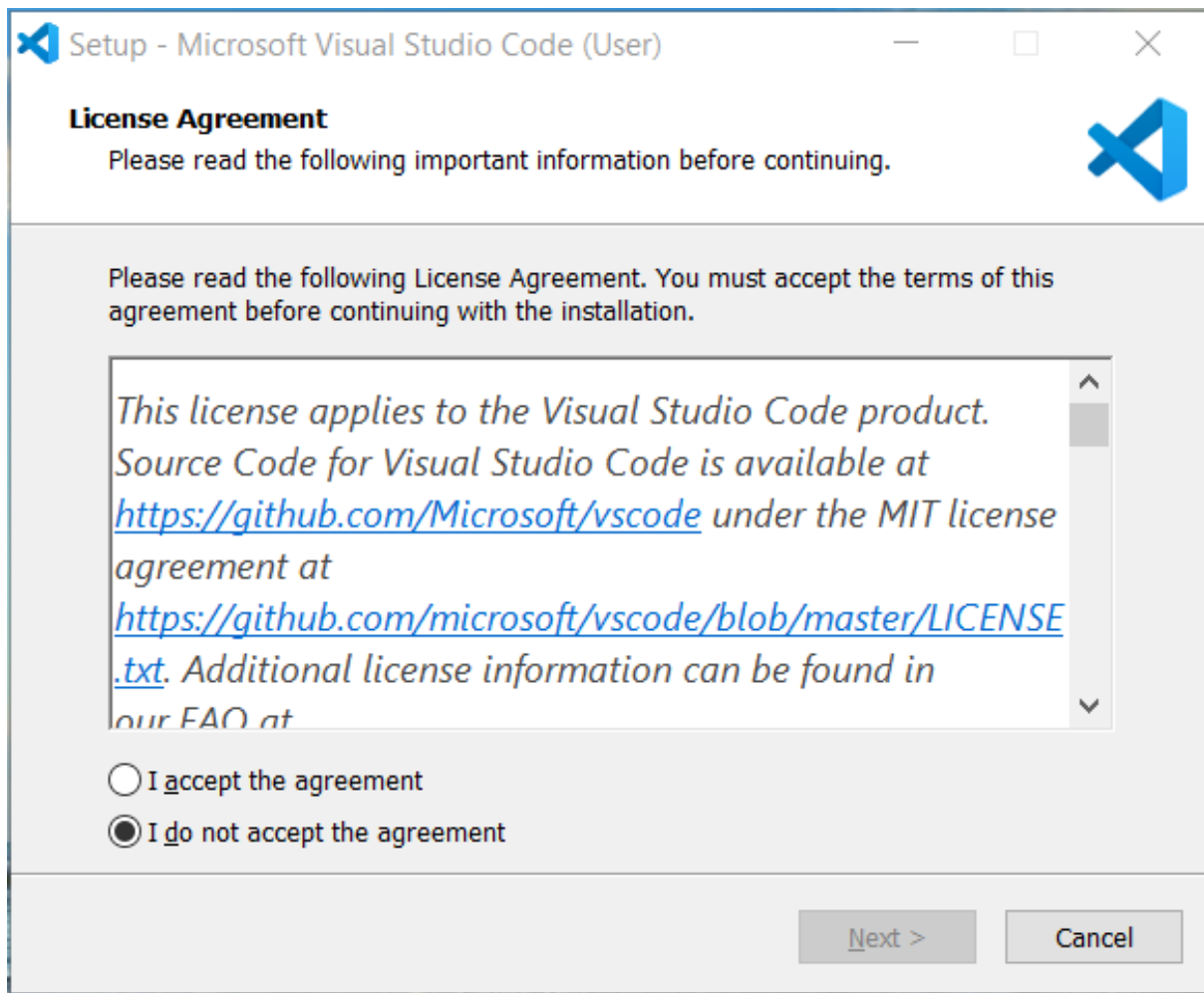


Εικόνα 3. Διάγραμμα οντοτήτων-συσχετίσεων (E-R)

Το εν λόγω διάγραμμα, λέγεται διάγραμμα οντοτήτων-συσχετίσεων (E-R)[[Παραπομπές](#)] και δημιουργήθηκε με τη βοήθεια της διαδικτυακής εφαρμογής Lucidchart. Η εφαρμογή βοηθάει τους χρήστες να δημιουργήσουν διαγράμματα, βασισμένα σε οντότητες που υπάρχουν σε πολλά λογισμικά και είναι εύκολα στη διαχείρισή τους. Εδώ συγκεκριμένα, δημιουργήθηκε το διάγραμμα, σύμφωνα με τις συσχετίσεις των οντοτήτων της βάσης δεδομένων “Rodos Flights”.

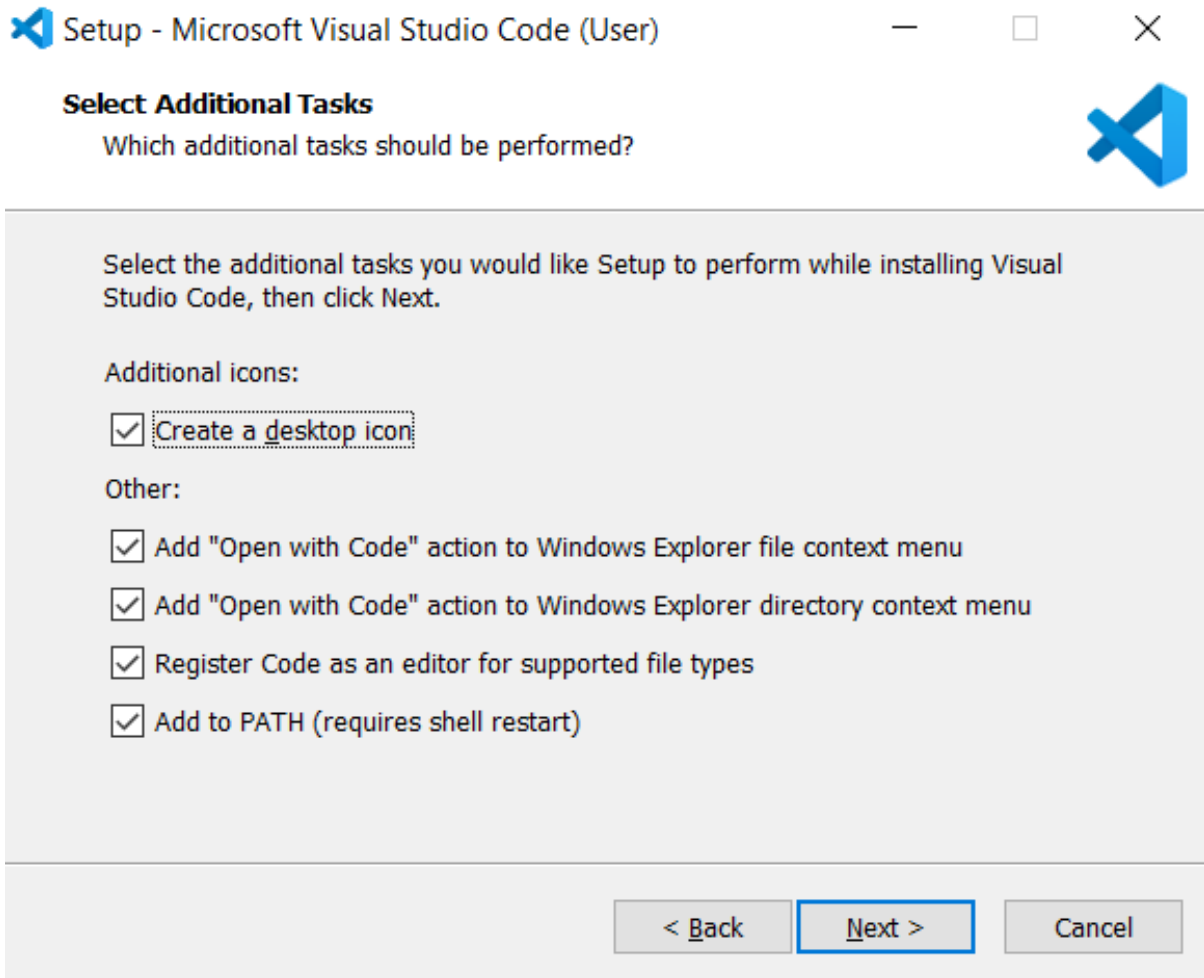
2.3.1 Περιγραφή ροής εργασίας

Αρχικά, πρέπει να εγκαταστήσουμε το συντάκτη (editor) Visual Studio Code.



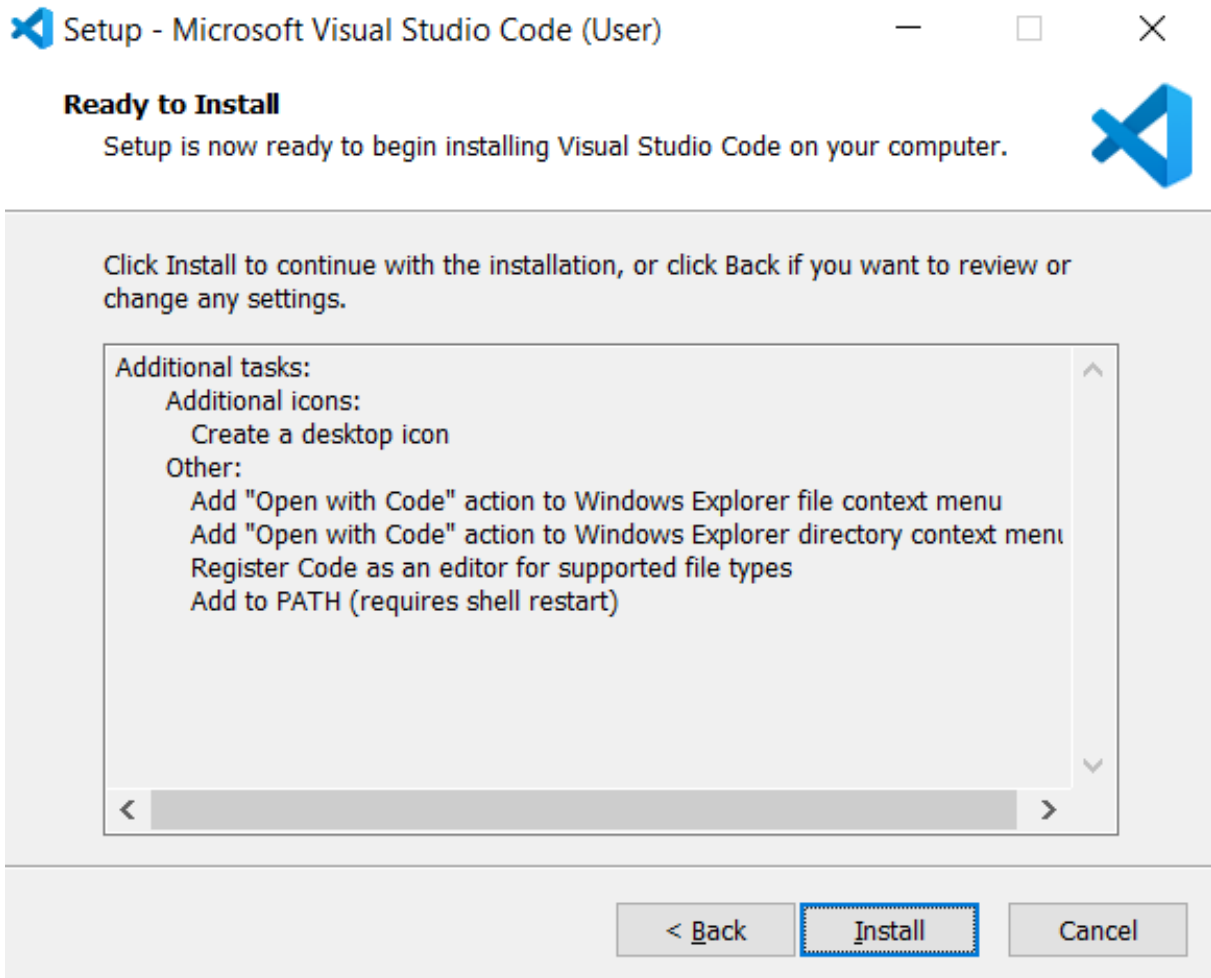
Εικόνα 4. Εγκατάσταση Visual Studio Code Βήμα 1^ο

Στο πρώτο βήμα, πρέπει να συμφωνήσουμε με τους κανόνες χρήσης της εφαρμογής, επιλέγοντας την επιλογή "I accept the agreement" και στη συνέχεια πατώντας το κουμπί "Next".



Εικόνα 5. Εγκατάσταση Visual Studio Code Βήμα 2^ο

Στο δεύτερο βήμα, επιλέγουμε όλα τα πεδία για τη δημιουργία συντόμευσης του προγράμματος στην επιφάνεια εργασίας, τη προσθήκη στη μεταβλητή περιβάλλοντος PATH, τη διαδρομή του συντάκτη Visual Studio Code, ώστε να μπορεί να εκτελείται από τον φλοιό των Windows (command prompt) και την επιλογή του συντάκτη αυτού ως προεπιλεγμένου για υποστηριζόμενα αρχεία (για παράδειγμα: .py, .html, .js, .css, .cs και τα λοιπά).



Εικόνα 6. Εγκατάσταση Visual Studio Code Βήμα 3^ο

Τέλος, πατάμε το κουμπί “Install” για εκκίνηση της εγκατάστασης.



Εικόνα 7. Εγκατάσταση Visual Studio Code Βήμα 4^ο

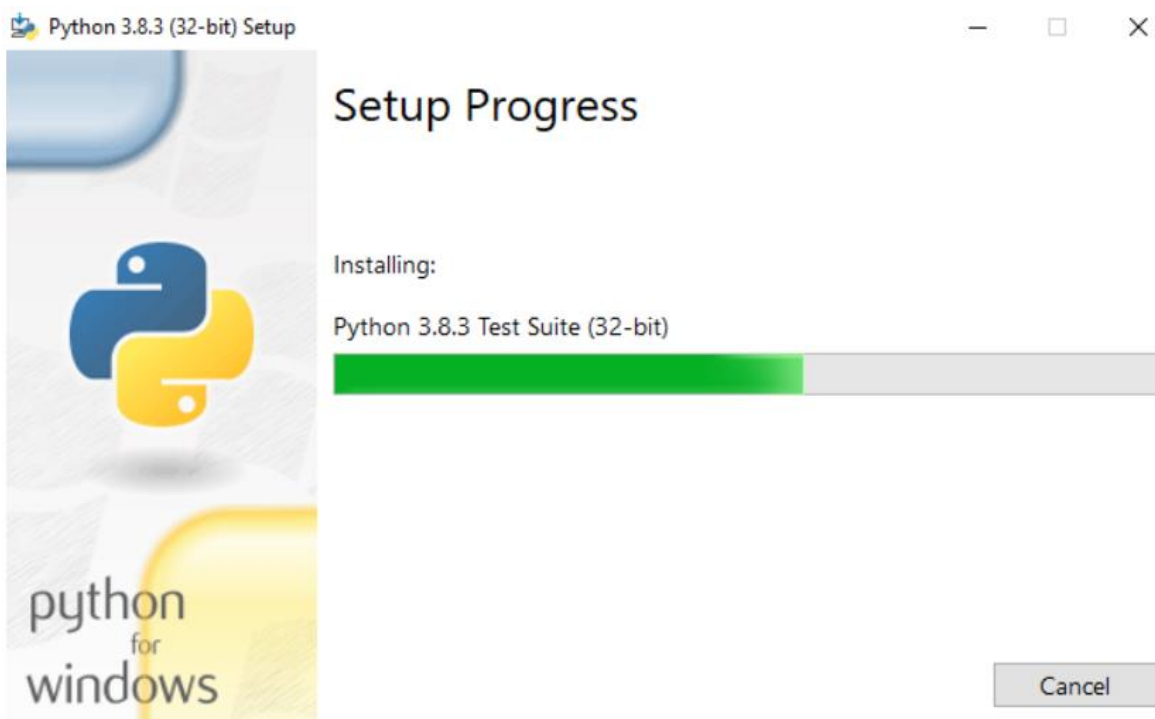
Όταν ολοκληρωθεί η εγκατάσταση, μπορούμε πλέον να κλείσουμε το παράθυρο της εγκατάστασης ή να κάνουμε εκκίνηση του προγράμματος, κάνοντας κλικ στην επιλογή “Launch Visual Studio Code” και έπειτα, πατάμε το κουμπί “Finish”.

Αφού ολοκληρώσουμε την εγκατάσταση του Visual Studio Code, προχωράμε στην εγκατάσταση του εργαλείου της Python.



Εικόνα 8. Εγκατάσταση Python Βήμα 1^ο

Προσθέτουμε τη διαδρομή (PATH) του διερμηνευτή της Python, στη μεταβλητή περιβάλλοντος PATH (που χρησιμοποιείται για τη γραμμή εντολών). Στη συνέχεια, πατάμε το κουμπί “Install now”.



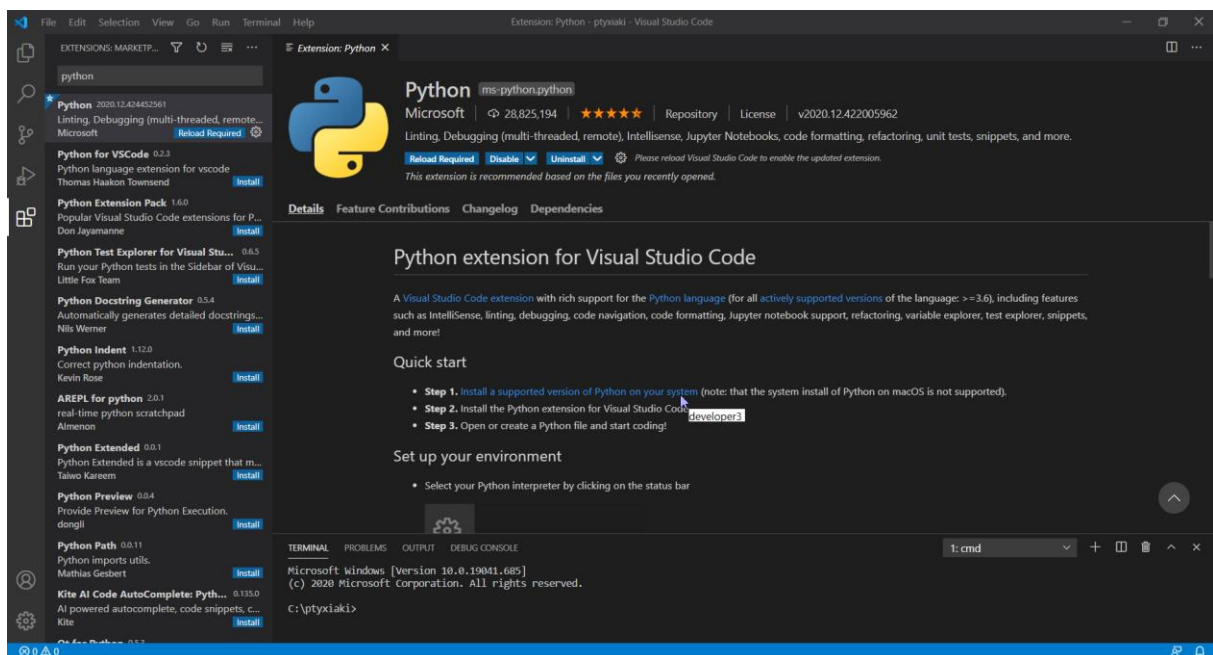
Εικόνα 9. Εγκατάσταση Python Βήμα 2^ο

Τώρα θα εγκαταστήσει όλα τα απαραίτητα αρχεία για την εκτέλεση του Python στον υπολογιστή. Περιμένουμε να ολοκληρωθεί η εγκατάσταση.



Εικόνα 10. Εγκατάσταση Python Βήμα 3^ο

Μόλις ολοκληρωθεί η εγκατάσταση, στη συνέχεια πατάμε το κουμπί “close”. Στη συνέχεια, θα πρέπει να εγκαταστήσουμε την επέκταση (extension) της Python στο Visual Studio Code.



Εικόνα 11. Εγκατάσταση Python στο Visual Studio Code Βήμα 1^ο

Δημιουργούμε ένα εικονικό περιβάλλον εγκατάστασης του framework Django ακολουθώντας τα παρακάτω βήματα:

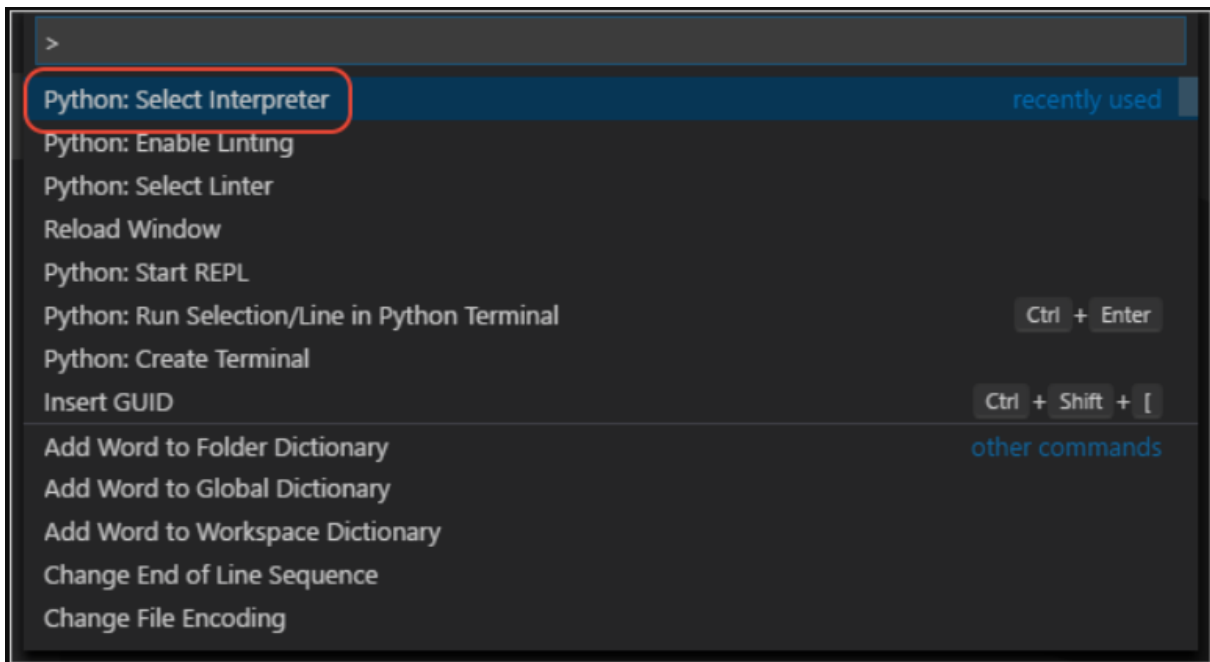
Βήμα 1^ο: Δημιουργούμε ένα φάκελο στο σκληρό δίσκο του υπολογιστή για την εργασία μας.

Βήμα 2^ο: Μπαίνουμε στο φάκελο με την εντολή cd (change directory).

Βήμα 3^ο: Εγκαθιστούμε το εικονικό περιβάλλον, με όνομα “env”, κάνοντας χρήση της εντολής: `python -m venv env`.

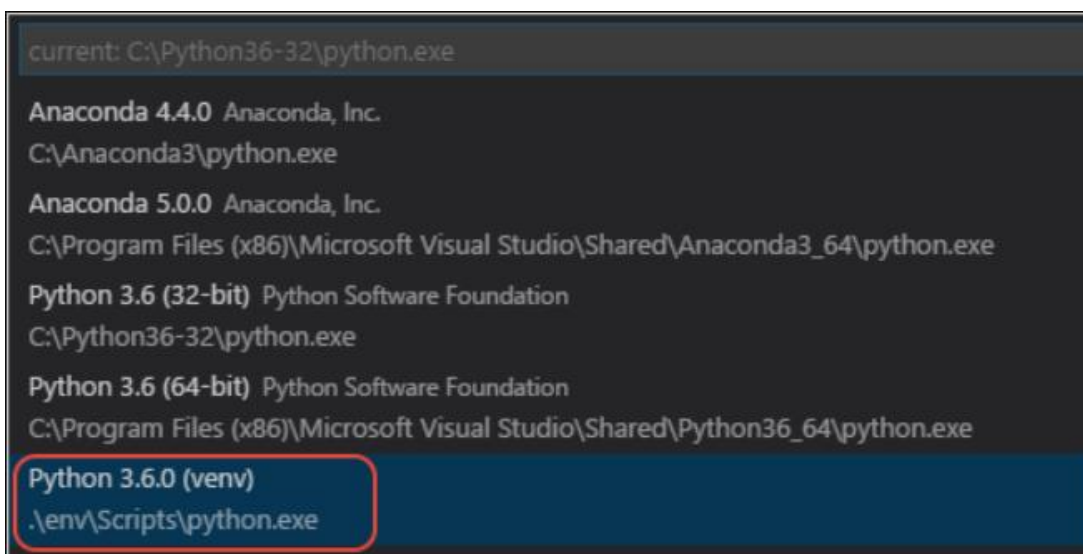
Βήμα 4^ο: Ανοίγουμε το φάκελο της εργασίας μας από το Visual Studio Code, πηγαίνοντας στο File > Open folder.

Βήμα 5^ο: Επιλέγουμε τον διερμηνευτή της Python, πηγαίνοντας στο View > Command Palette και στη συνέχεια επιλέγουμε “Python: Select Interpreter”.



Εικόνα 12. Εγκατάσταση Python στο Visual Studio Code Βήμα 2^ο

Τέλος, επιλέγουμε τον διερμηνευτή της Python από τη λίστα των διερμηνευτών.



Εικόνα 13. Εγκατάσταση Python στο Visual Studio Code Βήμα 3^ο

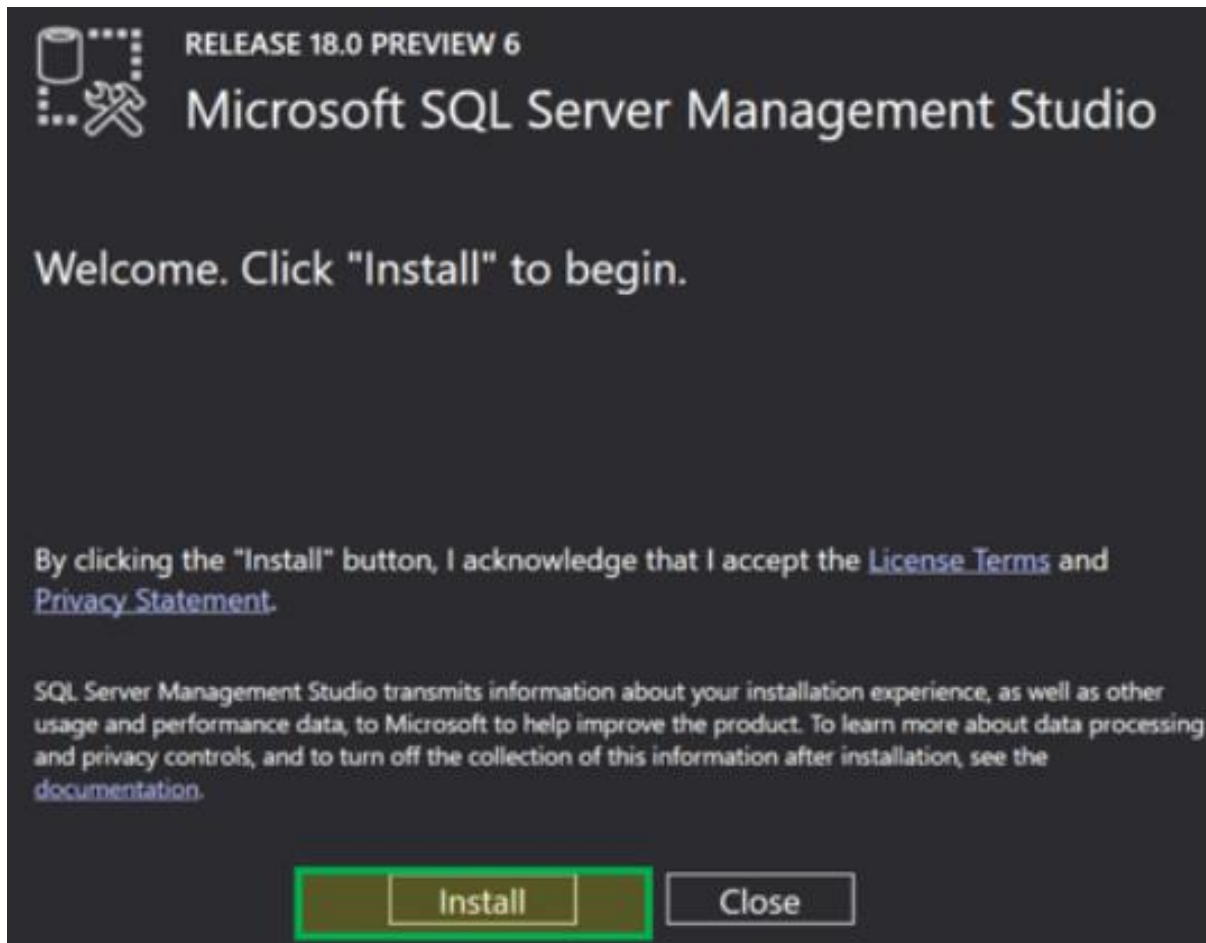
Βήμα 6°: Πηγαίνουμε στο μενού Terminal > New Terminal.

Βήμα 7°: Εγκαθιστούμε το Django με αυτή την εντολή: `python -m pip install django`.

Βήμα 8°: Δημιουργούμε την εργασία με το framework Django με την εξής εντολή: `django-admin startproject ptychiaki`.

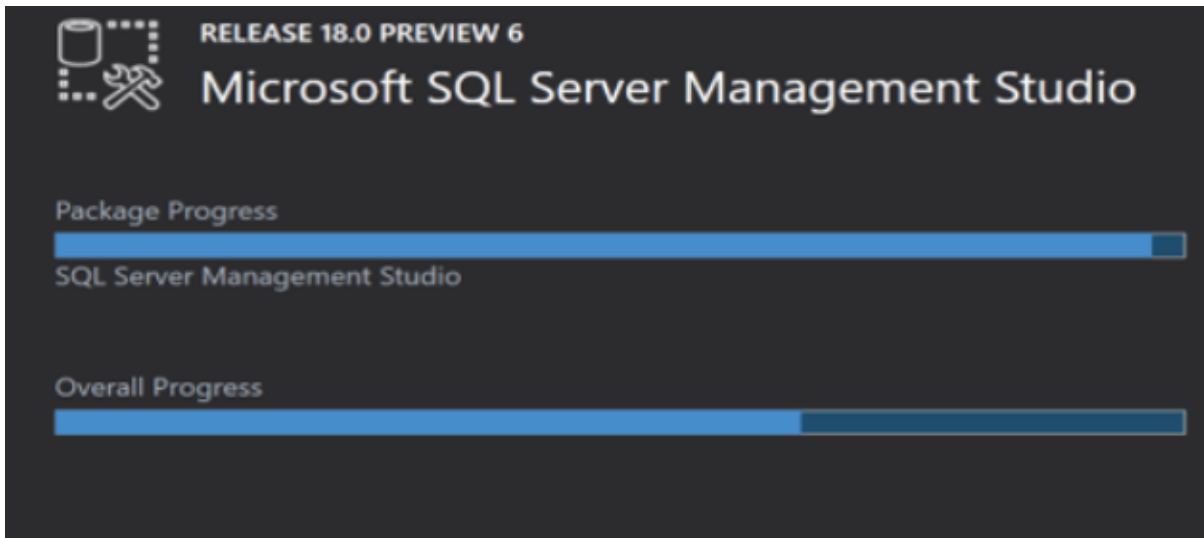
Βήμα 9°: Στο project που δημιουργήσαμε στο προηγούμενο βήμα, πρέπει να δημιουργήσουμε την εφαρμογή της πτυχιακής με αυτή την εντολή: `python manage.py startapp FlightManagementSystem`.

Τέλος, μένει να εγκαταστήσουμε το πρόγραμμα της βάσης δεδομένων, το Microsoft SQL Server Management Studio 18.



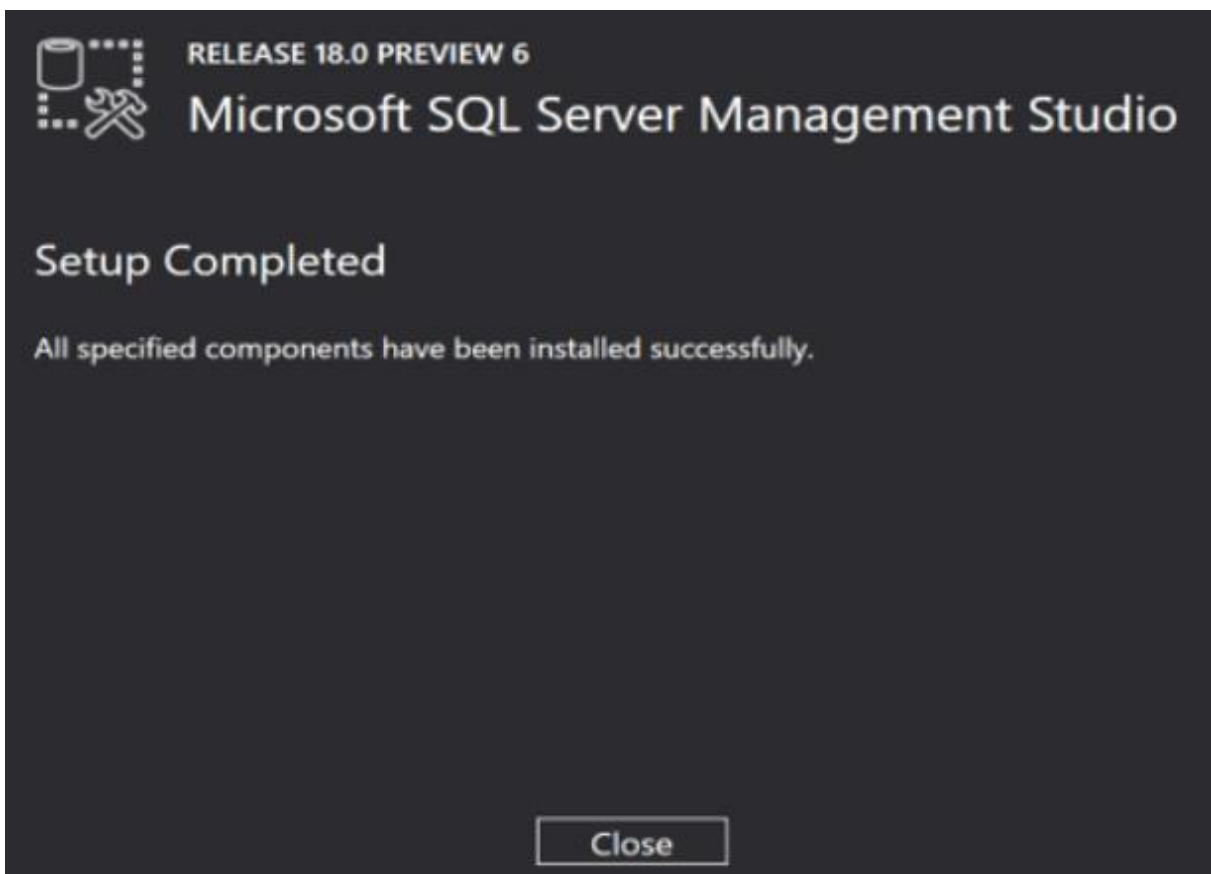
Εικόνα 14. Εγκατάσταση Microsoft SQL Server Management Studio Βήμα 1ο

Πατάμε το κουμπί “Install”, για να γίνει η εγκατάσταση του προγράμματος στο προεπιλεγμένο φάκελο του συστήματος του υπολογιστή (συνήθως είναι ο φάκελος Program Files).



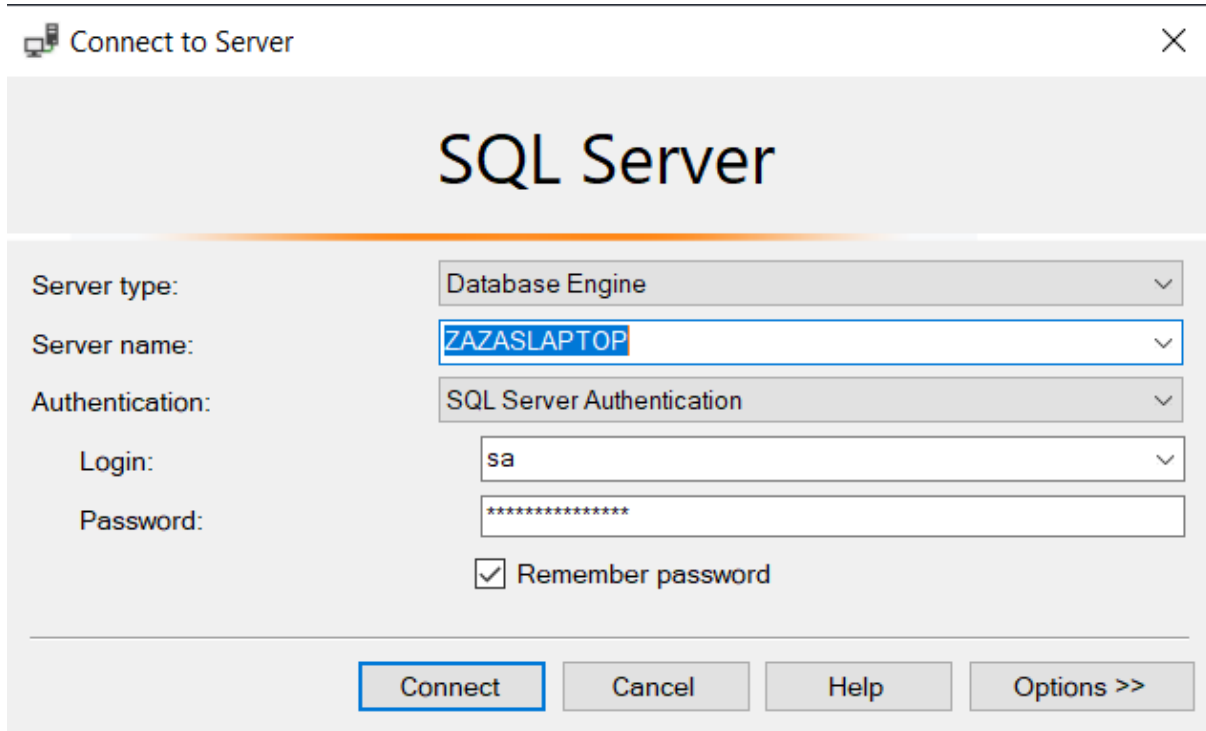
Εικόνα 15. Εγκατάσταση Microsoft SQL Server Management Studio Βήμα 2ο

Η εγκατάσταση θα ξεκινήσει. Περιμένουμε μέχρι να ολοκληρωθεί.



Εικόνα 16. Εγκατάσταση Microsoft SQL Server Management Studio Βήμα 3ο

Μόλις ολοκληρωθεί η εγκατάσταση, πατάμε το κουμπί “Close” για να κλείσει το παράθυρο.



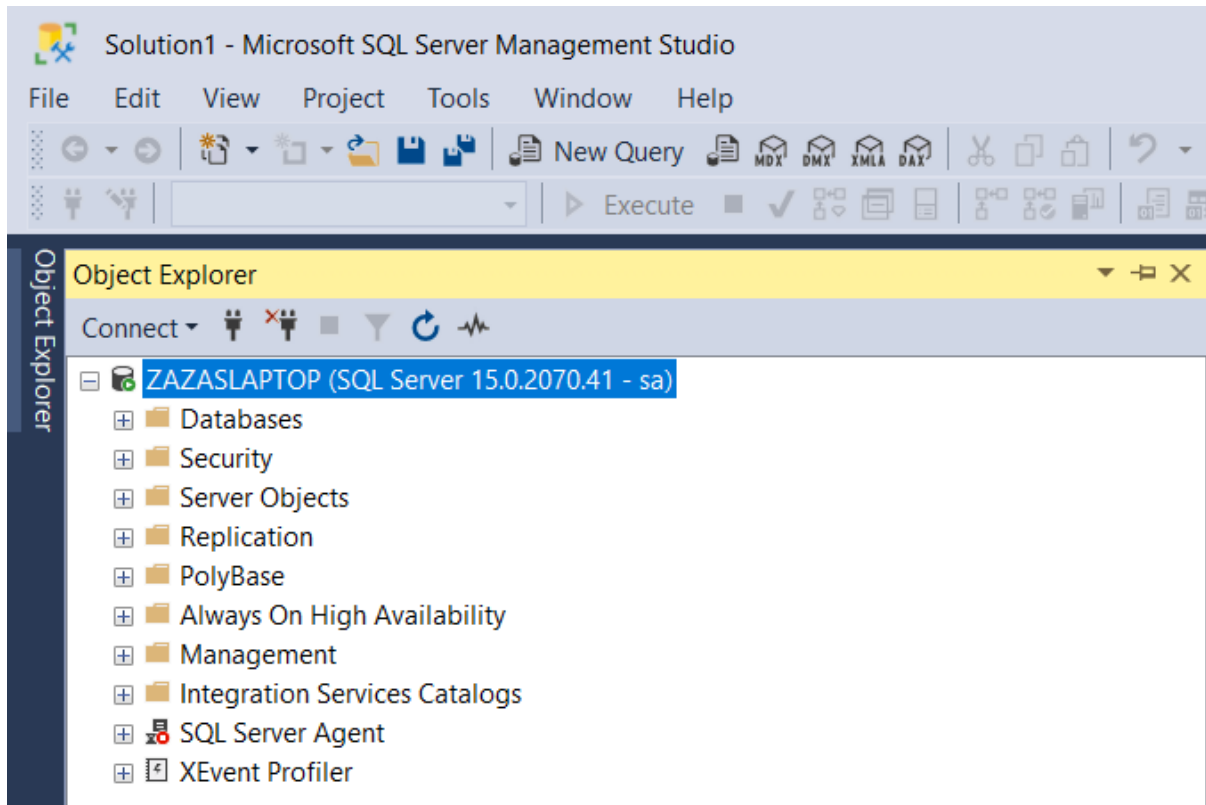
Εικόνα 17. Εγκατάσταση Microsoft SQL Server Management Studio Βήμα 4ο

Όταν ανοίξουμε το πρόγραμμα, μας βγάζει το παράθυρο “Connect to Server”. Θα πρέπει δηλαδή να συνδεθούμε με το διακομιστή της βάσης δεδομένων. Για να είναι η σύνδεση επιτυχής, θα πρέπει να συμπληρώσουμε τα εξής πεδία:

1. **Τύπος διακομιστή (Server type):** Αυτή είναι μια επιλογή, για να επιλέξουμε μία από τις τέσσερις διαθέσιμες επιλογές υπηρεσιών του προγράμματος. Θα εργαζόμαστε στο “Database Engine” (μηχανή βάσης δεδομένων), για τη δημιουργία και τη συνεργασία με τη βάση δεδομένων. Άλλος τύπος διακομιστή περιλαμβάνει υπηρεσίες ανάλυσης, αναφοράς και ενοποίησης.
2. **Όνομα διακομιστή (Server name):** Αυτό είναι το όνομα του διακομιστή, όπου είναι εγκατεστημένος ο MS SQL Server και πρέπει να δημιουργήσει τη σύνδεση με αυτόν τον διακομιστή. Γενικά, χρησιμοποιούμε το όνομα διακομιστή ως το όνομα του υπολογιστή.
3. **Έλεγχος ταυτότητας (Authentication):** Αυτό είναι προεπιλεγμένο ως «Έλεγχος ταυτότητας των Windows» κατά την εγκατάσταση του προγράμματος. Διαφορετικά, αν επιλέξουμε «Mixed Mode (Windows Authentication & Windows Authentication)», τότε ο έλεγχος ταυτότητας, θα προεπιλεγεί ως «Έλεγχος ταυτότητας του διακομιστή SQL».
4. **Σύνδεση/Κωδικός πρόσβασης (Login/Password):** Εάν έχει επιλεγεί έλεγχος ταυτότητας διαφορετικός από τον έλεγχο ταυτότητας των Windows, όπως «Έλεγχος ταυτότητας διακομιστή SQL» (πού στη προκειμένη περίπτωση είναι προεπιλεγμένος), θα απαιτηθούν αυτά τα δύο πεδία.

Εάν το επιθυμούμε, επιλέγουμε το κουτάκι «Απομνημόνευσης κωδικού», έτσι ώστε κάθε φορά που ανοίγουμε το πρόγραμμα, να μην χρειάζεται να συμπληρώνουμε τον κωδικό πρόσβασης.

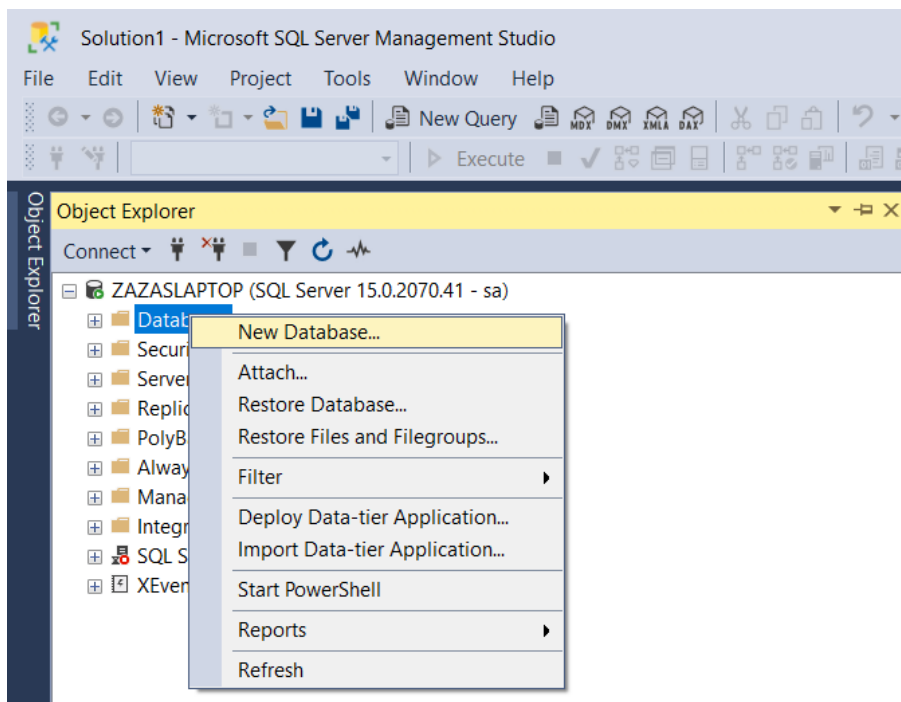
Τέλος, πατάμε το κουμπί “Connect”, για να πραγματοποιηθεί η σύνδεση.



Εικόνα 18. Εγκατάσταση Microsoft SQL Server Management Studio Βήμα 5ο

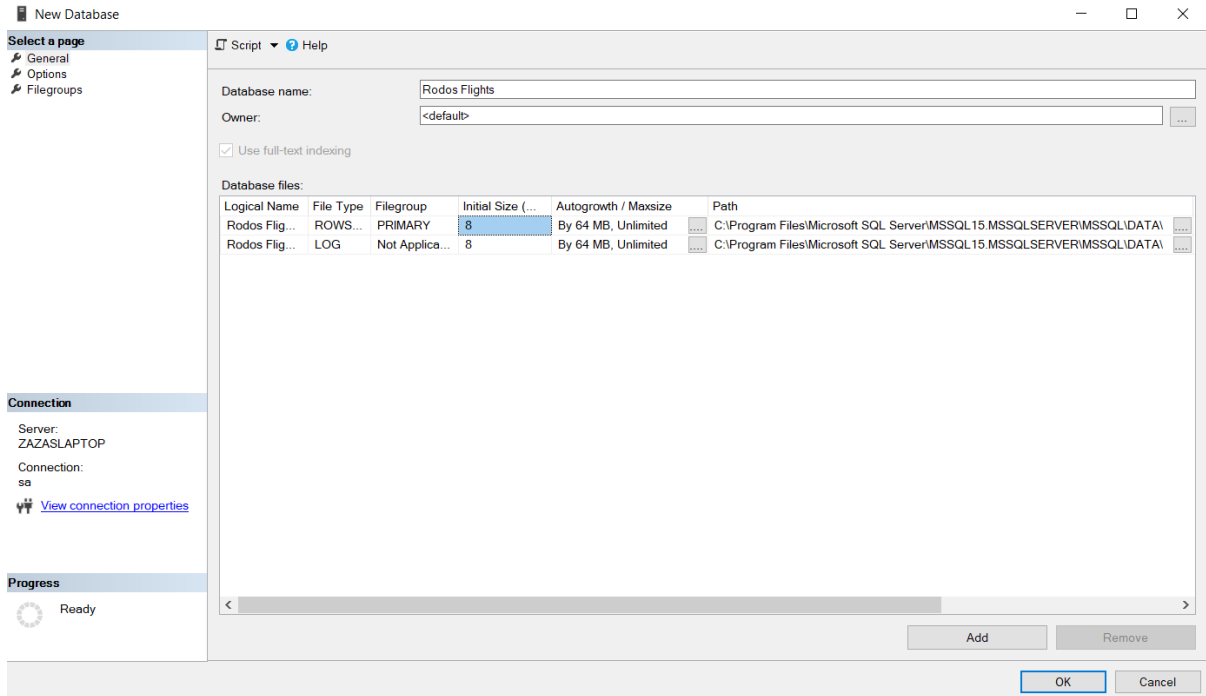
Μόλις γίνει η σύνδεση, μεταφερόμαστε στο εικονικό περιβάλλον του προγράμματος. Για να ολοκληρώσουμε την εγκατάσταση, μένει να δημιουργήσουμε τη βάση δεδομένων. Για να το κάνουμε αυτό, θα πρέπει να ακολουθήσουμε τα εξής βήματα:

Βήμα 1^ο: Κάνουμε δεξί κλικ πάνω στο φάκελο Βάσεις δεδομένων > Νέα Βάση Δεδομένων



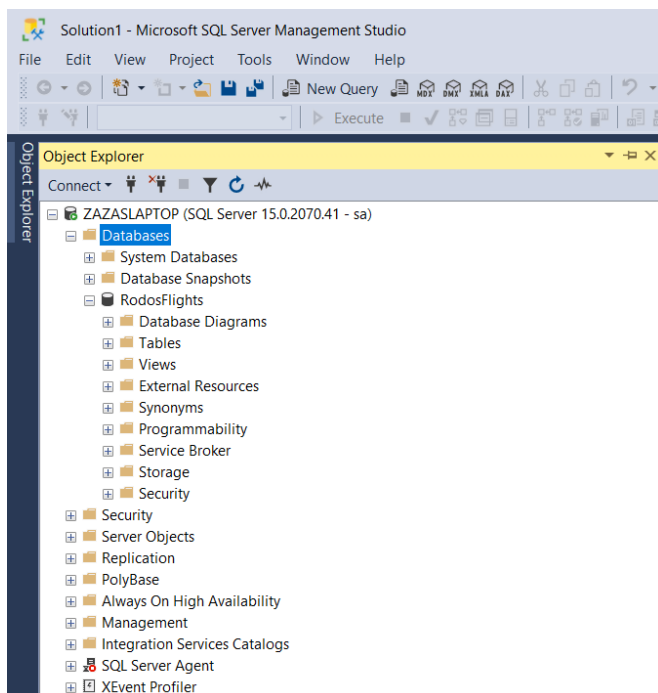
Εικόνα 19. Εγκατάσταση Βάσης Δεδομένων Βήμα 1ο

Βήμα 2^ο: Στη συνέχεια, δίνουμε το όνομα της βάσης δεδομένων (Rodos Flights), αφήνουμε όλα τα υπόλοιπα στοιχεία προεπιλεγμένα, καθώς και τον φάκελο όπου θα αποθηκευτεί η βάση δεδομένων και πατάμε το κουμπί OK.



Εικόνα 20. Εγκατάσταση Βάσης Δεδομένων Βήμα 2ο

Η βάση δεδομένων είναι έτοιμη προς χρήση. Γι' αυτή την εργασία, θα ασχοληθούμε με τους πίνακες (tables) και τις αποθηκευμένες διαδικασίες (stored procedures). Στα επόμενα κεφάλαια, υπάρχει η περιγραφή και η υλοποίηση για τα παραπάνω.



Εικόνα 21. Βάση Δεδομένων του Microsoft SQL Server Management Studio

3 Μοντελοποίηση της εφαρμογής

Για να γίνει κατανοητή η σχεδίαση όλης της εφαρμογής, είναι απαραίτητο να γνωρίζουμε τα βήματα, τα οποία αποτελούν την ιεραρχία μοντελοποίησης της εφαρμογής. Παρακάτω, περιγράφονται τα στάδια ανάλυσης της εφαρμογής.

3.1 Ανάλυση εφαρμογής

Η εφαρμογή, περιέχει την δημιουργία της κάθε εγγραφής, πού αποτελείται από τα εξής στάδια:

Στάδια	Γλώσσα Προγραμματισμού
Δημιουργία πινάκων στη βάση δεδομένων	SQL
Εισαγωγή εικόνων στον editor	-
Δημιουργία σελίδων HTML	HTML
Δημιουργία σελίδων JavaScript	JavaScript
Δημιουργία σελίδων CSS	CSS
Δημιουργία models	Python & Django
Δημιουργία URLs	Python & Django
Δημιουργία stored procedures στη βάση δεδομένων	SQL
Δημιουργία views	Python, Django & JSON

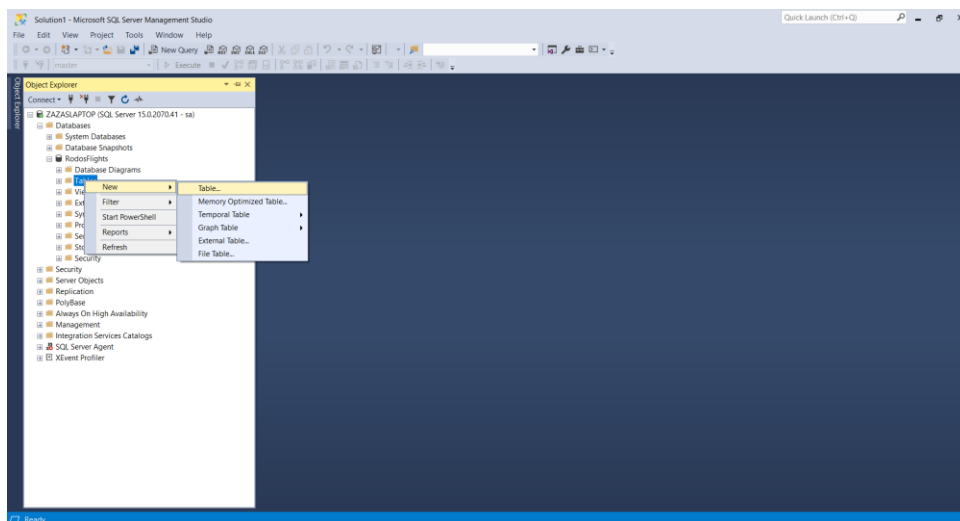
Πίνακας 1. Στάδια δημιουργίας της εφαρμογής

3.2 Σχεδιασμός Υλοποίησης

3.2.1 Δημιουργία πινάκων στη βάση δεδομένων

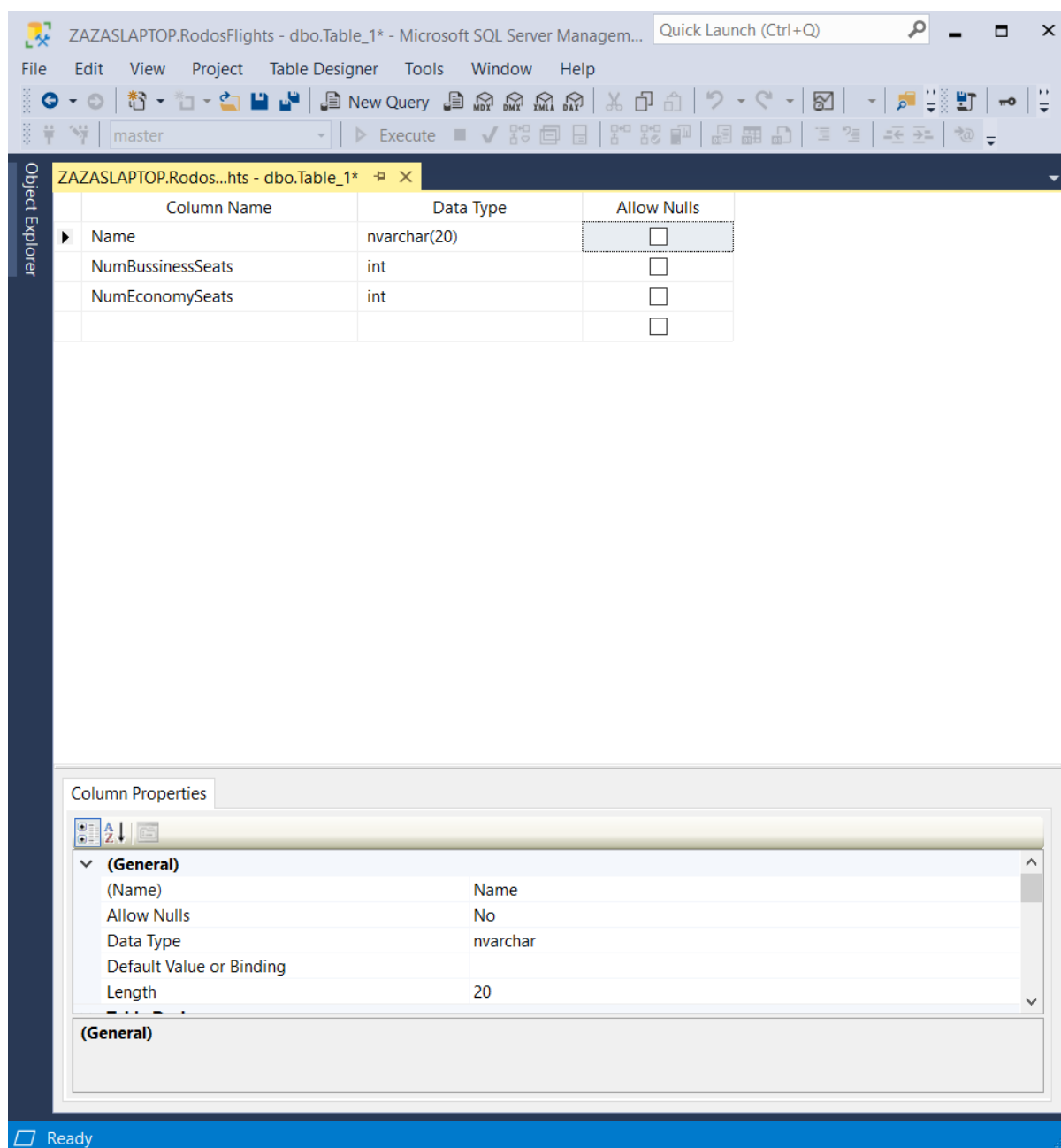
Υπάρχουν δύο τρόποι για να δημιουργήσουμε πίνακες στη βάση δεδομένων. Ο πρώτος και πιο απλός τρόπος, είναι να χρησιμοποιήσουμε άμεσες ενέργειες του προγράμματος Microsoft SQL Management Studio ως εξής:

Βήμα 1^ο: Πηγαίνουμε στη βάση δεδομένων και πατάμε δεξί κλικ στο φάκελο Tables > New > Table...




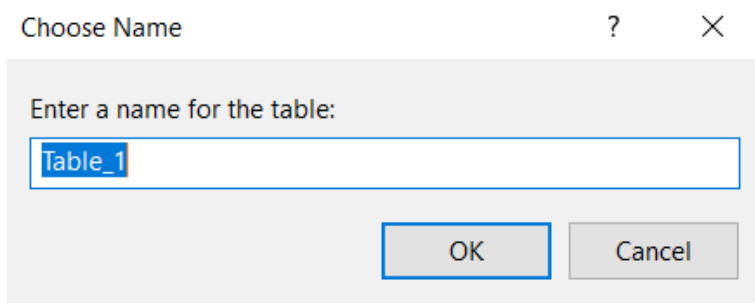
Εικόνα 22. Δημιουργία πινάκων στη βάση δεδομένων με άμεσες ενέργειες Βήμα 1ο

Βήμα 2º: Μετά, εισάγουμε τα ονόματα των πεδίων του πίνακα, τον τύπο τους και το allow nulls, είναι προτιμότερο να το αφήνουμε κενό, για να μην αφήνουμε αυτά τα πεδία κενά.



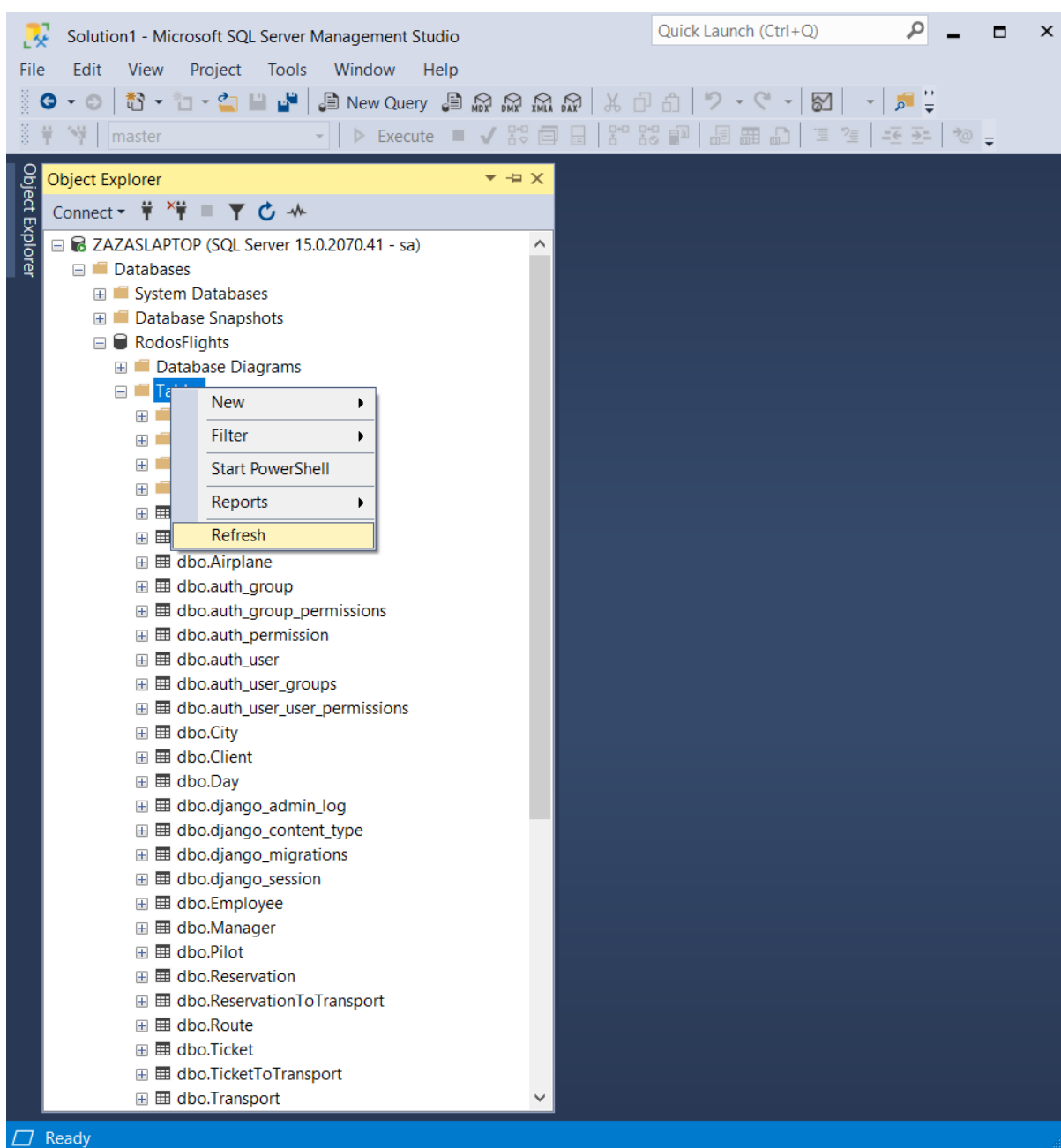
Εικόνα 23. Δημιουργία πινάκων στη βάση δεδομένων με άμεσες ενέργειες Βήμα 2ο

Βήμα 3º: Μόλις τελειώσουμε με τη συμπλήρωση των πεδίων, πατάμε το κουμπί της αποθήκευσης , εισάγουμε το όνομα του πίνακα και πατάμε OK.



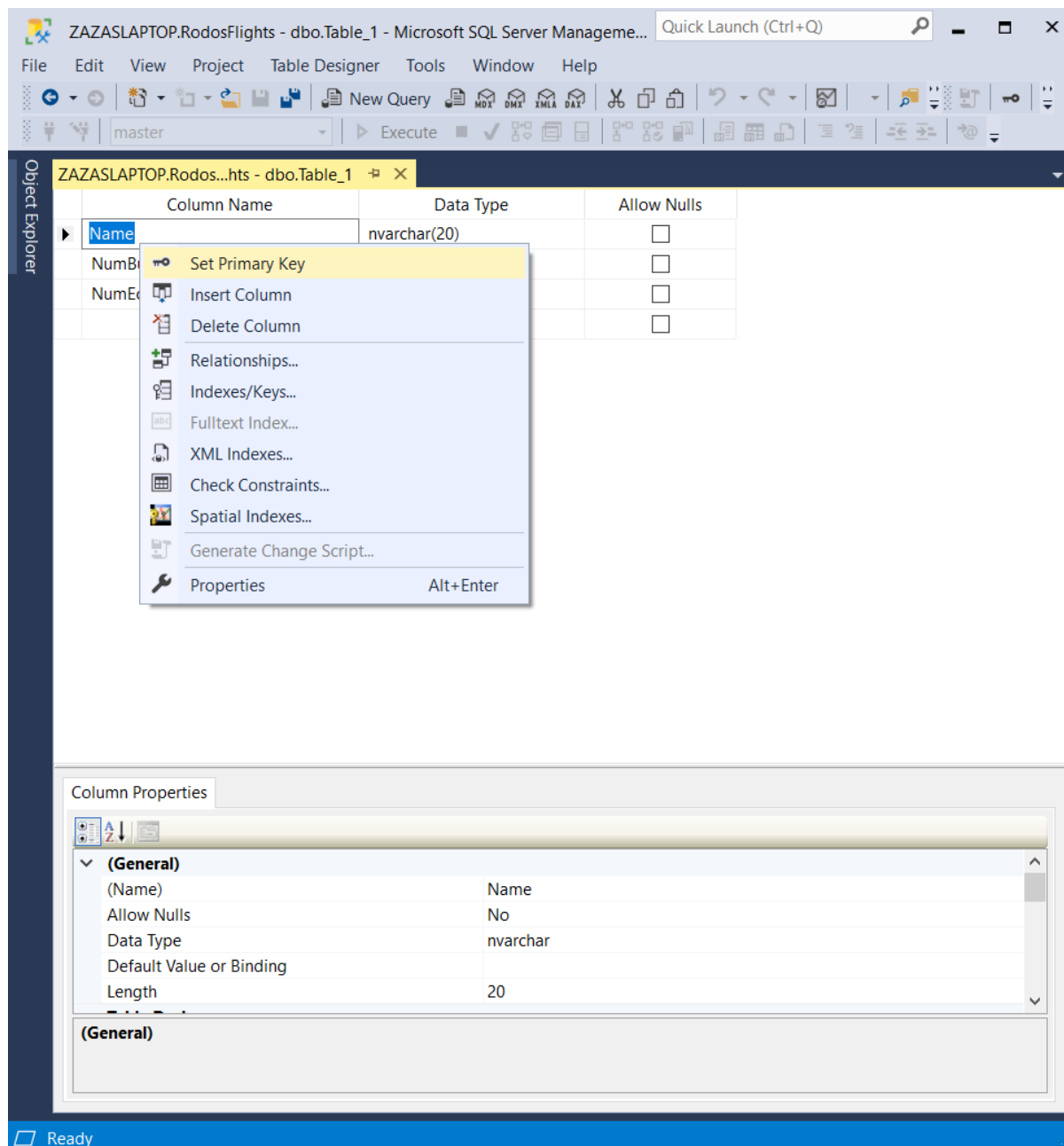
Εικόνα 24. Δημιουργία πινάκων στη βάση δεδομένων με άμεσες ενέργειες Βήμα 3^ο

Υπάρχει περίπτωση να μην δούμε τον νέο πίνακα μέσα στον φάκελο Tables. Γι' αυτό θα πρέπει να κάνουμε ανανέωση, κάνοντας δεξί κλικ πάνω στον εν λόγω φάκελο.



Εικόνα 25. Ανανέωση λίστας πινάκων

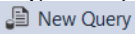
Βήμα 4^ο: Το επόμενο βήμα, είναι να θέσουμε το πρωτεύον κλειδί του πίνακα, το οποίο είναι μοναδικό για κάθε πίνακα. Για να γίνει αυτό, θα πρέπει να πάμε στα πεδία του πίνακα (Δεξί κλικ στον πίνακα > Columns > Δεξί κλικ σε οποιοδήποτε Πεδίο > Modify. Μετά, κάνουμε ξανά δεξί κλικ πάνω στο πεδίο, που θέλουμε να του ορίσουμε το πρωτεύον κλειδί και επιλέγουμε Set Primary Key.



Εικόνα 26. Δημιουργία πινάκων στη βάση δεδομένων με άμεσες ενέργειες Βήμα 4^ο

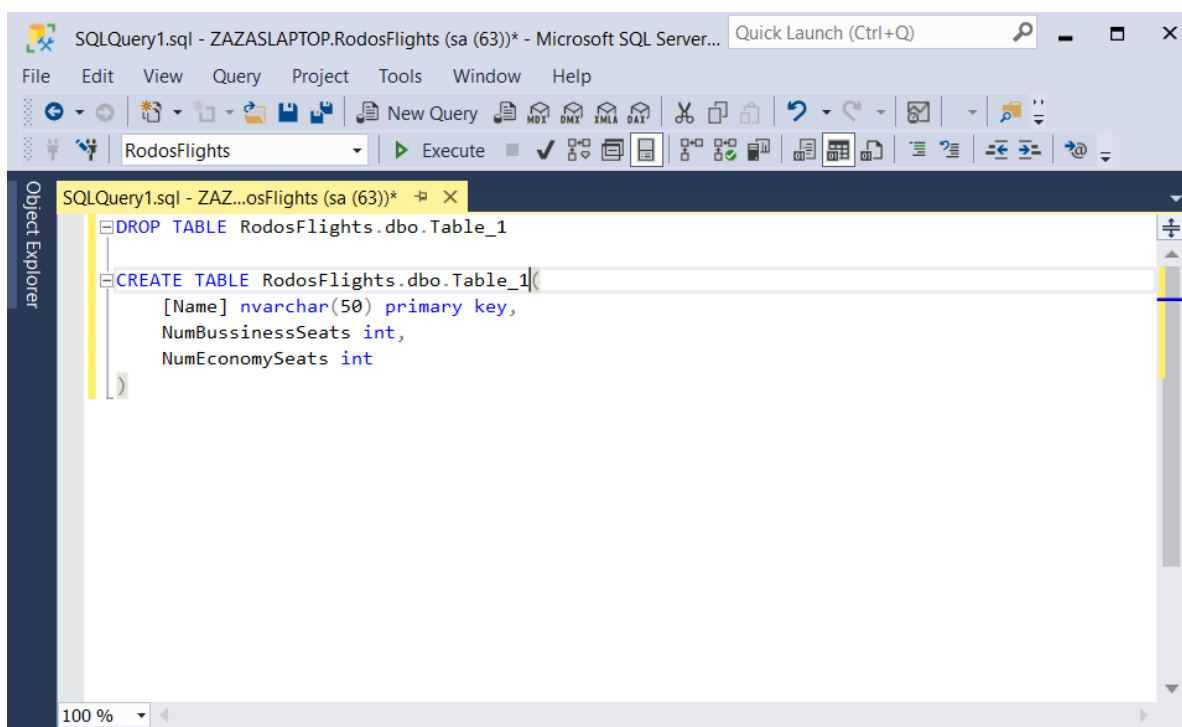
Είναι σημαντικό να μην ξεχνάμε να αποθηκεύουμε όποιες αλλαγές κάνουμε, καθώς να κάνουμε και ανανέωση των πινάκων.

Ο δεύτερος τρόπος δημιουργίας πινάκων, γίνεται με χρήση του κώδικα SQL (τα λεγόμενα Queries). Η σύνταξή του έχει ως εξής:

Βήμα 1^ο: Πηγαίνουμε στη γραμμή εργαλείων του προγράμματος στη πάνω μεριά και πατάμε το κουμπί .

Εκεί μπορούμε να γράψουμε κώδικα με τη γλώσσα προγραμματισμού SQL, πού μας επιτρέπει να εισάγουμε πίνακες, να τροποποιούμε τα πεδία τους, καθώς και να ορίζουμε τις συσχετίσεις τους με άλλους πίνακες. Όλα αυτά, μπορούμε να τα δούμε και παρακάτω στο κεφάλαιο της υλοποίησης της εργασίας.

Προς το παρόν, για τη δημιουργία νέου πίνακα, πρέπει να γράψουμε τον παρακάτω κώδικα:




Εικόνα 27. Δημιουργία πινάκων στη βάση δεδομένων με χρήση κώδικα SQL

Ο παραπάνω κώδικας, κάνει τις ίδιες ακριβώς ενέργειες με αυτές του πρώτου τρόπου δημιουργίας πίνακα πού είδαμε παραπάνω.

Για να βεβαιωθούμε ότι ο συγκεκριμένος πίνακας δεν υπάρχει ήδη ή δεν χρησιμοποιείται κάπου, χρησιμοποιούμε την εντολή DROP TABLE ΌνομαΒάσηςΔεδομένων.dbo.ΌνομαΠίνακα.

Επίσης, παρατηρούμε ότι το ένα από τα πεδία είναι μέσα σε αγκύλες [] (πεδίο Name). Αυτό γίνεται για τον λόγο ότι υπάρχουν έτοιμα πεδία και συναρτήσεις στην SQL γλώσσα, τα οποία αναγνωρίζουμε από το μπλε φόντο τους. Για να διαφοροποιήσουμε λοιπόν τα πεδία πού ορίζουμε εμείς από αυτά της SQL, τα βάζουμε μέσα σε αγκύλες.

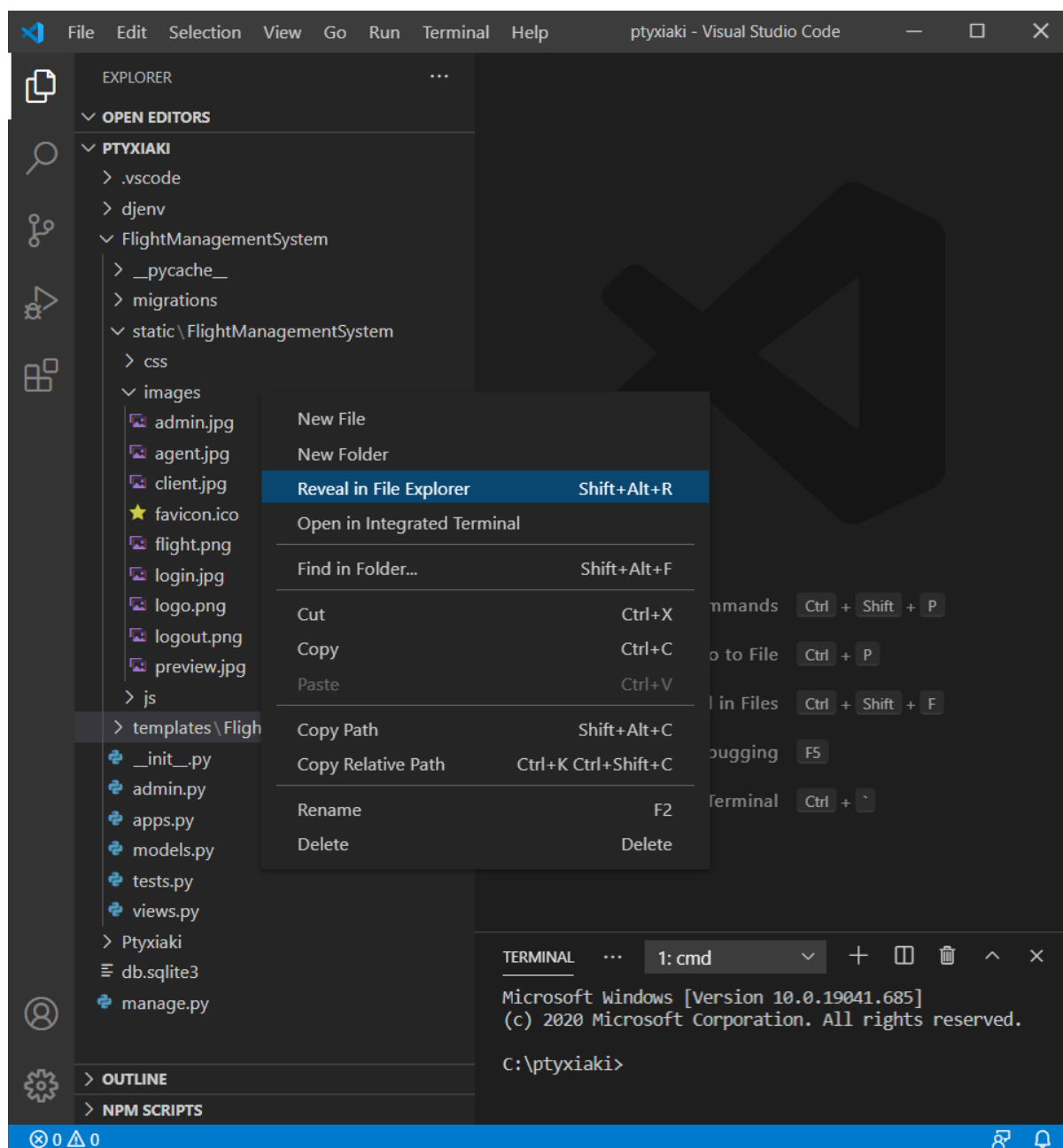
Για να εκτελέσουμε το Query, πατάμε το κουμπί  και εάν δεν υπάρξουν συντακτικά λάθη, η εκτέλεση έγινε επιτυχώς.

3.2.2 Εισαγωγή εικόνων στον editor

Μπορούμε να εισάγουμε εικόνες στον editor, πού να φαίνονται είτε στο φόντο, είτε σαν απλά εικονίδια, είτε σαν λογότυπα δίπλα από τον τίτλο πάνω στη καρτέλα της σελίδας. Τις εικόνες μπορούμε να τις εισάγουμε με τον παρακάτω τρόπο:

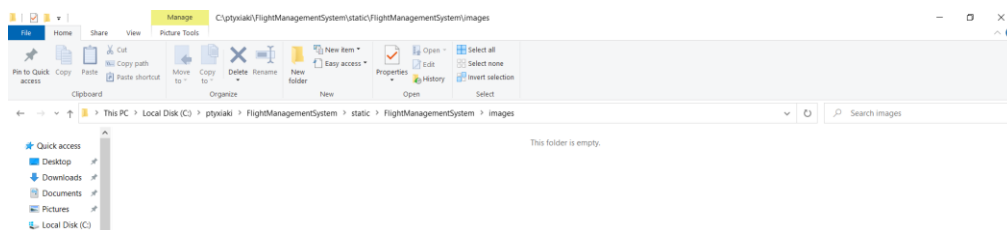
Πηγαίνουμε στην αριστερή καρτέλα του προγράμματος Visual Studio Code.

Βρίσκουμε τον φάκελο `static`, πού περιέχει τον υποφάκελο `FlightManagementSystem`, τον οποίο δημιουργήσαμε κατά την εγκατάσταση του προγράμματος. Κάνουμε δεξί κλικ πάνω στον υποφάκελο `images > Reveal in File Explorer`.



Εικόνα 28. Εισαγωγή εικόνων στον editor (1)

Πρόκειται να μας μεταφέρει στον φάκελο `images`, πού βρίσκεται μέσα στον φάκελο της πτυχιακής εργασίας, πού είναι αποθηκευμένος στον σκληρό δίσκο του υπολογιστή. Εκεί μέσα, μπορούμε να αποθηκεύσουμε τις εικόνες και να τις φορτώσουμε αργότερα, μέσω των σελίδων `html` με τη κατάλληλη εντολή κώδικα, ανατρέχοντας το μονοπάτι προορισμού (`destination path`), όπου βρίσκονται αποθηκευμένες οι εικόνες στον υπολογιστή.



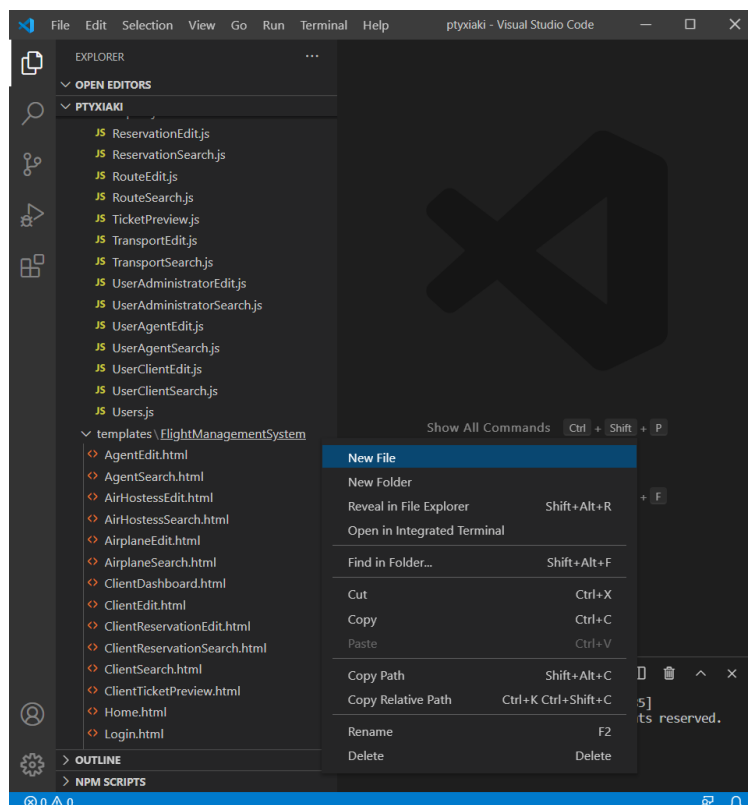
Εικόνα 29. Εισαγωγή εικόνων στον editor (2)

3.2.3 Δημιουργία σελίδων HTML

Για τη δημιουργία των σελίδων HTML, υπάρχουν πολλές εντολές με κώδικα, πού μπορούμε να χρησιμοποιήσουμε, ανάλογα με τη φύση των ζητούμενων. Στη παρούσα εργασία, χρησιμοποιούμε το μοντέλο των λιστών. Δηλαδή όταν μπαίνουμε σε μια σελίδα, βλέπουμε τις κατηγορίες των εγγραφών ομαδοποιημένες και όποτε θέλουμε να ανατρέξουμε τα στοιχεία τους, ή να δημιουργήσουμε νέα, χρησιμοποιούμε τα κουμπιά για την αντίστοιχη ενέργεια. Όλα τα παραπάνω, μπορούμε να τα δούμε στο κεφάλαιο υλοποίησης της εργασίας. Επίσης, μπορούμε να φορτώσουμε εικόνες οποιουδήποτε τύπου μέσα στη σελίδα και να ορίσουμε τις διαστάσεις τους.

Εδώ τα συγκεκριμένα βήματα δημιουργίας μιας HTML σελίδας, είναι τα εξής:

Βήμα 1^ο: Πηγαίνουμε στην αριστερή καρτέλα του προγράμματος Visual Studio Code. Βρίσκουμε τον φάκελο templates, πού περιέχει τον υποφάκελο FlightManagementSystem, τον οποίο δημιουργήσαμε κατά την εγκατάσταση του προγράμματος. Κάνουμε δεξί κλικ πάνω στον υποφάκελο > New File. Τότε, γράφουμε το όνομα του αρχείου, μαζί με τη κατάληξη .html, για να αναγνωριστεί το αρχείο ως σελίδα html από τον editor. Το πρώτο γράμμα του ονόματος, πρέπει να είναι πάντα κεφαλαίο.



Εικόνα 30. Δημιουργία σελίδων HTML

Βήμα 2^ο: Στη σελίδα που δημιουργήσαμε, χρησιμοποιούμε συγκεκριμένους κώδικες (στοιχεία HTML), που βασίζονται στη μορφοποίηση και στη χρήση συναρτήσεων-αλγορίθμων, τις οποίες υλοποιούμε με κώδικα JavaScript σε ξεχωριστά αρχεία τύπου .js, τα οποία περιγράφονται παρακάτω.

Σε όλες τις σελίδες HTML, χρησιμοποιούνται τα εξής στοιχεία:

- **<html>**: Αποτελούν το «σκελετό» όλης της σελίδας, όπου μέσα σε αυτό, συμπεριλαμβάνονται τα υπόλοιπα στοιχεία της σελίδας.
- **<head>**: Αποτελούν τη «κεφαλή» όλης της σελίδας. Χρησιμοποιούνται για την εισαγωγή έτοιμων βιβλιοθηκών μορφοποίησης της σελίδας (<meta>), συνδέσμων (<link>) για την εισαγωγή περαιτέρω σελίδων μορφοποίησης, όπως σελίδες με κώδικα CSS και εικόνες συντόμευσης (favicon), καθώς και σενάρια <script>, για την εισαγωγή σελίδων επεξεργασίας συναρτήσεων-αλγορίθμων, με κώδικα γραμμένο σε γλώσσα JavaScript. Τέλος, περιέχει τον τίτλο <title>, που εμφανίζεται στο περίγραμμα στη πάνω μεριά της σελίδας.
- **<body>**: Αποτελούν το «σώμα» όλης της σελίδας. Χρησιμοποιούνται για την εισαγωγή διάφορων στοιχείων (<div>, εικόνων και <buttons> περισσότερο), τα οποία σχετίζονται με έτοιμες κλάσεις (class). Δηλαδή, καλούν συναρτήσεις από τις σελίδες με κώδικα JavaScript και δημιουργούν πεδία για να μπορεί να εισάγει ο πελάτης (client) στοιχεία και να διευκολύνει την επικοινωνία του με τον χρήστη (server).

```
{% load static %}
<!DOCTYPE html>

<html>

  <head>
    <meta ...ΚΩΔΙΚΑΣ...>

    <link ...ΚΩΔΙΚΑΣ...>

    <script>ΚΩΔΙΚΑΣ</script>

    <title>ΠΑΡΑΔΕΙΓΜΑ</title>
  </head>

  <body>
    <input ...ΚΩΔΙΚΑΣ.../>

    <div>
      .
      .
      .
      ΚΩΔΙΚΑΣ
      .
      .
      .
    </div>

    <div>
      <div>
        <button>
          .
          .
          .
          ΚΩΔΙΚΑΣ
          .
          .
          .
        </button>
      </div>
    </div>
  </body>

</html>
```

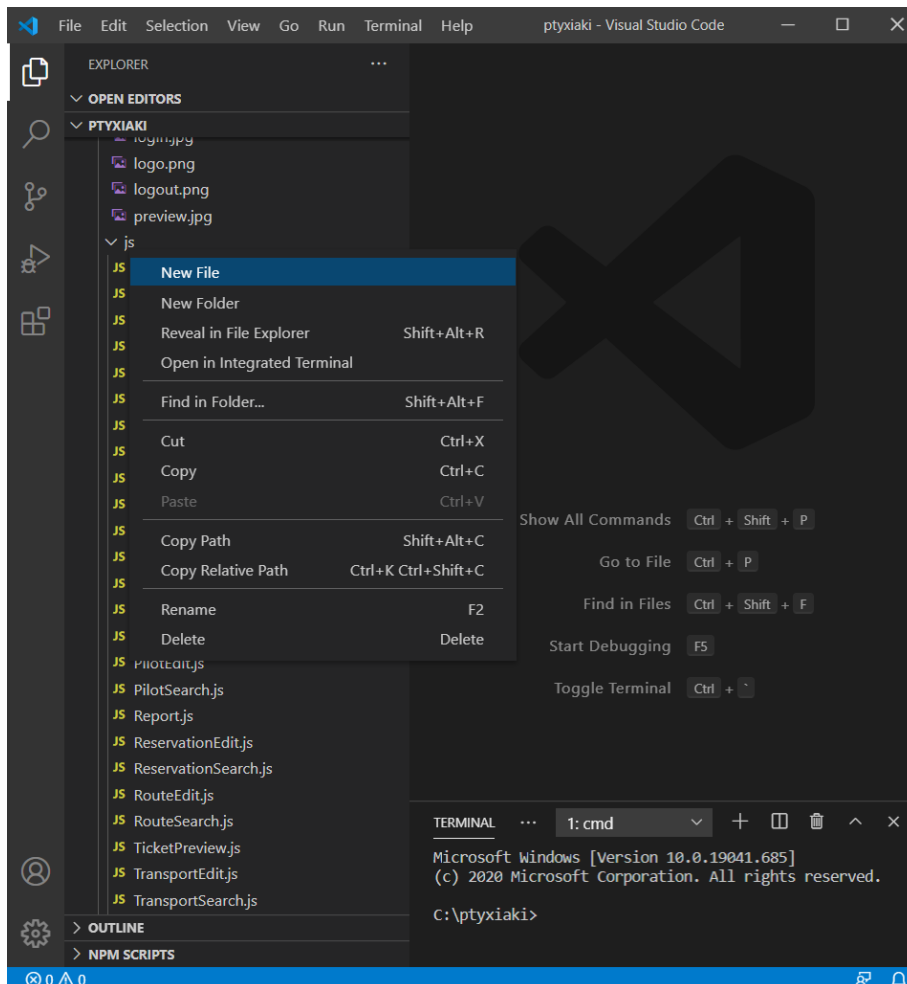
Εικόνα 31. Βασική σύνταξη σελίδας HTML

3.2.4 Δημιουργία σελίδων JavaScript

Για τη δημιουργία των σελίδων JavaScript, υπάρχουν εξίσου πολλές συναρτήσεις-αλγόριθμοι, πού μπορούμε να χρησιμοποιήσουμε, ανάλογα με τη φύση των ζητούμενων. Για παράδειγμα, δημιουργούμε μια συνάρτηση, πού η εφαρμογή μας ζητάει με τη μορφή μηνύματος, να εισάγουμε τα στοιχεία ενός πελάτη. Μόλις τα εισάγουμε, τότε η εφαρμογή κάνει έλεγχο εγκυρότητας των στοιχείων (μη κενά πεδία λόγου χάρη) και μας βγάζει επίσης μήνυμα αν τα στοιχεία πού δώσαμε είναι έγκυρα κι αν ναι, αποθηκεύονται στη βάση δεδομένων. Αλλιώς το μήνυμα γράφει ότι τα στοιχεία δεν είναι έγκυρα.

Εδώ τα συγκεκριμένα βήματα δημιουργίας μιας JavaScript σελίδας, είναι τα εξής:

Βήμα 1^ο: Πηγαίνουμε στην αριστερή καρτέλα του προγράμματος Visual Studio Code. Βρίσκουμε τον φάκελο static, πού περιέχει τον υποφάκελο FlightManagementSystem, τον οποίο δημιουργήσαμε κατά την εγκατάσταση του προγράμματος. Κάνουμε δεξί κλικ πάνω στον υποφάκελο js > New File. Τότε, γράφουμε το όνομα του αρχείου, μαζί με τη κατάληξη .js, για να αναγνωριστεί το αρχείο ως σελίδα javascript από τον editor. Το πρώτο γράμμα του ονόματος, πρέπει να είναι πάντα κεφαλαίο.



Εικόνα 32. Δημιουργία σελίδων JavaScript

Βήμα 2^ο: Στη σελίδα πού δημιουργήσαμε, χρησιμοποιούμε συναρτήσεις (έτοιμες και μη), πού βασίζονται στην επικοινωνία μεταξύ των σελίδων html και της βάσης δεδομένων.

Μέσω των συναρτήσεων (functions), χρησιμοποιούνται τα πεδία από τη βάση δεδομένων και εκτελούμε συγκεκριμένες ενέργειες.

Σε αυτή τη σελίδα επίσης, χρησιμοποιούμε μια τάξη (class). Εκεί δημιουργείται το αντικείμενο της εγγραφής, πού θέλουμε να δημιουργήσουμε συμπληρώνοντας τα πεδία, τα οποία την αποτελούν.

```
let myValue1 = false;
let myValue2 = 0;
let myValue3 = 'hello';

function insertElements(){
  /*
  ΚΩΔΙΚΑΣ
  */
}

function validateElements(element1, element2){
  /*
  ΚΩΔΙΚΑΣ
  */
}

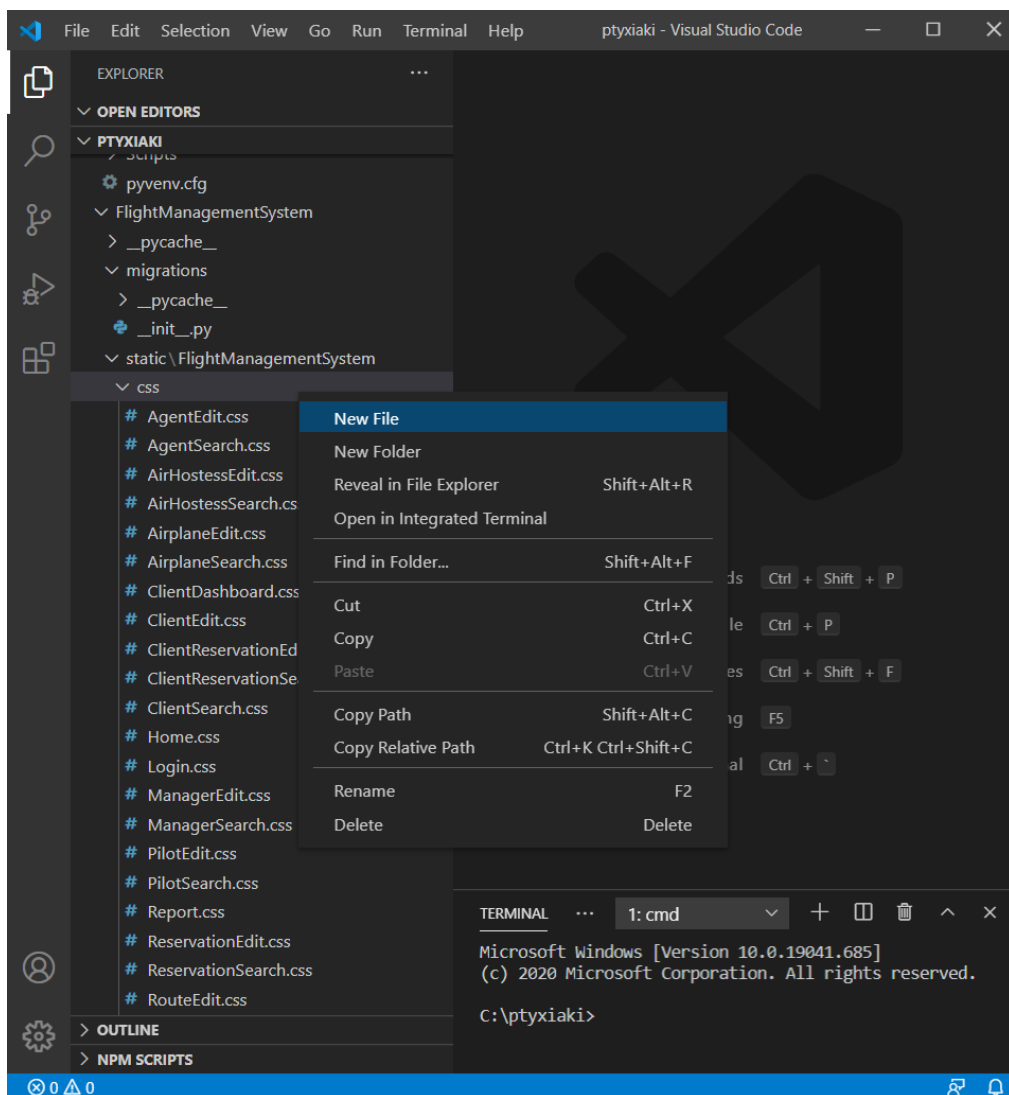
class Elements{
  /*
  ΚΩΔΙΚΑΣ
  */
}
```

Εικόνα 33. Βασική σύνταξη σελίδας JavaScript

3.2.5 Δημιουργία σελίδων CSS

Για τη δημιουργία των σελίδων CSS, υπάρχουν οι κλάσεις των πεδίων των σελίδων html. Δηλαδή, χρησιμοποιούμε τα ονόματα των κλάσεων, πού ορίζουμε στα πεδία div της σελίδας html, για να επεξεργαστούμε την ωραιοποίηση της σελίδας της εφαρμογής. Εδώ τα συγκεκριμένα βήματα δημιουργίας μιας CSS σελίδας, είναι τα εξής:

Βήμα 1^ο: Πηγαίνουμε στην αριστερή καρτέλα του προγράμματος Visual Studio Code. Βρίσκουμε τον φάκελο static, πού περιέχει τον υποφάκελο FlightManagementSystem, τον οποίο δημιουργήσαμε κατά την εγκατάσταση του προγράμματος. Κάνουμε δεξί κλικ πάνω στον υποφάκελο css > New File. Τότε, γράφουμε το όνομα του αρχείου, μαζί με τη κατάληξη .css, για να αναγνωριστεί το αρχείο ως σελίδα javascript από τον editor. Το πρώτο γράμμα του ονόματος, πρέπει να είναι πάντα κεφαλαίο.



Εικόνα 34. Δημιουργία σελίδων CSS

Βήμα 2^ο: Στη σελίδα πού δημιουργήσαμε, χρησιμοποιούμε κλάσεις, πού βασίζονται στην ωραιοποίηση της σελίδας της εφαρμογής. Όπως για παράδειγμα τη μορφοποίηση της γραμματοσειράς των γραμμάτων, το σχήμα των κουμπιών, τη στοίχιση των πεδίων στην οθόνη ή το χρώμα των ετικετών.


```
.field{  
  display: flex;  
  margin-bottom: 10px;  
}  
  
.lbl{  
  width: 230px;  
  text-align: right;  
  margin-right: 10px;  
}  
  
#txtNumBussinessSeats{  
  width: 60px;  
}  
  
.buttons{  
  display: flex;  
  justify-content: flex-end;  
  margin-right: 20px;  
  margin-bottom: 20px;  
  margin-top: 20px;  
}
```

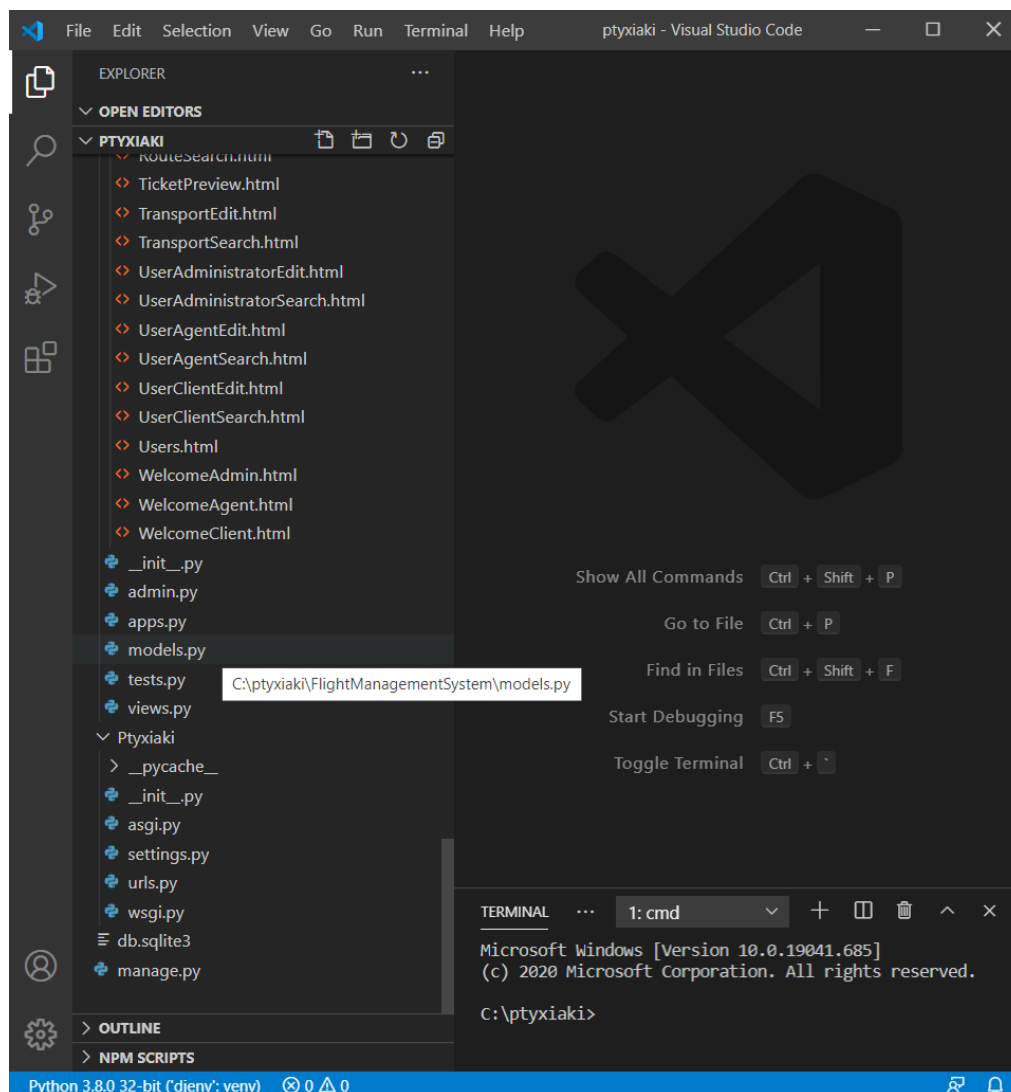
Εικόνα 35. Βασική σύνταξη σελίδας CSS

3.2.6 Δημιουργία models

Για κάθε εγγραφή που δημιουργούμε, θα πρέπει να δημιουργούμε και το αντίστοιχο μοντέλο του (model). Πρόκειται για τη τάξη, που δημιουργούμε και στις σελίδες JavaScript. Το μοντέλο επιτρέπει στον server της Django, να επικοινωνεί με τη βάση δεδομένων και να ορίζει τα πεδία της εγγραφής.

Εδώ τα συγκεκριμένα βήματα δημιουργίας ενός μοντέλου, είναι τα εξής:

Βήμα 1^ο: Πηγαίνουμε στην αριστερή καρτέλα του προγράμματος Visual Studio Code. Βρίσκουμε τον φάκελο FlightManagementSystem, τον οποίο δημιουργήσαμε κατά την εγκατάσταση του προγράμματος. Εκεί θα βρούμε τη σελίδα models.py.



Εικόνα 36. Σελίδα models

Βήμα 2^ο:

```
from django.db import models
```

```
# Create your models here.
```

```
class Elements:
```

```
    def __init__(this, id="", firstName="", lastName="", street="", streetNo="", city=""):
        this.id = id
        this.firstName = firstName
        this.lastName = lastName
        this.street = street
        this.streetNo = streetNo
        this.city = city
```

Εικόνα 37. Βασική σύνταξη σελίδας models

3.2.7 Δημιουργία URLs

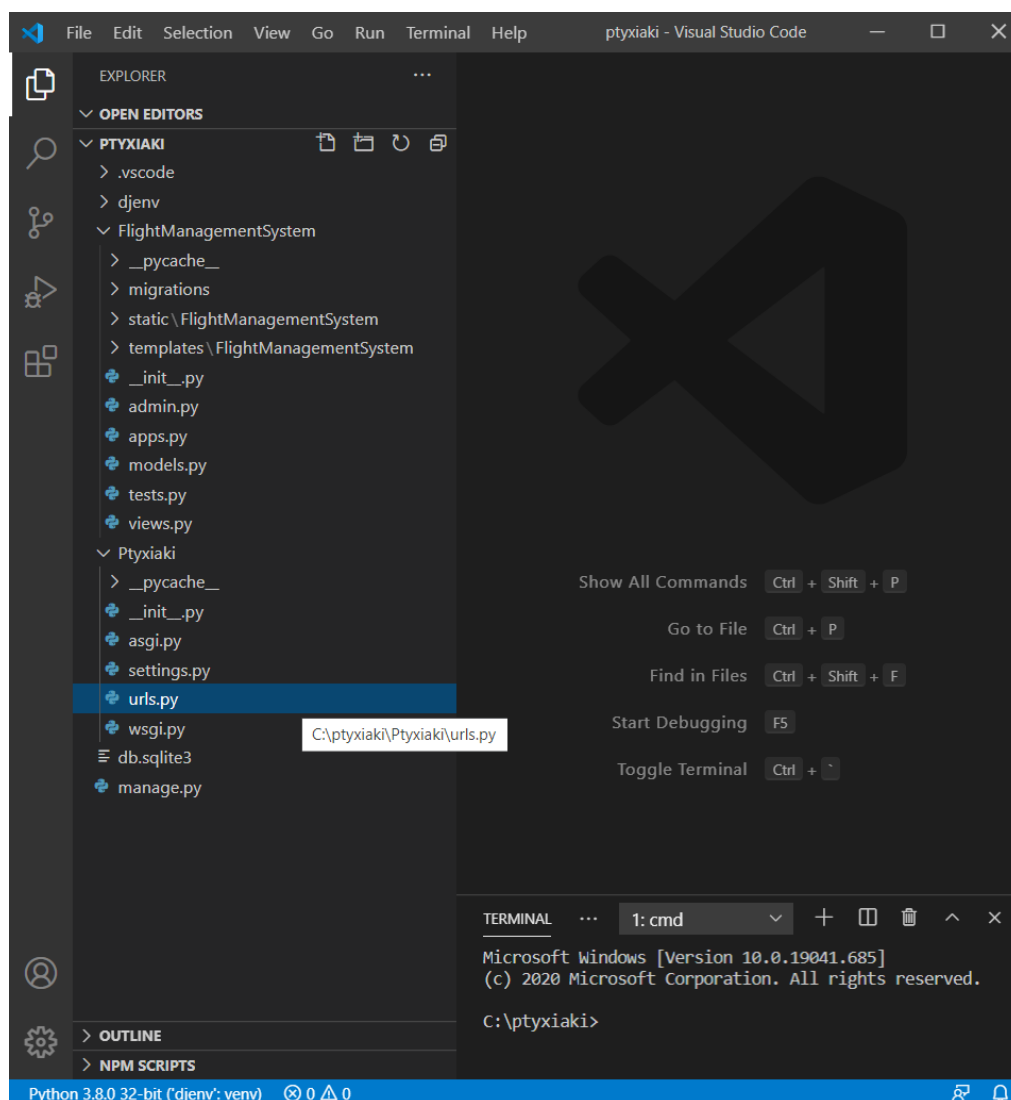
Κάθε εγγραφή, εκτός από τα models και τις σελίδες του, έχει και τις δικές του ιστοσελίδες. Για να τις ανοίξουμε στην εφαρμογή, θα πρέπει μέσω του editor, να ορίσουμε τα URL τους. Σ' αυτή την εργασία, τα URL για κάθε εγγραφή είναι παραπάνω από ένα. Για την απλοϊκότητα και τη γρηγορότερη φόρτωση των σελίδων, είναι προτιμότερο να χρησιμοποιούμε πολλά URL, παρά περισσότερο κώδικα σε ένα και μόνο URL.

Πρέπει επίσης να τους ορίσουμε τουλάχιστον ένα απαραίτητο πεδίο, επειδή στη βάση δεδομένων, κάθε εγγραφή έχει από ένα πρωτεύον κλειδί. Αυτό το κλειδί, το χρησιμοποιούμε μέσω του πεδίου του στον editor, για να αναζητήσουμε τη σχετική εγγραφή στη βάση δεδομένων.

Τα URL, χρησιμοποιούνται επίσης και για την επικοινωνία με τη σελίδα views, καλώντας τη συνάρτηση που χρησιμοποιείται για την υλοποίηση του συγκεκριμένου URL. Τη σελίδα views θα την αναλύσουμε παρακάτω.

Εδώ τα συγκεκριμένα βήματα δημιουργίας των URL, είναι τα εξής:

Βήμα 1^ο: Πηγαίνουμε στην αριστερή καρτέλα του προγράμματος Visual Studio Code. Βρίσκουμε τον φάκελο Ptyxiaki, τον οποίο δημιουργήσαμε κατά την εγκατάσταση του προγράμματος. Εκεί θα βρούμε τη σελίδα urls.py.



Εικόνα 38. Σελίδα URL

Βήμα 2^ο:

```

from django.contrib import admin
from django.urls import path
from FlightManagementSystem import views

urlpatterns = [
    path("ΌνομαΙστοσελίδας/", admin.site.urls),
    path('όνομαΙστοσελίδας/<str:όνομαΑπαραίτητουΠεδίου>',views.όνομαΣυνάρτησης),
]

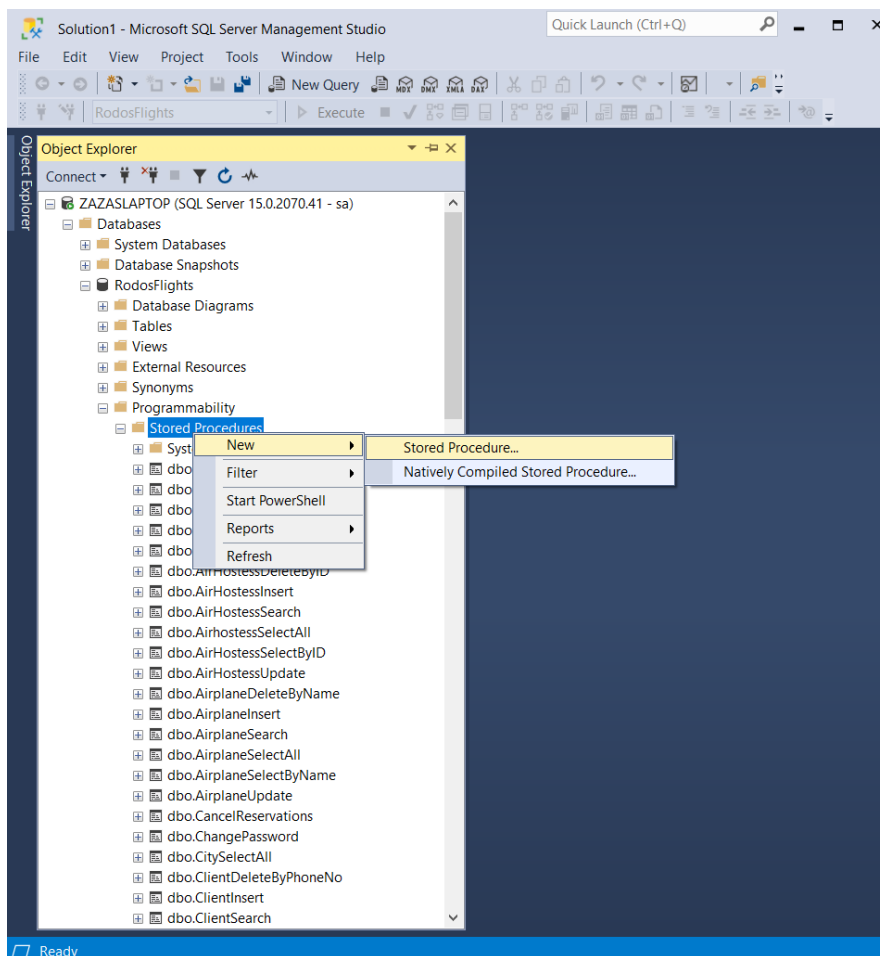
```

*Εικόνα 39. Βασική σύνταξη σελίδας urls***3.2.8 Δημιουργία stored procedures στη βάση δεδομένων**

Τα stored procedures, είναι σελίδες πού περιέχουν εντολές και συναρτήσεις, οι οποίες διασυνδέονται με τον editor, μέσω του Django framework. Προϋποθέτουν συγκεκριμένες ενέργειες για την επεξεργασία της εγγραφής, ανάλογα με τα ζητούμενά μας (εισαγωγή, διαγραφή, ανανέωση, έλεγχος στοιχείων).

Εδώ τα συγκεκριμένα βήματα δημιουργίας των stored procedures, είναι τα εξής:

Βήμα 1^ο: Ανοίγουμε το Microsoft SQL Server Management Studio. Βρίσκουμε τη βάση δεδομένων RodosFlights > Programmability > Stored Procedures > δεξί κλικ > New > Stored Procedure...

*Εικόνα 40. Δημιουργία stored procedure Βήμα 1ο*

Βήμα 2^ο: Στη σελίδα πού δημιουργήσαμε, θα πρέπει να δώσουμε:

- Το όνομα του stored procedure
- Τις παραμέτρους αν υπάρχουν, αλλιώς τις διαγράφουμε
- Τις μεταβλητές, πού θα σταλούν από τη βάση στον editor, αφού τις επεξεργαστούμε σύμφωνα με τα ζητούμενά της εργασίας

```

-- =====
-- Template generated from Template Explorer using:
-- Create Procedure (New Menu).SQL
--
-- Use the Specify Values for Template Parameters
-- command (Ctrl-Shift-M) to fill in the parameter
-- values below.
--
-- This block of comments will not be included in
-- the definition of the procedure.
-- =====
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
CREATE PROCEDURE <Procedure_Name, sysname, ProcedureName>
-- Add the parameters for the stored procedure here
<@Param1, sysname, @p1> <Datatype_For_Param1, , int> = <Default_Value_For_Param1, , 0>,
<@Param2, sysname, @p2> <Datatype_For_Param2, , int> = <Default_Value_For_Param2, , 0>
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

-- Insert statements for procedure here
SELECT <@Param1, sysname, @p1>, <@Param2, sysname, @p2>
END
GO
    
```

Εικόνα 41. Δημιουργία stored procedure Βήμα 2^ο

Για να εκτελέσουμε και να αποθηκεύσουμε το stored procedure στη βάση δεδομένων, πατάμε το κουμπί **Execute** και εάν δεν υπάρξουν συντακτικά λάθη, η εκτέλεση και αποθήκευση έγινε επιτυχώς.

3.2.9 Δημιουργία views

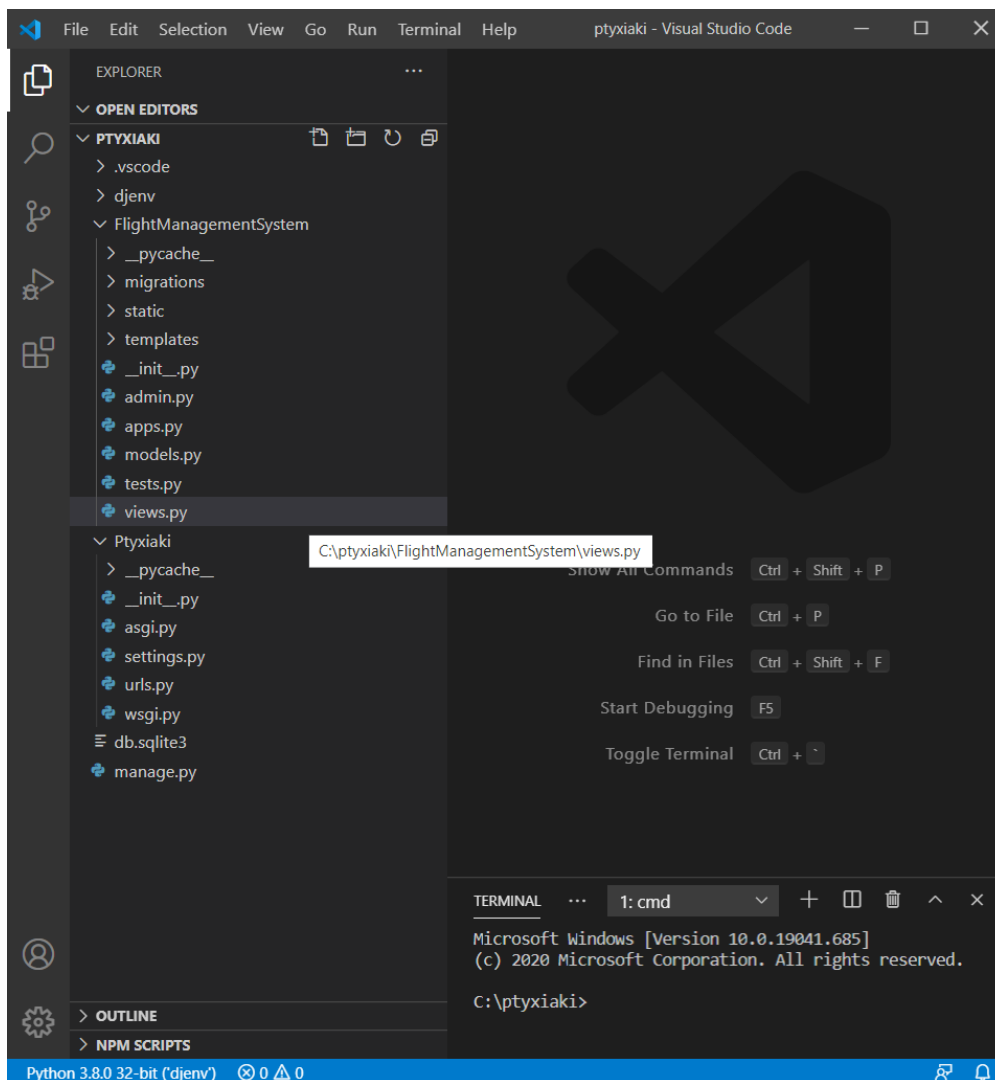
Πρόκειται για τη σελίδα με τις περισσότερες διασυνδέσεις όλης της εφαρμογής. Δηλαδή, περιέχει εντολές πού συσχετίζονται με τους πίνακες και τα stored procedures της βάσης δεδομένων και των σελίδων html, χρησιμοποιώντας αντικείμενα τύπου JSON, βιβλιοθήκες έτοιμες του Django framework και τα μοντέλα των εγγραφών.

Σύμφωνα με τους πίνακες και τα stored procedures, πού δημιουργήσαμε για κάθε εγγραφή στη βάση δεδομένων, δημιουργούμε και τις αντίστοιχες συναρτήσεις στη σελίδα

views.

Εδώ τα συγκεκριμένα βήματα δημιουργίας του view μιας εγγραφής, είναι τα εξής:

Βήμα 1^ο: Πηγαίνουμε στην αριστερή καρτέλα του προγράμματος Visual Studio Code. Βρίσκουμε τον φάκελο FlightManagementSystem, τον οποίο δημιουργήσαμε κατά την εγκατάσταση του προγράμματος. Εκεί θα βρούμε τη σελίδα views.py.



Εικόνα 42. Σελίδα views

Βήμα 2^ο: Ανοίγουμε το Microsoft SQL Server Management Studio, πηγαίνουμε στα stored procedures της εγγραφής που θέλουμε, και με βάση τα στοιχεία της εγγραφής, γράφουμε τις κατάλληλες εντολές στη σελίδα views, για να γίνει η διασύνδεση μεταξύ του editor και της βάσης δεδομένων, καθώς και οι απαραίτητες ενέργειες της εγγραφής (όπως η ενημέρωση των στοιχείων ενός πελάτη).

```

from django.shortcuts import render
from django.views.decorators.csrf import csrf_exempt
from django.db import connection
import json
from django.http import JsonResponse
from .models import Elements

@csrf_exempt
def όνομαΣυνάρτησης(request):
    όνομαΠεδίουΜεΠρωτεύονΚλειδί = request.POST['όνομαΠεδίουΜεΠρωτεύονΚλειδίΣτηΒάση']
    διαφορετικόΌνομαΠεδίουΜεΠρωτεύονΚλειδί = request.POST['διαφορετικόΌνομαΠεδίουΜεΠρωτεύονΚλειδίΣτηΒάση']
    όνομαΆλλουΠεδίου = request.POST['όνομαΆλλουΠεδίουΣτηΒάση']

    if διαφορετικόΌνομαΠεδίουΜεΠρωτεύονΚλειδί == '':
        with connection.cursor() as cursor:
            cursor.execute('exec dbo.ΌνομαStoredProcedureΕισαγωγήςΣτοιχείωνΕγγραφής @όνομαΠεδίουΜεΠρωτεύονΚλειδίΣτηΒάση=%s,
[όνομαΠεδίουΜεΠρωτεύονΚλειδί,όνομαΆλλουΠεδίου])
    else:
        with connection.cursor() as cursor:
            cursor.execute('exec dbo.ΌνομαStoredProcedureΑνανέωσηςΣτοιχείωνΕγγραφής @διαφορετικόΌνομαΠεδίουΜεΠρωτεύονΚλειδίΣτηΒάση=%s,
@όνομαΠεδίουΜεΠρωτεύονΚλειδίΣτηΒάση=%s, @όνομαΆλλουΠεδίουΣτηΒάση=%s',
[διαφορετικόΌνομαΠεδίουΜεΠρωτεύονΚλειδί,όνομαΠεδίουΜεΠρωτεύονΚλειδί,όνομαΆλλουΠεδίου])

    return JsonResponse(json.dumps(True),safe=False)

```

Εικόνα 43. Δημιουργία view

Η παραπάνω συνάρτηση, εκτελεί την εισαγωγή των στοιχείων της εγγραφής στη βάση δεδομένων, παίρνοντας τις τιμές των πεδίων της, εάν το πεδίο της εγγραφής με μοναδική τιμή (πρωτεύον κλειδί στη βάση δεδομένων) είναι κενό (δεν υπάρχει η εγγραφή). Αλλιώς αν δεν είναι κενό (υπάρχει η εγγραφή), γίνεται η ανανέωση των στοιχείων της εγγραφής και φυσικά η τιμή του πεδίου με πρωτεύον κλειδί δεν μεταβάλλεται.

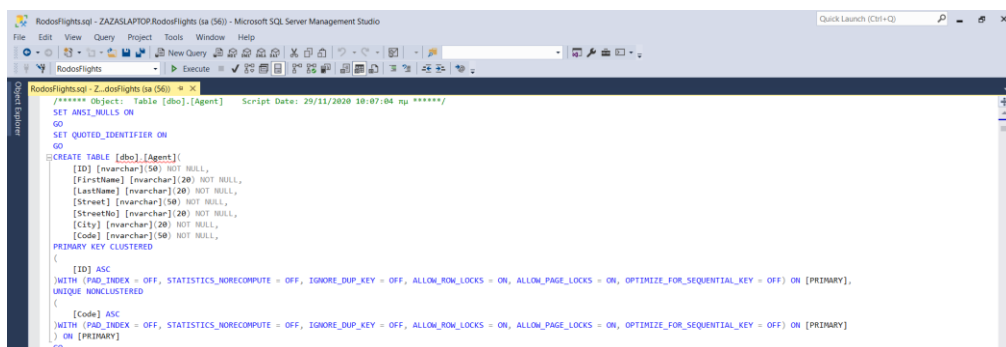
Παρατηρούμε επίσης, ότι χρησιμοποιείται η εντολή @csrf_exempt. Η συγκεκριμένη εντολή, χρησιμοποιείται για να επιτραπεί η εκτέλεση της συνάρτησης, χωρίς να γίνει η χρήση των cookies στην ιστοσελίδα της εφαρμογής, για λόγους ασφαλείας.

3.3 Υλοποίηση

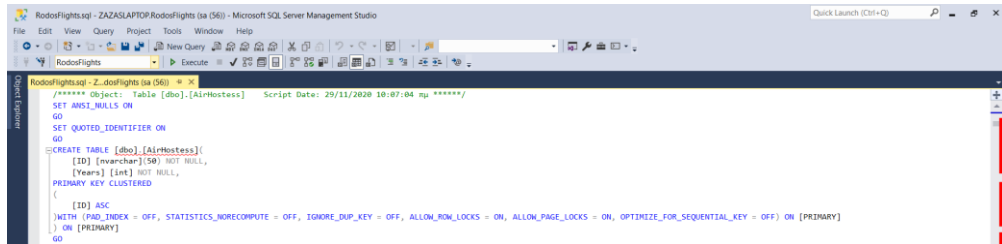
Παρακάτω, θα δούμε την υλοποίηση όλης της εφαρμογής με τη χρήση κώδικα στα προγράμματα Microsoft SQL Server Management Studio και Visual Studio Code.

3.3.1 Microsoft SQL Server Management Studio

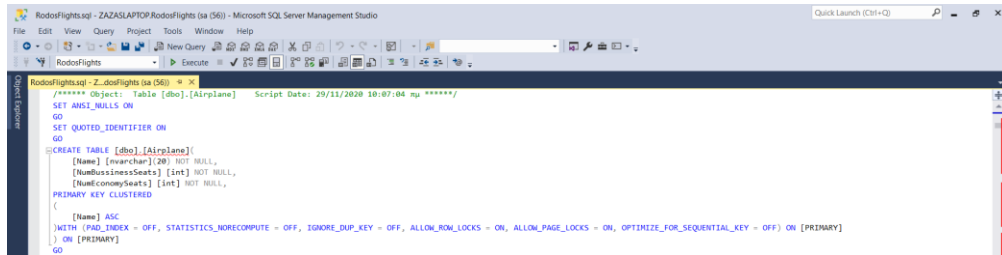
Δημιουργία πινάκων



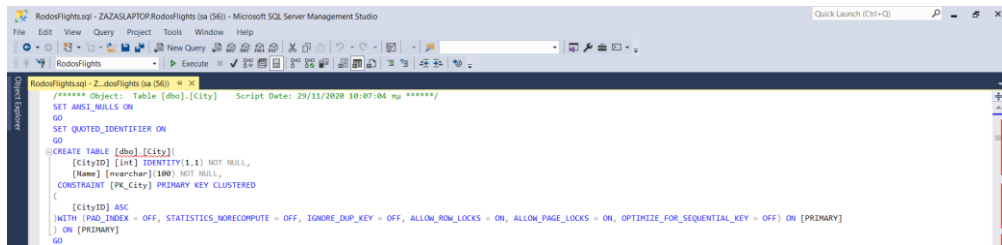
Εικόνα 44. Δημιουργία πίνακα ταξιδιωτικών πρακτόρων



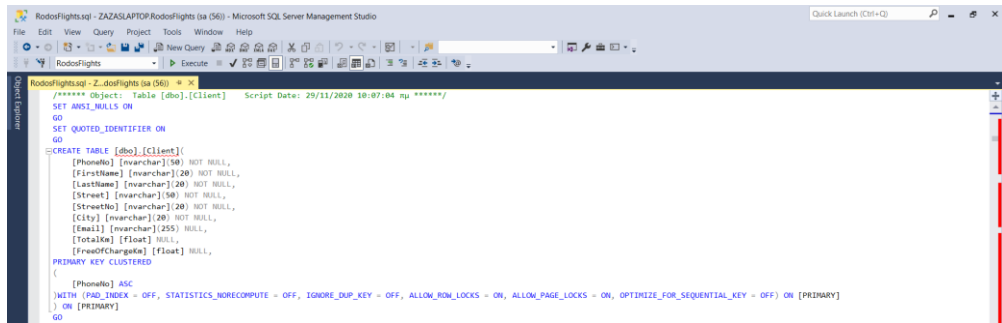
Εικόνα 45. Δημιουργία πίνακα αεροσυνοδών



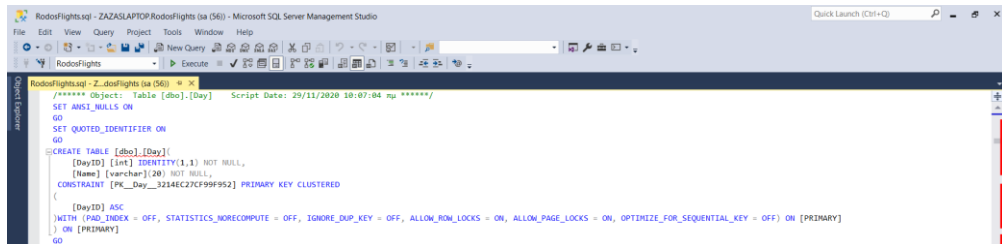
Εικόνα 46. Δημιουργία πίνακα αεροπλάνων



Εικόνα 47. Δημιουργία πίνακα πόλεων



Εικόνα 48. Δημιουργία πίνακα πελατών



Εικόνα 49 Δημιουργία πίνακα μερών


```
***** Object: Table [dbo].[Employee] Script Date: 29/11/2020 10:07:04 μμ *****  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
CREATE TABLE [dbo].[Employee](  
[ID] [nvarchar](50) NOT NULL,  
[FirstName] [nvarchar](20) NOT NULL,  
[LastName] [nvarchar](20) NOT NULL,  
[Street] [nvarchar](50) NOT NULL,  
[StreetNo] [nvarchar](20) NOT NULL,  
[City] [nvarchar](20) NOT NULL,  
[PhoneNo] [nvarchar](50) NOT NULL,  
[BirthDate] [date] NOT NULL,  
PRIMARY KEY CLUSTERED  
(  
[ID] ASC  
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]  
GO
```

Εικόνα 50. Δημιουργία πίνακα υπαλλήλων

```
***** Object: Table [dbo].[Manager] Script Date: 29/11/2020 10:07:04 μμ *****  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
CREATE TABLE [dbo].[Manager](  
[ID] [nvarchar](50) NOT NULL,  
[Education] [nvarchar](20) NOT NULL,  
[Years] [int] NOT NULL,  
PRIMARY KEY CLUSTERED  
(  
[ID] ASC  
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]  
GO
```

Εικόνα 51. Δημιουργία πίνακα διοικητικών υπαλλήλων

```
***** Object: Table [dbo].[Pilot] Script Date: 29/11/2020 10:07:04 μμ *****  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
CREATE TABLE [dbo].[Pilot](  
[ID] [nvarchar](50) NOT NULL,  
[FlightHours] [int] NOT NULL,  
PRIMARY KEY CLUSTERED  
(  
[ID] ASC  
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]  
GO
```

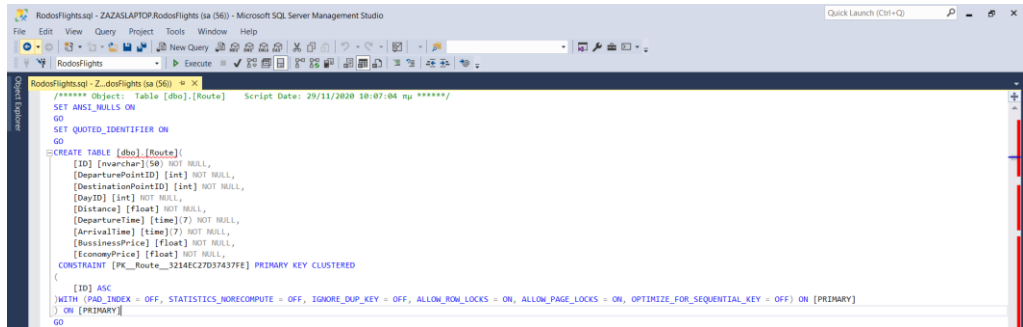
Εικόνα 52. Δημιουργία πίνακα πιλότων

```
***** Object: Table [dbo].[Reservation] Script Date: 29/11/2020 10:07:04 μμ *****  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
CREATE TABLE [dbo].[Reservation](  
[ReservationID] [int] IDENTITY(1,1) NOT NULL,  
[PhoneNo] [nvarchar](50) NOT NULL,  
[ReservationDate] [date] NOT NULL,  
[ExpirationDate] [date] NOT NULL,  
[BusinessClass] [bit] NOT NULL,  
[AgentID] [nvarchar](50) NOT NULL,  
[DepartureCityID] [int] NOT NULL,  
[ArrivalCityID] [int] NOT NULL,  
CONSTRAINT [PK_Reservation] PRIMARY KEY CLUSTERED  
(  
[ReservationID] ASC  
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]  
GO
```

Εικόνα 53. Δημιουργία πίνακα κρατήσεων

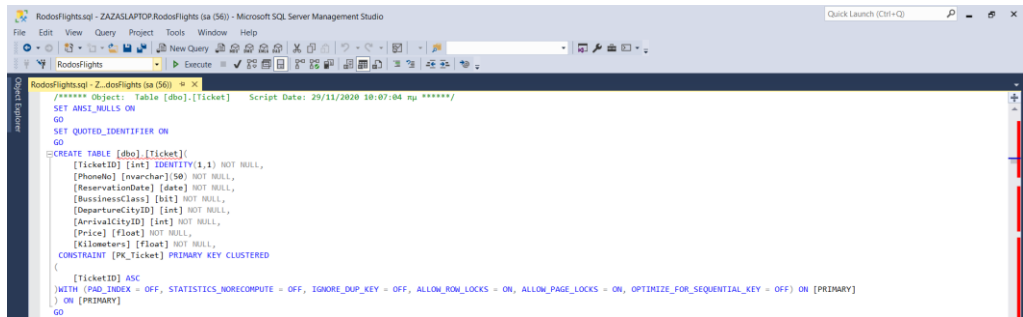
```
***** Object: Table [dbo].[ReservationToTransport] Script Date: 29/11/2020 10:07:04 μμ *****  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
CREATE TABLE [dbo].[ReservationToTransport](  
[ReservationID] [int] NOT NULL,  
[TransportID] [int] NOT NULL,  
[TransportIndex] [int] NOT NULL,  
CONSTRAINT [PK_ReservationToTransport] PRIMARY KEY CLUSTERED  
(  
[ReservationID] ASC,  
[TransportID] ASC  
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]  
GO
```

Εικόνα 54. Δημιουργία πίνακα κρατήσεων σε μεταφορές



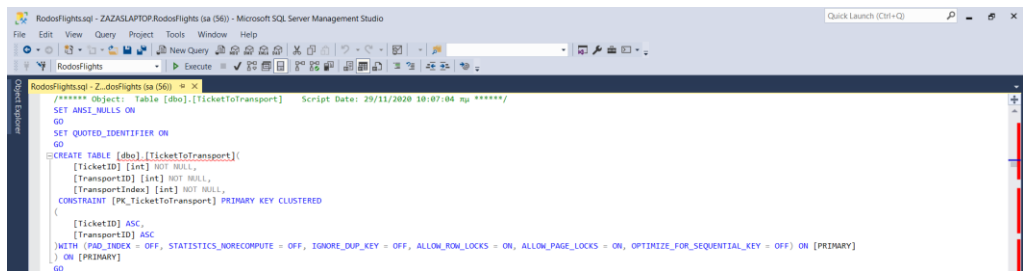
```
***** Object: Table [dbo].[Route] Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Route](
    [ID] [nvarchar](50) NOT NULL,
    [DeparturePointID] [int] NOT NULL,
    [DestinationPointID] [int] NOT NULL,
    [DayID] [int] NOT NULL,
    [Distance] [float] NOT NULL,
    [DepartureTime] [time](7) NOT NULL,
    [ArrivalTime] [time](7) NOT NULL,
    [BusinessPrice] [float] NOT NULL,
    [EconomyPrice] [float] NOT NULL,
    CONSTRAINT [PK_Route_3214EC27037437FE] PRIMARY KEY CLUSTERED
    (
        [ID] ASC
    ) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
```

Εικόνα 55. Δημιουργία πίνακα δρομολογίων



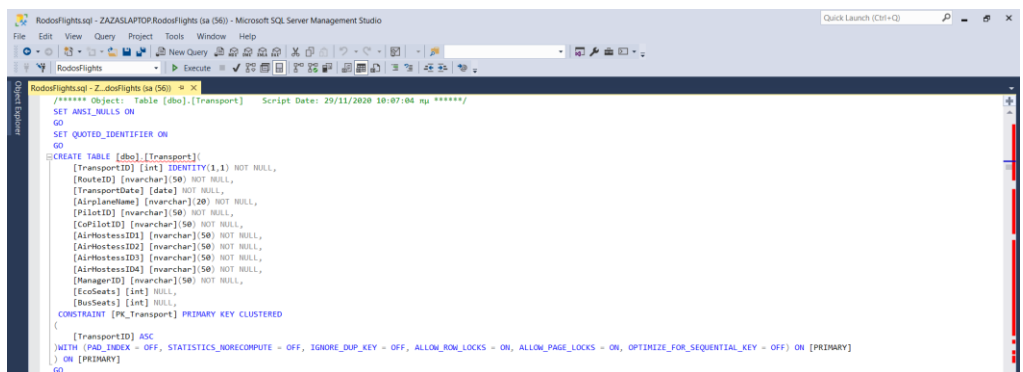
```
***** Object: Table [dbo].[Ticket] Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Ticket](
    [TicketID] [int] IDENTITY(1,1) NOT NULL,
    [PhoneNo] [nvarchar](50) NOT NULL,
    [ReservationDate] [date] NOT NULL,
    [BusinessClass] [bit] NOT NULL,
    [DepartureCityID] [int] NOT NULL,
    [ArrivalCityID] [int] NOT NULL,
    [Price] [float] NOT NULL,
    [Kilometers] [float] NOT NULL,
    CONSTRAINT [PK_Ticket] PRIMARY KEY CLUSTERED
    (
        [TicketID] ASC
    ) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
```

Εικόνα 56. Δημιουργία πίνακα εισιτηρίων



```
***** Object: Table [dbo].[TicketToTransport] Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[TicketToTransport](
    [TicketID] [int] NOT NULL,
    [TransportID] [int] NOT NULL,
    [TransportIndex] [int] NOT NULL,
    CONSTRAINT [PK_TicketToTransport] PRIMARY KEY CLUSTERED
    (
        [TicketID] ASC,
        [TransportID] ASC
    ) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
```

Εικόνα 57. Δημιουργία πίνακα εισιτηρίων σε μεταφορές



```
***** Object: Table [dbo].[Transport] Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Transport](
    [TransportID] [int] IDENTITY(1,1) NOT NULL,
    [RouteID] [nvarchar](50) NOT NULL,
    [TransportDate] [date] NOT NULL,
    [AirplaneName] [nvarchar](20) NOT NULL,
    [PilotID] [nvarchar](50) NOT NULL,
    [COPilotID] [nvarchar](50) NOT NULL,
    [AirHostessID1] [nvarchar](50) NOT NULL,
    [AirHostessID2] [nvarchar](50) NOT NULL,
    [AirHostessID3] [nvarchar](50) NOT NULL,
    [AirHostessID4] [nvarchar](50) NOT NULL,
    [ManagerID] [nvarchar](50) NOT NULL,
    [EcoSeats] [int] NULL,
    [BusSeats] [int] NULL,
    CONSTRAINT [PK_Transport] PRIMARY KEY CLUSTERED
    (
        [TransportID] ASC
    ) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
```

Εικόνα 58 Δημιουργία πίνακα μεταφορών

```

***** Object: Table [dbo].[User] Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[User](
    [Username] [nvarchar](50) NOT NULL,
    [ClientID] [nvarchar](50) NULL,
    [AgentID] [nvarchar](50) NULL,
    [AdministratorID] [nvarchar](50) NULL,
    [Password] [nvarchar](max) NOT NULL,
    CONSTRAINT [PK_User_536C8E527F64368] PRIMARY KEY CLUSTERED
    (
        [Username] ASC
    )
    WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
    
```

Εικόνα 59. Δημιουργία πίνακα χρηστών

Δημιουργία αποθηκευμένων διαδικασιών

```

***** Object: StoredProcedure [dbo].[AgentDeleteByID] Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[AgentDeleteByID]
    @ID nvarchar(50)
)
as
begin
    delete
    from
        [Agent]
    where
        [ID]=@ID;

    delete
    from
        [User]
    where
        [AgentID] = @ID;
end;
GO
    
```

Εικόνα 60. Αποθηκευμένη διαδικασία διαγραφής ταξιδιωτικού πράκτορα

```

***** Object: StoredProcedure [dbo].[AgentInsert] Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[AgentInsert]
    (
        @ID nvarchar(50),
        @FirstName nvarchar(20),
        @LastName nvarchar(20),
        @Street nvarchar(50),
        @StreetNo nvarchar(20),
        @City nvarchar(20),
        @Code nvarchar(50)
    )
as
begin
    insert into [Agent]([ID],[FirstName],[LastName],[Street],[StreetNo],[City],[Code])
    values(@ID,@FirstName,@LastName,@Street,@StreetNo,@City,@Code);
end;
GO
    
```

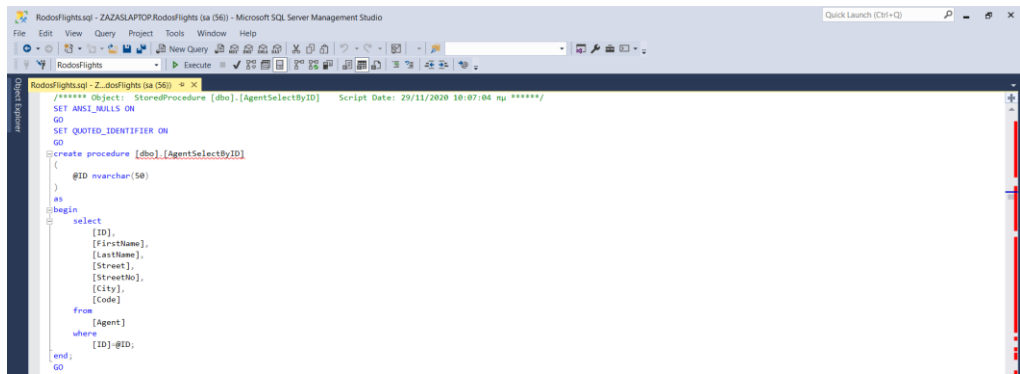
Εικόνα 61. Αποθηκευμένη διαδικασία εισαγωγής ταξιδιωτικού πράκτορα

```

***** Object: StoredProcedure [dbo].[AgentSearch] Script Date: 29/11/2020 10:07:04 μμ *****
-- Author: Zacharoulis
-- Create Date: 29/11/2020
-- Description: <Description>
CREATE PROCEDURE [dbo].[AgentSearch]
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

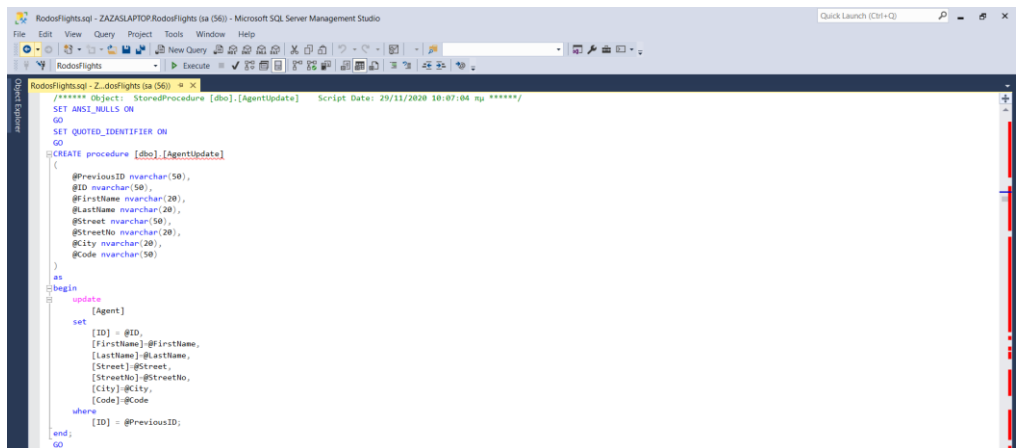
    select
        ID,
        FirstName,
        LastName,
        Street,
        StreetNo,
        City,
        Code
    from
        [Agent]
END
GO
    
```

Εικόνα 62. Αποθηκευμένη διαδικασία αναζήτησης ταξιδιωτικού πράκτορα



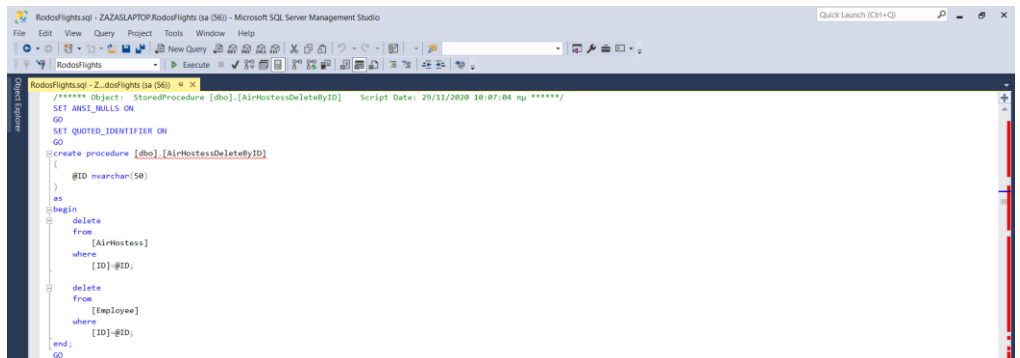
```
***** Object: StoredProcedure [dbo].[AgentSelectByID] Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[AgentSelectByID]
(
    @ID nvarchar(50)
)
as
begin
    select
        [ID],
        [FirstName],
        [LastName],
        [Street],
        [StreetNo],
        [City],
        [Code]
    from
        [Agent]
    where
        [ID]=@ID;
end;
GO
```

Εικόνα 63. Αποθηκευμένη διαδικασία επιλογής ταξιδιωτικού πράκτορα



```
***** Object: StoredProcedure [dbo].[AgentUpdate] Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[AgentUpdate]
(
    @PreviousID nvarchar(50),
    @ID nvarchar(50),
    @FirstName nvarchar(20),
    @LastName nvarchar(20),
    @Street nvarchar(50),
    @StreetNo nvarchar(20),
    @City nvarchar(20),
    @Code nvarchar(50)
)
as
begin
    update
        [Agent]
    set
        [ID] = @ID,
        [FirstName]=@FirstName,
        [LastName]=@LastName,
        [Street]=@Street,
        [StreetNo]=@StreetNo,
        [City]=@City,
        [Code]=@Code
    where
        [ID] = @PreviousID;
end;
GO
```

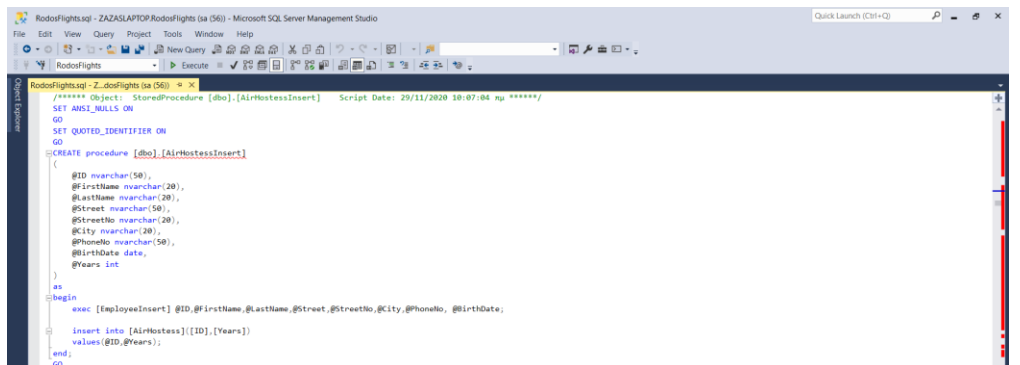
Εικόνα 64. Αποθηκευμένη διαδικασία ανανέωσης ταξιδιωτικού πράκτορα



```
***** Object: StoredProcedure [dbo].[AirHostessDeleteByID] Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[AirHostessDeleteByID]
(
    @ID nvarchar(50)
)
as
begin
    delete
        from
        [AirHostess]
    where
        [ID]=@ID;

    delete
        from
        [Employee]
    where
        [ID]=@ID;
end;
GO
```

Εικόνα 65. Αποθηκευμένη διαδικασία διαγραφής αεροσυνοδού



```
***** Object: StoredProcedure [dbo].[AirHostessInsert] Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[AirHostessInsert]
(
    @ID nvarchar(50),
    @FirstName nvarchar(20),
    @LastName nvarchar(20),
    @Street nvarchar(50),
    @StreetNo nvarchar(20),
    @City nvarchar(20),
    @PhoneNo nvarchar(50),
    @BirthDate date,
    @Years int
)
as
begin
    exec [EmployeeInsert] @ID,@FirstName,@LastName,@Street,@StreetNo,@City,@PhoneNo, @BirthDate;

    insert into [AirHostess]([ID],[Years])
    values(@ID,@Years);
end;
GO
```

Εικόνα 66. Αποθηκευμένη διαδικασία εισαγωγής αεροσυνοδού

```

Object Explorer
RodosFlights.sql - ZAZASLAPIOR.RodosFlights (sa (56)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
RodosFlights
Execute
Object Explorer
RodosFlights.sql - Z_dosFlights (sa (56))
Object Explorer
***** Object: StoredProcedure [dbo].[AirHostessSearch] Script Date: 29/11/2020 18:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
--
-- Author: Zacharoula
-- Create date: 2/5/2020
-- Description: <Description>
--
CREATE PROCEDURE [dbo].[AirHostessSearch]
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

select
empl.ID,
empl.FirstName,
empl.LastName,
empl.Street,
empl.StreetNo,
empl.City,
empl.PhoneNo,
empl.BirthDate,
airh.Years
from
[Employee] empl inner join
[AirHostess] airh on airh.[ID] = empl.[ID];
END
GO

```

Εικόνα 67. Αποθηκευμένη διαδικασία αναζήτησης αεροσυνοδού

```

Object Explorer
RodosFlights.sql - ZAZASLAPIOR.RodosFlights (sa (56)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
RodosFlights
Execute
Object Explorer
RodosFlights.sql - Z_dosFlights (sa (56))
Object Explorer
***** Object: StoredProcedure [dbo].[AirHostessSelectAll] Script Date: 29/11/2020 18:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
--
-- Author: <Author,Name>
-- Create date: <Create Date,>
-- Description: <Description,>
--
CREATE PROCEDURE [dbo].[AirHostessSelectAll]
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

select
a.[ID],
a.[FirstName] + ' ' + a.[LastName]
from
[AirHostess] a inner join
[Employee] e on e.ID = a.ID;
END
GO

```

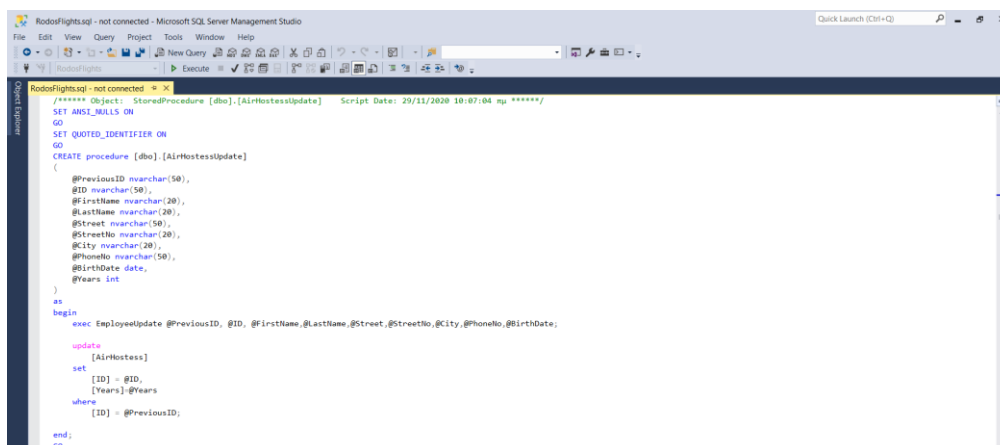
Εικόνα 68. Αποθηκευμένη διαδικασία επιλογής όλων των αεροσυνοδών

```

Object Explorer
RodosFlights.sql - not connected - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
RodosFlights
Execute
Object Explorer
RodosFlights.sql - not connected
Object Explorer
***** Object: StoredProcedure [dbo].[AirHostessSelectByID] Script Date: 29/11/2020 18:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[AirHostessSelectByID]
(
@ID nvarchar(50)
)
as
begin
select
empl.[ID],
empl.[FirstName],
empl.[LastName],
empl.[Street],
empl.[StreetNo],
empl.[City],
empl.[PhoneNo],
empl.[BirthDate],
airh.[Years]
from
[Employee] empl inner join
[AirHostess] airh on airh.[ID]=empl.[ID]
where
airh.[ID]=@ID;
end;
GO

```

Εικόνα 69. Αποθηκευμένη διαδικασία επιλογής αεροσυνοδού



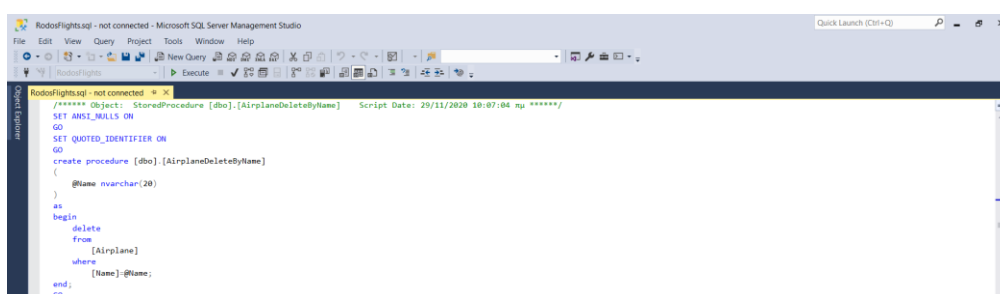
```

/***** Object: StoredProcedure [dbo].[AirHostessUpdate]  Script Date: 29/11/2020 10:07:04 μμ *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[AirHostessUpdate]
(
    @PreviousID nvarchar(50),
    @ID nvarchar(50),
    @FirstName nvarchar(20),
    @LastName nvarchar(20),
    @Street nvarchar(50),
    @StreetNo nvarchar(20),
    @City nvarchar(50),
    @PhoneNo nvarchar(50),
    @BirthDate date,
    @Years int
)
as
begin
    exec EmployeeUpdate @PreviousID, @ID, @FirstName, @LastName, @Street, @StreetNo, @City, @PhoneNo, @BirthDate;

    update
    [AirHostess]
    set
    [ID] = @ID,
    [Years] = @Years
    where
    [ID] = @PreviousID;
end;
GO

```

Εικόνα 70. Αποθηκευμένη διαδικασία ανανέωσης αεροσυνοδού

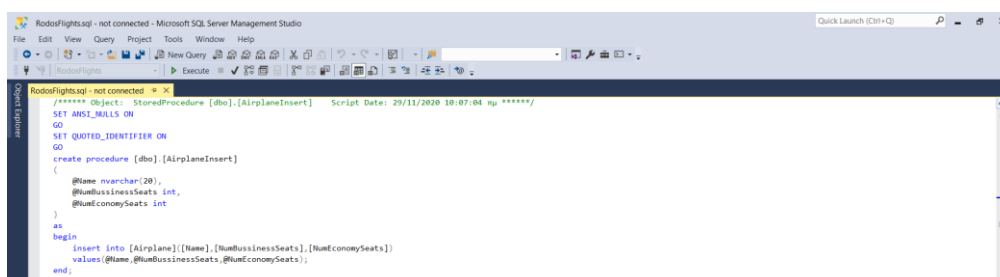


```

/***** Object: StoredProcedure [dbo].[AirplaneDeleteByName]  Script Date: 29/11/2020 10:07:04 μμ *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[AirplaneDeleteByName]
(
    @Name nvarchar(20)
)
as
begin
    delete
    from
    [Airplane]
    where
    [Name] = @Name;
end;
GO

```

Εικόνα 71. Αποθηκευμένη διαδικασία διαγραφής αεροπλάνου

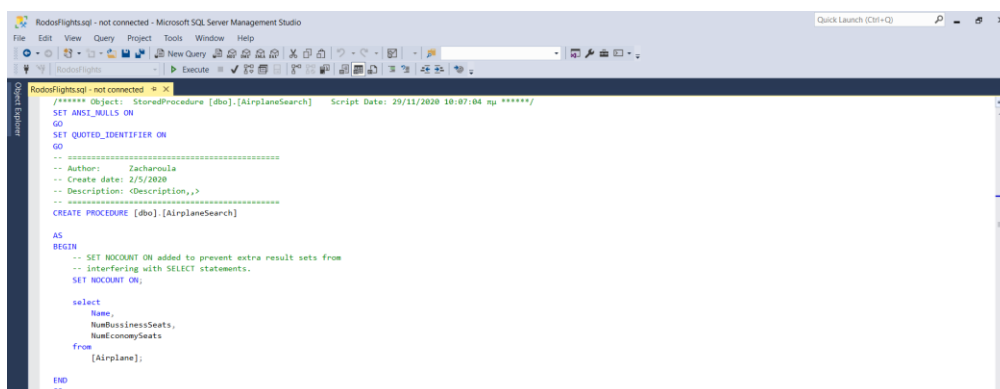


```

/***** Object: StoredProcedure [dbo].[AirplaneInsert]  Script Date: 29/11/2020 10:07:04 μμ *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[AirplaneInsert]
(
    @Name nvarchar(20),
    @NumBusinessSeats int,
    @NumEconomySeats int
)
as
begin
    insert into [Airplane]([Name],[NumBusinessSeats],[NumEconomySeats])
    values(@Name,@NumBusinessSeats,@NumEconomySeats);
end;
GO

```

Εικόνα 72. Αποθηκευμένη διαδικασία εισαγωγής αεροπλάνου



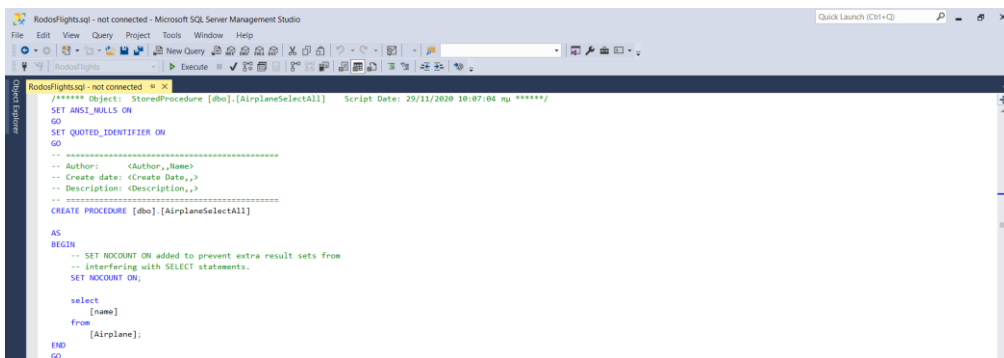
```

/***** Object: StoredProcedure [dbo].[AirplaneSearch]  Script Date: 29/11/2020 10:07:04 μμ *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author: Zacharoula
-- Create date: 2/5/2020
-- Description: <Description,>
-- =====
CREATE PROCEDURE [dbo].[AirplaneSearch]
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    select
    Name,
    NumBusinessSeats,
    NumEconomySeats
    from
    [Airplane];
END
GO

```

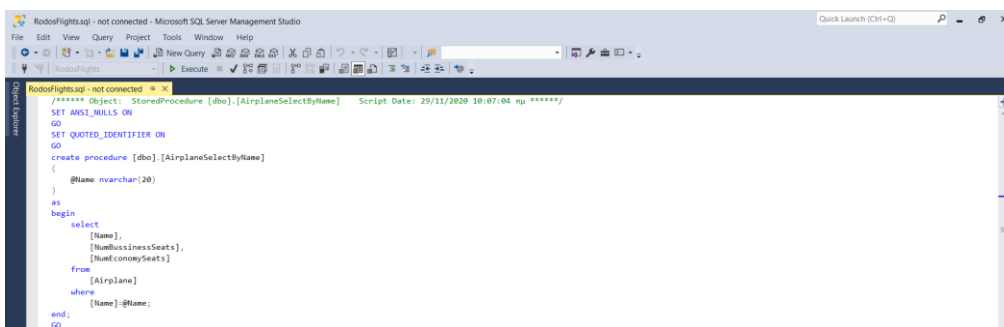
Εικόνα 73. Αποθηκευμένη διαδικασία αναζήτησης αεροπλάνου



```
***** Object: StoredProcedure [dbo].[AirplaneSelectAll] Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-----
-- Author: <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-----
CREATE PROCEDURE [dbo].[AirplaneSelectAll]
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

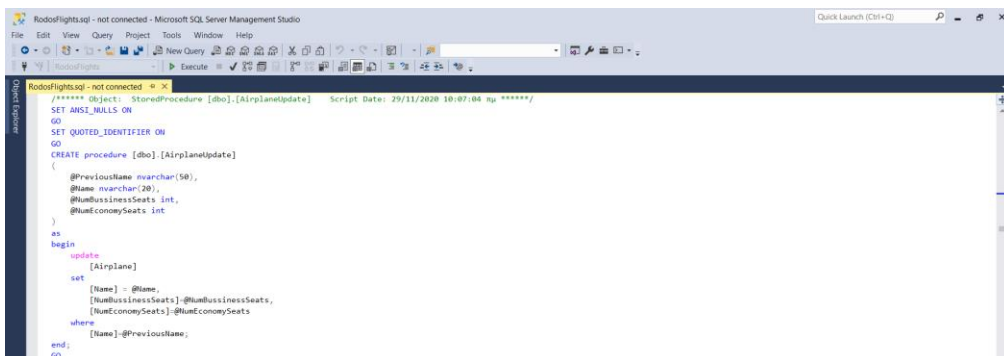
select
    [name]
from
    [Airplane];
END
GO
```

Εικόνα 74. Αποθηκευμένη διαδικασία επιλογής όλων των αεροπλάνων



```
***** Object: StoredProcedure [dbo].[AirplaneSelectByName] Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[AirplaneSelectByName]
(
    @Name nvarchar(20)
)
as
begin
select
    [Name],
    [NumBusinessSeats],
    [NumEconomySeats]
from
    [Airplane]
where
    [Name]=@Name;
end;
GO
```

Εικόνα 75. Αποθηκευμένη διαδικασία επιλογής αεροπλάνου



```
***** Object: StoredProcedure [dbo].[AirplaneUpdate] Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[AirplaneUpdate]
(
    @PreviousName nvarchar(50),
    @Name nvarchar(20),
    @NumBusinessSeats int,
    @NumEconomySeats int
)
as
begin
update
    [Airplane]
set
    [Name] = @Name,
    [NumBusinessSeats]=@NumBusinessSeats,
    [NumEconomySeats]=@NumEconomySeats
where
    [Name]=@PreviousName;
end;
GO
```

Εικόνα 76. Αποθηκευμένη διαδικασία ανανέωσης αεροπλάνου

```

/***** Object: StoredProcedure [dbo].[CancelReservations]  Script Date: 29/11/2020 10:07:04 μμ *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author: Zacharia
-- Create date: <Create Date>,
-- Description: <Description>,
-- =====
CREATE PROCEDURE [dbo].[CancelReservations]
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

-- δημιουργούμε έναν όρομνα για να πάρουμε όλες τις κρατήσεις που λήγουν την τρέχουσα ημερομηνία. Συγκεκριμένα για κάθε κράτηση φέρνουμε το ID και το flag που δείχνει εάν είναι business class.
declare c cursor for
select
    [ReservationID],
    [BusinessClass]
from
    [Reservation]
where
    [ExpirationDate] = cast(getdate() as date);

declare @resID int;
declare @busClass bit;
declare @currentTransportID1 int;
declare @currentTransportID2 int;

open c; -- άνοιξε τον όρομνα

-- φέρν την πρώτη εγγραφή από το query και βάλε το ReservationID στη μεταβλητή @resID
fetch next from c
into @resID, @busClass;

while @@FETCH_STATUS = 0 -- όσο υπάρχουν εγγραφές
begin
    set @currentTransportID1 = (select [TransportID] from [ReservationToTransport] where [ReservationID] = @resID and [TransportIndex] = 1);
    set @currentTransportID2 = (select [TransportID] from [ReservationToTransport] where [ReservationID] = @resID and [TransportIndex] = 2);

    if @busClass = 1 -- ελέγχουμε εάν η τρέχουσα κατηγορία θέσης είναι η business
    begin
        -- αυξάνω κατά ένα τις διαθέσιμες διακεκριμένες θέσεις του transport1
        update
            [Transport]
        set
            [BusSeats] = [BusSeats] + 1
        where
            [TransportID] = @currentTransportID1;

        -- αυξάνω κατά ένα τις διαθέσιμες διακεκριμένες θέσεις του transport2 σε περίπτωση που έχει οριστεί
        if not @currentTransportID2 is null
            update
                [Transport]
            set
                [BusSeats] = [BusSeats] + 1
            where
                [TransportID] = @currentTransportID2;
    end
    else
    begin
        -- αυξάνω κατά ένα τις διαθέσιμες οικονομικές θέσεις του transport1
        update
            [Transport]
        set
            [EcoSeats] = [EcoSeats] + 1
        where
            [TransportID] = @currentTransportID1;

        -- αυξάνω κατά ένα τις διαθέσιμες οικονομικές θέσεις του transport2 σε περίπτωση που έχει οριστεί
        if not @currentTransportID2 is null
            update
                [Transport]
            set
                [EcoSeats] = [EcoSeats] + 1
            where
                [TransportID] = @currentTransportID2;
    end;

    -- διαγράφουμε την κράτηση
    delete
    from
        [Reservation]
    where
        [ReservationID] = @resID;

    -- διαγράφουμε τις συσχετίσεις με τις μεταφορές
    delete
    from
        [ReservationToTransport]
    where
        [ReservationID] = @resID;

    -- ματρνουμε την επόμενη εγγραφή
    fetch next from c
    into @resID, @busClass;
end;

close c;
deallocate c;
END
GO

```

Εικόνα 77. Αποθηκευμένη διαδικασία ακύρωσης κρατήσεων

```

/***** Object: StoredProcedure [dbo].[ChangePassword]  Script Date: 29/11/2020 10:07:04 μμ *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author: Zacharia
-- Create date: 4/5/2020
-- Description: <Description>,
-- =====
CREATE PROCEDURE [dbo].[ChangePassword]
    @Username nvarchar(50),
    @OldPassword nvarchar(50),
    @NewPassword nvarchar(50)
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

declare @counter int;

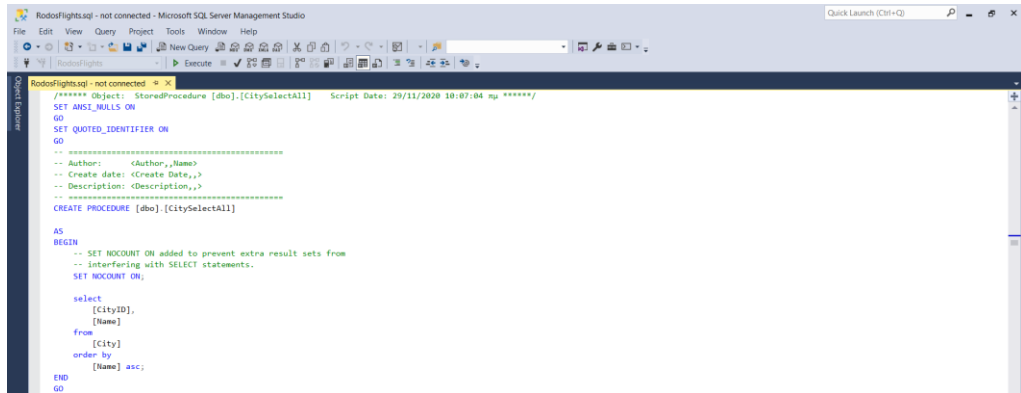
select
    @counter = count(*)
from
    [User]
where
    [Username] = @Username and
    [Password] = @OldPassword;

if @counter = 1
    update
        [User]
    set
        [Password] = @NewPassword
    where
        [Username] = @Username;

select @counter;
END
GO

```

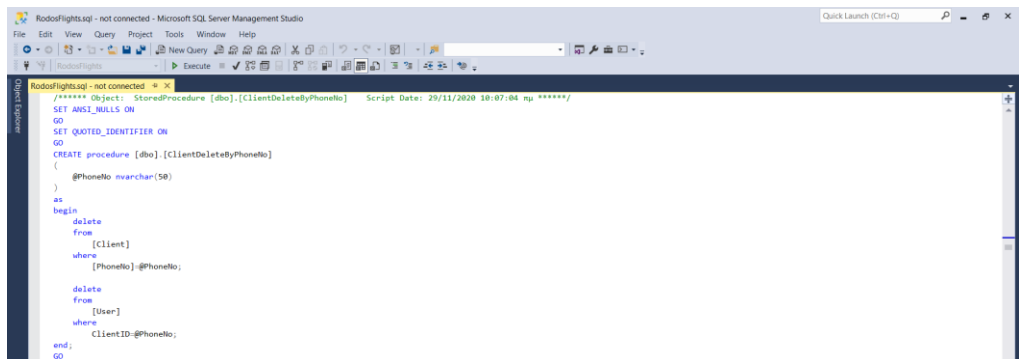
Εικόνα 78. Αποθηκευμένη διαδικασία αλλαγής κωδικού



```
***** Object: StoredProcedure [dbo].[CitySelectAll] Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
--
-- Author: <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
--
CREATE PROCEDURE [dbo].[CitySelectAll]
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

select
    [cityID],
    [Name]
from
    [City]
order by
    [Name] asc;
END
GO
```

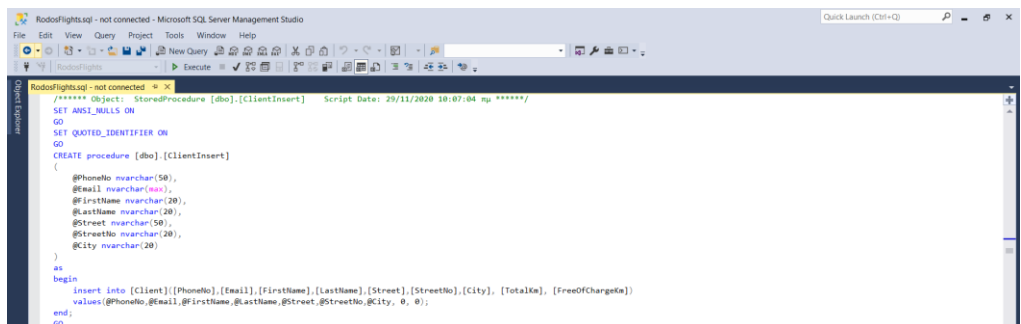
Εικόνα 79. Αποθηκευμένη διαδικασία επιλογής όλων των πόλεων



```
***** Object: StoredProcedure [dbo].[ClientDeleteByPhoneNo] Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[ClientDeleteByPhoneNo]
(
    @PhoneNo nvarchar(50)
)
as
begin
delete
from
    [Client]
where
    [PhoneNo]=@PhoneNo;

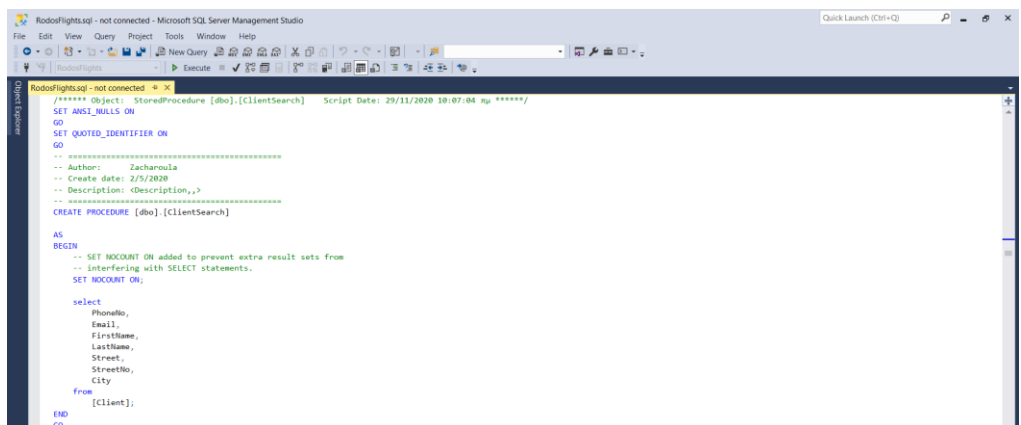
delete
from
    [User]
where
    ClientID=@PhoneNo;
end;
GO
```

Εικόνα 80. Αποθηκευμένη διαδικασία διαγραφής πελάτη



```
***** Object: StoredProcedure [dbo].[ClientInsert] Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[ClientInsert]
(
    @PhoneNo nvarchar(50),
    @Email nvarchar(50),
    @FirstName nvarchar(20),
    @LastName nvarchar(20),
    @Street nvarchar(50),
    @StreetNo nvarchar(20),
    @City nvarchar(20)
)
as
begin
insert into [Client]([PhoneNo],[Email],[FirstName],[LastName],[Street],[StreetNo],[City],[TotalKm],[FreeOfChargeKm])
values(@PhoneNo,@Email,@FirstName,@LastName,@Street,@StreetNo,@City, 0, 0);
end;
GO
```

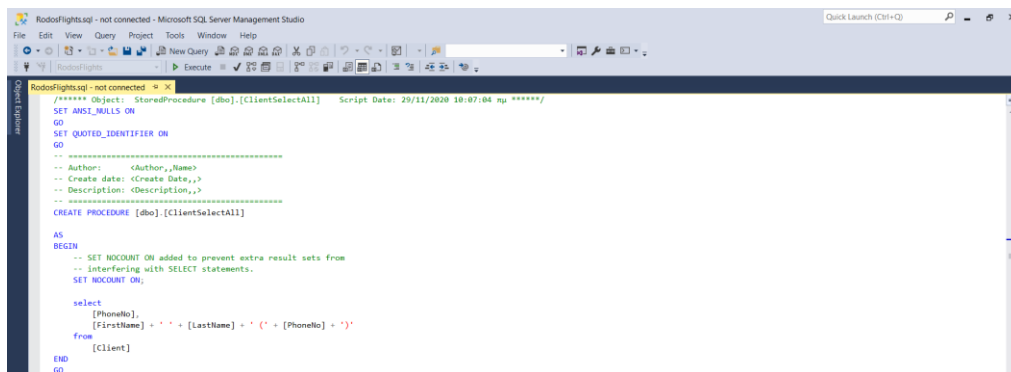
Εικόνα 81. Αποθηκευμένη διαδικασία εισαγωγής πελάτη



```
***** Object: StoredProcedure [dbo].[ClientSearch] Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
--
-- Author: Zacharoula
-- Create date: 2/5/2020
-- Description: <Description,,>
--
CREATE PROCEDURE [dbo].[ClientSearch]
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

select
    PhoneNo,
    Email,
    FirstName,
    LastName,
    Street,
    StreetNo,
    City
from
    [Client];
END
GO
```

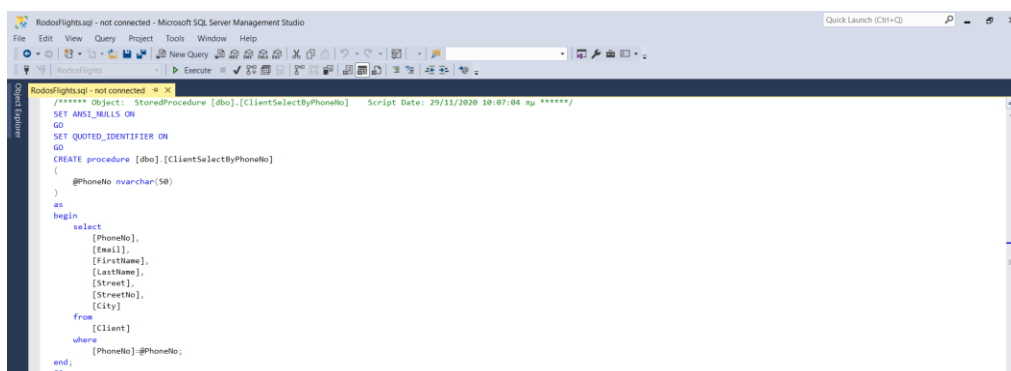
Εικόνα 82. Αποθηκευμένη διαδικασία αναζήτησης πελάτη



```
/****** Object: StoredProcedure [dbo].[ClientSelectAll] Script Date: 29/11/2020 10:07:04 μμ ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
--
-- Author: <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
--
CREATE PROCEDURE [dbo].[ClientSelectAll]
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

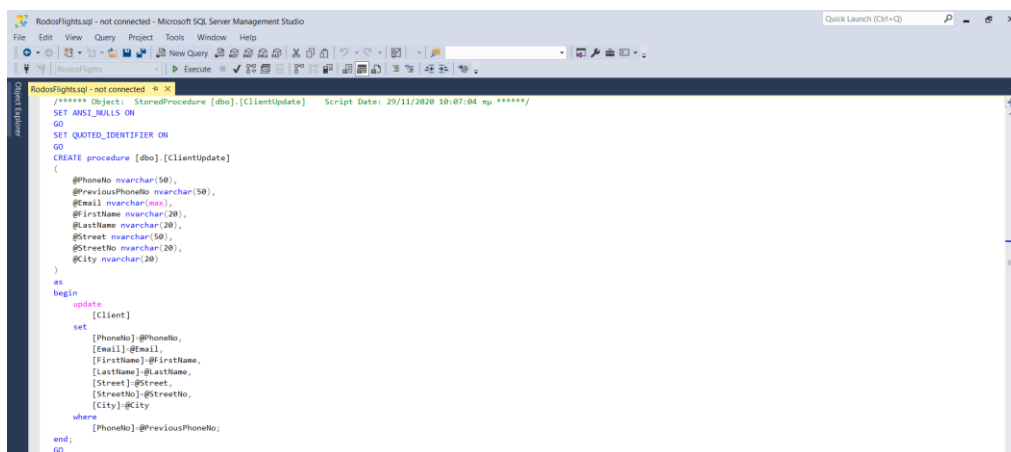
select
[PhoneNo],
[FirstName] + ' ' + [LastName] + ' (' + [PhoneNo] + ')'
from
[Client]
END
GO
```

Εικόνα 83. Αποθηκευμένη διαδικασία επιλογής όλων των πελατών



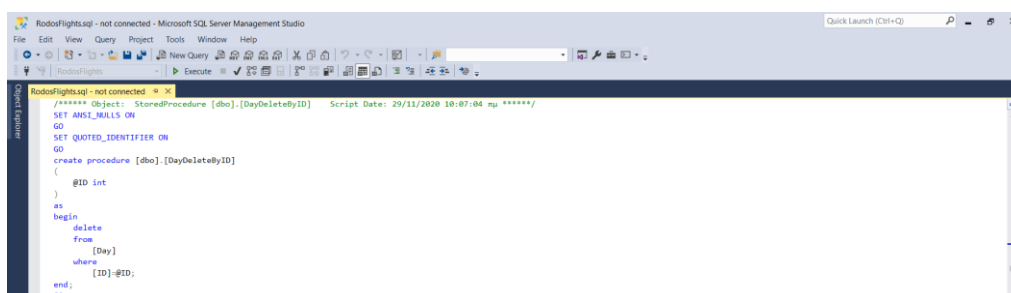
```
/****** Object: StoredProcedure [dbo].[ClientSelectByPhoneNo] Script Date: 29/11/2020 10:07:04 μμ ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[ClientSelectByPhoneNo]
(
@PhoneNo nvarchar(50)
)
as
begin
select
[PhoneNo],
[Email],
[FirstName],
[LastName],
[Street],
[StreetNo],
[City]
from
[Client]
where
[PhoneNo]=@PhoneNo;
end;
GO
```

Εικόνα 84. Αποθηκευμένη διαδικασία επιλογής πελάτη



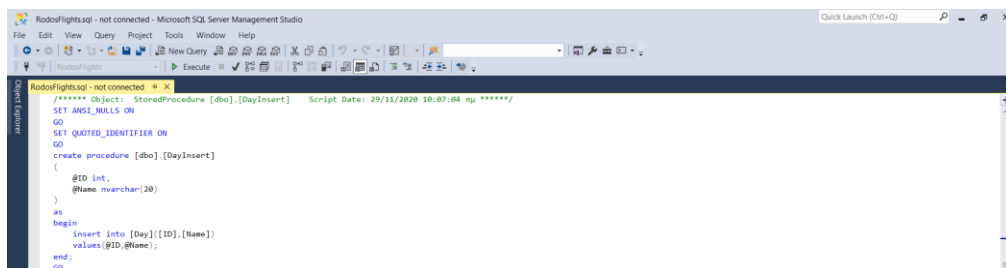
```
/****** Object: StoredProcedure [dbo].[ClientUpdate] Script Date: 29/11/2020 10:07:04 μμ ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[ClientUpdate]
(
@PhoneNo nvarchar(50),
@PreviousPhoneNo nvarchar(50),
@email nvarchar(100),
@FirstName nvarchar(20),
@LastName nvarchar(20),
@Street nvarchar(50),
@StreetNo nvarchar(20),
@City nvarchar(20)
)
as
begin
update
[Client]
set
[PhoneNo]=@PhoneNo,
[Email]=@email,
[FirstName]=@FirstName,
[LastName]=@LastName,
[Street]=@Street,
[StreetNo]=@StreetNo,
[City]=@City
where
[PhoneNo]=@PreviousPhoneNo;
end;
GO
```

Εικόνα 85. Αποθηκευμένη διαδικασία ανανέωσης πελάτη



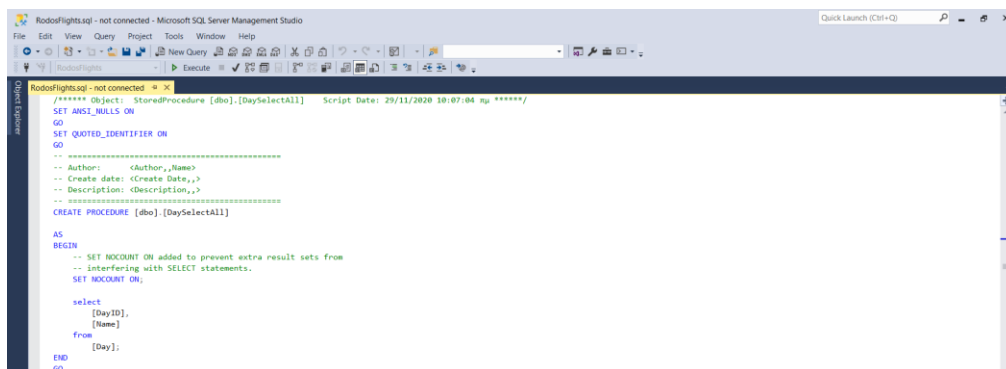
```
/****** Object: StoredProcedure [dbo].[DayDeletebyID] Script Date: 29/11/2020 10:07:04 μμ ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[DayDeletebyID]
(
@ID int
)
as
begin
delete
from
[Day]
where
[ID]=@ID;
end;
GO
```

Εικόνα 86. Αποθηκευμένη διαδικασία διαγραφής μέρας



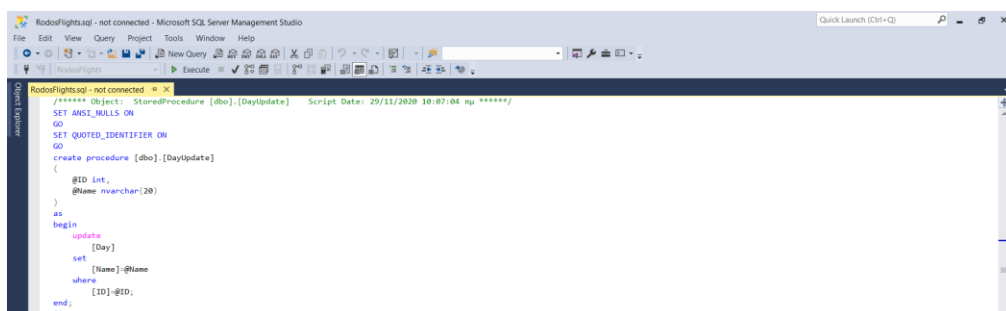
```
***** Object: StoredProcedure [dbo].[DayInsert] Script Date: 29/11/2020 10:07:04 μμ *****  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
create procedure [dbo].[DayInsert]  
(  
    @ID int,  
    @Name nvarchar(20)  
)  
as  
begin  
    insert into [Day]([ID],[Name])  
    values(@ID,@Name);  
end;  
GO
```

Εικόνα 87. Αποθηκευμένη διαδικασία εισαγωγής μέρας



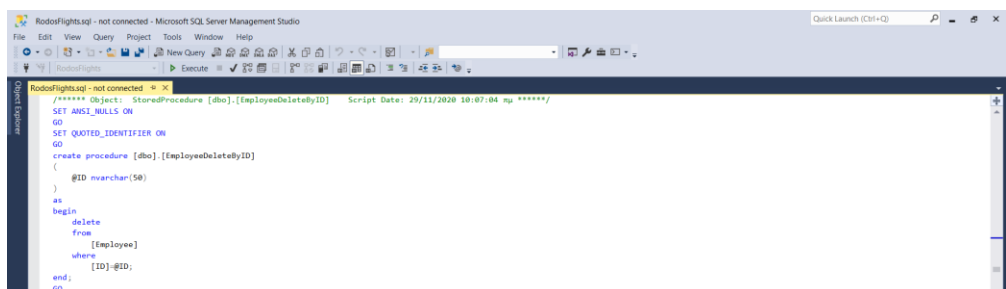
```
***** Object: StoredProcedure [dbo].[DaySelectAll] Script Date: 29/11/2020 10:07:04 μμ *****  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
-- Author: <Author,,Name>  
-- Create date: <Create Date,,>  
-- Description: <Description,,>  
CREATE PROCEDURE [dbo].[DaySelectAll]  
  
AS  
BEGIN  
    -- SET NOCOUNT ON added to prevent extra result sets from  
    -- interfering with SELECT statements.  
    SET NOCOUNT ON;  
  
    select  
        [DayID],  
        [Name]  
    from  
        [Day];  
END  
GO
```

Εικόνα 88. Αποθηκευμένη διαδικασία επιλογής όλων των ημερών



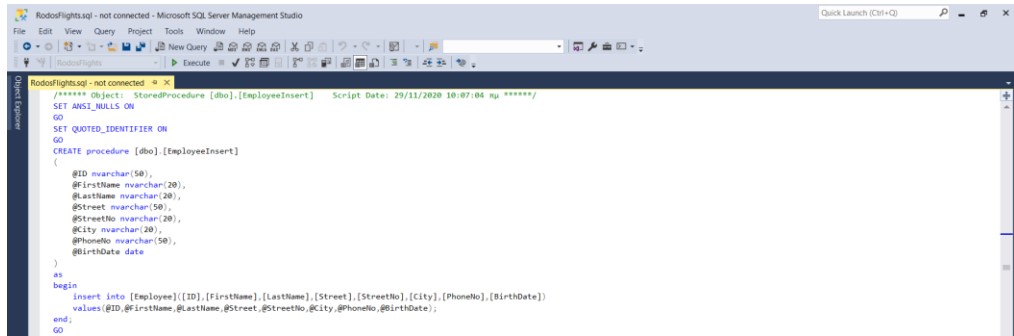
```
***** Object: StoredProcedure [dbo].[DayUpdate] Script Date: 29/11/2020 10:07:04 μμ *****  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
create procedure [dbo].[DayUpdate]  
(  
    @ID int,  
    @Name nvarchar(20)  
)  
as  
begin  
    update  
        [Day]  
    set  
        [Name]=@Name  
    where  
        [ID]=@ID;  
end;  
GO
```

Εικόνα 89. Αποθηκευμένη διαδικασία ανανέωσης μέρας



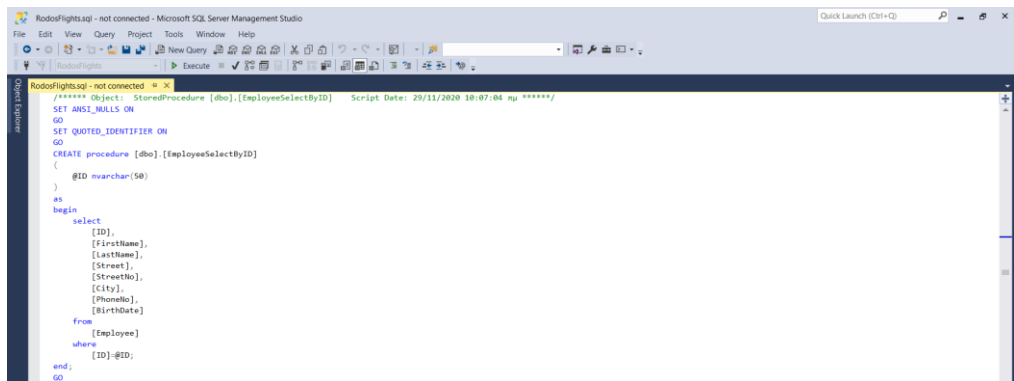
```
***** Object: StoredProcedure [dbo].[EmployeeDeleteByID] Script Date: 29/11/2020 10:07:04 μμ *****  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
create procedure [dbo].[EmployeeDeleteByID]  
(  
    @ID nvarchar(50)  
)  
as  
begin  
    delete  
    from  
        [Employee]  
    where  
        [ID]=@ID;  
end;  
GO
```

Εικόνα 90. Αποθηκευμένη διαδικασία διαγραφής υπαλλήλου



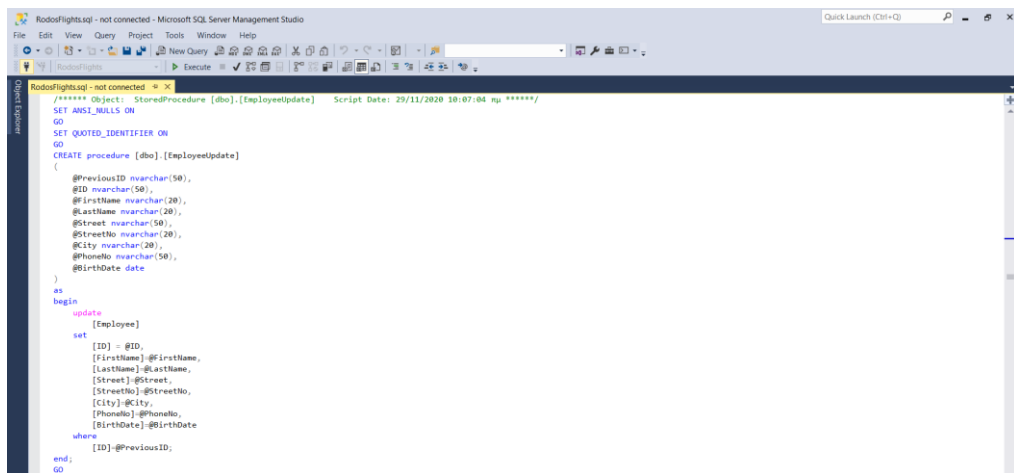
```
Object Explorer
RodosFlights.sq - not connected - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
RodosFlights
Execute
Object Explorer
RodosFlights.sq - not connected
Object Explorer
RodosFlights.sq - not connected - Microsoft SQL Server Management Studio
Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[EmployeeInsert]
(
    @ID nvarchar(50),
    @FirstName nvarchar(20),
    @LastName nvarchar(20),
    @Street nvarchar(50),
    @StreetNo nvarchar(20),
    @City nvarchar(20),
    @PhoneNo nvarchar(50),
    @BirthDate date
)
as
begin
insert into [Employee]([ID],[FirstName],[LastName],[Street],[StreetNo],[City],[PhoneNo],[BirthDate])
values(@ID,@FirstName,@LastName,@Street,@StreetNo,@City,@PhoneNo,@BirthDate);
end;
GO
```

Εικόνα 91. Αποθηκευμένη διαδικασία εισαγωγής υπαλλήλου



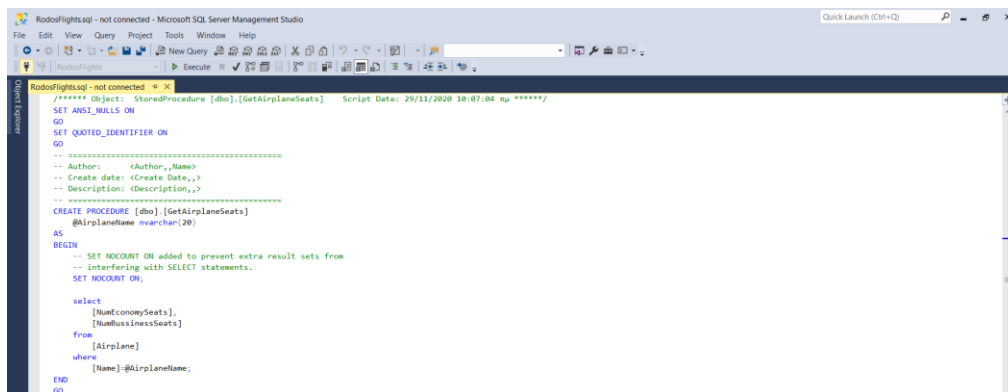
```
Object Explorer
RodosFlights.sq - not connected - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
RodosFlights
Execute
Object Explorer
RodosFlights.sq - not connected
Object Explorer
RodosFlights.sq - not connected - Microsoft SQL Server Management Studio
Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[EmployeeSelectByID]
(
    @ID nvarchar(50)
)
as
begin
select
    [ID],
    [FirstName],
    [LastName],
    [Street],
    [StreetNo],
    [City],
    [PhoneNo],
    [BirthDate]
from
    [Employee]
where
    [ID]=@ID;
end;
GO
```

Εικόνα 92. Αποθηκευμένη διαδικασία επιλογής υπαλλήλου



```
Object Explorer
RodosFlights.sq - not connected - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
RodosFlights
Execute
Object Explorer
RodosFlights.sq - not connected
Object Explorer
RodosFlights.sq - not connected - Microsoft SQL Server Management Studio
Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[EmployeeUpdate]
(
    @PreviousID nvarchar(50),
    @ID nvarchar(50),
    @FirstName nvarchar(20),
    @LastName nvarchar(20),
    @Street nvarchar(50),
    @StreetNo nvarchar(20),
    @City nvarchar(20),
    @PhoneNo nvarchar(50),
    @BirthDate date
)
as
begin
update
    [Employee]
set
    [ID] = @ID,
    [FirstName]=@FirstName,
    [LastName]@LastName,
    [Street]@Street,
    [StreetNo]@StreetNo,
    [City]@City,
    [PhoneNo]@PhoneNo,
    [BirthDate]@BirthDate
where
    [ID]=@PreviousID;
end;
GO
```

Εικόνα 93. Αποθηκευμένη διαδικασία ανανέωσης υπαλλήλου



Εικόνα 94. Αποθηκευμένη διαδικασία επιλογής αριθμού θέσεων αεροπλάνου

```

/***** Object: StoredProcedure [dbo].[GetClientInfo]    Script Date: 29/11/2020 10:07:04 πμ *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
CREATE procedure [dbo].[GetClientInfo]
    @phoneNo nvarchar(50)
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    select
        @phoneNo,
        [Email],
        [FirstName] + ' ' + [LastName] as [FullName],
        [Street] + ' ' + [StreetNo] as [Street],
        [City],
        [TotalKm] = (select
                        sum([Kilometers])
                    from
                        [Ticket]
                    where
                        [PhoneNo] = @phoneNo
                    ),
        u.[Username]
    from
        [Client] c inner join
        [User] u on u.[ClientID] = c.[PhoneNo]
    where
        [PhoneNo] = @phoneNo;

END
GO

```

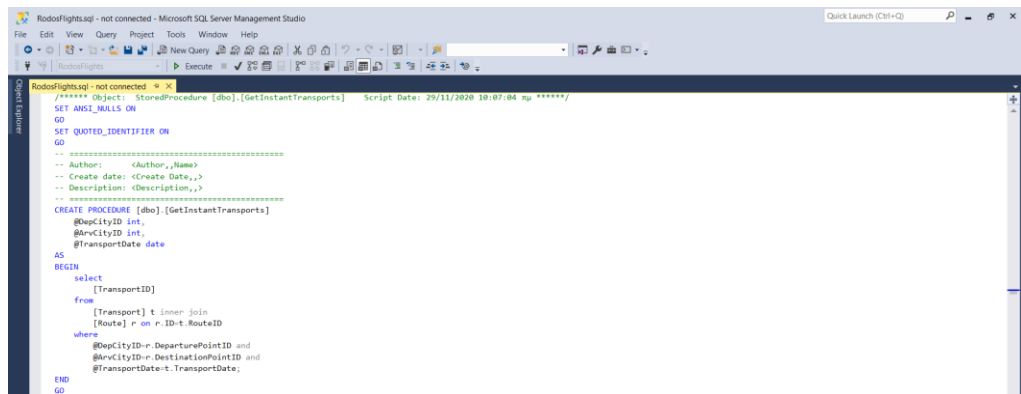
Εικόνα 95. Αποθηκευμένη διαδικασία επιλογής στοιχείων πελάτη

```

/***** Object: StoredProcedure [dbo].[GetIndirectTransports] Script Date: 29/11/2020 10:07:04 πμ *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author: <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
CREATE PROCEDURE [dbo].[GetIndirectTransports]
    @DepCityID int,
    @ArvCityID int,
    @ReservationDate date
AS
BEGIN
    select
        t1.[TransportID] as [TransportID1],
        dc1.[Name] as [DepartureCity1],
        ac1.[Name] as [ArrivalCity1],
        r1.[DepartureTime] as [DepartureTime1],
        r1.[ArrivalTime] as [ArrivalTime1],
        t1.[AirplaneName] as [AirplaneName1],
        r1.[EconomyPrice] as [EconomyPrice1],
        r1.[BusinessPrice] as [BusinessPrice1],
        t1.[EcoSeats] as [EcoSeats1],
        t1.[BusSeats] as [BusSeats1],
        t2.[TransportID] as [TransportID2],
        dc2.[Name] as [DepartureCity2],
        ac2.[Name] as [ArrivalCity2],
        r2.[DepartureTime] as [DepartureTime2],
        r2.[ArrivalTime] as [ArrivalTime2],
        t2.[AirplaneName] as [AirplaneName2],
        r2.[EconomyPrice] as [EconomyPrice2],
        r2.[BusinessPrice] as [BusinessPrice2],
        t2.[EcoSeats] as [EcoSeats2],
        t2.[BusSeats] as [BusSeats2]
    from
        [Transport] t2 inner join
        [Route] r2 on r2.ID=t2.RouteID inner join
        [City] dc2 on dc2.CityID=r2.DeparturePointID inner join
        [City] ac2 on ac2.CityID=r2.DestinationPointID inner join
        [Route] r1 on r1.DestinationPointID=r2.DeparturePointID inner join
        [City] dc1 on dc1.CityID=r1.DeparturePointID inner join
        [City] ac1 on ac1.CityID=r1.DestinationPointID inner join
        [Transport] t1 on t1.RouteID=r1.ID
    where
        r2.DeparturePointID<>@DepCityID and
        r2.DestinationPointID=@ArvCityID and
        t2.TransportDate=@ReservationDate and
        r1.DeparturePointID=@DepCityID
END
GO

```

Εικόνα 96. Αποθηκευμένη διαδικασία επιλογής μεταφορών με ενδιάμεσες στάσεις



Εικόνα 97. Αποθηκευμένη διαδικασία επιλογής άμεσων μεταφορών

```

Microsoft SQL Server Management Studio
RodosFlights.sql - not connected*
Object Explorer
Query Editor
***** Object: StoredProcedure [dbo].[GetLastTransportID] Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- Author: <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
CREATE PROCEDURE [dbo].[GetLastTransportID]
    @RouteID nvarchar(50),
    @TransportDate date,
    @AirplaneName nvarchar(20),
    @PilotID nvarchar(50),
    @COPilotID nvarchar(50),
    @AirHostessID1 nvarchar(50),
    @AirHostessID2 nvarchar(50),
    @AirHostessID3 nvarchar(50),
    @AirHostessID4 nvarchar(50),
    @ManagerID nvarchar(50),
    @EcoSeats int,
    @BusSeats int
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;
    declare @transportID int;

    exec [dbo].[TransportInsert] @transportID output, @RouteID, @TransportDate, @AirplaneName, @PilotID, @COPilotID, @AirHostessID1, @AirHostessID2, @AirHostessID3, @AirHostessID4, @ManagerID, @EcoSeats, @BusSeats;

    select @transportID;
END
GO
    
```

Εικόνα 98. Αποθηκευμένη διαδικασία επιλογής τελευταίας μεταφοράς

```

Microsoft SQL Server Management Studio
RodosFlights.sql - not connected*
Object Explorer
Query Editor
***** Object: StoredProcedure [dbo].[GetReport] Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- Author: <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
CREATE PROCEDURE [dbo].[GetReport]
    @dateFrom date,
    @dateTo date
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    select
        r.[ID] as [RouteID],
        dc.[Name] as [DepartureCity],
        ac.[Name] as [ArrivalCity],
        count(t.[TicketID]) as [TotalTickets]
    from
        [Ticket] t inner join
        [TicketToTransport] tzt on tzt.[TicketID] = t.[TicketID] inner join
        [Transport] tr on tr.[TransportID] = tzt.[TransportID] inner join
        [Route] r on r.[ID] = tr.[RouteID] inner join
        [City] dc on dc.[CityID] = r.[DeparturePointID] inner join
        [City] ac on ac.[CityID] = r.[DestinationPointID]
    where
        t.[ReservationDate] >= @dateFrom and t.[ReservationDate] <= @dateTo
    group by
        r.[ID], dc.[Name], ac.[Name]
END
GO
    
```

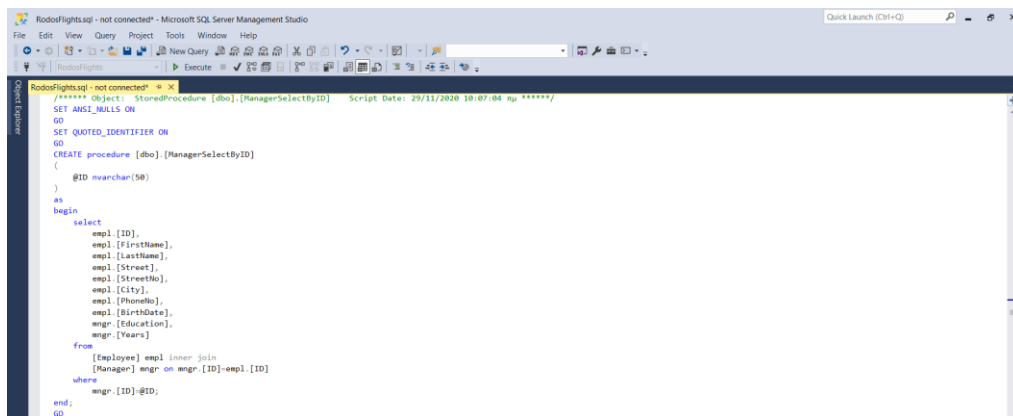
Εικόνα 99. Αποθηκευμένη διαδικασία αναφοράς πληρότητας πτήσεων

```

Microsoft SQL Server Management Studio
RodosFlights.sql - not connected*
Object Explorer
Query Editor
***** Object: StoredProcedure [dbo].[ManagerDeleteByID] Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[ManagerDeleteByID]
(
    @ID nvarchar(50)
)
as
begin
    delete
    from
        [Manager]
    where
        [ID]=@ID;

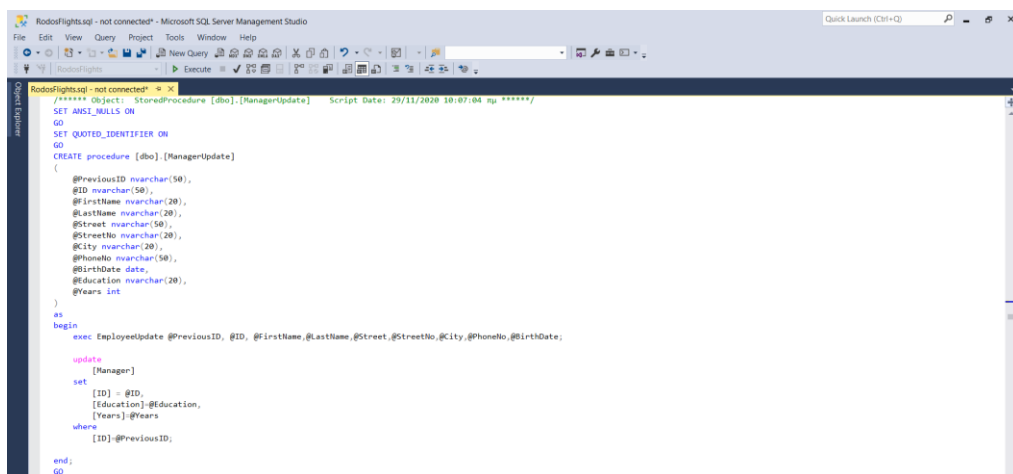
    delete
    from
        [Employee]
    where
        [ID]=@ID;
end;
GO
    
```

Εικόνα 100. Αποθηκευμένη διαδικασία διαγραφής διοικητικού υπαλλήλου



```
Object Explorer
Rodoflights.sql - not connected* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query Execute
Rodoflights
Object Explorer
Rodoflights.sql - not connected*
Object Explorer
***** Object: StoredProcedure [dbo].[ManagerSelectByID] Script Date: 29/11/2020 10:07:04 πμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[ManagerSelectByID]
(
    @ID nvarchar(50)
)
as
begin
select
    empl.[ID],
    empl.[FirstName],
    empl.[LastName],
    empl.[Street],
    empl.[StreetNo],
    empl.[City],
    empl.[Phonelo],
    empl.[BirthDate],
    mngr.[Education],
    mngr.[Years]
from
    [Employee] empl inner join
    [Manager] mngr on mngr.[ID]=empl.[ID]
where
    mngr.[ID]=@ID;
end;
GO
```

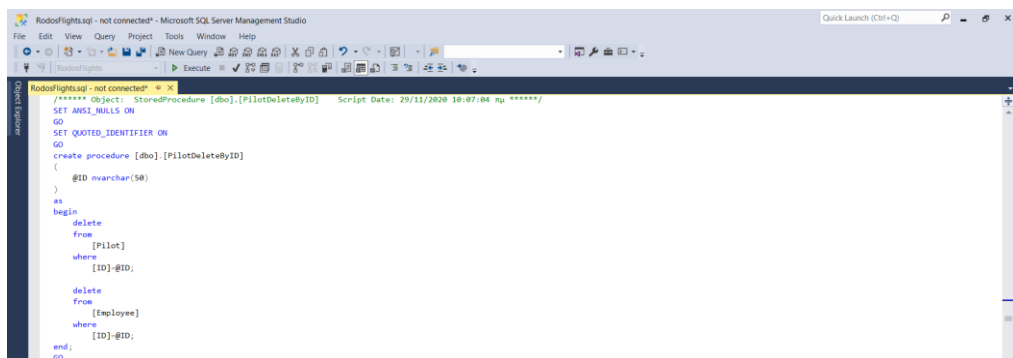
Εικόνα 104. Αποθηκευμένη διαδικασία επιλογής διοικητικού υπαλλήλου



```
Object Explorer
Rodoflights.sql - not connected* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query Execute
Rodoflights
Object Explorer
Rodoflights.sql - not connected*
Object Explorer
***** Object: StoredProcedure [dbo].[ManagerUpdate] Script Date: 29/11/2020 10:07:04 πμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[ManagerUpdate]
(
    @PreviousID nvarchar(50),
    @ID nvarchar(50),
    @FirstName nvarchar(20),
    @LastName nvarchar(20),
    @Street nvarchar(50),
    @StreetNo nvarchar(20),
    @City nvarchar(20),
    @Phonelo nvarchar(50),
    @BirthDate date,
    @Education nvarchar(20),
    @Years int
)
as
begin
exec EmployeeUpdate @PreviousID, @ID, @FirstName,@LastName,@Street,@StreetNo,@City,@Phonelo,@BirthDate;

update
[Manager]
set
    [ID] = @ID,
    [Education]=@Education,
    [Years]=@Years
where
    [ID]=@PreviousID;
end;
GO
```

Εικόνα 105. Αποθηκευμένη διαδικασία ανανέωσης διοικητικού υπαλλήλου



```
Object Explorer
Rodoflights.sql - not connected* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query Execute
Rodoflights
Object Explorer
Rodoflights.sql - not connected*
Object Explorer
***** Object: StoredProcedure [dbo].[PilotDeleteByID] Script Date: 29/11/2020 10:07:04 πμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[PilotDeleteByID]
(
    @ID nvarchar(50)
)
as
begin
delete
from
    [Pilot]
where
    [ID]=@ID;

delete
from
    [Employee]
where
    [ID]=@ID;
end;
GO
```

Εικόνα 106. Αποθηκευμένη διαδικασία διαγραφής πιλότου


```
Object Explorer
RodosFlights.sql - not connected* - Microsoft SQL Server Management Studio
Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[PilotSelectByID]
(
    @ID nvarchar(50)
)
as
begin
select
    empl.[ID],
    empl.[FirstName],
    empl.[LastName],
    empl.[Street],
    empl.[StreetNo],
    empl.[City],
    empl.[PhoneNo],
    empl.[BirthDate],
    p.[FlightHours]
from
    [Employee] empl inner join
    [Pilot] p on p.[ID]=empl.[ID]
where
    p.[ID]=@ID;
end;
GO
```

Εικόνα 110. Αποθηκευμένη διαδικασία επιλογής πιλότου

```
Object Explorer
RodosFlights.sql - not connected* - Microsoft SQL Server Management Studio
Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[PilotUpdate]
(
    @PreviousID nvarchar(50),
    @ID nvarchar(50),
    @FirstName nvarchar(20),
    @LastName nvarchar(20),
    @Street nvarchar(50),
    @StreetNo nvarchar(20),
    @City nvarchar(20),
    @PhoneNo nvarchar(50),
    @BirthDate date,
    @FlightHours int
)
as
begin
exec EmployeeUpdate @PreviousID, @ID, @FirstName,@LastName,@Street,@StreetNo,@City,@PhoneNo,@BirthDate;

update
    [Pilot]
set
    [ID] = @ID,
    [FlightHours] = @FlightHours
where
    [ID] = @PreviousID;
end;
GO
```

Εικόνα 111. Αποθηκευμένη διαδικασία ανανέωσης πιλότου

```

/***** Object: StoredProcedure [dbo].[ReservationDelete]   Script Date: 29/11/2020 10:07:04 πμ *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
CREATE PROCEDURE [dbo].[ReservationDelete]
    @ReservationID int
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    declare @busClass bit;
    declare @currentTransportID1 int;
    declare @currentTransportID2 int;

    select
        @busClass=BussinessClass
    from
        Reservation
    where
        ReservationID=@ReservationID;

    set @currentTransportID1 = (select [TransportID] from [ReservationToTransport] where [ReservationID] = @ReservationID and [TransportIndex] = 1);
    set @currentTransportID2 = (select [TransportID] from [ReservationToTransport] where [ReservationID] = @ReservationID and [TransportIndex] = 2);

    if @busClass = 1 -- ελέγχουμε εάν η τρέχουσα κατηγορία θέσης είναι η bussiness
    begin
        -- αυξάνω κατά ένα τις διαθέσιμες διακεκριμένες θέσεις του transport1
        update
            [Transport]
        set
            [BusSeats] = [BusSeats] + 1
        where
            [TransportID] = @currentTransportID1;

        -- αυξάνω κατά ένα τις διαθέσιμες διακεκριμένες θέσεις του transport2 σε περίπτωση που έχει οριστεί
        if not @currentTransportID2 is null
            update
                [Transport]
            set
                [BusSeats] = [BusSeats] + 1
            where
                [TransportID] = @currentTransportID2;

    end
    else
    begin
        -- αυξάνω κατά ένα τις διαθέσιμες οικονομικές θέσεις του transport1
        update
            [Transport]
        set
            [EcoSeats] = [EcoSeats] + 1
        where
            [TransportID] = @currentTransportID1;

        -- αυξάνω κατά ένα τις διαθέσιμες οικονομικές θέσεις του transport2 σε περίπτωση που έχει οριστεί
        if not @currentTransportID2 is null
            update
                [Transport]
            set
                [EcoSeats] = [EcoSeats] + 1
            where
                [TransportID] = @currentTransportID2;

    end;

    delete
    from
        [Reservation]
    where
        [ReservationID]=@ReservationID;

    delete
    from
        [ReservationToTransport]
    where
        [ReservationID]=@ReservationID;
END
GO

```

Εικόνα 112. Αποθηκευμένη διαδικασία διαγραφής κράτησης

```

/***** Object: StoredProcedure [dbo].[ReservationInsert]    Script Date: 29/11/2020 10:07:04 πμ *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-----
-- Author:            <Author,,Name>
-- Create date:       <Create Date,,>
-- Description:       <Description,,>
-----
CREATE PROCEDURE [dbo].[ReservationInsert]
    @PhoneNo nvarchar(50),
    @ReservationDate date,
    @ExpirationDate date,
    @DepartureCityID int,
    @ArrivalCityID int,
    @TransportID1 int,
    @TransportID2 int,
    @BusinessClass bit,
    @AgentID nvarchar(50)
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    insert into Reservation
    (
        [PhoneNo],
        [ReservationDate],
        [ExpirationDate],
        [DepartureCityID],
        [ArrivalCityID],
        [BusinessClass],
        [AgentID]
    )
    values
    (
        @PhoneNo,
        @ReservationDate,
        @ExpirationDate,
        @DepartureCityID,
        @ArrivalCityID,
        @BusinessClass,
        @AgentID
    );

    declare @reservationID int = scope_identity();

    insert into ReservationToTransport
    (
        [ReservationID],
        [TransportID],
        [TransportIndex]
    )
    values
    (
        @reservationID,
        @TransportID1,
        1
    );

    if @BusinessClass = 1
        update
            [Transport]
        set
            [BusSeats] = [BusSeats] - 1
        where
            [TransportID] = @TransportID1
    else
        update
            [Transport]
        set
            [EcoSeats] = [EcoSeats] - 1
        where
            [TransportID] = @TransportID1;

    if @TransportID2 > 0
    begin
        insert into ReservationToTransport
        (
            [ReservationID],
            [TransportID],
            [TransportIndex]
        )
        values
        (
            @reservationID,
            @TransportID2,
            2
        );

        if @BusinessClass = 1
            update
                [Transport]
            set
                [BusSeats] = [BusSeats] - 1
            where
                [TransportID] = @TransportID2
        else
            update
                [Transport]
            set
                [EcoSeats] = [EcoSeats] - 1
            where
                [TransportID] = @TransportID2;
    end;

    select @reservationID;
END
GO

```

Εικόνα 113. Αποθηκευμένη διαδικασία δημιουργίας κράτησης

```

/***** Object: StoredProcedure [dbo].[ReservationSearch]    Script Date: 29/11/2020 10:07:04 πμ *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-----
-- Author:          <Author,,Name>
-- Create date:    <Create Date,,>
-- Description:    <Description,,>
-----
CREATE PROCEDURE [dbo].[ReservationSearch]
    @AgentID nvarchar(50),
    @PhoneNo nvarchar(50)
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

select
    r.[ReservationID],
    c.[FirstName] + ' ' + c.[LastName] as Client,
    [ReservationDate],
    [ExpirationDate],
    case [BussinessClass]
        when 1 then 'Διακεκριμένη'
        else 'Οικονομική'
    end as SeatCategory,
    isNull(a.[FirstName] + ' ' + a.[LastName], '') as Agent,
    dc.[Name] as [DepartureCity],
    ac.[Name] as [ArrivalCity],
    case [BussinessClass]
        when 1 then r1.BussinessPrice+isNull(r2.BussinessPrice,0)
        else r1.EconomyPrice+isNull(r2.EconomyPrice,0)
    end as Price
from
    [Reservation] r inner join
    [City] dc on dc.CityID=r.DepartureCityID inner join
    [City] ac on ac.CityID=r.ArrivalCityID inner join
    [Client] c on c.PhoneNo=r.PhoneNo left join
    [Agent] a on a.ID=r.AgentID left join
    [ReservationToTransport] rt1 on (rt1.ReservationID=r.ReservationID and rt1.TransportIndex=1) left join
    [Transport] t1 on t1.TransportID=rt1.TransportID left join
    [Route] r1 on r1.ID=t1.RouteID left join
    [ReservationToTransport] rt2 on (rt2.ReservationID=r.ReservationID and rt2.TransportIndex=2) left join
    [Transport] t2 on t2.TransportID=rt2.TransportID left join
    [Route] r2 on r2.ID=t2.RouteID
where
    (r.AgentID=@AgentID and @AgentID<>'0')
    or
    (r.PhoneNo=@PhoneNo and @PhoneNo<>'0')
END
GO

```

Εικόνα 114. Αποθηκευμένη διαδικασία αναζήτησης κράτησης

The screenshot shows the SQL Server Enterprise Edition interface. The 'Query Editor' window displays the following T-SQL code for the stored procedure 'ReservationSelectByID':

```

/***** Object: StoredProcedure [dbo].[ReservationSelectByID]    Script Date: 29/11/2020 10:07:04 πμ *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-----
-- Author:          <Author,,Name>
-- Create date:    <Create Date,,>
-- Description:    <Description,,>
-----
CREATE PROCEDURE [dbo].[ReservationSelectByID]
    @ReservationID int
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

select
    r.[ReservationID],
    r.[PhoneNo],
    r.[ReservationDate],
    r.[ExpirationDate],
    r.[BussinessClass],
    r.[AgentID],
    r.[DepartureCityID],
    r.[ArrivalCityID],
    rt1.[TransportID] as [TransportID1],
    isNull(rt2.[TransportID], 0) as [TransportID2]
from
    [Reservation] r inner join
    [ReservationToTransport] rt1 on (rt1.[ReservationID] = r.[ReservationID] and rt1.[TransportIndex] = 1) left join
    [ReservationToTransport] rt2 on (rt2.[ReservationID] = r.[ReservationID] and rt2.[TransportIndex] = 2)
where
    r.[ReservationID] = @ReservationID
END
GO

```

Εικόνα 115. Αποθηκευμένη διαδικασία επιλογής κράτησης

```

/***** Object: StoredProcedure [dbo].[ReservationUpdate]  Script Date: 29/11/2020 10:07:04 μμ *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      <Author, Name>
-- Create date: <Create Date,>
-- Description: <Description,>
-- =====
CREATE PROCEDURE [dbo].[ReservationUpdate]
    @ReservationID int,
    @PhoneNo nvarchar(50),
    @ReservationDate date,
    @ExpirationDate date,
    @DepartureCityID int,
    @ArrivalCityID int,
    @TransportID1 int,
    @TransportID2 int,
    @BusinessClass bit,
    @AgentID nvarchar(50)
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    declare @currentBusClass bit = (select [BusinessClass] from Reservation where [ReservationID] = @ReservationID);
    declare @currentTransportID1 int = (select [TransportID] from [ReservationToTransport] where [ReservationID] = @ReservationID and [TransportIndex] = 1);
    declare @currentTransportID2 int = (select [TransportID] from [ReservationToTransport] where [ReservationID] = @ReservationID and [TransportIndex] = 2);

    if @currentBusClass = 1 -- ελέγχουμε εάν η τρέχουσα κατηγορία θέσης είναι η bussiness
    begin
        -- αυθνα κατά ένα τις διαθέσιμες διακεκριμένες θέσεις του transport1
        update
            [Transport]
        set
            [BusSeats] = [BusSeats] + 1
        where
            [TransportID] = @currentTransportID1;

        -- αυθνα κατά ένα τις διαθέσιμες διακεκριμένες θέσεις του transport2 σε περίπτωση που έχει οριστεί
        if not @currentTransportID2 is null
            update
                [Transport]
            set
                [BusSeats] = [BusSeats] + 1
            where
                [TransportID] = @currentTransportID2;
    end
    else
    begin
        -- αυθνα κατά ένα τις διαθέσιμες οικονομικές θέσεις του transport1
        update
            [Transport]
        set
            [EcoSeats] = [EcoSeats] + 1
        where
            [TransportID] = @currentTransportID1;

        -- αυθνα κατά ένα τις διαθέσιμες οικονομικές θέσεις του transport2 σε περίπτωση που έχει οριστεί
        if not @currentTransportID2 is null
            update
                [Transport]
            set
                [EcoSeats] = [EcoSeats] + 1
            where
                [TransportID] = @currentTransportID2;
    end;

    update
        [Reservation]
    set
        [PhoneNo] = @PhoneNo,
        [ReservationDate] = @ReservationDate,
        [ExpirationDate] = @ExpirationDate,
        [DepartureCityID] = @DepartureCityID,
        [ArrivalCityID] = @ArrivalCityID,
        [BusinessClass] = @BusinessClass,
        [AgentID] = @AgentID
    where
        [ReservationID] = @ReservationID;

    delete
    from
        [ReservationToTransport]
    where
        [ReservationID] = @ReservationID;

    insert into ReservationToTransport
    (
        [ReservationID],
        [TransportID],
        [TransportIndex]
    )
    values
    (
        @ReservationID,
        @TransportID1,
        1
    );

    if @BusinessClass = 1
        update
            [Transport]
        set
            [BusSeats] = [BusSeats] - 1
        where
            [TransportID] = @TransportID1
    else
        update
            [Transport]
        set
            [EcoSeats] = [EcoSeats] - 1
        where
            [TransportID] = @TransportID1;

    if @TransportID2 > 0
    begin
        insert into ReservationToTransport
        (
            [ReservationID],
            [TransportID],
            [TransportIndex]
        )
        values
        (
            @ReservationID,
            @TransportID2,
            2
        );

        if @BusinessClass = 1
            update
                [Transport]
            set
                [BusSeats] = [BusSeats] - 1
            where
                [TransportID] = @TransportID2
        else
            update
                [Transport]
            set
                [EcoSeats] = [EcoSeats] - 1
            where
                [TransportID] = @TransportID2;
    end;

END
GO

```

Εικόνα 116. Αποθηκευμένη διαδικασία ανανέωσης κράτησης

```
Object Explorer
Rodoflights.sql - not connected* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
Rodoflights
Execute
Rodoflights.sql - not connected*
Object Explorer
***** Object: StoredProcedure [dbo].[ReservationValidate] Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
--
-- Author: <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
--
CREATE PROCEDURE [dbo].[ReservationValidate]
    @ReservationID int,
    @resDate date,
    @phoneNo nvarchar(50)
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    select
        count([ReservationID])
    from
        [Reservation]
    where
        ([phoneNo] = @phoneNo and
        [ReservationDate] = @resDate and
        [ReservationID] <> @ReservationID);
END
GO
```

Εικόνα 117. Αποθηκευμένη διαδικασία ελέγχου εγκυρότητας κράτησης

```
Object Explorer
Rodoflights.sql - not connected* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
Rodoflights
Execute
Rodoflights.sql - not connected*
Object Explorer
***** Object: StoredProcedure [dbo].[RouteDeleteByID] Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[RouteDeleteByID]
(
    @ID nvarchar(50)
)
as
begin
delete
from
[Route]
where
[ID]=@ID;
end;
GO
```

Εικόνα 118. Αποθηκευμένη διαδικασία διαγραφής δρομολογίου

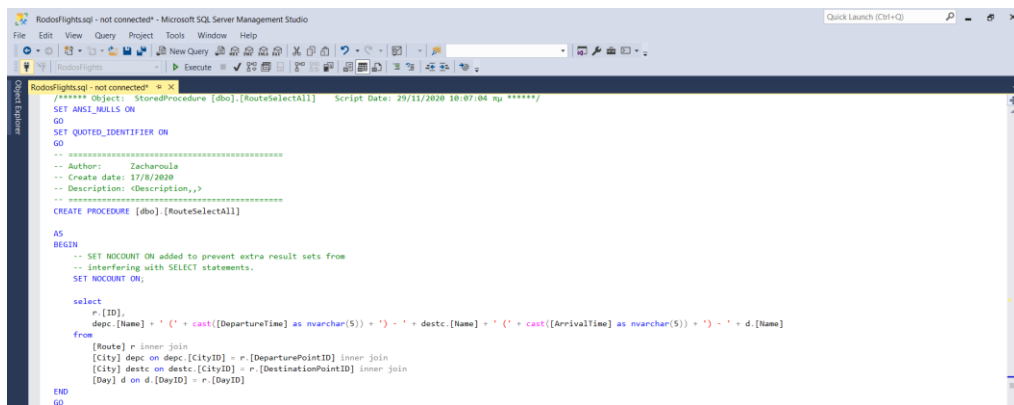
```
Object Explorer
Rodoflights.sql - not connected* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
Rodoflights
Execute
Rodoflights.sql - not connected*
Object Explorer
***** Object: StoredProcedure [dbo].[RouteInsert] Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[RouteInsert]
(
    @ID nvarchar(50),
    @DeparturePointID int,
    @DestinationPointID int,
    @DayID int,
    @Distance float,
    @DepartureTime time(7),
    @ArrivalTime time(7),
    @BusinessPrice float,
    @EconomyPrice float
)
as
begin
insert into [Route]([ID],[DeparturePointID],[DestinationPointID],[DayID],[Distance],[DepartureTime],[ArrivalTime],[BusinessPrice],[EconomyPrice])
values(@ID,@DeparturePointID,@DestinationPointID,@DayID,@Distance,@DepartureTime,@ArrivalTime,@BusinessPrice,@EconomyPrice);
end;
GO
```

Εικόνα 119. Αποθηκευμένη διαδικασία εισαγωγής δρομολογίου

```
Object Explorer
Rodoflights.sql - not connected* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
Rodoflights
Execute
Rodoflights.sql - not connected*
Object Explorer
***** Object: StoredProcedure [dbo].[RouteSearch] Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
--
-- Author: Zacharoula
-- Create date: 9/8/2020
-- Description: <Description,,>
--
CREATE PROCEDURE [dbo].[RouteSearch]
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    select
        r.[ID],
        dep.[Name],
        dest.[Name],
        d.[Name],
        r.[Distance],
        r.[DepartureTime],
        r.[ArrivalTime],
        r.[BusinessPrice],
        r.[EconomyPrice]
    from
        [Route] r inner join
        [Day] d on d.[DayID] = r.[DayID] inner join
        [City] dep on dep.[CityID] = r.[DeparturePointID] inner join
        [City] dest on dest.[CityID] = r.[DestinationPointID]
END
GO
```

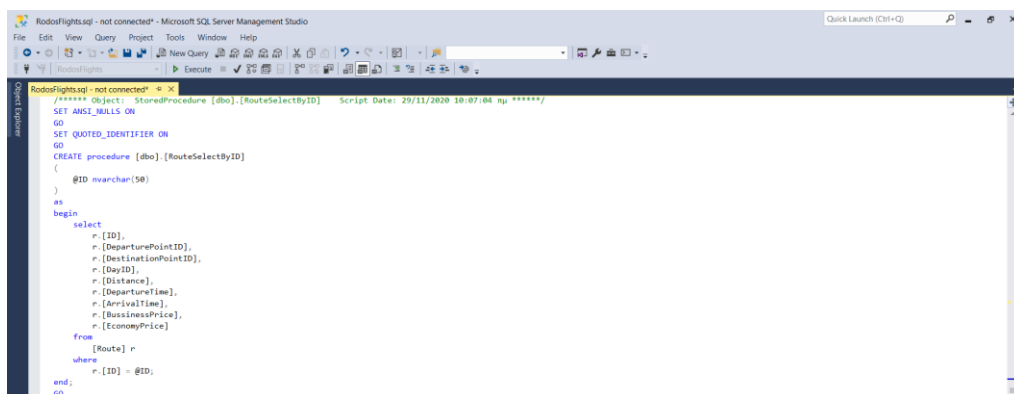
Εικόνα 120. Αποθηκευμένη διαδικασία αναζήτησης δρομολογίου



```
Object Explorer
RodosFlights.sql - not connected* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
Execute
RodosFlights
Object Explorer
RodosFlights.sql - not connected*
/***** Object: StoredProcedure [dbo].[RouteSelectAll] Script Date: 29/11/2020 10:07:04 μμ *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- Author: Zacharoula
-- Create date: 17/8/2020
-- Description: (Description, )
-- =====
CREATE PROCEDURE [dbo].[RouteSelectAll]
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

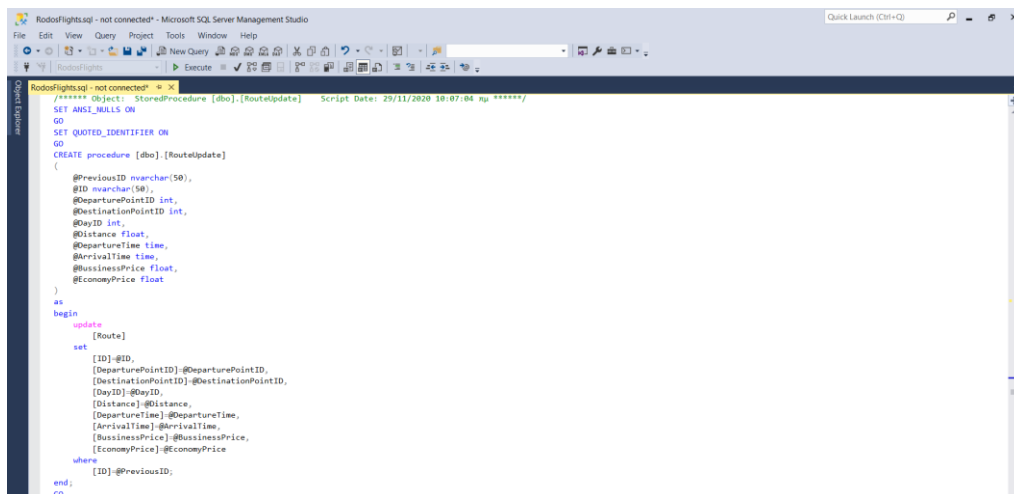
select
r.[ID],
depc.[Name] + ' (' + cast([DepartureTime] as nvarchar(5)) + ') - ' + destc.[Name] + ' (' + cast([ArrivalTime] as nvarchar(5)) + ') - ' + d.[Name]
from
[Route] r inner join
[City] depc on depc.[CityID] = r.[DeparturePointID] inner join
[City] destc on destc.[CityID] = r.[DestinationPointID] inner join
[Day] d on d.[DayID] = r.[DayID]
END
GO
```

Εικόνα 121. Αποθηκευμένη διαδικασία επιλογής όλων των δρομολογίων



```
Object Explorer
RodosFlights.sql - not connected* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
Execute
RodosFlights
Object Explorer
RodosFlights.sql - not connected*
/***** Object: StoredProcedure [dbo].[RouteSelectByID] Script Date: 29/11/2020 10:07:04 μμ *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[RouteSelectByID]
(
@ID nvarchar(50)
)
as
begin
select
r.[ID],
r.[DeparturePointID],
r.[DestinationPointID],
r.[DayID],
r.[Distance],
r.[DepartureTime],
r.[ArrivalTime],
r.[BusinessPrice],
r.[EconomyPrice]
from
[Route] r
where
r.[ID] = @ID;
end;
GO
```

Εικόνα 122. Αποθηκευμένη διαδικασία επιλογής δρομολογίου



```
Object Explorer
RodosFlights.sql - not connected* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
Execute
RodosFlights
Object Explorer
RodosFlights.sql - not connected*
/***** Object: StoredProcedure [dbo].[RouteUpdate] Script Date: 29/11/2020 10:07:04 μμ *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[RouteUpdate]
(
@previousID nvarchar(50),
@ID nvarchar(50),
@DeparturePointID int,
@DestinationPointID int,
@DayID int,
@Distance float,
@DepartureTime time,
@ArrivalTime time,
@BusinessPrice float,
@EconomyPrice float
)
as
begin
update
[Route]
set
[ID]=@ID,
[DeparturePointID]=@DeparturePointID,
[DestinationPointID]=@DestinationPointID,
[DayID]=@DayID,
[Distance]=@Distance,
[DepartureTime]=@DepartureTime,
[ArrivalTime]=@ArrivalTime,
[BusinessPrice]=@BusinessPrice,
[EconomyPrice]=@EconomyPrice
where
[ID]=@previousID;
end;
GO
```

Εικόνα 123. Αποθηκευμένη διαδικασία ανανέωσης δρομολογίου

```

/***** Object: StoredProcedure [dbo].[SignIn]    Script Date: 29/11/2020 10:07:04 πμ *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Author,,Name>
-- Create date:    <Create Date,,>
-- Description:    <Description,,>
-- =====
CREATE PROCEDURE [dbo].[SignIn]
    @Username nvarchar(50)
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    select
        u.[Password],
        [FirstName] = case
                        when not u.[ClientID] is null then c.[FirstName]
                        when not u.[AgentID] is null then a.[FirstName]
                        else e.[FirstName]
                    end,
        [LastName] = case
                        when not u.[ClientID] is null then c.[LastName]
                        when not u.[AgentID] is null then a.[LastName]
                        else e.[LastName]
                    end,
        u.[ClientID],
        u.[AgentID],
        u.[AdministratorID]
    from
        [User] u left join
        [Client] c on c.[PhoneNo] = u.ClientID left join
        [Agent] a on a.[ID] = u.[AgentID] left join
        [Employee] e on e.[ID] = u.AdministratorID
    where
        u.[Username] = @Username;
END
GO

```

Εικόνα 124. Αποθηκευμένη διαδικασία εισόδου χρήστη

```

/***** Object: StoredProcedure [dbo].[TicketInsert]    Script Date: 29/11/2020 10:07:04 πμ *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
CREATE PROCEDURE [dbo].[TicketInsert]
    @ReservationID int
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    declare @phoneNo nvarchar(50);
    declare @reservationDate date;
    declare @BussinessClass bit;
    declare @DepartureCityID int;
    declare @ArrivalCityID int;
    declare @Price float;
    declare @Kilometers float;
    declare @email nvarchar(255);
    declare @clientName nvarchar(max);
    declare @freeOfChargeKm float;
    declare @totalKm float;

    select
        @phoneNo = r.[PhoneNo],
        @reservationDate = r.[ReservationDate],
        @BussinessClass = r.[BussinessClass],
        @DepartureCityID = r.[DepartureCityID],
        @ArrivalCityID = r.[ArrivalCityID]
    from
        [Reservation] r
    where
        r.[ReservationID] = @ReservationID;

    select
        @email = [Email],
        @clientName = [FirstName] + ' ' + [LastName],
        @freeOfChargeKm = [FreeOfChargeKm],
        @totalKm = [TotalKm]
    from
        [Client]
    where
        [PhoneNo] = @phoneNo;

    select
        @Kilometers = sum(r.[Distance])
    from
        [Route] r
    where
        r.[ID] in (select
            [RouteID]
            from
                [Transport]
            where
                [TransportID] in (select
                    [TransportID]
                    from
                        [ReservationToTransport]
                    where
                        [ReservationID] = @ReservationID
                )
            );

```

Εικόνα 125. Αποθηκευμένη διαδικασία δημιουργίας εισιτηρίου (1)

```

if @BusinessClass = 1
    select
        @price = sum([BusinessPrice])
    from
        [Route] r
    where
        r.[ID] in (select
                    [RouteID]
                from
                    [Transport]
                where
                    [TransportID] in (select
                                    [TransportID]
                                from
                                    [ReservationToTransport]
                                where
                                    [ReservationID] = @ReservationID
                                )
                    )
else
    select
        @price = sum([EconomyPrice])
    from
        [Route] r
    where
        r.[ID] in (select
                    [RouteID]
                from
                    [Transport]
                where
                    [TransportID] in (select
                                    [TransportID]
                                from
                                    [ReservationToTransport]
                                where
                                    [ReservationID] = @ReservationID
                                )
                    )
);

if @Kilometers < 300
    if @FreeOfChargeKm >= @Kilometers
        begin
            set @Price = 0;
            set @FreeOfChargeKm = @FreeOfChargeKm - @Kilometers;

            update
                [Client]
            set
                [FreeOfChargeKm] = @FreeOfChargeKm
            where
                [PhoneNo] = @PhoneNo;

        end
    else
        begin
            set @totalKm = @totalKm + @Kilometers;

            set @FreeOfChargeKm = cast(@totalKm / 10000 as int) * 300;
            set @totalKm = @totalKm - cast(@totalKm / 10000 as int) * 10000;

            update
                [Client]
            set
                [TotalKm] = @totalKm,
                [FreeOfChargeKm] = @FreeOfChargeKm
            where
                [PhoneNo] = @PhoneNo;

        end;

insert into [Ticket]
(
    [PhoneNo]
    , [ReservationDate]
    , [BusinessClass]
    , [DepartureCityID]
    , [ArrivalCityID]
    , [Price]
    , [Kilometers]
)
values
(
    @PhoneNo,
    @ReservationDate,
    @BusinessClass,
    @DepartureCityID,
    @ArrivalCityID,
    @Price,
    @Kilometers
);

declare @ticketID int = scope_identity();

insert into [TicketToTransport]
(
    [TicketID],
    [TransportID],
    [TransportIndex]
)
select
    @ticketID,
    [TransportID],
    [TransportIndex]
from
    [ReservationToTransport]
where
    [ReservationID] = @ReservationID;

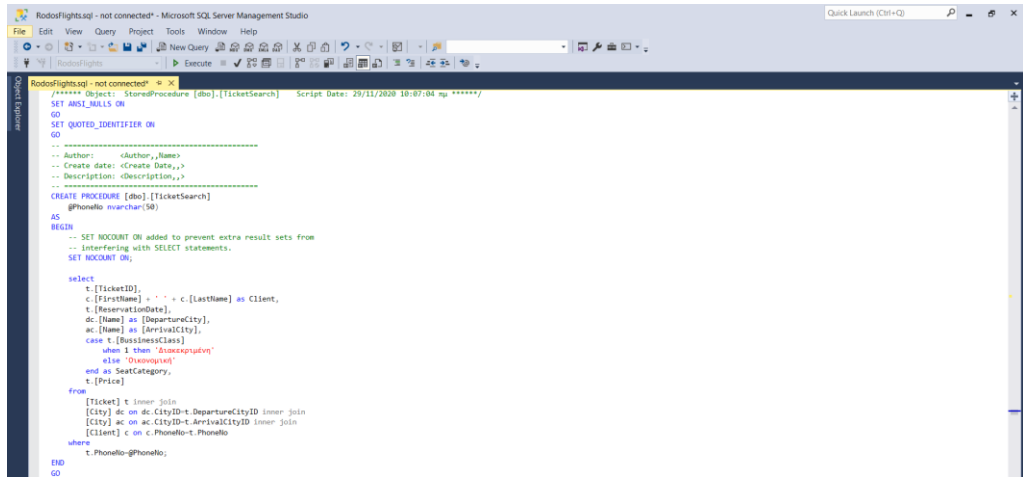
delete
from
    [Reservation]
where
    [ReservationID] = @ReservationID;

select @TicketID, @email, @clientName;

END
GO

```

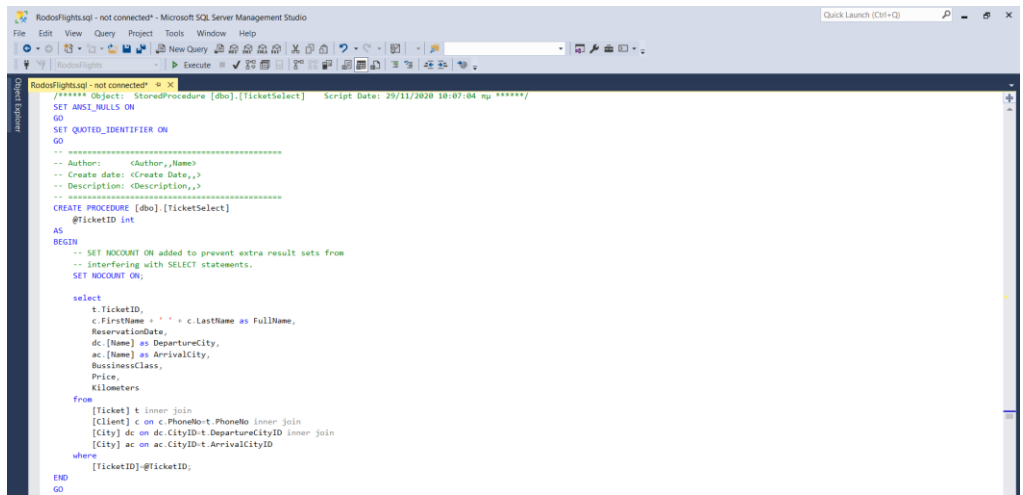
Εικόνα 126. Αποθηκευμένη διαδικασία δημιουργίας εισιτηρίου (2)



```
Microsoft SQL Server Management Studio
Object Explorer
RodosFlights.sq - not connected* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query Execute
RodosFlights
Object Explorer
RodosFlights.sq - not connected*
/***** Object: StoredProcedure [dbo].[TicketSearch] Script Date: 29/11/2020 10:07:04 μμ *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- Author: <Author, Name>
-- Create date: <Create Date, >
-- Description: <Description, >
CREATE PROCEDURE [dbo].[TicketSearch]
    @PhoneNo nvarchar(50)
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    select
        t.[TicketID],
        c.[FirstName] + ' ' + c.[LastName] as Client,
        t.[ReservationDate],
        dc.[Name] as [DepartureCity],
        ac.[Name] as [ArrivalCity],
        case t.[BusinessClass]
            when 1 then 'Διακεκριμένη'
            else 'Οικονομική'
        end as SeatCategory,
        t.[Price]
    from
        [Ticket] t inner join
        [Client] c on c.PhoneNo=t.PhoneNo inner join
        [City] dc on dc.CityID=t.DepartureCityID inner join
        [City] ac on ac.CityID=t.ArrivalCityID inner join
    where
        t.PhoneNo=@PhoneNo;
END
GO
```

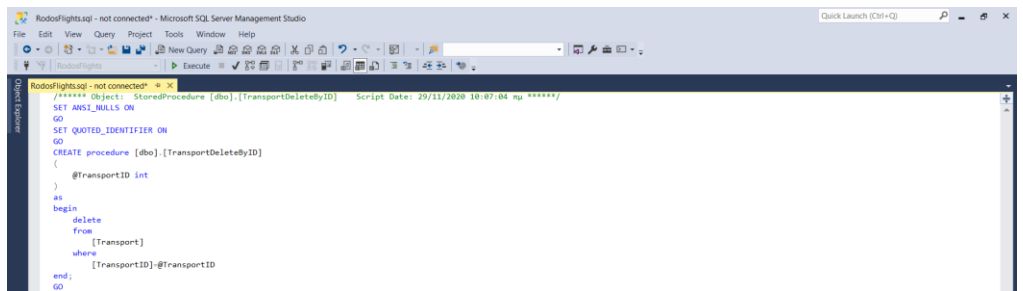
Εικόνα 127. Αποθηκευμένη διαδικασία αναζήτησης εισιτηρίου



```
Microsoft SQL Server Management Studio
Object Explorer
RodosFlights.sq - not connected* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query Execute
RodosFlights
Object Explorer
RodosFlights.sq - not connected*
/***** Object: StoredProcedure [dbo].[TicketSelect] Script Date: 29/11/2020 10:07:04 μμ *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- Author: <Author, Name>
-- Create date: <Create Date, >
-- Description: <Description, >
CREATE PROCEDURE [dbo].[TicketSelect]
    @TicketID int
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    select
        t.[TicketID],
        c.[FirstName] + ' ' + c.[LastName] as FullName,
        t.[ReservationDate],
        dc.[Name] as DepartureCity,
        ac.[Name] as ArrivalCity,
        t.[Price],
        t.[BusinessClass],
        Kilometers
    from
        [Ticket] t inner join
        [Client] c on c.PhoneNo=t.PhoneNo inner join
        [City] dc on dc.CityID=t.DepartureCityID inner join
        [City] ac on ac.CityID=t.ArrivalCityID
    where
        [TicketID]=@TicketID;
END
GO
```

Εικόνα 128. Αποθηκευμένη διαδικασία επιλογής εισιτηρίου



```
Microsoft SQL Server Management Studio
Object Explorer
RodosFlights.sq - not connected* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query Execute
RodosFlights
Object Explorer
RodosFlights.sq - not connected*
/***** Object: StoredProcedure [dbo].[TransportDeletebyID] Script Date: 29/11/2020 10:07:04 μμ *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[TransportDeletebyID]
(
    @TransportID int
)
as
begin
delete
from
[Transport]
where
[TransportID]=@TransportID
end,
GO
```

Εικόνα 129. Αποθηκευμένη διαδικασία διαγραφής εισιτηρίου

```

/***** Object: StoredProcedure [dbo].[TransportFindRoute]    Script Date: 29/11/2020 10:07:04 πμ
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Author,,Name>
-- Create date:    <Create Date,,>
-- Description:    <Description,,>
-- =====
CREATE PROCEDURE [dbo].[TransportFindRoute]
    @DepCityID int,
    @ArvCityID int,
    @ReservationDate date
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

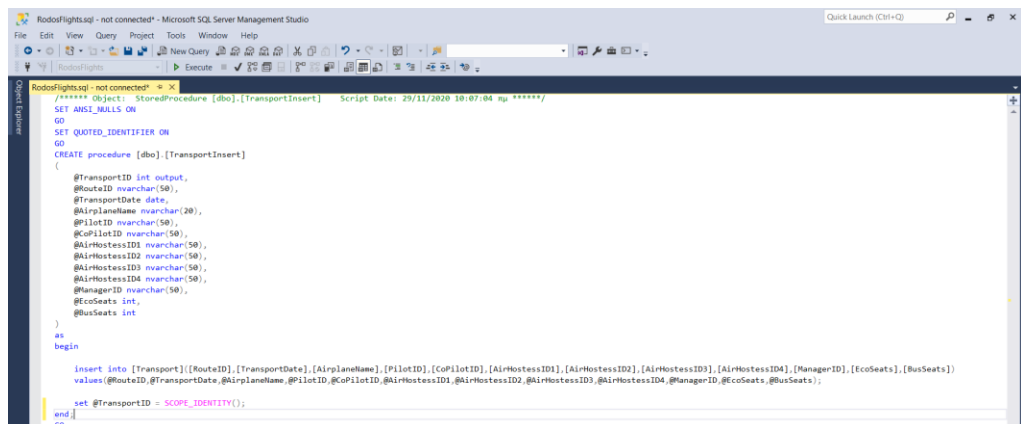
    select
        t.[TransportID],
        dc.Name as DepartureCity,
        ac.Name as ArrivalCity,
        r.DepartureTime,
        r.ArrivalTime,
        t.AirplaneName,
        r.EconomyPrice,
        r.BussinessPrice,
        t.EcoSeats,
        t.BusSeats

    from
        [Transport] t inner join
        [Route] r on r.ID=t.RouteID inner join
        [City] dc on dc.CityID=r.DeparturePointID inner join
        [City] ac on ac.CityID=r.DestinationPointID

    where
        r.DeparturePointID=@DepCityID and
        r.DestinationPointID=@ArvCityID and
        t.TransportDate=@ReservationDate;
END
GO

```

Εικόνα 130. Αποθηκευμένη διαδικασία αναζήτησης διαθέσιμων μεταφορών δρομολογίου



Εικόνα 131. Αποθηκευμένη διαδικασία εισαγωγής μεταφοράς

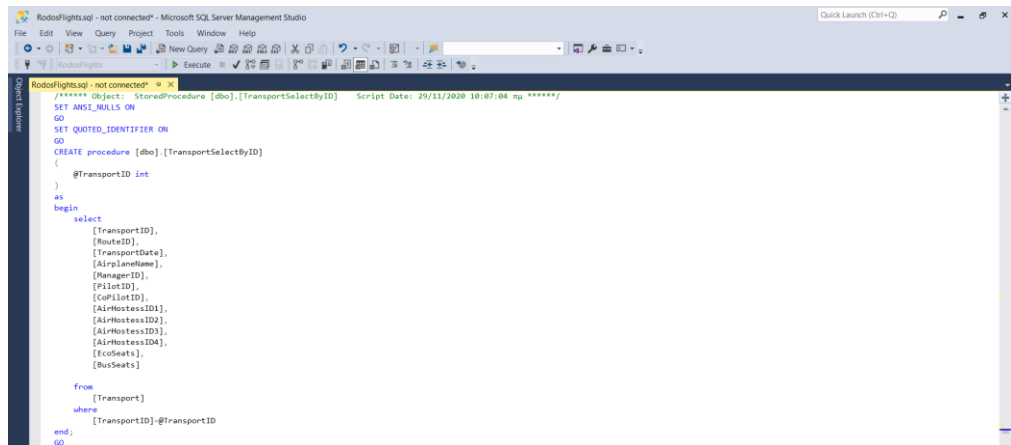
```

/***** Object: StoredProcedure [dbo].[TransportSearch]   Script Date: 29/11/2020 10:07:04 μμ *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
--
-- Author: <Author,Name>
-- Create date: <Create Date,>
-- Description: <Description,>
--
CREATE PROCEDURE [dbo].[TransportSearch]
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

select
t.[TransportID],
depc.[Name] = '(' + cast([DepartureTime] as nvarchar(5)) + ') - ' + destd.[Name] + '(' + cast([ArrivalTime] as nvarchar(5)) + ') - ' + d.[Name] as [Route],
t.[TransportDate],
t.[AirplaneName],
p1.[FirstName] + ' ' + p1.[LastName] + '(' + p1.ID + ')' as [Pilot],
p2.[FirstName] + ' ' + p2.[LastName] + '(' + p2.ID + ')' as [CoPilot],
m.[FirstName] + ' ' + m.[LastName] + '(' + m.ID + ')' as [Manager],
ah1.[FirstName] + ' ' + ah1.[LastName] + '(' + ah1.ID + ')' as [AirHostess1],
ah2.[FirstName] + ' ' + ah2.[LastName] + '(' + ah2.ID + ')' as [AirHostess2],
ah3.[FirstName] + ' ' + ah3.[LastName] + '(' + ah3.ID + ')' as [AirHostess3],
ah4.[FirstName] + ' ' + ah4.[LastName] + '(' + ah4.ID + ')' as [AirHostess4]
from
[Transport] t inner join
[Employee] p1 on p1.[ID] = t.[PilotID] inner join
[Employee] p2 on p2.[ID] = t.[CoPilotID] inner join
[Employee] m on m.[ID] = t.[ManagerID] inner join
[Employee] ah1 on ah1.[ID] = t.[AirHostessID1] inner join
[Employee] ah2 on ah2.[ID] = t.[AirHostessID2] inner join
[Employee] ah3 on ah3.[ID] = t.[AirHostessID3] inner join
[Employee] ah4 on ah4.[ID] = t.[AirHostessID4] inner join
[Route] r on r.[ID] = t.[RouteID] inner join
[City] depc on depc.[CityID] = r.[DeparturePointID] inner join
[City] destd on destd.[CityID] = r.[DestinationPointID] inner join
[Day] d on d.[DayID] = r.[DayID]
END
GO

```

Εικόνα 132. Αποθηκευμένη διαδικασία αναζήτησης μεταφοράς



Εικόνα 133. Αποθηκευμένη διαδικασία επιλογής μεταφοράς

```
/****** Object: StoredProcedure [dbo].[TransportUpdate]    Script Date: 29/11/2020 10:07:04 πμ *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[TransportUpdate]
(
    @TransportID int,
    @RouteID nvarchar(50),
    @TransportDate date,
    @AirplaneName nvarchar(20),
    @PilotID nvarchar(50),
    @CoPilotID nvarchar(50),
    @AirHostessID1 nvarchar(50),
    @AirHostessID2 nvarchar(50),
    @AirHostessID3 nvarchar(50),
    @AirHostessID4 nvarchar(50),
    @ManagerID nvarchar(50),
    @EcoSeats int,
    @BusSeats int
)
as
begin
    update
        [Transport]
    set
        [RouteID]=@RouteID,
        [TransportDate]=@TransportDate,
        [AirplaneName]=@AirplaneName,
        [PilotID]=@PilotID,
        [CoPilotID]=@CoPilotID,
        [AirHostessID1]=@AirHostessID1,
        [AirHostessID2]=@AirHostessID2,
        [AirHostessID3]=@AirHostessID3,
        [AirHostessID4]=@AirHostessID4,
        [ManagerID]=@ManagerID,
        [EcoSeats]=@EcoSeats,
        [BusSeats]=@BusSeats
    where
        [TransportID]=@TransportID;
end;
GO
```

Εικόνα 134. Αποθηκευμένη διαδικασία ανανέωσης μεταφοράς

Τμήμα Μηχανικών Πληροφορικής, ΕΛ.ΜΕ.ΠΑ Κρήτης

```

/***** Object: StoredProcedure [dbo].[TransportValidation]    Script Date: 29/11/2020 10:07:04 πμ *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Author,,Name>
-- Create date:    <Create Date,,>
-- Description:    <Description,,>
-- =====
CREATE PROCEDURE [dbo].[TransportValidation]

    @routeID nvarchar(50),
    @airplaneName nvarchar(20),
    @pilotID nvarchar(50),
    @coPilotID nvarchar(50),
    @airHostessID1 nvarchar(50),
    @airHostessID2 nvarchar(50),
    @airHostessID3 nvarchar(50),
    @airHostessID4 nvarchar(50),
    @transportDate date,
    @transportID int

AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

declare @dt time;
declare @at time;

select
    @dt = [DepartureTime],
    @at = [ArrivalTime]
from
    [Route]
where
    [ID] = @routeID;

declare @count1 int = 0;
declare @count2 int = 0;
declare @count3 int = 0;

-- Θέρνωμε το πληθος των transports στην επιλεγμένη ημερομηνία με το ίδιο αεροπλάνο που επικαλύπτονται
select
    @count1 = count([TransportID])
from
    [Transport] t inner join
    [Route] r on t.[RouteID] = r.[ID]
where
    (
        [AirplaneName] = @airplaneName or
        @pilotID in ([PilotID], [CoPilotID]) or
        @coPilotID in ([PilotID], [CoPilotID]) or
        @airHostessID1 in ([AirHostessID1], [AirHostessID2], [AirHostessID3], [AirHostessID4]) or
        @airHostessID2 in ([AirHostessID1], [AirHostessID2], [AirHostessID3], [AirHostessID4]) or
        @airHostessID3 in ([AirHostessID1], [AirHostessID2], [AirHostessID3], [AirHostessID4]) or
        @airHostessID4 in ([AirHostessID1], [AirHostessID2], [AirHostessID3], [AirHostessID4])
    )
    and
    [TransportDate] = @transportDate
    and
    (
        (@dt <= r.[DepartureTime] and @at >= r.[DepartureTime]) or
        (@dt >= r.[DepartureTime] and @at <= r.[ArrivalTime]) or
        (@dt >= r.[DepartureTime] and @dt <= r.[ArrivalTime] and @at >= r.[ArrivalTime]) or
        (@dt <= r.[DepartureTime] and @at >= r.[ArrivalTime])
    )
    and
    [TransportID] <> @transportID;

select
    @count2 = count(r2.DestinationPointID)
from
    [Transport] t2 inner join [Route] r2 on t2.[RouteID] = r2.[ID]
where
    r2.[DestinationPointID] in (select [DeparturePointID] from [Route] where [ID] = @routeID) and
    t2.[TransportDate] = @transportDate and
    t2.[AirplaneName] = @airplaneName and
    abs(DATEDIFF(MINUTE, r2.[ArrivalTime], @dt)) < 40 and
    t2.[TransportID] <> @transportID;

select
    @count3 = count(r2.DestinationPointID)
from
    [Transport] t2 inner join [Route] r2 on t2.[RouteID] = r2.[ID]
where
    r2.DestinationPointID in (select [DestinationPointID] from [Route] where [ID] = @routeID) and
    t2.[TransportDate] = @transportDate and
    t2.[AirplaneName] = @airplaneName and
    abs(DATEDIFF(MINUTE, @at, r2.[DepartureTime])) < 40 and
    t2.[TransportID] <> @transportID;

select @count1 + @count2 + @count3;

END
GO

```

Εικόνα 135. Αποθηκευμένη διαδικασία ελέγχου εγκυρότητας μεταφοράς

```

/***** Object: StoredProcedure [dbo].[UserAdministratorInsert] Script Date: 29/11/2020 10:07:04 μμ *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[UserAdministratorInsert]
(
    @Username nvarchar(50),
    @Password nvarchar(max),
    @ID nvarchar(50),
    @FirstName nvarchar(20),
    @LastName nvarchar(20),
    @Street nvarchar(50),
    @StreetNo nvarchar(20),
    @City nvarchar(20),
    @PhoneNo nvarchar(50),
    @BirthDate date
)
as
begin
    exec EmployeeInsert @ID, @FirstName, @LastName, @Street, @StreetNo, @City, @PhoneNo, @BirthDate;

    insert into [User]([Username],[Password],[ClientID],[AgentID],[AdministratorID])
    values(@Username,@Password,NULL,NULL,@ID);

end;
GO
    
```

Εικόνα 136. Αποθηκευμένη διαδικασία εισαγωγής χρήστη-διαχειριστή

```

/***** Object: StoredProcedure [dbo].[UserAdministratorSearch] Script Date: 29/11/2020 10:07:04 μμ *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author: Zacharoula
-- Create date: 21/6/2020
-- Description: <Description,>
-- =====
CREATE PROCEDURE [dbo].[UserAdministratorSearch]
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    select
        u.[Username],
        e.[ID],
        e.[FirstName],
        e.[LastName],
        e.[Street],
        e.[StreetNo],
        e.[City],
        e.[PhoneNo],
        e.[BirthDate]
    from
        [Employee] e inner join
        [User] u on u.[AdministratorID] = e.[ID]
    where
        not u.[AdministratorID] is null;

END
GO
    
```

Εικόνα 137. Αποθηκευμένη διαδικασία αναζήτησης χρήστη-διαχειριστή

```

/***** Object: StoredProcedure [dbo].[UserAdministratorUpdate] Script Date: 29/11/2020 10:07:04 μμ *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author: Zacharoula
-- Create date: 21/6/2020
-- Description: <Description,>
-- =====
CREATE PROCEDURE [dbo].[UserAdministratorUpdate]
@Username nvarchar(50),
@PreviousUsername nvarchar(50),
@ID nvarchar(50),
@PreviousID nvarchar(50),
@FirstName nvarchar(20),
@LastName nvarchar(20),
@Street nvarchar(50),
@StreetNo nvarchar(20),
@City nvarchar(20),
@PhoneNo nvarchar(50),
@BirthDate date
AS
BEGIN
    exec EmployeeUpdate @PreviousID, @ID, @FirstName, @LastName, @Street, @StreetNo, @City, @PhoneNo, @BirthDate;

    update
        [User]
    set
        [AdministratorID] = @ID,
        [Username] = @Username
    where
        [Username] = @PreviousUsername;

END
GO
    
```

Εικόνα 138. Αποθηκευμένη διαδικασία ανανέωσης χρήστη-διαχειριστή

```
Object Explorer
Rodosflights.sql - not connected* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
Quick Launch (Ctrl+Q)
Rodosflights
New Query
Execute
Object Explorer
Rodosflights.sql - not connected*
***** Object: StoredProcedure [dbo].[UserAgentInsert] Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[UserAgentInsert]
(
    @Username nvarchar(50),
    @Password nvarchar(max),
    @ID nvarchar(50),
    @FirstName nvarchar(20),
    @LastName nvarchar(20),
    @Street nvarchar(50),
    @StreetNo nvarchar(20),
    @City nvarchar(20),
    @Code nvarchar(50)
)
as
begin
    exec AgentInsert @ID, @FirstName, @LastName, @Street, @StreetNo, @City, @Code;
    insert into [User]([Username],[Password],[ClientID],[AgentID],[AdministratorID])
    values(@Username,@Password,NULL,@ID,NULL);
end;
GO
```

Εικόνα 139. Αποθηκευμένη διαδικασία εισαγωγής χρήστη-ταξιδιωτικού πράκτορα

```
Object Explorer
Rodosflights.sql - not connected* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
Quick Launch (Ctrl+Q)
Rodosflights
New Query
Execute
Object Explorer
Rodosflights.sql - not connected*
***** Object: StoredProcedure [dbo].[UserAgentSearch] Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author: Zacharoula
-- Create date: 21/5/2020
-- Description: <Description,,>
-- =====
CREATE PROCEDURE [dbo].[UserAgentSearch]
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

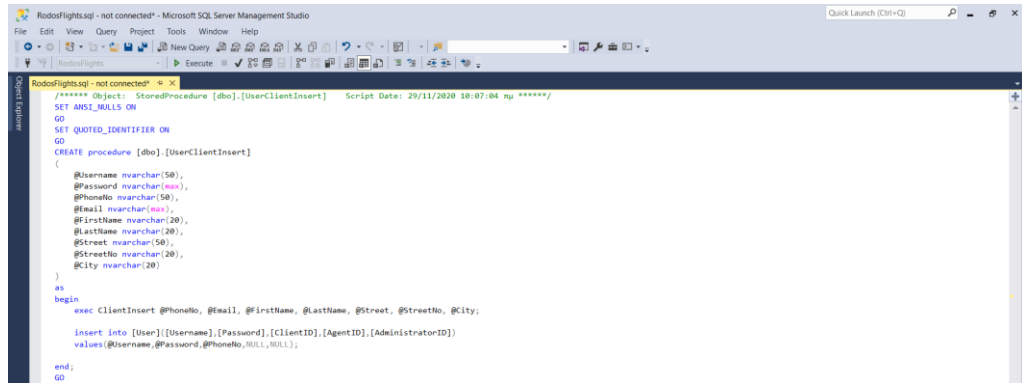
    select
        u.[Username],
        a.[ID],
        a.[FirstName],
        a.[LastName],
        a.[Street],
        a.[StreetNo],
        a.[City],
        a.[Code]
    from
        [User] u inner join
        [Agent] a on a.[ID] = u.[AgentID]
    where
        not u.[AgentID] is null;
END
GO
```

Εικόνα 140. Αποθηκευμένη διαδικασία αναζήτησης χρήστη-ταξιδιωτικού πράκτορα

```
Object Explorer
Rodosflights.sql - not connected* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
Quick Launch (Ctrl+Q)
Rodosflights
New Query
Execute
Object Explorer
Rodosflights.sql - not connected*
***** Object: StoredProcedure [dbo].[UserAgentUpdate] Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author: Zacharoula
-- Create date: 21/5/2020
-- Description: <Description,,>
-- =====
CREATE PROCEDURE [dbo].[UserAgentUpdate]
(
    @Username nvarchar(50),
    @PreviousUsername nvarchar(50),
    @ID nvarchar(50),
    @PreviousID nvarchar(50),
    @FirstName nvarchar(20),
    @LastName nvarchar(20),
    @Street nvarchar(50),
    @StreetNo nvarchar(20),
    @City nvarchar(20),
    @Code nvarchar(50)
)
AS
BEGIN
    exec AgentUpdate @PreviousID, @ID, @FirstName, @LastName, @Street, @StreetNo, @City, @Code;

    update
    [User]
    set
        [AgentID] = @ID,
        [Username] = @Username
    where
        [Username] = @PreviousUsername;
END
GO
```

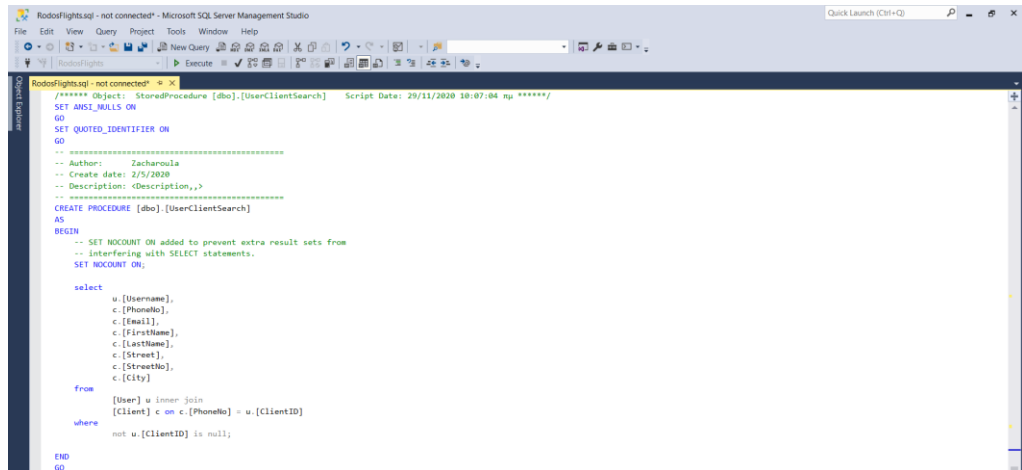
Εικόνα 141. Αποθηκευμένη διαδικασία ανανέωσης χρήστη-ταξιδιωτικού πράκτορα



```
Object Explorer
Rodoflights.sql - not connected* - Microsoft SQL Server Management Studio
Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[UserClientInsert]
(
    @Username nvarchar(50),
    @Password nvarchar(max),
    @PhoneNo nvarchar(50),
    @Email nvarchar(max),
    @FirstName nvarchar(20),
    @LastName nvarchar(20),
    @Street nvarchar(50),
    @StreetNo nvarchar(20),
    @City nvarchar(20)
)
as
begin
    exec ClientInsert @PhoneNo, @Email, @FirstName, @LastName, @Street, @StreetNo, @City;

    insert into [User]([Username],[Password],[ClientID],[AgentID],[AdministratorID]
    values(@Username,@Password,@PhoneNo,NULL,NULL);
end;
GO
```

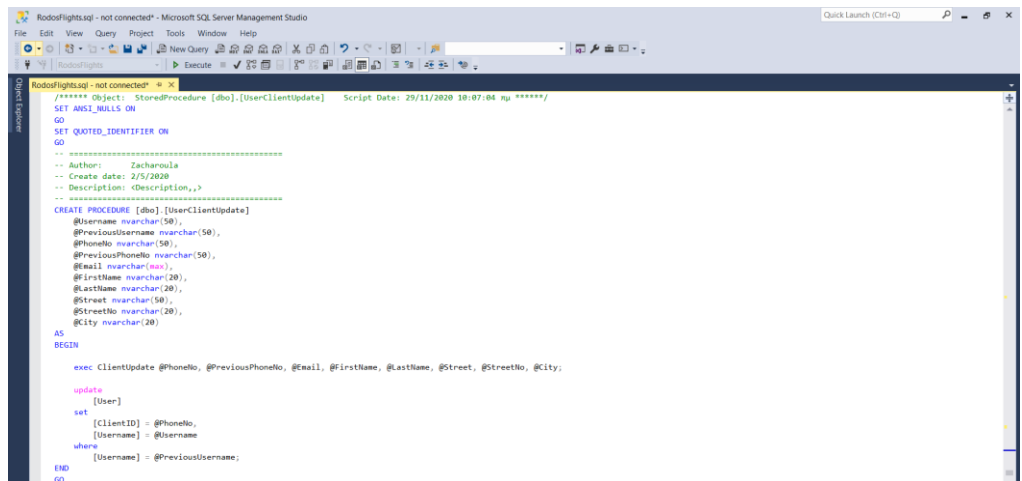
Εικόνα 142. Αποθηκευμένη διαδικασία εισαγωγής χρήστη-πελάτη



```
Object Explorer
Rodoflights.sql - not connected* - Microsoft SQL Server Management Studio
Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- Author: Zacharoula
-- Create date: 2/5/2020
-- Description: (Description,)>
CREATE PROCEDURE [dbo].[UserClientSearch]
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    select
        u.[Username],
        c.[PhoneNo],
        c.[Email],
        c.[FirstName],
        c.[LastName],
        c.[Street],
        c.[StreetNo],
        c.[City]
    from
        [User] u inner join
        [Client] c on c.[PhoneNo] = u.[ClientID]
    where
        not u.[ClientID] is null;
END
GO
```

Εικόνα 143. Αποθηκευμένη διαδικασία αναζήτησης χρήστη-πελάτη



```
Object Explorer
Rodoflights.sql - not connected* - Microsoft SQL Server Management Studio
Script Date: 29/11/2020 10:07:04 μμ *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- Author: Zacharoula
-- Create date: 2/5/2020
-- Description: (Description,)>
CREATE PROCEDURE [dbo].[UserClientUpdate]
(
    @Username nvarchar(50),
    @PreviousUsername nvarchar(50),
    @PhoneNo nvarchar(50),
    @PreviousPhoneNo nvarchar(50),
    @Email nvarchar(max),
    @FirstName nvarchar(20),
    @LastName nvarchar(20),
    @Street nvarchar(50),
    @StreetNo nvarchar(20),
    @City nvarchar(20)
)
AS
BEGIN
    exec ClientUpdate @PhoneNo, @PreviousPhoneNo, @Email, @FirstName, @LastName, @Street, @StreetNo, @City;

    update
        [User]
    set
        [ClientID] = @PhoneNo,
        [Username] = @Username
    where
        [Username] = @PreviousUsername;
END
GO
```

Εικόνα 144. Αποθηκευμένη διαδικασία ανανέωσης χρήστη-πελάτη

```
/****** Object: StoredProcedure [dbo].[UserDeleteByUsername] Script Date: 29/11/2020
10:07:04 πμ *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[UserDeleteByUsername]
(
    @Username nvarchar(50)
)
as
begin
    declare @ClientID nvarchar(50);
    declare @AgentID nvarchar(50);
    declare @AdministratorID nvarchar(50);

    select
        @ClientID=[ClientID],
        @AgentID=[AgentID],
        @AdministratorID=[AdministratorID]
    from
        [User]
    where
        [Username]=@Username;

    delete
    from
        [User]
    where
        [Username]=@Username;

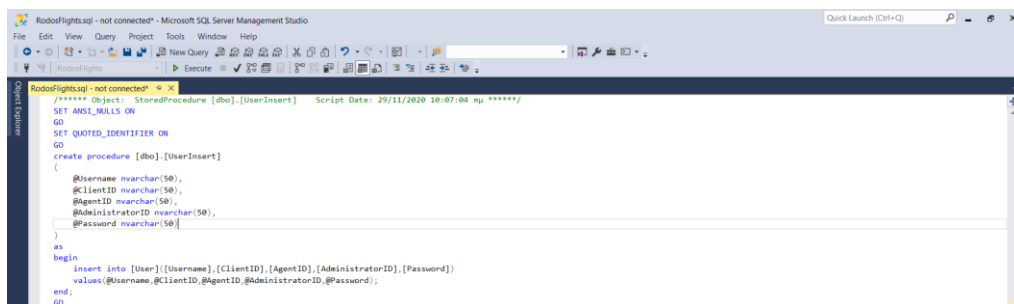
    delete
    from
        [Client]
    where
        [PhoneNo]=@ClientID;

    delete
    from
        [Agent]
    where
        [ID]=@AgentID;

    delete
    from
        [Employee]
    where
        [ID]=@AdministratorID;

end;
GO
```

Εικόνα 145. Αποθηκευμένη διαδικασία διαγραφής χρήστη



Εικόνα 146. Αποθηκευμένη διαδικασία εισαγωγής χρήστη

```

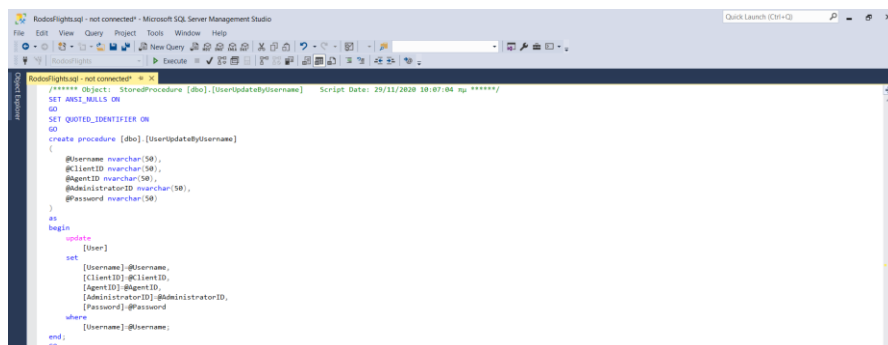
/***** Object: StoredProcedure [dbo].[UserSelectByUserName]    Script Date: 29/11/2020 10:07:04 πμ *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE procedure [dbo].[UserSelectByUserName]
(
    @userName nvarchar(50)
)
as
begin
    declare @clientID nvarchar(50);
    declare @agentID nvarchar(50);
    declare @administratorID nvarchar(50);

    select
        @clientID = [ClientID],
        @agentID = [AgentID],
        @administratorID = [AdministratorID]
    from
        [User]
    where
        [Username] = @username;

    if not @clientID is null
        select
            @userName,
            [PhoneNo],
            [Email],
            [FirstName],
            [LastName],
            [Street],
            [StreetNo],
            [City]
        from
            [Client]
        where
            [PhoneNo] = @clientID
    else if not @agentID is null
        select
            @userName,
            [ID],
            [FirstName],
            [LastName],
            [Street],
            [StreetNo],
            [City],
            [Code]
        from
            [Agent]
        where
            [ID] = @agentID
    else
        select
            @userName,
            [ID],
            [FirstName],
            [LastName],
            [Street],
            [StreetNo],
            [City],
            [PhoneNo],
            [BirthDate]
        from
            [Employee]
        where
            [ID] = @administratorID;
end;
GO

```

Εικόνα 147. Αποθηκευμένη διαδικασία επιλογής χρήστη



Εικόνα 148. Αποθηκευμένη διαδικασία ανανέωσης χρήστη

Τμήμα Μηχανικών Πληροφορικής, ΕΛ.ΜΕ.ΠΑ Κρήτης

```
{% load static %}
<!DOCTYPE html>

<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
    <link href="{% static 'FlightManagementSystem/css/AgentSearch.css' %}" rel="stylesheet" />
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>
    <script src="{% static 'FlightManagementSystem/js/AgentSearch.js' %}"></script>

  <title>ΠΡΑΚΤΟΡΑΣ</title>
</head>
<body>

  <div class="modal" id="ppMsg">
    <div class="modal-dialog">
      <div class="modal-content">
        <div class="modal-header">
          ΠΡΟΣΟΧΗ
        </div>
        <div class="modal-body">
          <div id="lblMsg">

          </div>
        </div>
        <div class="modal-footer">
          <div>
            <button type="button" class="btn btn-sm btn-secondary" onclick="onOKClick()">OK</button>
          </div>
        </div>
      </div>
    </div>
  </div>

  <div class="modal" id="ppWarning">
    <div class="modal-dialog">
      <div class="modal-content">
        <div class="modal-header">
          ΕΠΙΒΕΒΑΙΩΣΗ
        </div>
        <div class="modal-body">
          <div>
            Θέλετε σίγουρα να διαγράψετε τον ταξιδιωτικό πράκτορα;
          </div>
        </div>
        <div class="modal-footer">
          <div>
            <button type="button" class="btn btn-sm btn-dark" onclick="onYesClick()">Ναι</button>
          </div>
          <div>
            <button type="button" class="btn btn-sm btn-success" onclick="onNoClick()">Όχι</button>
          </div>
        </div>
      </div>
    </div>
  </div>

  <div style="margin-bottom:10px">
    <button type="button" class="btn btn-sm btn-success" onclick="onNewClick()">ΔΗΜΙΟΥΡΓΙΑ</button>
  </div>

  <table class="tbl-search">
    <tr>
      <th>Ενέργεια</th>
      <th>Αριθμός Ταυτότητας</th>
      <th>Όνομα</th>
      <th>Επώνυμο</th>
      <th>Οδός</th>
      <th>Αριθμός</th>
      <th>Πόλη</th>
      <th>Κωδικός</th>
    </tr>

    {% for a in model %}
      <tr>
        <td>
          <a href="javascript:void(0)" title="Προβολή" data-toggle="tooltip" onclick="viewAgent('{{ a.id }}')">
            <span class="material-icons" style="color:green">
              visibility
            </span>
          </a>
          <a href="javascript:void(0)" title="Ενεξέγρυσά" data-toggle="tooltip" onclick="editAgent('{{ a.id }}')">
            <span class="material-icons" style="color:black">
              create
            </span>
          </a>
          <a href="javascript:void(0)" onclick="deleteAgent('{{ a.id }}')" title="Διαγραφή" data-toggle="tooltip">
            <span class="material-icons" style="color:red">
              delete
            </span>
          </a>
        </td>
        <td>{{ a.id }}</td>
        <td>{{ a.firstName }}</td>
        <td>{{ a.lastName }}</td>
        <td>{{ a.street }}</td>
        <td class="td-centered">{{ a.streetNo }}</td>
        <td>{{ a.city }}</td>
        <td class="td-centered">{{ a.code }}</td>
      </tr>
    {% endfor %}
  </table>
</body>
</html>
```

Εικόνα 151. Σελίδα html αναζήτησης ταξιδιωτικού πράκτορα

Τμήμα Μηχανικών Πληροφορικής, ΕΛ.ΜΕ.ΠΑ Κρήτης

```
{% load static %}
<!DOCTYPE html>

<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
<link href="{% static 'FlightManagementSystem/css/AirHostessSearch.css' %}" rel="stylesheet" />
<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>
<script src="{% static 'FlightManagementSystem/js/AirHostessSearch.js' %}"></script>

<title>ΑΕΡΟΣΥΝΟΔΟΣ</title>
</head>
<body>

<div class="modal" id="ppMsg">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
ΠΡΟΣΟΧΗ
</div>
<div class="modal-body">
<div id="lblMsg">
</div>
</div>
<div class="modal-footer">
<div>
<button type="button" class="btn btn-sm btn-secondary" onclick="onOKClick()">OK</button>
</div>
</div>
</div>
</div>

<div class="modal" id="ppWarning">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
ΕΠΙΒΕΒΑΙΩΣΗ
</div>
<div class="modal-body">
<div>
Θέλετε σίγουρα να διαγράψετε την αεροσυνοδό;
</div>
</div>
<div class="modal-footer">
<div>
<button type="button" class="btn btn-sm btn-dark" onclick="onYesClick()">Ναι</button>
</div>
<div>
<button type="button" class="btn btn-sm btn-success" onclick="onNoClick()">Όχι</button>
</div>
</div>
</div>
</div>

<div style="margin-bottom:10px">
<button type="button" class="btn btn-sm btn-success" onclick="onNewClick()">ΔΗΜΙΟΥΡΓΙΑ</button>
</div>

<table class="tbl-search">
<tr>
<th>Ενέργεια</th>
<th>Αριθμός Ταυτότητας</th>
<th>Όνομα</th>
<th>Επίσημο</th>
<th>Όδος</th>
<th>Αριθμός</th>
<th>Πόλη</th>
<th>Τηλέφωνο</th>
<th>Ημερομηνία Γέννησης</th>
<th>Χρόνια Υπηρεσίας</th>
</tr>

{% for ah in model %}
<tr>
<td>
<a href="javascript:void(0)" title="Προβολή" data-toggle="tooltip" onclick="viewAirHostess('{{ ah.id }}')">
<span class="material-icons" style="color:green">
visibility
</span>
</a>
<a href="javascript:void(0)" title="Ενεξεργασία" data-toggle="tooltip" onclick="editAirHostess('{{ ah.id }}')">
<span class="material-icons" style="color:black">
create
</span>
</a>
<a href="javascript:void(0)" onclick="deleteAirHostess('{{ ah.id }}')" title="Αναγραφή" data-toggle="tooltip">
<span class="material-icons" style="color:red">
delete
</span>
</a>
</td>
<td>{{ ah.id }}</td>
<td>{{ ah.firstName }}</td>
<td>{{ ah.lastName }}</td>
<td>{{ ah.street }}</td>
<td class="td-centered">{{ ah.streetNo }}</td>
<td>{{ ah.city }}</td>
<td>{{ ah.phoneNo }}</td>
<td class="td-centered">{{ ah.birthDate }}</td>
<td class="td-centered">{{ ah.years }}</td>
</tr>
{% endfor %}
</table>
</body>
</html>
```

Εικόνα 153. Σελίδα html αναζήτησης αεροσυνοδού

Τμήμα Μηχανικών Πληροφορικής, ΕΛ.ΜΕ.ΠΑ Κρήτης

```
{% load static %}
<!DOCTYPE html>

<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
    <link href="{% static 'FlightManagementSystem/css/AirplaneSearch.css' %}" rel="stylesheet" />
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>
    <script src="{% static 'FlightManagementSystem/js/AirplaneSearch.js' %}"></script>

    <title>ΑΕΡΟΠΛΑΝΟ</title>
  </head>
  <body>
    <div class="modal" id="ppMsg">
      <div class="modal-dialog">
        <div class="modal-content">
          <div class="modal-header">
            ΠΡΟΣΟΧΗ
          </div>
          <div class="modal-body">
            <div id="lblMsg">
            </div>
          </div>
          <div class="modal-footer">
            <div>
              <button type="button" class="btn btn-sm btn-secondary" onclick="onOKClick()">OK</button>
            </div>
          </div>
        </div>
      </div>
    </div>

    <div class="modal" id="ppWarning">
      <div class="modal-dialog">
        <div class="modal-content">
          <div class="modal-header">
            ΕΠΙΒΕΒΑΙΩΣΗ
          </div>
          <div class="modal-body">
            <div>
              Θέλετε σίγουρα να διαγράψετε το αεροπλάνο;
            </div>
          </div>
          <div class="modal-footer">
            <div>
              <button type="button" class="btn btn-sm btn-dark" onclick="onYesClick()">Ναι</button>
            </div>
            <div>
              <button type="button" class="btn btn-sm btn-success" onclick="onNoClick()">Όχι</button>
            </div>
          </div>
        </div>
      </div>
    </div>

    <div style="margin-bottom:10px">
      <button type="button" class="btn btn-sm btn-success" onclick="onNewClick()">ΔΗΜΙΟΥΡΓΙΑ</button>
    </div>

    <table class="tbl-search">
      <tr>
        <th>Ενέργεια</th>
        <th>Όνομασία</th>
        <th>Αριθμός Διακεκριμένων Θέσεων</th>
        <th>Αριθμός Οικονομικών Θέσεων</th>
      </tr>
    </table>

    {% for ap in model %}
      <tr>
        <td>
          <a href="javascript:void(0)" title="Προβολή" data-toggle="tooltip" onclick="viewAirplane('{{ ap.name }}')">
            <span class="material-icons" style="color:green">
              visibility
            </span>
          </a>
          <a href="javascript:void(0)" title="Επεξεργασία" data-toggle="tooltip" onclick="editAirplane('{{ ap.name }}')">
            <span class="material-icons" style="color:black">
              create
            </span>
          </a>
          <a href="javascript:void(0)" onclick="deleteAirplane('{{ ap.name }}'" title="Διαγραφή" data-toggle="tooltip">
            <span class="material-icons" style="color:red">
              delete
            </span>
          </a>
        </td>
        <td class="td-centered">{{ ap.name }}</td>
        <td class="td-centered">{{ ap.numBusinessSeats }}</td>
        <td class="td-centered">{{ ap.numEconomySeats }}</td>
      </tr>
    </table>
  </body>
</html>
```

Εικόνα 155. Σελίδα html αναζήτησης αεροπλάνου

```
{% load static %}
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
    <link href="{% static 'FlightManagementSystem/css/ClientDashboard.css' %}" rel="stylesheet" />
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>
    <script src="{% static 'FlightManagementSystem/js/ClientDashboard.js' %}"></script>
    <title>ΤΑΜΙΩ</title>
  </head>
  <body>
    <div class="card">
      <div class="card-header">
        ΠΡΟΣΩΠΙΚΑ ΣΤΟΙΧΕΙΑ
      </div>
      <div class="card-body">
        <table class="tbl-info">
          <tr>
            <td style="font-weight: bold;text-align:right">
              Ονοματεπώνυμο:
            </td>
            <td>
              {{ info.fullName }}
            </td>
          </tr>
          <tr>
            <td style="font-weight: bold;text-align:right">
              Διεύθυνση:
            </td>
            <td>
              {{ info.street }}
            </td>
          </tr>
          <tr>
            <td style="font-weight: bold;text-align:right">
              Πόλη:
            </td>
            <td>
              {{ info.city }}
            </td>
          </tr>
          <tr>
            <td style="font-weight: bold;text-align:right">
              Τηλέφωνο:
            </td>
            <td>
              {{ info.phoneNo }}
            </td>
          </tr>
          <tr>
            <td style="font-weight: bold;text-align:right">
              Ηλεκτρονική Διεύθυνση:
            </td>
            <td>
              {{ info.eMail }}
            </td>
          </tr>
          <tr>
            <td style="font-weight: bold;text-align:right">
              Username:
            </td>
            <td>
              {{ info.username }}
            </td>
          </tr>
          <tr>
            <td style="font-weight: bold;text-align:right">
              Κιλιόμετρα:
            </td>
            <td>
              {{ info.totalKm }}
            </td>
          </tr>
        </table>
      </div>
    </div>
    <div class="card" style="margin-top:20px">
      <div class="card-header bg-secondary" style="color:white">
        ΚΡΑΤΗΣΕΙΣ
      </div>
      <div class="card-body">
        <table class="tbl-search">
          <tr>
            <th style="min-width:100px">Κωδικός κράτησης</th>
            <th style="min-width:100px">Ημερομηνία κράτησης</th>
            <th style="min-width:200px">Πόλη αναχώρησης</th>
            <th style="min-width:200px">Πόλη άφιξης</th>
          </tr>
          <tr>
            <td colspan="4">
              {% for res in reservations %}
            <td class="td-centered">{{ res.reservationID }}</td>
            <td class="td-centered">{{ res.resDate }}</td>
            <td>{{ res.depCity}}</td>
            <td>{{ res.arrCity}}</td>
          </tr>
          <tr>
            <td colspan="4">
              {% endfor %}
            </td>
          </tr>
        </table>
      </div>
    </div>
    <div class="card" style="margin-top:20px">
      <div class="card-header bg-info" style="color:white">
        ΕΙΣΗΤΗΡΙΑ
      </div>
      <div class="card-body">
        <table class="tbl-search">
          <tr>
            <th style="min-width:100px">Κωδικός εισιτηρίου</th>
            <th style="min-width:100px">Ημερομηνία έκδοσης</th>
            <th style="min-width:200px">Πόλη αναχώρησης</th>
            <th style="min-width:200px">Πόλη άφιξης</th>
          </tr>
          <tr>
            <td colspan="4">
              {% for t in tickets %}
            <td class="td-centered">{{ t.ticketID }}</td>
            <td class="td-centered">{{ t.resDate }}</td>
            <td>{{ t.depCity}}</td>
            <td>{{ t.arrCity}}</td>
          </tr>
          <tr>
            <td colspan="4">
              {% endfor %}
            </td>
          </tr>
        </table>
      </div>
    </div>
  </body>
</html>
```

Εικόνα 156. Σελίδα html ταμπλό πελάτη

Τμήμα Μηχανικών Πληροφορικής, ΕΛ.ΜΕ.ΠΑ Κρήτης

```
{% load static %}
<!DOCTYPE html>

<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
<link href="{% static 'FlightManagementSystem/css/ClientSearch.css' %}" rel="stylesheet" />
<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>
<script src="{% static 'FlightManagementSystem/js/ClientSearch.js' %}"></script>

<title>ΠΕΛΑΤΗΣ</title>
</head>

<body>
<div class="modal" id="ppMsg">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
ΠΡΟΣΟΧΗ
</div>
<div class="modal-body">
<div id="lblMsg">
</div>
</div>
<div class="modal-footer">
<div>
<button type="button" class="btn btn-sm btn-secondary" onclick="onOKClick()">OK</button>
</div>
</div>
</div>
</div>

<div class="modal" id="ppWarning">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
ΕΠΙΒΕΒΑΙΩΣΗ
</div>
<div class="modal-body">
<div>
Θέλετε σίγουρα να διαγράψετε το πελάτη;
</div>
</div>
<div class="modal-footer">
<div>
<button type="button" class="btn btn-sm btn-dark" onclick="onYesClick()">Ναι</button>
</div>
<div>
<button type="button" class="btn btn-sm btn-success" onclick="onNoClick()">Όχι</button>
</div>
</div>
</div>
</div>

<div style="margin-bottom:10px">
<button type="button" class="btn btn-sm btn-success" onclick="onNewClick()">ΔΗΜΙΟΥΡΓΙΑ</button>
</div>

<table class="tbl-search">
<tr>
<th>Ενέργεια</th>
<th>Αριθμός Τηλεφώνου</th>
<th>Ηλεκτρονική διεύθυνση</th>
<th>Όνομα</th>
<th>Επίσημο</th>
<th>Όδος</th>
<th>Αριθμός</th>
<th>Πόλη</th>
</tr>

{% for c in model %}
<tr>
<td>
<a href="javascript:void(0)" title="Προβολή" data-toggle="tooltip" onclick="viewClient('{{ c.phoneNo }}')">
<span class="material-icons" style="color:green">
visibility
</span>
</a>
<a href="javascript:void(0)" title="Επεξεργασία" data-toggle="tooltip" onclick="editClient('{{ c.phoneNo }}')">
<span class="material-icons" style="color:black">
create
</span>
</a>
<a href="javascript:void(0)" onclick="deleteClient('{{ c.phoneNo }}')" title="Διαγραφή" data-toggle="tooltip">
<span class="material-icons" style="color:red">
delete
</span>
</a>
</td>
<td>{{ c.phoneNo }}</td>
<td>{{ c.eMail }}</td>
<td>{{ c.firstName }}</td>
<td>{{ c.lastName }}</td>
<td>{{ c.street }}</td>
<td class="td-centered">{{ c.streetNo }}</td>
<td>{{ c.city }}</td>
</tr>
{% endfor %}
</table>
</body>
</html>
```

Εικόνα 158. Σελίδα html αναζήτησης πελάτη


```
{% load static %}
<!DOCTYPE html>

<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
    <link href="{% static 'FlightManagementSystem/css/ClientReservationSearch.css' %}" rel="stylesheet" />
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>
    <script src="{% static 'FlightManagementSystem/js/ClientReservationSearch.js' %}"></script>

    <title>ΚΡΑΤΗΣΗ</title>
  </head>
  <body>
    <div class="modal" id="ppMsg">
      <div class="modal-dialog">
        <div class="modal-content">
          <div class="modal-header">
            ΠΡΟΣΟΧΗ
          </div>
          <div class="modal-body">
            <div id="lblMsg">
              </div>
            </div>
          <div class="modal-footer">
            <div>
              <button type="button" class="btn btn-sm btn-secondary" onclick="onOKClick()">OK</button>
            </div>
          </div>
        </div>
      </div>
    </div>

    <div class="modal" id="ppWarning">
      <div class="modal-dialog">
        <div class="modal-content">
          <div class="modal-header">
            ΕΠΙΒΕΒΑΙΩΣΗ
          </div>
          <div class="modal-body">
            <div>
              Θέλετε σίγουρα να διαγράψετε τη κράτηση;
            </div>
          </div>
          <div class="modal-footer">
            <div>
              <button type="button" class="btn btn-sm btn-dark" onclick="onYesClick()">Ναι</button>
            </div>
            <div>
              <button type="button" class="btn btn-sm btn-success" onclick="onNoClick()">Όχι</button>
            </div>
          </div>
        </div>
      </div>
    </div>

    <div style="margin-bottom:10px">
      <button type="button" class="btn btn-sm btn-success" onclick="onNewClick()">ΔΗΜΙΟΥΡΓΙΑ</button>
    </div>

    <table class="tbl-search">
      <tr>
        <th style="min-width:102px">Ενέργεια</th>
        <th style="min-width:100px">Κωδικός κράτησης</th>
        <th style="min-width:100px">Ημερομηνία κράτησης</th>
        <th style="min-width:100px">Ημερομηνία λήξης</th>
        <th style="min-width:200px">Πόλη αναχώρησης</th>
        <th style="min-width:200px">Πόλη άφιξης</th>
        <th style="min-width:100px">Κατηγορία θέσης</th>
        <th style="min-width:100px">Τιμή</th>
      </tr>

      {% for res in model %}
      <tr>
        <td>
          <a href="javascript:void(0)" title="Προβολή" data-toggle="tooltip" onclick="viewReservation('{{ res.reservationID }}')">
            <span class="material-icons" style="color:green">
              visibility
            </span>
          </a>
          <a href="javascript:void(0)" title="Επεξεργασία" data-toggle="tooltip" onclick="editReservation('{{ res.reservationID }}')">
            <span class="material-icons" style="color:black">
              create
            </span>
          </a>
          <a href="javascript:void(0)" onclick="deleteReservation('{{ res.reservationID }}')" title="Διαγραφή" data-toggle="tooltip">
            <span class="material-icons" style="color:red">
              delete
            </span>
          </a>
        </td>
        <td class="td-centered">{{ res.reservationID }}</td>
        <td class="td-centered">{{ res.resDate }}</td>
        <td class="td-centered">{{ res.expDate }}</td>
        <td>{{ res.depCity }}</td>
        <td>{{ res.arvCity }}</td>
        <td>{{ res.seatCategory }}</td>
        <td>{{ res.price }}</td>
      </tr>
      {% endfor %}
    </table>
  </body>
</html>
```

Εικόνα 160. Σελίδα html αναζήτησης κράτησης πελάτη

```
{% load static %}
<!DOCTYPE html>

<html>

<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
<link href="{% static 'FlightManagementSystem/css/top-banner.css' %}" rel="stylesheet" />
<link href="{% static 'FlightManagementSystem/css/ticketPreview.css' %}" rel="stylesheet" />
<link rel="shortcut icon" href="{% static 'FlightManagementSystem/images/favicon.ico' %}" />

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>
<script src="{% static 'FlightManagementSystem/js/ClientReservationEdit.js' %}"></script>

<title>ΕΙΛΗΘΡΗ</title>
</head>

<body>
<div>
<div class="top-section">
<div>

</div>
<div class="user-details-section">
<div class="username">
{% if currentUser.isClient == True %}
<a style="color:white" href="/userClientEdit/{{ currentUser.username }}/edit" target="_blank">{{ currentUser.firstName }} {{ currentUser.lastName }} ({{ currentUser.username }})</a>
{% elif currentUser.isAgent == True %}
<a style="color:white" href="/userAgentEdit/{{ currentUser.username }}/edit" target="_blank">{{ currentUser.firstName }} {{ currentUser.lastName }} ({{ currentUser.username }})</a>
{% else %}
<a style="color:white" href="/userAdministratorEdit/{{ currentUser.username }}/edit" target="_blank">{{ currentUser.firstName }} {{ currentUser.lastName }} ({{ currentUser.username }})</a>
{% endif %}

</div>
<div>

</div>
</div>
</div>
<div class="ticket">

<div style="margin-top:20px">
<div class="field">
<div class="lbl">
Αριθμός εισιτηρίου
</div>
<div>
{{ model.ticketID }}
</div>
</div>
<div class="field">
<div class="lbl">
Όνοματίς
</div>
<div>
{{ model.fullName }}
</div>
</div>
<div class="field">
<div class="lbl">
Ημερομηνία κράτησης
</div>
<div>
{{ model.resDate }}
</div>
</div>
<div class="field">
<div class="lbl">
Πόλη αναχώρησης
</div>
<div>
{{ model.depCity }}
</div>
</div>
<div class="field">
<div class="lbl">
Πόλη άφιξης
</div>
<div>
{{ model.arrCity }}
</div>
</div>
<div class="field">
<div class="lbl">
Διακεκρυμμένη θέση
</div>
<div>
{% if model.busClass == True %}
Ηαυ
{% else %}
Οχι
{% endif %}
</div>
</div>
<div class="field">
<div class="lbl">
Τιμή
</div>
<div>
{{ model.price }}€
</div>
</div>
<div class="field">
<div class="lbl">
Χιλιόμετρα
</div>
<div>
{{ model.kilometers }}km
</div>
</div>
</div>
</div>
</body>
</html>
```

Εικόνα 161. Σελίδα html προεπισκόπησης εισιτηρίου στην οθόνη του πελάτη

```
(% load static %)
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
    <link href="{% static 'FlightManagementSystem/css/top-banner.css' %}" rel="stylesheet" />
    <link href="{% static 'FlightManagementSystem/css/TicketPreview.css' %}" rel="stylesheet" />
    <link rel="shortcut icon" href="{% static 'FlightManagementSystem/images/favicon.ico' %}" />
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>
    <script src="{% static 'FlightManagementSystem/js/TicketPreview.js' %}"></script>
    <title>ΕΙΠΗΠΙΟ</title>
  </head>
  <body style="background-image:url({% static 'FlightManagementSystem/images/preview.jpg' %});">
    <input type="hidden" id="hf" value="{% model.price %}">
    <div>
      <div class="top-section">
        <div>
          
        </div>
        <div class="user-details-section">
          <div class="username">
            {% if currentUser.isClient == True %}
            <a style="color:white" href="/userClientEdit/{% currentUser.username %}/edit" target="_blank">{{ currentUser.firstName }} {{ currentUser.lastName }} ({{ currentUser.username }})</a>
            {% elif currentUser.isAgent == True %}
            <a style="color:white" href="/userAgentEdit/{% currentUser.username %}/edit" target="_blank">{{ currentUser.firstName }} {{ currentUser.lastName }} ({{ currentUser.username }})</a>
            {% else %}
            <a style="color:white" href="/userAdministratorEdit/{% currentUser.username %}/edit" target="_blank">{{ currentUser.firstName }} {{ currentUser.lastName }} ({{ currentUser.username }})</a>
            {% endif %}
          </div>
          <div>
            
          </div>
        </div>
      </div>
      <div class="modal" id="ppMsg">
        <div class="modal-dialog">
          <div class="modal-content">
            <div class="modal-header">
              ΠΡΟΞΕΧΗ
            </div>
            <div class="modal-body">
              <div id="lblMsg">
                Ανακοίνωση από το εστιατήριο &#pgrv;
              </div>
            </div>
            <div class="modal-footer">
              <div>
                <button type="button" class="btn btn-sm btn-secondary" onclick="onOKClick()">OK</button>
              </div>
            </div>
          </div>
        </div>
      </div>
      <div class="ticket">
        
        <div style="margin-top:20px">
          <div class="field">
            <div class="lbl">
              Αριθμός εισιτηρίου
            </div>
            <div>
              {{ model.ticketID }}
            </div>
          </div>
          <div class="field">
            <div class="lbl">
              Πλεόνασμα
            </div>
            <div>
              {{ model.fullName }}
            </div>
          </div>
          <div class="field">
            <div class="lbl">
              Ημερομηνία πτήσης
            </div>
            <div>
              {{ model.resDate }}
            </div>
          </div>
          <div class="field">
            <div class="lbl">
              Πόλη αναχώρησης
            </div>
            <div>
              {{ model.depCity }}
            </div>
          </div>
          <div class="field">
            <div class="lbl">
              Πόλη άφιξης
            </div>
            <div>
              {{ model.arrCity }}
            </div>
          </div>
          <div class="field">
            <div class="lbl">
              Διακεκριμένη θέση
            </div>
            <div>
              {% if model.busClass == True %}
              Ναι
              {% else %}
              Όχι
              {% endif %}
            </div>
          </div>
          <div class="field">
            <div class="lbl">
              Τιμή
            </div>
            <div>
              {{ model.price }} &#euro;
            </div>
          </div>
          <div class="field">
            <div class="lbl">
              Χιλιόμετρα
            </div>
            <div>
              {{ model.kilometers }}
            </div>
          </div>
        </div>
      </div>
    </body>
  </html>
```

Εικόνα 162. Σελίδα html προεπισκόπησης εισιτηρίου στην οθόνη του ταξιδιωτικού πράκτορα

```

{% load static %}
<!DOCTYPE html>

<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
    <link href="{% static 'FlightManagementSystem/css/top-banner.css' %}" rel="stylesheet" />
    <link href="{% static 'FlightManagementSystem/css/Home.css' %}" rel="stylesheet" />
    <link rel="shortcut icon" href="{% static 'FlightManagementSystem/images/favicon.ico' %}" />

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>
    <script src="{% static 'FlightManagementSystem/js/Home.js' %}"></script>

    <title>ΑΡΧΙΚΗ ΣΕΛΙΔΑ</title>
  </head>
  <body>
    <div class="modal" id="ppMsg">
      <div class="modal-dialog">
        <div class="modal-content">
          <div class="modal-header">
            ΠΡΟΣΟΧΗ
          </div>
          <div class="modal-body">
            <div id="lblMsg">
            </div>
          </div>
          <div class="modal-footer">
            <div>
              <button type="button" class="btn btn-sm btn-secondary" onClick="onOKClick()">OK</button>
            </div>
          </div>
        </div>
      </div>
    </div>
    <div class="top-section">
      <div>
        
      </div>
      <div class="user-details-section">
        <div class="username">
          {% if currentUser.isClient == True %}
            <a style="color:white" href="/userClientEdit/{{ currentUser.username }}/edit" target="_blank">{{ currentUser.firstName }} {{ currentUser.lastName }} ({{ currentUser.username }})</a>
          {% elif currentUser.isAgent == True %}
            <a style="color:white" href="/userAgentEdit/{{ currentUser.username }}/edit" target="_blank">{{ currentUser.firstName }} {{ currentUser.lastName }} ({{ currentUser.username }})</a>
          {% else %}
            <a style="color:white" href="/userAdministratorEdit/{{ currentUser.username }}/edit" target="_blank">{{ currentUser.firstName }} {{ currentUser.lastName }} ({{ currentUser.username }})</a>
          {% endif %}
        </div>
        <div>
          
        </div>
      </div>
    </div>
    <div class="main-section">
      <div class="left-section">
        {% if currentUser.isAdministrator %}
          <div id="tabPilots" class="tab" onClick="onPilotsClick()">
            ΠΙΛΟΤΟΙ
          </div>
          <div id="tabAirHostesses" class="tab" onClick="onAirHostessesClick()">
            ΑΕΡΟΠΗΓΑΙΩΤΕΣ
          </div>
          <div id="tabAirplanes" class="tab" onClick="onAirplanesClick()">
            ΑΕΡΟΠΛΑΝΑ
          </div>
          <div id="tabAgents" class="tab" onClick="onAgentsClick()">
            ΤΑΞΙΑΓΟΡΕΥΤΕΣ ΠΡΑΚΤΟΡΕΣ
          </div>
          <div id="tabManagers" class="tab" onClick="onManagersClick()">
            ΔΙΟΙΚΗΤΙΚΟΙ ΥΠΑΜΜΟΙ
          </div>
        {% endif %}
        {% if currentUser.isAgent %}
          <div id="tabClients" class="tab" onClick="onClientsClick()">
            ΠΕΛΑΤΕΣ
          </div>
        {% endif %}
        {% if currentUser.isAdministrator %}
          <div id="tabUsers" class="tab" onClick="onUsersClick()">
            ΧΡΗΣΤΕΣ
          </div>
          <div id="tabRoutes" class="tab" onClick="onRoutesClick()">
            ΔΡΟΜΟΛΟΓΙΑ
          </div>
          <div id="tabTransports" class="tab" onClick="onTransportsClick()">
            ΜΕΤΑΦΟΡΕΣ
          </div>
          <div id="tabReport" class="tab" onClick="onReportClick()">
            ΠΑΡΗΡΗΤΗΣΕΙΣ
          </div>
        {% endif %}
        {% if currentUser.isAgent %}
          <div id="tabReservations" class="tab" onClick="onReservationsClick()">
            ΚΡΑΤΗΣΕΙΣ
          </div>
        {% endif %}
        {% if currentUser.isClient %}
          <div id="tabDashboard" class="tab" onClick="onClientDashboardClick()">
            ΤΑΜΙΩ
          </div>
          <div id="tabReservations" class="tab" onClick="onClientReservationsClick()">
            ΚΡΑΤΗΣΕΙΣ
          </div>
        {% endif %}
      </div>
      <div class="right-section">
        <iframe id="frm"></iframe>
      </div>
    </div>
  </body>
</html>

```

Εικόνα 163. Σελίδα html αρχικής σελίδας

```
{% load static %}
<!DOCTYPE html>

<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
    <link href="{% static 'FlightManagementSystem/css/Login.css' %}" rel="stylesheet" />
    <link rel="shortcut icon" href="{% static 'FlightManagementSystem/images/favicon.ico' %}" />

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>
    <script src="{% static 'FlightManagementSystem/js/Login.js' %}"></script>

    <title>ΕΙΣΟΔΟΣ ΧΡΗΣΤΗ</title>
  </head>

  <body style="background-image:url('{% static 'FlightManagementSystem/images/login.jpg' %});">

    <div class="modal" id="ppMsg">
      <div class="modal-dialog">
        <div class="modal-content">
          <div class="modal-header">
            ΠΡΟΣΟΧΗ
          </div>
          <div class="modal-body">
            <div id="lblMsg">

            </div>
          </div>
          <div class="modal-footer">
            <div>
              <button type="button" class="btn btn-sm btn-secondary" onclick="onOKClick()">OK</button>
            </div>
          </div>
        </div>
      </div>
    </div>

    <div style="display:flex;justify-content: center;">
      <div class="panel">
        <div>
          
        </div>
        <div style="margin-bottom:15px;margin-top:20px">
          <div>
            Όνομα Χρήστη
          </div>
          <div>
            <input type="text" id="txtUserName" class="form-control form-control-sm txt" />
          </div>
        </div>

        <div>
          <div>
            Κωδικός Πρόσβασης
          </div>
          <div>
            <input type="password" id="txtPassword" class="form-control form-control-sm txt" />
          </div>
        </div>

        <div class="button-section">
          <div style="margin-right:10px">
            <button type="button" class="btn btn-sm btn-success" onclick="onSignInClick()">
              Σύνδεση
            </button>
          </div>
          <div style="margin-right:10px">
            <button type="button" class="btn btn-sm btn-info" onclick="onSignUpClick()">
              Εγγραφή
            </button>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

Εικόνα 164. Σελίδα html εισόδου χρήστη

Τμήμα Μηχανικών Πληροφορικής, ΕΛ.ΜΕ.ΠΑ Κρήτης

```
{% load static %}
<!DOCTYPE html>

<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
    <link href="{% static 'FlightManagementSystem/css/ManagerSearch.css' %}" rel="stylesheet" />
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>
    <script src="{% static 'FlightManagementSystem/js/ManagerSearch.js' %}"></script>

    <title>ΔΙΟΙΚΗΤΙΚΟΣ ΥΠΑΛΛΗΛΟΣ</title>
  </head>
  <body>

    <div class="modal" id="ppMsg">
      <div class="modal-dialog">
        <div class="modal-content">
          <div class="modal-header">
            ΠΡΟΣΟΧΗ
          </div>
          <div class="modal-body">
            <div id="lblMsg">
            </div>
          </div>
          <div class="modal-footer">
            <div>
              <button type="button" class="btn btn-sm btn-secondary" onclick="onOKClick()">OK</button>
            </div>
          </div>
        </div>
      </div>
    </div>

    <div class="modal" id="ppWarning">
      <div class="modal-dialog">
        <div class="modal-content">
          <div class="modal-header">
            ΕΠΙΒΕΒΑΙΩΣΗ
          </div>
          <div class="modal-body">
            <div>
              Θέλετε σίγουρα να διαγράψετε το διοικητικό υπάλληλο;
            </div>
          </div>
          <div class="modal-footer">
            <div>
              <button type="button" class="btn btn-sm btn-dark" onclick="onYesClick()">Ναι</button>
            </div>
            <div>
              <button type="button" class="btn btn-sm btn-success" onclick="onNoClick()">Όχι</button>
            </div>
          </div>
        </div>
      </div>
    </div>

    <div style="margin-bottom:10px">
      <button type="button" class="btn btn-sm btn-success" onclick="onNewClick()">ΔΗΜΙΟΥΡΓΙΑ</button>
    </div>

    <table class="tbl-search">
      <tr>
        <th>Ενέργεια</th>
        <th>Αριθμός Ταυτότητας</th>
        <th>Όνομα</th>
        <th>Επώνυμο</th>
        <th>Οδός</th>
        <th>Αριθμός</th>
        <th>Πόλη</th>
        <th>Τηλέφωνο</th>
        <th>Ημερομηνία Γέννησης</th>
        <th>Εκπαίδευση</th>
        <th>Χρόνια Υπηρεσίας</th>
      </tr>

      {% for m in model %}
      <tr>
        <td style="min-width: 105px;width:105px">
          <a href="javascript:void(0)" title="Προβολή" data-toggle="tooltip" onclick="viewManager('{{ m.id }}')">
            <span class="material-icons" style="color:green">
              visibility
            </span>
          </a>
          <a href="javascript:void(0)" title="Επεξεργασία" data-toggle="tooltip" onclick="editManager('{{ m.id }}')">
            <span class="material-icons" style="color:black">
              create
            </span>
          </a>
          <a href="javascript:void(0)" onclick="deleteManager('{{ m.id }}')" title="Διαγραφή" data-toggle="tooltip">
            <span class="material-icons" style="color:red">
              delete
            </span>
          </a>
        </td>
        <td>{{ m.id }}</td>
        <td>{{ m.firstName }}</td>
        <td>{{ m.lastName }}</td>
        <td>{{ m.street }}</td>
        <td class="td-centered">{{ m.streetNo }}</td>
        <td>{{ m.city }}</td>
        <td>{{ m.phoneno }}</td>
        <td class="td-centered">{{ m.birthDate }}</td>
        <td class="td-centered">{{ m.education }}</td>
        <td class="td-centered">{{ m.years }}</td>
      </tr>
      {% endfor %}
    </table>
  </body>
</html>
```

Εικόνα 166. Σελίδα html αναζήτησης διοικητικού υπαλλήλου

Τμήμα Μηχανικών Πληροφορικής, ΕΛ.ΜΕ.ΠΑ Κρήτης

```
{% load static %}
<!DOCTYPE html>

<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
    <link href="{% static 'FlightManagementSystem/css/PilotSearch.css' %}" rel="stylesheet" />
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>
    <script src="{% static 'FlightManagementSystem/js/PilotSearch.js' %}"></script>

    <title>ΠΙΛΩΤΟΣ</title>
  </head>

  <body>
    <div class="modal" id="ppMsg">
      <div class="modal-dialog">
        <div class="modal-content">
          <div class="modal-header">
            ΠΡΟΣΟΧΗ
          </div>
          <div class="modal-body">
            <div id="lblMsg">
            </div>
          </div>
          <div class="modal-footer">
            <div>
              <button type="button" class="btn btn-sm btn-secondary" onclick="onOKClick()">OK</button>
            </div>
          </div>
        </div>
      </div>
    </div>

    <div class="modal" id="ppWarning">
      <div class="modal-dialog">
        <div class="modal-content">
          <div class="modal-header">
            ΕΠΙΒΕΒΑΙΩΣΗ
          </div>
          <div class="modal-body">
            <div>
              Θέλετε σίγουρα να διαγράψετε το πιλότο;
            </div>
          </div>
          <div class="modal-footer">
            <div>
              <button type="button" class="btn btn-sm btn-dark" onclick="onYesClick()">Ναι</button>
            </div>
            <div>
              <button type="button" class="btn btn-sm btn-success" onclick="onNoClick()">Όχι</button>
            </div>
          </div>
        </div>
      </div>
    </div>

    <div style="margin-bottom:10px">
      <button type="button" class="btn btn-sm btn-success" onclick="onNewClick()">ΔΗΜΙΟΥΡΓΙΑ</button>
    </div>

    <table class="tbl-search">
      <tr>
        <th>Ενέργεια</th>
        <th>Αριθμός Ταυτότητας</th>
        <th>Όνομα</th>
        <th>Επώνυμο</th>
        <th>Όδος</th>
        <th>Αριθμός</th>
        <th>Πόλη</th>
        <th>Τηλέφωνο</th>
        <th>Ημερομηνία Γέννησης</th>
        <th>Ώρες Πτήσης</th>
      </tr>

      {% for p in model %}
        <tr>
          <td>
            <a href="javascript:void(0)" title="Προβολή" data-toggle="tooltip" onclick="viewPilot('{{ p.id }}')">
              <span class="material-icons" style="color:green">
                visibility
              </span>
            </a>
            <a href="javascript:void(0)" title="Επεξεργασία" data-toggle="tooltip" onclick="editPilot('{{ p.id }}')">
              <span class="material-icons" style="color:black">
                create
              </span>
            </a>
            <a href="javascript:void(0)" onclick="deletePilot('{{ p.id }}')" title="Διαγραφή" data-toggle="tooltip">
              <span class="material-icons" style="color:red">
                delete
              </span>
            </a>
          </td>
          <td>{{ p.id }}</td>
          <td>{{ p.firstName }}</td>
          <td>{{ p.lastName }}</td>
          <td>{{ p.street }}</td>
          <td class="td-centered">{{ p.streetNo }}</td>
          <td>{{ p.city }}</td>
          <td>{{ p.phonello }}</td>
          <td class="td-centered">{{ p.birthDate }}</td>
          <td class="td-centered">{{ p.flightHours }}</td>
        </tr>
      {% endfor %}
    </table>
  </body>
</html>
```

Εικόνα 168. Σελίδα html αναζήτησης πιλότου

```
{% load static %}
<!DOCTYPE html>

<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
    <link href="{% static 'FlightManagementSystem/css/Report.css' %}" rel="stylesheet" />
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script src="https://cdn.jsdelivr.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>
    <script src="{% static 'FlightManagementSystem/js/Report.js' %}"></script>

    <title>ΣΥΓΚΕΝΤΡΩΤΙΚΗ ΑΝΑΦΟΡΑ</title>
  </head>

  <body>
    <div>
      <h4>ΣΥΓΚΕΝΤΡΩΤΙΚΗ ΑΝΑΦΟΡΑ</h4>
    </div>
    <div class="container-1">
      <div class="data-field">
        <div class="lbl">
          Από
        </div>

        <div>
          <input type="date" id="dtFrom" class="form-control form-control-sm" onchange="disableToDates()" />
        </div>
      </div>

      <div class="data-field" style="margin-left:20px">
        <div class="lbl">
          Μέχρι
        </div>

        <div>
          <input type="date" id="dtTo" class="form-control form-control-sm" onchange="disableFromDates()" />
        </div>
      </div>

      <div style="margin-left:20px">
        <button type="button" class="btn btn-primary btn-sm" onclick="onShowClick()">Προβολή</button>
      </div>
    </div>

    <div id='report'>
    </div>
  </body>
</html>
```

Εικόνα 169. Σελίδα html συγκεντρωτικής αναφοράς πτήσεων

Τμήμα Μηχανικών Πληροφορικής, ΕΛ.ΜΕ.ΠΑ Κρήτης

```
{% load static %}
<!DOCTYPE html>

<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
    <link href="{% static 'FlightManagementSystem/css/ReservationSearch.css' %}" rel="stylesheet" />
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>
    <script src="{% static 'FlightManagementSystem/js/ReservationSearch.js' %}"></script>

    <title>ΚΡΑΤΗΣΗ</title>
  </head>
  <body>
    <div class="modal" id="ppMsg">
      <div class="modal-dialog">
        <div class="modal-content">
          <div class="modal-header">
            ΠΡΟΣΟΧΗ
          </div>
          <div class="modal-body">
            <div id="lblMsg">
            </div>
          </div>
          <div class="modal-footer">
            <div>
              <button type="button" class="btn btn-sm btn-secondary" onclick="onOKClick()">OK</button>
            </div>
          </div>
        </div>
      </div>
    </div>

    <div class="modal" id="ppWarning">
      <div class="modal-dialog">
        <div class="modal-content">
          <div class="modal-header">
            ΕΠΙΒΕΒΑΙΩΣΗ
          </div>
          <div class="modal-body">
            <div>
              Θέλετε σίγουρα να διαγράψετε τη κράτηση;
            </div>
          </div>
          <div class="modal-footer">
            <div>
              <button type="button" class="btn btn-sm btn-dark" onclick="onYesClick()">Ναι</button>
            </div>
            <div>
              <button type="button" class="btn btn-sm btn-success" onclick="onNoClick()">Όχι</button>
            </div>
          </div>
        </div>
      </div>
    </div>

    <div style="margin-bottom:10px">
      <button type="button" class="btn btn-sm btn-success" onclick="onNewClick()">ΔΗΜΙΟΥΡΓΙΑ</button>
    </div>

    <table class="tbl-search">
      <tr>
        <th style="min-width:102px">Ενέργεια</th>
        <th style="min-width:100px">Κωδικός κράτησης</th>
        <th style="min-width:100px">Ημερομηνία κράτησης</th>
        <th style="min-width:100px">Ημερομηνία λήξης</th>
        <th style="min-width:200px">Πόλη αναχώρησης</th>
        <th style="min-width:200px">Πόλη άφιξης</th>
        <th style="min-width:100px">Κατηγορία θέσης</th>
        <th style="min-width:100px">Τιμή</th>
        <th style="min-width:200px">Πελάτης</th>
        <th style="min-width:200px">Ταξιδιωτικός πράκτορας</th>
      </tr>

      {% for res in model %}
        <tr>
          <td>
            <a href="javascript:void(0)" title="Προβολή" data-toggle="tooltip" onclick="viewReservation('{{ res.reservationID }}')">
              <span class="material-icons" style="color:green">
                visibility
              </span>
            </a>
            <a href="javascript:void(0)" title="Επεξεργασία" data-toggle="tooltip" onclick="editReservation('{{ res.reservationID }}')">
              <span class="material-icons" style="color:black">
                create
              </span>
            </a>
            <a href="javascript:void(0)" onclick="deleteReservation('{{ res.reservationID }}')" title="Διαγραφή" data-toggle="tooltip">
              <span class="material-icons" style="color:red">
                delete
              </span>
            </a>
          </td>
          <td class="td-centered">{{ res.reservationID }}</td>
          <td class="td-centered">{{ res.resDate }}</td>
          <td class="td-centered">{{ res.expDate }}</td>
          <td>{{ res.depCity }}</td>
          <td>{{ res.arrCity }}</td>
          <td>{{ res.seatCategory }}</td>
          <td>{{ res.price }}</td>
          <td>{{ res.client }}</td>
          <td>{{ res.agent }}</td>
        </tr>
      {% endfor %}
    </table>
  </body>
</html>
```

Εικόνα 171. Σελίδα html αναζήτησης κράτησης ταξιδιωτικού πράκτορα

Τμήμα Μηχανικών Πληροφορικής, ΕΛ.ΜΕ.ΠΑ Κρήτης

```
{% load static %}
<!DOCTYPE html>

<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
    <link href="{% static 'FlightManagementSystem/css/RouteSearch.css' %}" rel="stylesheet" />
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>
    <script src="{% static 'FlightManagementSystem/js/RouteSearch.js' %}"></script>

    <title>ΔΡΟΜΟΛΟΓΙΟ</title>
  </head>
  <body>
    <div class="modal" id="ppMsg">
      <div class="modal-dialog">
        <div class="modal-content">
          <div class="modal-header">
            ΠΡΟΣΟΧΗ
          </div>
          <div class="modal-body">
            <div id="lblMsg">
            </div>
          </div>
          <div class="modal-footer">
            <div>
              <button type="button" class="btn btn-sm btn-secondary" onclick="onOKClick()">OK</button>
            </div>
          </div>
        </div>
      </div>
    </div>

    <div class="modal" id="ppWarning">
      <div class="modal-dialog">
        <div class="modal-content">
          <div class="modal-header">
            ΕΠΙΒΕΒΑΙΩΣΗ
          </div>
          <div class="modal-body">
            <div>
              Θέλετε σίγουρα να διαγράψετε το δρομολόγιο;
            </div>
          </div>
          <div class="modal-footer">
            <div>
              <button type="button" class="btn btn-sm btn-dark" onclick="onYesClick()">Ναι</button>
            </div>
            <div>
              <button type="button" class="btn btn-sm btn-success" onclick="onNoClick()">Όχι</button>
            </div>
          </div>
        </div>
      </div>
    </div>

    <div style="margin-bottom:10px">
      <button type="button" class="btn btn-sm btn-success" onclick="onNewClick()">ΔΗΜΙΟΥΡΓΙΑ</button>
    </div>

    <table class="tbl-search">
      <tr>
        <th>Ενέργεια</th>
        <th style="width:100px">Κωδικός</th>
        <th style="width:240px">Αφετηρία</th>
        <th style="width:240px">Προορισμός</th>
        <th style="width:110px">Ημέρα πρoγμ/σης</th>
        <th style="width:110px">Απόσταση (km)</th>
        <th style="width:80px">Ήρα αναγ/σης</th>
        <th style="width:80px">Ήρα άφιξης</th>
        <th style="width:100px">Τιμή διακεκρυμένης θέσης (€euro)</th>
        <th style="width:100px">Τιμή οικονομικής θέσης (€euro)</th>
      </tr>

      {% for r in model %}
        <tr>
          <td>
            <a href="javascript:void(0)" title="Προβολή" data-toggle="tooltip" onclick="viewRoute('{{ r.id }}')">
              <span class="material-icons" style="color:green">visibility</span>
            </a>
            <a href="javascript:void(0)" title="Επεξεργασία" data-toggle="tooltip" onclick="editRoute('{{ r.id }}')">
              <span class="material-icons" style="color:black">create</span>
            </a>
            <a href="javascript:void(0)" onclick="deleteRoute('{{ r.id }}')" title="Διαγραφή" data-toggle="tooltip">
              <span class="material-icons" style="color:red">delete</span>
            </a>
          </td>
          <td>{{ r.id }}</td>
          <td>{{ r.depPoint }}</td>
          <td>{{ r.destPoint }}</td>
          <td class="td-centered">{{ r.day }}</td>
          <td class="td-centered">{{ r.distance }}</td>
          <td class="td-centered">{{ r.depTime }}</td>
          <td class="td-centered">{{ r.arrTime }}</td>
          <td class="td-centered">{{ r.busPrice }}</td>
          <td class="td-centered">{{ r.ecoPrice }}</td>
        </tr>
      {% endfor %}
    </table>
  </body>
</html>
```

Εικόνα 173. Σελίδα html αναζήτησης δρομολογίου

```
[% load static %]
<!DOCTYPE html>

<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
    <link href="{% static 'flightManagementSystem/css/top-hammer.css' %}" rel="stylesheet" />
    <link href="{% static 'flightManagementSystem/css/transportEdit.css' %}" rel="stylesheet" />
    <link rel="shortcut icon" href="{% static 'flightManagementSystem/images/favicon.ico' %}" />
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/umd/popper.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>
    <script src="{% static 'flightManagementSystem/js/transportEdit.js' %}"></script>
  </head>
  <body>
    <input type="hidden" id="hf" value="{ model.id }" />
    <div class="modal" id="ppMag">
      <div class="modal-dialog">
        <div class="modal-content">
          <div class="modal-header">
            ΠΡΟΒΟΗ
          </div>
          <div class="modal-body">
            <div id="tblMag">
            </div>
          </div>
          <div class="modal-footer">
            <div>
              <button type="button" class="btn btn-sm btn-secondary" onclick="onClick()">OK</button>
            </div>
          </div>
        </div>
      </div>
    </div>
    <div class="modal" id="ppMarving">
      <div class="modal-dialog">
        <div class="modal-content">
          <div class="modal-header">
            ΕΠΙΒΕΒΑΙΩΣΗ
          </div>
          <div class="modal-body">
            <div>
              Θέλτε σίγουρα να διαγράψετε τη μεταφορά;
            </div>
          </div>
          <div class="modal-footer">
            <div>
              <button type="button" class="btn btn-sm btn-dark" onclick="onYesClick()">Ναι</button>
            </div>
            <div>
              <button type="button" class="btn btn-sm btn-success" onclick="onClick()">Όχι</button>
            </div>
          </div>
        </div>
      </div>
    </div>
    <div>
      <div class="top-section">
        <div>
          
        </div>
        <div class="user-details-section">
          <div class="username">
            {% if currentUser.isClient == True %}
            <a style="color:white" href="/userClientEdit/{{ currentUser.username }}/edit" target="_blank">{{ currentUser.firstName }} {{ currentUser.lastName }} ({{ currentUser.username }})</a>
            {% elif currentUser.isAgent == True %}
            <a style="color:white" href="/userAgentEdit/{{ currentUser.username }}/edit" target="_blank">{{ currentUser.firstName }} {{ currentUser.lastName }} ({{ currentUser.username }})</a>
            {% else %}
            <a style="color:white" href="/userAdministratorEdit/{{ currentUser.username }}/edit" target="_blank">{{ currentUser.firstName }} {{ currentUser.lastName }} ({{ currentUser.username }})</a>
            {% endif %}
          </div>
          <div>
            
          </div>
        </div>
      </div>
      <div class="buttons">
        <div {% if model.mode == 'edit' or model.isNew == True %} hidden {% endif %}>
          <button type="button" class="btn btn-success btn-sm" onclick="onEditClick(false)">Επεξεργασία</button>
        </div>
        <div {% if model.mode == 'view' %} hidden {% endif %}>
          <button type="button" class="btn btn-success btn-sm" onclick="onSaveClick(false)">Αποθήκευση</button>
        </div>
        <div style="margin-left: 10px; {% if model.mode == 'view' %} hidden {% endif %}>
          <button type="button" class="btn btn-success btn-sm" onclick="onSaveClick(true)">Αποθήκευση &
            Μεταφορά</button>
        </div>
        <div style="margin-left: 10px; {% if model.isNew == True %} hidden {% endif %}>
          <button type="button" class="btn btn-dark btn-sm" onclick="onDeleteClick()">Διαγραφή</button>
        </div>
        <div style="margin-left: 10px; {% if model.mode == 'view' or model.isNew == True %} hidden {% endif %}>
          <button type="button" class="btn btn-dark btn-sm" onclick="onCancelClick()">Ακύρωση</button>
        </div>
      </div>
      <div style="margin-left: 20px">
        <div class="panel">
          <div class="field">
            <div class="tbl">
              Κωδικός μεταφοράς
            </div>
            <div>
              <input type="text" id="txtTransportID" disabled value="{ model.transportID }"
                class="form-control-sm">
            </div>
          </div>
          <div class="field">
            <div class="tbl">
              Αεροπλάνο
            </div>
            <div>
              <select id="cmbRoute" class="cmb form-control" {% if model.mode == 'view' %} disabled {% endif %}>
                {% for r in routes %}
                <option value="{ r.routeID }" selected="{ r.name }"></option>
                <option value="{ r.routeID }">{{ r.name }}</option>
                {% endfor %}
              </select>
            </div>
          </div>
          <div class="field">
            <div class="tbl">
              Ημερομηνία
            </div>
            <div>
              <input type="date" id="date" value="{ model.transportDate }"
                class="form-control form-control-sm" {% if model.mode == 'view' %} disabled {% endif %}>
            </div>
          </div>
          <div style="display: flex">
            <div class="field">
              <div class="tbl">
                Αεροπλάνο
              </div>
              <div>
                <select onchange="onPlaneChange()" id="cmbPlane" class="cmb form-control"
                  {% if model.mode == 'view' %} disabled {% endif %}>
                  {% for a in airplanes %}
                  <option value="{ model.airplaneName %}" selected="{ a.name }"></option>
                  <option value="{ a.name }">{{ a.name }}</option>
                  {% endfor %}
                </select>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

Εικόνα 174. Σελίδα html επεξεργασίας μεταφοράς (1)

```

<div class="field">
  <div class="lbl">
    Οικονομικές θέσεις
  </div>
  <div>
    <input type="text" id="txtEcoSeats" class="form-control-sm" disabled
      value="{{ model.ecoSeats }}" />
  </div>
</div>
</div>
<div class="field">
  <div class="lbl">
    Διατεταγμένες θέσεις
  </div>
  <div>
    <input type="text" id="txtBusSeats" class="form-control-sm" disabled
      value="{{ model.busSeats }}" />
  </div>
</div>
</div>
<div class="field">
  <div class="lbl">
    Υμouθuoc
  </div>
  <div>
    <select id="cmbManager" class="cmb form-control" {% if model.mode == 'view' %} disabled {% endif %}>
      {% for m in managers %}
      {% if m.managerID == model.managerID %}
      <option value="{{ m.managerID }}" selected>{{ m.name }}</option>
      {% else %}
      <option value="{{ m.managerID }}">{{ m.name }}</option>
      {% endif %}
      {% endfor %}
    </select>
  </div>
</div>
</div>
<div class="panel">
  <div class="field">
    <div class="lbl">
      Κυβερνήτης
    </div>
    <div>
      <select id="cmbPilot" class="cmb form-control" {% if model.mode == 'view' %} disabled {% endif %}>
        {% for p in pilots %}
        {% if p.pilotID == model.pilotID %}
        <option value="{{ p.pilotID }}" selected>{{ p.name }}</option>
        {% else %}
        <option value="{{ p.pilotID }}">{{ p.name }}</option>
        {% endif %}
        {% endfor %}
      </select>
    </div>
  </div>
  <div class="field">
    <div class="lbl">
      Συνουβερνήτης
    </div>
    <div>
      <select id="cmbCoPilot" class="cmb form-control" {% if model.mode == 'view' %} disabled {% endif %}>
        {% for p in pilots %}
        {% if p.pilotID == model.coPilotID %}
        <option value="{{ p.pilotID }}" selected>{{ p.name }}</option>
        {% else %}
        <option value="{{ p.pilotID }}">{{ p.name }}</option>
        {% endif %}
        {% endfor %}
      </select>
    </div>
  </div>
</div>
<div class="panel-noheader">
  <div class="field">
    <div class="lbl">
      Αεροσυνοδός 1
    </div>
    <div>
      <select id="cmbHostess1" class="cmb form-control" {% if model.mode == 'view' %} disabled
        {% endif %}>
        {% for a in airHostesses %}
        {% if a.airHostessID == model.airHostessID1 %}
        <option value="{{ a.airHostessID }}" selected>{{ a.name }}</option>
        {% else %}
        <option value="{{ a.airHostessID }}">{{ a.name }}</option>
        {% endif %}
        {% endfor %}
      </select>
    </div>
  </div>
  <div class="field">
    <div class="lbl">
      Αεροσυνοδός 2
    </div>
    <div>
      <select id="cmbHostess2" class="cmb form-control" {% if model.mode == 'view' %} disabled
        {% endif %}>
        {% for a in airHostesses %}
        {% if a.airHostessID == model.airHostessID2 %}
        <option value="{{ a.airHostessID }}" selected>{{ a.name }}</option>
        {% else %}
        <option value="{{ a.airHostessID }}">{{ a.name }}</option>
        {% endif %}
        {% endfor %}
      </select>
    </div>
  </div>
  <div class="field">
    <div class="lbl">
      Αεροσυνοδός 3
    </div>
    <div>
      <select id="cmbHostess3" class="cmb form-control" {% if model.mode == 'view' %} disabled
        {% endif %}>
        {% for a in airHostesses %}
        {% if a.airHostessID == model.airHostessID3 %}
        <option value="{{ a.airHostessID }}" selected>{{ a.name }}</option>
        {% else %}
        <option value="{{ a.airHostessID }}">{{ a.name }}</option>
        {% endif %}
        {% endfor %}
      </select>
    </div>
  </div>
  <div class="field">
    <div class="lbl">
      Αεροσυνοδός 4
    </div>
    <div>
      <select id="cmbHostess4" class="cmb form-control" {% if model.mode == 'view' %} disabled
        {% endif %}>
        {% for a in airHostesses %}
        {% if a.airHostessID == model.airHostessID4 %}
        <option value="{{ a.airHostessID }}" selected>{{ a.name }}</option>
        {% else %}
        <option value="{{ a.airHostessID }}">{{ a.name }}</option>
        {% endif %}
        {% endfor %}
      </select>
    </div>
  </div>
</div>
</div>
</body>
</html>

```

Εικόνα 175. Σελίδα html επεξεργασίας μεταφοράς (2)

Τμήμα Μηχανικών Πληροφορικής, ΕΛ.ΜΕ.ΠΑ Κρήτης

```
{% load static %}
<!DOCTYPE html>

<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
    <link href="{% static 'FlightManagementSystem/css/TransportSearch.css' %}" rel="stylesheet" />
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>
    <script src="{% static 'FlightManagementSystem/js/TransportSearch.js' %}"></script>

    <title>ΜΕΤΑΦΟΡΑ</title>
  </head>

  <body>
    <div class="modal" id="ppMsg">
      <div class="modal-dialog">
        <div class="modal-content">
          <div class="modal-header">
            ΠΡΟΣΟΧΗ
          </div>
          <div class="modal-body">
            <div id="lblMsg">
            </div>
          </div>
          <div class="modal-footer">
            <div>
              <button type="button" class="btn btn-sm btn-secondary" onclick="onClick()">OK</button>
            </div>
          </div>
        </div>
      </div>
    </div>

    <div class="modal" id="ppWarning">
      <div class="modal-dialog">
        <div class="modal-content">
          <div class="modal-header">
            ΕΠΙΒΕΒΑΙΩΣΗ
          </div>
          <div class="modal-body">
            <div>
              Θέλετε σίγουρα να διαγράψετε τη μεταφορά;
            </div>
          </div>
          <div class="modal-footer">
            <div>
              <button type="button" class="btn btn-sm btn-dark" onclick="onYesClick()">Ναι</button>
            </div>
            <div>
              <button type="button" class="btn btn-sm btn-success" onclick="onNoClick()">Όχι</button>
            </div>
          </div>
        </div>
      </div>
    </div>

    <div style="margin-bottom:10px">
      <button type="button" class="btn btn-sm btn-success" onclick="onNewClick()">ΔΗΜΙΟΥΡΓΙΑ</button>
    </div>

    <table class="tbl-search">
      <tr>
        <th style="min-width:102px">Ενέργεια</th>
        <th style="min-width:100px">Κωδικός μεταφοράς</th>
        <th style="min-width:300px">Δρομολόγιο</th>
        <th style="min-width:100px">Ημερομηνία</th>
        <th style="min-width:200px">Αεροπλάνο</th>
        <th style="min-width:300px">Κυβερνήτης</th>
        <th style="min-width:300px">Συγκυβερνήτης</th>
        <th style="min-width:300px">1η Αεροσυνοδός</th>
        <th style="min-width:300px">2η Αεροσυνοδός</th>
        <th style="min-width:300px">3η Αεροσυνοδός</th>
        <th style="min-width:300px">4η Αεροσυνοδός</th>
        <th style="min-width:300px">Υπεύθυνος</th>
      </tr>

      {% for t in model %}
        <tr>
          <td>
            <a href="javascript:void(0)" title="Προβολή" data-toggle="tooltip" onclick="viewTransport('{{ t.transportID }}')">
              <span class="material-icons" style="color:green">
                visibility
              </span>
            </a>
            <a href="javascript:void(0)" title="Επεξεργασία" data-toggle="tooltip" onclick="editTransport('{{ t.transportID }}')">
              <span class="material-icons" style="color:black">
                create
              </span>
            </a>
            <a href="javascript:void(0)" onclick="deleteTransport('{{ t.transportID }}')" title="Διαγραφή" data-toggle="tooltip">
              <span class="material-icons" style="color:red">
                delete
              </span>
            </a>
          </td>
          <td class="td-centered">{{ t.transportID }}</td>
          <td>{{ t.route }}</td>
          <td class="td-centered">{{ t.transportDate }}</td>
          <td>{{ t.airplaneName }}</td>
          <td>{{ t.pilot }}</td>
          <td>{{ t.coPilot }}</td>
          <td>{{ t.airHostess1 }}</td>
          <td>{{ t.airHostess2 }}</td>
          <td>{{ t.airHostess3 }}</td>
          <td>{{ t.airHostess4 }}</td>
          <td>{{ t.manager }}</td>
        </tr>
      {% endfor %}
    </table>
  </body>
</html>
```

Εικόνα 176. Σελίδα html αναζήτησης μεταφοράς


```

<div style="margin-left:20px">
  <div class="field">
    <div class="lbl">
      Αριθμός Ταυτότητας
    </div>
    <div>
      <input type="text" id="txtID" class="form-control form-control-sm txt" value="{{ model.id }}"
        {% if model.mode == 'view' %} disabled {% endif %}/>
    </div>
  </div>

  <div class="field">
    <div class="lbl">
      Όνομα
    </div>
    <div>
      <input type="text" id="txtFirstName" class="form-control form-control-sm txt" value="{{ model.firstName }}"
        {% if model.mode == 'view' %} disabled {% endif %}/>
    </div>
  </div>

  <div class="field">
    <div class="lbl">
      Επώνυμο
    </div>
    <div>
      <input type="text" id="txtLastName" class="form-control form-control-sm txt" value="{{ model.lastName }}"
        {% if model.mode == 'view' %} disabled {% endif %}/>
    </div>
  </div>

  <div style="display:flex">
    <div class="field">
      <div class="lbl">
        Οδός
      </div>
      <div>
        <input type="text" id="txtStreet" class="form-control form-control-sm txt" value="{{ model.street }}"
          {% if model.mode == 'view' %} disabled {% endif %}/>
      </div>
    </div>

    <div class="field">
      <div class="lbl">
        Αριθμός
      </div>
      <div>
        <input type="text" id="txtStreetNo" class="form-control form-control-sm" value="{{ model.streetNo }}"
          {% if model.mode == 'view' %} disabled {% endif %}/>
      </div>
    </div>
  </div>

  <div class="field">
    <div class="lbl">
      Πόλη
    </div>
    <div>
      <input type="text" id="txtCity" class="form-control form-control-sm txt" value="{{ model.city }}"
        {% if model.mode == 'view' %} disabled {% endif %}/>
    </div>
  </div>

  <div class="field">
    <div class="lbl">
      Τηλέφωνο
    </div>
    <div>
      <input type="text" id="txtPhoneNo" class="form-control form-control-sm txt" value="{{ model.phoneNo }}"
        {% if model.mode == 'view' %} disabled {% endif %}/>
    </div>
  </div>

  <div class="field">
    <div class="lbl">
      Ημερομηνία Γέννησης
    </div>
    <div>
      <input type="date" id="txtBirthDate" class="form-control form-control-sm txt" value="{{ model.birthDate }}" {% if model.mode == 'view' %} disabled {% endif %} />
    </div>
  </div>

  <div class="field">
    <div class="lbl">
      Όνομα χρήστη
    </div>
    <div>
      <input type="text" id="txtUsername" class="form-control form-control-sm txt" value="{{ model.username }}"
        {% if model.mode == 'view' %} disabled {% endif %}/>
    </div>
  </div>

  <div class="field" {% if model.username != '' %} hidden {% endif %}>
    <div class="lbl">
      Κωδικός πρόσβασης
    </div>
    <div>
      <input type="password" id="txtPassword" class="form-control form-control-sm txt" />
    </div>
  </div>

  <div class="field" {% if model.username != '' %} hidden {% endif %}>
    <div class="lbl">
      Επβεβαίωση
    </div>
    <div>
      <input type="password" id="txtConfirm" class="form-control form-control-sm txt" />
    </div>
  </div>
</div>
</body>
</html>

```

Εικόνα 178. Σελίδα html επεξεργασίας χρήστη-διαχειριστή (2)

Τμήμα Μηχανικών Πληροφορικής, ΕΛ.ΜΕ.ΠΑ Κρήτης

```
{% load static %}
<!DOCTYPE html>

<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
<link href="{% static 'FlightManagementSystem/css/UserAdministratorSearch.css' %}" rel="stylesheet" />
<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>
<script src="{% static 'FlightManagementSystem/js/UserAdministratorSearch.js' %}"></script>

<title>ΧΡΗΣΤΗΣ</title>
</head>
<body>
<div class="modal" id="ppMsg">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
ΠΡΟΣΟΧΗ
</div>
<div class="modal-body">
<div id="lblMsg">
</div>
</div>
<div class="modal-footer">
<div>
<button type="button" class="btn btn-sm btn-secondary" onclick="onOKClick()">OK</button>
</div>
</div>
</div>
</div>
<div class="modal" id="ppWarning">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
ΕΠΙΒΕΒΑΙΩΣΗ
</div>
<div class="modal-body">
<div>
Θέλετε σίγουρα να διαγράψετε το διαχειριστή;
</div>
</div>
<div class="modal-footer">
<div>
<button type="button" class="btn btn-sm btn-dark" onclick="onYesClick()">Ναι</button>
</div>
<div>
<button type="button" class="btn btn-sm btn-success" onclick="onNoClick()">Όχι</button>
</div>
</div>
</div>
</div>
<div style="margin-bottom:10px">
<button type="button" class="btn btn-sm btn-success" onclick="onNewClick()">ΔΗΜΙΟΥΡΓΙΑ</button>
</div>
<table class="tbl-search">
<tr>
<th style="min-width:101px">Ενέργεια</th>
<th>Όνομα Χρήστη</th>
<th>Αριθμός Ταυτότητας</th>
<th>Όνομα</th>
<th>Επίνομο</th>
<th>Όδος</th>
<th>Αριθμός</th>
<th>Πόλη</th>
<th>Τηλέφωνο</th>
<th>Ημερομηνία Γέννησης</th>
</tr>
<tr>
<td>
<a href="javascript:void(0)" title="Προβολή" data-toggle="tooltip" onclick="viewUserAdministrator('{{ uad.username }}')">
<span class="material-icons" style="color:green">visibility</span>
</a>
<a href="javascript:void(0)" title="Επεξεργασία" data-toggle="tooltip" onclick="editUserAdministrator('{{ uad.username }}')">
<span class="material-icons" style="color:black">create</span>
</a>
<a href="javascript:void(0)" onclick="deleteUserAdministrator('{{ uad.username }}')" title="Διαγραφή" data-toggle="tooltip">
<span class="material-icons" style="color:red">delete</span>
</a>
</td>
<td class="td-centered">{{ uad.username }}</td>
<td class="td-centered">{{ uad.id }}</td>
<td class="td-centered">{{ uad.firstName }}</td>
<td class="td-centered">{{ uad.lastName }}</td>
<td class="td-centered">{{ uad.street }}</td>
<td class="td-centered">{{ uad.streetNo }}</td>
<td class="td-centered">{{ uad.city }}</td>
<td class="td-centered">{{ uad.phonello }}</td>
<td class="td-centered">{{ uad.birthDate }}</td>
</tr>
<tr>
<td colspan="10" style="text-align:center">
</td>
</tr>
</table>
</body>
</html>
```

Εικόνα 179. Σελίδα html αναζήτησης χρήστη-διαχειριστή


```

<div style="margin-left:20px">
  <div class="field">
    <div class="lbl">
      Αριθμός Ταυτότητας
    </div>
    <div>
      <input type="text" id="txtID" class="form-control form-control-sm txt" value="{{ model.id }}"
        {% if model.mode == 'view' %} disabled {% endif %} />
    </div>
  </div>

  <div class="field">
    <div class="lbl">
      Όνομα
    </div>
    <div>
      <input type="text" id="txtFirstName" class="form-control form-control-sm txt"
        value="{{ model.firstName }}" {% if model.mode == 'view' %} disabled {% endif %} />
    </div>
  </div>

  <div class="field">
    <div class="lbl">
      Επώνυμο
    </div>
    <div>
      <input type="text" id="txtLastName" class="form-control form-control-sm txt"
        value="{{ model.lastName }}" {% if model.mode == 'view' %} disabled {% endif %} />
    </div>
  </div>

  <div style="display:flex">
    <div class="field">
      <div class="lbl">
        Οδός
      </div>
      <div>
        <input type="text" id="txtAddress" class="form-control form-control-sm txt"
          value="{{ model.street }}" {% if model.mode == 'view' %} disabled {% endif %} />
      </div>
    </div>

    <div class="field">
      <div class="lbl">
        Αριθμός
      </div>
      <div>
        <input type="text" id="txtAddressNo" class="form-control form-control-sm"
          value="{{ model.streetNo }}" {% if model.mode == 'view' %} disabled {% endif %} />
      </div>
    </div>
  </div>

  <div class="field">
    <div class="lbl">
      Πόλη
    </div>
    <div>
      <input type="text" id="txtCity" class="form-control form-control-sm txt" value="{{ model.city }}"
        {% if model.mode == 'view' %} disabled {% endif %} />
    </div>
  </div>

  <div class="field">
    <div class="lbl">
      Κωδικός
    </div>
    <div>
      <input type="text" id="txtCode" class="form-control form-control-sm txt" value="{{ model.code }}"
        {% if model.mode == 'view' %} disabled {% endif %} />
    </div>
  </div>

  <div class="field">
    <div class="lbl">
      Όνομα χρήστη
    </div>
    <div>
      <input type="text" id="txtUsername" class="form-control form-control-sm txt"
        value="{{ model.username }}" {% if model.mode == 'view' %} disabled {% endif %} />
    </div>
  </div>

  <div class="field" {% if model.username != '' %} hidden {% endif %}>
    <div class="lbl">
      Κωδικός πρόσβασης
    </div>
    <div>
      <input type="password" id="txtPassword" class="form-control form-control-sm txt" />
    </div>
  </div>

  <div class="field" {% if model.username != '' %} hidden {% endif %}>
    <div class="lbl">
      Επιβεβαίωση
    </div>
    <div>
      <input type="password" id="txtConfirm" class="form-control form-control-sm txt" />
    </div>
  </div>
</div>
</body>
</html>

```

Εικόνα 181. Σελίδα html επεξεργασίας χρήστη-ταξιδιωτικού πράκτορα (2)

Τμήμα Μηχανικών Πληροφορικής, ΕΛ.ΜΕ.ΠΑ Κρήτης

```
{% load static %}
<!DOCTYPE html>

<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
    <link href="{% static 'FlightManagementSystem/css/UserAgentSearch.css' %}" rel="stylesheet" />
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>
    <script src="{% static 'FlightManagementSystem/js/UserAgentSearch.js' %}"></script>
  </head>
  <body>
    <div class="modal" id="ppMsg">
      <div class="modal-dialog">
        <div class="modal-content">
          <div class="modal-header">
            ΠΡΟΣΟΧΗ
          </div>
          <div class="modal-body">
            <div id="lblMsg">
            </div>
          </div>
          <div class="modal-footer">
            <div>
              <button type="button" class="btn btn-sm btn-secondary" onclick="onOKClick()">OK</button>
            </div>
          </div>
        </div>
      </div>
    </div>
    <div class="modal" id="ppWarning">
      <div class="modal-dialog">
        <div class="modal-content">
          <div class="modal-header">
            ΕΠΙΒΕΒΑΙΩΣΗ
          </div>
          <div class="modal-body">
            <div>
              Θέλετε σίγουρα να διαγράψετε το ταξιδιωτικό πράκτορα;
            </div>
          </div>
          <div class="modal-footer">
            <div>
              <button type="button" class="btn btn-sm btn-dark" onclick="onYesClick()">Ναι</button>
            </div>
            <div>
              <button type="button" class="btn btn-sm btn-success" onclick="onNoClick()">Όχι</button>
            </div>
          </div>
        </div>
      </div>
    </div>
    <div style="margin-bottom:10px">
      <button type="button" class="btn btn-sm btn-success" onclick="onNewClick()">ΔΗΜΙΟΥΡΓΙΑ</button>
    </div>
    <table class="tbl-search">
      <tr>
        <th>Ενέργεια</th>
        <th>Όνομα Χρήστη</th>
        <th>Αριθμός Ταυτότητας</th>
        <th>Όνομα</th>
        <th>Επίσημο</th>
        <th>Οδός</th>
        <th>Αριθμός</th>
        <th>Πόλη</th>
        <th>Κωδικός</th>
      </tr>
      <tr>
        <th colspan="9">{{ for ag in model %}}
      </th>
      <td>
        <a href="javascript:void(0)" title="Προβολή" data-toggle="tooltip" onclick="viewUserAgent('{{ ag.username }}')">
          <span class="material-icons" style="color:green">visibility</span>
        </a>
        <a href="javascript:void(0)" title="Επεξεργασία" data-toggle="tooltip" onclick="editUserAgent('{{ ag.username }}')">
          <span class="material-icons" style="color:black">create</span>
        </a>
        <a href="javascript:void(0)" onclick="deleteUserAgent('{{ ag.username }}'" title="Διαγραφή" data-toggle="tooltip">
          <span class="material-icons" style="color:red">delete</span>
        </a>
      </td>
        <td class="td-centered">{{ ag.username }}</td>
        <td class="td-centered">{{ ag.id }}</td>
        <td class="td-centered">{{ ag.firstName }}</td>
        <td class="td-centered">{{ ag.lastName }}</td>
        <td class="td-centered">{{ ag.street }}</td>
        <td class="td-centered">{{ ag.streetNo }}</td>
        <td class="td-centered">{{ ag.city }}</td>
        <td class="td-centered">{{ ag.code }}</td>
      </tr>
    </table>
  </body>
</html>
```

Εικόνα 182. Σελίδα html αναζήτησης χρήστη-ταξιδιωτικού πράκτορα

Τμήμα Μηχανικών Πληροφορικής, ΕΛ.ΜΕ.ΠΑ Κρήτης

```
<div style="margin-left:20px">
  <div class="field">
    <div class="lbl">
      Τηλέφωνο
    </div>
    <div>
      <input type="text" id="txtPhoneNo" class="form-control form-control-sm txt" value="{{ model.phoneNo }}"
        {% if model.mode == 'view' %} disabled {% endif %} />
    </div>
  </div>

  <div class="field">
    <div class="lbl">
      Όνομα
    </div>
    <div>
      <input type="text" id="txtFirstName" class="form-control form-control-sm txt" value="{{ model.firstName }}"
        {% if model.mode == 'view' %} disabled {% endif %} />
    </div>
  </div>

  <div class="field">
    <div class="lbl">
      Επώνυμο
    </div>
    <div>
      <input type="text" id="txtLastName" class="form-control form-control-sm txt" value="{{ model.lastName }}"
        {% if model.mode == 'view' %} disabled {% endif %} />
    </div>
  </div>

  <div style="display:flex">
    <div class="field">
      <div class="lbl">
        Οδός
      </div>
      <div>
        <input type="text" id="txtAddress" class="form-control form-control-sm txt" value="{{ model.street }}"
          {% if model.mode == 'view' %} disabled {% endif %} />
      </div>
    </div>

    <div class="field">
      <div class="lbl">
        Αριθμός
      </div>
      <div>
        <input type="text" id="txtAddressNo" class="form-control form-control-sm" value="{{ model.streetNo }}"
          {% if model.mode == 'view' %} disabled {% endif %} />
      </div>
    </div>
  </div>

  <div class="field">
    <div class="lbl">
      Πόλη
    </div>
    <div>
      <input type="text" id="txtCity" class="form-control form-control-sm txt" value="{{ model.city }}"
        {% if model.mode == 'view' %} disabled {% endif %} />
    </div>
  </div>

  <div class="field">
    <div class="lbl">
      Όνομα χρήστη
    </div>
    <div>
      <input type="text" id="txtUsername" class="form-control form-control-sm txt" value="{{ model.username }}"
        {% if model.mode == 'view' %} disabled {% endif %} />
    </div>
  </div>

  <div class="field" {% if model.username != '' %} hidden {% endif %}>
    <div class="lbl">
      Κωδικός πρόσβασης
    </div>
    <div>
      <input type="password" id="txtPassword" class="form-control form-control-sm txt"/>
    </div>
  </div>

  <div class="field" {% if model.username != '' %} hidden {% endif %}>
    <div class="lbl">
      Επιβεβαίωση
    </div>
    <div>
      <input type="password" id="txtConfirm" class="form-control form-control-sm txt"/>
    </div>
  </div>

  <div class="field">
    <div class="lbl">
      E-mail
    </div>
    <div>
      <input type="text" id="txtEmail" class="form-control form-control-sm txt" value="{{ model.eMail }}"
        {% if model.mode == 'view' %} disabled {% endif %} />
    </div>
  </div>
</div>
</body>
</html>
```

Εικόνα 184. Σελίδα html επεξεργασίας χρήστη-πελάτη (2)

Τμήμα Μηχανικών Πληροφορικής, ΕΛ.ΜΕ.ΠΑ Κρήτης

```
{% load static %}
<!DOCTYPE html>

<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
<link href="{% static 'FlightManagementSystem/css/UserClientSearch.css' %}" rel="stylesheet" />
<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>
<script src="{% static 'FlightManagementSystem/js/UserClientSearch.js' %}"></script>

<title>ΧΡΗΣΤΗΣ</title>
</head>

<body>
<div class="modal" id="ppMsg">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
ΠΡΟΣΟΧΗ
</div>
<div class="modal-body">
<div id="lblMsg">
</div>
</div>
<div class="modal-footer">
<div>
<button type="button" class="btn btn-sm btn-secondary" onclick="onOKClick()">OK</button>
</div>
</div>
</div>
</div>
</div>

<div class="modal" id="ppWarning">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
ΕΠΙΒΕΒΑΙΩΣΗ
</div>
<div class="modal-body">
<div>
Θέλετε σίγουρα να διαγράψετε το πελάτη;
</div>
</div>
<div class="modal-footer">
<div>
<button type="button" class="btn btn-sm btn-dark" onclick="onYesClick()">Ναι</button>
</div>
<div>
<button type="button" class="btn btn-sm btn-success" onclick="onNoClick()">Όχι</button>
</div>
</div>
</div>
</div>
</div>

<div style="margin-bottom:10px">
<button type="button" class="btn btn-sm btn-success" onclick="onNewClick()">ΔΗΜΙΟΥΡΓΙΑ</button>
</div>

<table class="tbl-search">
<tr>
<th>Ενέργεια</th>
<th>Όνομα Χρήστη</th>
<th>Αριθμός Τηλεφώνου</th>
<th>Ηλεκτρονική διεύθυνση</th>
<th>Όνομα</th>
<th>Επώνυμο</th>
<th>Οδός</th>
<th>Αριθμός</th>
<th>Πόλη</th>
</tr>
<tr>
{% for uc in model %}
<tr>
<td>
<a href="javascript:void(0)" title="Προβολή" data-toggle="tooltip" onclick="viewUserClient('{{ uc.username }}')">
<span class="material-icons" style="color:green">visibility
</span>
</a>
<a href="javascript:void(0)" title="Επεξεργασία" data-toggle="tooltip" onclick="editUserClient('{{ uc.username }}')">
<span class="material-icons" style="color:black">create
</span>
</a>
<a href="javascript:void(0)" onclick="deleteUserClient('{{ uc.username }}')" title="Διαγραφή" data-toggle="tooltip">
<span class="material-icons" style="color:red">delete
</span>
</a>
</td>
<td class="td-centered">{{ uc.username }}</td>
<td class="td-centered">{{ uc.phoneNo }}</td>
<td class="td-centered">{{ uc.eMail }}</td>
<td class="td-centered">{{ uc.firstName }}</td>
<td class="td-centered">{{ uc.lastName }}</td>
<td class="td-centered">{{ uc.street }}</td>
<td class="td-centered">{{ uc.streetNo }}</td>
<td class="td-centered">{{ uc.city }}</td>
</tr>
{% endfor %}
</table>
</body>
</html>
```

Εικόνα 185. Σελίδα html αναζήτησης χρήστη-πελάτη

```
{% load static %}
<!DOCTYPE html>

<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
    <link href="{% static 'FlightManagementSystem/css/Users.css' %}" rel="stylesheet" />
    <link rel="shortcut icon" href="{% static 'FlightManagementSystem/images/favicon.ico' %}" />

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>
    <script src="{% static 'FlightManagementSystem/js/Users.js' %}"></script>

    <title>ΧΡΗΣΤΕΣ</title>
  </head>

  <body>
    <div class="main-section">
      <div class="left-section">
        <div id="tabUserClients" class="tab" onclick="onClientsClick()">
          ΠΕΛΑΤΕΣ
        </div>

        <div id="tabUserAgents" class="tab" onclick="onAgentsClick()">
          ΤΑΞΙΔΙΩΤΙΚΟΙ ΠΡΑΚΤΟΡΕΣ
        </div>

        <div id="tabUserAdministrators" class="tab" onclick="onAdministratorsClick()">
          ΔΙΑΧΕΙΡΙΣΤΕΣ
        </div>
      </div>

      <div class="right-section">
        <iframe id="frmUsers"></iframe>
      </div>
    </body>
  </html>
```

Εικόνα 186. Σελίδα html χρηστών

```
{% load static %}
<!DOCTYPE html>

<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
    <link href="{% static 'FlightManagementSystem/css/top-banner.css' %}" rel="stylesheet" />
    <link href="{% static 'FlightManagementSystem/css/welcome.css' %}" rel="stylesheet" />
    <link rel="shortcut icon" href="{% static 'FlightManagementSystem/images/favicon.ico' %}" />

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>

    <title>WELCOME</title>
  </head>

  <body style="background-image:url('{% static 'FlightManagementSystem/images/admin.jpg' %}');">

    <div class="main-container">
      <div style="margin-left:170px;background-color:white;border:1px solid black;border-radius:5px; padding:5px;text-shadow:1px 1px 2px black, 0 0 25px blue, 0 0 5px darkblue">
        <h3>ΚΑΛΩΣΟΡΙΣΤΕΣ <span class="user">{{ currentUser.firstName }} {{ currentUser.lastName }}</span></h3>
      </div>
    </div>
  </body>
</html>
```

Εικόνα 187. Σελίδα html καλωσορίσματος χρήστη-διαχειριστή

```
{% load static %}
<!DOCTYPE html>

<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
    <link href="{% static 'FlightManagementSystem/css/top-banner.css' %}" rel="stylesheet" />
    <link href="{% static 'FlightManagementSystem/css/Welcome.css' %}" rel="stylesheet" />
    <link rel="shortcut icon" href="{% static 'FlightManagementSystem/images/favicon.ico' %}" />

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>

    <title>WELCOME</title>
  </head>

  <body style="background-image:url({% static 'FlightManagementSystem/images/agent.jpg' %});">

    <div class="main-container">
      <div style="margin-left:170px;margin-top:-300px">
        <h3>ΚΑΛΩΣΟΡΙΣΕΣ <span class="user">{{ currentUser.firstName }} {{ currentUser.lastName }}</span></h3>
      </div>
    </div>
  </body>
</html>
```

Εικόνα 188. Σελίδα html καλωσορίσματος χρήστη-ταξιδιωτικού πράκτορα

```
{% load static %}
<!DOCTYPE html>

<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
    <link href="{% static 'FlightManagementSystem/css/top-banner.css' %}" rel="stylesheet" />
    <link href="{% static 'FlightManagementSystem/css/Welcome.css' %}" rel="stylesheet" />
    <link rel="shortcut icon" href="{% static 'FlightManagementSystem/images/favicon.ico' %}" />

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>

    <title>WELCOME</title>
  </head>

  <body style="background-image:url({% static 'FlightManagementSystem/images/client.jpg' %});">

    <div class="main-container">
      <div style="margin-left:170px">
        <h3>ΚΑΛΩΣΟΡΙΣΕΣ <span class="user">{{ currentUser.firstName }} {{ currentUser.lastName }}</span></h3>
      </div>
    </div>
  </body>
</html>
```

Εικόνα 189. Σελίδα html καλωσορίσματος χρήστη-πελάτη

Δημιουργία σελίδων JavaScript

```

let openInView = false;
let closeWindow = false;

function onOKClick() {
    $('#ppMsg').modal('hide');

    if (openInView == true) {
        window.opener.updateGrid();
        let id = $('#txtID').val();
        window.open("/agentEdit/" + id + "/view", "_self");
    }
    else if (closeWindow == true) {
        try {
            window.opener.updateGrid();
            window.opener.closeWindow();
        }
        catch (ex) { window.close(); }
    }
}

function onDeleteClick() {
    $('#ppWarning').modal('show');
}

function onYesClick(){
    let id = $('#txtID').val();

    $.post('/deleteAgent/' + id, {}, function (data, status) {
        $('#ppWarning').modal('hide');
        $('#lblMsg').text('Ο ταξιδιωτικός πράκτορας διαγράφηκε επιτυχώς.');
```

Εικόνα 190. Σελίδα javascript επεξεργασίας ταξιδιωτικού πράκτορα (1)

```

$.ajax({
  type: 'POST',
  url: '/validateAgent/' + id + '/' + code + '/' + obj.id + '/' + obj.code,
  data: {},
  dataType: 'json',
  async: false,
  success: function (data, status, xhr) {
    let msg = '';

    if (data.idExists == true) {
      msg = '<li>0 αριθμός ταυτότητας υπάρχει ήδη.</li>';
    }

    if (data.codeExists == true) {
      msg += '<li>0 κωδικός υπάρχει ήδη.</li>';
    }

    if (msg != '') {
      $('#lblMsg').html('<ul>' + msg + '</ul>');
      $('#ppMsg').modal('show');

      isValid = false;
    }
    else {
      isValid = true;
    }
  }
});

return isValid;
}

function onSaveClick(close) {
  let obj = $.parseJSON($('#hf').val().replace(/\'/g, '\"'));
  closeWindow = close;
  let isValid = validate(obj);

  if (isValid == true) {
    let id = $('#txtID').val();
    let previousID = obj.id;
    let firstName = $('#txtFirstName').val();
    let lastName = $('#txtLastName').val();
    let street = $('#txtStreet').val();
    let streetNo = $('#txtStreetNo').val();
    let city = $('#txtCity').val();
    let code = $('#txtCode').val();
    let previousCode = obj.code;

    agent = new Agent(previousID, id, firstName, lastName, street, streetNo, city, code, previousCode);

    $.ajax({
      type: 'POST',
      url: '/saveAgent',
      data: agent,
      dataType: 'json',
      async: false,
      success: function (data, status, xhr) {
        $('#lblMsg').html('0 πράκτορας καταχωρήθηκε επιτυχώς. ');
        $('#ppMsg').modal('show');
        openInView = !close;
      }
    });
  }
}

function onEditClick() {
  let id = $('#txtID').val();
  window.open("/agentEdit/" + id + "/edit", "_self");
}

class Agent {
  constructor(previousID, id, firstName, lastName, street, streetNo, city, code, previousCode) {
    this.previousID = previousID;
    this.id = id;
    this.firstName = firstName;
    this.lastName = lastName;
    this.street = street;
    this.streetNo = streetNo;
    this.city = city;
    this.code = code;
    this.previousCode = previousCode;
  }
}

function onlogoutClick() {
  window.open('/logout', '_self');
}

function onhomeClick() {
  window.open('/home', '_self');
}

```

Εικόνα 191. Σελίδα javascript επεξεργασίας ταξιδιωτικού πράκτορα (2)

```

let myWindow = null;
let selectedID = null;

$(document).ready(function () {
  $('[data-toggle="tooltip"]').tooltip();
});

function editAgent(id) {
  myWindow = window.open('/agentEdit/' + id + '/edit');
}

function viewAgent(id) {
  myWindow = window.open('/agentEdit/' + id + '/view');
}

function onNewClick(){
  myWindow = window.open("/agentEdit/0/edit");
}

function closeWindow(){
  myWindow.close();
  window.open('/agentSearch', '_self');
}

function updateGrid(){
  window.open('/agentSearch', '_self');
}

function onYesClick(){
  $.post('/deleteAgent/' + selectedID, {}, function (data, status) {
    $('#ppWarning').modal('hide');
    $('#lblMsg').text('Ο ταξιδιωτικός πράκτορας διαγράφηκε επιτυχώς. ');
    $('#ppMsg').modal('show');
  });
}

function onNoClick(){
  $('#ppWarning').modal('hide');
}

function deleteAgent(ID)
{
  selectedID = ID;
  $('#ppWarning').modal('show');
}

function onOKClick(){
  updateGrid();
}

```

Εικόνα 192. Σελίδα javascript αναζήτησης ταξιδιωτικού πράκτορα

```

let openInView = false;
let closeWindow = false;

function onOKClick(){
    $('#ppMsg').modal('hide');

    if (openInView == true){
        window.opener.updateGrid();
        let id = $('#txtID').val();
        window.open("/airHostessEdit/" + id + "/view", "_self");
    }
    else if (closeWindow == true) {
        try {
            window.opener.updateGrid();
            window.opener.closeWindow();
        }
        catch (ex) { window.close(); }
    }
}

function onDeleteClick(){
    $('#ppWarning').modal('show');
}

function onYesClick(){
    let id = $('#txtID').val();

    $.post('/deleteAirHostess/' + id, {}, function(data, status){
        $('#ppWarning').modal('hide');
        if(data=='true'){
            $('#lblMsg').text('Η αεροσυνοδός διαγράφηκε επιτυχώς.');
```

Εικόνα 193. Σελίδα javascript επεξεργασίας αεροσυνοδού (1)


```

$.ajax({
  type: 'POST',
  url: '/validateAirHostess/' + id + '/' + prevID,
  data: {},
  dataType: 'json',
  async: false,
  success: function (data, status, xhr) {
    let msg = '';

    if (data.idExists == true) {
      msg = '<li>0 αριθμός ταυτότητας υπάρχει ήδη.</li>';
    }

    if (msg != '') {
      $('#lblMsg').html('<ul>' + msg + '</ul>');
      $('#ppMsg').modal('show');

      isValid = false;
    }
    else {
      isValid = true;
    }
  }
});

return isValid;
}

function onSaveClick(close){
  let prevID = $('#hf').val();

  if (prevID === '')
  {
    prevID = '0';
  }

  closeButton = close;
  let isValid = validate(prevID);

  if (isValid == true){
    let previousID = prevID;
    let id = $('#txtID').val();
    let fn = $('#txtFirstName').val();
    let ln = $('#txtLastName').val();
    let str = $('#txtStreet').val();
    let strNo = $('#txtStreetNo').val();
    let city = $('#txtCity').val();
    let phoneNo = $('#txtPhoneNo').val();
    let birthDate = $('#txtBirthDate').val();
    let years = $('#txtYears').val();

    hostess = new AirHostess(previousID, id, fn, ln, str, strNo, city, phoneNo, birthDate, years);

    $.ajax({
      type: 'POST',
      url: '/saveAirHostess',
      data: hostess,
      dataType: 'json',
      async: false,
      success: function (data, status, xhr){
        $('#lblMsg').html('Η αεροσυνοδός καταχωρήθηκε επιτυχώς.');
```

Εικόνα 194. Σελίδα javascript επεξεργασίας αεροσυνοδού (2)

```

let myWindow = null;
let selectedID = null;

$(document).ready(function () {
    $('[data-toggle="tooltip"]').tooltip();
});

function editAirHostess(id) {
    myWindow = window.open('/airHostessEdit/' + id + '/edit');
}

function viewAirHostess(id) {
    myWindow = window.open('/airHostessEdit/' + id + '/view');
}

function onNewClick(){
    myWindow = window.open("/airHostessEdit/0/edit");
}

function closeWindow(){
    myWindow.close();
    window.open('/airHostessSearch', '_self');
}

function updateGrid(){
    window.open('/airHostessSearch', '_self');
}

function onYesClick(){
    $.post('/deleteAirHostess/' + selectedID, {}, function (data, status) {
        $('#ppWarning').modal('hide');
        $('#lblMsg').text('Η αεροσυνοδός διαγράφηκε επιτυχώς.');
```

Εικόνα 195. Σελίδα javascript αναζήτησης αεροσυνοδού

```

let openInView = false;
let closeWindow = false;

function onOKClick(){
    $('#ppMsg').modal('hide');

    if (openInView == true){
        window.opener.updateGrid();
        let name = $('#txtName').val();
        window.open("/airplaneEdit/" + name + "/view", "_self");
    }
    else if (closeWindow == true) {
        try {
            window.opener.updateGrid();
            window.opener.closeWindow();
        }
        catch (ex) { window.close(); }
    }
}

function onDeleteClick(){
    $('#ppWarning').modal('show');
}

function onYesClick(){
    let name = $('#txtName').val();

    $.post('/deleteAirplane/' + name, {}, function (data, status) {
        $('#ppWarning').modal('hide');
        if (data == 'true')
            {
                $('#lblMsg').text('Το αεροπλάνο διαγράφηκε επιτυχώς. ');
                closeWindow = true;
            }
        else{
            $('#lblMsg').text('Δεν είναι δυνατή η διαγραφή του αεροπλάνου, επειδή συσχετίζεται με άλλες εγγραφές. ');
        }
        $('#ppMsg').modal('show');
    });
}

function onNoClick(){
    $('#ppWarning').modal('hide');
}

function onCancelClick(){
    window.location.reload();
}

function validate(prevName){
    let isValid = null;
    let message = '';

    let name = $('#txtName').val();
    if (name == ''){
        message = message + '<li>Συμπληρώστε την ονομασία.</li>';
    }

    let busSeats = $('#txtNumBussinessSeats').val();
    if (busSeats == '0'){
        message = message + '<li>Συμπληρώστε τον αριθμό διακεκριμένων θέσεων.</li>';
    }

    let ecoSeats = $('#txtNumEconomySeats').val();
    if (ecoSeats == '0'){
        message = message + '<li>Συμπληρώστε τον αριθμό οικονομικών θέσεων.</li>';
    }

    if (message != ''){
        message = '<ul>' + message + '</ul>';
        $('#lblMsg').html(message);
        $('#ppMsg').modal('show');

        isValid = false;
    }
    else {
        if (prevName === '')
            {
                prevName = '0';
            }
    }
}

```

Εικόνα 196. Σελίδα javascript επεξεργασίας αεροπλάνου (1)

```

$.ajax({
  type: 'POST',
  url: '/validateAirplane/' + name + '/' + prevName,
  data: {},
  dataType: 'json',
  async: false,
  success: function (data, status, xhr) {
    let msg = '';

    if (data.nameExists == true) {
      msg += '<li>Το αεροπλάνο υπάρχει ήδη.</li>';
    }

    if (msg != '') {
      $('#lblMsg').html('<ul>' + msg + '</ul>');
      $("#ppMsg").modal('show');

      isValid = false;
    }
    else {
      isValid = true;
    }
  }
});
}

return isValid;
}

function onSaveClick(close){
  let prevName = $('#hf').val();
  closeWindow = close;
  let isValid = validate(prevName);

  if (isValid == true){
    let previousName = prevName;
    let name = $('#txtName').val();
    let busSeats = $('#txtNumBussinessSeats').val();
    let ecoSeats = $('#txtNumEconomySeats').val();

    plane = new Airplane(previousName, name, busSeats, ecoSeats);

    $.ajax({
      type: 'POST',
      url: '/saveAirplane',
      data: plane,
      dataType: 'json',
      async: false,
      success: function(data,status,xhr){
        $('#lblMsg').html('Το αεροπλάνο καταχωρήθηκε επιτυχώς.');
```

Εικόνα 197. Σελίδα javascript επεξεργασίας αεροπλάνου (2)

```
let myWindow = null;
let selectedName = null;

$(document).ready(function () {
  $('[data-toggle="tooltip"]').tooltip();
});

function editAirplane(name) {
  myWindow = window.open('/airplaneEdit/' + name + '/edit');
}

function viewAirplane(name) {
  myWindow = window.open('/airplaneEdit/' + name + '/view');
}

function onNewClick(){
  myWindow = window.open("/airplaneEdit/0/edit");
}

function closeWindow(){
  myWindow.close();
  window.open('/airplaneSearch', '_self');
}

function updateGrid(){
  window.open('/airplaneSearch', '_self');
}

function onYesClick(){
  $.post('/deleteAirplane/' + selectedName, {}, function (data, status) {
    $('#ppWarning').modal('hide');
    $('#lblMsg').text('Το αεροπλάνο διαγράφηκε επιτυχώς.');
```

```
    $('#ppMsg').modal('show');
  });
}

function onNoClick(){
  $('#ppWarning').modal('hide');
}

function deleteAirplane(Name)
{
  selectedName = Name;
  $('#ppWarning').modal('show');
}

function onOKClick(){
  updateGrid();
}
```

Εικόνα 198. Σελίδα javascript αναζήτησης αεροπλάνου

```

let openInView = false;
let closeWindow = false;

function onOKClick(){
    $('#ppMsg').modal('hide');

    if (openInView == true){
        window.opener.updateGrid();
        let phoneNo = $('#txtPhoneNo').val();
        window.open("/clientEdit/" + phoneNo + "/view", "_self");
    }
    else if (closeWindow == true) {
        try {
            window.opener.updateGrid();
            window.opener.closeWindow();
        }
        catch (ex) { window.close(); }
    }
}

function onDeleteClick(){
    $('#ppWarning').modal('show');
}

function onYesClick(){
    let phoneNo = $('#txtPhoneNo').val();

    $.post('/deleteClient/' + phoneNo, {}, function (data, status) {
        $('#ppWarning').modal('hide');
        $('#lblMsg').text('Ο πελάτης διαγράφηκε επιτυχώς. ');
        $('#ppMsg').modal('show');
        closeWindow = true;
    });
}

function onNoClick(){
    $('#ppWarning').modal('hide');
}

function onCancelClick(){
    window.location.reload();
}

function validate(prevPhoneNo){
    let isValid = null;
    let message = '';

    let phoneNo = $('#txtPhoneNo').val();
    if (phoneNo == ''){
        message = message + '<li>Συμπληρώστε τον αριθμό τηλεφώνου.</li>';
    }

    let fName = $('#txtFirstName').val();
    if (fName == ''){
        message = message + '<li>Συμπληρώστε το όνομα.</li>';
    }

    let lName = $('#txtLastName').val();
    if (lName == ''){
        message = message + '<li>Συμπληρώστε το επώνυμο.</li>';
    }

    let street = $('#txtAddress').val();
    if (street == ''){
        message = message + '<li>Συμπληρώστε την οδό.</li>';
    }

    let streetNo = $('#txtAddressNo').val();
    if (streetNo == ''){
        message = message + '<li>Συμπληρώστε τον αριθμό της οδού.</li>';
    }

    let city = $('#txtCity').val();
    if (city == ''){
        message = message + '<li>Συμπληρώστε το όνομα της πόλης.</li>';
    }

    let email = $('#txtEmail').val();
    if (email == ''){
        message = message + '<li>Συμπληρώστε την ηλεκτρονική διεύθυνση.</li>';
    }

    if (message != ''){
        message = '<ul>' + message + '</ul>';
        $('#lblMsg').html(message);
        $('#ppMsg').modal('show');
    }

    isValid = false;
}
else {
    if (prevPhoneNo === '')
    {
        prevPhoneNo = '0';
    }
}
}

```

Εικόνα 199. Σελίδα javascript επεξεργασίας πελάτη (1)

```

$.ajax({
  type: 'POST',
  url: '/validateClient/' + phoneNo + '/' + prevPhoneNo,
  data: {},
  dataType: 'json',
  async: false,
  success: function (data, status, xhr) {
    let msg = '';

    if (data.phoneNoExists == true) {
      msg = '<li>Ο αριθμός τηλεφώνου υπάρχει ήδη.</li>';
    }

    if (msg != '') {
      $('#lblMsg').html('<ul>' + msg + '</ul>');
      $('#ppMsg').modal('show');

      isValid = false;
    }
    else {
      isValid = true;
    }
  }
});

return isValid;
}

function onSaveClick(close){
  let prevPhoneNo = $('#hf').val();

  if (prevPhoneNo === '')
  {
    prevPhoneNo = '0';
  }

  closeButton = close;
  let isValid = validate(prevPhoneNo);

  if (isValid == true){
    let previousPhoneNo = prevPhoneNo;
    let phoneNo = $('#txtPhoneNo').val();
    let eMail = $('#txtEmail').val();
    let fName = $('#txtFirstName').val();
    let lName = $('#txtLastName').val();
    let street = $('#txtAddress').val();
    let streetNo = $('#txtAddressNo').val();
    let city = $('#txtCity').val();

    client = new Client(previousPhoneNo, phoneNo, eMail, fName, lName, street, streetNo, city)

    $.ajax({
      type: 'POST',
      url: '/saveClient',
      data: client,
      dataType: 'json',
      async: false,
      success: function(data,status,xhr){
        $('#lblMsg').html('Ο πελάτης καταχωρήθηκε επιτυχώς. ');
        $('#ppMsg').modal('show');
        openInView = !close;
      }
    });
  }
}

function onEditClick(){
  let phoneNo = $('#txtPhoneNo').val();
  window.open("/clientEdit/" + phoneNo + "/edit", "_self");
}

class Client{
  constructor(previousPhoneNo,phoneNo,eMail,firstName,lastName,street,streetNo,city){
    this.previousPhoneNo = previousPhoneNo;
    this.phoneNo = phoneNo;
    this.eMail = eMail;
    this.firstName = firstName;
    this.lastName = lastName;
    this.street = street;
    this.streetNo = streetNo;
    this.city = city;
  }
}

function onlogoutClick(){
  window.open('/logout','_self');
}

function onhomeClick(){
  window.open('/home','_self');
}

```

Εικόνα 200. Σελίδα javascript επεξεργασίας πελάτη (2)

```

let myWindow = null;
let selectedPhoneNo = null;

$(document).ready(function () {
  $('[data-toggle="tooltip"]').tooltip();
});

function editClient(phoneNo) {
  myWindow = window.open('/clientEdit/' + phoneNo + '/edit');
}

function viewClient(phoneNo) {
  myWindow = window.open('/clientEdit/' + phoneNo + '/view');
}

function onNewClick(){
  myWindow = window.open("/clientEdit/0/edit");
}

function closeWindow(){
  myWindow.close();
  window.open('/clientSearch', '_self');
}

function updateGrid(){
  window.open('/clientSearch', '_self');
}

function onYesClick(){
  $.post('/deleteClient/' + selectedPhoneNo, {}, function (data, status) {
    $('#ppWarning').modal('hide');
    $('#lblMsg').text('Ο πελάτης διαγράφηκε επιτυχώς.');
```

Εικόνα 201. Σελίδα javascript αναζήτησης πελάτη


```

let openInView = false;
let closeWindow = false;
let selectedTransport = null;
let currentReservationID = null;
let openTicket = false;
let ticketID = 0;

$(document).ready(onReady);

function onReady() {
    currentReservationID = $('#hf').val();

    findRoute();

    let isBusinessClass = null;

    if ($('#hf2').val() == 'False') {
        isBusinessClass = false;
    }
    else {
        isBusinessClass = true;
    }

    let transportID1 = $('#hfT1').val();
    let transportID2 = $('#hfT2').val();

    let id = 'rdb_' + transportID1 + '_' + transportID2 + '_';
    selectedTransport = transportID1 + '_' + transportID2;

    if (isBusinessClass) {
        id = id + 'bus';
    }
    else {
        id = id + 'eco';
    }

    $('#'+ id).attr('checked', true);
}

function onOKClick() {
    $('#ppMsg').modal('hide');

    if (openTicket == true) {
        window.opener.updateGrid();
        window.open("/showTicket/" + ticketID, "_self");
    }
    else if (openInView == true) {
        window.opener.updateGrid();
        window.open("/clientReservationEdit/" + currentReservationID + "/view", "_self");
    }
    else if (closeWindow == true) {
        try {
            window.opener.updateGrid();
            window.opener.closeWindow();
        }
        catch (ex) {
            window.close();
        }
    }
}

function onDeleteClick() {
    $('#ppWarning').modal('show');
}

function onYesClick() {
    let id = $('#txtID').val();

    $.post('/deleteReservation/' + id, {}, function (data, status) {
        $('#ppWarning').modal('hide');
        $('#tblMsg').text('Η κράτηση διαγράφηκε επιτυχώς.');
```

Εικόνα 202. Σελίδα javascript επεξεργασίας κράτησης πελάτη (1)

Τμήμα Μηχανικών Πληροφορικής, ΕΛ.ΜΕ.ΠΑ Κρήτης

```
function createNoTransportPanel() {
    let panel = `<div style="text-align:center">Δεν βρέθηκαν μεταφορές.</div>`;
    return panel;
}

function createDirectTransportPanel(transportItem) {
    let value = transportItem.transportID;
    let panel = `<div class="panel">`;
    let panel1 = `<table class="tbl"><tr><td class="time bg-gray" style="border-top-left-radius:5px">` + transportItem.depTime + `</td><td class="bg-gray"></td><td class="time bg-gray">` + transportItem.arvTime + `</td><td class="category">Economy</td><td class="category"
style="border-top-right-radius:5px">Business</td>`;
    panel1 += `<tr><td class="city bg-gray">` + transportItem.depCity + `</td><td class="bg-gray"></td><td class="city bg-gray">` + transportItem.arvCity + `</td><td rowspan="2"
class="price">€euro;` + transportItem.ecoPrice + `</td><td rowspan="2" class="price">€euro;` + transportItem.busPrice + `</td></tr>`;
    let panel2 = `<tr><td class="time bg-gray" style="padding-top:10px"></td><td class="bg-gray" style="padding-top:10px"></td><td class="time bg-gray" style="padding-top:10px"></td>
</tr>`;
    panel2 += `<tr><td class="city bg-gray" style="border-bottom-left-radius:5px"></td><td class="bg-gray"></td><td class="city bg-gray"></td><td class="radio"><input ` +
(transportItem.ecoSeats == 0 ? `disabled` : ``) + ` id="rdb_` + value + `_eco` type="radio" name="selection" onclick="setSelectedOption(` + value + `_eco`)"` + `</td><td class="radio"
style="border-bottom-right-radius:5px"><input ` + (transportItem.busSeats == 0 ? `disabled` : ``) + ` type="radio" id="rdb_` + value + `_bus` name="selection"
onclick="setSelectedOption(` + value + `_bus`)"` + `</td></tr>`;
    panel += panel1 + panel2 + `</div>`;
    return panel;
}

function createIndirectTransportPanel(transportItem1, transportItem2) {
    let value = transportItem1.transportID + `.` + transportItem2.transportID;
    let panel = `<div class="panel">`;
    let panel1 = `<table class="tbl"><tr><td class="time bg-gray" style="border-top-left-radius:5px">` + transportItem1.depTime + `</td><td class="bg-gray"></td><td class="time bg-gray">` + transportItem1.arvTime + `</td><td class="category">Economy</td><td class="category"
style="border-top-right-radius:5px">Business</td>`;
    panel1 += `<tr><td class="city bg-gray">` + transportItem1.depCity + `</td><td class="bg-gray"></td><td class="city bg-gray">` + transportItem1.arvCity + `</td><td rowspan="2"
class="price">€euro;` + (transportItem1.ecoPrice + transportItem2.ecoPrice) + `</td><td rowspan="2" class="price">€euro;` + (transportItem1.busPrice + transportItem2.busPrice) + `</td>
</tr>`;
    let panel2 = `<tr><td class="time bg-gray" style="padding-top:10px">` + transportItem2.depTime + `</td><td class="bg-gray" style="padding-top:10px"></td><td class="time bg-gray" style="padding-top:10px">` + transportItem2.arvTime + `</td></tr>`;
    panel2 += `<tr><td class="city bg-gray" style="border-bottom-left-radius:5px">` + transportItem2.depCity + `</td><td class="bg-gray"></td><td class="city bg-gray">` +
transportItem2.arvCity + `</td><td class="radio"><input type="radio" id="rdb_` + transportItem1.transportID + `.` + transportItem2.transportID + `_eco` name="selection"
onclick="setSelectedOption(` + value + `_eco`)"` + `</td><td class="radio" style="border-bottom-right-radius:5px"><input type="radio" id="rdb_` + transportItem1.transportID + `.` +
transportItem2.transportID + `_bus` name="selection" onclick="setSelectedOption(` + value + `_bus`)"` + `</td></tr>`;
    panel += panel1 + panel2 + `</div>`;
    return panel;
}

function setSelectedOption(value) {
    selectedTransport = value;
}

function onPrintClick() {
    $.ajax({
        type: 'POST',
        url: '/printTicket',
        data: {
            reservationID: currentReservationID
        },
        success: function (data, status) {
            ticketID = data;
            openTicket = true;
            $('#tblMsg').html(`<div>Η έκδοση του εισιτηρίου έγινε επιτυχώς.<br /></div> αριθμός εισιτηρίου σας είναι: ` + data + `<br />Καλό σας ταξίδι!`);
            $('#ppMsg').modal('show');
        },
        dataType: 'json',
        async: false
    });
}

class ClientReservation {
    constructor(reservationID, phoneNo, resDate, expDate, busSeat, depCityID, arvCityID, transportID1, transportID2) {
        this.reservationID = reservationID;
        this.phoneNo = phoneNo;
        this.resDate = resDate;
        this.expDate = expDate;
        this.busSeat = busSeat;
        this.depCityID = depCityID;
        this.arvCityID = arvCityID;
        this.transportID1 = transportID1;
        this.transportID2 = transportID2;
    }
}

function onLogoutClick() {
    window.open('/logout', '_self');
}

function onHomeClick() {
    window.open('/home', '_self');
}
```

Εικόνα 204. Σελίδα javascript επεξεργασίας κράτησης πελάτη (3)

```

let myWindow = null;
let selectedID = null;

$(document).ready(function () {
    $('[data-toggle="tooltip"]').tooltip();
});

function editReservation(reservationID) {
    myWindow = window.open('/clientReservationEdit/' + reservationID + '/edit');
}

function viewReservation(reservationID) {
    myWindow = window.open('/clientReservationEdit/' + reservationID + '/view');
}

function onNewClick(){
    myWindow = window.open("/clientReservationEdit/0/edit");
}

function closeWindow(){
    myWindow.close();
    window.open('/clientReservationSearch', '_self');
}

function updateGrid(){
    window.open('/clientReservationSearch', '_self');
}

function onYesClick(){
    $.post('/deleteReservation/' + selectedID, {}, function (data, status) {
        $('#ppWarning').modal('hide');
        $('#lblMsg').text('Η κράτηση διαγράφηκε επιτυχώς.');
```

Εικόνα 205. Σελίδα javascript αναζήτησης κράτησης πελάτη

```

function onPilotsClick()
{
    selectTab('tabPilots');
    $('#frm').attr('src', './pilotSearch');
}

function onAirhostessesClick()
{
    selectTab('tabAirhostesses');
    $('#frm').attr('src', './airHostessSearch');
}

function onAirplanesClick()
{
    selectTab('tabAirplanes');
    $('#frm').attr('src', './airplaneSearch');
}

function onAgentsClick()
{
    selectTab('tabAgents');
    $('#frm').attr('src', './agentSearch');
}

function onManagersClick()
{
    selectTab('tabManagers');
    $('#frm').attr('src', './managerSearch');
}

function onClientsClick()
{
    selectTab('tabClients');
    $('#frm').attr('src', './clientSearch');
}

function onUsersClick()
{
    selectTab('tabUsers');
    $('#frm').attr('src', './users');
}

function onRoutesClick()
{
    selectTab('tabRoutes');
    $('#frm').attr('src', './routeSearch');
}

function onTransportsClick()
{
    selectTab('tabTransports');
    $('#frm').attr('src', './transportSearch');
}

function onReservationsClick()
{
    selectTab('tabReservations');
    $('#frm').attr('src', './reservationSearch');
}

function onClientReservationsClick()
{
    selectTab('tabReservations');
    $('#frm').attr('src', './clientReservationSearch');
}

```

Εικόνα 206. Σελίδα javascript αρχικής σελίδας (1)

```

function deselectAllTabs(){
    $('#tabPilots').css('background-color', '#8fa3cc');
    $('#tabAirhostesses').css('background-color', '#8fa3cc');
    $('#tabAirplanes').css('background-color', '#8fa3cc');
    $('#tabAgents').css('background-color', '#8fa3cc');
    $('#tabManagers').css('background-color', '#8fa3cc');
    $('#tabClients').css('background-color', '#8fa3cc');
    $('#tabUsers').css('background-color', '#8fa3cc');
    $('#tabRoutes').css('background-color', '#8fa3cc');
    $('#tabTransports').css('background-color', '#8fa3cc');
    $('#tabReservations').css('background-color', '#8fa3cc');
    $('#tabDashboard').css('background-color', '#8fa3cc');
    $('#tabReport').css('background-color', '#8fa3cc');
}

function selectTab(tabID)
{
    deselectAllTabs();
    $('#' + tabID).css('background-color', '#6e7d9c');
}

function onlogoutClick(){
    window.open('/logout', '_self');
}

function onhomeClick(){
    window.open('/home', '_self');
}

function onClientDashboardClick(){
    selectTab('tabDashboard');
    $('#frm').attr('src', './clientDashboard');
}

function onReportClick(){
    selectTab('tabReport');
    $('#frm').attr('src', './report');
}

$(document).ready(onReady);

function onReady()
{
    $('#frm').attr('src', './welcome');
}

```

Εικόνα 207. Σελίδα javascript αρχικής σελίδας (2)

```

function onOKClick() {
    $('#ppMsg').modal('hide');
}

function onSignUpClick() {
    window.open('/userClientEdit/0/edit', '_blank');
}

class User {
    constructor(username, password) {
        this.username = username;
        this.password = password;
    }

    isValid() {
        let message = '';

        if (this.username == '') {
            message = '<li>Συμπληρώστε το όνομα χρήστη.</li>';
        }

        if (this.password == '') {
            message = message + '<li>Συμπληρώστε τον κωδικό πρόσβασης.</li>';
        }

        if (message != '') {
            message = '<ul>' + message + '</ul>';
            $('#lblMsg').html(message);
            $('#ppMsg').modal('show');

            return false;
        }
        else {
            return true;
        }
    }
}

function onSignInClick() {
    let username = $('#txtUserName').val();
    let password = $('#txtPassword').val();

    let usr = new User(username, password);

    if (usr.isValid()) {
        $.ajax({
            type: 'POST',
            url: '/signIn',
            data: usr,
            dataType: 'json',
            async: false,
            success: function (data, status, xhr) {
                let valid = $.parseJSON(data);

                if (valid == false) {
                    $('#lblMsg').html('Τα στοιχεία εισόδου είναι λανθασμένα.<br />Παρακαλώ, προσπαθήστε ξανά.');
```

Εικόνα 208. Σελίδα javascript εισόδου χρήστη

```

let openInView = false;
let closeWindow = false;

function onOKClick(){
    $('#ppMsg').modal('hide');

    if (openInView == true){
        window.opener.updateGrid();
        let id = $('#txtID').val();
        window.open("/managerEdit/" + id + "/view", "_self");
    }
    else if (closeWindow == true) {
        try {
            window.opener.updateGrid();
            window.opener.closeWindow();
        }
        catch (ex) { window.close(); }
    }
}

function onDeleteClick(){
    $('#ppWarning').modal('show');
}

function onYesClick(){
    let id = $('#txtID').val();

    $.post('/deleteManager/' + id, {}, function(data, status){
        $('#ppWarning').modal('hide');
        if(data=="true"){
            $('#lblMsg').text('Ο διοικητικός υπάλληλος διαγράφηκε επιτυχώς.');

```

Εικόνα 209. Σελίδα javascript επεξεργασίας διοικητικού υπαλλήλου (1)


```

if (message != ''){
    message = '<ul>' + message + '</ul>';
    $('#lblMsg').html(message);
    $('#ppMsg').modal('show');

    isValid = false;
}
else {
    if (previousID === '')
    {
        previousID = '0';
    }

    $.ajax({
        type: 'POST',
        url: '/validateManager/' + id + '/' + previousID,
        data: {},
        dataType: 'json',
        async: false,
        success: function (data, status, xhr) {
            let msg = '';

            if (data.idExists == true) {
                msg = '<li>0 αριθμός ταυτότητας υπάρχει ήδη.</li>';
            }

            if (msg != '') {
                $('#lblMsg').html('<ul>' + msg + '</ul>');
                $('#ppMsg').modal('show');

                isValid = false;
            }
            else {
                isValid = true;
            }
        }
    });
}

return isValid;
}

function onSaveClick(close){
    let prevID = $('#hf').val();
    closeWindow = close;
    let isValid = validate(prevID);

    if (isValid == true){
        let previousID = prevID;
        let id = $('#txtID').val();
        let fn = $('#txtFirstName').val();
        let ln = $('#txtLastName').val();
        let str = $('#txtStreet').val();
        let strNo = $('#txtStreetNo').val();
        let city = $('#txtCity').val();
        let phoneNo = $('#txtPhoneNo').val();
        let birthDate = $('#txtBirthDate').val();
        let education = $('#txtEducation').val();
        let years = $('#txtYears').val();

        manager = new Manager(previousID, id, fn, ln, str, strNo, city, phoneNo, birthDate, education, years);

        $.ajax({
            type: 'POST',
            url: '/saveManager',
            data: manager,
            dataType: 'json',
            async: false,
            success: function(data,status,xhr){
                $('#lblMsg').html('0 διοικητικός υπάλληλος καταχωρήθηκε επιτυχώς. ');
                $('#ppMsg').modal('show');
                openInView = !close;
            }
        });
    }
}

function onEditClick(){
    let id = $('#txtID').val();
    window.open("/managerEdit/" + id + "/edit", "_self");
}

class Manager{
    constructor(previousID,id,firstName,lastName,street,streetNo,city,phoneNo,birthDate,education,years){
        this.previousID = previousID;
        this.id = id;
        this.firstName = firstName;
        this.lastName = lastName;
        this.street = street;
        this.streetNo = streetNo;
        this.city = city;
        this.phoneNo = phoneNo;
        this.birthDate = birthDate;
        this.education = education;
        this.years = years;
    }
}

function onlogoutClick(){
    window.open('/logout', '_self');
}

function onhomeClick(){
    window.open('/home', '_self');
}

```

Εικόνα 210. Σελίδα javascript επεξεργασίας διοικητικού υπαλλήλου (2)

```

let myWindow = null;
let selectedID = null;

$(document).ready(function () {
    $('[data-toggle="tooltip"]').tooltip();
});

function editManager(id) {
    myWindow = window.open('/managerEdit/' + id + '/edit');
}

function viewManager(id) {
    myWindow = window.open('/managerEdit/' + id + '/view');
}

function onNewClick(){
    myWindow = window.open("/managerEdit/0/edit");
}

function closeWindow(){
    myWindow.close();
    window.open('/managerSearch', '_self');
}

function updateGrid(){
    window.open('/managerSearch', '_self');
}

function onYesClick(){
    $.post('/deleteManager/' + selectedID, {}, function (data, status) {
        $('#ppWarning').modal('hide');
        $('#lblMsg').text('Ο διοικητικός υπάλληλος διαγράφηκε επιτυχώς. ');
        $('#ppMsg').modal('show');
    });
}

function onNoClick(){
    $('#ppWarning').modal('hide');
}

function deleteManager(ID)
{
    selectedID = ID;
    $('#ppWarning').modal('show');
}

function onOKClick(){
    updateGrid();
}

```

Εικόνα 211. Σελίδα javascript αναζήτησης διοικητικού υπαλλήλου

```

let openInView = false;
let closeWindow = false;

function onOKClick() {
    $('#ppMsg').modal('hide');

    if (openInView == true) {
        window.opener.updateGrid();
        let id = $('#txtID').val();
        window.open("/pilotEdit/" + id + "/view", "_self");
    }
    else if (closeWindow == true) {
        try {
            window.opener.updateGrid();
            window.opener.closeWindow();
        }
        catch (ex) { window.close(); }
    }
}

function onDeleteClick() {
    $('#ppWarning').modal('show');
}

function onYesClick() {
    let id = $('#txtID').val();

    $.post('/deletePilot/' + id, {}, function (data, status) {
        $('#ppWarning').modal('hide');

        if (data=='true')
        {
            $('#lblMsg').text('Ο πιλότος διαγράφηκε επιτυχώς.');
```

Εικόνα 212. Σελίδα javascript επεξεργασίας πιλότου (1)

```

if (message != '') {
    message = '<ul>' + message + '</ul>';
    $('#lblMsg').html(message);
    $('#ppMsg').modal('show');

    isValid = false;
}
else {
    if (previousID === '')
    {
        previousID = '0';
    }

    $.ajax({
        type: 'POST',
        url: '/validatePilot/' + id + '/' + previousID,
        data: {},
        dataType: 'json',
        async: false,
        success: function (data, status, xhr) {
            let msg = '';

            if (data.idExists == true) {
                msg = '<li>0 αριθμός ταυτότητας υπάρχει ήδη.</li>';
            }

            if (msg != '') {
                $('#lblMsg').html('<ul>' + msg + '</ul>');
                $('#ppMsg').modal('show');

                isValid = false;
            }
            else {
                isValid = true;
            }
        }
    });
}

return isValid;
}

function onSaveClick(close) {
    let prevID = $('#hf').val();

    if (prevID === '')
    {
        prevID = '0';
    }

    closeWindow = close;
    let isValid = validate(prevID);

    if (isValid == true) {
        let previousID = prevID;
        let id = $('#txtID').val();
        let fn = $('#txtFirstName').val();
        let ln = $('#txtLastName').val();
        let str = $('#txtStreet').val();
        let strNo = $('#txtStreetNo').val();
        let city = $('#txtCity').val();
        let phoneNo = $('#txtPhoneNo').val();
        let birthDate = $('#txtBirthDate').val();
        let flightHours = $('#txtHours').val();

        pilot = new Pilot(previousID, id, fn, ln, str, strNo, city, phoneNo, birthDate, flightHours);

        $.ajax({
            type: 'POST',
            url: '/savePilot',
            data: pilot,
            dataType: 'json',
            async: false,
            success: function (data, status, xhr) {
                $('#lblMsg').html('0 πιλότος καταχωρήθηκε επιτυχώς. ');
                $('#ppMsg').modal('show');
                openInView = !close;
            }
        });
    }
}

function onEditClick() {
    let id = $('#txtID').val();
    window.open("/pilotEdit/" + id + "/edit", "_self");
}

class Pilot {
    constructor(previousID, id, firstName, lastName, street, streetNo, city, phoneNo, birthDate, flightHours) {
        this.previousID = previousID;
        this.id = id;
        this.firstName = firstName;
        this.lastName = lastName;
        this.street = street;
        this.streetNo = streetNo;
        this.city = city;
        this.phoneNo = phoneNo;
        this.birthDate = birthDate;
        this.flightHours = flightHours;
    }
}

function onLogoutClick(){
    window.open('/logout', '_self');
}

function onHomeClick(){
    window.open('/home', '_self');
}

```

Εικόνα 213. Σελίδα javascript επεξεργασίας πιλότου (2)

```

let myWindow = null;
let selectedID = null;

$(document).ready(function () {
    $('[data-toggle="tooltip"]').tooltip();
});

function editPilot(id) {
    myWindow = window.open('/pilotEdit/' + id + '/edit');
}

function viewPilot(id) {
    myWindow = window.open('/pilotEdit/' + id + '/view');
}

function onNewClick(){
    myWindow = window.open("/pilotEdit/0/edit");
}

function closeWindow(){
    myWindow.close();
    window.open('/pilotSearch', '_self');
}

function updateGrid(){
    window.open('/pilotSearch', '_self');
}

function onYesClick(){
    $.post('/deletePilot/' + selectedID, {}, function (data, status) {
        $('#ppWarning').modal('hide');
        $('#lblMsg').text('Ο πιλότος διαγράφηκε επιτυχώς.');
```

Εικόνα 214. Σελίδα javascript αναζήτησης πιλότου

```

function onShowClick() {

    let request = {
        dateFrom: $('#dtFrom').val(),
        dateTo: $('#dtTo').val()
    };

    $.post('/getReport', request, function (data, status) {
        data = $.parseJSON(data);
        let item = null;

        let html = "<table class='tbl-search'>";
        html += "<th>Κωδικός δρομολογίου</th>";
        html += "<th>Πόλη αναχώρησης</th>";
        html += "<th>Πόλη άφιξης</th>";
        html += "<th>Πληρότητα</th>";

        for (let i = 0; i < data.length; i++) {
            item = $.parseJSON(data[i]);
            html += '<tr><td>' + item.routeID + '</td>';
            html += '<td>' + item.depCity + '</td>';
            html += '<td>' + item.arvCity + '</td>';
            html += '<td class="td-right">' + item.tickets + '</td></tr>';

        }

        html += '</table>';
        $('#report').html(html);
    }

    );

}

function disableToDates()
{
    let dateFrom = $('#dtFrom').val();
    $('#dtTo').prop('min', dateFrom);
}

function disableFromDates()
{
    let dateTo = $('#dtTo').val();
    $('#dtFrom').prop('max', dateTo);
}

```

Εικόνα 215. Σελίδα javascript συγκεντρωτικής αναφοράς πτήσεων

```

let openInView = false;
let closeWindow = false;
let selectedTransport = null;
let currentReservationID = null;
let openTicket = false;
let ticketID = 0;

$(document).ready(onReady);

function onReady() {
    currentReservationID = $('#f').val();

    findRoute();

    let isBusinessClass = null;

    if ($('#f2').val() == 'false') {
        isBusinessClass = false;
    }
    else {
        isBusinessClass = true;
    }

    let transportID1 = $('#f1').val();
    let transportID2 = $('#f2').val();

    let id = 'rb_' + transportID1 + '_' + transportID2 + '_';
    selectedTransport = transportID1 + '_' + transportID2;

    if (isBusinessClass) {
        id = id + 'bus';
    }
    else {
        id = id + 'eco';
    }

    $('#'+ id).attr('checked', true);
}

function onOKClick() {
    $('#ppMsg').modal('hide');

    if (openTicket == true) {
        window.opener.updateGrid();
        window.open("/showTicket/" + ticketID, "_self");
    }
    else if (openInView == true) {
        window.opener.updateGrid();
        window.open("/reservationEdit/" + currentReservationID + "/view", "_self");
    }
    else if (closeWindow == true) {
        try {
            window.opener.updateGrid();
            window.opener.closeWindow();
        }
        catch (ex) { window.close(); }
    }
}

function onDeleteClick() {
    $('#ppWarning').modal('show');
}

function onYesClick() {
    let id = $('#txtID').val();

    $.post('/deleteReservation/' + id, {}, function (data, status) {
        $('#ppWarning').modal('hide');
        $('#lblMsg').text('Η κράτηση διαγράφηκε επιτυχώς.');
```

Εικόνα 216. Σελίδα javascript επεξεργασίας κράτησης ταξιδιωτικού πράκτορα (1)

```

if (message != '') {
    message = '<ul>' + message + '</ul>';
    $('#lblMsg').html(message);
    $('#pppMsg').modal('show');
}

isValid = false;
}
else {
$.ajax({
    type: 'POST',
    url: '/validateReservation',
    data: {
        PhoneNo: phoneNo,
        ResDate: resDate,
        ReservationID: currentReservationID
    },
    success: function (data, status) {
        if (data == '1') {
            message = '<ul><li>Έχετε κάνει ήδη κράτηση για τη συγκεκριμένη ημερομηνία.</li></ul>';
            $('#lblMsg').html(message);
            $('#pppMsg').modal('show');
            isValid = false;
        }
        else {
            isValid = true;
        }
    },
    dataType: 'json',
    async: false
});
return isValid;
}

function onSaveClick(close) {
    if (validate()) {
        let reservationID = $('#txtID').val();
        let phoneNo = $('#cmbClient').val();
        let resDate = $('#dtResDate').val();
        let expDate = $('#dtExpDate').val();
        let depCityID = $('#cmbDepCity').val();
        let arvCityID = $('#cmbArvCity').val();

        let values = selectedTransport.split('_');

        let transportID1 = 0;
        let transportID2 = 0;
        let busSeat = null;

        if (values.length == 2) {
            transportID1 = values[0];

            if (values[1] == 'eco') {
                busSeat = false;
            }
            else {
                busSeat = true;
            }
        }
        else {
            transportID1 = values[0];
            transportID2 = values[1];

            if (values[2] == 'eco') {
                busSeat = false;
            }
            else {
                busSeat = true;
            }
        }

        let reservation = new Reservation(reservationID, phoneNo, resDate, expDate, depCityID, arvCityID, transportID1, transportID2, busSeat);

$.ajax({
    type: 'POST',
    url: '/saveReservation',
    data: reservation,
    dataType: 'json',
    async: false,
    success: function (data, status, xhr) {
        currentReservationID = data;
        $('#lblMsg').html('Η κράτηση καταχωρήθηκε επιτυχώς.');
```

Εικόνα 217. Σελίδα javascript επεξεργασίας κράτησης ταξιδιωτικού πράκτορα (2)

Τμήμα Μηχανικών Πληροφορικής, ΕΛ.ΜΕ.Π.Α Κρήτης

```
function createNoTransportPanel() {
    let panel = '<div style="text-align:center">Δεν βρέθηκαν μεταφορές.</div>';
    return panel;
}

function createDirectTransportPanel(transportItem) {
    let value = transportItem.transportID;
    let panel = '<div class="panel">';
    let panel1 = '<table class="tbl"><tr><td class="time bg-gray" style="border-top-left-radius:5px"> + transportItem.depTime + '</td><td class="bg-gray"></td><td class="time bg-gray"> + transportItem.arrTime + '</td><td class="category">Economy</td><td class="category" style="border-top-right-
radius:5px">Business</td>';
    panel1 += '<tr><td class="city bg-gray"> + transportItem.depCity + '</td><td class="bg-gray"></td><td class="city bg-gray"> + transportItem.arrCity + '</td><td rowspan="2" class="price">&euro;'; +
transportItem.ecoPrice + '</td><td rowspan="2" class="price">&euro; + transportItem.busPrice + '</td></tr>';

    let panel2 = '<tr><td class="time bg-gray" style="padding-top:10px"></td><td class="bg-gray" style="padding-top:10px"></td><td class="time bg-gray" style="padding-top:10px"></td></tr>';
    panel2 += '<tr><td class="city bg-gray" style="border-bottom-left-radius:5px"></td><td class="bg-gray"></td><td class="city bg-gray"></td><td class="radio"><input + (transportItem.ecoSeats == 0 ?
'disabled' : '') + ' id="rdb_' + value + '_eco' type="radio" name="selection" onclick="setSelectedOption('\'' + value + '_eco')"/></td><td class="radio" style="border-bottom-right-radius:5px"><input
+ (transportItem.busSeats == 0 ? 'disabled' : '') + ' type="radio" id="rdb_' + value + '_bus' name="selection" onclick="setSelectedOption('\'' + value + '_bus')"/></td></tr>';

    panel += panel1 + panel2 + '</div>';
}

return panel;
}

function createIndirectTransportPanel(transportItem1, transportItem2) {
    let value = transportItem1.transportID + '_' + transportItem2.transportID;
    let panel = '<div class="panel">';
    let panel1 = '<table class="tbl"><tr><td class="time bg-gray" style="border-top-left-radius:5px"> + transportItem1.depTime + '</td><td class="bg-gray"></td><td class="time bg-gray"> + transportItem1.arrTime + '</td><td class="category">Economy</td><td class="category" style="border-top-right-
radius:5px">Business</td>';
    panel1 += '<tr><td class="city bg-gray"> + transportItem1.depCity + '</td><td class="bg-gray"></td><td class="city bg-gray"> + transportItem1.arrCity + '</td><td rowspan="2" class="price">&euro;'; +
(transportItem1.ecoPrice + transportItem2.ecoPrice) + '</td><td rowspan="2" class="price">&euro; + (transportItem1.busPrice + transportItem2.busPrice) + '</td></tr>';

    let panel2 = '<tr><td class="time bg-gray" style="padding-top:10px"> + transportItem2.depTime + '</td><td class="bg-gray" style="padding-top:10px"></td><td class="time bg-gray" style="padding-top:10px"> + transportItem2.arrTime + '</td></tr>';
    panel2 += '<tr><td class="city bg-gray" style="border-bottom-left-radius:5px"> + transportItem1.depCity + '</td><td class="bg-gray"></td><td class="city bg-gray"> + transportItem2.arrCity + '</td>
<td class="radio"><input type="radio" id="rdb_' + transportItem1.transportID + '_' + transportItem2.transportID + '_eco' name="selection" onclick="setSelectedOption('\'' + value + '_eco')"/></td><td
class="radio" style="border-bottom-right-radius:5px"><input type="radio" id="rdb_' + transportItem1.transportID + '_' + transportItem2.transportID + '_bus' name="selection"
onclick="setSelectedOption('\'' + value + '_bus')"/></td></tr>';

    panel += panel1 + panel2 + '</div>';
}

return panel;
}

function setSelectedOption(value) {
    selectedTransport = value;
}

function onPrintClick() {
    $.ajax({
        type: 'POST',
        url: '/printTicket',
        data: {
            reservationID: currentReservationID
        },
        success: function (data, status) {
            ticketID = data;
            openTicket = true;
            $('#tblMsg').html('Η έκδοση του εισιτηρίου έγινε επιτυχώς.<br />0 αριθμός εισιτηρίου σας είναι: ' + data + '<br />Καλό σας ταξίδι!');
            $('#ppMsg').modal('show');
        },
        dataType: 'json',
        async: false
    });
}

class Reservation {
    constructor(reservationID, phoneNo, resDate, expDate, depCityID, arrCityID, transportID1, transportID2, busSeat) {
        this.reservationID = reservationID;
        this.phoneNo = phoneNo;
        this.resDate = resDate;
        this.expDate = expDate;
        this.depCityID = depCityID;
        this.arrCityID = arrCityID;
        this.transportID1 = transportID1;
        this.transportID2 = transportID2;
        this.busSeat = busSeat;
    }
}

function onLogoutClick() {
    window.open('/logout', '_self');
}

function onHomeClick() {
    window.open('/home', '_self');
}

function disableExpirationDates() {
    let dateFrom = $('#dtResDate').val();
    let d = new Date(dateFrom);

    let minDate = d;
    minDate.setDate(d.getDate() + 1);

    let month = minDate.getMonth() + 1;
    if (month < 10) {
        month = '0' + month.toString();
    }
}

$('#dtExpDate').prop('min', minDate.getFullYear() + '-' + month.toString() + '-' + (minDate.getDate() < 10 ? '0' + minDate.getDate().toString() : minDate.getDate()));
}

function disableReservationDates() {
    let dateTo = $('#dtExpDate').val();
    let d = new Date(dateTo);

    let maxDate = d;
    maxDate.setDate(d.getDate() - 1);

    let month = maxDate.getMonth() + 1;
    if (month < 10) {
        month = '0' + month.toString();
    }
}

$('#dtResDate').prop('max', maxDate.getFullYear() + '-' + month.toString() + '-' + (maxDate.getDate() < 10 ? '0' + maxDate.getDate().toString() : maxDate.getDate()));
}
```

Εικόνα 218. Σελίδα javascript επεξεργασίας κράτησης ταξιδιωτικού πράκτορα (3)

```

let myWindow = null;
let selectedID = null;

$(document).ready(function () {
  $('[data-toggle="tooltip"]').tooltip();
});

function editReservation(reservationID) {
  myWindow = window.open('/reservationEdit/' + reservationID + '/edit');
}

function viewReservation(reservationID) {
  myWindow = window.open('/reservationEdit/' + reservationID + '/view');
}

function onNewClick(){
  myWindow = window.open("/reservationEdit/0/edit");
}

function closeWindow(){
  myWindow.close();
  window.open('/reservationSearch', '_self');
}

function updateGrid(){
  window.open('/reservationSearch', '_self');
}

function onYesClick(){
  $.post('/deleteReservation/' + selectedID, {}, function (data, status) {
    $('#ppWarning').modal('hide');
    $('#lblMsg').text('Η κράτηση διαγράφηκε επιτυχώς.');
```

Εικόνα 219. Σελίδα javascript αναζήτησης κράτησης ταξιδιωτικού πράκτορα

```

let openInView = false;
let closeWindow = false;

function onOKClick(){
    $('#ppMsg').modal('hide');

    if (openInView == true){
        window.opener.updateGrid();
        let id = $('#txtID').val();
        window.open("/routeEdit/" + id + "/view", "_self");
    }
    else if (closeWindow == true) {
        try {
            window.opener.updateGrid();
            window.opener.closeWindow();
        }
        catch (ex) { window.close(); }
    }
}

function onDeleteClick(){
    $('#ppWarning').modal('show');
}

function onYesClick(){
    let id = $('#txtID').val();

    $.post('/deleteRoute/' + id, {}, function(data, status){
        $('#ppWarning').modal('hide');
        if(data=='true'){
            $('#lblMsg').text('Το δρομολόγιο διαγράφηκε επιτυχώς.');
            closeWindow = true;
        }
        else{
            $('#lblMsg').text('Δεν είναι δυνατή η διαγραφή του δρομολογίου, επειδή συσχετίζεται με άλλες εγγραφές.');
        }
        $('#ppMsg').modal('show');
    });
}

function onNoClick(){
    $('#ppWarning').modal('hide');
}

function onCancelClick(){
    window.location.reload();
}

function validate(prevID){
    let isValid = null;
    let message = '';

    let id = $('#txtID').val();
    if (id == ''){
        message = message + '<li>Συμπληρώστε τον κωδικό.</li>';
    }

    let depPointID = $('#cmbDepPoint').val();
    if (depPointID == 0){
        message = message + '<li>Συμπληρώστε την αφετηρία.</li>';
    }

    let destPointID = $('#cmbDestPoint').val();
    if (destPointID == 0){
        message = message + '<li>Συμπληρώστε τον προορισμό.</li>';
    }

    let dayID = $('#cmbDay').val();
    if (dayID == 0){
        message = message + '<li>Συμπληρώστε την ημέρα πραγματοποίησης του ταξιδιού.</li>';
    }

    let distance = $('#txtDistance').val();
    if (distance <= 0){
        message = message + '<li>Η απόσταση πρέπει να είναι θετικός αριθμός.</li>';
    }

    let depTime = $('#txtDepartureTime').val();
    if (depTime == ''){
        message = message + '<li>Συμπληρώστε την ώρα αναχώρησης.</li>';
    }

    let arvTime = $('#txtArrivalTime').val();
    if (arvTime == ''){
        message = message + '<li>Συμπληρώστε την ώρα άφιξης.</li>';
    }

    let busPrice = $('#txtBusPrice').val();
    if (busPrice <= 0){
        message = message + '<li>Η τιμή διακεκριμένης θέσης πρέπει να είναι θετικός αριθμός.</li>';
    }

    let ecoPrice = $('#txtEcoPrice').val();
    if (ecoPrice <= 0){
        message = message + '<li>Η τιμή οικονομικής θέσης πρέπει να είναι θετικός αριθμός.</li>';
    }

    if (depPointID == destPointID){
        message = message + '<li>Επιλέξτε διαφορετικό προορισμό.</li>';
    }

    let depTimeValues = depTime.split(':');
    let arvTimeValues = arvTime.split(':');

    let dtDepTime = new Date(2020,01,01, depTimeValues[0], depTimeValues[1]);
    let dtArvTime = new Date(2020,01,01, arvTimeValues[0], arvTimeValues[1]);

    if (dtArvTime.getTime() < dtDepTime.getTime())
    {
        message = message + '<li>Η ώρα αναχώρησης δεν μπορεί να είναι μεταγενέστερη της ώρας άφιξης.</li>';
    }
    else if(dtArvTime.getTime() - dtDepTime.getTime() < 30 * 60 * 1000)
    {
        message = message + '<li>Η ώρα αναχώρησης και η ώρα άφιξης πρέπει να διαφέρουν τουλάχιστον μισή ώρα.</li>';
    }
}

```

Εικόνα 220. Σελίδα javascript επεξεργασίας δρομολογίου (1)

```

if (message != ''){
    message = '<ul>' + message + '</ul>';
    $('#lblMsg').html(message);
    $('#ppMsg').modal('show');

    isValid = false;
}
else {
    if (prevID === '')
    {
        prevID = '0';
    }

    $.ajax({
        type: 'POST',
        url: '/validateRoute/' + id + '/' + prevID,
        data: {},
        dataType: 'json',
        async: false,
        success: function (data, status, xhr) {
            let msg = '';

            if (data.idExists == true) {
                msg += '<li>Το δρομολόγιο υπάρχει ήδη.</li>';
            }

            if (msg != '') {
                $('#lblMsg').html('<ul>' + msg + '</ul>');
                $('#ppMsg').modal('show');

                isValid = false;
            }
            else {
                isValid = true;
            }
        }
    });
}

return isValid;
}

function onSaveClick(close){
    let prevID = $('#hf').val();
    if (prevID == '')
    {
        prevID = '0';
    }

    closeWindow = close;
    let isValid = validate(prevID);

    if (isValid == true){
        let previousID = prevID;
        let id = $('#txtID').val();
        let depPointID = $('#cmbDepPoint').val();
        let destPointID = $('#cmbDestPoint').val();
        let dayID = $('#cmbDay').val();
        let distance = $('#txtDistance').val();
        let depTime = $('#txtDepartureTime').val();
        let arvTime = $('#txtArrivalTime').val();
        let busPrice = $('#txtBusPrice').val();
        let ecoPrice = $('#txtEcoPrice').val();

        route = new Route(previousID, id, depPointID, destPointID, dayID, distance, depTime, arvTime, busPrice, ecoPrice);

        $.ajax({
            type: 'POST',
            url: '/saveRoute',
            data: route,
            dataType: 'json',
            async: false,
            success: function(data,status,xhr){
                $('#lblMsg').html('Το δρομολόγιο καταχωρήθηκε επιτυχώς. ');
                $('#ppMsg').modal('show');
                openInView = !close;
            }
        });
    }
}

function onEditClick(){
    let id = $('#txtID').val();
    window.open("/routeEdit/" + id + "/edit", "_self");
}

class Route{
    constructor(previousID, id, depPointID, destPointID, dayID, distance, depTime, arvTime, busPrice, ecoPrice){
        this.previousID = previousID;
        this.id = id;
        this.depPointID = depPointID;
        this.destPointID = destPointID;
        this.dayID = dayID;
        this.distance = distance;
        this.depTime = depTime;
        this.arvTime = arvTime;
        this.busPrice = busPrice;
        this.ecoPrice = ecoPrice;
    }
}

function onlogoutClick(){
    window.open('/logout', '_self');
}

function onhomeClick(){
    window.open('/home', '_self');
}

```

Εικόνα 221. Σελίδα javascript επεξεργασίας δρομολογίου (2)

```

let myWindow = null;
let selectedID = null;

$(document).ready(function () {
    $('[data-toggle="tooltip"]').tooltip();
});

function editRoute(id) {
    myWindow = window.open('/routeEdit/' + id + '/edit');
}

function viewRoute(id) {
    myWindow = window.open('/routeEdit/' + id + '/view');
}

function onNewClick(){
    myWindow = window.open("/routeEdit/0/edit");
}

function closeWindow(){
    myWindow.close();
    window.open('/routeSearch', '_self');
}

function updateGrid(){
    window.open('/routeSearch', '_self');
}

function onYesClick(){
    $.post('/deleteRoute/' + selectedID, {}, function (data, status) {
        $('#ppWarning').modal('hide');
        $('#lblMsg').text('Το δρομολόγιο διαγράφηκε επιτυχώς.');
```

Εικόνα 222. Σελίδα javascript αναζήτησης δρομολογίου

```
$(document).ready(onReady);  
  
function onReady(){  
    let price = parseFloat($('#hf').val());  
  
    if (price == 0)  
    {  
        $('#ppMsg').modal('show');  
    }  
}  
  
function onOKClick(){  
    $('#ppMsg').modal('hide');  
}  
  
function onlogoutClick(){  
    window.open('/logout', '_self');  
}  
  
function onhomeClick(){  
    window.open('/home', '_self');  
}
```

Εικόνα 223. Σελίδα javascript προεπισκόπησης εισιτηρίου

```

let openInView = false;
let closeWindow = false;
let newTransportID = null;

function onOKClick() {
    $('#ppMsg').modal('hide');

    if (openInView == true) {
        window.opener.updateGrid();
        window.open("/transportEdit/" + newTransportID + "/view", "_self");
    }
    else if (closeWindow == true) {
        try {
            window.opener.updateGrid();
            window.opener.closeWindow();
        }
        catch (ex) { window.close(); }
    }
}

function onDeleteClick() {
    $('#ppWarning').modal('show');
}

function onYesClick(){
    let routeID = $('#cmbRoute').val();

    $.post('/deleteTransport/' + routeID, {}, function (data, status) {
        $('#ppWarning').modal('hide');
        $('#tblMsg').text('Η μεταφορά διαγράφηκε επιτυχώς.');
```

}

```

        $('#ppMsg').modal('show');
        closeWindow = true;
    });
}

function onNoClick() {
    $('#ppWarning').modal('hide');
}

function onCancelClick() {
    window.location.reload();
}

function validate() {
    let isValid = null;
    let message = '';

    let routeID = $('#cmbRoute').val();
    if (routeID == '0') {
        message = message + '<li>Επιλέξτε δρομολόγιο.</li>';
    }

    let transportDate = $('#dtDate').val();
    if (transportDate == '') {
        message = message + '<li>Επιλέξτε ημερομηνία.</li>';
    }

    let tokens = $('#cmbRoute option:selected').text().split(' - ');
    let dayName = tokens[2];
    let dayIndex = null;

    switch (dayName) {
        case 'ΔΕΥΤΕΡΑ':
            dayIndex = 1;
            break;

        case 'ΤΡΙΤΗ':
            dayIndex = 2;
            break;

        case 'ΤΕΤΑΡΤΗ':
            dayIndex = 3;
            break;

        case 'ΠΕΜΠΤΗ':
            dayIndex = 4;
            break;

        case 'ΠΑΡΑΣΚΕΥΗ':
            dayIndex = 5;
            break;

        case 'ΣΑΒΒΑΤΟ':
            dayIndex = 6;
            break;

        case 'ΚΥΡΙΑΚΗ':
            dayIndex = 0;
            break;
    }
}

```

Εικόνα 224. Σελίδα javascript επεξεργασίας μεταφοράς (1)

```

let transDate = new Date(transportDate);
let dayIndex2 = transDate.getDay();

if (dayIndex != dayIndex2) {
    message = message + '<li>Η ημερομηνία που επέλεξε δεν αντιστοιχεί στην ημέρα του δρομολογίου.</li>';
}

let airplaneName = $('#cmbPlane').val();
if (airplaneName == '') {
    message = message + '<li>Επιλέξτε αεροπλάνο.</li>';
}

let managerID = $('#cmbManager').val();
if (managerID == '0') {
    message = message + '<li>Επιλέξτε υπεύθυνο.</li>';
}

let pilotID = $('#cmbPilot').val();
if (pilotID == '0') {
    message = message + '<li>Επιλέξτε κυβερνήτη.</li>';
}

let coPilotID = $('#cmbCoPilot').val();
if (coPilotID == '0') {
    message = message + '<li>Επιλέξτε συγκυβερνήτη.</li>';
}

let airHostessID1 = $('#cmbHostess1').val();
if (airHostessID1 == '0') {
    message = message + '<li>Επιλέξτε 1η αεροσυνοδό.</li>';
}

let airHostessID2 = $('#cmbHostess2').val();
if (airHostessID2 == '0') {
    message = message + '<li>Επιλέξτε 2η αεροσυνοδό.</li>';
}

let airHostessID3 = $('#cmbHostess3').val();
if (airHostessID3 == '0') {
    message = message + '<li>Επιλέξτε 3η αεροσυνοδό.</li>';
}

let airHostessID4 = $('#cmbHostess4').val();
if (airHostessID4 == '0') {
    message = message + '<li>Επιλέξτε 4η αεροσυνοδό.</li>';
}

if (pilotID == coPilotID) {
    message = message + '<li>Επιλέξτε διαφορετικό συγκυβερνήτη.</li>';
}

let airHostessList = [airHostessID1];
if (airHostessList.indexOf(airHostessID2) != -1) {
    message = message + '<li>Η 2η αεροσυνοδός έχει ανατεθεί πολλαπλές φορές.</li>';
} else {
    airHostessList.push(airHostessID2);
}

if (airHostessList.indexOf(airHostessID3) != -1) {
    message = message + '<li>Η 3η αεροσυνοδός έχει ανατεθεί πολλαπλές φορές.</li>';
} else {
    airHostessList.push(airHostessID3);
}

if (airHostessList.indexOf(airHostessID4) != -1) {
    message = message + '<li>Η 4η αεροσυνοδός έχει ανατεθεί πολλαπλές φορές.</li>';
} else {
    airHostessList.push(airHostessID4);
}

if (message != '') {
    message = '<ul>' + message + '</ul>';
    $('#tblMsg').html(message);
    $('#tblMsg').modal('show');
    isValid = false;
} else {
    let transportID = $('#txtTransportID').val();

    $.ajax({
        type: 'POST',
        url: '/validateTransport/' + transportID + '/' + routeID + '/' + pilotID + '/' + coPilotID + '/' + transportDate + '/' + airHostessID1 + '/' + airHostessID2 + '/' + airHostessID3 + '/' + airHostessID4 + '/' + airplaneName,
        dataType: 'json',
        async: false,
        success: function (data, status, xhr) {
            if (data.hasConflict == true) { //εαν έχουμε conflict
                isValid = false;

                $('#tblMsg').html('<div>Είναι δυνατή η αποθήκευση της μεταφοράς, διότι υπάρχει χρονική επικάλυψη με άλλες μεταφορές.</div>');
                $('#tblMsg').modal('show');
            } else {
                isValid = true;
            }
        }
    });
}

return isValid;
}

```

Εικόνα 225. Σελίδα javascript επεξεργασίας μεταφοράς (2)


```

function onSaveClick(close) {
    let isValid = validate();

    if (isValid == true) {
        let transportID = $('#txtTransportID').val();
        let routeID = $('#cmbRoute').val();
        let pilotID = $('#cmbPilot').val();
        let coPilotID = $('#cmbCoPilot').val();
        let transportDate = $('#dtDate').val();
        let airHostessID1 = $('#cmbHostess1').val();
        let airHostessID2 = $('#cmbHostess2').val();
        let airHostessID3 = $('#cmbHostess3').val();
        let airHostessID4 = $('#cmbHostess4').val();
        let airplaneName = $('#cmbPlane').val();
        let managerID = $('#cmbManager').val();
        let ecoSeats = $('#txtEcoSeats').val();
        let busSeats = $('#txtBusSeats').val();

        let transport = new Transport(transportID, routeID, pilotID, coPilotID, transportDate, airHostessID1, airHostessID2, airHostessID3, airHostessID4, airplaneName, managerID, ecoSeats, busSeats);

        $.ajax({
            type: 'POST',
            url: '/saveTransport',
            data: transport,
            dataType: 'json',
            async: false,
            success: function (data, status, xhr) {
                newTransportID = data;
                $('#tblMsg').html('Η μεταφορά καταχωρήθηκε επιτυχώς.');
```

Εικόνα 226. Σελίδα javascript επεξεργασίας μεταφοράς (3)

```

let myWindow = null;
let selectedID = null;

$(document).ready(function () {
    $('[data-toggle="tooltip"]').tooltip();
});

function editTransport(transportID) {
    myWindow = window.open('/transportEdit/' + transportID + '/edit');
}

function viewTransport(transportID) {
    myWindow = window.open('/transportEdit/' + transportID + '/view');
}

function onNewClick(){
    myWindow = window.open("/transportEdit/0/edit");
}

function closeWindow(){
    myWindow.close();
    window.open('/transportSearch', '_self');
}

function updateGrid(){
    window.open('/transportSearch', '_self');
}

function onYesClick(){
    $.post('/deleteTransport/' + selectedID, {}, function (data, status) {
        $('#ppWarning').modal('hide');
        $('#lblMsg').text('Η μεταφορά διαγράφηκε επιτυχώς. ');
        $('#ppMsg').modal('show');
    });
}

function onNoClick(){
    $('#ppWarning').modal('hide');
}

function deleteTransport(id)
{
    selectedID = id;
    $('#ppWarning').modal('show');
}

function onOKClick(){
    updateGrid();
}

```

Εικόνα 227. Σελίδα javascript αναζήτησης μεταφοράς

```

let openInView = false;
let closeWindow = false;

$(document).ready(function () {
    isNew = $.parseJSON($('#hf').val().toLowerCase());
});

function onOKClick() {
    $('#ppMsg').modal('hide');

    if(openInView == true){
        window.opener.updateGrid();
        let username = $('#txtUsername').val();
        window.open('/userAdministratorEdit/' + username + '/view', "_self");
    }
    else if(closeWindow == true){
        try {
            window.opener.updateGrid();
            window.opener.closeWindow();
        }
        catch (ex) { window.close(); }
    }
}

function onDeleteClick() {
    $('#ppWarning').modal('show');
}

function onYesClick() {
    let username = $('#txtUsername').val();

    $.post('/deleteUser/' + username, {}, function (data, status) {
        $('#ppWarning').modal('hide');
        $('#lblMsg').text('Ο χρήστης διαγράφηκε επιτυχώς.');
```

Εικόνα 228. Σελίδα javascript επεξεργασίας χρήση-διαχειριστή (1)

```

if (message != '') {
    message = '<ul>' + message + '</ul>';
    $('#lblMsg').html(message);
    $('#pppMsg').modal('show');

    isValid = false;
}
else if (isNew == true && password != confirmPassword) {
    message = '<ul><li>Οι κωδικοί πρόσβασης δεν συμφωνούν. Προσπαθήστε ξανά.</li></ul>';
    $('#lblMsg').html(message);
    $('#pppMsg').modal('show');

    isValid = false;
}
else {
    if (obj.previousUsername === '')
    {
        obj.previousUsername = '0';
    }

    if (obj.previousID === '')
    {
        obj.previousID = '0';
    }

    $.ajax({
        type: 'POST',
        url: '/validateUserAdministrator/' + username + '/' + id + '/' + obj.previousUsername + '/' + obj.previousID,
        data: {},
        dataType: 'json',
        async: false,
        success: function (data, status, xhr) {
            let msg = '';

            if (data.idExists == true) {
                msg = '<li>Ο αριθμός ταυτότητας υπάρχει ήδη.</li>';
            }

            if (data.usernameExists == true) {
                msg += '<li>Το όνομα χρήστη υπάρχει ήδη.</li>';
            }

            if (msg != '') {
                $('#lblMsg').html('<ul>' + msg + '</ul>');
                $('#pppMsg').modal('show');

                isValid = false;
            }
            else {
                isValid = true;
            }
        }
    });
}

return isValid;
}

function onSaveClick(close) {
    let obj = $.parseJSON($('#hfPrev').val().replace(/\'/g, ''));
    closeWindow = close;
    let isValid = validate(obj);

    if (isValid == true) {
        let previousUsername = obj.previousUsername;
        let previousID = obj.previousID;
        let username = $('#txtUsername').val();
        let password = $('#txtPassword').val();
        let id = $('#txtID').val();
        let fn = $('#txtFirstName').val();
        let ln = $('#txtLastName').val();
        let str = $('#txtStreet').val();
        let strNo = $('#txtStreetNo').val();
        let city = $('#txtCity').val();
        let phoneNo = $('#txtPhoneNo').val();
        let birthDate = $('#txtBirthDate').val();

        userAdministrator = new UserAdministrator(previousUsername, previousID, username, password, id, fn, ln, str, strNo, city, phoneNo, birthDate);

        $.ajax({
            type: 'POST',
            url: '/saveUserAdministrator',
            data: userAdministrator,
            dataType: 'json',
            async: false,
            success: function (data, status, xhr) {
                openInView = true;
                $('#lblMsg').html('Ο χρήστης καταχωρήθηκε επιτυχώς.');
```

Εικόνα 229. Σελίδα javascript επεξεργασίας χρήστη-διαχειριστή (2)

```

function onSavePasswordClick() {
    let isValid = null;
    let message = '';

    let username = $('#txtUsername').val();

    let oldPassword = $('#txtOldPassword').val();
    let newPassword = $('#txtNewPassword').val();
    let confirmPassword = $('#txtConfirmPassword').val();

    if (oldPassword == '' || newPassword == '' || confirmPassword == '') {
        $('#lblPasswordMsg').html('Συμπληρώστε τα απαραίτητα πεδία.');
```

\$('#lblPasswordMsg').show();

```
    }
    else {
        if (newPassword == confirmPassword) {
            $.ajax({
                type: 'POST',
                url: '/changePassword',
                data: { 'username': username, 'oldPassword': oldPassword, 'newPassword': newPassword },
                dataType: 'json',
                async: false,
                success: function (data, status, xhr) {
                    let counter = $.parseJSON(data);

                    if (counter == 1) {
                        $('#lblMsg').html('Η αλλαγή του κωδικού έγινε με επιτυχία.');
```

\$('#ppPassword').modal('hide');

\$('#ppMsg').modal('show');

```
                    }
                    else {
                        $('#lblPasswordMsg').html('Ο παλαιός κωδικός δεν είναι σωστός.');
```

\$('#lblPasswordMsg').show();

```
                    }
                });
            }
            else {
                $('#lblPasswordMsg').html('Η επιβεβαίωση κωδικού δεν είναι σωστή.');
```

\$('#lblPasswordMsg').show();

```
            }
        }
    }
}

function onCancelPasswordClick() {
    $('#ppPassword').modal('hide');
}

class UserAdministrator {
    constructor(previousUsername, previousID, username, password, id, firstName, lastName, street, streetNo, city, phoneNo, birthDate) {
        this.previousUsername = previousUsername;
        this.previousID = previousID;
        this.username = username;
        this.password = password;
        this.id = id;
        this.firstName = firstName;
        this.lastName = lastName;
        this.street = street;
        this.streetNo = streetNo;
        this.city = city;
        this.phoneNo = phoneNo;
        this.birthDate = birthDate;
    }
}

function onlogoutClick(){
    window.open('/logout', '_self');
}

function onhomeClick(){
    window.open('/home', '_self');
}

```

Εικόνα 230. Σελίδα javascript επεξεργασίας χρήση-διαχειριστή (3)

```

let myWindow = null;
let selectedUsername = null;

$(document).ready(function () {
    $('[data-toggle="tooltip"]').tooltip();
});

function editUserAdministrator(username) {
    myWindow = window.open('/userAdministratorEdit/' + username + '/edit', '_blank');
}

function viewUserAdministrator(username) {
    myWindow = window.open('/userAdministratorEdit/' + username + '/view', '_blank');
}

function onNewClick(){
    myWindow = window.open('/userAdministratorEdit/0/edit', '_blank');
}

function closeWindow(){
    myWindow.close();
    window.open('/userAdministratorSearch', '_self');
}

function updateGrid(){
    window.open('/userAdministratorSearch', '_self');
}

function onYesClick(){
    $.post('/deleteUser/' + selectedUsername, {}, function (data, status) {
        $('#ppWarning').modal('hide');
        $('#lblMsg').text('Ο διαχειριστής διαγράφηκε επιτυχώς.');
```

Εικόνα 231. Σελίδα javascript αναζήτησης χρήστη-διαχειριστή

```

let isNew = null;
let openInView = false;
let closeWindow = false;

$(document).ready(function () {
    isNew = $.parseJSON($('#hf').val().toLowerCase());
});

function onOKClick() {
    $('#ppMsg').modal('hide');

    if(openInView == true){
        window.opener.updateGrid();
        let username = $('#txtUsername').val();
        window.open('/userAgentEdit/' + username + '/view', "_self");
    }
    else if(closeWindow == true){
        try {
            window.opener.updateGrid();
            window.opener.closeWindow();
        }
        catch (ex) { window.close(); }
    }
}

function onDeleteClick() {
    $('#ppWarning').modal('show');
}

function onYesClick() {
    let username = $('#txtUsername').val();

    $.post('/deleteUser/' + username, {}, function (data, status) {
        $('#ppWarning').modal('hide');
        $('#tblMsg').text('Ο χρήστης διαγράφηκε επιτυχώς. ');
        $('#ppMsg').modal('show');
        closeWindow = true;
    });
}

function onNoClick() {
    $('#ppWarning').modal('hide');
}

function onCancelClick() {
    window.location.reload();
}

function validate(obj) {
    let isValid = null;
    let message = '';

    let id = $('#txtID').val();
    if (id == '') {
        message = message + '<li>Συμπληρώστε τον αριθμό ταυτότητας.</li>';
    }

    let fname = $('#txtFirstName').val();
    if (fname == '') {
        message = message + '<li>Συμπληρώστε το όνομά σας.</li>';
    }

    let lname = $('#txtLastName').val();
    if (lname == '') {
        message = message + '<li>Συμπληρώστε το επώνυμό σας.</li>';
    }

    let street = $('#txtAddress').val();
    if (street == '') {
        message = message + '<li>Συμπληρώστε την οδό.</li>';
    }

    let streetNo = $('#txtAddressNo').val();
    if (streetNo == '') {
        message = message + '<li>Συμπληρώστε τον αριθμό της οδού.</li>';
    }

    let city = $('#txtCity').val();
    if (city == '') {
        message = message + '<li>Συμπληρώστε το όνομα της πόλης.</li>';
    }

    let code = $('#txtCode').val();
    if (code == '') {
        message = message + '<li>Συμπληρώστε τον κωδικό σας.</li>';
    }

    let username = $('#txtUsername').val();
    if (username == '') {
        message = message + '<li>Συμπληρώστε το όνομα χρήστη.</li>';
    }

    let password = null;
    let confirmPassword = null;

    if (isNew == true) {
        password = $('#txtPassword').val();
        if (password == '') {
            message = message + '<li>Συμπληρώστε τον κωδικό.</li>';
        }

        confirmPassword = $('#txtConfirm').val();
        if (confirmPassword == '') {
            message = message + '<li>Επιβεβαιώστε τον κωδικό πρόσβασης.</li>';
        }
    }
}

```

Εικόνα 232. Σελίδα javascript επεξεργασίας χρήστη-ταξιδιωτικού πράκτορα (1)

```

if (message != '') {
    message = '<ul>' + message + '</ul>';
    $('#lblMsg').html(message);
    $('#ppMsg').modal('show');

    isValid = false;
}
else if (isNew == true && password != confirmPassword) {
    message = '<ul><li>Οι κωδικοί πρόσβασης δεν συμφωνούν. Προσπαθήστε ξανά.</li></ul>';
    $('#lblMsg').html(message);
    $('#ppMsg').modal('show');

    isValid = false;
}
else {
    if (obj.previousUsername === '')
    {
        obj.previousUsername = '0';
    }

    if (obj.previousCode === '')
    {
        obj.previousCode = '0';
    }

    if (obj.previousID === '')
    {
        obj.previousID = '0';
    }

    $.ajax({
        type: 'POST',
        url: '/validateUserAgent/' + username + '/' + code + '/' + id + '/' + obj.previousUsername + '/' + obj.previousCode + '/' + obj.previousID,
        data: {},
        dataType: 'json',
        async: false,
        success: function (data, status, xhr) {
            let msg = '';

            if (data.idExists == true) {
                msg = '<li>Ο αριθμός ταυτότητας υπάρχει ήδη.</li>';
            }

            if (data.usernameExists == true) {
                msg += '<li>Το όνομα χρήστη υπάρχει ήδη.</li>';
            }

            if (data.codeExists == true) {
                msg += '<li>Ο κωδικός υπάρχει ήδη.</li>';
            }

            if (msg != '') {
                $('#lblMsg').html('<ul>' + msg + '</ul>');
                $('#ppMsg').modal('show');

                isValid = false;
            }
            else {
                isValid = true;
            }
        }
    });

    return isValid;
}

function onSaveClick(close) {
    let obj = $.parseJSON($('#hfPrev').val().replace(/\'/g, '\\\''));
    closeWindow = close;
    let isValid = validate(obj);

    if (isValid == true) {
        let username = $('#txtUsername').val();
        let previousUsername = obj.previousUsername;
        let password = $('#txtPassword').val();
        let id = $('#txtID').val();
        let previousID = obj.previousID;
        let fname = $('#txtFirstName').val();
        let lname = $('#txtLastName').val();
        let street = $('#txtAddress').val();
        let streetNo = $('#txtAddressNo').val();
        let city = $('#txtCity').val();
        let code = $('#txtCode').val();
        let previousCode = obj.previousCode;

        userAgent = new UserAgent(previousUsername, previousCode, previousID, username, password, id, fname, lname, street, streetNo, city, code)

        $.ajax({
            type: 'POST',
            url: '/saveUserAgent',
            data: userAgent,
            dataType: 'json',
            async: false,
            success: function (data, status, xhr) {
                $('#lblMsg').html('Ο χρήστης καταχωρήθηκε επιτυχώς.');
```

Εικόνα 233. Σελίδα javascript επεξεργασίας χρήστη-ταξιδιωτικού πράκτορα (2)


```

function onChangePasswordClick() {
    $("#lblPasswordMsg").hide();
    $("#ppPassword").modal('show');
}

function onSavePasswordClick() {
    let isValid = null;
    let message = '';

    let username = $('#txtUsername').val();

    let oldPassword = $('#txtOldPassword').val();
    let newPassword = $('#txtNewPassword').val();
    let confirmPassword = $('#txtConfirmPassword').val();

    if (oldPassword == '' || newPassword == '' || confirmPassword == '') {
        $('#lblPasswordMsg').html('Συμπληρώστε τα απαραίτητα πεδία.');
```

}
{

```

        $('#lblPasswordMsg').show();
    }
    else {
        if (newPassword == confirmPassword) {
            $.ajax({
                type: 'POST',
                url: '/changePassword',
                data: { 'username': username, 'oldPassword': oldPassword, 'newPassword': newPassword },
                dataType: 'json',
                async: false,
                success: function (data, status, xhr) {
                    let counter = $.parseJSON(data);

                    if (counter == 1) {
                        $('#lblMsg').html('Η αλλαγή του κωδικού έγινε με επιτυχία.');
```

}
{

```

                        $('#ppPassword').modal('hide');
                        $('#ppMsg').modal('show');
                    }
                    else {
                        $('#lblPasswordMsg').html('Ο παλιός κωδικός δεν είναι σωστός.');
```

}
{

```

                    $('#lblPasswordMsg').show();
                }
            });
        }
        else {
            $('#lblPasswordMsg').html('Η επιβεβαίωση κωδικού δεν είναι σωστή.');
```

}
{

```

            $('#lblPasswordMsg').show();
        }
    }
}

function onCancelPasswordClick() {
    $("#ppPassword").modal('hide');
}

function onEditClick(){
    let username = $('#txtUsername').val();
    window.open("/userAgentEdit/" + username + "/edit", "_self");
}

class UserAgent {
    constructor(previousUsername, previousCode, previousID, username, password, id, firstName, lastName, street, streetNo, city, code) {
        this.username = username;
        this.previousUsername = previousUsername;
        this.previousCode = previousCode;
        this.previousID = previousID;
        this.password = password;
        this.id = id;
        this.firstName = firstName;
        this.lastName = lastName;
        this.street = street;
        this.streetNo = streetNo;
        this.city = city;
        this.code = code;
    }
}

function onlogoutClick(){
    window.open('/logout', '_self');
}

function onhomeClick(){
    window.open('/home', '_self');
}

```

Εικόνα 234. Σελίδα javascript επεξεργασίας χρήστη-ταξιδιωτικού πράκτορα (3)

```

let myWindow = null;

$(document).ready(function () {
  $('[data-toggle="tooltip"]').tooltip();
});

function editUserAgent(username) {
  myWindow = window.open('/userAgentEdit/' + username + '/edit');
}

function viewUserAgent(username) {
  myWindow = window.open('/userAgentEdit/' + username + '/view');
}

function onNewClick(){
  myWindow = window.open("/userAgentEdit/0/edit");
}

function closeWindow(){
  myWindow.close();
  window.open('/userAgentSearch', '_self');
}

function updateGrid(){
  window.open('/userAgentSearch', '_self');
}

function onYesClick(){
  $.post('/deleteUser/' + selectedUsername, {}, function (data, status) {
    $('#ppWarning').modal('hide');
    $('#lblMsg').text('Ο ταξιδιωτικός πράκτορας διαγράφηκε επιτυχώς.');
```

Εικόνα 235. Σελίδα javascript αναζήτησης χρήστη-ταξιδιωτικού πράκτορα

```

let isNew = null;
let openInView = false;
let closeWindow = false;

$(document).ready(function () {
    isNew = $.parseJSON($('#hf').val().toLowerCase());
});

function onOKClick() {
    $('#ppMsg').modal('hide');

    if(openInView == true){
        window.opener.updateGrid();
        let username = $('#txtUsername').val();
        window.open("/userClientEdit/" + username + "/view", "_self");
    }
    else if (closeWindow == true) {
        try {
            window.opener.updateGrid();
            window.opener.closeWindow();
        }
        catch (ex) { window.close(); }
    }
}

function onDeleteClick() {
    $('#ppWarning').modal('show');
}

function onYesClick() {
    let username = $('#txtUsername').val();

    $.post('/deleteUser/' + username, {}, function (data, status) {
        $('#ppWarning').modal('hide');
        $('#tblMsg').text('Ο χρήστης διαγράφηκε επιτυχώς.');
```

Εικόνα 236. Σελίδα javascript επεξεργασίας χρήστη-πελάτη (1)

```

if (message != '') {
    message = '<ul>' + message + '</ul>';
    $('#lblMsg').html(message);
    $('#ppMsg').modal('show');

    isValid = false;
}
else if (password != confirmPassword){
    message = '<ul><li>Οι κωδικοί πρόσβασης δεν συμφωνούν. Προσπαθήστε ξανά.</li></ul>';
    $('#lblMsg').html(message);
    $('#ppMsg').modal('show');

    isValid = false;
}
else {
    if (obj.previousUsername === '')
    {
        obj.previousUsername = '0';
    }

    if (obj.previousPhoneNo === '')
    {
        obj.previousPhoneNo = '0';
    }

    $.ajax({
        type: 'POST',
        url: '/validateUserClient/' + username + '/' + phoneNo + '/' + obj.previousUsername + '/' + obj.previousPhoneNo,
        data: {},
        dataType: 'json',
        async: false,
        success: function (data, status, xhr) {
            let msg = '';

            if (data.phoneNoExists == true) {
                msg = '<li>Ο αριθμός τηλεφώνου υπάρχει ήδη.</li>';
            }

            if (data.usernameExists == true) {
                msg += '<li>Το όνομα χρήστη υπάρχει ήδη.</li>';
            }

            if (msg != '') {
                $('#lblMsg').html('<ul>' + msg + '</ul>');
                $('#ppMsg').modal('show');

                isValid = false;
            }
            else {
                isValid = true;
            }
        }
    });

    return isValid;
}

function onSaveClick(close) {
    let obj = $.parseJSON($('#hfPrev').val().replace(/\\/g, '\\'));
    closeWindow = close;
    let isValid = validate(obj);

    if (isValid == true) {
        let username = $('#txtUsername').val();
        let previousUsername = obj.previousUsername;
        let password = $('#txtPassword').val();
        let previousPhoneNo = obj.previousPhoneNo;
        let phoneNo = $('#txtPhoneNo').val();
        let eMail = $('#txtEmail').val();
        let fName = $('#txtFirstName').val();
        let lName = $('#txtLastName').val();
        let street = $('#txtAddress').val();
        let streetNo = $('#txtAddressNo').val();
        let city = $('#txtCity').val();

        userClient = new UserClient(username, previousUsername, password, previousPhoneNo, phoneNo, eMail, fName, lName, street, streetNo, city)

        $.ajax({
            type: 'POST',
            url: '/saveUserClient',
            data: userClient,
            dataType: 'json',
            async: false,
            success: function (data, status, xhr) {
                $('#lblMsg').html('Ο χρήστης καταχωρήθηκε επιτυχώς.');
```

Εικόνα 237. Σελίδα javascript επεξεργασίας χρήστη-πελάτη (2)

```

function onChangePasswordClick() {
    $("#lblPasswordMsg").hide();
    $("#ppPassword").modal('show');
}

function onSavePasswordClick() {
    let isValid = null;
    let message = '';

    let username = $('#txtUsername').val();

    let oldPassword = $('#txtOldPassword').val();
    let newPassword = $('#txtNewPassword').val();
    let confirmPassword = $('#txtConfirmPassword').val();

    if (oldPassword == '' || newPassword == '' || confirmPassword == '') {
        $("#lblPasswordMsg").html('Συμπληρώστε τα απαραίτητα πεδία.');
```

```

        $("#lblPasswordMsg").show();
    }
    else {
        if (newPassword == confirmPassword) {
            $.ajax({
                type: 'POST',
                url: '/changePassword',
                data: { 'username': username, 'oldPassword': oldPassword, 'newPassword': newPassword },
                dataType: 'json',
                async: false,
                success: function (data, status, xhr) {
                    let counter = $.parseJSON(data);

                    if (counter == 1) {
                        $("#lblMsg").html('Η αλλαγή του κωδικού έγινε με επιτυχία.');
```

```

                        $("#ppPassword").modal('hide');
                        $("#ppMsg").modal('show');
                    }
                    else {
                        $("#lblPasswordMsg").html('Ο παλαιός κωδικός δεν είναι σωστός.');
```

```

                    }
                    $("#lblPasswordMsg").show();
                }
            });
        }
        else {
            $("#lblPasswordMsg").html('Η επιβεβαίωση κωδικού δεν είναι σωστή.');
```

```

            $("#lblPasswordMsg").show();
        }
    }
}

function onCancelPasswordClick() {
    $("#ppPassword").modal('hide');
}

function onEditClick(){
    let username = $('#txtUsername').val();
    window.open("/userClientEdit/" + username + "/edit","_self");
}

class UserClient {
    constructor(username, previousUsername, password, previousPhoneNo, phoneNo, eMail, firstName, lastName, street, streetNo, city) {
        this.username = username;
        this.previousUsername = previousUsername;
        this.password = password;
        this.previousPhoneNo = previousPhoneNo;
        this.phoneNo = phoneNo;
        this.eMail = eMail;
        this.firstName = firstName;
        this.lastName = lastName;
        this.street = street;
        this.streetNo = streetNo;
        this.city = city;
    }
}

function onlogoutClick(){
    window.open('/logout','_self');
}

function onhomeClick(){
    window.open('/home','_self');
}

```

Εικόνα 238. Σελίδα javascript επεξεργασίας χρήστη-πελάτη (3)

```
let myWindow = null;

$(document).ready(function () {
  $('[data-toggle="tooltip"]').tooltip();
});

function editUserClient(username) {
  myWindow = window.open('/userClientEdit/' + username + '/edit');
}

function viewUserClient(username) {
  myWindow = window.open('/userClientEdit/' + username + '/view');
}

function onNewClick(){
  myWindow = window.open("/userClientEdit/0/edit");
}

function closeWindow(){
  myWindow.close();
  window.open('/userClientSearch', '_self');
}

function updateGrid(){
  window.open('/userClientSearch', '_self');
}

function onYesClick(){
  $.post('/deleteUser/' + selectedUsername, {}, function (data, status) {
    $('#ppWarning').modal('hide');
    $('#lblMsg').text('Ο πελάτης διαγράφηκε επιτυχώς.');
```

```
    $('#ppMsg').modal('show');
  });
}

function onNoClick(){
  $('#ppWarning').modal('hide');
}

function deleteUserClient(username)
{
  selectedUsername = username;
  $('#ppWarning').modal('show');
}

function onOKClick(){
  updateGrid();
}
```

Εικόνα 239. Σελίδα javascript αναζήτησης χρήστη-πελάτη

```
function onClientsClick(){
    selectTab('tabUserClients');
    $('#frmUsers').attr('src', './userClientSearch');
}

function onAgentsClick(){
    selectTab('tabUserAgents');
    $('#frmUsers').attr('src', './userAgentSearch');
}

function onAdministratorsClick(){
    selectTab('tabUserAdministrators');
    $('#frmUsers').attr('src', './userAdministratorSearch');
}

function deselectAllTabs(){
    $('#tabUserClients').css('background-color', '#483D8B');
    $('#tabUserAgents').css('background-color', '#483D8B');
    $('#tabUserAdministrators').css('background-color', '#483D8B');
}

function selectTab(tabID)
{
    deselectAllTabs();
    $('#' + tabID).css('background-color', '#6e7d9c');
}
```

Εικόνα 240. Σελίδα javascript χρηστών

Δημιουργία σελίδων CSS

```
.field{
    display: flex;
    margin-bottom: 10px;
}

.lbl{
    width: 150px;
    text-align: right;
    margin-right: 10px;
}

.txt{
    width: 250px;
}

#txtAddressNo{
    width: 150px;
}

.buttons{
    display: flex;
    justify-content: flex-end;
    margin-right: 20px;
    margin-bottom: 20px;
    margin-top: 20px;
}

.home{
    cursor: pointer;
}
```

Εικόνα 241. Σελίδα css επεξεργασίας ταξιδιωτικού πράκτορα

```
body{
  padding-left:10px;
  padding-right:10px;
  padding-top: 10px;
  padding-bottom: 10px;
}

.tbl-search
{
  border: 1px solid black;
}

.tbl-search td
{
  border: 1px solid black;
  padding-left: 10px;
  padding-right: 10px;
}

.tbl-search th
{
  border: 1px solid black;
  padding-left: 10px;
  padding-right: 10px;
  background-color: blue;
  color:white;
  text-align: center;
}

.td-centered
{
  text-align: center;
}

a:hover
{
  text-decoration: none;
}
```

Εικόνα 242. Σελίδα css αναζήτησης ταξιδιωτικού πράκτορα

```
.field{
  display: flex;
  margin-bottom: 10px;
}

.lbl{
  width: 150px;
  text-align: right;
  margin-right: 10px;
}

.txt{
  width: 250px;
}

#txtAddressNo{
  width: 150px;
}

.buttons{
  display: flex;
  justify-content: flex-end;
  margin-right: 20px;
  margin-bottom: 20px;
  margin-top: 20px;
}

.home{
  cursor: pointer;
}
```

Εικόνα 243. Σελίδα css επεξεργασίας αεροσυννοδού


```
body{
  padding-left:10px;
  padding-right:10px;
  padding-top: 10px;
  padding-bottom: 10px;
}

.tbl-search
{
  border: 1px solid black;
}

.tbl-search td
{
  border: 1px solid black;
  padding-left: 10px;
  padding-right: 10px;
}

.tbl-search th
{
  border: 1px solid black;
  padding-left: 10px;
  padding-right: 10px;
  background-color: blue;
  color:white;
  text-align: center;
}

.td-centered
{
  text-align: center;
}

a:hover
{
  text-decoration: none;
}
```

Εικόνα 244. Σελίδα css αναζήτησης αεροσυνοδού

```
.field{
  display: flex;
  margin-bottom: 10px;
}

.lbl{
  width: 230px;
  text-align: right;
  margin-right: 10px;
}

.txt{
  width: 250px;
}

#txtNumBusinessSeats{
  width: 60px;
}

#txtNumEconomySeats{
  width: 60px;
}

.buttons{
  display: flex;
  justify-content: flex-end;
  margin-right: 20px;
  margin-bottom: 20px;
  margin-top:20px;
}

.home{
  cursor: pointer;
}
```

Εικόνα 245. Σελίδα css επεξεργασίας αεροπλάνου

```
body{
  padding-left:10px;
  padding-right:10px;
  padding-top: 10px;
  padding-bottom: 10px;
}

.tbl-search
{
  border: 1px solid black;
}

.tbl-search td
{
  border: 1px solid black;
  padding-left: 10px;
  padding-right: 10px;
}

.tbl-search th
{
  border: 1px solid black;
  padding-left: 10px;
  padding-right: 10px;
  background-color: blue;
  color:white;
  text-align: center;
}

.td-centered
{
  text-align: center;
}

a:hover
{
  text-decoration: none;
}
```

Εικόνα 246. Σελίδα css αναζήτησης αεροπλάνου

```
.tbl-search
{
  border: 1px solid black;
}

.tbl-search td
{
  border: 1px solid black;
  padding-left: 10px;
  padding-right: 10px;
}

.tbl-search th
{
  border: 1px solid black;
  padding-left: 10px;
  padding-right: 10px;
  background-color: blue;
  color:white;
  text-align: center;
}

.td-centered
{
  text-align: center;
}

body{
  padding-left:10px;
  padding-right:10px;
  padding-top: 10px;
  padding-bottom: 10px;
}

.tbl-info td
{
  padding-right: 10px;
}
```

Εικόνα 247. Σελίδα css ταμπλό πελάτη

```
.field{
  display: flex;
  margin-bottom: 10px;
}

.lbl{
  width: 75px;
  text-align: right;
  margin-right: 10px;
}

.txt{
  width: 250px;
}

#txtAddressNo{
  width: 150px;
}

.buttons{
  display: flex;
  justify-content: flex-end;
  margin-right: 20px;
  margin-bottom: 20px;
  margin-top: 20px;
}

.home{
  cursor: pointer;
}
```

Εικόνα 248. Σελίδα css επεξεργασίας πελάτη

```
body{
  padding-left: 10px;
  padding-right: 10px;
  padding-top: 10px;
  padding-bottom: 10px;
}

.tbl-search
{
  border: 1px solid black;
}

.tbl-search td
{
  border: 1px solid black;
  padding-left: 10px;
  padding-right: 10px;
}

.tbl-search th
{
  border: 1px solid black;
  padding-left: 10px;
  padding-right: 10px;
  background-color: blue;
  color: white;
  text-align: center;
}

.td-centered
{
  text-align: center;
}

a:hover
{
  text-decoration: none;
}
```

Εικόνα 249. Σελίδα css αναζήτησης πελάτη

```

.field{
  display: flex;
  margin-bottom: 10px;
}

.lbl{
  width: 230px;
  text-align: right;
  margin-right: 10px;
}

.cmb, .dt{
  width: 300px;
}

.buttons{
  display: flex;
  justify-content: flex-end;
  margin-right: 20px;
  margin-bottom: 20px;
  margin-top: 20px;
}

legend
{
  text-align:center;
  border: 1px solid black;
  background-color: #8fa3cc;
  font-size:18px;
  color:white;
}

.panel
{
  border: 1px solid black;
  border-radius: 5px;
}

/*Θέτω στο δεύτερο και στα επόμενά του panels, margin-top=10px*/
.panel:nth-child(n+2)
{
  margin-top:10px;
}

.time
{
  font-size: 18px;
}

.city
{
  font-size:12px;
}

.category
{
  font-size: 14px;
  vertical-align: center;
  padding-left: 10px;
  padding-right: 10px;
  color: black;
  background-color: lightblue;
}

.price
{
  font-size: 16px;
  vertical-align: center;
  text-align: center;
  color:black;
  font-weight: bold;
  background-color: lightblue;
}

.radio
{
  background-color: lightblue;
  text-align: center;
  vertical-align: center;
}

.radio > input:hover
{
  cursor:pointer;
}

.bg-gray
{
  background-color: #669999;
  color:white;
}

.tbl td
{
  padding-left: 10px;
  padding-right: 10px;
}

.home{
  cursor: pointer;
}

```

Εικόνα 250. Σελίδα css επεξεργασίας κράτησης πελάτη

```
body{
  padding-left:10px;
  padding-right:10px;
  padding-top: 10px;
  padding-bottom: 10px;
}

.tbl-search
{
  border: 1px solid black;
}

.tbl-search td
{
  border: 1px solid black;
  padding-left: 10px;
  padding-right: 10px;
}

.tbl-search th
{
  border: 1px solid black;
  padding-left: 10px;
  padding-right: 10px;
  background-color: blue;
  color:white;
  text-align: center;
}

.td-centered
{
  text-align: center;
}

a:hover
{
  text-decoration: none;
}
```

Εικόνα 251. Σελίδα css αναζήτησης κράτησης πελάτη

```
.main-section{
  display:flex;
  flex-flow: row;
  height: calc(100vh - 65px);
}

.tab
{
  background-color: #8fa3cc;
  width: 170px;
  padding-top: 30px;
  padding-bottom: 30px;
  text-align:center;
  color:white;
  font-weight: bold;
}

.tab:hover
{
  cursor:pointer;
  background-color:#6e7d9c!important;
}

.right-section{
  flex-grow: 1;
  height: calc(100vh - 65px);
  overflow-y: auto;
}

#frm
{
  height: calc(100vh - 65px);
  border: none;
  width: 100%;
  display: block;
}

.home{
  cursor: pointer;
}

.left-section{
  background-color:#8fa3cc;
  height: calc(100vh - 65px);
  overflow-y:auto;
}

html,
body{
  min-height: 100%;
}
```

Εικόνα 252. Σελίδα css αρχικής σελίδας

```
.txt
{
  width: 250px;
}

.button-section
{
  margin-top: 20px;
  display: flex;
  justify-content: center;
}

.panel
{
  box-shadow: 3px 3px 3px 3px #888888;
  border: 1px solid black;
  border-radius: 5px;
  padding-top: 20px;
  padding-left: 20px;
  padding-right: 20px;
  padding-bottom: 20px;
  margin-top: 50px;
  background-color: white;
}
```

Εικόνα 253. Σελίδα css εισόδου χρήστη

```
.field{
  display: flex;
  margin-bottom: 10px;
}

.lbl{
  width: 159px;
  text-align: right;
  margin-right: 10px;
}

.txt{
  width: 250px;
}

#txtAddressNo{
  width: 150px;
}

.buttons{
  display: flex;
  justify-content: flex-end;
  margin-right: 20px;
  margin-bottom: 20px;
  margin-top: 20px;
}

.home{
  cursor: pointer;
}
```

Εικόνα 254. Σελίδα css επεξεργασίας διοικητικού υπαλλήλου

```
body{
  padding-left:10px;
  padding-right:10px;
  padding-top: 10px;
  padding-bottom: 10px;
}

.tbl-search
{
  border: 1px solid black;
}

.tbl-search td
{
  border: 1px solid black;
  padding-left: 10px;
  padding-right: 10px;
}

.tbl-search th
{
  border: 1px solid black;
  padding-left: 10px;
  padding-right: 10px;
  background-color: blue;
  color:white;
  text-align: center;
}

.td-centered
{
  text-align: center;
}

a:hover
{
  text-decoration: none;
}
```

Εικόνα 255. Σελίδα css αναζήτησης διοικητικού υπαλλήλου

```
.field{
  display: flex;
  margin-bottom: 10px;
}

.lbl{
  width: 159px;
  text-align: right;
  margin-right: 10px;
}

.txt{
  width: 250px;
}

#txtAddressNo{
  width: 150px;
}

.buttons{
  display: flex;
  justify-content: flex-end;
  margin-right: 20px;
  margin-bottom: 20px;
  margin-top: 20px;
}

.home{
  cursor: pointer;
}
```

Εικόνα 256. Σελίδα css επεξεργασίας πιλότου

```
body{
  padding-left:10px;
  padding-right:10px;
  padding-top: 10px;
  padding-bottom: 10px;
}

.tbl-search
{
  border: 1px solid black;
}

.tbl-search td
{
  border: 1px solid black;
  padding-left: 10px;
  padding-right: 10px;
}

.tbl-search th
{
  border: 1px solid black;
  padding-left: 10px;
  padding-right: 10px;
  background-color: blue;
  color:white;
  text-align: center;
}

.td-centered
{
  text-align: center;
}

a:hover
{
  text-decoration: none;
}
```

Εικόνα 257. Σελίδα css αναζήτησης πιλότου

```
body{
  padding-left:10px;
  padding-right:10px;
  padding-top: 10px;
  padding-bottom: 10px;
}

.data-field{
  display:flex;
  align-content: center;
}

.lbl{
  font-weight:bold;
  margin-right:5px;
}

.container-1{
  display:flex;
  border-radius: 5px;
  border:1px solid black;
  background-color: lightcyan;
  padding:5px;
  margin-bottom:20px;
  margin-top:20px;
}

.tbl-search
{
  border: 1px solid black;
}

.tbl-search td
{
  border: 1px solid black;
  padding-left: 10px;
  padding-right: 10px;
}

.tbl-search th
{
  border: 1px solid black;
  padding-left: 10px;
  padding-right: 10px;
  background-color: blue;
  color:white;
  text-align: center;
}

.td-right
{
  text-align: right;
}
```

Εικόνα 258. Σελίδα css συγκεντρωτικής αναφοράς πτήσεων


```

.field{
  display: flex;
  margin-bottom: 10px;
}

.lbl{
  width: 230px;
  text-align: right;
  margin-right: 10px;
}

.cmb, .dt{
  width: 300px;
}

.buttons{
  display: flex;
  justify-content: flex-end;
  margin-right: 20px;
  margin-bottom: 20px;
  margin-top: 20px;
}

legend
{
  text-align:center;
  border: 1px solid black;
  background-color: #8fa3cc;
  font-size:18px;
  color:white;
}

.panel
{
  border: 1px solid black;
  border-radius: 5px;
}

/*Θέτω στο δεύτερο και στα επόμενά του panels, margin-top=10px*/
.panel:nth-child(n+2)
{
  margin-top:10px;
}

.time
{
  font-size: 18px;
}

.city
{
  font-size:12px;
}

.category
{
  font-size: 14px;
  vertical-align: center;
  padding-left: 10px;
  padding-right: 10px;
  color: black;
  background-color: lightblue;
}

.price
{
  font-size: 16px;
  vertical-align: center;
  text-align: center;
  color:black;
  font-weight: bold;
  background-color: lightblue;
}

.radio
{
  background-color: lightblue;
  text-align: center;
  vertical-align: center;
}

.radio > input:hover
{
  cursor:pointer;
}

.bg-gray
{
  background-color: #669999;
  color:white;
}

.tbl td
{
  padding-left: 10px;
  padding-right: 10px;
}

.home{
  cursor: pointer;
}

```

Εικόνα 259. Σελίδα css επεξεργασίας κράτησης ταξιδιωτικού πράκτορα

```
body{
  padding-left:10px;
  padding-right:10px;
  padding-top: 10px;
  padding-bottom: 10px;
}

.tbl-search
{
  border: 1px solid black;
}

.tbl-search td
{
  border: 1px solid black;
  padding-left: 10px;
  padding-right: 10px;
}

.tbl-search th
{
  border: 1px solid black;
  padding-left: 10px;
  padding-right: 10px;
  background-color: blue;
  color:white;
  text-align: center;
}

.td-centered
{
  text-align: center;
}

a:hover
{
  text-decoration: none;
}
```

Εικόνα 260. Σελίδα css αναζήτησης κράτησης ταξιδιωτικού πράκτορα

```
.field{
  display: flex;
  margin-bottom: 10px;
}

.lbl{
  width: 185px;
  text-align: right;
  margin-right: 10px;
}

.txt{
  width: 250px;
}

.cmb {
  width: 250px;
  height: 31px !important;
  padding:.2rem !important;
}

#txtAddressNo{
  width: 150px;
}

.buttons{
  display: flex;
  justify-content: flex-end;
  margin-right: 20px;
  margin-bottom: 20px;
  margin-top: 20px;
}

.home{
  cursor: pointer;
}
```

Εικόνα 261. Σελίδα css επεξεργασίας δρομολογίου

```
body{
  padding-left:10px;
  padding-right:10px;
  padding-top: 10px;
  padding-bottom: 10px;
}

.tbl-search
{
  border: 1px solid black;
}

.tbl-search td
{
  border: 1px solid black;
  padding-left: 10px;
  padding-right: 10px;
}

.tbl-search th
{
  border: 1px solid black;
  padding-left: 10px;
  padding-right: 10px;
  background-color: blue;
  color:white;
  text-align: center;
}

.td-centered
{
  text-align: center;
}

a:hover
{
  text-decoration: none;
}
```

Εικόνα 262. Σελίδα css αναζήτησης δρομολογίου

```
.field{
  display: flex;
  margin-bottom: 10px;
}

.field > div:first-child
{
  font-weight: bold;
}

.field > div:nth-child(2)
{
  border: 1px solid black;
  background-color: #8fa3cc;
  border-radius: 3px;
  color:white;
  font-weight: bold;
  padding-left: 10px;
  padding-right: 10px;
  width:300px;
}

.lbl{
  width: 150px;
  text-align: right;
  margin-right: 10px;
}

.buttons{
  display: flex;
  justify-content: flex-end;
  margin-right: 20px;
  margin-bottom: 20px;
  margin-top: 20px;
}

.home{
  cursor: pointer;
}

.ticket{
  box-shadow: 3px 3px 3px 3px #888888;
  border: 1px solid black;
  border-radius: 5px;
  width: 550px;
  padding-top: 20px;
  padding-left: 20px;
  padding-right: 20px;
  padding-bottom: 20px;
  margin-top: 20px!important;
  margin:0 auto;
  background-color: white;
}
```

Εικόνα 263. Σελίδα css προεπισκόπησης εισιτηρίου

```
.top-section {
  display: flex;
  flex-flow: row;
  flex-grow: 1;
  background-color: #8fa3cc;
}

.user-details-section {
  display: flex;
  flex-flow: row;
  flex-grow: 1;
  justify-content: flex-end;
  align-items: center;
  padding-right: 10px;
}

.username {
  color: white;
  font-weight: bold;
  margin-right: 20px;
  font-size: 18px;
}

.logout{
  width: 32px;
  cursor: pointer;
}
```

Εικόνα 264. Σελίδα css πάνω πανό στοιχείων χρήστη & εξόδου του

```
.field{
  display: flex;
  margin-bottom: 10px;
}

.lbl{
  width: 185px;
  text-align: right;
  margin-right: 10px;
}

.cmb {
  width: 450px;
  height: 31px !important;
  padding: .2rem !important;
}

.buttons{
  display: flex;
  justify-content: flex-end;
  margin-right: 20px;
  margin-bottom: 20px;
  margin-top: 20px;
}

.panel{
  padding: 10px 10px 10px 10px;
  border-bottom: 1px solid black;
}

.panel-noborder{
  padding: 10px 10px 10px 10px;
}

.home{
  cursor: pointer;
}
```

Εικόνα 265. Σελίδα css επεξεργασίας μεταφοράς

```
body{
  padding-left:10px;
  padding-right:10px;
  padding-top: 10px;
  padding-bottom: 10px;
}

.tbl-search
{
  border: 1px solid black;
}

.tbl-search td
{
  border: 1px solid black;
  padding-left: 10px;
  padding-right: 10px;
}

.tbl-search th
{
  border: 1px solid black;
  padding-left: 10px;
  padding-right: 10px;
  background-color: blue;
  color:white;
  text-align: center;
}

.td-centered
{
  text-align: center;
}

a:hover
{
  text-decoration: none;
}
```

Εικόνα 266. Σελίδα css αναζήτησης μεταφοράς

```
.field{
  display: flex;
  margin-bottom: 10px;
}

.lbl{
  width: 150px;
  text-align: right;
  margin-right: 10px;
}

.lbl-214{
  width:214px!important;
}

.txt{
  width: 250px;
}

#txtAddressNo{
  width: 150px;
}

.buttons{
  display: flex;
  justify-content: flex-end;
  margin-right: 20px;
  margin-bottom: 20px;
  margin-top: 20px;
}

.home{
  cursor: pointer;
}
```

Εικόνα 267. Σελίδα css επεξεργασίας χρήστη-διαχειριστή

```
body{
  padding-left:10px;
  padding-right:10px;
  padding-top: 10px;
  padding-bottom: 10px;
}

.tbl-search
{
  border: 1px solid black;
}

.tbl-search td
{
  border: 1px solid black;
  padding-left: 10px;
  padding-right: 10px;
}

.tbl-search th
{
  border: 1px solid black;
  padding-left: 10px;
  padding-right: 10px;
  background-color: blue;
  color:white;
  text-align: center;
}

.td-centered
{
  text-align: center;
}

a:hover
{
  text-decoration: none;
}
```

Εικόνα 268. Σελίδα css αναζήτησης χρήστη-διαχειριστή

```
.field{
  display: flex;
  margin-bottom: 10px;
}

.lbl{
  width: 159px;
  text-align: right;
  margin-right: 10px;
}

.lbl-214{
  width:214px!important;
}

.txt{
  width: 250px;
}

#txtAddressNo{
  width: 150px;
}

.buttons{
  display: flex;
  justify-content: flex-end;
  margin-right: 10px;
  margin-bottom: 20px;
  margin-top: 20px;
}

.home{
  cursor: pointer;
}
```

Εικόνα 269. Σελίδα css επεξεργασίας χρήστη-ταξιδιωτικού πράκτορα

```
body{
  padding-left:10px;
  padding-right:10px;
  padding-top: 10px;
  padding-bottom: 10px;
}

.tbl-search
{
  border: 1px solid black;
}

.tbl-search td
{
  border: 1px solid black;
  padding-left: 10px;
  padding-right: 10px;
}

.tbl-search th
{
  border: 1px solid black;
  padding-left: 10px;
  padding-right: 10px;
  background-color: blue;
  color:white;
  text-align: center;
}

.td-centered
{
  text-align: center;
}

a:hover
{
  text-decoration: none;
}
```

Εικόνα 270. Σελίδα css αναζήτησης χρήστη-ταξιδιωτικού πράκτορα

```
.field{
  display: flex;
  margin-bottom: 10px;
}

.lbl{
  width: 159px;
  text-align: right;
  margin-right: 10px;
}

.lbl-214{
  width:214px!important;
}

.txt{
  width: 250px;
}

#txtAddressNo{
  width: 150px;
}

.buttons{
  display: flex;
  justify-content: flex-end;
  margin-right: 10px;
  margin-bottom: 20px;
  margin-top: 20px;
}

.home{
  cursor: pointer;
}
```

Εικόνα 271. Σελίδα css επεξεργασίας χρήστη-πελάτη

```
body{
  padding-left:10px;
  padding-right:10px;
  padding-top: 10px;
  padding-bottom: 10px;
}

.tbl-search
{
  border: 1px solid black;
}

.tbl-search td
{
  border: 1px solid black;
  padding-left: 10px;
  padding-right: 10px;
}

.tbl-search th
{
  border: 1px solid black;
  padding-left: 10px;
  padding-right: 10px;
  background-color: blue;
  color:white;
  text-align: center;
}

.td-centered
{
  text-align: center;
}

a:hover
{
  text-decoration: none;
}
```

Εικόνα 272. Σελίδα css αναζήτησης χρήστη-πελάτη

```
.main-section{
  display:flex;
  flex-flow: row;
}

.left-section
{
  height: 100vh;
  background-color: #483D8B;
}

.tab
{
  background-color: #483D8B;
  width: 150px;
  padding-top: 30px;
  padding-bottom: 30px;
  text-align:center;
  color:white;
  font-weight: bold;
}

.tab:hover
{
  cursor:pointer;
  background-color:#6e7d9c;
}

.right-section{
  flex-grow:1;
  padding-left:10px;
  padding-right:10px;
  padding-top: 10px;
  padding-bottom: 10px;
}

#frmUsers
{
  height: calc(100vh - 90px);
  border: none;
  width: 100%;
}
```

Εικόνα 273. Σελίδα css χρηστών

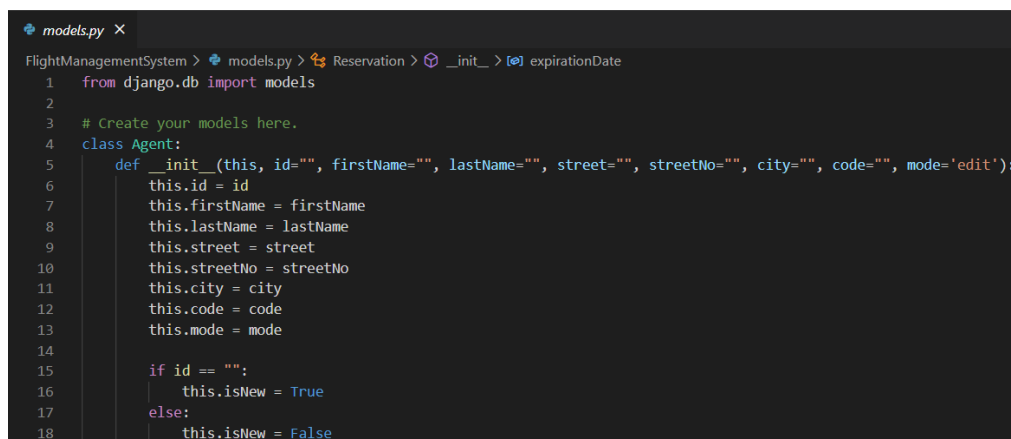

```
.main-container
{
  padding: 0;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100%;
}

.user
{
  color: blue;
  font-weight: bold;
}

body,html
{
  height: 100%;
  padding: 0;
  background-size: cover;
}
```

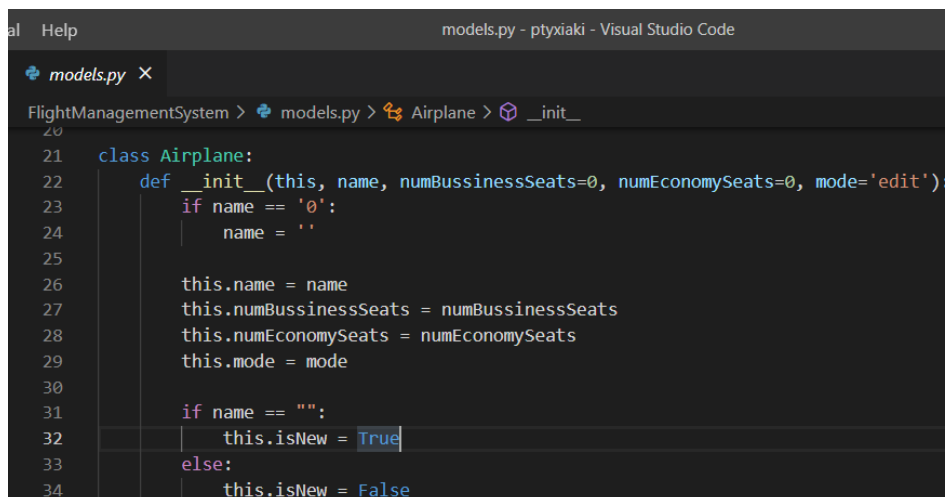
Εικόνα 274. Σελίδα css καλωσορίσματος χρήστη

Δημιουργία models



```
models.py X
FlightManagementSystem > models.py > Reservation > __init__ > expirationDate
1 from django.db import models
2
3 # Create your models here.
4 class Agent:
5     def __init__(this, id="", firstName="", lastName="", street="", streetNo="", city="", code="", mode='edit'):
6         this.id = id
7         this.firstName = firstName
8         this.lastName = lastName
9         this.street = street
10        this.streetNo = streetNo
11        this.city = city
12        this.code = code
13        this.mode = mode
14
15        if id == "":
16            this.isNew = True
17        else:
18            this.isNew = False
```

Εικόνα 275. Μοντέλο ταξιδιωτικού πράκτορα



```
models.py X
FlightManagementSystem > models.py > Airplane > __init__
20
21 class Airplane:
22     def __init__(this, name, numBussinessSeats=0, numEconomySeats=0, mode='edit'):
23         if name == '0':
24             name = ''
25
26         this.name = name
27         this.numBussinessSeats = numBussinessSeats
28         this.numEconomySeats = numEconomySeats
29         this.mode = mode
30
31         if name == "":
32             this.isNew = True
33         else:
34             this.isNew = False
```

Εικόνα 276. Μοντέλο αεροπλάνου

```
al Help models.py - ptyxiaki - Visual Studio Code
models.py x
FlightManagementSystem > models.py > Airplane > __init__
36
37 class Client:
38     def __init__(this, phoneNo="", eMail="", firstName="", lastName="", street="", streetNo="", city="", mode='edit'):
39         this.phoneNo = phoneNo
40         this.eMail = eMail
41         this.firstName = firstName
42         this.lastName = lastName
43         this.street = street
44         this.streetNo = streetNo
45         this.city = city
46         this.mode = mode
47
48         if phoneNo == "":
49             this.isNew = True
50         else:
51             this.isNew = False
```

Εικόνα 277. Μοντέλο πελάτη

```
al Help models.py - ptyxiaki - Visual Studio Code
models.py x
FlightManagementSystem > models.py > Airplane > __init__
54 class Employee:
55     def __init__(this, id="", firstName="", lastName="", street="", streetNo="", city="", phoneNo="", birthDate='', mode='edit'):
56         this.id = id
57         this.firstName = firstName
58         this.lastName = lastName
59         this.street = street
60         this.streetNo = streetNo
61         this.city = city
62         this.phoneNo = phoneNo
63         this.birthDate = birthDate
64         this.mode = mode
65
66         if id == "":
67             this.isNew = True
68         else:
69             this.isNew = False
```

Εικόνα 278. Μοντέλο υπαλλήλου

```
al Help models.py - ptyxiaki - Visual Studio Code
models.py x
FlightManagementSystem > models.py > Airplane > __init__
72 class Pilot(Employee):
73     def __init__(this, id="", firstName="", lastName="", street="", streetNo="", city="", phoneNo="", birthDate="", flightHours=0, mode="edit"):
74         Employee.__init__(this, id, firstName, lastName,
75                             street, streetNo, city, phoneNo, birthDate)
76         this.flightHours = flightHours
77         this.mode = mode
78
79         if id == "":
80             this.isNew = True
81         else:
82             this.isNew = False
```

Εικόνα 279. Μοντέλο πιλότου

```
al Help models.py - ptyxiaki - Visual Studio Code
models.py x
FlightManagementSystem > models.py > Airplane > __init__
85 class AirHostess(Employee):
86     def __init__(this, id="", firstName="", lastName="", street="", streetNo="", city="", phoneNo="", birthDate="", years=0, mode="edit"):
87         Employee.__init__(this, id, firstName, lastName,
88                             street, streetNo, city, phoneNo, birthDate)
89         this.years = years
90         this.mode = mode
91
92         if id == "":
93             this.isNew = True
94         else:
95             this.isNew = False
```

Εικόνα 280. Μοντέλο αεροσυνοδού

```
models.py X
FlightManagementSystem > models.py > Airplane > __init__
98 class Manager(Employee):
99     def __init__(this, id="", firstName="", lastName="", street="", streetNo="", city="", phoneNo="", birthDate="", education="", years=0, mode="edit"):
100         Employee.__init__(this, id, firstName, lastName,
101                             street, streetNo, city, phoneNo, birthDate)
102         this.education = education
103         this.years = years
104         this.mode = mode
105
106         if id == "":
107             this.isNew = True
108         else:
109             this.isNew = False
```

Εικόνα 281. Μοντέλο διοικητικού υπαλλήλου

```
models.py X
FlightManagementSystem > models.py > Airplane > __init__
112 class UserAgent(Agent):
113     def __init__(this, username="", id="", firstName="", lastName="", street="", streetNo="", city="", code="", mode="edit"):
114         Agent.__init__(this, id, firstName, lastName,
115                         street, streetNo, city, code, mode)
116         this.username = username
117         this.mode = mode
```

Εικόνα 282. Μοντέλο χρήστη-ταξιδιωτικού πράκτορα

```
models.py X
FlightManagementSystem > models.py > Airplane > __init__
120 class UserClient(Client):
121     def __init__(this, username="", phoneNo="", eMail="", firstName="", lastName="", street="", streetNo="", city="", mode="edit"):
122         Client.__init__(this, phoneNo, eMail, firstName,
123                         lastName, street, streetNo, city, mode)
124         this.username = username
125         this.mode = mode
```

Εικόνα 283. Μοντέλο χρήστη-πελάτη

```
models.py X
FlightManagementSystem > models.py > Airplane > __init__
128 class UserAdministrator(Employee):
129     def __init__(this, username="", id="", firstName="", lastName="", street="", streetNo="", city="", phoneNo="", birthDate="", mode="edit"):
130         Employee.__init__(this, id, firstName, lastName,
131                             street, streetNo, city, phoneNo, birthDate)
132         this.username = username
133         this.mode = mode
```

Εικόνα 284. Μοντέλο χρήστη-διαχειριστή

```
models.py X
FlightManagementSystem > models.py > Airplane > __init__
136 class User:
137     def __init__(this, username="", firstName="", lastName="", isAgent=False, isClient=False, isAdministrator=False, clientID="", agentID="", administratorID=""):
138         this.username = username
139         this.firstName = firstName
140         this.lastName = lastName
141         this.isAgent = isAgent
142         this.isClient = isClient
143         this.isAdministrator = isAdministrator
144         this.clientID = clientID
145         this.agentID = agentID
146         this.administratorID = administratorID
```

Εικόνα 285. Μοντέλο χρηστών

```
models.py x
FlightManagementSystem > models.py > Airplane > _init_
149 class Route:
150     def __init__(this, id="", depPointID=0, destPointID=0, dayID=0, distance=0, depTime=None, arvTime=None, busPrice=0, ecoPrice=0, mode='edit'):
151         this.id = id
152         this.depPointID = depPointID
153         this.destPointID = destPointID
154         this.dayID = dayID
155         this.distance = distance
156
157         if depTime != None:
158             this.depTime = str(depTime)
159         else:
160             this.depTime = depTime
161
162         if arvTime != None:
163             this.arvTime = str(arvTime)
164         else:
165             this.arvTime = arvTime
166
167         this.busPrice = busPrice
168         this.ecoPrice = ecoPrice
169         this.mode = mode
```

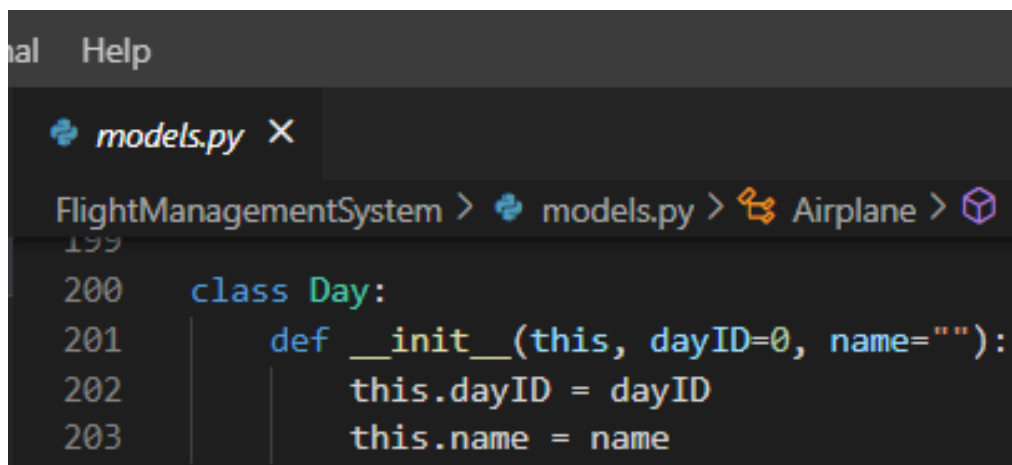
Εικόνα 286. Μοντέλο δρομολογίου

```
models.py x
FlightManagementSystem > models.py > Airplane > _init_
172 class RouteRow:
173     def __init__(this, id="", depPoint="", destPoint="", day="", distance=0, depTime=None, arvTime=None, busPrice=0, ecoPrice=0):
174         this.id = id
175         this.depPoint = depPoint
176         this.destPoint = destPoint
177         this.day = day
178         this.distance = distance
179
180         if depTime != None:
181             this.depTime = str(depTime)
182         else:
183             this.depTime = depTime
184
185         if arvTime != None:
186             this.arvTime = str(arvTime)
187         else:
188             this.arvTime = arvTime
189
190         this.busPrice = busPrice
191         this.ecoPrice = ecoPrice
```

Εικόνα 287. Μοντέλο γραμμής δρομολογίου

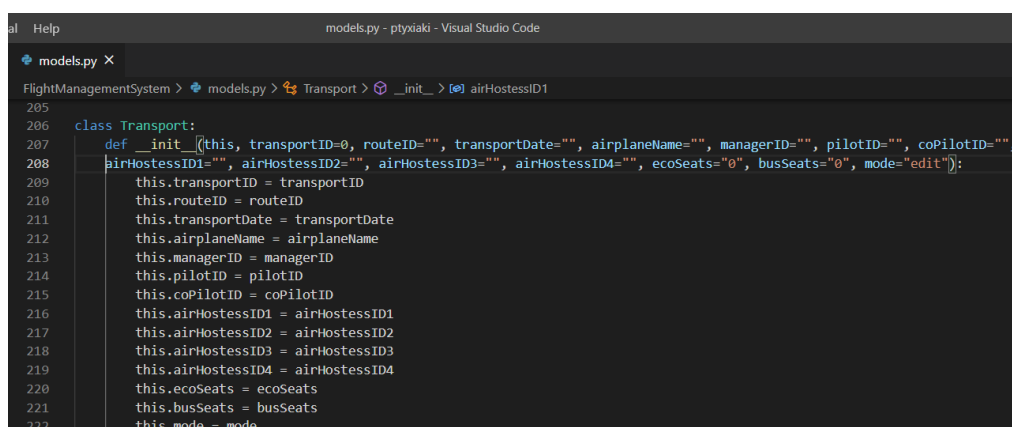
```
models.py x
FlightManagementSystem > models.py > Airplane > _
193
194 class City:
195     def __init__(this, cityID=0, name=""):
196         this.cityID = cityID
197         this.name = name
```

Εικόνα 288. Μοντέλο πόλης



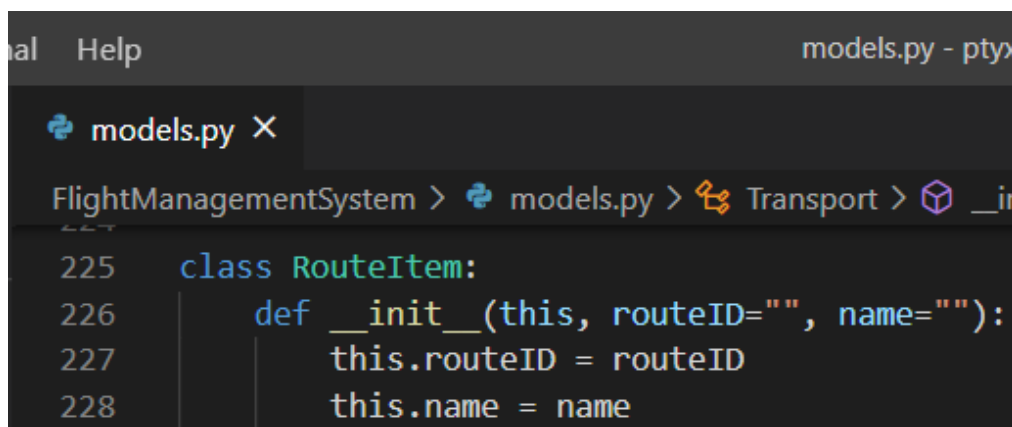
```
FlightManagementSystem > models.py > Airplane >
200 class Day:
201     def __init__(this, dayID=0, name=""):
202         this.dayID = dayID
203         this.name = name
```

Εικόνα 289. Μοντέλο μέρας



```
models.py - ptyxiaki - Visual Studio Code
models.py X
FlightManagementSystem > models.py > Transport > __init__ > airHostessID1
205
206 class Transport:
207     def __init__(this, transportID=0, routeID="", transportDate="", airplaneName="", managerID="", pilotID="", coPilotID="",
208     airHostessID1="", airHostessID2="", airHostessID3="", airHostessID4="", ecoSeats="0", busSeats="0", mode="edit"):
209         this.transportID = transportID
210         this.routeID = routeID
211         this.transportDate = transportDate
212         this.airplaneName = airplaneName
213         this.managerID = managerID
214         this.pilotID = pilotID
215         this.coPilotID = coPilotID
216         this.airHostessID1 = airHostessID1
217         this.airHostessID2 = airHostessID2
218         this.airHostessID3 = airHostessID3
219         this.airHostessID4 = airHostessID4
220         this.ecoSeats = ecoSeats
221         this.busSeats = busSeats
222         this.mode = mode
```

Εικόνα 290. Μοντέλο μεταφοράς



```
models.py - ptyxiaki
models.py X
FlightManagementSystem > models.py > Transport > __in
225 class RouteItem:
226     def __init__(this, routeID="", name=""):
227         this.routeID = routeID
228         this.name = name
```

Εικόνα 291. Μοντέλο αντικειμένου δρομολογίου

```
al Help models.py - ptyx
models.py X
FlightManagementSystem > models.py > Transport > _in
231 class PilotItem:
232     def __init__(this, pilotID="", name=""):
233         this.pilotID = pilotID
234         this.name = name
```

Εικόνα 292. Μοντέλο αντικειμένου πιλότου

```
al Help models.py - ptyxiaki - Vi
models.py X
FlightManagementSystem > models.py > Transport > _init_ > D
237 class AirhostessItem:
238     def __init__(this, airHostessID="", name=""):
239         this.airHostessID = airHostessID
240         this.name = name
```

Εικόνα 293. Μοντέλο αντικειμένου αεροσυνοδού

```
al Help models.py - ptyxiaki
models.py X
FlightManagementSystem > models.py > Transport > _init_
243 class ManagerItem:
244     def __init__(this, managerID="", name=""):
245         this.managerID = managerID
246         this.name = name
```

Εικόνα 294. Μοντέλο αντικειμένου διοικητικού υπαλλήλου

```
al Help
models.py X
FlightManagementSystem > models.py > Tra
249 class AirplaneItem:
250     def __init__(this, name=""):
251         this.name = name
```

Εικόνα 295. Μοντέλο αντικειμένου αεροπλάνου

```
al Help models.py - ptyxiaki - Visual Studio Code
models.py X
FlightManagementSystem > models.py > TransportRow > _init_ > airHostess1
254 class TransportRow:
255     def __init__(this, transportID="", route="", transportDate="", airplaneName="", pilot="", coPilot="", manager="",
256     airHostess1="", airHostess2="", airHostess3="", airHostess4=""):
257         this.transportID = transportID
258         this.route = route
259         this.transportDate = transportDate
260         this.airplaneName = airplaneName
261         this.pilot = pilot
262         this.coPilot = coPilot
263         this.manager = manager
264         this.airHostess1 = airHostess1
265         this.airHostess2 = airHostess2
266         this.airHostess3 = airHostess3
267         this.airHostess4 = airHostess4
```

Εικόνα 296. Μοντέλο γραμμής μεταφοράς

```
al Help models.py - ptyxiaki - Visual Studio Code
models.py
FlightManagementSystem > models.py > Reservation > _init_ > departureCityID
270 class Reservation:
271     def __init__(this, reservationID="", phoneNo="", reservationDate="", expirationDate="", bussinessClass=False, agentID="",
272     departureCityID="", arrivalCityID="", transportID1="", transportID2="", mode="edit"):
273         this.reservationID = reservationID
274         this.phoneNo = phoneNo
275         this.reservationDate = reservationDate
276         this.expirationDate = expirationDate
277         this.bussinessClass = bussinessClass
278         this.agentID = agentID
279         this.departureCityID = departureCityID
280         this.arrivalCityID = arrivalCityID
281         this.transportID1 = transportID1
282         this.transportID2 = transportID2
283         this.mode = mode
```

Εικόνα 297. Μοντέλο κράτησης

```

models.py ×
FlightManagementSystem > models.py > Reservation > __init__ > [
285
286 class ClientItem:
287     def __init__(this, phoneNo="", displayName=""):
288         this.phoneNo = phoneNo
289         this.displayName = displayName
    
```

Εικόνα 298. Μοντέλο αντικειμένου πελάτη

```

models.py ×
FlightManagementSystem > models.py > TransportItem > toJson
292 class TransportItem:
293     def __init__(this, transportID, depCity, arvCity, depTime, arvTime, airplaneName, ecoPrice, busPrice, ecoSeats, busSeats):
294         this.transportID = transportID
295         this.depCity = depCity
296         this.arvCity = arvCity
297         this.depTime = depTime
298         this.arvTime = arvTime
299         this.airplaneName = airplaneName
300         this.ecoPrice = ecoPrice
301         this.busPrice = busPrice
302         this.ecoSeats = ecoSeats
303         this.busSeats = busSeats
304
305     def toJson(this):
306         return '{"transportID":' + str(this.transportID) + ', "depCity":' + this.depCity + ', "arvCity":' + this.arvCity + ',
307             "depTime":' + str(this.depTime) + ', "arvTime":' + str(this.arvTime) + ', "airplaneName":' + this.airplaneName + ',
308             "ecoPrice":' + str(this.ecoPrice) + ', "busPrice":' + str(this.busPrice) + ', "ecoSeats":' + str(this.ecoSeats) + ',
309             "busSeats":' + str(this.busSeats) + '}'
    
```

Εικόνα 299. Μοντέλο αντικειμένου μεταφοράς

```

models.py ×
FlightManagementSystem > models.py > TransportItem > toJson
312 class ReservationRow:
313     def __init__(this, reservationID, client, resdate, expDate, seatCategory, agent, depCity, arvCity, price):
314         this.reservationID = reservationID
315         this.client = client
316         this.resDate = resdate
317         this.expDate = expDate
318         this.seatCategory = seatCategory
319         this.agent = agent
320         this.depCity = depCity
321         this.arvCity = arvCity
322         this.price = price
    
```

Εικόνα 300. Μοντέλο γραμμής κράτησης

```

models.py ×
FlightManagementSystem > models.py > TransportItem > toJson
325 class Ticket:
326     def __init__(this, ticketID, fullName, resDate, depCity, arvCity, busClass, price, kilometers):
327         this.ticketID = ticketID
328         this.fullName = fullName
329         this.resDate = resDate
330         this.depCity = depCity
331         this.arvCity = arvCity
332         this.busClass = busClass
333         this.price = price
334         this.kilometers = kilometers
    
```

Εικόνα 301. Μοντέλο εισιτηρίου


```

models.py ×
FlightManagementSystem > models.py > TransportItem > toJson
337 class ClientInfo:
338     def __init__(this, phoneNo, eMail, fullName, street, city, totalKm, username):
339         this.phoneNo = phoneNo
340         this.eMail = eMail
341         this.fullName = fullName
342         this.street = street
343         this.city = city
344         this.totalKm = totalKm
345         this.username = username
    
```

Εικόνα 302. Μοντέλο πληροφοριών πελάτη

```

models.py ×
FlightManagementSystem > models.py > Occupancy > __init__
348 class Occupancy:
349     def __init__(this, routeID, depCity, arvCity, tickets):
350         this.routeID = routeID
351         this.depCity = depCity
352         this.arvCity = arvCity
353         this.tickets = tickets
354
355     def toJson(this):
356         return '{"routeID":' + str(this.routeID) + ', "depCity":"' + this.depCity + ', "arvCity":"' + this.arvCity + ', "tickets":'
357         + str(this.tickets) + '}'
    
```

Εικόνα 303. Μοντέλο πληρότητας πτήσεων

Δημιουργία URLs

```

urls.py ×
Ptyxiaki > urls.py > ...
1 from django.contrib import admin
2 from django.urls import path
3 from FlightManagementSystem import views
4
5 urlpatterns = []
6     path('admin/', admin.site.urls),
7
8     path('agentEdit/<str:id>/<str:mode>', views.agentEdit),
9     path('saveAgent', views.saveAgent),
10    path('', views.login),
11    path('deleteAgent/<str:id>', views.deleteAgent),
12    path('validateAgent/<str:id>/<str:code>/<str:prevID>/<str:prevCode>', views.validateAgent),
    
```

Εικόνα 304. URLs εισόδου διαχειριστή & επεξεργασίας ταξιδιωτικού πράκτορα

```

urls.py ×
Ptyxiaki > urls.py > ...
13
14    path('airplaneEdit/<str:name>/<str:mode>', views.airplaneEdit),
15    path('saveAirplane', views.saveAirplane),
16    path('deleteAirplane/<str:name>', views.deleteAirplane),
17    path('validateAirplane/<str:name>/<str:prevName>', views.validateAirplane),
    
```

Εικόνα 305. URLs επεξεργασίας αεροπλάνου

```

al Help urls.py - ptyxiaki - Visual Studio Code
urls.py X
Ptyxiaki > urls.py > ...
18
19 path('clientEdit/<str:phoneNo>/<str:mode>',views.clientEdit),
20 path('saveClient', views.saveClient),
21 path('deleteClient/<str:phoneNo>',views.deleteClient),
22 path('validateClient/<str:phoneNo>/<str:prevPhoneNo>', views.validateClient),
    
```

Εικόνα 306. URLs επεξεργασίας πελάτη

```

al Help urls.py - ptyxiaki - Visual Studio Code
urls.py X
Ptyxiaki > urls.py > ...
24 path('pilotEdit/<str:id>/<str:mode>',views.pilotEdit),
25 path('savePilot',views.savePilot),
26 path('deletePilot/<str:id>',views.deletePilot),
27 path('validatePilot/<str:id>/<str:prevID>',views.validatePilot),
    
```

Εικόνα 307. URLs επεξεργασίας πιλότου

```

al Help urls.py - ptyxiaki - Visual Studio Code
urls.py X
Ptyxiaki > urls.py > ...
29 path('airHostessEdit/<str:id>/<str:mode>',views.airHostessEdit),
30 path('saveAirHostess', views.saveAirHostess),
31 path('deleteAirHostess/<str:id>',views.deleteAirHostess),
32 path('validateAirHostess/<str:id>/<str:prevID>', views.validateAirHostess),
    
```

Εικόνα 308. URLs επεξεργασίας αεροσυνοδού

```

al Help urls.py - ptyxiaki - Visual Studio Code
urls.py X
Ptyxiaki > urls.py > ...
34 path('managerEdit/<str:id>/<str:mode>',views.managerEdit),
35 path('saveManager', views.saveManager),
36 path('deleteManager/<str:id>',views.deleteManager),
37 path('validateManager/<str:id>/<str:prevID>', views.validateManager),
    
```

Εικόνα 309. URLs επεξεργασίας διοικητικού υπαλλήλου

```

al Help urls.py - ptyxiaki - Visual Studio Code
urls.py X
Ptyxiaki > urls.py > ...
39 path('routeEdit/<str:id>/<str:mode>',views.routeEdit),
40 path('saveRoute',views.saveRoute),
41 path('deleteRoute/<str:id>',views.deleteRoute),
42 path('validateRoute/<str:id>/<str:prevID>',views.validateRoute),
    
```

Εικόνα 310. URLs επεξεργασίας δρομολογίου

```
al Help urls.py - ptyxiaki - Visual Studio Code
urls.py X
Ptyxiaki > urls.py > ...
44 path('userClientEdit/<str:username>/<str:mode>', views.userClientEdit),
45 path('saveUserClient', views.saveUserClient),
46 path('validateUserClient/<str:username>/<str:phoneNo>/<str:prevUsername>/<str:prevPhoneNo>', views.validateUserClient),
```

Εικόνα 311. URLs επεξεργασίας χρήστη-πελάτη

```
al Help urls.py - ptyxiaki - Visual Studio Code
urls.py X
Ptyxiaki > urls.py > ...
48 path('userAgentEdit/<str:username>/<str:mode>', views.userAgentEdit),
49 path('saveUserAgent', views.saveUserAgent),
50 path('validateUserAgent/<str:username>/<str:code>/<str:id>/<str:previousUsername>/<str:previousCode>/<str:previousID>',
51 views.validateUserAgent),
```

Εικόνα 312. URLs επεξεργασίας χρήστη-ταξιδιωτικού πράκτορα

```
al Help urls.py - ptyxiaki - Visual Studio Code
urls.py X
Ptyxiaki > urls.py > ...
53 path('userAdministratorEdit/<str:username>/<str:mode>', views.userAdministratorEdit),
54 path('saveUserAdministrator', views.saveUserAdministrator),
55 path('validateUserAdministrator/<str:username>/<str:id>/<str:previousUsername>/<str:previousID>',
56 views.validateUserAdministrator),
```

Εικόνα 313. URLs επεξεργασίας χρήστη-διαχειριστή

```
al Help urls.py - ptyxiaki - Visual Studio Code
urls.py X
Ptyxiaki > urls.py > ...
58 path('transportEdit/<int:id>/<str:mode>', views.transportEdit),
59 path('validateTransport/<int:transportID>/<str:routeID>/<str:pilotID>/<str:coPilotID>/<str:transportDate>/<str:airHostessID1>/
60 <str:airHostessID2>/<str:airHostessID3>/<str:airHostessID4>/<str:airplaneName>', views.validateTransport),
61 path('deleteTransport/<int:id>', views.deleteTransport),]
62 path('saveTransport', views.saveTransport),
63 path('getTransports', views.getTransports),
64 path('getAirplaneSeats', views.getAirplaneSeats),
```

Εικόνα 314. URLs επεξεργασίας μεταφοράς

```
al Help urls.py - ptyxiaki - Visual Studio Code
urls.py X
Ptyxiaki > urls.py > ...
65 path('reservationEdit/<int:id>/<str:mode>', views.reservationEdit),
66 path('saveReservation', views.saveReservation),
67 path('deleteReservation/<int:id>', views.deleteReservation),
68 path('validateReservation', views.validateReservation),
```

Εικόνα 315. URLs επεξεργασίας κράτησης ταξιδιωτικού πράκτορα

```
al Help urls.py - ptyxiaki - Visual Studio Code
urls.py X
Ptyxiaki > urls.py > ...
70 path('clientReservationEdit/<int:id>/<str:mode>', views.clientReservationEdit),
71 path('saveClientReservation', views.saveClientReservation),
```

Εικόνα 316. URLs επεξεργασίας κράτησης πελάτη

```

al Help urls.py - ptyxiaki - Visual Studio Code
urls.py X
Ptyxiaki > urls.py > ...
73 path('deleteUser/<str:username>',views.deleteUser),
74 path('validateUser/<str:username>',views.validateUser),
75 path('users',views.users),
76 path('signIn',views.signIn),
77 path('logout',views.logout),
78 path('home',views.home),
79 path('changePassword',views.changePassword),
80 path('welcome', views.welcome),|
    
```

Εικόνα 317. URLs επεξεργασίας & εισόδου χρήστη

```

al Help urls.py - ptyxiaki - Visual Studio Code
urls.py X
Ptyxiaki > urls.py > ...
82 path('pilotSearch',views.pilotSearch),
83 path('agentSearch',views.agentSearch),
84 path('airHostessSearch',views.airHostessSearch),
85 path('airplaneSearch',views.airplaneSearch),
86 path('clientSearch',views.clientSearch),
87 path('managerSearch',views.managerSearch),
88 path('userAgentSearch',views.userAgentSearch),
89 path('userClientSearch',views.userClientSearch),
90 path('userAdministratorSearch',views.userAdministratorSearch),
91 path('routeSearch',views.routeSearch),
92 path('transportSearch',views.transportSearch),
93 path('reservationSearch',views.reservationSearch),
94 path('clientReservationSearch',views.clientReservationSearch),
    
```

Εικόνα 318. URLs αναζήτησης των εγγραφών

```

al Help urls.py - ptyxiaki - Visual Studio Code
urls.py X
Ptyxiaki > urls.py > ...
96 path('printTicket',views.printTicket),
97 path('showTicket/<int:ticketID>',views.showTicket),
98 path('showClientTicket/<int:ticketID>',views.showClientTicket),
    
```

Εικόνα 319. URLs εισιτηρίου

```

al Help urls.py - Ptyxiaki - Visual Studio Code
urls.py
Ptyxiaki > urls.py > ...
100 path('clientDashboard', views.clientDashboard),
    
```

Εικόνα 320. URL ταμπλό πελάτη

```

al Help urls.py - Ptyxiaki - Visual Studio Code
urls.py
Ptyxiaki > urls.py > ...
102 path('report', views.report),
103 path('getReport', views.getReport)
104
    
```

Εικόνα 321. URLs συγκεντρωτικής αναφοράς πτήσεων

Δημιουργία views

```

from django.shortcuts import render
from django.views.decorators.csrf import csrf_exempt
from django.db import connection
import json
from django.http import JsonResponse
from .models import Airplane
from .models import Agent
from .models import Client
from .models import Pilot
from .models import AirHostess
from .models import Manager
from .models import UserAgent
from .models import UserClient
from .models import UserAdministrator
from .models import User
from .models import Route
from .models import RouteRow
from .models import RouteItem
#Κάθε αντικείμενο της τάξης RouteItem χρησιμοποιείται για τη δημιουργία μιας επιλογής (option) στο combobox route της σελίδας transport
from .models import City
from .models import Day
from .models import Transport
from .models import TransportRow
from .models import PilotItem
from .models import AirHostessItem
from .models import ManagerItem
from .models import AirplaneItem
from .models import Reservation
from .models import ClientItem
from .models import TransportItem
from .models import ReservationRow
from .models import Ticket
from .models import ClientInfo
from .models import Occupancy
import datetime
import hashlib
from django.shortcuts import redirect
from django.core.mail import send_mail
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
import ssl
    
```

Εικόνα 322. Εισαγωγή βιβλιοθηκών της Django, JSON, Email & μοντέλων εγγραφών

```
al Help views.py - ptyxiaki - Visual Studio Code
views.py x
FlightManagementSystem > views.py > changePassword
44 # Create your views here.
45
46 # -----APXIKΑ-----
47
48 def login(request):
49     return render(request, 'FlightManagementSystem/Login.html', {})
```

Εικόνα 323. View αρχικής σελίδας εισόδου

```
al Help views.py - ptyxiaki - Visual Studio Code
views.py x
FlightManagementSystem > views.py > ...
51 def encodePassword(password):
52     hashedPassword = hashlib.sha256(str.encode(password)).hexdigest()
53     return hashedPassword
```

Εικόνα 324. View κρυπτογράφηση κωδικού

```
al Help views.py - ptyxiaki - V
views.py x
FlightManagementSystem > views.py > ...
55 def checkPassword(password, dbHashedPassword):
56     hashedPassword = encodePassword(password)
57
58     if hashedPassword == dbHashedPassword:
59         return True
60     else:
61         return False
```

Εικόνα 325. View ελέγχου εγκυρότητας κωδικού

```
@csrf_exempt
def signIn(request):
    username = request.POST['username']
    password = request.POST['password']

    with connection.cursor() as cursor:
        cursor.execute('exec dbo.SignIn @Username=%s', [username])

    row = cursor.fetchone()
    if not row is None: #ελέγχουμε αν βρέθηκε εγγραφή στον πίνακα user με το username που έδωσε ο χρήστης
        dbPassword = row[0]
        valid = checkPassword(password,dbPassword)
    else:
        valid = False

    if valid == True:
        firstName = row[1]
        lastName = row[2]
        clientID = row[3]
        agentID = row[4]
        administratorID = row[5]

        isClient = False
        if not clientID is None:
            isClient = True

        isAgent = False
        if not agentID is None:
            isAgent = True

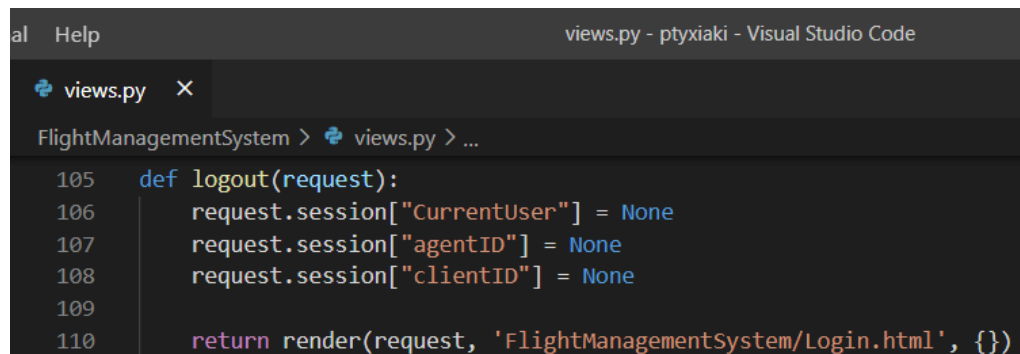
        isAdmin = False
        if not administratorID is None:
            isAdmin = True

        currentUser = User(username, firstName, lastName, isAgent, isClient, isAdmin, clientID, agentID, administratorID)

        request.session["CurrentUser"] = json.dumps(currentUser.__dict__)
        request.session["agentID"] = agentID
        request.session["clientID"] = clientID

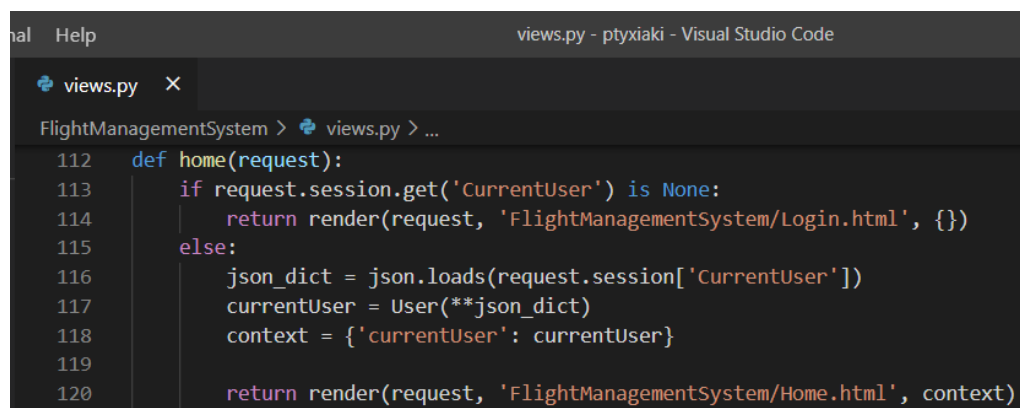
    return JsonResponse(json.dumps(valid),safe=False)
```

Εικόνα 326. View ελέγχου εγκυρότητας στοιχείων χρήστη



```
105 def logout(request):
106     request.session["CurrentUser"] = None
107     request.session["agentID"] = None
108     request.session["clientID"] = None
109
110     return render(request, 'FlightManagementSystem/Login.html', {})
```

Εικόνα 327. View αποσύνδεσης χρήστη



```
112 def home(request):
113     if request.session.get('CurrentUser') is None:
114         return render(request, 'FlightManagementSystem/Login.html', {})
115     else:
116         json_dict = json.loads(request.session['CurrentUser'])
117         currentUser = User(**json_dict)
118         context = {'currentUser': currentUser}
119
120     return render(request, 'FlightManagementSystem/Home.html', context)
```

Εικόνα 328. View αρχικής σελίδας εφαρμογής

```

views.py - ptyxiaki - Visual Studio Code
views.py x
FlightManagementSystem > views.py > changePassword
122 @csrf_exempt
123 def changePassword(request):
124     username = request.POST['username']
125     oldPassword = request.POST['oldPassword']
126     newPassword = request.POST['newPassword']
127
128     # κρυπτογραφώ τον παλιό και νέο κωδικό
129     encodedOldPassword = encodePassword(oldPassword)
130     encodedNewPassword = encodePassword(newPassword)
131
132     with connection.cursor() as cursor:
133         cursor.execute('exec dbo.ChangePassword @UserName=%s, @OldPassword=%s, @NewPassword=%s', [[username, encodedOldPasword,
134             encodedNewPassword]])
135
136         row = cursor.fetchone()
137         counter = row[0]
138
139     return JsonResponse(json.dumps(counter), safe=False)

```

Εικόνα 329. View αλλαγής κωδικού χρήστη

```

views.py - ptyxiaki - Visual Studio Code
views.py x
FlightManagementSystem > views.py > agentEdit
141 # -----AGENT-----
142
143 def agentEdit(request, id, mode):
144     if request.session.get('currentUser') is None:
145         return redirect('/')
146     else:
147         model = Agent()
148
149         if id != '0':
150             with connection.cursor() as cursor:
151                 cursor.execute('exec dbo.AgentSelectByID @ID=%s', [id])
152
153                 row = cursor.fetchone()
154                 model = Agent(id,row[1],row[2],row[3],row[4],row[5],row[6],mode)
155
156         json_dict = json.loads(request.session['currentUser'])
157         currentUser = User(**json_dict)
158
159         context = {'model': model, 'currentUser': currentUser } #Η μεταβλητή context είναι ένα αντικείμενο json, το οποίο θα
160         #χρησιμοποιηθεί για την αρχικοποίηση των controls της ιστοσελίδας
161     return render(request, 'FlightManagementSystem/AgentEdit.html', context)

```

Εικόνα 330. View επεξεργασίας ταξιδιωτικού πράκτορα

```

views.py - ptyxiaki - Visual Studio Code
views.py x
FlightManagementSystem > views.py > agentEdit
163 def agentSearch(request):
164     if request.session.get('currentUser') is None:
165         return redirect('/')
166     else:
167         with connection.cursor() as cursor:
168             cursor.execute('exec dbo.AgentSearch')
169
170             agents = []
171
172             row = cursor.fetchone() # διαβάζω την πρώτη εγγραφή
173
174             while row != None:
175                 agent = Agent(row[0],row[1],row[2],row[3],row[4],row[5],row[6])
176                 agents.append(agent)
177                 row = cursor.fetchone() # διαβάζω την επόμενη εγγραφή
178
179             context = {'model': agents}
180             return render(request, 'FlightManagementSystem/AgentSearch.html', context)

```

Εικόνα 331. View αναζήτησης ταξιδιωτικού πράκτορα


```

182 @csrf_exempt
183 def saveAgent(request):
184     id = request.POST['id']
185     previousID = request.POST['previousID']
186     firstName = request.POST['firstName']
187     lastName = request.POST['lastName']
188     street = request.POST['street']
189     streetNo = request.POST['streetNo']
190     city = request.POST['city']
191     code = request.POST['code']
192     previousCode = request.POST['previousCode']
193
194     if previousID == '0' and previousCode == '0':
195         with connection.cursor() as cursor:
196             cursor.execute('exec dbo.AgentInsert @ID=%s, @FirstName=%s, @LastName=%s, @Street=%s, @StreetNo=%s, @City=%s, @Code=%s',
197                             [id, firstName, lastName, street, streetNo, city, code])
198     else:
199         with connection.cursor() as cursor:
200             cursor.execute('exec dbo.AgentUpdate @PreviousID=%s, @ID=%s, @FirstName=%s, @LastName=%s, @Street=%s, @StreetNo=%s, @City=%s, @Code=%s',
201                             [previousID, id, firstName, lastName, street, streetNo, city, code])
202
203     return JsonResponse(json.dumps(True), safe=False)

```

Εικόνα 332. View αποθήκευσης ταξιδιωτικού πράκτορα

```

205 @csrf_exempt
206 def deleteAgent(request, id):
207
208     with connection.cursor() as cursor:
209         cursor.execute('exec dbo.AgentDeleteById @ID=%s', [id])
210
211     return JsonResponse(json.dumps(True), safe=False)

```

Εικόνα 333. View διαγραφής ταξιδιωτικού πράκτορα

```

views.py - ptyxiaki - Visual Studio Code
views.py x
FlightManagementSystem > views.py > saveAgent

213 @csrf_exempt
214 def validateAgent(request, id, code, prevID, prevCode):
215
216     if id != prevID:
217         query = 'select count(*) from Agent where ID = %s'
218
219         with connection.cursor() as cursor:
220             cursor.execute(query, [id])
221
222             row = cursor.fetchone()
223             count1 = row[0]
224     else:
225         count1 = 0
226
227     if prevCode != code:
228         query = 'select count(*) from Agent where Code = %s'
229
230         with connection.cursor() as cursor:
231             cursor.execute(query, [code])
232
233             row = cursor.fetchone()
234             count2 = row[0]
235     else:
236         count2 = 0
237
238     data = {'idExists': count1 != 0, 'codeExists': count2 != 0}
239
240     return JsonResponse(data, safe = False)

```

Εικόνα 334. View ελέγχου εγκυρότητας ταξιδιωτικού πράκτορα

```

views.py - ptyxiaki - Visual Studio Code
views.py x
FlightManagementSystem > views.py > airplaneEdit

244 def airplaneEdit(request, name, mode):
245     if request.session.get('currentUser') is None:
246         return redirect('/')
247     else:
248         model = Airplane('0')
249
250         if name != '0':
251             with connection.cursor() as cursor:
252                 cursor.execute('exec dbo.AirplaneSelectByName @Name=%s', [name])
253                 row = cursor.fetchone()
254
255                 model = Airplane(name,row[1],row[2],mode)
256
257         json_dict = json.loads(request.session['currentUser'])
258         currentUser = User(**json_dict)
259
260         context = {'model': model, 'currentUser': currentUser} #Η μεταβλητή context είναι ένα αντικείμενο json, το οποίο θα
261         #χρησιμοποιηθεί για την αρχικοποίηση των controls της ιστοσελίδας
262         return render(request, 'FlightManagementSystem/AirplaneEdit.html', context)

```

Εικόνα 335. View επεξεργασίας αεροπλάνου

```

al Help views.py - ptyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > airplaneEdit
264 def airplaneSearch(request):
265     if request.session.get('currentUser') is None:
266         return redirect('/')
267     else:
268         with connection.cursor() as cursor:
269             cursor.execute('exec dbo.AirplaneSearch')
270
271             airplanes = []
272
273             row = cursor.fetchone() # διαβάζω την πρώτη εγγραφή
274
275             while row != None:
276                 airplane = Airplane(row[0],row[1],row[2])
277                 airplanes.append(airplane)
278                 row = cursor.fetchone() # διαβάζω την επόμενη εγγραφή
279
280             context = {'model': airplanes}
281             return render(request, 'FlightManagementSystem/AirplaneSearch.html', context)

```

Εικόνα 336. View αναζήτησης αεροπλάνου

```

al Help views.py - ptyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > ...
283 @csrf_exempt
284 def saveAirplane(request):
285     name = request.POST['name']
286     previousName = request.POST['previousName']
287     numBusinessSeats = request.POST['busSeats']
288     numEconomySeats = request.POST['ecoSeats']
289
290     if previousName == '':
291         with connection.cursor() as cursor:
292             cursor.execute('exec dbo.AirplaneInsert @Name=%s, @NumBusinessSeats=%s, @NumEconomySeats=%s',
293                 [name,numBusinessSeats,numEconomySeats])
294     else:
295         with connection.cursor() as cursor:
296             cursor.execute('exec dbo.AirplaneUpdate @PreviousName=%s, @Name=%s, @NumBusinessSeats=%s, @NumEconomySeats=%s',
297                 [previousName,name,numBusinessSeats,numEconomySeats])
298
299     return JsonResponse(json.dumps(True),safe=False)

```

Εικόνα 337. View αποθήκευσης αεροπλάνου

```

al Help views.py - ptyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > ...
301 @csrf_exempt
302 def deleteAirplane(request, name):
303     try:
304         with connection.cursor() as cursor:
305             cursor.execute('exec dbo.AirplaneDeleteByName @Name=%s',[name])
306             success = True
307     except:
308         success = False
309
310     return JsonResponse(json.dumps(success),safe=False)

```

Εικόνα 338. View διαγραφής αεροπλάνου

```

al Help views.py - ptyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > ...
312 @csrf_exempt
313 def validateAirplane(request, name, prevName):
314
315     if name != prevName:
316         query = 'select count(*) from Airplane where name = %s'
317
318         with connection.cursor() as cursor:
319             cursor.execute(query, [name])
320
321             row = cursor.fetchone()
322             count = row[0]
323     else:
324         count = 0
325
326     data = {'nameExists': count}
327
328     return JsonResponse(data, safe = False)

```

Εικόνα 339. View ελέγχου εγκυρότητας αεροπλάνου

```

al Help views.py - ptyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > ...
330 # -----CLIENT-----
331
332 def clientEdit(request, phoneNo, mode):
333     if request.session.get('currentUser') is None:
334         return redirect('/')
335     else:
336         model = Client()
337
338         if phoneNo != '0':
339             with connection.cursor() as cursor:
340                 cursor.execute('exec dbo.ClientSelectByPhoneNo @PhoneNo=%s', [phoneNo])
341
342                 row = cursor.fetchone()
343                 model = Client(phoneNo,row[1],row[2],row[3],row[4],row[5],row[6],mode)
344
345             json_dict = json.loads(request.session['currentUser'])
346             currentUser = User(**json_dict)
347
348             context = {'model': model, 'currentUser' : currentUser} #Η μεταβλητή context είναι ένα αντικείμενο json, το οποίο θα
349             #χρησιμοποιηθεί για την αρχικοποίηση των controls της ιστοσελίδας
350     return render(request, 'FlightManagementSystem/ClientEdit.html', context)

```

Εικόνα 340. View επεξεργασίας πελάτη

```

al Help views.py - ptyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > ...
352 def clientSearch(request):
353     if request.session.get('CurrentUser') is None:
354         return redirect('/')
355     else:
356         with connection.cursor() as cursor:
357             cursor.execute('exec dbo.ClientSearch')
358
359             clients = []
360
361             row = cursor.fetchone() # διαβάζω την πρώτη εγγραφή
362
363             while row != None:
364                 client = Client(row[0],row[1],row[2],row[3],row[4],row[5],row[6])
365                 clients.append(client)
366                 row = cursor.fetchone() # διαβάζω την επόμενη εγγραφή
367
368             context = {'model': clients}
369             return render(request, 'FlightManagementSystem/ClientSearch.html', context)
    
```

Εικόνα 341. View αναζήτησης πελάτη

```

al Help views.py - ptyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > saveClient
371 @csrf_exempt
372 def saveClient(request):
373     phoneNo = request.POST['phoneNo']
374     previousPhoneNo = request.POST['previousPhoneNo']
375     email = request.POST['email']
376     firstName = request.POST['firstName']
377     lastName = request.POST['lastName']
378     street = request.POST['street']
379     streetNo = request.POST['streetNo']
380     city = request.POST['city']
381
382     if previousPhoneNo == '0':
383         with connection.cursor() as cursor:
384             cursor.execute('exec dbo.ClientInsert @PhoneNo=%s, @Email=%s, @FirstName=%s, @LastName=%s, @Street=%s, @StreetNo=%s, @City=%s',
385                             [phoneNo,email,firstName,lastName,street,streetNo,city])
386     else:
387         with connection.cursor() as cursor:
388             cursor.execute('exec dbo.ClientUpdate @PhoneNo=%s, @PreviousPhoneNo=%s, @Email=%s, @FirstName=%s, @LastName=%s, @Street=%s, @StreetNo=%s, @City=%s',
389                             [phoneNo,previousPhoneNo,email,firstName,lastName,street,streetNo,city])
390
391     return JsonResponse(json.dumps(True),safe=False)
    
```

Εικόνα 342. View αποθήκευσης πελάτη

```

al Help views.py - ptyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > saveClient
393 @csrf_exempt
394 def deleteClient(request, phoneNo):
395
396     with connection.cursor() as cursor:
397         cursor.execute('exec dbo.ClientDeleteByPhoneNo @PhoneNo=%s', [phoneNo])
398
399     return JsonResponse(json.dumps(True),safe=False)
    
```

Εικόνα 343. View διαγραφής πελάτη

```

al Help views.py - ptyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > saveClient
401 @csrf_exempt
402 def validateClient(request, phoneNo, prevPhoneNo):
403     if phoneNo != prevPhoneNo:
404         query = 'select count(*) from Client where phoneNo = %s'
405
406         with connection.cursor() as cursor:
407             cursor.execute(query, [phoneNo])
408
409             row = cursor.fetchone()
410             count = row[0]
411     else:
412         count = 0
413
414     data = {'phoneNoExists': count != 0}
415
416     return JsonResponse(data, safe = False)
    
```

Εικόνα 344. View ελέγχου εγκυρότητας πελάτη

```

al Help views.py - ptyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > pilotEdit
418 # -----PILOT-----
419
420 def pilotEdit(request, id, mode):
421     if request.session.get('currentUser') is None:
422         return redirect('/')
423     else:
424         model = Pilot()
425
426         if id != '0':
427             with connection.cursor() as cursor:
428                 cursor.execute('exec dbo.PilotSelectByID @ID=%s', [id])
429
430                 row = cursor.fetchone()
431                 model = Pilot(id,row[1],row[2],row[3],row[4],row[5],row[6],row[7].strftime("%Y-%m-%d"),row[8],mode)
432
433         json_dict = json.loads(request.session['currentUser'])
434         currentUser = User(**json_dict)
435
436         context = {'model': model, 'currentUser': currentUser} #Η μεταβλητή context είναι ένα αντικείμενο json, το οποίο θα
437         #χρησιμοποιηθεί για την αρχικοποίηση των controls της ιστοσελίδας
438         return render(request, 'FlightManagementSystem/PilotEdit.html', context)
    
```

Εικόνα 345. View επεξεργασίας πιλότου

```

al Help views.py - ptyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > pilotEdit
440 def pilotSearch(request):
441     if request.session.get('currentUser') is None:
442         return redirect('/')
443     else:
444         with connection.cursor() as cursor:
445             cursor.execute('exec dbo.PilotSearch')
446
447             pilots = []
448
449             row = cursor.fetchone() # διαβάζω την πρώτη εγγραφή
450
451             while row != None:
452                 pilot = Pilot(row[0],row[1],row[2],row[3],row[4],row[5],row[6],row[7].strftime("%Y-%m-%d"),row[8])
453                 pilots.append(pilot)
454                 row = cursor.fetchone() # διαβάζω την επόμενη εγγραφή
455
456             context = {'model': pilots}
457             return render(request, 'FlightManagementSystem/PilotSearch.html', context)
    
```

Εικόνα 346. View αναζήτησης πιλότου

```

FlightManagementSystem > views.py > savePilot
457 @csrf_exempt
458 def savePilot(request):
459     id = request.POST['id']
460     previousID = request.POST['previousID']
461     firstName = request.POST['firstName']
462     lastName = request.POST['lastName']
463     street = request.POST['street']
464     streetNo = request.POST['streetNo']
465     city = request.POST['city']
466     phoneNo = request.POST['phoneNo']
467     birthDate = request.POST['birthDate']
468     flightHours = request.POST['flightHours']
469
470     if previousID == '0':
471         with connection.cursor() as cursor:
472             cursor.execute('exec dbo.PilotInsert @ID=%s, @FirstName=%s, @LastName=%s, @Street=%s, @StreetNo=%s, @City=%s, @PhoneNo=%s, @BirthDate=%s, @FlightHours=%s',
473                             [id, firstName, lastName, street, streetNo, city, phoneNo, birthDate, flightHours])
474     else:
475         with connection.cursor() as cursor:
476             cursor.execute('exec dbo.PilotUpdate @PreviousID=%s, @ID=%s, @FirstName=%s, @LastName=%s, @Street=%s, @StreetNo=%s, @City=%s, @PhoneNo=%s, @BirthDate=%s, @FlightHours=%s',
477                             [previousID, id, firstName, lastName, street, streetNo, city, phoneNo, birthDate, flightHours])
478
479     return JsonResponse(json.dumps(True), safe=False)

```

Εικόνα 347. View αποθήκευσης πιλότου

```

FlightManagementSystem > views.py > ...
483 @csrf_exempt
484 def deletePilot(request, id):
485     try:
486         with connection.cursor() as cursor:
487             cursor.execute('exec dbo.PilotDeleteByID @ID=%s', [id])
488             success = True
489     except:
490         success = False
491
492     return JsonResponse(json.dumps(success), safe=False)

```

Εικόνα 348. View διαγραφής πιλότου

```

FlightManagementSystem > views.py > ...
494 @csrf_exempt
495 def validatePilot(request, id, prevID):
496
497     if id != prevID:
498         query = 'select count(*) from Employee where id = %s'
499
500         with connection.cursor() as cursor:
501             cursor.execute(query, [id])
502
503             row = cursor.fetchone()
504             count = row[0]
505     else:
506         count = 0
507
508     data = {'idExists': count != 0}
509
510     return JsonResponse(data, safe = False)

```

Εικόνα 349. View ελέγχου εγκυρότητας πιλότου

```

views.py - ptyxiaki - Visual Studio Code
views.py x
FlightManagementSystem > views.py > airHostessEdit
512 # -----AIRHOSTESS-----
513
514 def airHostessEdit(request, id, mode):
515     if request.session.get('currentUser') is None:
516         return redirect('/')
517     else:
518         model = AirHostess()
519
520         if id != '0':
521             with connection.cursor() as cursor:
522                 cursor.execute('exec dbo.AirHostessSelectByID @ID=%s', [id])
523
524                 row = cursor.fetchone()
525                 model = AirHostess(id,row[1],row[2],row[3],row[4],row[5],row[6],row[7].strftime("%Y-%m-%d"),row[8],mode)
526
527             json_dict = json.loads(request.session['currentUser'])
528             currentUser = User(**json_dict)
529
530             context = {'model': model, 'currentUser': currentUser} #Η μεταβλητή context είναι ένα αντικείμενο json, το οποίο θα
531             #χρησιμοποιηθεί για την αρχικοποίηση των controls της ιστοσελίδας
532             return render(request, 'FlightManagementSystem/AirHostessEdit.html', context)

```

Εικόνα 350. View επεξεργασίας αεροσυνοδού

```

views.py - ptyxiaki - Visual Studio Code
views.py x
FlightManagementSystem > views.py > airHostessSearch
534 def airHostessSearch(request):
535     if request.session.get('currentUser') is None:
536         return redirect('/')
537     else:
538         with connection.cursor() as cursor:
539             cursor.execute('exec dbo.AirHostessSearch')
540
541             airhostesses = []
542
543             row = cursor.fetchone() # διαβάζω την πρώτη εγγραφή
544
545             while row != None:
546                 airhostess = AirHostess(row[0],row[1],row[2],row[3],row[4],row[5],row[6],row[7].strftime("%Y-%m-%d"),row[8])
547                 airhostesses.append(airhostess)
548                 row = cursor.fetchone() # διαβάζω την επόμενη εγγραφή
549
550             context = {'model': airhostesses}
551             return render(request, 'FlightManagementSystem/AirHostessSearch.html', context)

```

Εικόνα 351. View αναζήτησης αεροσυνοδού

```

views.py - ptyxiaki - Visual Studio Code
views.py x
FlightManagementSystem > views.py > saveAirHostess
553 @csrf_exempt
554 def saveAirHostess(request):
555
556     id = request.POST['id']
557     previousID = request.POST['previousID']
558     firstName = request.POST['firstName']
559     lastName = request.POST['lastName']
560     street = request.POST['street']
561     streetNo = request.POST['streetNo']
562     city = request.POST['city']
563     phoneNo = request.POST['phoneNo']
564     birthDate = request.POST['birthDate']
565     years = request.POST['years']
566
567     if previousID == '0':
568         with connection.cursor() as cursor:
569             cursor.execute('exec dbo.AirHostessInsert @ID=%s, @FirstName=%s, @LastName=%s, @Street=%s, @StreetNo=%s, @City=%s, @PhoneNo=%s, @BirthDate=%s, @Years=%s',
570                 [id,firstName,lastName,street,streetNo,city,phoneNo,birthDate,years])
571     else:
572         with connection.cursor() as cursor:
573             cursor.execute('exec dbo.AirHostessUpdate @PreviousID=%s, @ID=%s, @FirstName=%s, @LastName=%s, @Street=%s, @StreetNo=%s, @City=%s, @PhoneNo=%s, @BirthDate=%s, @Years=%s',
574                 [previousID,id,firstName,lastName,street,streetNo,city,phoneNo,birthDate,years])
575
576     return JsonResponse(json.dumps(True),safe=False)

```

Εικόνα 352. View αποθήκευσης αεροσυνοδού


```

al Help views.py - ptyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > saveAirHostess
578 @csrf_exempt
579 def deleteAirHostess(request, id):
580     try:
581         with connection.cursor() as cursor:
582             cursor.execute('exec dbo.AirHostessDeleteByID @ID=%s',[id])
583             success = True
584     except:
585         success = False
586
587     return JsonResponse(json.dumps(success),safe=False)
    
```

Εικόνα 353. View διαγραφής αεροσυνοδού

```

al Help views.py - ptyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > saveAirHostess
589 @csrf_exempt
590 def validateAirHostess(request, id, prevID):
591
592     if id != prevID:
593         query = 'select count(*) from Employee where id = %s'
594
595         with connection.cursor() as cursor:
596             cursor.execute(query, [id])
597
598             row = cursor.fetchone()
599             count = row[0]
600     else:
601         count = 0
602
603     data = {'idExists': count != 0}
604
605     return JsonResponse(data, safe = False)
    
```

Εικόνα 354. View ελέγχου εγκυρότητας αεροσυνοδού

```

views.py - ptyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > managerEdit
607 # -----MANAGER-----
608
609 def managerEdit(request, id, mode):
610     if request.session.get('currentUser') is None:
611         return redirect('/')
612     else:
613         model = Manager()
614
615         if id != '0':
616             with connection.cursor() as cursor:
617                 cursor.execute('exec dbo.ManagerSelectByID @ID=%s', [id])
618
619                 row = cursor.fetchone()
620                 model = Manager(id,row[1],row[2],row[3],row[4],row[5],row[6],row[7].strftime("%Y-%m-%d"),row[8],row[9], mode)
621
622         json_dict = json.loads(request.session['currentUser'])
623         currentUser = User(**json_dict)
624
625         context = {'model': model, 'currentUser': currentUser} #Η μεταβλητή context είναι ένα αντικείμενο json, το οποίο θα
626         #χρησιμοποιηθεί για την αρχικοποίηση των controls της ιστοσελίδας
627         return render(request, 'FlightManagementSystem/ManagerEdit.html', context)

```

Εικόνα 355. View επεξεργασίας διοικητικού υπαλλήλου

```

views.py - ptyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > managerEdit
629 def managerSearch(request):
630     if request.session.get('currentUser') is None:
631         return redirect('/')
632     else:
633         with connection.cursor() as cursor:
634             cursor.execute('exec dbo.ManagerSearch')
635
636             managers = []
637
638             row = cursor.fetchone() # διαβάζω την πρώτη εγγραφή
639
640             while row != None:
641                 manager = Manager(row[0],row[1],row[2],row[3],row[4],row[5],row[6],row[7].strftime("%Y-%m-%d"),row[8],row[9])
642                 managers.append(manager)
643                 row = cursor.fetchone() # διαβάζω την επόμενη εγγραφή
644
645             context = {'model': managers}
646             return render(request, 'FlightManagementSystem/ManagerSearch.html', context)

```

Εικόνα 356. View αναζήτησης διοικητικού υπαλλήλου

```

views.py - ptyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > saveManager
648 @csrf_exempt
649 def saveManager(request):
650     id = request.POST['id']
651     previousID = request.POST['previousID']
652     firstName = request.POST['firstName']
653     lastName = request.POST['lastName']
654     street = request.POST['street']
655     streetNo = request.POST['streetNo']
656     city = request.POST['city']
657     phoneNo = request.POST['phoneNo']
658     birthDate = request.POST['birthDate']
659     education = request.POST['education']
660     years = request.POST['years']
661
662     if previousID == '':
663         with connection.cursor() as cursor:
664             cursor.execute('exec dbo.ManagerInsert @ID=%s, @FirstName=%s, @LastName=%s, @Street=%s, @StreetNo=%s, @City=%s, @PhoneNo=%s, @BirthDate=%s, @Education=%s, @Years=%s',
665                 [id,firstName,lastName,street,streetNo,city,phoneNo,birthDate,education,years])
666     else:
667         with connection.cursor() as cursor:
668             cursor.execute('exec dbo.ManagerUpdate @PreviousID=%s, @ID=%s, @FirstName=%s, @LastName=%s, @Street=%s, @StreetNo=%s, @City=%s, @PhoneNo=%s, @BirthDate=%s, @Education=%s, @Years=%s',
669                 [previousID,id,firstName,lastName,street,streetNo,city,phoneNo,birthDate,education,years])
670
671     return JsonResponse(json.dumps(True), safe=False)

```

Εικόνα 357. View αποθήκευσης διοικητικού υπαλλήλου

```

al Help views.py - ptyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > saveManager
673 @csrf_exempt
674 def deleteManager(request, id):
675     try:
676         with connection.cursor() as cursor:
677             cursor.execute('exec dbo.ManagerDeleteByID @ID=%s',[id])
678             success = True
679         except:
680             success = False
681
682     return JsonResponse(json.dumps(success),safe=False)
    
```

Εικόνα 358. View διαγραφής διοικητικού υπαλλήλου

```

al Help views.py - ptyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > saveManager
684 @csrf_exempt
685 def validateManager(request, id, prevID):
686
687     if id != prevID:
688         query = 'select count(*) from Employee where id = %s'
689
690         with connection.cursor() as cursor:
691             cursor.execute(query, [id])
692
693             row = cursor.fetchone()
694             count = row[0]
695         else:
696             count = 0
697
698         data = {'idExists': count != 0}
699
700     return JsonResponse(data, safe = False)
    
```

Εικόνα 359. View ελέγχου εγκυρότητας διοικητικού υπαλλήλου

```

al Help views.py - ptyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > validateUser
702 # -----USER-----
703
704 def users(request):
705     context = {}
706     return render(request, 'FlightManagementSystem/Users.html', context)
    
```

Εικόνα 360. View σελίδας χρηστών

```

al Help views.py - ptyxiaki - Visual Studio Code
views.py x
FlightManagementSystem > views.py > validateUser
708 @csrf_exempt
709 def deleteUser(request, username):
710
711     with connection.cursor() as cursor:
712         cursor.execute('exec dbo.UserDeleteByUsername @Username=%s', [username])
713
714     return JsonResponse(json.dumps(True), safe=False)

```

Εικόνα 361. View διαγραφής χρήστη

```

al Help views.py - ptyxiaki - Visual Studio Code
views.py x
FlightManagementSystem > views.py > saveManager
712 @csrf_exempt
713 def validateUser(request, username):
714
715     query = 'select count(*) from [User] where username = %s'
716
717     with connection.cursor() as cursor:
718         cursor.execute(query, [username])
719
720         row = cursor.fetchone()
721         count = row[0]
722
723     data = {'countUsername': count}
724
725     return JsonResponse(data, safe = False)

```

Εικόνα 362. View ελέγχου εγκυρότητας χρήστη

```

al Help views.py - ptyxiaki - Visual Studio Code
views.py x
FlightManagementSystem > views.py > userClientEdit
728 # -----USER CLIENT-----
729
730 def userClientEdit(request, username, mode):
731     if request.session.get('currentUser') is None and username != '0':
732         return redirect('/')
733     else:
734         model = UserClient()
735
736         if username != '0':
737             with connection.cursor() as cursor:
738                 cursor.execute('exec dbo.UserSelectByUserName @UserName=%s', [username])
739
740                 row = cursor.fetchone()
741                 model = UserClient(username, row[1], row[2], row[3], row[4], row[5], row[6], row[7], mode)
742
743         if not request.session.get('currentUser') is None:
744             json_dict = json.loads(request.session['currentUser'])
745             currentUser = User(**json_dict)
746         else:
747             currentUser = User()
748             currentUser.firstName = 'Νέος χρήστης'
749
750     context = {'model': model, 'currentUser': currentUser} #Η μεταβλητή context είναι ένα αντικείμενο json, το οποίο θα
751     #χρησιμοποιηθεί για την αρχικοποίηση των controls της ιστοσελίδας
752     return render(request, 'FlightManagementSystem/UserClientEdit.html', context)

```

Εικόνα 363. View επεξεργασίας χρήστη-πελάτη

```

al Help views.py - ptyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > saveUserClient
754 @csrf_exempt
755 def saveUserClient(request):
756     username = request.POST['username']
757     previousUsername = request.POST['previousUsername']
758     phoneNo = request.POST['phoneNo']
759     previousPhoneNo = request.POST['previousPhoneNo']
760     email = request.POST['email']
761     firstName = request.POST['firstName']
762     lastName = request.POST['lastName']
763     street = request.POST['street']
764     streetNo = request.POST['streetNo']
765     city = request.POST['city']
766
767     if previousUsername == '' and previousPhoneNo == '':
768         password = request.POST['password']
769         hashedPassword = encodePassword(password) # γίνεται η κρυπτογράφηση του κωδικού
770
771         with connection.cursor() as cursor:
772             cursor.execute('exec dbo.UserClientInsert @UserName=%s, @password=%s, @PhoneNo=%s, @Email=%s, @FirstName=%s, @LastName=%s, @Street=%s, @StreetNo=%s, @City=%s',
773                 [username, hashedPassword, phoneNo, email, firstName, lastName, street, streetNo, city])
774         else:
775             with connection.cursor() as cursor:
776                 cursor.execute('exec dbo.UserClientUpdate @UserName=%s, @previousUsername=%s, @phoneNo=%s, @previousPhoneNo=%s, @email=%s, @firstName=%s, @lastName=%s, @Street=%s, @StreetNo=%s, @City=%s',
777                     [username, previousUsername, phoneNo, previousPhoneNo, email, firstName, lastName, street, streetNo, city])
778
779     return JsonResponse(json.dumps(True), safe=False)
    
```

Εικόνα 364. View αποθήκευσης χρήστη-πελάτη

```

al Help views.py - ptyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > saveUserClient
781 def userClientSearch(request):
782     if request.session.get('currentUser') is None:
783         return redirect('/')
784     else:
785         with connection.cursor() as cursor:
786             cursor.execute('exec dbo.UserClientSearch')
787
788             clients = []
789
790             row = cursor.fetchone() # διαβάζω την πρώτη εγγραφή
791
792             while row != None:
793                 client = UserClient(row[0],row[1],row[2],row[3],row[4],row[5],row[6],row[7])
794                 clients.append(client)
795                 row = cursor.fetchone() # διαβάζω την επόμενη εγγραφή
796
797             context = {'model': clients}
798             return render(request, 'FlightManagementSystem/UserClientSearch.html', context)
    
```

Εικόνα 365. View αναζήτησης χρήστη-πελάτη

```

views.py - ptyxiaki - Visual Studio Code
views.py x
FlightManagementSystem > views.py > saveUserClient
800 @csrf_exempt
801 def validateUserClient(request, username, phoneNo, prevUsername, prevPhoneNo):
802
803     if prevUsername != username:
804         query = 'select count(*) from [User] where [Username] = %s'
805
806         with connection.cursor() as cursor:
807             cursor.execute(query, [username])
808
809             row = cursor.fetchone()
810             count1 = row[0]
811     else:
812         count1 = 0
813
814
815     if prevPhoneNo != phoneNo:
816         query = 'select count(*) from [Client] where [PhoneNo] = %s'
817
818         with connection.cursor() as cursor:
819             cursor.execute(query, [phoneNo])
820
821             row = cursor.fetchone()
822             count2 = row[0]
823     else:
824         count2 = 0
825
826     data = {'usernameExists': count1 != 0, 'phoneNoExists': count2 != 0 }
827
828     return JsonResponse(data, safe = False)

```

Εικόνα 366. View ελέγχου εγκυρότητας χρήστη-πελάτη

```

views.py - ptyxiaki - Visual Studio Code
views.py x
FlightManagementSystem > views.py > userAgentEdit
830 # -----USER_AGENT-----
831
832 def userAgentEdit(request, username, mode):
833     if request.session.get('currentUser') is None:
834         return redirect('/')
835     else:
836         model = UserAgent()
837
838         if username != '0':
839             with connection.cursor() as cursor:
840                 cursor.execute('exec dbo.UserSelectByUserName @UserName=%s', [username])
841
842                 row = cursor.fetchone()
843                 model = UserAgent(username, row[1], row[2], row[3], row[4], row[5], row[6], row[7], mode)
844
845         json_dict = json.loads(request.session['currentUser'])
846         currentUser = User(**json_dict)
847
848         context = {'model': model, 'currentUser': currentUser} #Η μεταβλητή context είναι ένα αντικείμενο json, το οποίο θα
849         #χρησιμοποιηθεί για την αρχικοποίηση των controls της ιστοσελίδας
850     return render(request, 'FlightManagementSystem/UserAgentEdit.html', context)

```

Εικόνα 367. View επεξεργασίας χρήστη-ταξιδιωτικού πράκτορα

```

al Help views.py - ptyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > saveUserAgent
852 @csrf_exempt
853 def saveUserAgent(request):
854     username = request.POST['username']
855     previousUsername = request.POST['previousUsername']
856     id = request.POST['id']
857     previousID = request.POST['previousID']
858     firstName = request.POST['firstName']
859     lastName = request.POST['lastName']
860     street = request.POST['street']
861     streetNo = request.POST['streetNo']
862     city = request.POST['city']
863     code = request.POST['code']
864     previousCode = request.POST['previousCode']
865
866     if previousID == '0' and previousUsername == '0' and previousCode == '0':
867         password = request.POST['password']
868         hashedPassword = encodePassword(password) #livesat η κρυπτογράφηση του κωδικού
869
870         with connection.cursor() as cursor:
871             cursor.execute('exec dbo.UserAgentInsert @UserName=%s, @password=%s, @ID=%s, @firstName=%s, @lastName=%s, @Street=%s, @StreetNo=%s, @City=%s, @Code=%s',
872                 (username,hashedPassword,id,firstName,lastName,street,streetNo,city,code))
873     else:
874         with connection.cursor() as cursor:
875             cursor.execute('exec dbo.UserAgentUpdate @UserName=%s, @previousUsername=%s, @ID=%s, @previousID=%s, @firstName=%s, @lastName=%s, @Street=%s, @StreetNo=%s, @City=%s, @Code=%s',
876                 (username,previousUsername,id,previousID,firstName,lastName,street,streetNo,city,code))
877
878     return JsonResponse(json.dumps(True),safe=False)
    
```

Εικόνα 368. View αποθήκευσης χρήστη-ταξιδιωτικού πράκτορα

```

al Help views.py - ptyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > saveUserAgent
880 def userAgentSearch(request):
881     if request.session.get('CurrentUser') is None:
882         return redirect('/')
883     else:
884         with connection.cursor() as cursor:
885             cursor.execute('exec dbo.UserAgentSearch')
886
887             agents = []
888
889             row = cursor.fetchone() # διαβάζω την πρώτη εγγραφή
890
891             while row != None:
892                 agent = UserAgent(row[0],row[1],row[2],row[3],row[4],row[5],row[6],row[7])
893                 agents.append(agent)
894                 row = cursor.fetchone() # διαβάζω την επόμενη εγγραφή
895
896             context = {'model': agents}
897             return render(request, 'FlightManagementSystem/UserAgentSearch.html', context)
    
```

Εικόνα 369. View αναζήτησης χρήστη-ταξιδιωτικού πράκτορα

```
@csrf_exempt
def validateUserAgent(request, username, code, id, previousUsername, previousCode, previousID):

    if (username != previousUsername):
        query = 'select count(*) from [User] where [Username] = %s'

        with connection.cursor() as cursor:
            cursor.execute(query, [username])

            row = cursor.fetchone()
            count1 = row[0]
    else:
        count1 = 0

    if (code != previousCode):
        query = 'select count(*) from [Agent] where [Code] = %s'

        with connection.cursor() as cursor:
            cursor.execute(query, [code])

            row = cursor.fetchone()
            count2 = row[0]
    else:
        count2 = 0

    if (id != previousID):
        query = 'select count(*) from [Agent] where [ID] = %s'

        with connection.cursor() as cursor:
            cursor.execute(query, [id])

            row = cursor.fetchone()
            count3 = row[0]
    else:
        count3 = 0

    data = {'usernameExists': count1 != 0, 'codeExists': count2 != 0, 'idExists': count3 != 0 }

    return JsonResponse(data, safe = False)
```

Εικόνα 370. View ελέγχου εγκυρότητας χρήστη-ταξιδιωτικού πράκτορα

```
views.py - ptyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > userAdministratorEdit
# -----USER ADMINISTRATOR-----
941 def userAdministratorEdit(request, username, mode):
942
943     if request.session.get('currentUser') is None:
944         return redirect('/')
945     else:
946         model = UserAdministrator()
947
948         if username != '0':
949             with connection.cursor() as cursor:
950                 cursor.execute('exec dbo.UserSelectByUserName @UserName=%s', [username])
951
952                 row = cursor.fetchone()
953                 model = UserAdministrator(username,row[1],row[2],row[3],row[4],row[5],row[6],row[7],row[8].strftime("%Y-%m-%d"),mode)
954
955     json_dict = json.loads(request.session['currentUser'])
956     currentUser = User(**json_dict)
957
958     context = {'model': model, 'currentUser': currentUser} #Η μεταβλητή context είναι ένα αντικείμενο json, το οποίο θα
959     #χρησιμοποιηθεί για την αρχικοποίηση των controls της ιστοσελίδας
960     return render(request, 'FlightManagementSystem/UserAdministratorEdit.html', context)
```

Εικόνα 371. View επεξεργασίας χρήστη-διαχειριστή


```

views.py
FlightManagementSystem > views.py > saveUserAdministrator
962 @csrf_exempt
963 def saveUserAdministrator(request):
964     username = request.POST['username']
965     previousUsername = request.POST['previousUsername']
966     id = request.POST['id']
967     previousID = request.POST['previousID']
968     firstName = request.POST['firstName']
969     lastName = request.POST['lastName']
970     street = request.POST['street']
971     streetNo = request.POST['streetNo']
972     city = request.POST['city']
973     phoneNo = request.POST['phoneNo']
974     birthDate = request.POST['birthDate']
975
976     if previousID == '0' and previousUsername == '0':
977         password = request.POST['password']
978         hashedPassword = encodePassword(password) #Γίνεται η κρυπτογράφηση του κωδικού
979
980         with connection.cursor() as cursor:
981             cursor.execute('exec dbo.UserAdministratorInsert @Username=%s, @Password=%s, @ID=%s, @FirstName=%s, @LastName=%s, @Street=%s, @StreetNo=%s, @City=%s, @PhoneNo=%s, @BirthDate=%s',
982                 [username, hashedPassword, id, firstName, lastName, street, streetNo, city, phoneNo, birthDate])
983     else:
984         with connection.cursor() as cursor:
985             cursor.execute('exec dbo.UserAdministratorUpdate @Username=%s, @PreviousUsername=%s, @ID=%s, @PreviousID=%s, @FirstName=%s, @LastName=%s, @Street=%s, @StreetNo=%s, @City=%s, @PhoneNo=%s, @BirthDate=%s',
986                 [username, previousUsername, id, previousID, firstName, lastName, street, streetNo, city, phoneNo, birthDate])
987
988     return JsonResponse(json.dumps(True), safe=False)
    
```

Εικόνα 372. View αποθήκευσης χρήστη-διαχειριστή

```

views.py
FlightManagementSystem > views.py > saveUserAdministrator
990 def userAdministratorSearch(request):
991     if request.session.get('currentUser') is None:
992         return redirect('/')
993     else:
994         with connection.cursor() as cursor:
995             cursor.execute('exec dbo.UserAdministratorSearch')
996
997             administrators = []
998
999             row = cursor.fetchone() # διαβάζω την πρώτη εγγραφή
1000
1001             while row != None:
1002                 administrator = UserAdministrator(row[0],row[1],row[2],row[3],row[4],row[5],row[6],row[7],row[8].strftime("%d-%m-%Y"))
1003                 administrators.append(administrator)
1004                 row = cursor.fetchone() # διαβάζω την επόμενη εγγραφή
1005
1006             context = {'model': administrators}
1007             return render(request, 'FlightManagementSystem/UserAdministratorSearch.html', context)
    
```

Εικόνα 373. View αναζήτησης χρήστη-διαχειριστή

```

views.py
FlightManagementSystem > views.py > saveUserAdministrator
1009 @csrf_exempt
1010 def validateUserAdministrator(request, username, id, previousUsername, previousID):
1011
1012     if (username != previousUsername):
1013         query = 'select count(*) from [User] where [Username] = %s'
1014
1015         with connection.cursor() as cursor:
1016             cursor.execute(query, [username])
1017
1018             row = cursor.fetchone()
1019             count1 = row[0]
1020     else:
1021         count1 = 0
1022
1023     if (id != previousID):
1024         query = 'select count(*) from [Employee] where [ID] = %s'
1025
1026         with connection.cursor() as cursor:
1027             cursor.execute(query, [id])
1028
1029             row = cursor.fetchone()
1030             count2 = row[0]
1031     else:
1032         count2 = 0
1033
1034     data = {'usernameExists': count1 != 0, 'idExists': count2 != 0 }
1035
1036     return JsonResponse(data, safe = False)
    
```

Εικόνα 374. View ελέγχου εγκυρότητας χρήστη-διαχειριστή

```
def routeEdit(request, id, mode):
    if request.session.get('currentUser') is None:
        return redirect('/')
    else:
        model = Route('0')

        if id != '0':
            with connection.cursor() as cursor:
                cursor.execute('exec dbo.RouteSelectByID @ID=%s', [id])
                row = cursor.fetchone()
                model = Route(id,row[1],row[2],row[3],row[4],row[5],row[6],row[7],row[8],mode)

        if id == '0':
            model.isNew = True
            model.id = ''
        else:
            model.isNew = False

        cities = [City(0,'')]
        with connection.cursor() as cursor:
            cursor.execute('exec dbo.CitySelectAll')
            row = cursor.fetchone() # διαβάζω την πρώτη εγγραφή

            while row != None:
                city = City(row[0],row[1])
                cities.append(city)
                row = cursor.fetchone() # διαβάζω την επόμενη εγγραφή

        days = [Day(0,'')]
        with connection.cursor() as cursor:
            cursor.execute('exec dbo.DaySelectAll')
            row = cursor.fetchone() # διαβάζω την πρώτη εγγραφή

            while row != None:
                day = Day(row[0],row[1])
                days.append(day)
                row = cursor.fetchone() # διαβάζω την επόμενη εγγραφή

        json_dict = json.loads(request.session['currentUser'])
        currentUser = User(**json_dict)

        context = {'model': model, 'currentUser': currentUser, 'cities': cities, 'days': days} #Η μεταβλητή context είναι ένα
        #αντικείμενο json, το οποίο θα χρησιμοποιηθεί για την αρχικοποίηση των controls της ιστοσελίδας
        return render(request, 'FlightManagementSystem/RouteEdit.html', context)
```

Εικόνα 375. View επεξεργασίας δρομολογίου

```
@csrf_exempt
def saveRoute(request):
    id = request.POST['id']
    previousID = request.POST['previousID']
    depPointID = request.POST['depPointID']
    destPointID = request.POST['destPointID']
    dayID = request.POST['dayID']
    distance = request.POST['distance']
    depTime = request.POST['depTime']
    arrTime = request.POST['arrTime']
    busPrice = request.POST['busPrice']
    ecoPrice = request.POST['ecoPrice']

    if previousID == '0':
        with connection.cursor() as cursor:
            cursor.execute('exec dbo.RouteInsert @ID=%s, @DeparturePointID=%s, @DestinationPointID=%s, @DayID=%s, @Distance=%s, @DepartureTime=%s, @ArrivalTime=%s, @BusinessPrice=%s, @EconomyPrice=%s',
                [id,depPointID,destPointID,dayID,distance,depTime,arrTime,busPrice,ecoPrice])
        else:
            with connection.cursor() as cursor:
                cursor.execute('exec dbo.RouteUpdate @PreviousID=%s, @ID=%s, @DeparturePointID=%s, @DestinationPointID=%s, @DayID=%s, @Distance=%s, @DepartureTime=%s, @ArrivalTime=%s, @BusinessPrice=%s, @EconomyPrice=%s',
                    [previousID,id,depPointID,destPointID,dayID,distance,depTime,arrTime,busPrice,ecoPrice])

    return JsonResponse(json.dumps(True),safe=False)
```

Εικόνα 376. View αποθήκευσης δρομολογίου

```
views.py - phyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > ...
1111 def routeSearch(request):
1112     if request.session.get('currentUser') is None:
1113         return redirect('/')
1114     else:
1115         with connection.cursor() as cursor:
1116             cursor.execute('exec dbo.RouteSearch')
1117
1118             routes = []
1119
1120             row = cursor.fetchone() # διαβάζω την πρώτη εγγραφή
1121
1122             while row != None:
1123                 route = RouteRow(row[0],row[1],row[2],row[3],row[4],row[5],row[6],row[7],row[8])
1124                 routes.append(route)
1125                 row = cursor.fetchone() # διαβάζω την επόμενη εγγραφή
1126
1127             context = {'model': routes}
1128             return render(request, 'FlightManagementSystem/RouteSearch.html', context)
```

Εικόνα 377. View αναζήτησης δρομολογίου

```

1130 @csrf_exempt
1131 def deleteRoute(request, id):
1132     try:
1133         with connection.cursor() as cursor:
1134             cursor.execute('exec dbo.RouteDeleteByID @ID=%s', [id])
1135             success = True
1136     except:
1137         success = False
1138
1139     return JsonResponse(json.dumps(success), safe=False)
    
```

Εικόνα 378. View διαγραφής δρομολογίου

```

1141 @csrf_exempt
1142 def validateRoute(request, id, prevID):
1143     if id != prevID:
1144         query = 'select count(*) from Route where id = %s'
1145
1146         with connection.cursor() as cursor:
1147             cursor.execute(query, [id])
1148
1149             row = cursor.fetchone()
1150             count = row[0]
1151     else:
1152         count = 0
1153
1154     data = {'idExists': count != 0}
1155
1156     return JsonResponse(data, safe = False)
    
```

Εικόνα 379. View ελέγχου εγκυρότητας δρομολογίου

```

def transportEdit(request, id, mode):
    if request.session.get('currentUser') is None:
        return redirect('/')
    else:
        model = Transport(0)

        if id != 0:
            with connection.cursor() as cursor:
                cursor.execute('exec dbo.TransportSelectByID @TransportID=%s', [id])

                row = cursor.fetchone()
                model = Transport(id,row[1],row[2].strftime("%Y-%m-%d"),row[3],row[4],row[5],row[6],row[7],row[8],row[9],row[10],
                row[11],row[12],mode)

        if id == 0:
            model.isNew = True
        else:
            model.isNew = False

        routes = [RouteItem('0','')]
        with connection.cursor() as cursor:
            cursor.execute('exec dbo.RouteSelectAll')
            row = cursor.fetchone() # διαβάζω την πρώτη εγγραφή

            while row != None:
                route = RouteItem(row[0],row[1])
                routes.append(route)
                row = cursor.fetchone() # διαβάζω την επόμενη εγγραφή

        pilots = [PilotItem('0','')]
        with connection.cursor() as cursor:
            cursor.execute('exec dbo.PilotSelectAll')
            row = cursor.fetchone() # διαβάζω την πρώτη εγγραφή

            while row != None:
                pilot = PilotItem(row[0],row[1])
                pilots.append(pilot)
                row = cursor.fetchone() # διαβάζω την επόμενη εγγραφή

        airhostesses = [AirhostessItem('0','')]
        with connection.cursor() as cursor:
            cursor.execute('exec dbo.AirhostessSelectAll')
            row = cursor.fetchone() # διαβάζω την πρώτη εγγραφή

            while row != None:
                airhostess = AirhostessItem(row[0],row[1])
                airhostesses.append(airhostess)
                row = cursor.fetchone() # διαβάζω την επόμενη εγγραφή

        managers = [ManagerItem('0','')]
        with connection.cursor() as cursor:
            cursor.execute('exec dbo.ManagerSelectAll')
            row = cursor.fetchone() # διαβάζω την πρώτη εγγραφή

            while row != None:
                manager = ManagerItem(row[0],row[1])
                managers.append(manager)
                row = cursor.fetchone() # διαβάζω την επόμενη εγγραφή

        airplanes = [AirplaneItem('')]
        with connection.cursor() as cursor:
            cursor.execute('exec dbo.AirplaneSelectAll')
            row = cursor.fetchone() # διαβάζω την πρώτη εγγραφή

            while row != None:
                airplane = AirplaneItem(row[0])
                airplanes.append(airplane)
                row = cursor.fetchone() # διαβάζω την επόμενη εγγραφή

        json_dict = json.loads(request.session['currentUser'])
        currentUser = User(**json_dict)

        context = {'model': model, 'currentUser': currentUser, 'routes': routes, 'pilots': pilots, 'airhostesses': airhostesses,
        'managers': managers, 'airplanes': airplanes} #Η μεταβλητή context είναι ένα αντικείμενο json, το οποίο θα χρησιμοποιηθεί
        #για την αρχικοποίηση των controls της ιστοσελίδας
        return render(request, 'FlightManagementSystem/TransportEdit.html', context)

```

Εικόνα 380. View επεξεργασίας μεταφοράς

```

@csrf_exempt
def saveTransport(request):
    transportID = request.POST['transportID']
    routeID = request.POST['routeID']
    transportDate = request.POST['transportDate']
    airplaneName = request.POST['airplaneName']
    pilotID = request.POST['pilotID']
    coPilotID = request.POST['coPilotID']
    airHostessID1 = request.POST['airHostessID1']
    airHostessID2 = request.POST['airHostessID2']
    airHostessID3 = request.POST['airHostessID3']
    airHostessID4 = request.POST['airHostessID4']
    managerID = request.POST['managerID']
    accSeats = request.POST['accSeats']
    busSeats = request.POST['busSeats']

    if transportID == '0':
        with connection.cursor() as cursor:
            cursor.execute('exec dbo.InsertTransportID @routeID=%s, @transportDate=%s, @airplaneName=%s, @pilotID=%s, @coPilotID=%s, @airHostessID1=%s, @airHostessID2=%s, @airHostessID3=%s, @airHostessID4=%s, @managerID=%s, @accSeats=%s, @busSeats=%s',
            [routeID,transportDate,airplaneName,pilotID,coPilotID,airHostessID1,airHostessID2,airHostessID3,airHostessID4,managerID,accSeats,busSeats])
        row = cursor.fetchone() # διαβάζω την πρώτη εγγραφή
    else:
        with connection.cursor() as cursor:
            cursor.execute('exec dbo.TransportUpdate @transportID=%s, @routeID=%s, @transportDate=%s, @airplaneName=%s, @pilotID=%s, @coPilotID=%s, @airHostessID1=%s, @airHostessID2=%s, @airHostessID3=%s, @airHostessID4=%s, @managerID=%s, @accSeats=%s, @busSeats=%s',
            [transportID,routeID,transportDate,airplaneName,pilotID,coPilotID,airHostessID1,airHostessID2,airHostessID3,airHostessID4,managerID,accSeats,busSeats])
            transportID=int(transportID)

    return JsonResponse(json.dumps(transportID),safe=False)

```

Εικόνα 381. View αποθήκευσης μεταφοράς

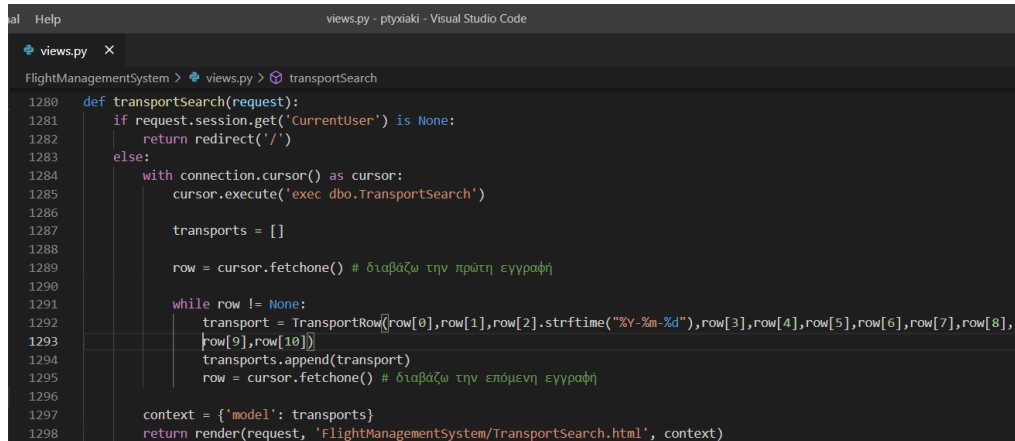
```
@csrf_exempt
def validateTransport(request, transportID, routeID, pilotID, copilotID, transportDate, airHostessID1, airHostessID2,
airHostessID3, airHostessID4, airplaneName):

with connection.cursor() as cursor:
    cursor.execute('exec dbo.TransportValidation @routeID=%s, @airplaneName=%s, @pilotID=%s, @copilotID=%s, @airHostessID1=%s, @airHostessID2=%s, @airHostessID3=%s, @airHostessID4=%s, @transportDate=%s, @transportID=%s',
    [routeID,airplaneName,pilotID,copilotID,airHostessID1,airHostessID2,airHostessID3,airHostessID4,transportDate,transportID])

    row = cursor.fetchone()
    count = row[0]
    data = {'hasConflict': count != 0}

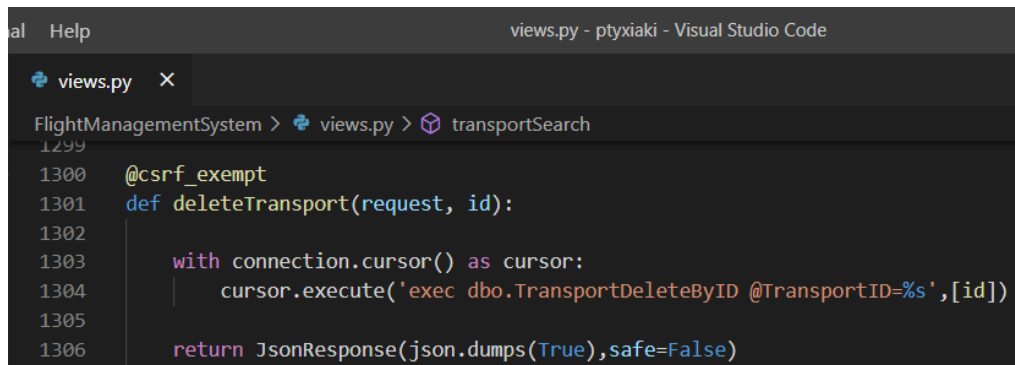
return JsonResponse(data, safe = False)
```

Εικόνα 382. View ελέγχου εγκυρότητας μεταφοράς



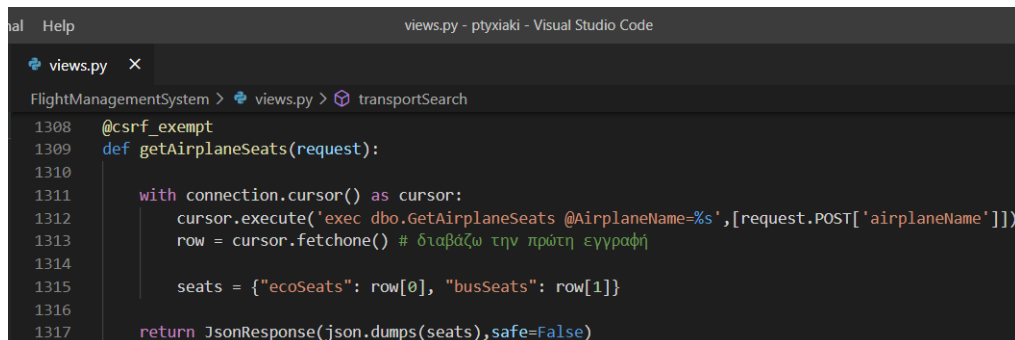
```
views.py - ptyxiaki - Visual Studio Code
views.py x
FlightManagementSystem > views.py > transportSearch
1280 def transportSearch(request):
1281     if request.session.get('currentUser') is None:
1282         return redirect('/')
1283     else:
1284         with connection.cursor() as cursor:
1285             cursor.execute('exec dbo.TransportSearch')
1286
1287             transports = []
1288
1289             row = cursor.fetchone() # διαβάζω την πρώτη εγγραφή
1290
1291             while row != None:
1292                 transport = TransportRow(row[0],row[1],row[2].strftime("%Y-%m-%d"),row[3],row[4],row[5],row[6],row[7],row[8],
1293                 [row[9],row[10]])
1294                 transports.append(transport)
1295                 row = cursor.fetchone() # διαβάζω την επόμενη εγγραφή
1296
1297             context = {'model': transports}
1298             return render(request, 'FlightManagementSystem/TransportSearch.html', context)
```

Εικόνα 383. View αναζήτησης μεταφοράς



```
views.py - ptyxiaki - Visual Studio Code
views.py x
FlightManagementSystem > views.py > transportSearch
1299
1300 @csrf_exempt
1301 def deleteTransport(request, id):
1302
1303     with connection.cursor() as cursor:
1304         cursor.execute('exec dbo.TransportDeleteByID @TransportID=%s',[id])
1305
1306     return JsonResponse(json.dumps(True),safe=False)
```

Εικόνα 384. View διαγραφής μεταφοράς



```
views.py - ptyxiaki - Visual Studio Code
views.py x
FlightManagementSystem > views.py > transportSearch
1308 @csrf_exempt
1309 def getAirplaneSeats(request):
1310
1311     with connection.cursor() as cursor:
1312         cursor.execute('exec dbo.GetAirplaneSeats @AirplaneName=%s',[request.POST['airplaneName']])
1313         row = cursor.fetchone() # διαβάζω την πρώτη εγγραφή
1314
1315         seats = {"ecoSeats": row[0], "busSeats": row[1]}
1316
1317     return JsonResponse(json.dumps(seats),safe=False)
```

Εικόνα 385. View εισαγωγής τύπου θέσεων αεροπλάνου στη μεταφορά

```

# -----
RESERVATION-----
def reservationEdit(request, id, mode):
    if request.session.get('currentUser') is None:
        return redirect('/')
    else:
        model = Reservation()

        if id != 0:
            with connection.cursor() as cursor:
                cursor.execute('exec dbo.ReservationSelectByID @ReservationID=%s', [id])

                row = cursor.fetchone()
                model = Reservation(id,row[1],row[2].strftime("%Y-%m-%d"),row[3].strftime("%Y-%m-%d"),row[4],row[5],row[6],row[7],
                row[8],row[9],mode)

        if id == 0:
            model.isNew = True
        else:
            model.isNew = False

        cities = [City(0,'')]
        with connection.cursor() as cursor:
            cursor.execute('exec dbo.CitySelectAll')
            row = cursor.fetchone() # διαβάζω την πρώτη εγγραφή

            while row != None:
                city = City(row[0],row[1])
                cities.append(city)
                row = cursor.fetchone() # διαβάζω την επόμενη εγγραφή

        clients = [ClientItem('0','')]
        with connection.cursor() as cursor:
            cursor.execute('exec dbo.ClientSelectAll')
            row = cursor.fetchone() # διαβάζω την πρώτη εγγραφή

            while row != None:
                client = ClientItem(row[0],row[1])
                clients.append(client)
                row = cursor.fetchone() # διαβάζω την επόμενη εγγραφή

        json_dict = json.loads(request.session['currentUser'])
        currentUser = User(**json_dict)

        context = {'model': model, 'currentUser' : currentUser, 'cities': cities, 'clients': clients} #Η μεταβλητή context είναι
        #ένα αντικείμενο json, το οποίο θα χρησιμοποιηθεί για την αρχικοποίηση των controls της ιστοσελίδας
        return render(request, 'FlightManagementSystem/ReservationEdit.html', context)

```

Εικόνα 386. View επεξεργασίας κράτησης χρήστη-ταξιδιωτικού πράκτορα

```

@csrf_exempt
def getTransports(request):
    depCityID = request.POST['depCityID']
    arvCityID = request.POST['arvCityID']
    resDate = request.POST['resDate']

    transports = []
    with connection.cursor() as cursor:
        cursor.execute('exec dbo.TransportFindRoute @DepCityID=%s, @ArvCityID=%s, @ReservationDate=%s',
        [depCityID,arvCityID,resDate])
        row = cursor.fetchone() # διαβάζω την πρώτη εγγραφή

        while row != None:
            transportItem = TransportItem(row[0],row[1],row[2],row[3],row[4],row[5],row[6],row[7],row[8],row[9])
            transports.append(transportItem.toJson())
            row = cursor.fetchone() # διαβάζω την επόμενη εγγραφή

    transports2 = []
    with connection.cursor() as cursor:
        cursor.execute('exec dbo.GetIndirectTransports @DepCityID=%s, @ArvCityID=%s, @ReservationDate=%s',
        [depCityID,arvCityID,resDate])
        row = cursor.fetchone() # διαβάζω την πρώτη εγγραφή

        while row != None:
            transportItem1 = TransportItem(row[0],row[1],row[2],row[3],row[4],row[5],row[6],row[7],row[8],row[9])
            transportItem2 = TransportItem(row[10],row[11],row[12],row[13],row[14],row[15],row[16],row[17],row[18],row[19])
            transports2.append(transportItem1.toJson())
            transports2.append(transportItem2.toJson())
            row = cursor.fetchone() # διαβάζω την επόμενη εγγραφή

    data = {'result1': transports, 'result2': transports2}
    return JsonResponse(json.dumps(data),safe=False)

```

Εικόνα 387. View εισαγωγής μεταφορών στη κράτηση του χρήστη-ταξιδιωτικού πράκτορα

```

@csrf_exempt
def saveReservation(request):
    reservationID = request.POST['reservationID']
    phoneNo = request.POST['phoneNo']
    resDate = request.POST['resDate']
    expDate = request.POST['expDate']
    depCityID = request.POST['depCityID']
    arrCityID = request.POST['arrCityID']
    transportID1 = request.POST['transportID1']
    transportID2 = request.POST['transportID2']
    busSeat = request.POST['busSeat']
    agentID = request.session.get('agentID')

    if reservationID == '0':
        with connection.cursor() as cursor:
            cursor.execute('exec dbo.ReservationInsert @PhoneNo=%s, @ReservationDate=%s, @ExpirationDate=%s, @DepartureCityID=%s, @ArrivalCityID=%s, @TransportID1=%s, @TransportID2=%s, @BusinessClass=%s, @AgentID=%s',
                [phoneNo, resDate, expDate, depCityID, arrCityID, transportID1, transportID2, busSeat, agentID])
            row = cursor.fetchone() # διαβάζω τον πρώτο εγγραφή
            reservationID = row[0]
    else:
        with connection.cursor() as cursor:
            cursor.execute('exec dbo.ReservationUpdate @ReservationID=%s, @PhoneNo=%s, @ReservationDate=%s, @ExpirationDate=%s, @DepartureCityID=%s, @ArrivalCityID=%s, @TransportID1=%s, @TransportID2=%s, @BusinessClass=%s, @AgentID=%s',
                [reservationID, phoneNo, resDate, expDate, depCityID, arrCityID, transportID1, transportID2, busSeat, agentID])

    return JsonResponse(json.dumps(int(reservationID)), safe=False)

```

Εικόνα 388. View αποθήκευσης κράτησης χρήστη-ταξιδιωτικού πράκτορα

```

views.py - ptyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > reservationSearch
1426 def reservationSearch(request):
1427     if request.session.get('currentUser') is None:
1428         return redirect('/')
1429     else:
1430         with connection.cursor() as cursor:
1431             cursor.execute('exec dbo.ReservationSearch @AgentID=%s, @PhoneNo=%s',
1432                 [request.session.get('agentID'), request.session.get('clientID')])
1433             reservations = []
1434
1435             row = cursor.fetchone() # διαβάζω την πρώτη εγγραφή
1436
1437             while row != None:
1438                 reservation = ReservationRow([row[0], row[1], row[2].strftime("%Y-%m-%d"), row[3].strftime("%Y-%m-%d"), row[4], row[5],
1439                     row[6], row[7], row[8]])
1440                 reservations.append(reservation)
1441                 row = cursor.fetchone() # διαβάζω την επόμενη εγγραφή
1442
1443             context = {'model': reservations}
1444             return render(request, 'FlightManagementSystem/ReservationSearch.html', context)

```

Εικόνα 389. View αναζήτησης κράτησης χρήστη-ταξιδιωτικού πράκτορα

```

views.py - ptyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > reservationSearch
1446 @csrf_exempt
1447 def deleteReservation(request, id):
1448
1449     with connection.cursor() as cursor:
1450         cursor.execute('exec dbo.ReservationDelete @ReservationID=%s', [id])
1451
1452     return JsonResponse(json.dumps(True), safe=False)

```

Εικόνα 390. View διαγραφής κράτησης χρήστη-ταξιδιωτικού πράκτορα

```

views.py - ptyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > validateReservation
1454 @csrf_exempt
1455 def validateReservation(request):
1456     phoneNo = request.POST['PhoneNo']
1457     resDate = request.POST['ResDate']
1458     reservationID = request.POST['ReservationID']
1459
1460     with connection.cursor() as cursor:
1461         cursor.execute('exec dbo.ReservationValidate @ReservationID=%s, @PhoneNo=%s, @ResDate=%s',
1462             [reservationID, phoneNo, resDate])
1463         row = cursor.fetchone()
1464
1465         counter = row[0]
1466
1467     return JsonResponse(json.dumps(int(counter)), safe=False)

```

Εικόνα 391. View ελέγχου εγκυρότητας κράτησης χρήστη-ταξιδιωτικού πράκτορα

```

# -----CLIENT
RESERVATION-----
def clientReservationEdit(request, id, mode):
    if request.session.get('currentUser') is None:
        return redirect('/')
    else:
        model = Reservation()
        model.phoneNo = request.session.get("clientID")

        if id != 0:
            with connection.cursor() as cursor:
                cursor.execute('exec dbo.ReservationSelectByID @ReservationID=%s', [id])

                row = cursor.fetchone()
                model = Reservation(id,row[1],row[2].strftime("%Y-%m-%d"),row[3].strftime("%Y-%m-%d"),row[4],row[5],row[6],row[7],
                row[8],row[9],mode)

        if id == 0:
            model.isNew = True
        else:
            model.isNew = False

        cities = [City(0,'')]
        with connection.cursor() as cursor:
            cursor.execute('exec dbo.CitySelectAll')
            row = cursor.fetchone() # διαβάζω την πρώτη εγγραφή

            while row != None:
                city = City(row[0],row[1])
                cities.append(city)
                row = cursor.fetchone() # διαβάζω την επόμενη εγγραφή

        json_dict = json.loads(request.session['currentUser'])
        currentUser = User(**json_dict)

        context = {'model': model, 'currentUser' : currentUser, 'cities': cities} #Η μεταβλητή context είναι ένα αντικείμενο json,
        # το οποίο θα χρησιμοποιηθεί για την αρχικοποίηση των controls της ιστοσελίδας
        return render(request, 'FlightManagementSystem/ClientReservationEdit.html', context)

```

Εικόνα 392. View επεξεργασίας κράτησης χρήστη-πελάτη

```

@csrf_exempt
def saveClientReservation(request):
    reservationID = request.POST['reservationID']
    phoneNo = request.session.get('clientID')
    resDate = request.POST['resDate']
    expDate = request.POST['expDate']
    depCityID = request.POST['depCityID']
    arrCityID = request.POST['arrCityID']
    transportID1 = request.POST['transportID1']
    transportID2 = request.POST['transportID2']
    busSeat = request.POST['busSeat']
    agentID = 0

    if reservationID == '0':
        with connection.cursor() as cursor:
            cursor.execute('exec dbo.ReservationInsert @PhoneNo=%s, @ReservationDate=%s, @ExpirationDate=%s, @DepartureCityID=%s, @ArrivalCityID=%s, @TransportID1=%s, @TransportID2=%s, @BusinessClass=%s, @AgentID=%s',
            [phoneNo,resDate,expDate,depCityID,arrCityID,transportID1,transportID2,busSeat,agentID])
            row = cursor.fetchone() # διαβάζω την πρώτη εγγραφή
            reservationID = row[0]
    else:
        with connection.cursor() as cursor:
            cursor.execute('exec dbo.ReservationUpdate @ReservationID=%s, @PhoneNo=%s, @ReservationDate=%s, @ExpirationDate=%s, @DepartureCityID=%s, @ArrivalCityID=%s, @TransportID1=%s, @TransportID2=%s, @BusinessClass=%s, @AgentID=%s',
            [reservationID,phoneNo,resDate,expDate,depCityID,arrCityID,transportID1,transportID2,busSeat,agentID])

    return JsonResponse(json.dumps(int(reservationID)),safe=False)

```

Εικόνα 393. View αποθήκευσης κράτησης χρήστη-πελάτη

```

views.py - ptyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > clientReservationSearch
1534 def clientReservationSearch(request):
1535     if request.session.get('currentUser') is None:
1536         return redirect('/')
1537     else:
1538         with connection.cursor() as cursor:
1539             cursor.execute('exec dbo.ReservationSearch @AgentID=%s, @PhoneNo=%s',
1540             [request.session.get('agentID'),request.session.get('clientID')])
1541
1542             reservations = []
1543
1544             row = cursor.fetchone() # διαβάζω την πρώτη εγγραφή
1545
1546             while row != None:
1547                 reservation = ReservationRow([row[0],row[1],row[2].strftime("%Y-%m-%d"),row[3].strftime("%Y-%m-%d"),row[4],row[5],
1548                 row[6],row[7],row[8]])
1549                 reservations.append(reservation)
1550                 row = cursor.fetchone() # διαβάζω την επόμενη εγγραφή
1551
1552             context = {'model': reservations}
1553             return render(request, 'FlightManagementSystem/ClientReservationSearch.html', context)

```

Εικόνα 394. View αναζήτησης κράτησης χρήστη-πελάτη


```

al Help views.py - ptyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > sendEmail
1555 # -----TICKET-----
1556 def sendEmail(emailTo, ticketID, clientName):
1557     msg = MIMEMultipart('alternative')
1558     msg['Subject'] = 'ΕΚΔΟΣΗ ΕΙΣΙΤΗΡΙΟΥ'
1559     msg['From'] = 'zaza031193@gmail.com'
1560     msg['To'] = emailTo
1561
1562     html = "<html><body>Αγαπητέ/ή " + clientName + ", το εισιτήριό σας εκδόθηκε με επιτυχία."
1563     html += "<br />Ο αριθμός εισιτηρίου είναι: <a href='\"http://localhost:8000/showticket/\" + str(ticketID) +
1564     '\" target='\"_blank\">\" + str(ticketID) + </a>"
1565     html += "<br /><br />Καλό σας ταξίδι</body></html>"
1566
1567     part1 = MIMEText(html,"html")
1568     msg.attach(part1)
1569
1570     context = ssl.create_default_context()
1571     with smtplib.SMTP_SSL("smtp.gmail.com", 465, context=context) as server:
1572         server.login("zaza031193@gmail.com", 'z_achariou93')
1573         server.sendmail('zaza031193@gmail.com', emailTo, msg.as_string())

```

Εικόνα 395. View διαμόρφωσης μηνύματος email του εισιτηρίου

```

al Help views.py - ptyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > sendEmail
1575 # Μετασχηματισμός κράτησης σε εισιτήριο
1576 @csrf_exempt
1577 def printTicket(request):
1578     reservationID = request.POST['reservationID']
1579
1580     with connection.cursor() as cursor:
1581         cursor.execute('exec dbo.TicketInsert @ReservationID=%s', [reservationID])
1582         row = cursor.fetchone() # διαβάζω την επόμενη εγγραφή
1583
1584         ticketID = int(row[0])
1585         email = str(row[1])
1586         clientName = str(row[2])
1587         sendEmail(email, ticketID, clientName)

```

Εικόνα 396. View μετασχηματισμού κράτησης σε εισιτήριο

```

al Help views.py - ptyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > sendEmail
1591 def showTicket(request, ticketID):
1592     if request.session.get('currentUser') is None:
1593         return redirect('/')
1594     else:
1595         with connection.cursor() as cursor:
1596             cursor.execute('exec dbo.TicketSelect @TicketID=%s', [ticketID])
1597             row = cursor.fetchone() # διαβάζω την επόμενη εγγραφή
1598
1599             price = float(row[6])
1600
1601             if price == int(price):
1602                 price = int(price)
1603
1604             km = float(row[7])
1605
1606             if km == int(km):
1607                 km = int(km)
1608
1609             ticket = Ticket(row[0], row[1], row[2].strftime("%Y-%m-%d"), row[3], row[4], row[5], price, km)
1610
1611             json_dict = json.loads(request.session['currentUser'])
1612             currentUser = User(**json_dict)
1613
1614             context = {'model': ticket, 'currentUser': currentUser}
1615             return render(request, 'FlightManagementSystem/TicketPreview.html', context)

```

Εικόνα 397. View έκδοσης εισιτηρίου από τον ταξιδιωτικό πράκτορα

```

al Help views.py - ptyxiaki - Visual Studio Code
views.py x
FlightManagementSystem > views.py > sendEmail
1617 def showClientTicket(request,ticketID):
1618     if request.session.get('currentUser') is None:
1619         return redirect('/')
1620     else:
1621         with connection.cursor() as cursor:
1622             cursor.execute('exec dbo.TicketSelect @TicketID=%s', [ticketID])
1623             row = cursor.fetchone() # διαβάζω την επόμενη εγγραφή
1624
1625             ticket = Ticket(row[0],row[1],row[2].strftime("%Y-%m-%d"),row[3],row[4],row[5],row[6],row[7])
1626
1627             json_dict = json.loads(request.session['currentUser'])
1628             currentUser = User(**json_dict)
1629
1630             context = {'model': ticket, 'currentUser' : currentUser}
1631             return render(request, 'FlightManagementSystem/ClientTicketPreview.html', context)

```

Εικόνα 398. View έκδοσης εισιτηρίου από τον πελάτη

```

# -----
DASHBOARD-----
def clientDashboard(request):
    if request.session.get('currentUser') is None:
        return redirect('/')
    else:
        with connection.cursor() as cursor:
            cursor.execute('exec dbo.ReservationSearch @AgentID=%s, @PhoneNo=%s',[request.session.get('agentID'),
            request.session.get('clientID')])

            reservations = []

            row = cursor.fetchone() # διαβάζω την πρώτη εγγραφή

            while row != None:
                reservation = ReservationRow(row[0],row[1],row[2].strftime("%Y-%m-%d"),row[3].strftime("%Y-%m-%d"),row[4],row[5],
                row[6],row[7],row[8])
                reservations.append(reservation)
                row = cursor.fetchone() # διαβάζω την επόμενη εγγραφή

            with connection.cursor() as cursor:
                cursor.execute('exec dbo.TicketSearch @PhoneNo=%s', [request.session.get('clientID')])

                tickets = []

                row = cursor.fetchone() # διαβάζω την πρώτη εγγραφή

                while row != None:
                    ticket = Ticket(row[0], row[1], row[2].strftime("%Y-%m-%d"), row[3], row[4], row[5], row[6], 0)
                    tickets.append(ticket)
                    row = cursor.fetchone() # διαβάζω την επόμενη εγγραφή

            with connection.cursor() as cursor:
                cursor.execute('exec dbo.GetClientInfo @PhoneNo=%s', [request.session.get('clientID')])
                row = cursor.fetchone() # διαβάζω την πρώτη εγγραφή
                info = ClientInfo(row[0],row[1],row[2],row[3],row[4],row[5],row[6])

            context = {'reservations': reservations, 'tickets': tickets, 'info': info}
            return render(request, 'FlightManagementSystem/ClientDashboard.html', context)

```

Εικόνα 399. View ταμπλό πελάτη

```

al Help views.py - ptyxiaki - Visual Studio Code
views.py x
FlightManagementSystem > views.py > ...
1673 # -----REPORT-----
1674
1675 def report(request):
1676     if request.session.get('currentUser') is None:
1677         return redirect('/')
1678     else:
1679         return render(request, 'FlightManagementSystem/Report.html', {})

```

Εικόνα 400. View σελίδας συγκεντρωτικής αναφοράς πτήσεων

```

al Help views.py - ptyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > ...
1681 @csrf_exempt
1682 def getReport(request):
1683     dateFrom = request.POST['dateFrom']
1684     dateTo = request.POST['dateTo']
1685
1686     with connection.cursor() as cursor:
1687         cursor.execute('exec dbo.GetReport @dateFrom=%s, @dateTo=%s', [dateFrom, dateTo])
1688
1689         occupancies = []
1690
1691         row = cursor.fetchone() # διαβάζω την πρώτη εγγραφή
1692
1693         while row != None:
1694             occupancy = Occupancy(row[0], row[1], row[2], row[3])
1695             occupancies.append(occupancy.toJson())
1696             row = cursor.fetchone() # διαβάζω την επόμενη εγγραφή
1697
1698     return JsonResponse(json.dumps(occupancies), safe=False)
    
```

Εικόνα 401. View εισαγωγής συγκεντρωτικής αναφοράς πτήσεων

```

al Help views.py - ptyxiaki - Visual Studio Code
views.py X
FlightManagementSystem > views.py > ...
1700 # -----WELCOME-----
1701
1702 def welcome(request):
1703     json_dict = json.loads(request.session['currentUser'])
1704     currentUser = User(**json_dict)
1705
1706     context = {'currentUser': currentUser}
1707
1708     if currentUser.isAdministrator:
1709         return render(request, 'FlightManagementSystem/WelcomeAdmin.html', context)
1710     elif currentUser.isClient:
1711         return render(request, 'FlightManagementSystem/WelcomeClient.html', context)
1712     elif currentUser.isAgent:
1713         return render(request, 'FlightManagementSystem/WelcomeAgent.html', context)
    
```

Εικόνα 402. View σελίδας καλωσορίσματος χρήστη

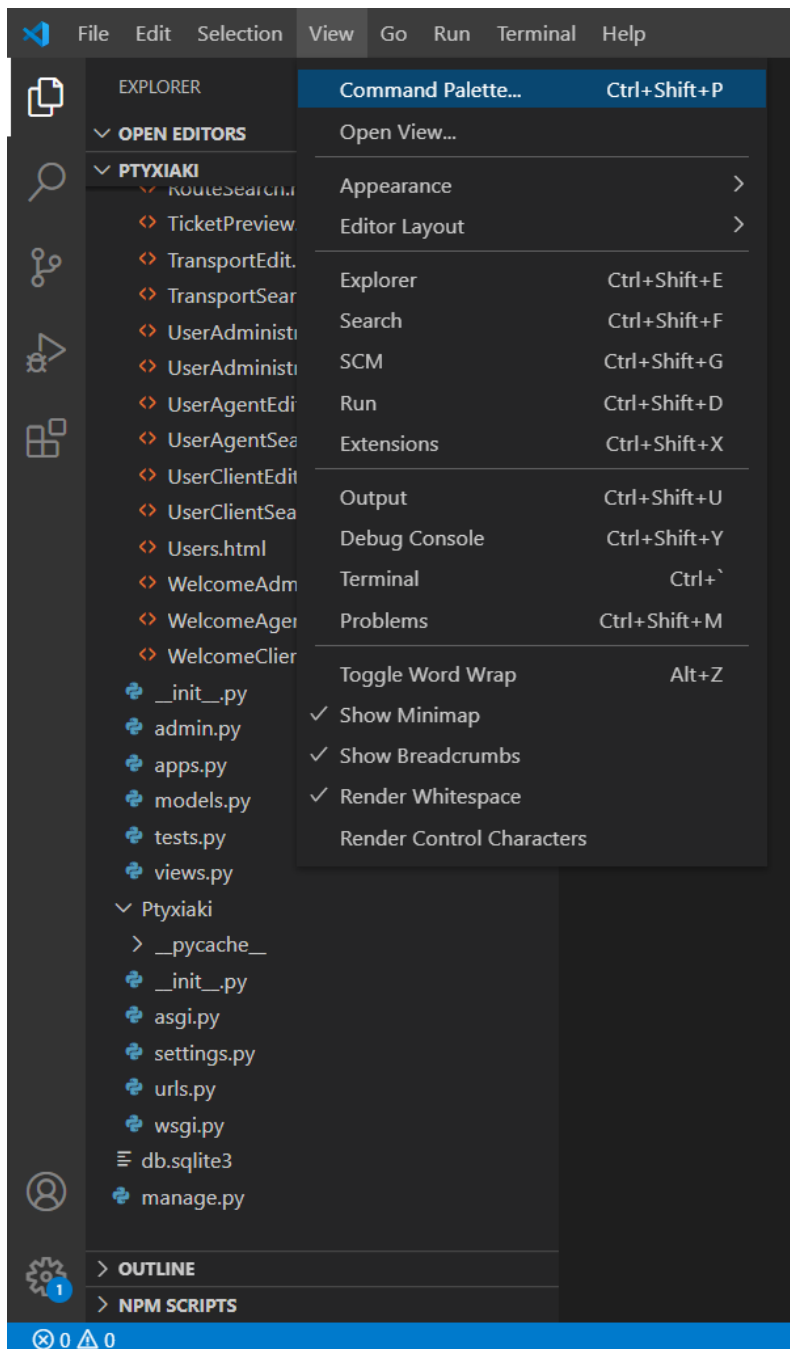
4 Αποτελέσματα

4.1 Εκτέλεση της εφαρμογής

Πλέον, για να εκτελέσουμε την εφαρμογή, θα πρέπει να φορτώσουμε την αρχική ιστοσελίδα (<http://127.0.0.1:8000>). Για να το κάνουμε αυτό, θα πρέπει να εκτελέσουμε τα εξής βήματα:

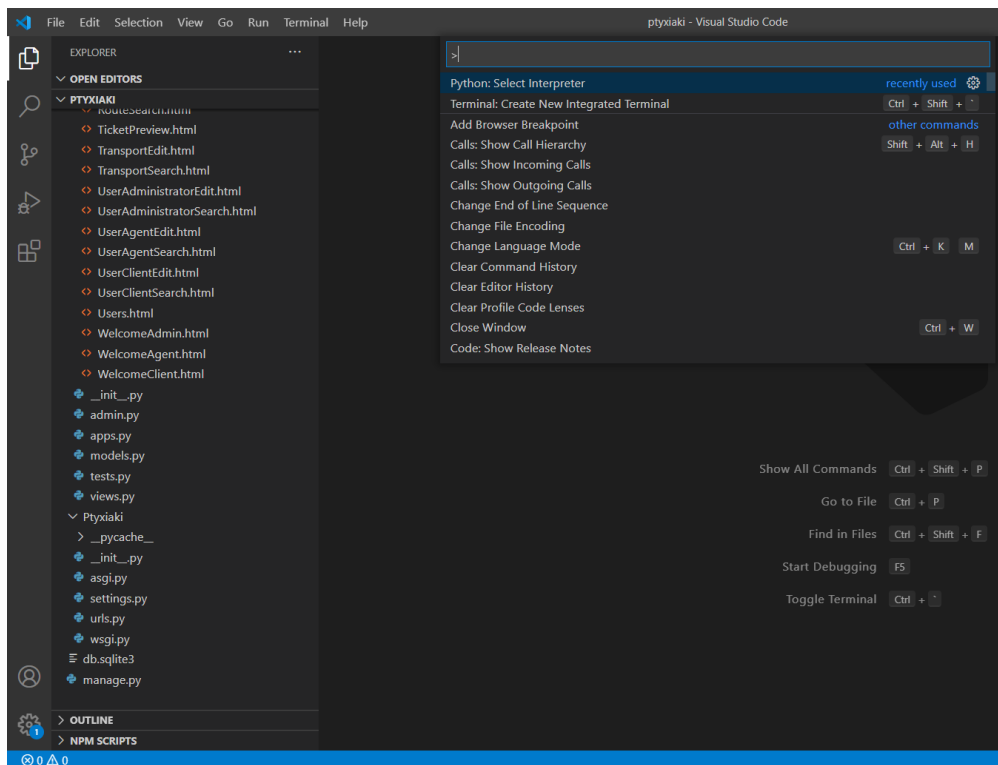
Βήμα 1^ο: Ανοίγουμε το πρόγραμμα Visual Studio Code.

Βήμα 2^ο: Πάμε στη καρτέλα View > Command Palette.



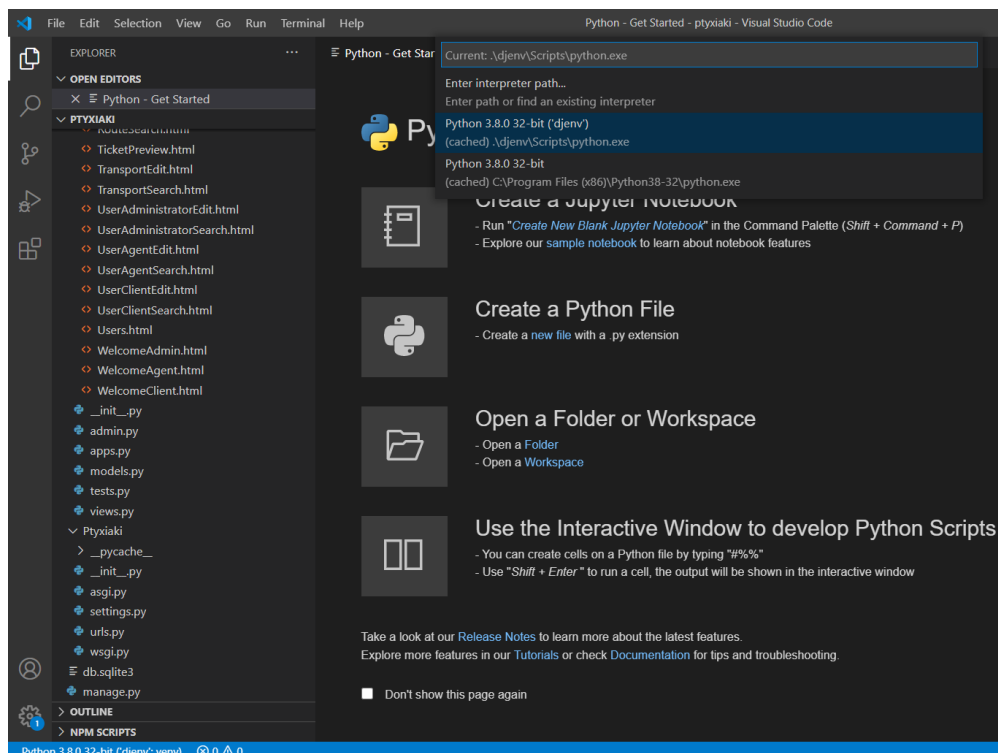
Εικόνα 403. Εκτέλεση εφαρμογής Βήμα 1ο & 2ο

Βήμα 3^ο: Επιλέγουμε την επιλογή: Python: Select Interpreter.



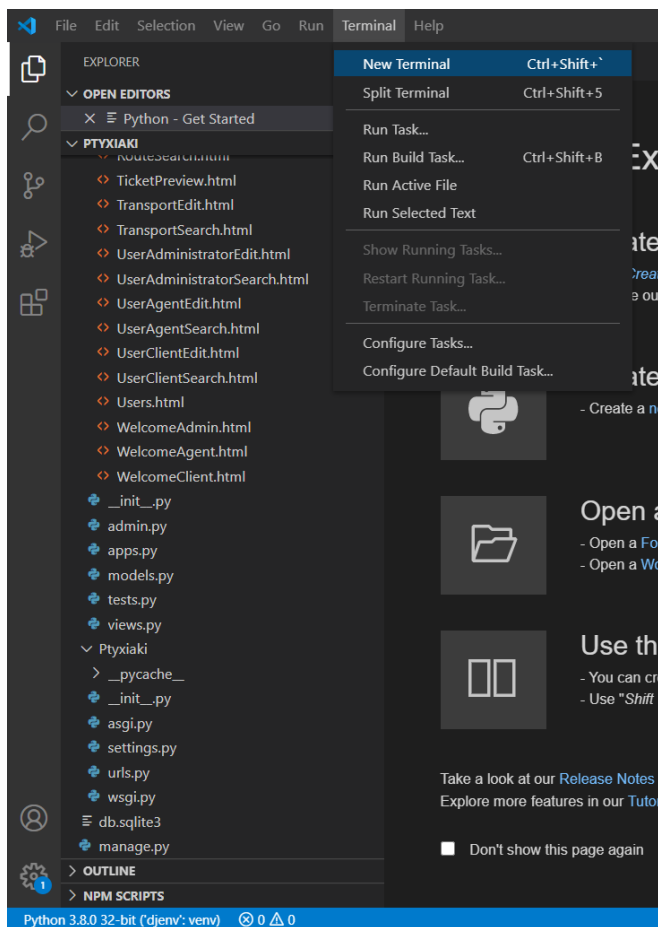
Εικόνα 404. Εκτέλεση εφαρμογής Βήμα 3ο

Βήμα 4^ο: Ύστερα, επιλέγουμε την επιλογή: Python 3.8.0 32-bit ('djenv').



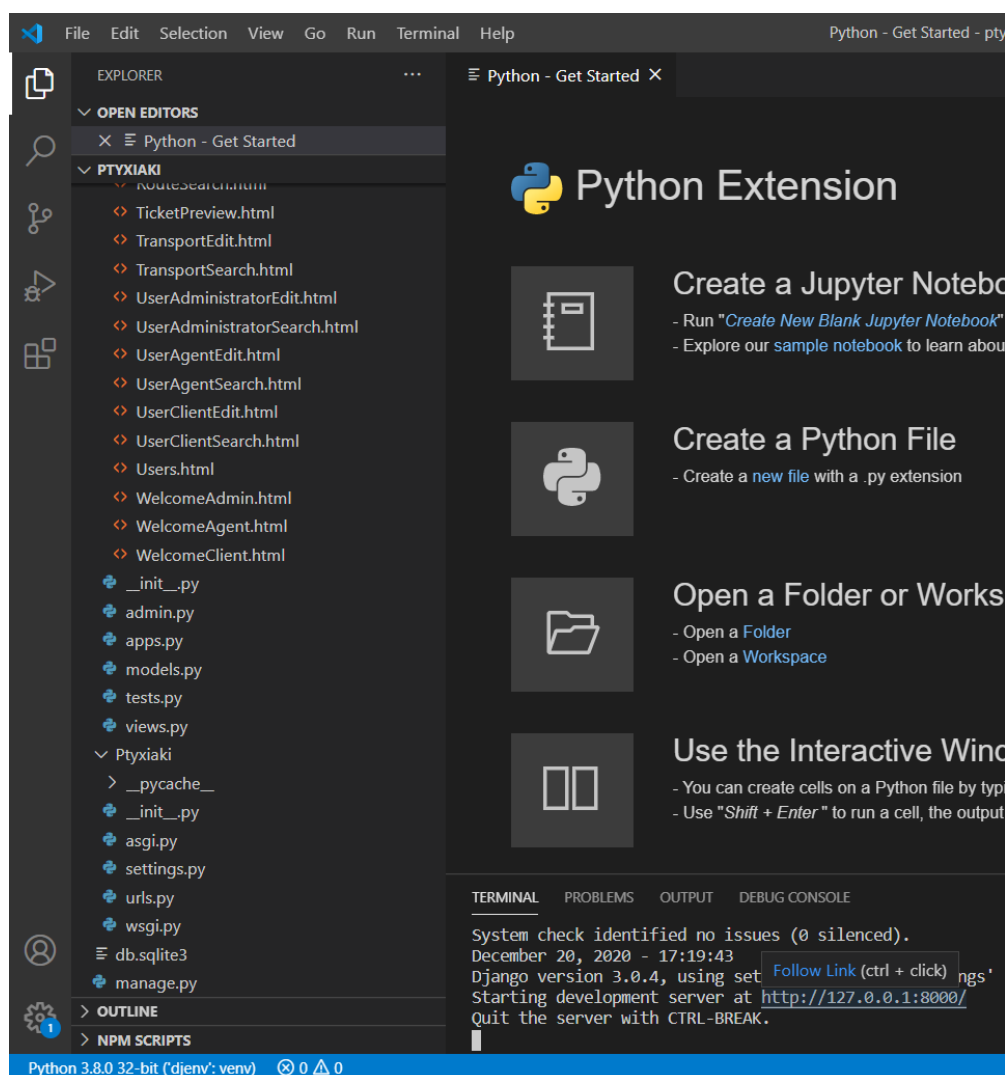
Εικόνα 405. Εκτέλεση εφαρμογής Βήμα 4ο

Βήμα 5^ο: Αφού φορτώσαμε πλέον το εικονικό περιβάλλον της Django (djenv), πάμε στη καρτέλα Terminal > New Terminal.



Εικόνα 406. Εκτέλεση εφαρμογής Βήμα 5ο

Βήμα 6^ο: Τέλος, αφού φορτώσουμε και το νέο τερματικό, γράφουμε την εξής εντολή: `python manage.py runserver`. Με το που τρέξει το τερματικό και δεν υπάρξουν σφάλματα, θα εμφανιστεί το URL της εφαρμογής (<http://127.0.0.1:8000>), στο οποίο πατάμε Ctrl και αριστερό κλικ, για να ανοίξουμε την εφαρμογή στον προεπιλεγμένο περιηγητή διαδικτύου (ο οποίος εδώ, είναι ο Google Chrome).

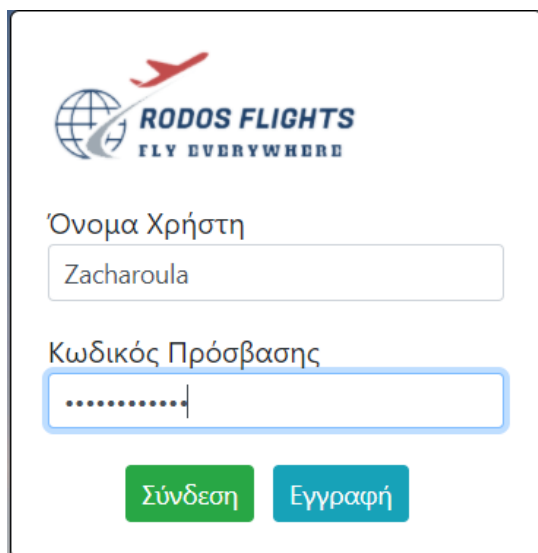


Εικόνα 407. Εκτέλεση εφαρμογής Βήμα 6ο

4.2 Χρήση της εφαρμογής

Είσοδος χρήστη

Ένας χρήστης, προκειμένου να κάνει χρήση του πληροφοριακού συστήματος, πρέπει να πραγματοποιήσει είσοδο στο σύστημα, κάνοντας χρήση ενός ονόματος χρήστη (username) και κωδικού πρόσβασης (password). Με αυτόν τον τρόπο, εξακριβώνεται η αυθεντικότητα του χρήστη. Σε περίπτωση που ο χρήστης δεν εισάγει το όνομα χρήστη και τον κωδικό πρόσβασης, εμφανίζεται σχετικό μήνυμα. Επίσης, αν τα στοιχεία που εισάγει είναι λανθασμένα, εμφανίζεται εξίσου σχετικό μήνυμα.



Εικόνα 408. Είσοδος χρήστη

ΠΡΟΣΟΧΗ

- Συμπληρώστε το όνομα χρήστη.
- Συμπληρώστε τον κωδικό πρόσβασης.

OK

Εικόνα 409. Έλεγχος κενών στοιχείων εισόδου χρήστη

ΠΡΟΣΟΧΗ

Τα στοιχεία εισόδου είναι λανθασμένα.
Παρακαλώ, προσπαθήστε ξανά.

OK

Εικόνα 410. Έλεγχος εγκυρότητας στοιχείων εισόδου χρήστη

Εγγραφή χρήστη

Ένας χρήστης-πελάτης, μπορεί να κάνει εγγραφή πατώντας το κουμπί «Εγγραφή». Αφού πατήσει το κουμπί «Εγγραφή», θα πρέπει να συμπληρώσει τα στοιχεία του.

RODOS FLIGHTS
FLY EVERYWHERE

Νέος χρήστης

Αποθήκευση Αποθήκευση & Κλείσιμο

Τηλέφωνο

Όνομα

Επώνυμο

Οδός Αριθμός

Πόλη

Όνομα χρήστη

Κωδικός πρόσβασης

Επιβεβαίωση

E-mail


Εικόνα 411. Εγγραφή χρήστη-πελάτη

Αρχική οθόνη διαχειριστή

Στη παρακάτω εικόνα, φαίνεται η αρχική σελίδα του χρήστη-διαχειριστή, δηλαδή η σελίδα την οποία θα δει ο διαχειριστής αφού κάνει είσοδο. Όποιες ενέργειες επιθυμεί να κάνει ο κάθε χρήστης, μπορεί όποια στιγμή θέλει, να επιστρέψει στην αρχική σελίδα, πατώντας το κουμπί με το λογότυπο της αεροπορικής εταιρείας, που βρίσκεται στην πάνω αριστερή γωνία της σελίδας.



Εικόνα 412. Αρχική σελίδα διαχειριστή & λογότυπο-κουμπί μετάβασης στην αρχική σελίδα




Τέλος, ο χρήστης μπορεί οποιαδήποτε στιγμή να κάνει έξοδο από την εφαρμογή, πατώντας το εικονίδιο , που βρίσκεται στην πάνω δεξιά γωνία της σελίδας και να γυρίσει στη σελίδα εισόδου χρήστη.

Λίστα πιλότων

Στη σελίδα αυτή, ο διαχειριστής μπορεί να δει τα στοιχεία όλων των πιλότων, που είναι καταχωρημένοι στο σύστημα, καθώς και να καταχωρήσει νέους πιλότους.

RODOS FLIGHTS FLY EVERYWHERE		GRIGORIS GRIGORIADIS (GREGORY96)								
ΠΙΛΟΤΟΙ		ΔΗΜΙΟΥΡΓΙΑ								
Ενέργεια	Αριθμός Ταυτότητας	Όνομα	Επώνυμο	Οδός	Αριθμός	Πόλη	Τηλέφωνο	Ημερομηνία Γέννησης	Ώρες Πτήσης	
	32432	dfsdfsdfsdfsdf	3423	fsdfs3432	34	sdfs	32432ds	2020-02-12	43	
	AF415636	Vasilakis	Vasilakis	Apterwn	16	Irakleio	2810314942	1990-05-31	8	
	AG226677	Pavlos	Pavlidis	Trikoupi	42	Rodos	2241033667	1991-03-28	8	
	AX900145	Anastasios	Anastasiadis	Pernou	17	Athina	2103614529	1996-09-20	6	
	gdgfg	gdgd	dfg	dgdgd	45353	gdgfgdf	dfgd	2020-09-04	454	

Εικόνα 413. Λίστα πιλότων

Από τη σελίδα αυτή, ο χρήστης μπορεί να προβάλλει τα στοιχεία ενός πιλότου σε κατάσταση προβολής (view mode), χωρίς τη δυνατότητα τροποποίησής τους, πατώντας το εικονίδιο . Μπορεί να επεξεργαστεί τα στοιχεία ενός πιλότου, πατώντας το εικονίδιο . Τέλος, μπορεί να διαγράψει ένα πιλότο με το εικονίδιο , αφού επιβεβαιώσει τη διαγραφή.

RODOS FLIGHTS FLY EVERYWHERE		GRIGORIS GRIGORIADIS (GREGORY96)	
		<input type="button" value="Επεξεργασία"/>	<input type="button" value="Διαγραφή"/>
Αριθμός Ταυτότητας	<input type="text" value="AX900145"/>		
Όνομα	<input type="text" value="Anastasios"/>		
Επώνυμο	<input type="text" value="Anastasiadis"/>		
Οδός	<input type="text" value="Pernou"/>	Αριθμός	<input type="text" value="17"/>
Πόλη	<input type="text" value="Athina"/>		
Τηλέφωνο	<input type="text" value="2103614529"/>		
Ημερομηνία Γέννησης	<input type="text" value="20/09/1996"/>		
Ώρες Πτήσεων	<input type="text" value="6"/>		

Εικόνα 414. Προβολή στοιχείων πιλότου

Αριθμός Ταυτότητας: AX900145

Όνομα: Anastasios

Επώνυμο: Anastasiadis

Οδός: Pernou

Αριθμός: 17

Πόλη: Athina

Τηλέφωνο: 2103614529

Ημερομηνία Γέννησης: 20/09/1996

Ώρες Πτήσεων: 6

Εικόνα 415. Επεξεργασία στοιχείων πιλότου

ΕΠΙΒΕΒΑΙΩΣΗ

Θέλετε σίγουρα να διαγράψετε το πιλότο;

Ναι Όχι

Εικόνα 416. Επιβεβαίωση διαγραφής πιλότου

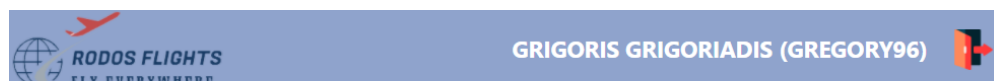
Λίστα αεροσυνοδών

Στη σελίδα αυτή, ο διαχειριστής μπορεί να δει τα στοιχεία όλων των αεροσυνοδών, που είναι καταχωρημένοι στο σύστημα, καθώς και να καταχωρήσει νέους αεροσυνοδούς.

Ενέργεια	Αριθμός Ταυτότητας	Όνομα	Επώνυμο	Οδός	Αριθμός	Πόλη	Τηλέφωνο	Ημερομηνία Γέννησης	Χρόνια Υπηρεσίας
	A5459789	Ilias	Zachariadis	Lakonias	25	Athina	2105674873	1984-04-24	4
	jyutre	retrfyuiy	etrytyuio	qretrfyguhilj	786435	sedrfyguhjlo	876645	2020-01-01	4587
	kjihyutrsy	iuoiop	ytfuygihjoj	lkjibhcgf	45768	fyguhijok	487	2020-01-01	478
	kjiutr	qwteyrdt	zq3we4r5yt6	tyru5ye4	7845	wetrytyi	78451	2019-02-04	78
	mkhjghdfr	quyuiopj	qwetrtyguhi	qwttryguh	45/7	yuio	45	2020-01-01	450

Εικόνα 417. Λίστα αεροσυνοδών

Από τη σελίδα αυτή, ο χρήστης μπορεί να προβάλλει τα στοιχεία ενός αεροσυνοδού σε κατάσταση προβολής (view mode), χωρίς τη δυνατότητα τροποποίησής τους, πατώντας το εικονίδιο . Μπορεί να επεξεργαστεί τα στοιχεία ενός αεροσυνοδού, πατώντας το εικονίδιο . Τέλος, μπορεί να διαγράψει ένα αεροσυνοδό με το εικονίδιο , αφού επιβεβαιώσει τη διαγραφή.



Επεξεργασία Διαγραφή

Αριθμός Ταυτότητας AS459789

Όνομα Ilias

Επώνυμο Zachariadis

Οδός Lakonias Αριθμός 25

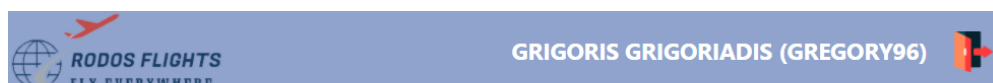
Πόλη Athina

Τηλέφωνο 2105674873

Ημερομηνία Γέννησης 24/04/1984

Χρόνια Υπηρεσίας 4

Εικόνα 418. Προβολή στοιχείων αεροσυνοδού



Αποθήκευση Αποθήκευση & Κλείσιμο Διαγραφή Άκυρο

Αριθμός Ταυτότητας AS459789

Όνομα Ilias

Επώνυμο Zachariadis

Οδός Lakonias Αριθμός 25

Πόλη Athina

Τηλέφωνο 2105674873

Ημερομηνία Γέννησης 24/04/1984

Χρόνια Υπηρεσίας 4

Εικόνα 419. Επεξεργασία στοιχείων αεροσυνοδού

ΕΠΙΒΕΒΑΙΩΣΗ

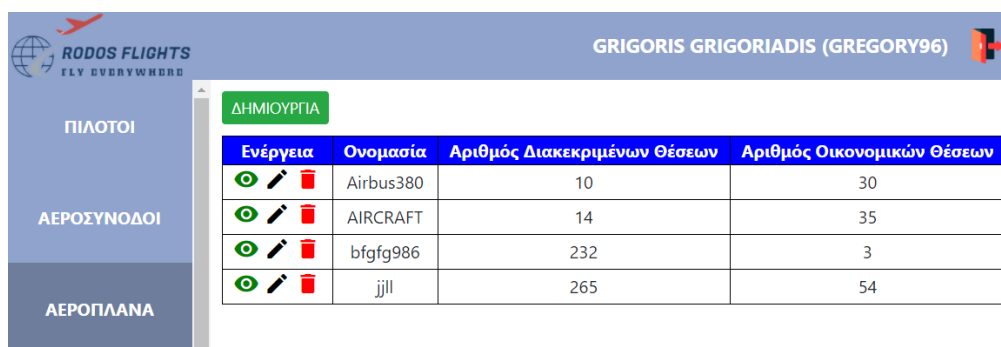
Θέλετε σίγουρα να διαγράψετε την αεροσυνοδό;

Ναι Όχι

Εικόνα 420. Επιβεβαίωση διαγραφής αεροσυνοδού

Λίστα αεροπλάνων

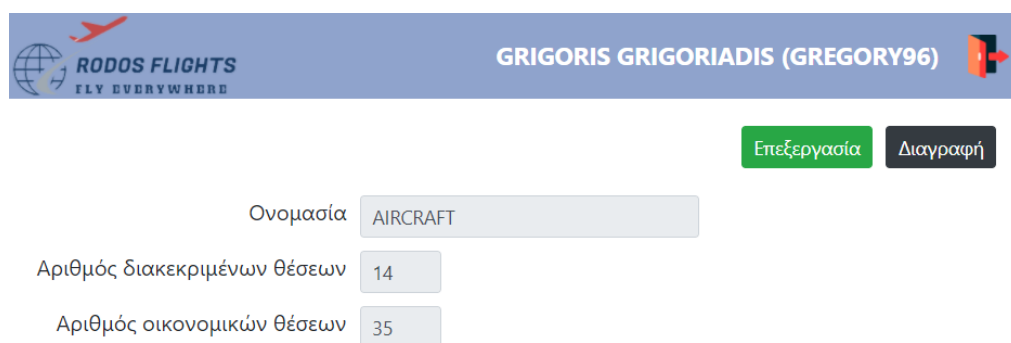
Στη σελίδα αυτή, ο διαχειριστής μπορεί να δει τα στοιχεία όλων των αεροπλάνων, που είναι καταχωρημένα στο σύστημα, καθώς και να καταχωρήσει νέα αεροπλάνα.



Ενέργεια	Όνομασία	Αριθμός Διακεκριμένων Θέσεων	Αριθμός Οικονομικών Θέσεων
	Airbus380	10	30
	AIRCRAFT	14	35
	bfgfg986	232	3
	jll	265	54

Εικόνα 421. Λίστα αεροπλάνων

Από τη σελίδα αυτή, ο χρήστης μπορεί να προβάλλει τα στοιχεία ενός αεροπλάνου σε κατάσταση προβολής (view mode), χωρίς τη δυνατότητα τροποποίησής τους, πατώντας το εικονίδιο . Μπορεί να επεξεργαστεί τα στοιχεία ενός αεροπλάνου, πατώντας το εικονίδιο . Τέλος, μπορεί να διαγράψει ένα αεροπλάνο με το εικονίδιο , αφού επιβεβαιώσει τη διαγραφή.



Ονομασία

Αριθμός διακεκριμένων θέσεων

Αριθμός οικονομικών θέσεων

Εικόνα 422. Προβολή στοιχείων αεροπλάνου



Ονομασία

Αριθμός διακεκριμένων θέσεων

Αριθμός οικονομικών θέσεων

Εικόνα 423. Επεξεργασία στοιχείων αεροπλάνου

ΕΠΙΒΕΒΑΙΩΣΗ

Θέλετε σίγουρα να διαγράψετε το αεροπλάνο;

Ναι

Όχι

Εικόνα 424. Επιβεβαίωση διαγραφής αεροπλάνου

Λίστα ταξιδιωτικών πρακτόρων

Στη σελίδα αυτή, ο διαχειριστής μπορεί να δει τα στοιχεία όλων των ταξιδιωτικών πρακτόρων, που είναι καταχωρημένα στο σύστημα, καθώς και να καταχωρήσει νέους ταξιδιωτικούς πράκτορες.

Ενέργεια	Αριθμός Ταυτότητας	Όνομα	Επώνυμο	Οδός	Αριθμός	Πόλη	Κωδικός
	AP785412	FANOURLS	TATAKIS	ATHITAKI	48	IRAKLEIO	363539
	AQ774455	Lampros	Lamprakis	Grigoriou	74	Rodos	365252

Εικόνα 425. Λίστα ταξιδιωτικών πρακτόρων

Από τη σελίδα αυτή, ο χρήστης μπορεί να προβάλει τα στοιχεία ενός ταξιδιωτικού πράκτορα σε κατάσταση προβολής (view mode), χωρίς τη δυνατότητα τροποποίησής τους, πατώντας το εικονίδιο . Μπορεί να επεξεργαστεί τα στοιχεία ενός ταξιδιωτικού πράκτορα, πατώντας το εικονίδιο . Τέλος, μπορεί να διαγράψει έναν ταξιδιωτικό πράκτορα με το εικονίδιο , αφού επιβεβαιώσει τη διαγραφή.

Αριθμός Ταυτότητας

Όνομα

Επώνυμο

Οδός Αριθμός

Πόλη

Κωδικός

Εικόνα 426. Προβολή στοιχείων ταξιδιωτικού πράκτορα

RODOS FLIGHTS
FLY EVERYWHERE

GRIGORIS GRIGORIADIS (GREGORY96)

Αποθήκευση Αποθήκευση & Κλείσιμο Διαγραφή Άκυρο

Αριθμός Ταυτότητας

Όνομα

Επώνυμο

Οδός Αριθμός

Πόλη

Κωδικός

Εικόνα 427. Επεξεργασία στοιχείων ταξιδιωτικού πράκτορα

ΕΠΙΒΕΒΑΙΩΣΗ

Θέλετε σίγουρα να διαγράψετε τον ταξιδιωτικό πράκτορα;

Ναι

Όχι

Εικόνα 428. Επιβεβαίωση διαγραφής ταξιδιωτικού πράκτορα

Λίστα διοικητικών υπαλλήλων

Στη σελίδα αυτή, ο διαχειριστής μπορεί να δει τα στοιχεία όλων των διοικητικών υπαλλήλων, που είναι καταχωρημένα στο σύστημα, καθώς και να καταχωρήσει νέους διοικητικούς υπαλλήλους.

RODOS FLIGHTS
FLY EVERYWHERE

GRIGORIS GRIGORIADIS (GREGORY96)

ΠΙΛΟΤΟΙ

ΔΗΜΙΟΥΡΓΙΑ

Ενέργεια	Αριθμός Ταυτότητας	Όνομα	Επώνυμο	Οδός	Αριθμός	Πόλη	Τηλέφωνο	Ημερομηνία Γέννησης	Εκπαίδευση	Χρόνια Υπηρεσίας
	AF152478	Zacharias	Papantwniou	Grigoraki	52	Lasithi	2831036452	1980-01-12	Τριτοβάθμια	11

ΑΕΡΟΣΥΝΟΔΟΙ

Εικόνα 429. Λίστα διοικητικών υπαλλήλων

Από τη σελίδα αυτή, ο χρήστης μπορεί να προβάλλει τα στοιχεία ενός διοικητικού υπαλλήλου σε κατάσταση προβολής (view mode), χωρίς τη δυνατότητα τροποποίησής τους, πατώντας το εικονίδιο . Μπορεί να επεξεργαστεί τα στοιχεία ενός διοικητικού υπαλλήλου, πατώντας το εικονίδιο . Τέλος, μπορεί να διαγράψει έναν διοικητικό υπάλληλο με το εικονίδιο , αφού επιβεβαιώσει τη διαγραφή.



Επεξεργασία Διαγραφή

Αριθμός Ταυτότητας

Όνομα

Επώνυμο

Οδός Αριθμός

Πόλη

Τηλέφωνο

Ημερομηνία Γέννησης

Εκπαίδευση

Χρόνια Υπηρεσίας

Εικόνα 430. Προβολή στοιχείων διοικητικού υπαλλήλου



Αποθήκευση Αποθήκευση & Κλείσιμο Διαγραφή Άκυρο

Αριθμός Ταυτότητας

Όνομα

Επώνυμο

Οδός Αριθμός

Πόλη

Τηλέφωνο

Ημερομηνία Γέννησης

Εκπαίδευση

Χρόνια Υπηρεσίας

Εικόνα 431. Επεξεργασία στοιχείων διοικητικού υπαλλήλου

ΕΠΙΒΕΒΑΙΩΣΗ

Θέλετε σίγουρα να διαγράψετε τον διοικητικό υπάλληλο;

Ναι Όχι

Εικόνα 432. Επιβεβαίωση διαγραφής διοικητικού υπαλλήλου

Λίστα χρηστών-πελατών

Στη σελίδα αυτή, ο διαχειριστής μπορεί να δει τα στοιχεία όλων των χρηστών-πελατών, που είναι καταχωρημένα στο σύστημα, καθώς και να καταχωρήσει νέους χρήστες-πελάτες.

Ενέργεια	Όνομα Χρήστη	Αριθμός Τηλεφώνου	Ηλεκτρονική διεύθυνση	Όνομα	Επώνυμο	Οδός	Αριθμός	Πόλη
	KKKKK	56478534	zaza031193@gmail.com	JHKGFHFG	OKJILHUFG	OLJIHUJGF	89	JKLHGF
	Zacharoula93	2241024568	zaza031193@gmail.com	Zacharoula	Zachariou	Rodini	48	Rodos

Εικόνα 433. Λίστα χρηστών-πελατών

Από τη σελίδα αυτή, ο χρήστης μπορεί να προβάλλει τα στοιχεία ενός χρήστη-πελάτη σε κατάσταση προβολής (view mode), χωρίς τη δυνατότητα τροποποίησής τους, πατώντας το εικονίδιο . Μπορεί να επεξεργαστεί τα στοιχεία ενός χρήστη-πελάτη, πατώντας το εικονίδιο . Τέλος, μπορεί να διαγράψει έναν χρήστη-πελάτη με το εικονίδιο , αφού επιβεβαιώσει τη διαγραφή.

GRIGORIS GRIGORIADIS (GREGORY96)

Επεξεργασία Διαγραφή

Τηλέφωνο

Όνομα

Επώνυμο

Οδός Αριθμός

Πόλη

Όνομα χρήστη

E-mail

Εικόνα 434. Προβολή στοιχείων χρήστη-πελάτη

RODOS FLIGHTS
FLY EVERYWHERE

GRIGORIS GRIGORIADIS (GREGORY96)

Αλλαγή Κωδικού Αποθήκευση Αποθήκευση & Κλείσιμο Διαγραφή Άκυρο

Τηλέφωνο 2241024568

Όνομα Zacharoula

Επώνυμο Zachariou

Οδός Rodini Αριθμός 48

Πόλη Rodos

Όνομα χρήστη Zacharoula93

E-mail zaza031193@gmail.com

Εικόνα 435. Επεξεργασία στοιχείων χρήστη-πελάτη

ΕΠΙΒΕΒΑΙΩΣΗ

Θέλετε σίγουρα να διαγράψετε τον πελάτη;

Ναι Όχι

Εικόνα 436. Επιβεβαίωση διαγραφής χρήστη-πελάτη

Λίστα χρηστών-ταξιδιωτικών πρακτόρων

Στη σελίδα αυτή, ο διαχειριστής μπορεί να δει τα στοιχεία όλων των χρηστών-ταξιδιωτικών πρακτόρων, που είναι καταχωρημένα στο σύστημα, καθώς και να καταχωρήσει νέους χρήστες-ταξιδιωτικούς πράκτορες.

RODOS FLIGHTS
FLY EVERYWHERE

GRIGORIS GRIGORIADIS (GREGORY96)

ΠΙΛΟΤΟΙ ΠΕΛΑΤΕΣ ΔΗΜΙΟΥΡΓΙΑ

Ενέργεια	Όνομα Χρήστη	Αριθμός Ταυτότητας	Όνομα	Επώνυμο	Οδός	Αριθμός	Πόλη	Κωδικός
	Lampros78	AQ774455	Lampros	Lamprakis	Grigoriou	74	Rodos	365252

Εικόνα 437. Λίστα χρηστών-ταξιδιωτικών πρακτόρων

Από τη σελίδα αυτή, ο χρήστης μπορεί να προβάλλει τα στοιχεία ενός χρήστη-ταξιδιωτικού πράκτορα σε κατάσταση προβολής (view mode), χωρίς τη δυνατότητα τροποποίησής του, πατώντας το εικονίδιο . Μπορεί να επεξεργαστεί τα στοιχεία ενός χρήστη-ταξιδιωτικού πράκτορα, πατώντας το εικονίδιο . Τέλος, μπορεί να διαγράψει έναν χρήστη-ταξιδιωτικό πράκτορα με το εικονίδιο , αφού επιβεβαιώσει τη διαγραφή.



Επεξεργασία

Διαγραφή

Αριθμός Ταυτότητας

Όνομα

Επώνυμο

Οδός

Αριθμός

Πόλη

Κωδικός

Όνομα χρήστη

Εικόνα 438. Προβολή στοιχείων χρήστη-ταξιδιωτικού πράκτορα



Αλλαγή Κωδικού

Αποθήκευση

Αποθήκευση & Κλείσιμο

Διαγραφή

Άκυρο

Αριθμός Ταυτότητας

Όνομα

Επώνυμο

Οδός

Αριθμός

Πόλη

Κωδικός

Όνομα χρήστη

Εικόνα 439. Επεξεργασία στοιχείων χρήστη-ταξιδιωτικού πράκτορα

ΕΠΙΒΕΒΑΙΩΣΗ

Θέλετε σίγουρα να διαγράψετε τον πράκτορα;

Ναι

Όχι

Εικόνα 440. Επιβεβαίωση διαγραφής χρήστη-ταξιδιωτικού πράκτορα

Λίστα χρηστών-διαχειριστών

Στη σελίδα αυτή, ο διαχειριστής μπορεί να δει τα στοιχεία όλων των χρηστών-διαχειριστών, που είναι καταχωρημένα στο σύστημα, καθώς και να καταχωρήσει νέους χρήστες-διαχειριστές.

RODOS FLIGHTS FLY EVERYWHERE		GRIGORIS GRIGORIADIS (GREGORY96)									
ΠΕΛΑΤΕΣ		ΔΗΜΙΟΥΡΓΙΑ									
Ενέργεια	Όνομα Χρήστη	Αριθμός Ταυτότητας	Όνομα	Επώνυμο	Οδός	Αριθμός	Πόλη	Τηλέφωνο	Ημερομηνία Γέννησης		
	GREGORY96	AW457825	GRIGORIS	GRIGORIADIS	STAMATIADI	28	RODOS	2241021475	06-06-1996		
	MARINA	L293	fg	dfg	df	34	dg	d4234	13-07-2020		
	STAMOS93	AS654798	STAMATIS	STAMATIOU	LEMESOU	36	KARPATIOS	2241032456	03-07-1993		
	tonytgh	kkkk	fgdgd	dgdgdf	gdgdgd	24323	dfgdf	dfgdgfdgd	03-07-2020		
	ZZZZZ	RT754215	DFYUKL	TFGHJU	KJLUYT	11	ERDFHGJH	45784521	07-08-1978		

Εικόνα 441. Λίστα χρηστών-διαχειριστών

Από τη σελίδα αυτή, ο χρήστης μπορεί να προβάλλει τα στοιχεία ενός χρήστη-διαχειριστή σε κατάσταση προβολής (view mode), χωρίς τη δυνατότητα τροποποίησής τους, πατώντας το εικονίδιο . Μπορεί να επεξεργαστεί τα στοιχεία ενός χρήστη-διαχειριστή, πατώντας το εικονίδιο . Τέλος, μπορεί να διαγράψει έναν χρήστη-διαχειριστή με το εικονίδιο , αφού επιβεβαιώσει τη διαγραφή.

Επεξεργασία Διαγραφή

Αριθμός Ταυτότητας

Όνομα

Επώνυμο

Οδός

Πόλη

Τηλέφωνο

Ημερομηνία Γέννησης

Όνομα χρήστη

Αριθμός

Εικόνα 442. Προβολή στοιχείων χρήστη-διαχειριστή



Αλλαγή Κωδικού | Αποθήκευση | Αποθήκευση & Κλείσιμο | Διαγραφή | Άκυρο

Αριθμός Ταυτότητας: AW457825
 Όνομα: GRIGORIS
 Επώνυμο: GRIGORIADIS
 Οδός: STAMATIADI | Αριθμός: 28
 Πόλη: RODOS
 Τηλέφωνο: 2241021475
 Ημερομηνία Γέννησης: 06/06/1996
 Όνομα χρήστη: GREGORY96

Εικόνα 443. Επεξεργασία στοιχείων χρήστη-διαχειριστή

ΕΠΙΒΕΒΑΙΩΣΗ

Θέλετε σίγουρα να διαγράψετε τον χρήστη;

Ναι Όχι

Εικόνα 444. Επιβεβαίωση διαγραφής χρήστη-διαχειριστή

Λίστα δρομολογίων

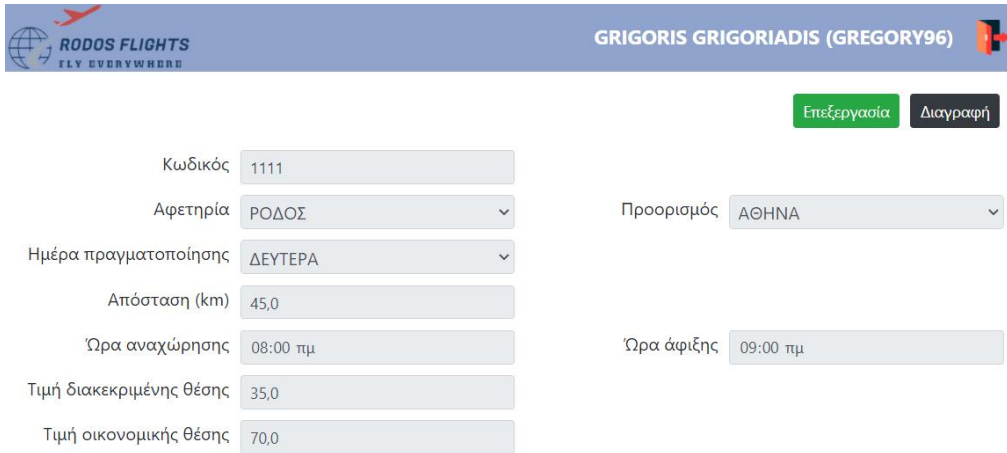
Στη σελίδα αυτή, ο διαχειριστής μπορεί να δει τα στοιχεία όλων των δρομολογίων, που είναι καταχωρημένα στο σύστημα, καθώς και να καταχωρήσει νέα δρομολόγια.



Ενέργεια	Κωδικός	Αφετηρία	Προορισμός	Ημέρα παραγ/σης	Απόσταση (km)	Ώρα αναγ/σης	Ώρα αφίξης	Τιμή διακεκριμένης θέσης (€)	Τιμή οικονομικής θέσης (€)
	1111	ΡΟΔΟΣ	ΑΘΗΝΑ	ΔΕΥΤΕΡΑ	45.0	08:00:00	09:00:00	35.0	70.0
	1112	ΑΘΗΝΑ	ΘΕΣΣΑΛΟΝΙΚΗ	ΔΕΥΤΕΡΑ	70.0	09:45:00	10:45:00	70.0	120.0
	1211	ΑΘΗΝΑ	ΘΕΣΣΑΛΟΝΙΚΗ	ΤΡΙΤΗ	70.0	10:00:00	11:00:00	80.0	45.0
	1222	ΡΟΔΟΣ	ΑΘΗΝΑ	ΤΡΙΤΗ	45.0	08:00:00	09:00:00	90.0	60.0
	8963	ΑΘΗΝΑ	ΗΡΑΚΛΕΙΟ	ΔΕΥΤΕΡΑ	2100.0	14:15:00	14:50:00	150.0	90.0
	8975	ΧΑΝΙΑ	ΧΙΟΣ	ΠΕΜΠΤΗ	85.0	09:00:00	09:50:00	110.0	75.0
	9856	ΡΟΔΟΣ	ΣΑΜΟΣ	ΚΥΡΙΑΚΗ	68.0	11:00:00	11:40:00	80.0	65.0

Εικόνα 445. Λίστα δρομολογίων

Από τη σελίδα αυτή, ο χρήστης μπορεί να προβάλλει τα στοιχεία ενός δρομολογίου σε κατάσταση προβολής (view mode), χωρίς τη δυνατότητα τροποποίησής τους, πατώντας το εικονίδιο . Μπορεί να επεξεργαστεί τα στοιχεία ενός δρομολογίου, πατώντας το εικονίδιο . Τέλος, μπορεί να διαγράψει ένα δρομολόγιο με το εικονίδιο , αφού επιβεβαιώσει τη διαγραφή.



RODOS FLIGHTS
FLY EVERYWHERE

GRIGORIS GRIGORIADIS (GREGORY96)

Επεξεργασία Διαγραφή

Κωδικός: 1111

Αφετηρία: ΡΟΔΟΣ

Προορισμός: ΑΘΗΝΑ

Ημέρα πραγματοποίησης: ΔΕΥΤΕΡΑ

Απόσταση (km): 45,0

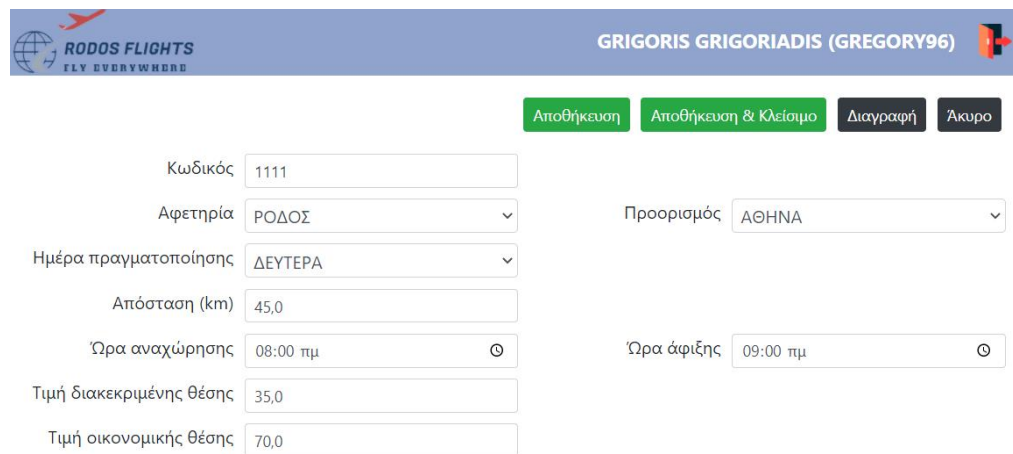
Ώρα αναχώρησης: 08:00 πμ

Ώρα άφιξης: 09:00 πμ

Τιμή διακεκριμένης θέσης: 35,0

Τιμή οικονομικής θέσης: 70,0

Εικόνα 446. Προβολή στοιχείων δρομολογίου



RODOS FLIGHTS
FLY EVERYWHERE

GRIGORIS GRIGORIADIS (GREGORY96)

Αποθήκευση Αποθήκευση & Κλείσιμο Διαγραφή Άκυρο

Κωδικός: 1111

Αφετηρία: ΡΟΔΟΣ

Προορισμός: ΑΘΗΝΑ

Ημέρα πραγματοποίησης: ΔΕΥΤΕΡΑ

Απόσταση (km): 45,0

Ώρα αναχώρησης: 08:00 πμ

Ώρα άφιξης: 09:00 πμ

Τιμή διακεκριμένης θέσης: 35,0

Τιμή οικονομικής θέσης: 70,0

Εικόνα 447. Επεξεργασία στοιχείων δρομολογίου

ΕΠΙΒΕΒΑΙΩΣΗ

Θέλετε σίγουρα να διαγράψετε τη διαδρομή;

Ναι Όχι

Εικόνα 448. Επιβεβαίωση διαγραφής δρομολογίου




Λίστα μεταφορών

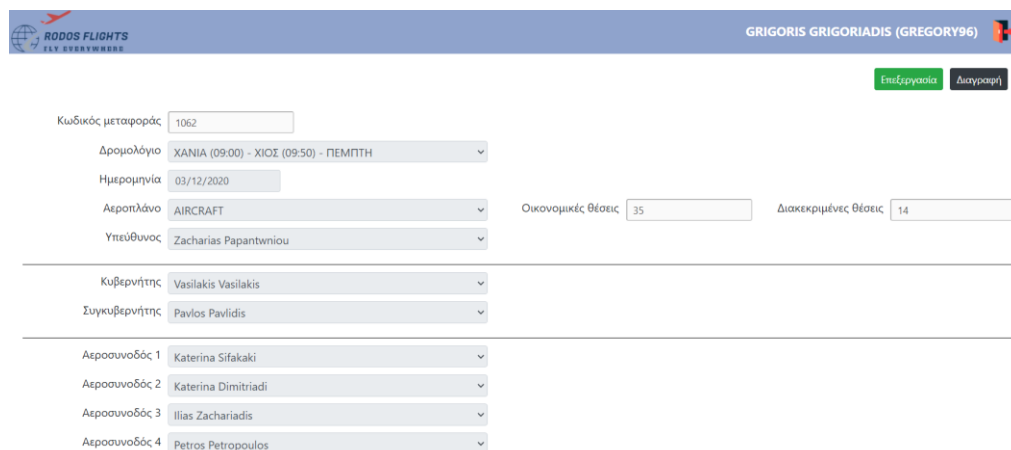
Στη σελίδα αυτή, ο διαχειριστής μπορεί να δει τα στοιχεία όλων των μεταφορών, που είναι καταχωρημένα στο σύστημα, καθώς και να καταχωρήσει νέες μεταφορές.



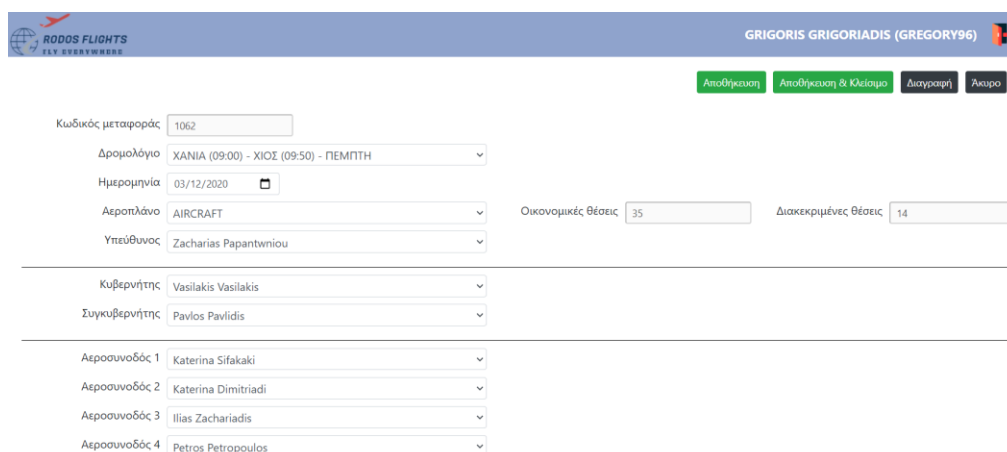
Επίγλυφο	Κωδικός μεταφοράς	Δρομολόγιο	Ημερομηνία	Αεροπλάνο	Κοιμητήριο	Συνεπιβάτης	1η Αεροπορικός	2η Αεροπορικός	3η Αεροπορικός	4η Αεροπορικός	Υπερήκοος
	1062	ΑΘΗΝΑ (09:00) - ΡΟΣ (09:50) - ΠΕΜΠΤΗ	2020-12-03	AIRKRAFT	Vasilakis Vasilakis (AF415636)	Pavlos Pavlides (A0226077)	Katerina Sfakali (AL229011)	Katerina Dimitraki (AR782326)	Ilias Zachariadis (AS459789)	Petros Petropoulos (AZ334966)	Zacharias Papantoniou (AF152478)
	1063	ΡΟΣ (11:00) - ΣΑΜΟΣ (11:40) - ΚΥΡΙΑΚΗ	2020-12-06	Airbus380	Anastasios Anastasiadis (A0900148)	Vasilakis Vasilakis (AF415636)	Petros Petropoulos (AZ334966)	Katerina Sfakali (AL229011)	Ilias Zachariadis (AS459789)	Katerina Dimitraki (AR782326)	Zacharias Papantoniou (AF152478)

Εικόνα 449. Λίστα μεταφορών

Από τη σελίδα αυτή, ο χρήστης μπορεί να προβάλλει τα στοιχεία μιας μεταφοράς σε κατάσταση προβολής (view mode), χωρίς τη δυνατότητα τροποποίησής τους, πατώντας το εικονίδιο . Μπορεί να επεξεργαστεί τα στοιχεία μιας μεταφοράς, πατώντας το εικονίδιο . Τέλος, μπορεί να διαγράψει μια μεταφορά με το εικονίδιο , αφού επιβεβαιώσει τη διαγραφή.



Εικόνα 450. Προβολή στοιχείων μεταφοράς



Εικόνα 451. Επεξεργασία στοιχείων μεταφοράς

ΕΠΙΒΕΒΑΙΩΣΗ

Θέλετε σίγουρα να διαγράψετε τη διαδρομή;

Εικόνα 452. Επιβεβαίωση διαγραφής μεταφοράς

Λίστα κρατήσεων

Στη σελίδα αυτή, ο χρήστης-πελάτης και ο χρήστης-ταξιδιωτικός πράκτορας, μπορούν να δουν τα στοιχεία όλων των κρατήσεων, που είναι καταχωρημένα στο σύστημα, καθώς και να καταχωρήσουν νέες κρατήσεις.

Ενέργεια	Κωδικός κράτησης	Ημερομηνία κράτησης	Ημερομηνία λήξης	Πόλη αναχώρησης	Πόλη άφιξης	Κατηγορία θέσης	Τιμή
	1132	2020-12-03	2020-12-01	ΧΑΝΙΑ	ΧΙΟΣ	Οικονομική	75.0
	1133	2020-12-07	2020-12-05	ΑΘΗΝΑ	ΘΕΣΣΑΛΟΝΙΚΗ	Διακεκριμένη	70.0

Εικόνα 453. Λίστα κρατήσεων

Από τη σελίδα αυτή, ο χρήστης μπορεί να προβάλλει τα στοιχεία μιας κράτησης σε κατάσταση προβολής (view mode), χωρίς τη δυνατότητα τροποποίησής τους, πατώντας το εικονίδιο . Μπορεί να επεξεργαστεί τα στοιχεία μιας κράτησης, πατώντας το εικονίδιο . Τέλος, μπορεί να διαγράψει μια κράτηση με το εικονίδιο , αφού επιβεβαιώσει τη διαγραφή.

Κωδικός κράτησης: 1132
 Ημερομηνία κράτησης: 03/12/2020
 Πόλη αναχώρησης: ΧΑΝΙΑ
 Πόλη άφιξης: ΧΙΟΣ

09:00:00	09:50:00	Economy	Business
ΧΑΝΙΑ	ΧΙΟΣ	€75	€110

Εικόνα 454. Προβολή στοιχείων κράτησης

Κωδικός κράτησης: 1132
 Ημερομηνία κράτησης: 03/12/2020
 Πόλη αναχώρησης: ΧΑΝΙΑ
 Πόλη άφιξης: ΧΙΟΣ

09:00:00	09:50:00	Economy	Business
ΧΑΝΙΑ	ΧΙΟΣ	€75	€110

Εικόνα 455. Επεξεργασία στοιχείων κράτησης

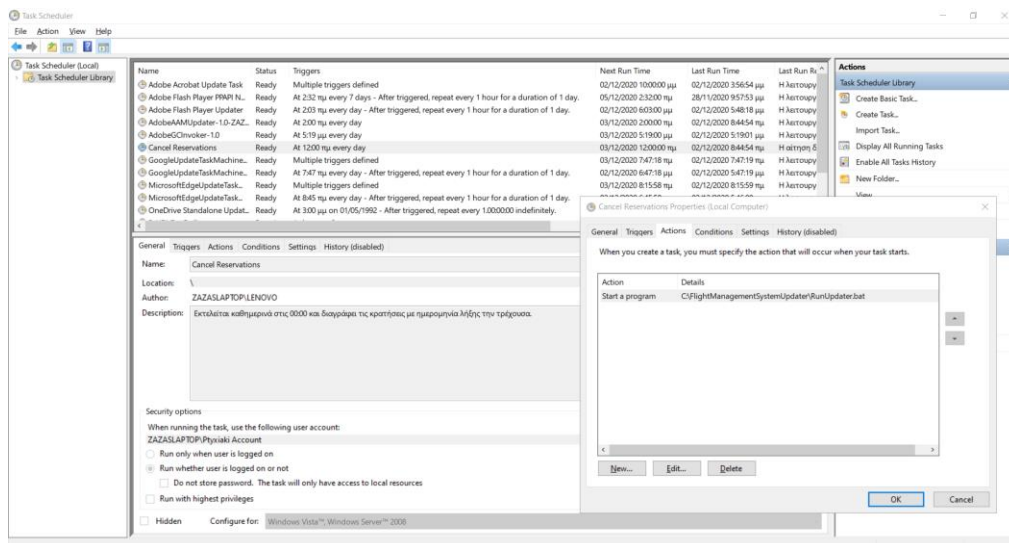
Ενέργεια	Κωδικός κράτησης	Ημερομηνία κράτησης	Ημερομηνία λήξης	Πόλη αναχώρησης	Πόλη άφιξης	Κατηγορία θέσης	Τιμή
	1132	2020-12-03	2020-12-01	ΧΑΝΙΑ	ΧΙΟΣ	Οικονομική	75.0
	1133	2020-12-07	2020-12-05	ΑΘΗΝΑ	ΘΕΣΣΑΛΟΝΙΚΗ	Διακεκριμένη	70.0

Εικόνα 456. Επιβεβαίωση διαγραφής κράτησης

Χρονοπρογραμματισμός κρατήσεων

Για κάθε κράτηση, έχει οριστεί είτε από τον ταξιδιωτικό πράκτορα, είτε από το πληροφοριακό σύστημα, μια ημερομηνία λήξης. Γι' αυτό το σκοπό, έχει υλοποιηθεί μια standalone εφαρμογή σε Python, η οποία εκτελείται καθημερινά στις 00:00 και η οποία διαγράφει όλες τις κρατήσεις, που έχουν ημερομηνία λήξης την τρέχουσα ημερομηνία, με τη βοήθεια ενός stored procedure. Με τον τρόπο αυτό, διατηρούμε στη βάση δεδομένων, κρατήσεις που έχει παρέλθει η ημερομηνία λήξης τους. Η καθημερινή εκτέλεση αυτής της

εφαρμογής, επιτυγχάνεται με τη δημιουργία ενός scheduled task των Windows το οποίο εκτελεί ένα αρχείο δέσμης (RunUpdater.bat).



Εικόνα 457. Χρονοπρογραμματισμένη διεργασία (scheduled task)

Λίστα πελατών

Στη σελίδα αυτή, ο χρήστης-ταξιδιωτικός πράκτορας, μπορεί να δει τα στοιχεία όλων των πελατών, που είναι καταχωρημένα στο σύστημα, καθώς και να καταχωρήσουν νέους πελάτες.

RODOS FLIGHTS FLY EVERYWHERE		Lampros Lamprakis (Lampros78)							
ΠΕΛΑΤΕΣ	ΔΗΜΙΟΥΡΓΙΑ	Ενέργεια	Αριθμός Τηλεφώνου	Ηλεκτρονική διεύθυνση	Όνομα	Επώνυμο	Οδός	Αριθμός	Πόλη
			2104563789	mpampis63@gmail.com	Xaralampos	Tseliou	Pernou	36	Athina
			2241024568	zaza031193@gmail.com	Zacharoula	Zachariou	Rodini	48	Rodos
			2241036645	gregory71@hotmail.com	Grigoris	Athanasiou	Aleksandrou Diakou	14	Rodos
			2310451012	stellitapam@yahoo.com	Stella	Pampali	Megalou Aleksandou	18	Thessaloniki

Εικόνα 458. Λίστα πελατών

Από τη σελίδα αυτή, ο χρήστης μπορεί να προβάλλει τα στοιχεία ενός πελάτη σε κατάσταση προβολής (view mode), χωρίς τη δυνατότητα τροποποίησής τους, πατώντας το εικονίδιο . Μπορεί να επεξεργαστεί τα στοιχεία ενός πελάτη, πατώντας το εικονίδιο . Τέλος, μπορεί να διαγράψει ένα πελάτη με το εικονίδιο , αφού επιβεβαιώσει τη διαγραφή.

RODOS FLIGHTS
FLY EVERYWHERE

Lampros Lamprakis (Lampros78)

Επεξεργασία Διαγραφή

Τηλέφωνο 2241024568

Όνομα Zacharoula

Επώνυμο Zachariou

Οδός Rodini Αριθμός 48

Πόλη Rodos

E-mail zaza031193@gmail.com

Εικόνα 459. Προβολή στοιχείων πελάτη

RODOS FLIGHTS
FLY EVERYWHERE

Lampros Lamprakis (Lampros78)

Αποθήκευση Αποθήκευση & Κλείσιμο Διαγραφή Άκυρο

Τηλέφωνο 2241024568

Όνομα Zacharoula

Επώνυμο Zachariou

Οδός Rodini Αριθμός 48

Πόλη Rodos

E-mail zaza031193@gmail.com

Εικόνα 460. Επεξεργασία στοιχείων πελάτη

RODOS FLIGHTS
FLY EVERYWHERE

Lampros Lamprakis (Lampros78)

ΔΗΜΙΟΥΡΓΙΑ

ΠΕΛΑΤΕΣ

ΚΡΑΤΗΣΕΙΣ

ΕΠΙΒΕΒΑΙΩΣΗ

Θέλετε σίγουρα να διαγράψετε το πελάτη;

Ναι Όχι

Ενέργεια	Αριθμός Τηλεφώνου	Αριθμός	Πόλη
Διαγραφή	104563789	36	Athina
	2241024568	48	Rodos
	2241036645	14	Rodos
	2310451012	18	Thessaloniki

Εικόνα 461. Επιβεβαίωση διαγραφής πελάτη

Καταχώρηση πιλότου

Στη σελίδα αυτή, ο διαχειριστής μπορεί να καταχωρήσει τα στοιχεία ενός νέου πιλότου. Κατά την αποθήκευση, η εφαρμογή ελέγχει αν ο χρήστης έχει συμπληρώσει όλα τα απαραίτητα

πεδία. Σε περίπτωση που δεν τα έχει συμπληρώσει όλα, εμφανίζει σχετικό μήνυμα. Επίσης, ελέγχει αν ο αριθμός ταυτότητας είναι ήδη καταχωρημένος και αντιστοιχεί σε άλλο υπάλληλο. Αφού γίνει η αποθήκευση του πιλότου, η εφαρμογή ανοίγει την σελίδα του πιλότου σε κατάσταση προβολής (view mode). Σε περίπτωση όμως, που ο χρήστης πάτησε το κουμπί «Αποθήκευση & κλείσιμο», η εφαρμογή αποθηκεύει τον πιλότο και κλείνει την σελίδα. Και στις δύο περιπτώσεις, ενημερώνει ταυτόχρονα την σελίδα αναζήτησης των πιλότων.

RODOS FLIGHTS
FLY EVERYWHERE

GRIGORIS GRIGORIADIS (GREGORY96)

Αποθήκευση Αποθήκευση & Κλείσιμο

Αριθμός Ταυτότητας

Όνομα

Επώνυμο

Οδός

Πόλη

Τηλέφωνο

Αριθμός

Ημερομηνία Γέννησης ηη/μμ/εεεε

Ώρες Πτήσεων

Εικόνα 462. Δημιουργία νέου πιλότου

ΠΡΟΣΟΧΗ

- Συμπληρώστε τον αριθμό ταυτότητας.
- Συμπληρώστε το όνομά σας.
- Συμπληρώστε το επώνυμό σας.
- Συμπληρώστε το όνομα της πόλης.
- Συμπληρώστε τον αριθμό τηλεφώνου σας.
- Συμπληρώστε την ημερομηνία γέννησής σας.

OK

Εικόνα 463. Έλεγχος εγκυρότητας στοιχείων πιλότου

ΠΡΟΣΟΧΗ

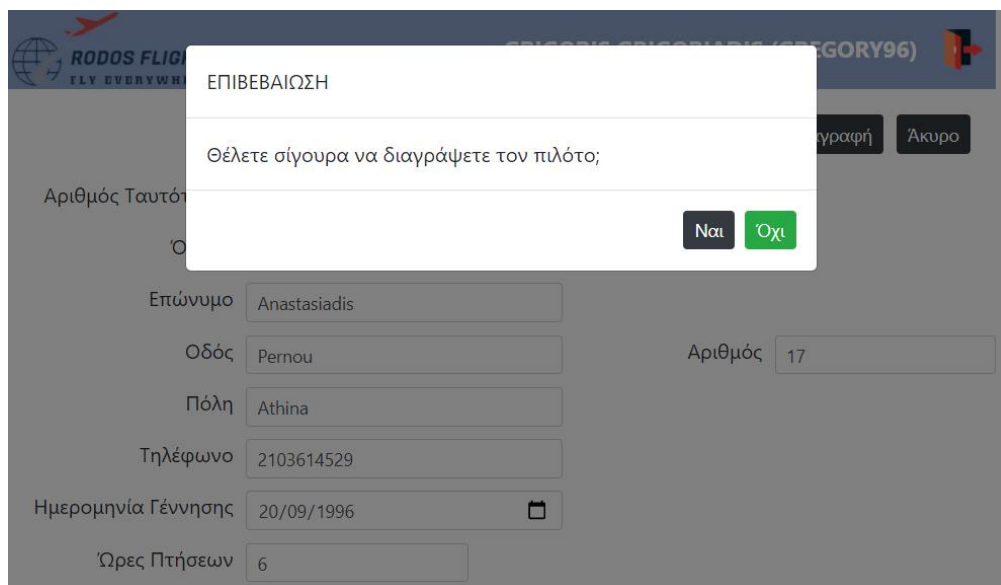
- Ο αριθμός ταυτότητας υπάρχει ήδη.

OK

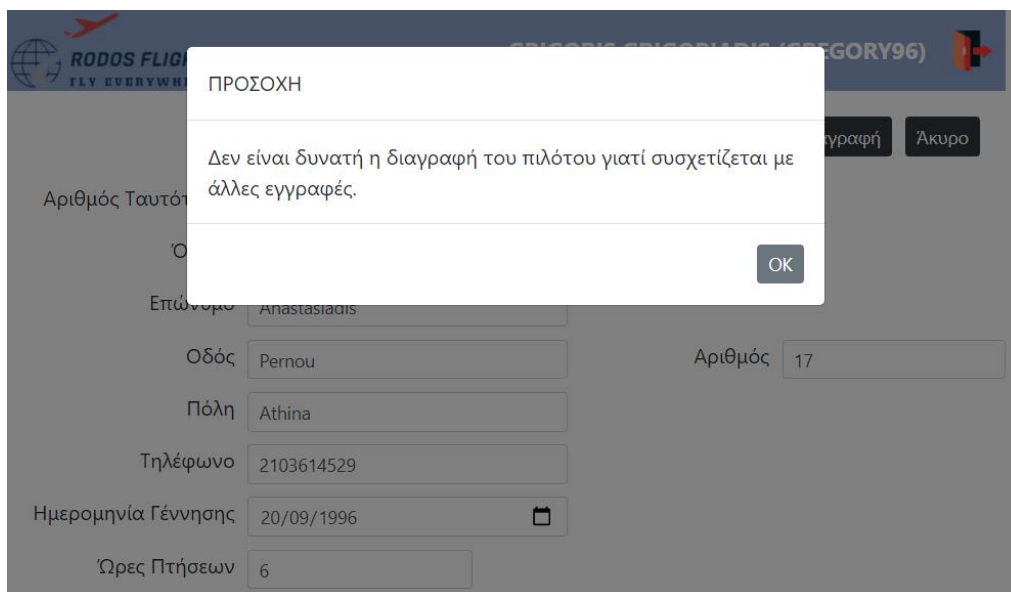
Εικόνα 464. Έλεγχος ήδη καταχωρημένου πιλότου

Διαγραφή πιλότου

Ο διαχειριστής μπορεί να διαγράψει έναν πιλότο, πατώντας το κουμπί «Διαγραφή». Στη συνέχεια, η εφαρμογή ζητάει επιβεβαίωση από το διαχειριστή, ώστε να διαγράψει τη σχετική εγγραφή. Σε περίπτωση που ο πιλότος συσχετίζεται με άλλες εγγραφές, όπως για παράδειγμα με μια μεταφορά, η εφαρμογή εμφανίζει σχετικό μήνυμα και δεν προχωράει στη διαγραφή του.



Εικόνα 465. Επιβεβαίωση διαγραφής πιλότου




Εικόνα 466. Συσχέτιση πιλότου με άλλες εγγραφές

Ακύρωση αλλαγών στα στοιχεία του πιλότου

Ο διαχειριστής μπορεί να ακυρώσει τυχόν αλλαγές στα στοιχεία του πιλότου πατώντας το κουμπί «Άκυρο». Στην περίπτωση αυτή, η εφαρμογή επαναφορτώνει τη σελίδα του συγκεκριμένου πιλότου.

Καταχώρηση αεροσυνοδού

Στη σελίδα αυτή, ο διαχειριστής μπορεί να καταχωρήσει τα στοιχεία ενός νέου αεροσυνοδού. Κατά την αποθήκευση, η εφαρμογή ελέγχει αν ο χρήστης έχει συμπληρώσει όλα τα απαραίτητα πεδία. Σε περίπτωση που δεν τα έχει συμπληρώσει όλα, εμφανίζει σχετικό μήνυμα. Επίσης, ελέγχει αν ο αριθμός ταυτότητας είναι ήδη καταχωρημένος και αντιστοιχεί σε άλλο υπάλληλο. Αφού γίνει η αποθήκευση του αεροσυνοδού, η εφαρμογή ανοίγει την σελίδα του αεροσυνοδού σε κατάσταση προβολής (view mode). Σε περίπτωση όμως, που ο χρήστης πάτησε το κουμπί «Αποθήκευση & κλείσιμο», η εφαρμογή αποθηκεύει τον αεροσυνοδό και κλείνει την σελίδα. Και στις δύο περιπτώσεις, ενημερώνει ταυτόχρονα την σελίδα αναζήτησης των αεροσυνοδών.

 **RODOS FLIGHTS**
FLY EVERYWHERE
GRIGORIS GRIGORIADIS (GREGORY96)

Αποθήκευση
Αποθήκευση & Κλείσιμο

Αριθμός Ταυτότητας

Όνομα

Επώνυμο

Οδός

Πόλη

Τηλέφωνο

Ημερομηνία Γέννησης 

Χρόνια Υπηρεσίας

Αριθμός

Εικόνα 467. Δημιουργία νέου αεροσυνοδού

ΠΡΟΣΟΧΗ

- Συμπληρώστε τον αριθμό ταυτότητας.
- Συμπληρώστε το όνομά σας.
- Συμπληρώστε το επώνυμό σας.
- Συμπληρώστε το όνομα της πόλης.
- Συμπληρώστε τον αριθμό τηλεφώνου σας.
- Συμπληρώστε την ημερομηνία γέννησής σας.

OK

Εικόνα 468. Έλεγχος εγκυρότητας στοιχείων αεροσυνοδού

ΠΡΟΣΟΧΗ

- Ο αριθμός ταυτότητας υπάρχει ήδη.

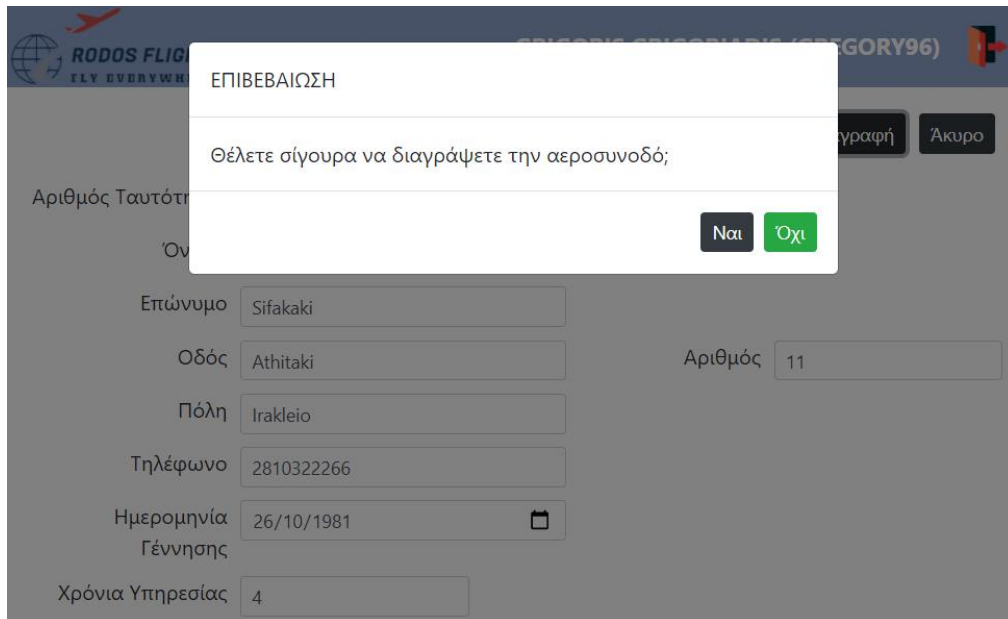
OK

Εικόνα 469. Έλεγχος ήδη καταχωρημένου αεροσυνοδού

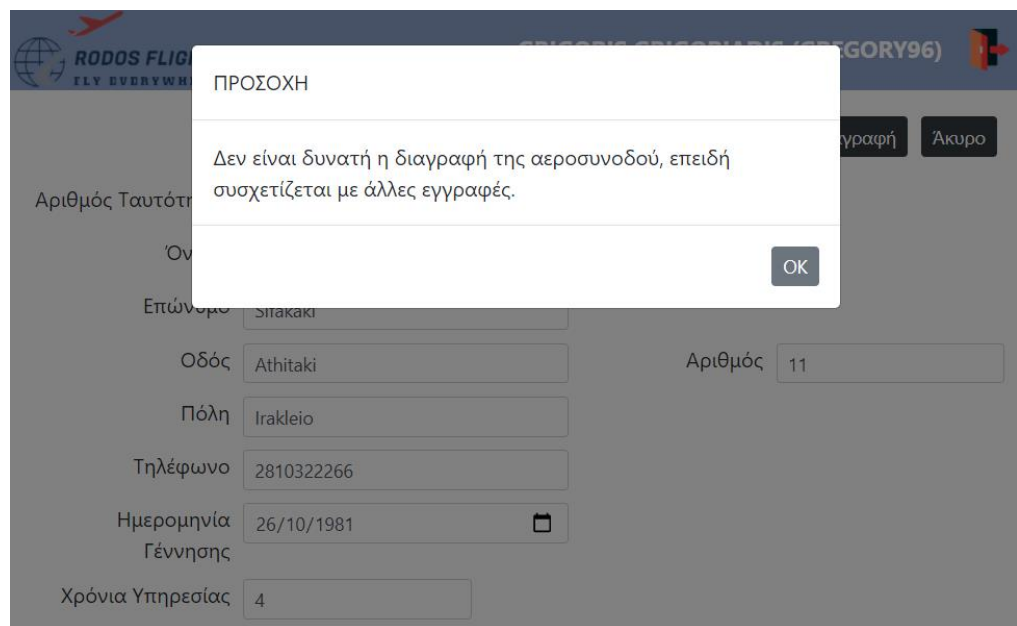
Διαγραφή αεροσυνοδού

Ο διαχειριστής μπορεί να διαγράψει έναν αεροσυνοδό, πατώντας το κουμπί «Διαγραφή». Στη συνέχεια, η εφαρμογή ζητάει επιβεβαίωση από το διαχειριστή, ώστε να διαγράψει τη σχετική εγγραφή. Σε περίπτωση που ο αεροσυνοδός συσχετίζεται με άλλες εγγραφές, όπως για

παράδειγμα με μια μεταφορά, η εφαρμογή εμφανίζει σχετικό μήνυμα και δεν προχωράει στη διαγραφή του.



Εικόνα 470. Επιβεβαίωση διαγραφής αεροσυνοδού



Εικόνα 471. Συσχέτιση αεροσυνοδού με άλλες εγγραφές

Ακύρωση αλλαγών στα στοιχεία του αεροσυνοδού

Ο διαχειριστής μπορεί να ακυρώσει τυχόν αλλαγές στα στοιχεία του αεροσυνοδού πατώντας το κουμπί «Άκυρο». Στην περίπτωση αυτή, η εφαρμογή επαναφορτώνει τη σελίδα του συγκεκριμένου αεροσυνοδού.

Καταχώρηση αεροπλάνου

Στη σελίδα αυτή, ο διαχειριστής μπορεί να καταχωρήσει τα στοιχεία ενός νέου αεροπλάνου. Κατά την αποθήκευση, η εφαρμογή ελέγχει αν ο χρήστης έχει συμπληρώσει όλα τα

απαραίτητα πεδία. Σε περίπτωση που δεν τα έχει συμπληρώσει όλα, εμφανίζει σχετικό μήνυμα. Επίσης, ελέγχει αν η ονομασία είναι ήδη καταχωρημένη. Αφού γίνει η αποθήκευση του αεροπλάνου, η εφαρμογή ανοίγει την σελίδα του αεροπλάνου σε κατάσταση προβολής (view mode). Σε περίπτωση όμως, που ο χρήστης πάτησε το κουμπί «Αποθήκευση & κλείσιμο», η εφαρμογή αποθηκεύει το αεροπλάνο και κλείνει την σελίδα. Και στις δύο περιπτώσεις, ενημερώνει ταυτόχρονα την σελίδα αναζήτησης των αεροπλάνων.



Όνομασία

Αριθμός διακεκριμένων θέσεων

Αριθμός οικονομικών θέσεων

Εικόνα 472. Δημιουργία νέου αεροπλάνου

ΠΡΟΣΟΧΗ

- Συμπληρώστε την ονομασία.
- Συμπληρώστε τον αριθμό διακεκριμένων θέσεων.
- Συμπληρώστε τον αριθμό οικονομικών θέσεων.

OK

Εικόνα 473. Έλεγχος εγκυρότητας στοιχείων αεροπλάνου

ΠΡΟΣΟΧΗ

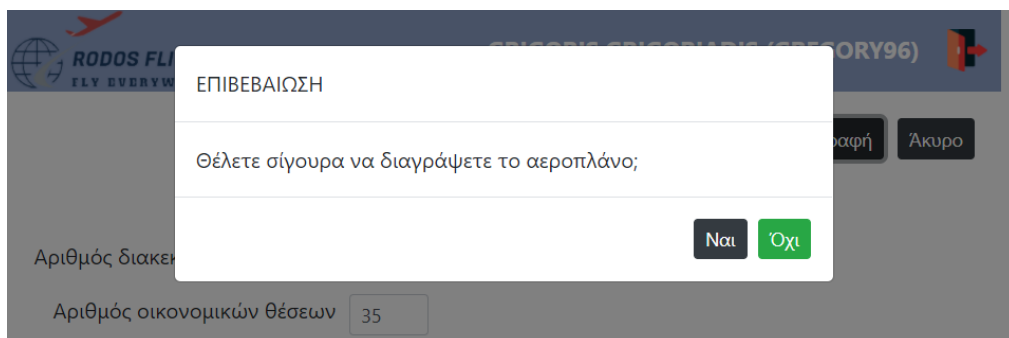
- Το αεροπλάνο υπάρχει ήδη.

OK

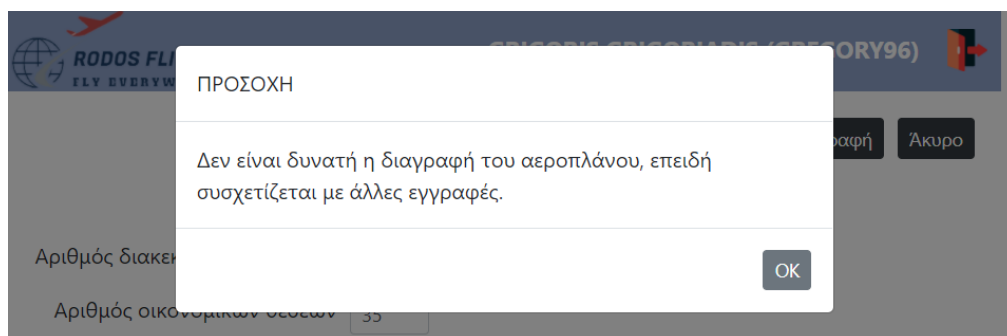
Εικόνα 474. Έλεγχος ήδη καταχωρημένου αεροπλάνου

Διαγραφή αεροπλάνου

Ο διαχειριστής μπορεί να διαγράψει ένα αεροπλάνο, πατώντας το κουμπί «Διαγραφή». Στη συνέχεια, η εφαρμογή ζητάει επιβεβαίωση από το διαχειριστή, ώστε να διαγράψει τη σχετική εγγραφή. Σε περίπτωση που το αεροπλάνο συσχετίζεται με άλλες εγγραφές, όπως για παράδειγμα με μια μεταφορά, η εφαρμογή εμφανίζει σχετικό μήνυμα και δεν προχωράει στη διαγραφή του.



Εικόνα 475. Επιβεβαίωση διαγραφής αεροπλάνου



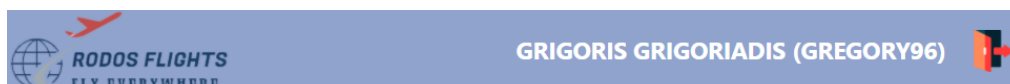
Εικόνα 476. Συσχέτιση αεροπλάνου με άλλες εγγραφές

Ακύρωση αλλαγών στα στοιχεία του αεροπλάνου

Ο διαχειριστής μπορεί να ακυρώσει τυχόν αλλαγές στα στοιχεία του αεροπλάνου πατώντας το κουμπί «Άκυρο». Στην περίπτωση αυτή, η εφαρμογή επαναφορτώνει τη σελίδα του συγκεκριμένου αεροπλάνου.

Καταχώρηση ταξιδιωτικού πράκτορα

Στη σελίδα αυτή, ο διαχειριστής μπορεί να καταχωρήσει τα στοιχεία ενός νέου ταξιδιωτικού πράκτορα. Κατά την αποθήκευση, η εφαρμογή ελέγχει αν ο χρήστης έχει συμπληρώσει όλα τα απαραίτητα πεδία. Σε περίπτωση που δεν τα έχει συμπληρώσει όλα, εμφανίζει σχετικό μήνυμα. Επίσης, ελέγχει αν ο αριθμός ταυτότητας και ο κωδικός είναι ήδη καταχωρημένα. Αφού γίνει η αποθήκευση του ταξιδιωτικού πράκτορα, η εφαρμογή ανοίγει την σελίδα του ταξιδιωτικού πράκτορα σε κατάσταση προβολής (view mode). Σε περίπτωση όμως, που ο χρήστης πάτησε το κουμπί «Αποθήκευση & κλείσιμο», η εφαρμογή αποθηκεύει τον ταξιδιωτικό πράκτορα και κλείνει την σελίδα. Και στις δύο περιπτώσεις, ενημερώνει ταυτόχρονα την σελίδα αναζήτησης των ταξιδιωτικών πρακτόρων.



Αποθήκευση

Αποθήκευση & Κλείσιμο

Αριθμός Ταυτότητας

Όνομα

Επώνυμο

Οδός Αριθμός

Πόλη

Κωδικός

Εικόνα 477. Δημιουργία νέου ταξιδιωτικού πράκτορα

ΠΡΟΣΟΧΗ

- Συμπληρώστε τον αριθμό ταυτότητας.
- Συμπληρώστε το όνομά σας.
- Συμπληρώστε το επώνυμό σας.
- Συμπληρώστε το όνομα της πόλης.
- Συμπληρώστε τον κωδικό σας.

OK

Εικόνα 478. Έλεγχος εγκυρότητας στοιχείων ταξιδιωτικού πράκτορα

ΠΡΟΣΟΧΗ

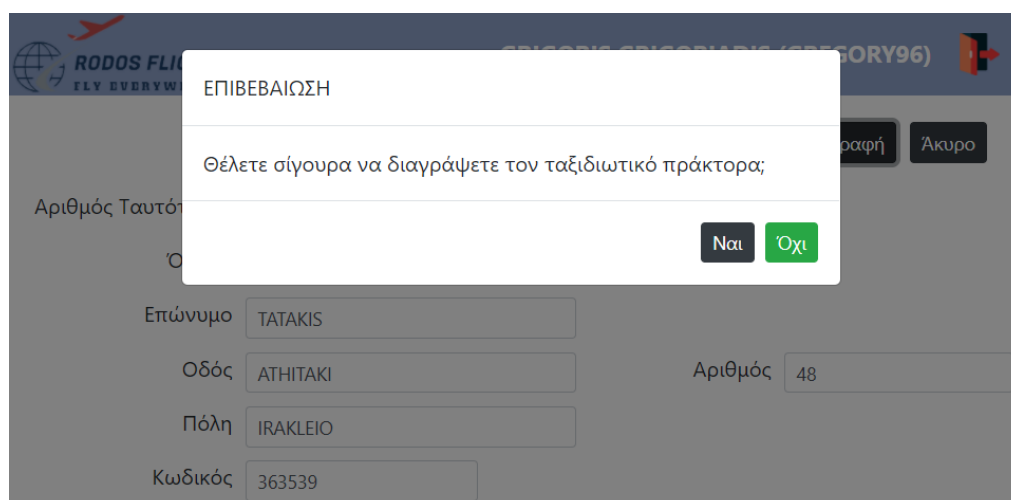
- Ο αριθμός ταυτότητας υπάρχει ήδη.
- Ο κωδικός υπάρχει ήδη.

OK

Εικόνα 479. Έλεγχος ήδη καταχωρημένου ταξιδιωτικού πράκτορα

Διαγραφή ταξιδιωτικού πράκτορα

Ο διαχειριστής μπορεί να διαγράψει ένα ταξιδιωτικό πράκτορα, πατώντας το κουμπί «Διαγραφή». Στη συνέχεια, η εφαρμογή ζητάει επιβεβαίωση από το διαχειριστή, ώστε να διαγράψει τη σχετική εγγραφή.



Εικόνα 480. Επιβεβαίωση διαγραφής ταξιδιωτικού πράκτορα

Ακύρωση αλλαγών στα στοιχεία του ταξιδιωτικού πράκτορα

Ο διαχειριστής μπορεί να ακυρώσει τυχόν αλλαγές στα στοιχεία του ταξιδιωτικού πράκτορα πατώντας το κουμπί «Άκυρο». Στην περίπτωση αυτή, η εφαρμογή επαναφορτώνει τη σελίδα του συγκεκριμένου ταξιδιωτικού πράκτορα.

Καταχώρηση διοικητικού υπαλλήλου

Στη σελίδα αυτή, ο διαχειριστής μπορεί να καταχωρήσει τα στοιχεία ενός νέου διοικητικού υπαλλήλου. Κατά την αποθήκευση, η εφαρμογή ελέγχει αν ο χρήστης έχει συμπληρώσει όλα τα απαραίτητα πεδία. Σε περίπτωση που δεν τα έχει συμπληρώσει όλα, εμφανίζει σχετικό μήνυμα. Επίσης, ελέγχει αν ο αριθμός ταυτότητας είναι ήδη καταχωρημένος. Αφού γίνει η αποθήκευση του διοικητικού υπαλλήλου, η εφαρμογή ανοίγει την σελίδα του διοικητικού υπαλλήλου σε κατάσταση προβολής (view mode). Σε περίπτωση όμως, που ο χρήστης πάτησε το κουμπί «Αποθήκευση & κλείσιμο», η εφαρμογή αποθηκεύει τον διοικητικό υπάλληλο και κλείνει την σελίδα. Και στις δύο περιπτώσεις, ενημερώνει ταυτόχρονα την σελίδα αναζήτησης των διοικητικών υπαλλήλων.



Αποθήκευση Αποθήκευση & Κλείσιμο

Αριθμός Ταυτότητας

Όνομα

Επώνυμο

Οδός

Πόλη

Τηλέφωνο

Αριθμός

Ημερομηνία Γέννησης

Εκπαίδευση

Χρόνια Υπηρεσίας

Εικόνα 481. Δημιουργία νέου διοικητικού υπαλλήλου

ΠΡΟΣΟΧΗ

- Συμπληρώστε τον αριθμό ταυτότητας.
- Συμπληρώστε το όνομά σας.
- Συμπληρώστε το επώνυμό σας.
- Συμπληρώστε το όνομα της πόλης.
- Συμπληρώστε τον αριθμό τηλεφώνου σας.
- Συμπληρώστε την ημερομηνία γέννησής σας.
- Συμπληρώστε τον τύπο εκπαίδευσής σας.

OK

Εικόνα 482. Έλεγχος εγκυρότητας στοιχείων διοικητικού υπαλλήλου

ΠΡΟΣΟΧΗ

- Ο αριθμός ταυτότητας υπάρχει ήδη.

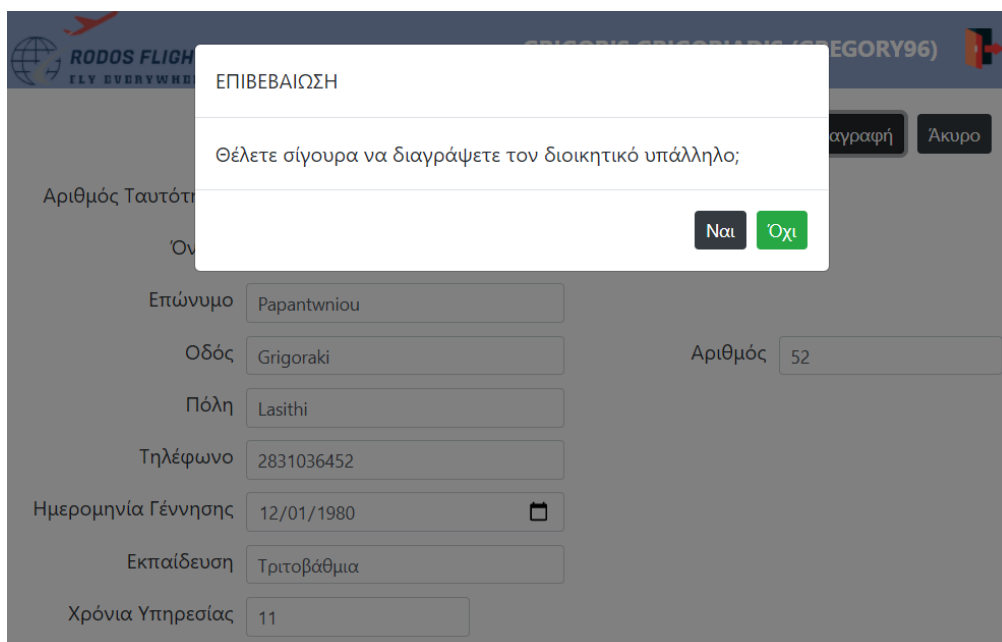
OK

Εικόνα 483. Έλεγχος ήδη καταχωρημένου διοικητικού υπαλλήλου

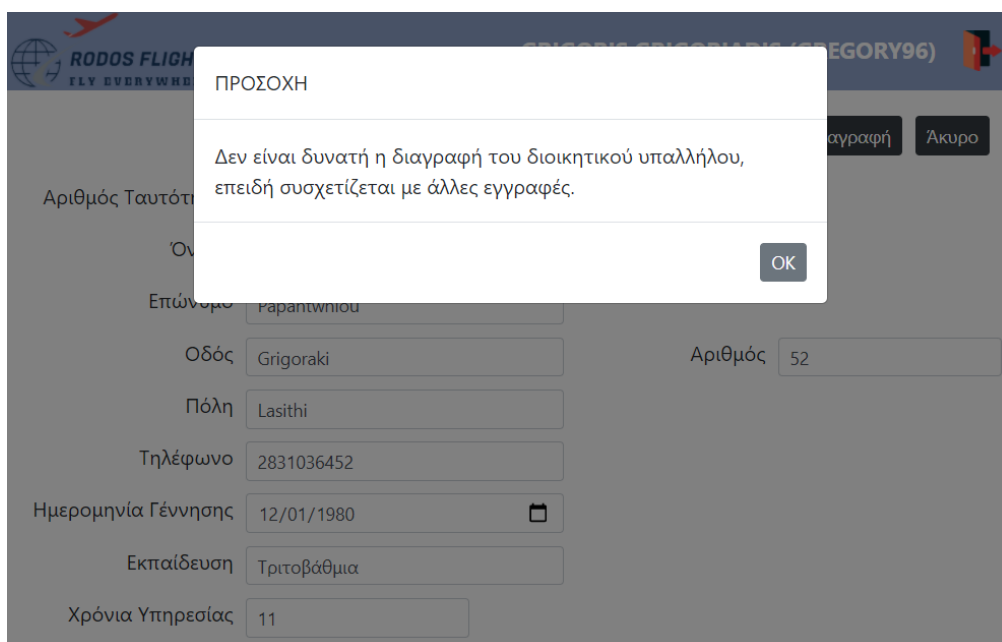
Διαγραφή διοικητικού υπαλλήλου

Ο διαχειριστής μπορεί να διαγράψει ένα διοικητικό υπάλληλο πατώντας το κουμπί «Διαγραφή». Στη συνέχεια, η εφαρμογή ζητάει επιβεβαίωση από το διαχειριστή, ώστε να διαγράψει τη σχετική εγγραφή. Σε περίπτωση που ο διοικητικός υπάλληλος συσχετίζεται με

άλλες εγγραφές, όπως για παράδειγμα με μια μεταφορά, η εφαρμογή εμφανίζει σχετικό μήνυμα και δεν προχωράει στη διαγραφή του.



Εικόνα 484. Επιβεβαίωση διαγραφής διοικητικού υπαλλήλου



Εικόνα 485. Συσχέτιση διοικητικού υπαλλήλου με άλλες εγγραφές

Ακύρωση αλλαγών στα στοιχεία του διοικητικού υπαλλήλου

Ο διαχειριστής μπορεί να ακυρώσει τυχόν αλλαγές στα στοιχεία του διοικητικού υπαλλήλου πατώντας το κουμπί «Άκυρο». Στην περίπτωση αυτή, η εφαρμογή επαναφορτώνει τη σελίδα του συγκεκριμένου διοικητικού υπαλλήλου.

Καταχώρηση χρήστη-πελάτη

Στη σελίδα αυτή, ο διαχειριστής μπορεί να καταχωρήσει τα στοιχεία ενός νέου χρήστη-

πελάτη. Κατά την αποθήκευση, η εφαρμογή ελέγχει αν ο χρήστης έχει συμπληρώσει όλα τα απαραίτητα πεδία. Σε περίπτωση που δεν τα έχει συμπληρώσει όλα, εμφανίζει σχετικό μήνυμα. Έπειτα, ελέγχει αν το όνομα χρήστη (username) και ο αριθμός τηλεφώνου είναι ήδη καταχωρημένα. Καθώς επίσης ελέγχει αν ο κωδικός πρόσβασης είναι ίδιος με την επιβεβαίωση του κωδικού. Αφού γίνει η αποθήκευση του χρήστη-πελάτη, η εφαρμογή ανοίγει την σελίδα του χρήστη-πελάτη σε κατάσταση προβολής (view mode). Σε περίπτωση όμως, που ο χρήστης πάτησε το κουμπί «Αποθήκευση & κλείσιμο», η εφαρμογή αποθηκεύει τον χρήστη-πελάτη και κλείνει την σελίδα. Και στις δύο περιπτώσεις, ενημερώνει ταυτόχρονα την σελίδα αναζήτησης των χρηστών-πελατών.

RODOS FLIGHTS
FLY EVERYWHERE

GRIGORIS GRIGORIADIS (GREGORY96)

Αποθήκευση Αποθήκευση & Κλείσιμο

Τηλέφωνο

Όνομα

Επώνυμο

Οδός

Πόλη

Όνομα χρήστη

Κωδικός πρόσβασης

Επιβεβαίωση

E-mail

Αριθμός

Εικόνα 486. Δημιουργία νέου χρήστη-πελάτη

ΠΡΟΣΟΧΗ

- Συμπληρώστε τον αριθμό τηλεφώνου.
- Συμπληρώστε το όνομά σας.
- Συμπληρώστε το επώνυμό σας.
- Συμπληρώστε την οδό.
- Συμπληρώστε τον αριθμό της οδού.
- Συμπληρώστε το όνομα της πόλης.
- Συμπληρώστε το όνομα χρήστη.
- Συμπληρώστε τον κωδικό.
- Επιβεβαιώστε τον κωδικό πρόσβασης.
- Συμπληρώστε την ηλεκτρονική σας διεύθυνση.

OK

Εικόνα 487. Έλεγχος εγκυρότητας στοιχείων χρήστη-πελάτη

ΠΡΟΣΟΧΗ

- Ο αριθμός τηλεφώνου υπάρχει ήδη.
- Το όνομα χρήστη υπάρχει ήδη.

OK

Εικόνα 488. Έλεγχος ήδη καταχωρημένου χρήστη-πελάτη

ΠΡΟΣΟΧΗ

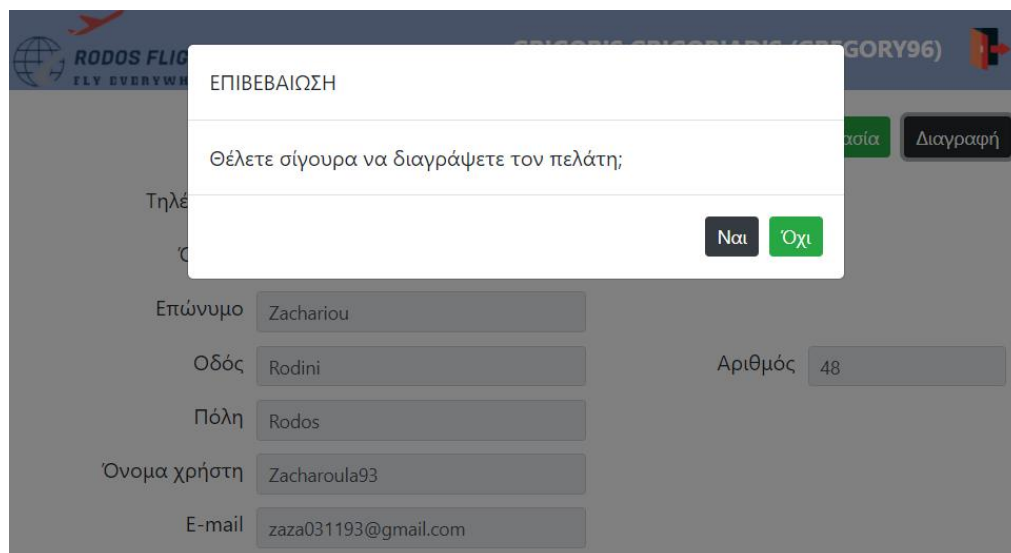
- Οι κωδικοί πρόσβασης δεν συμφωνούν. Προσπαθήστε ξανά.

OK

Εικόνα 489. Έλεγχος εγκυρότητας κωδικού πρόσβασης

Διαγραφή χρήστη-πελάτη

Ο διαχειριστής μπορεί να διαγράψει ένα χρήστη-πελάτη πατώντας το κουμπί «Διαγραφή». Στη συνέχεια, η εφαρμογή ζητάει επιβεβαίωση από το διαχειριστή, ώστε να διαγράψει τη σχετική εγγραφή.



Εικόνα 490. Επιβεβαίωση διαγραφής χρήστη-πελάτη

Ακύρωση αλλαγών στα στοιχεία του χρήστη-πελάτη

Ο διαχειριστής μπορεί να ακυρώσει τυχόν αλλαγές στα στοιχεία του χρήστη-πελάτη πατώντας το κουμπί «Άκυρο». Στην περίπτωση αυτή, η εφαρμογή επαναφορτώνει τη σελίδα του συγκεκριμένου χρήστη-πελάτη.

Καταχώρηση χρήστη-ταξιδιωτικού πράκτορα

Στη σελίδα αυτή, ο διαχειριστής μπορεί να καταχωρήσει τα στοιχεία ενός νέου χρήστη-ταξιδιωτικού πράκτορα. Κατά την αποθήκευση, η εφαρμογή ελέγχει αν ο χρήστης έχει συμπληρώσει όλα τα απαραίτητα πεδία. Σε περίπτωση που δεν τα έχει συμπληρώσει όλα, εμφανίζει σχετικό μήνυμα. Έπειτα, ελέγχει αν ο αριθμός ταυτότητας, το όνομα χρήστη (username) και ο κωδικός είναι ήδη καταχωρημένα. Καθώς επίσης ελέγχει αν ο κωδικός πρόσβασης είναι ίδιος με την επιβεβαίωση του κωδικού. Αφού γίνει η αποθήκευση του χρήστη-ταξιδιωτικού πράκτορα, η εφαρμογή ανοίγει την σελίδα του χρήστη-ταξιδιωτικού πράκτορα σε κατάσταση προβολής (view mode). Σε περίπτωση όμως, που ο χρήστης πάτησε το κουμπί «Αποθήκευση & κλείσιμο», η εφαρμογή αποθηκεύει τον χρήστη-ταξιδιωτικό πράκτορα και κλείνει την σελίδα. Και στις δύο περιπτώσεις, ενημερώνει ταυτόχρονα την σελίδα αναζήτησης των χρηστών-ταξιδιωτικών πρακτόρων.



RODOS FLIGHTS
FLY EVERYWHERE

GRIGORIS GRIGORIADIS (GREGORY96)

Αποθήκευση Αποθήκευση & Κλείσιμο

Αριθμός Ταυτότητας

Όνομα

Επώνυμο

Οδός

Πόλη

Κωδικός

Αριθμός

Όνομα χρήστη

Κωδικός πρόσβασης

Επιβεβαίωση

Εικόνα 491. Δημιουργία νέου χρήστη-ταξιδιωτικού πράκτορα

ΠΡΟΣΟΧΗ

- Συμπληρώστε τον αριθμό ταυτότητας.
- Συμπληρώστε το όνομά σας.
- Συμπληρώστε το επώνυμό σας.
- Συμπληρώστε την οδό.
- Συμπληρώστε τον αριθμό της οδού.
- Συμπληρώστε το όνομα της πόλης.
- Συμπληρώστε τον κωδικό σας.
- Συμπληρώστε το όνομα χρήστη.
- Συμπληρώστε τον κωδικό.
- Επιβεβαιώστε τον κωδικό πρόσβασης.

OK

Εικόνα 492. Έλεγχος εγκυρότητας στοιχείων χρήστη-ταξιδιωτικού πράκτορα

ΠΡΟΣΟΧΗ

- Ο αριθμός ταυτότητας υπάρχει ήδη.
- Το όνομα χρήστη υπάρχει ήδη.
- Ο κωδικός υπάρχει ήδη

OK

Εικόνα 493. Έλεγχος ήδη καταχωρημένου χρήστη-ταξιδιωτικού πράκτορα

ΠΡΟΣΟΧΗ

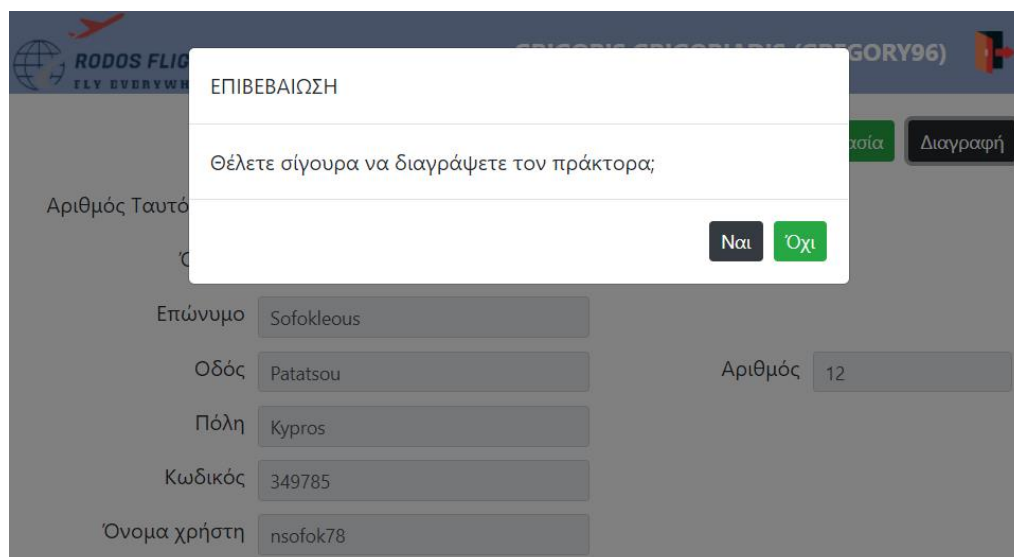
- Οι κωδικοί πρόσβασης δεν συμφωνούν. Προσπαθήστε ξανά.

OK

Εικόνα 494. Έλεγχος εγκυρότητας κωδικού πρόσβασης

Διαγραφή χρήστη-ταξιδιωτικού πράκτορα

Ο διαχειριστής μπορεί να διαγράψει ένα χρήστη-ταξιδιωτικό πράκτορα πατώντας το κουμπί «Διαγραφή». Στη συνέχεια, η εφαρμογή ζητάει επιβεβαίωση από το διαχειριστή, ώστε να διαγράψει τη σχετική εγγραφή.



Εικόνα 495. Επιβεβαίωση διαγραφής χρήστη-ταξιδιωτικού πράκτορα

Ακύρωση αλλαγών στα στοιχεία του χρήστη-ταξιδιωτικού πράκτορα

Ο διαχειριστής μπορεί να ακυρώσει τυχόν αλλαγές στα στοιχεία του χρήστη-ταξιδιωτικού πράκτορα πατώντας το κουμπί «Άκυρο». Στην περίπτωση αυτή, η εφαρμογή επαναφορτώνει τη σελίδα του συγκεκριμένου χρήστη-ταξιδιωτικού πράκτορα.

Καταχώρηση χρήστη-διαχειριστή

Στη σελίδα αυτή, ο διαχειριστής μπορεί να καταχωρήσει τα στοιχεία ενός νέου χρήστη-διαχειριστή. Κατά την αποθήκευση, η εφαρμογή ελέγχει αν ο χρήστης έχει συμπληρώσει όλα τα απαραίτητα πεδία. Σε περίπτωση που δεν τα έχει συμπληρώσει όλα, εμφανίζει σχετικό μήνυμα. Έπειτα, ελέγχει αν ο αριθμός ταυτότητας και το όνομα χρήστη (username) είναι ήδη καταχωρημένα. Καθώς επίσης ελέγχει αν ο κωδικός πρόσβασης είναι ίδιος με την επιβεβαίωση του κωδικού. Αφού γίνει η αποθήκευση του χρήστη-διαχειριστή, η εφαρμογή ανοίγει την σελίδα του χρήστη-διαχειριστή σε κατάσταση προβολής (view mode). Σε περίπτωση όμως, που ο χρήστης πάτησε το κουμπί «Αποθήκευση & κλείσιμο», η εφαρμογή αποθηκεύει τον χρήστη-ταξιδιωτικό πράκτορα και κλείνει την σελίδα. Και στις δύο περιπτώσεις, ενημερώνει ταυτόχρονα την σελίδα αναζήτησης των χρηστών-διαχειριστών.

RODOS FLIGHTS
FLY EVERYWHERE

GRIGORIS GRIGORIADIS (GREGORY96)

Αποθήκευση Αποθήκευση & Κλείσιμο

Αριθμός Ταυτότητας

Όνομα

Επώνυμο

Οδός

Πόλη

Τηλέφωνο

Ημερομηνία Γέννησης

Όνομα χρήστη

Κωδικός πρόσβασης

Επιβεβαίωση

Αριθμός

Εικόνα 496. Δημιουργία νέου χρήστη-διαχειριστή

ΠΡΟΣΟΧΗ

- Συμπληρώστε τον αριθμό ταυτότητας.
- Συμπληρώστε το όνομά σας.
- Συμπληρώστε το επώνυμό σας.
- Συμπληρώστε το όνομα της πόλης.
- Συμπληρώστε τον αριθμό τηλεφώνου σας.
- Συμπληρώστε την ημερομηνία γέννησής σας.
- Συμπληρώστε το όνομα χρήστη.
- Συμπληρώστε τον κωδικό.
- Επιβεβαιώστε τον κωδικό πρόσβασης.

OK

Εικόνα 497. Έλεγχος εγκυρότητας στοιχείων χρήστη-διαχειριστή

ΠΡΟΣΟΧΗ

- Ο αριθμός ταυτότητας υπάρχει ήδη.
- Το όνομα χρήστη υπάρχει ήδη.

OK

Εικόνα 498. Έλεγχος ήδη καταχωρημένου χρήστη-διαχειριστή

ΠΡΟΣΟΧΗ

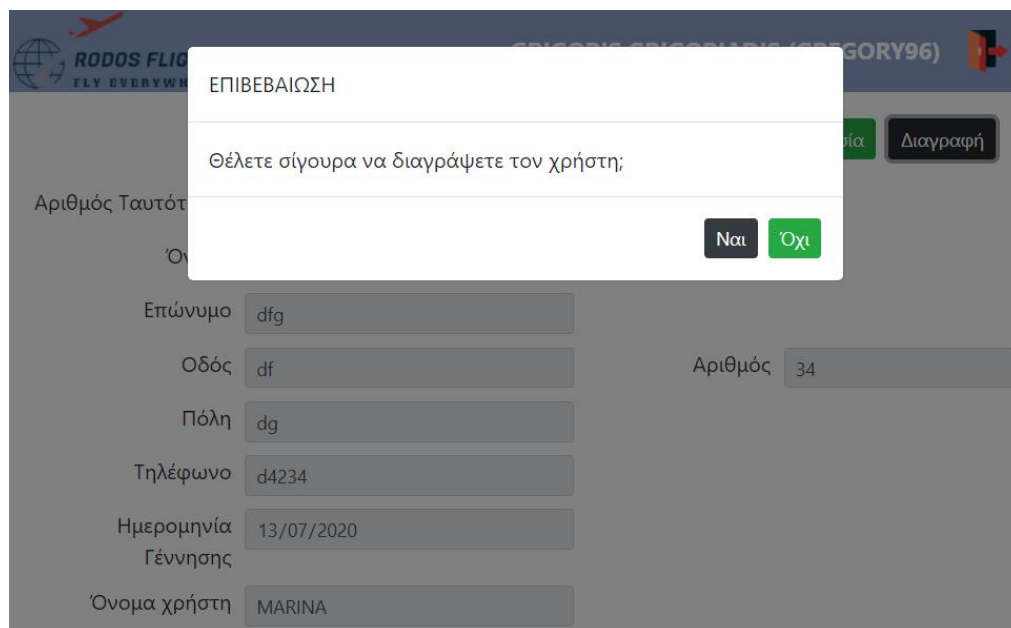
- Οι κωδικοί πρόσβασης δεν συμφωνούν. Προσπαθήστε ξανά.

OK

Εικόνα 499. Έλεγχος εγκυρότητας κωδικού πρόσβασης

Διαγραφή χρήστη-διαχειριστή

Ο διαχειριστής μπορεί να διαγράψει ένα χρήστη-διαχειριστή πατώντας το κουμπί «Διαγραφή». Στη συνέχεια, η εφαρμογή ζητάει επιβεβαίωση από το διαχειριστή, ώστε να διαγράψει τη σχετική εγγραφή.




Εικόνα 500. Επιβεβαίωση διαγραφής χρήστη-διαχειριστή


Ακύρωση αλλαγών στα στοιχεία του χρήστη-διαχειριστή

Ο διαχειριστής μπορεί να ακυρώσει τυχόν αλλαγές στα στοιχεία του χρήστη-διαχειριστή πατώντας το κουμπί «Άκυρο». Στην περίπτωση αυτή, η εφαρμογή επαναφορτώνει τη σελίδα του συγκεκριμένου χρήστη-διαχειριστή.

Καταχώρηση δρομολογίου

Στη σελίδα αυτή, ο διαχειριστής μπορεί να καταχωρήσει τα στοιχεία ενός νέου δρομολογίου. Κατά την αποθήκευση, η εφαρμογή ελέγχει αν ο χρήστης έχει συμπληρώσει όλα τα απαραίτητα πεδία. Σε περίπτωση που δεν τα έχει συμπληρώσει όλα, εμφανίζει σχετικό μήνυμα. Εξίσου, ελέγχει αν η απόσταση και οι τιμές των θέσεων είναι θετικοί αριθμοί. Επίσης, οι πόλεις αναχώρησης και άφιξης να είναι διαφορετικές. Έπειτα, ελέγχει αν ο κωδικός είναι ήδη καταχωρημένος. Επιπλέον, ελέγχει αν οι ώρες αναχώρησης και άφιξης να διαφέρουν τουλάχιστον μισή ώρα, καθώς και η ώρα άφιξης να είναι μεταγενέστερη της ώρας αναχώρησης. Αφού γίνει η αποθήκευση του δρομολογίου, η εφαρμογή ανοίγει την σελίδα του δρομολογίου σε κατάσταση προβολής (view mode). Σε περίπτωση όμως, που ο χρήστης πάτησε το κουμπί «Αποθήκευση & κλείσιμο», η εφαρμογή αποθηκεύει το δρομολόγιο και κλείνει την σελίδα. Και στις δύο περιπτώσεις, ενημερώνει ταυτόχρονα την σελίδα αναζήτησης των δρομολογίων.



GRIGORIS GRIGORIADIS (GREGORY96)


Αποθήκευση
Αποθήκευση & Κλείσιμο

Κωδικός <input style="width: 100%;" type="text"/>	Προορισμός <input style="width: 100%;" type="text"/>
Αφετηρία <input style="width: 100%;" type="text"/>	
Ημέρα πραγματοποίησης <input style="width: 100%;" type="text"/>	
Απόσταση (km) <input style="width: 100%;" type="text" value="0"/>	
Ώρα αναχώρησης <input style="width: 100%;" type="text" value="--:-- --"/>	Ώρα άφιξης <input style="width: 100%;" type="text" value="--:-- --"/>
Τιμή διακεκριμένης θέσης <input style="width: 100%;" type="text" value="0"/>	
Τιμή οικονομικής θέσης <input style="width: 100%;" type="text" value="0"/>	

Εικόνα 501. Δημιουργία νέου δρομολογίου

ΠΡΟΣΟΧΗ

- Συμπληρώστε τον κωδικό.
- Συμπληρώστε την αφετηρία.
- Συμπληρώστε τον προορισμό.
- Συμπληρώστε την ημέρα πραγματοποίησης του ταξιδιού.
- Η απόσταση πρέπει να είναι θετικός αριθμός.
- Συμπληρώστε την ώρα αναχώρησης.
- Συμπληρώστε την ώρα άφιξης.
- Η τιμή διακεκριμένης θέσης πρέπει να είναι θετικός αριθμός.
- Η τιμή οικονομικής θέσης πρέπει να είναι θετικός αριθμός.
- Επιλέξτε διαφορετικό προορισμό.

OK

Εικόνα 502. Έλεγχος εγκυρότητας στοιχείων δρομολογίου

ΠΡΟΣΟΧΗ

- Το δρομολόγιο υπάρχει ήδη.

OK

Εικόνα 503. Έλεγχος ήδη καταχωρημένου δρομολογίου

ΠΡΟΣΟΧΗ

- Η ώρα αναχώρησης και η ώρα άφιξης πρέπει να διαφέρουν τουλάχιστον μισή ώρα.

OK

Εικόνα 504. Έλεγχος ελάχιστης διαφοράς χρόνου πραγματοποίησης δρομολογίου

ΠΡΟΣΟΧΗ

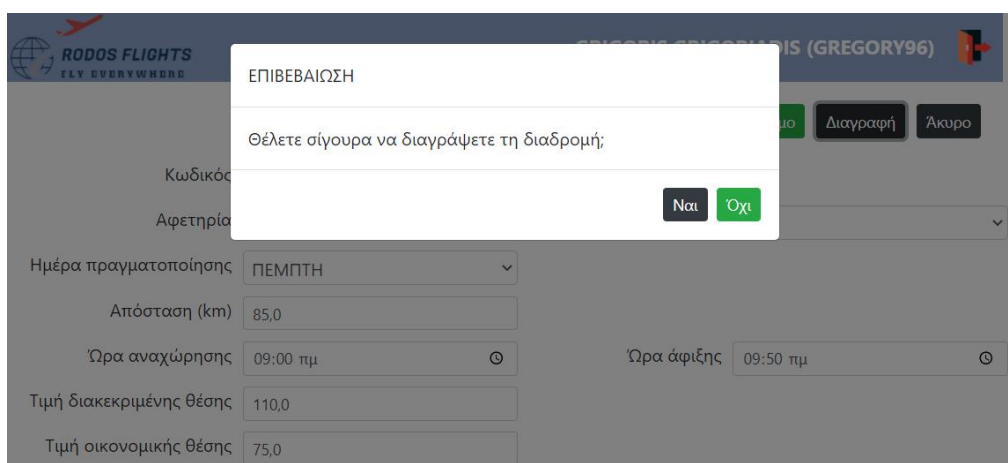
- Η ώρα αναχώρησης δεν μπορεί να είναι μεταγενέστερη της ώρας άφιξης.

OK

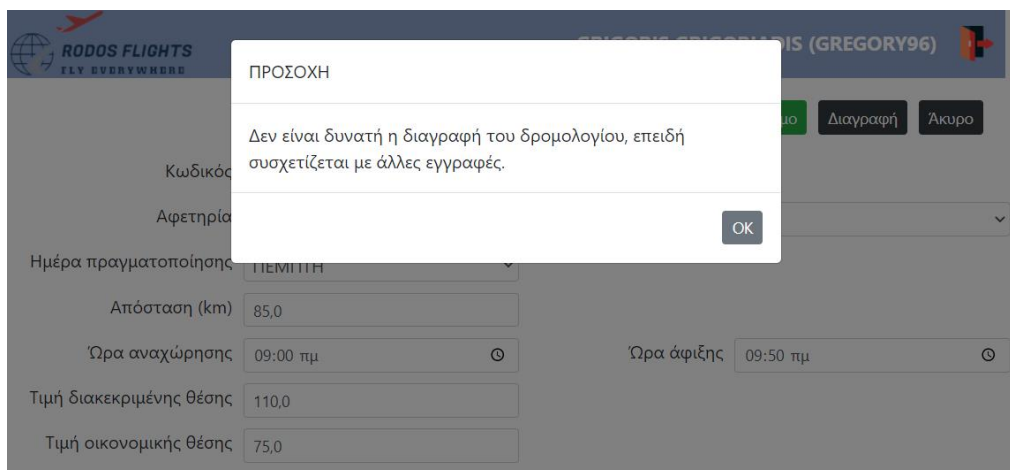
Εικόνα 505. Έλεγχος μεταγενέστερης ώρας άφιξης δρομολογίου

Διαγραφή δρομολογίου

Ο διαχειριστής μπορεί να διαγράψει ένα δρομολόγιο πατώντας το κουμπί «Διαγραφή». Στη συνέχεια, η εφαρμογή ζητάει επιβεβαίωση από το διαχειριστή, ώστε να διαγράψει τη σχετική εγγραφή. Σε περίπτωση που το δρομολόγιο συσχετίζεται με άλλες εγγραφές, όπως για παράδειγμα με μια μεταφορά, η εφαρμογή εμφανίζει σχετικό μήνυμα και δεν προχωράει στη διαγραφή του.



Εικόνα 506. Επιβεβαίωση διαγραφής δρομολογίου



Εικόνα 507. Συσχέτιση δρομολογίου με άλλες εγγραφές

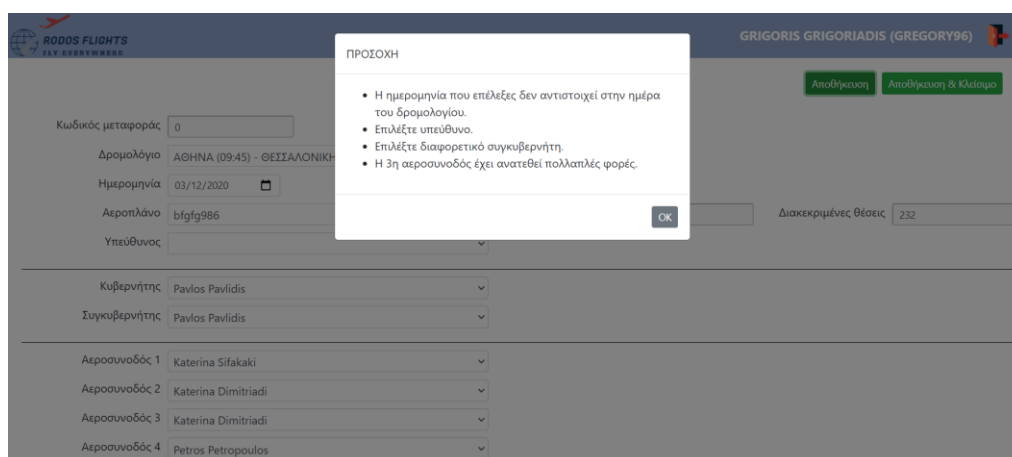
Ακύρωση αλλαγών στα στοιχεία του δρομολογίου

Ο διαχειριστής μπορεί να ακυρώσει τυχόν αλλαγές στα στοιχεία του δρομολογίου πατώντας το κουμπί «Άκυρο». Στην περίπτωση αυτή, η εφαρμογή επαναφορτώνει τη σελίδα του συγκεκριμένου δρομολογίου.

Καταχώρηση μεταφοράς

Στη σελίδα αυτή, ο διαχειριστής μπορεί να καταχωρήσει τα στοιχεία μιας νέας μεταφοράς. Κατά την αποθήκευση, η εφαρμογή ελέγχει αν ο χρήστης έχει συμπληρώσει όλα τα απαραίτητα πεδία. Σε περίπτωση που δεν τα έχει συμπληρώσει όλα, εμφανίζει σχετικό μήνυμα. Επίσης, ελέγχει αν η ημερομηνία αντιστοιχεί στη μέρα του δρομολογίου. Καθώς επίσης, ελέγχει αν οι πιλότοι και οι αεροσυνοδοί δεν επιλέχθηκαν πάνω από μία φορά. Τέλος, ελέγχει αν υπάρχει χρονική επικάλυψη της συγκεκριμένης μεταφοράς με άλλες μεταφορές. Αφού γίνει η αποθήκευση της μεταφοράς, η εφαρμογή ανοίγει την σελίδα της μεταφοράς σε κατάσταση προβολής (view mode). Σε περίπτωση όμως, που ο χρήστης πάτησε το κουμπί «Αποθήκευση & κλείσιμο», η εφαρμογή αποθηκεύει τη μεταφορά και κλείνει την σελίδα. Και στις δύο περιπτώσεις, ενημερώνει ταυτόχρονα την σελίδα αναζήτησης των μεταφορών.

Εικόνα 508. Δημιουργία νέας μεταφοράς



Εικόνα 509. Έλεγχος εγκυρότητας στοιχείων μεταφοράς

ΠΡΟΣΟΧΗ

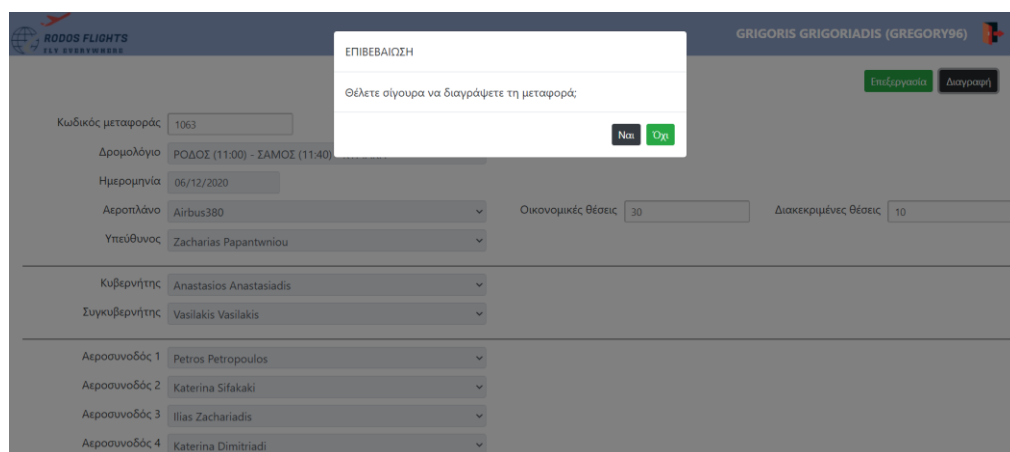
Δεν είναι δυνατή η αποθήκευση της μεταφοράς, διότι υπάρχει χρονική επικάλυψη με άλλες μεταφορές.

OK

Εικόνα 510. Έλεγχος χρονικής επικάλυψης μεταφοράς με άλλες

Διαγραφή μεταφοράς

Ο διαχειριστής μπορεί να διαγράψει μια μεταφορά πατώντας το κουμπί «Διαγραφή». Στη συνέχεια, η εφαρμογή ζητάει επιβεβαίωση από το διαχειριστή, ώστε να διαγράψει τη σχετική εγγραφή.



Εικόνα 511. Επιβεβαίωση διαγραφής μεταφοράς

Ακύρωση αλλαγών στα στοιχεία της μεταφοράς

Ο διαχειριστής μπορεί να ακυρώσει τυχόν αλλαγές στα στοιχεία της μεταφοράς πατώντας το κουμπί «Άκυρο». Στην περίπτωση αυτή, η εφαρμογή επαναφορτώνει τη σελίδα της συγκεκριμένης μεταφοράς.

Καταχώρηση κράτησης

Στη σελίδα αυτή, ο χρήστης-πελάτης και ο χρήστης-ταξιδιωτικός πράκτορας μπορούν να καταχωρήσουν τα στοιχεία μιας νέας κράτησης. Κατά την αποθήκευση, η εφαρμογή ελέγχει αν ο χρήστης έχει συμπληρώσει όλα τα απαραίτητα πεδία. Σε περίπτωση που δεν τα έχει συμπληρώσει όλα, εμφανίζει σχετικό μήνυμα. Στην περίπτωση που η κράτηση δημιουργείται από τον ταξιδιωτικό πράκτορα, η εφαρμογή ελέγχει αν η ημερομηνία της κράτησης είναι προγενέστερη αυτής της λήξης της, «παγώνοντας» τις προηγούμενες. Καθώς επίσης, γίνεται ο ίδιος έλεγχος και με την αντίθετη περίπτωση. Δηλαδή η ημερομηνία λήξης της κράτησης να είναι μεταγενέστερη αυτής της προηγούμενης. Τέλος, η εφαρμογή κάνει αναζήτηση στη βάση δεδομένων αν υπάρχουν διαθέσιμες μεταφορές για τη συγκεκριμένη ημερομηνία κράτησης, που ζητάει ο χρήστης. Εάν δεν βρεθούν διαθέσιμες μεταφορές, τότε εμφανίζεται σχετικό μήνυμα μέσα στο πλαίσιο δεξιά της σελίδας. Εάν βρεθούν διαθέσιμες μεταφορές, τότε εμφανίζονται μέσα στο πλαίσιο και ο χρήστης επιλέγει τη μεταφορά της αρεσκείας του, επιλέγοντας ταυτόχρονα και τον τύπο θέσης (οικονομική ή διακεκριμένη). Εάν η συγκεκριμένη θέση, που επιθυμεί να επιλέξει ο χρήστης δεν είναι διαθέσιμη λόγω πληρότητας, τότε η εφαρμογή «παγώνει» την επιλογή και δεν επιτρέπει στον χρήστη να την επιλέξει. Σε περίπτωση που δεν επιλέχθηκε η μεταφορά, εμφανίζεται σχετικό μήνυμα. Επίσης, όταν ο χρήστης πατήσει το κουμπί της αποθήκευσης, η εφαρμογή ελέγχει αν ο χρήστης είχε κάνει ξανά κράτηση για τη συγκεκριμένη ημερομηνία. Εάν είχε κάνει κι άλλη κράτηση, ανεξαρτήτου προορισμού, τότε εμφανίζεται σχετικό μήνυμα και δεν προχωράει στην αποθήκευση της κράτησης. Αφού γίνει η αποθήκευση της κράτησης, η εφαρμογή ανοίγει την σελίδα της κράτησης σε κατάσταση προβολής (view mode). Σε περίπτωση όμως, που ο χρήστης πάτησε το κουμπί «Αποθήκευση & κλείσιμο», η εφαρμογή αποθηκεύει τη κράτηση και κλείνει την σελίδα. Και στις δύο περιπτώσεις, ενημερώνει ταυτόχρονα την σελίδα αναζήτησης των κρατήσεων.

Εικόνα 512. Δημιουργία νέας κράτησης από τον ταξιδιωτικό πράκτορα

Εικόνα 513. Δημιουργία νέας κράτησης από τον πελάτη

ΠΡΟΣΟΧΗ

- Επιλέξτε τον πελάτη.
- Επιλέξτε ημερομηνία κράτησης.
- Επιλέξτε ημερομηνία λήξης της κράτησης.
- Επιλέξτε πόλη αναχώρησης.
- Επιλέξτε πόλη άφιξης.

OK

Εικόνα 514. Έλεγχος εγκυρότητας στοιχείων κράτησης από τον ταξιδιωτικό πράκτορα

ΠΡΟΣΟΧΗ

- Επιλέξτε ημερομηνία κράτησης.
- Επιλέξτε πόλη αναχώρησης.
- Επιλέξτε πόλη άφιξης.

OK

Εικόνα 515. Έλεγχος εγκυρότητας στοιχείων κράτησης από τον πελάτη

The screenshot shows the Rodos Flights website interface. At the top, there is a header with the Rodos Flights logo and the name 'Lampros Lamprakis (Lampros78)'. Below the header, there are two green buttons: 'Αποθήκευση' and 'Αποθήκευση & Κλείσιμο'. The main form contains several input fields: 'Κωδικός κράτησης' (0), 'Πελάτης' (dropdown), 'Ημερομηνία κράτησης' (10/12/2020), 'Ημερομηνία λήξης' (11/μμ/εεεε), 'Πόλη αναχώρησης' (Δεκέμβριος 2020), and 'Πόλη άφιξης'. A calendar dropdown is open for the departure date, showing the month of December 2020. The calendar grid shows days from 1 to 31, with the 11th highlighted in blue. To the right of the form, there is a box labeled 'ΔΙΑΘΕΣΙΜΕΣ ΜΕΤΑΦΟΡΕΣ' which is currently empty.

Εικόνα 516. Έλεγχος εγκυρότητας ημερομηνίας κράτησης

The screenshot shows the Rodos Flights search interface. The header includes the logo and the user name 'Zacharoula Zachariou (Zacharoula93)'. The search criteria are:

- Κωδικός κράτησης: 0
- Ημερομηνία κράτησης: 07/12/2020
- Πόλη αναχώρησης: ΘΕΣΣΑΛΟΝΙΚΗ
- Πόλη άφιξης: ΚΑΡΠΑΘΟΣ

 On the right, there are buttons for 'Αποθήκευση' and 'Αποθήκευση & Κλείσιμο'. A pop-up window titled 'ΔΙΑΘΕΣΙΜΕΣ ΜΕΤΑΦΟΡΕΣ' displays the message: 'Δεν βρέθηκαν μεταφορές.'

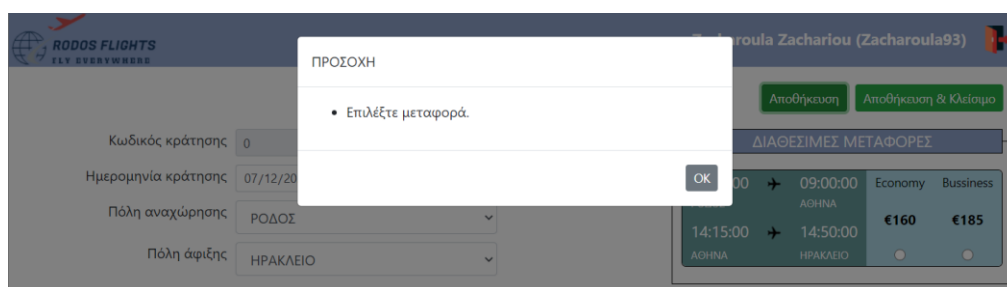
Εικόνα 517. Μήνυμα μη διαθέσιμων μεταφορών

The screenshot shows the Rodos Flights search interface with the same search criteria as Figure 517. The search results are displayed in a table under the heading 'ΔΙΑΘΕΣΙΜΕΣ ΜΕΤΑΦΟΡΕΣ':

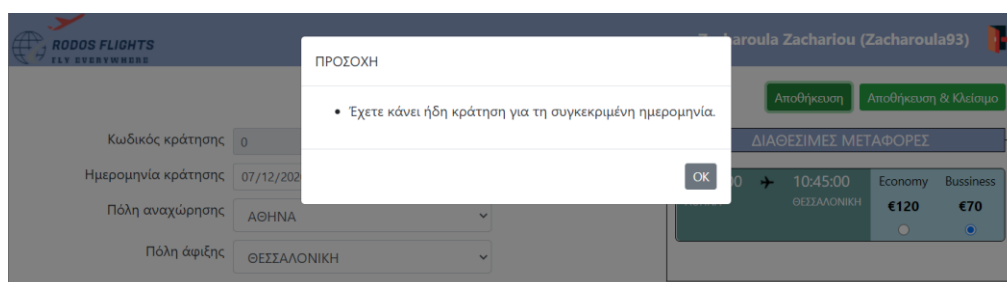
Time	From	To	Economy	Business
08:00:00	ΡΟΔΟΣ	ΑΘΗΝΑ		
14:15:00	ΑΘΗΝΑ	ΗΡΑΚΛΕΙΟ	€160	€185

 The 'Economy' and 'Business' columns have radio buttons below them, with 'Economy' selected.

Εικόνα 518. Μήνυμα διαθέσιμων μεταφορών



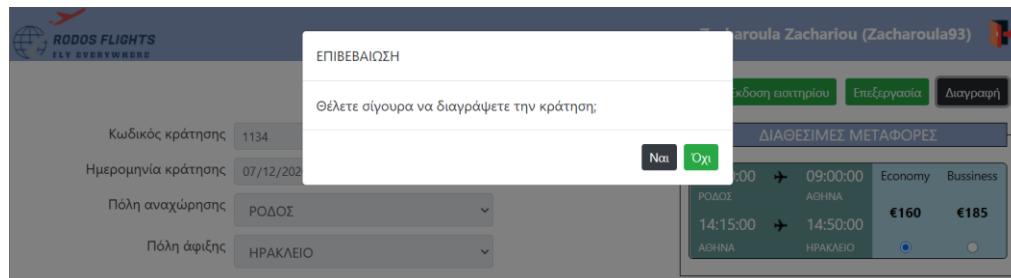
Εικόνα 519. Έλεγχος επιλογής μεταφοράς



Εικόνα 520. Έλεγχος ήδη καταχωρημένης κράτησης

Διαγραφή κράτησης

Ο χρήστης-πελάτης και ο χρήστης-ταξιδιωτικός πράκτορας μπορούν να διαγράψουν μια κράτηση πατώντας το κουμπί «Διαγραφή». Στη συνέχεια, η εφαρμογή ζητάει επιβεβαίωση από το χρήστη, ώστε να διαγράψει τη σχετική εγγραφή.



Εικόνα 521. Επιβεβαίωση διαγραφής κράτησης

Ακύρωση αλλαγών στα στοιχεία της κράτησης

Ο χρήστης-πελάτης και ο χρήστης-ταξιδιωτικός πράκτορας μπορούν να ακυρώσουν τυχόν αλλαγές στα στοιχεία της κράτησης πατώντας το κουμπί «Άκυρο». Στην περίπτωση αυτή, η εφαρμογή επαναφορτώνει τη σελίδα της συγκεκριμένης κράτησης.

Καταχώρηση πελάτη

Στη σελίδα αυτή, ο χρήστης-ταξιδιωτικός πράκτορας μπορεί να καταχωρήσει τα στοιχεία ενός νέου πελάτη. Κατά την αποθήκευση, η εφαρμογή ελέγχει αν ο χρήστης έχει συμπληρώσει όλα τα απαραίτητα πεδία. Σε περίπτωση που δεν τα έχει συμπληρώσει όλα, εμφανίζει σχετικό μήνυμα. Έπειτα, ελέγχει αν ο αριθμός τηλεφώνου είναι ήδη καταχωρημένος. Αφού γίνει η αποθήκευση του πελάτη, η εφαρμογή ανοίγει την σελίδα του πελάτη σε κατάσταση προβολής (view mode). Σε περίπτωση όμως, που ο χρήστης πάτησε το κουμπί «Αποθήκευση & κλείσιμο», η εφαρμογή αποθηκεύει τον πελάτη και κλείνει την σελίδα. Και στις δύο περιπτώσεις, ενημερώνει ταυτόχρονα την σελίδα αναζήτησης των

πελατών.

RODOS FLIGHTS
FLY EVERYWHERE

Lampros Lamprakis (Lampros78)

Αποθήκευση Αποθήκευση & Κλείσιμο

Τηλέφωνο

Όνομα

Επώνυμο

Οδός Αριθμός

Πόλη

E-mail

Εικόνα 522. Δημιουργία νέου πελάτη

ΠΡΟΣΟΧΗ

- Συμπληρώστε τον αριθμό τηλεφώνου.
- Συμπληρώστε το όνομα.
- Συμπληρώστε το επώνυμο.
- Συμπληρώστε την οδό.
- Συμπληρώστε τον αριθμό της οδού.
- Συμπληρώστε το όνομα της πόλης.
- Συμπληρώστε την ηλεκτρονική διεύθυνση.

OK

Εικόνα 523. Έλεγχος εγκυρότητας στοιχείων πελάτη

ΠΡΟΣΟΧΗ

- Ο αριθμός τηλεφώνου υπάρχει ήδη.

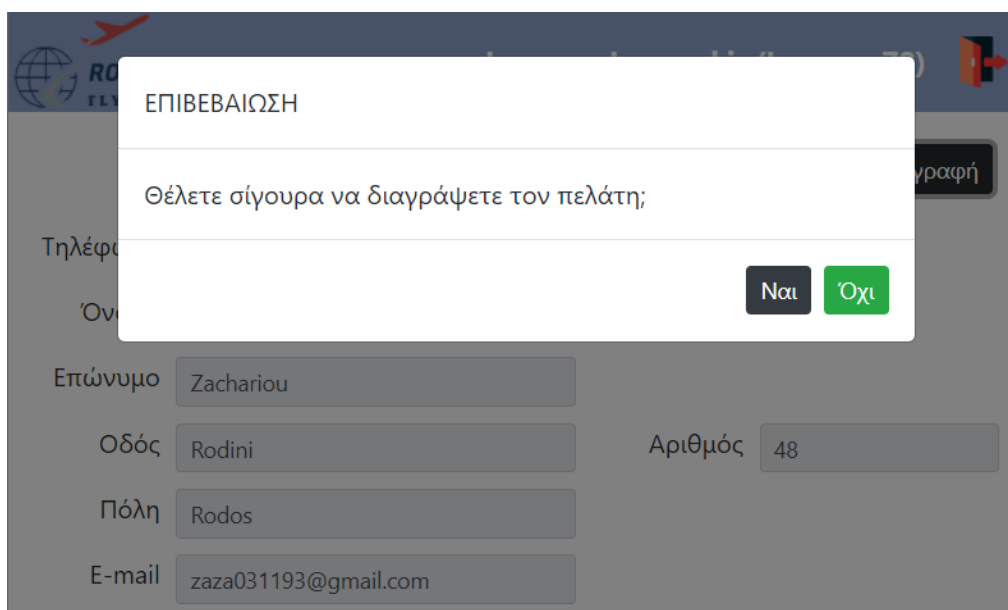
OK

Εικόνα 524. Έλεγχος ήδη καταχωρημένου πελάτη

Διαγραφή πελάτη

Ο χρήστης-ταξιδιωτικός πράκτορας μπορεί να διαγράψει ένα πελάτη πατώντας το κουμπί «Διαγραφή». Στη συνέχεια, η εφαρμογή ζητάει επιβεβαίωση από το χρήστη, ώστε να

διαγράψει τη σχετική εγγραφή.



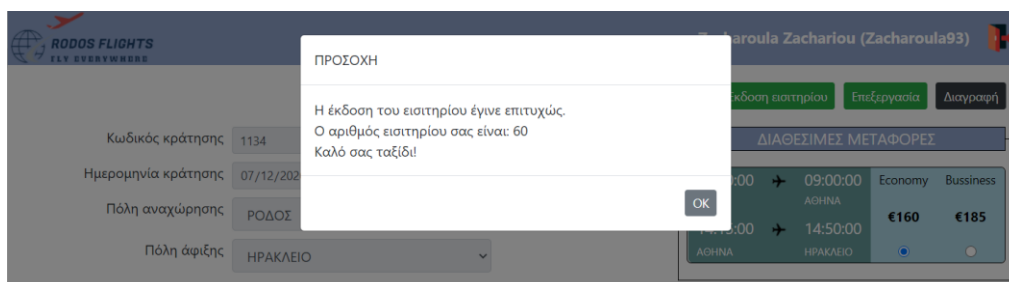
Εικόνα 525. Επιβεβαίωση διαγραφής πελάτη

Ακύρωση αλλαγών στα στοιχεία του πελάτη

Ο χρήστης-ταξιδιωτικός πράκτορας μπορεί να ακυρώσει τυχόν αλλαγές στα στοιχεία του πελάτη πατώντας το κουμπί «Άκυρο». Στην περίπτωση αυτή, η εφαρμογή επαναφορτώνει τη σελίδα του συγκεκριμένου πελάτη.

Έκδοση εισιτηρίου

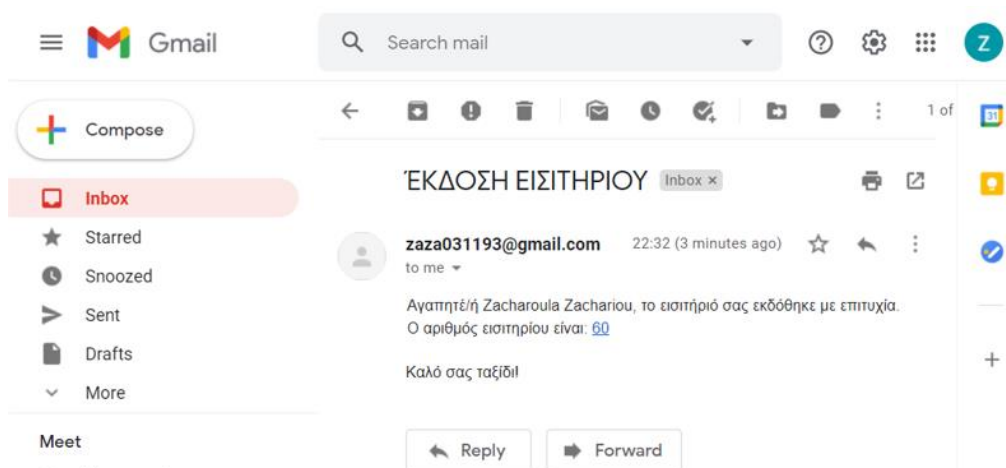
Στη σελίδα κρατήσεων, ο χρήστης-πελάτης και ο χρήστης ταξιδιωτικός πράκτορας μπορεί να εκδώσει το σχετικό εισιτήριο, πατώντας το κουμπί «Έκδοση εισιτηρίου». Σ' αυτή τη περίπτωση, η εφαρμογή εμφανίζει μια σελίδα με τη προεπισκόπηση του εισιτηρίου και στέλνει σχετικό e-mail στο λογαριασμό του χρήστη-πελάτη.



Εικόνα 526. Μήνυμα επιτυχούς έκδοσης εισιτηρίου



Εικόνα 527. Προεπισκόπηση εκδιδόμενου εισιτηρίου



Εικόνα 528. E-mail επιβεβαίωσης έκδοσης εισιτηρίου

Ταμπλό πελάτη

Στη σελίδα αυτή, ο χρήστης-πελάτης μπορεί να διακρίνει τρεις διαφορετικές περιοχές (panels). Η πρώτη περιέχει τα προσωπικά του στοιχεία καθώς και τον αριθμό των χιλιομέτρων που δικαιούται δωρεάν. Η δεύτερη περιέχει τις κρατήσεις που έχει πραγματοποιήσει και η τρίτη τα εισιτήρια που έχει εκδώσει.

RODOS FLIGHTS
FLY EVERYWHERE

Zacharoula Zachariou (Zacharoula93)

ΤΑΜΠΛΟ

ΚΡΑΤΗΣΕΙΣ

ΠΡΟΣΩΠΙΚΑ ΣΤΟΙΧΕΙΑ

Όνοματεπώνυμο: Zacharoula Zachariou
 Διεύθυνση: Rodini 48
 Πόλη: Rodos
 Τηλέφωνο: 2241024568
 Ηλεκτρονική διεύθυνση: zaza031193@gmail.com
 Username: Zacharoula93
 Χιλιόμετρα: 4290.0

ΚΡΑΤΗΣΕΙΣ

Κωδικός κράτησης	Ημερομηνία κράτησης	Πόλη αναχώρησης	Πόλη άφιξης
1136	2020-12-07	ΡΟΔΟΣ	ΘΕΣΣΑΛΟΝΙΚΗ
1137	2020-12-03	ΧΑΝΙΑ	ΧΙΟΣ

ΕΙΣΙΤΗΡΙΑ

Κωδικός εισιτηρίου	Ημερομηνία έκδοσης	Πόλη αναχώρησης	Πόλη άφιξης
60	2020-12-07	ΡΟΔΟΣ	ΗΡΑΚΛΕΙΟ
61	2020-12-07	ΡΟΔΟΣ	ΗΡΑΚΛΕΙΟ

Εικόνα 529. Ταμπλό πελάτη

Άμεση πρόσβαση επεξεργασίας στοιχείων χρήστη

Όλοι οι χρήστες (διαχειριστές, ταξιδιωτικοί πράκτορες και πελάτες), έχουν την δυνατότητα να επεξεργαστούν απευθείας τα προσωπικά στοιχεία τους. Αρκεί να κάνουν κλικ πάνω στο όνομα χρήστη στην πάνω δεξιά μεριά της σελίδας και να εμφανιστεί η σελίδα επεξεργασίας των στοιχείων χρήστη.

RODOS FLIGHTS
FLY EVERYWHERE

GRIGORIS GRIGORIADIS (GREGORY96)

Αλλαγή Κωδικού | Αποθήκευση | Αποθήκευση & Κλείσιμο | Διαγραφή | Άκυρο

Αριθμός Ταυτότητας: AW157825
 Όνομα: GRIGORIS
 Επώνυμο: GRIGORIADIS
 Οδός: ΣΤΑΜΑΤΙΑΔΙ
 Πόλη: ΚΕΚΚΙΣ
 Τηλέφωνο: 2241021475
 Ημερομηνία Γεννησης: 06/06/1996
 Όνομα χρήστη: GREGORY96

Αριθμός: 28

ΚΑΛΩΣΟΡΙΣΣΕΣ GRIGORIS GRIGORIADIS

Εικόνα 530. Άμεση επεξεργασία στοιχείων χρήστη-διαχειριστή

RODOS FLIGHTS
FLY EVERYWHERE

Lampros Lampraklis (Lampros78)

Αλλαγή Κωδικού | Αποθήκευση | Αποθήκευση & Κλείσιμο | Διαγραφή | Άκυρο

Αριθμός Ταυτότητας: AQ774455
 Όνομα: Lampros
 Επώνυμο: Lampraklis
 Οδός: Grigoriou
 Πόλη: Rodos
 Κωδικός: 365252
 Όνομα χρήστη: Lampros78

Αριθμός: 74

ΚΑΛΩΣΟΡΙΣΣΕΣ Lampros Lampraklis

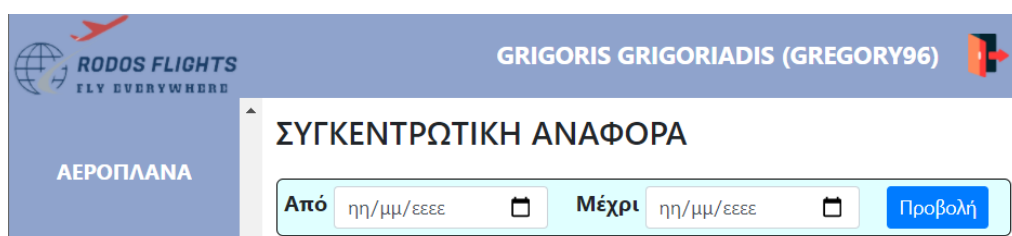
Εικόνα 531. Άμεση επεξεργασία στοιχείων χρήστη-ταξιδιωτικού πράκτορα



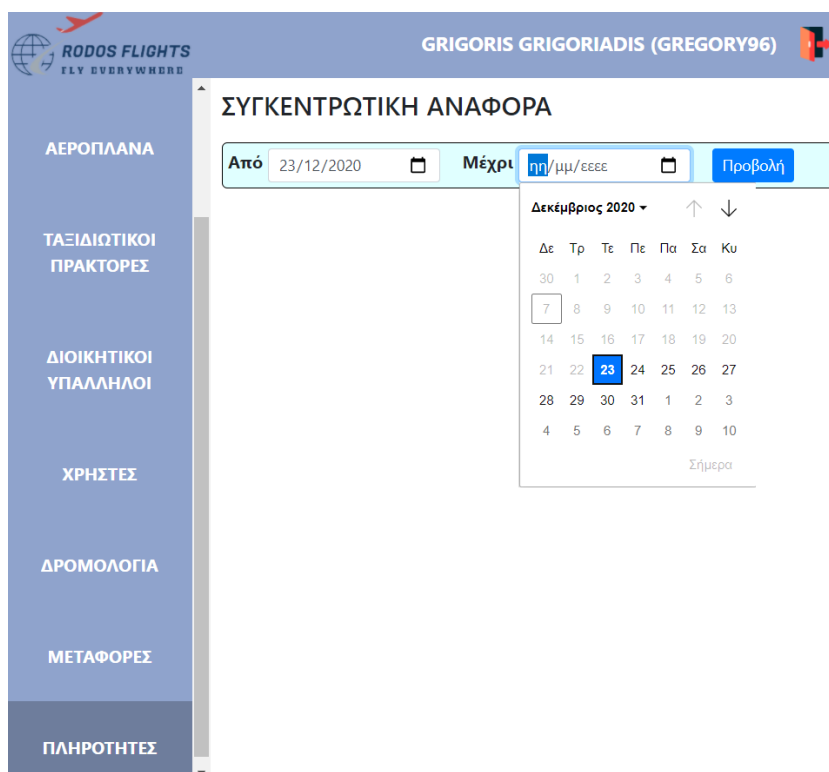
Εικόνα 532. Άμεση επεξεργασία στοιχείων χρήστη-πελάτη

Συγκεντρωτική αναφορά πληρότητας

Στη σελίδα αυτή, ο διαχειριστής έχει στη διάθεσή του μια γραμμή φίλτρων, όπου μπορεί να θέσει το διάστημα ημερομηνιών, για το οποίο θέλει να δει την πληρότητα των θέσεων. Στην περίπτωση που χρήστης επιλέξει την αρχική ημερομηνία, η εφαρμογή «παγώνει» τις προγενέστερες ημερομηνίες από το φίλτρο της τελικής ημερομηνίας και αντίστροφα. Στη συνέχεια, πατώντας το κουμπί «Προβολή», η εφαρμογή του παρουσιάζει μια συγκεντρωτική αναφορά, σχετικά με την πληρότητα των θέσεων για κάθε διαθέσιμο δρομολόγιο.



Εικόνα 533. Γραμμή φίλτρων συγκεντρωτικής αναφοράς



Εικόνα 534. Έλεγχος εγκυρότητας ημερομηνιών

RODOS FLIGHTS
FLY EVERYWHERE

GRIGORIS GRIGORIADIS (GREGORY96)

ΠΙΛΟΤΟΙ

ΑΕΡΟΣΥΝΟΔΟΙ

ΑΕΡΟΠΛΑΝΑ

ΤΑΞΙΔΙΩΤΙΚΟΙ ΠΡΑΚΤΟΡΕΣ

ΣΥΓΚΕΝΤΡΩΤΙΚΗ ΑΝΑΦΟΡΑ

Από 01/12/2020 Μέχρι 31/12/2020 Προβολή

Κωδικός δρομολογίου	Πόλη αναχώρησης	Πόλη άφιξης	Πληρότητα
1010	ΙΩΑΝΝΙΝΑ	ΚΑΡΠΑΘΟΣ	5
1111	ΡΟΔΟΣ	ΑΘΗΝΑ	8
1112	ΑΘΗΝΑ	ΘΕΣΣΑΛΟΝΙΚΗ	3
2222	ΚΩΣ	ΞΑΝΘΗ	4
3333	ΠΑΤΡΑ	ΠΥΡΓΟΣ	4
8963	ΑΘΗΝΑ	ΗΡΑΚΛΕΙΟ	5
8975	ΧΑΝΙΑ	ΧΙΟΣ	1
9856	ΡΟΔΟΣ	ΣΑΜΟΣ	5

Εικόνα 535. Λίστα πληροτήτων διαθέσιμων δρομολογίων

4.3 Συμπεράσματα

Με την παρούσα πτυχιακή εργασία, μπορεί ο καθένας πλέον να λάβει εμπλουτισμένες γνώσεις σχεδιασμού και υλοποίησης μιας ιστοσελίδας. Καθώς και της δημιουργίας βάσης δεδομένων για την δημιουργία και επεξεργασία των εγγραφών μιας εταιρείας ή μιας ιδιωτικής επιχείρησης. Πιο συγκεκριμένα, η παραπάνω εργασία μας βοήθησε να μάθουμε τα εξής:

- Γνώσεις για το σχεδιασμό ιστοσελίδων με τη χρήση κώδικα HTML
- Γνώσεις για τη μορφοποίηση των ιστοσελίδων με τη χρήση κώδικα CSS
- Γνώσεις για τη δημιουργία αλγορίθμων εκτέλεσης ενεργειών με τη χρήση του Django framework, της γλώσσας προγραμματισμού Python, καθώς και της γλώσσας προγραμματισμού JavaScript.
- Γνώσεις δημιουργίας βάσης δεδομένων και αποθηκευμένων διαδικασιών με τη χρήση γλώσσας προγραμματισμού SQL.

Βιβλιογραφία

https://en.wikipedia.org/wiki/SQL_Server_Management_Studio

https://en.wikipedia.org/wiki/Visual_Studio_Code

<https://www.lucidchart.com>

<https://www.w3schools.com>

<https://stackoverflow.com>

<https://code.visualstudio.com/docs>

<https://docs.djangoproject.com>

<https://docs.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver15>

<https://www.codecademy.com/articles/visual-studio-code>

Παραπομπές

[1] Microsoft SQL Server: Ο Microsoft SQL Server είναι ένα σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων που αναπτύχθηκε από τη Microsoft. Ως διακομιστής βάσης δεδομένων, είναι ένα προϊόν λογισμικού με την κύρια λειτουργία της αποθήκευσης και ανάκτησης δεδομένων, όπως απαιτείται από άλλες εφαρμογές λογισμικού, οι οποίες ενδέχεται να εκτελούνται είτε στον ίδιο υπολογιστή είτε σε άλλον υπολογιστή μέσω ενός δικτύου (συμπεριλαμβανομένου του Διαδικτύου). Η Microsoft διαθέτει τουλάχιστον δώδεκα διαφορετικές εκδόσεις του Microsoft SQL Server, που στοχεύουν σε διαφορετικά είδη κοινού και για φόρτο εργασίας, από μικρές εφαρμογές ενός μηχανήματος έως μεγάλες εφαρμογές που αντιμετωπίζουν το Διαδίκτυο με πολλούς ταυτόχρονους χρήστες.

[2] Windows Presentation Foundation (WPF): Το Windows Presentation Foundation, είναι ένα ελεύθερο και ανοιχτού κώδικα γραφικό υποσύστημα (παρόμοιο με το [WinForms](#)) που αναπτύχθηκε αρχικά από τη Microsoft για την απόδοση διεπαφών χρήστη σε εφαρμογές που βασίζονται σε Windows. Το WPF, προηγουμένως γνωστό ως "Avalon", κυκλοφόρησε αρχικά ως μέρος του [.NET Framework 3.0](#) το 2006. Το WPF χρησιμοποιεί το [DirectX](#) και προσπαθεί να παρέχει ένα συνεπές μοντέλο προγραμματισμού για εφαρμογές κτιρίων. Διαχωρίζει τη διεπαφή χρήστη από την επιχειρησιακή λογική και μοιάζει με παρόμοια μοντέλα αντικειμενοστραφούς επεκτάσιμης γλώσσας σήμανσης ([XML](#)), όπως αυτά που εφαρμόζονται στη γλώσσα διεπαφής χρήστη XML ([XUL](#)) και επεκτάσιμων διανυσματικών γραφικών ([SVG](#)).

[3] Git: Το Git είναι ένα κατακευματισμένο σύστημα ελέγχου έκδοσης για παρακολούθηση αλλαγών σε οποιοδήποτε σύνολο αρχείων, το οποίο αρχικά σχεδιάστηκε για το συντονισμό της εργασίας μεταξύ προγραμματιστών που συνεργάζονται στον πηγαίο κώδικα κατά την ανάπτυξη λογισμικού. Οι στόχοι του περιλαμβάνουν ταχύτητα, ακεραιότητα δεδομένων και υποστήριξη για κατακευματισμένες, μη γραμμικές ροές εργασίας.

Το Git δημιουργήθηκε από τον Linus Torvalds το 2005 για την ανάπτυξη του πυρήνα Linux, με άλλους προγραμματιστές του πυρήνα να συμβάλλουν στην αρχική του ανάπτυξη. Από το 2005, ο Junio Hamano είναι ο βασικός συντηρητής. Όπως συμβαίνει με τα περισσότερα άλλα συστήματα ελέγχου έκδοσης, και σε αντίθεση με τα περισσότερα συστήματα διακομιστή-πελάτη, κάθε κατάλογος Git σε κάθε υπολογιστή είναι ένα πλήρες αποθετήριο με πλήρες ιστορικό και πλήρεις δυνατότητες παρακολούθησης εκδόσεων, ανεξάρτητα από την πρόσβαση στο δίκτυο ή έναν κεντρικό διακομιστή. Το Git είναι δωρεάν και λογισμικό ανοιχτού κώδικα που διανέμεται στη γενική άδεια του κοινού έκδοσης 2 ([GNU](#)).

[4] GitHub: Η GitHub Inc. είναι θυγατρική της Microsoft που παρέχει φιλοξενία για ανάπτυξη λογισμικού και έλεγχο έκδοσης χρησιμοποιώντας το Git. Προσφέρει τη λειτουργική διαχείριση κατακευματισμένων εκδόσεων και διαχείριση πηγαίου κώδικα του Git, καθώς και τις δικές της δυνατότητες. Παρέχει έλεγχο πρόσβασης και πολλές δυνατότητες συνεργασίας, όπως παρακολούθηση σφαλμάτων, αιτήματα λειτουργιών, διαχείριση εργασιών, συνεχή ενσωμάτωση και περεταίρω πληροφορίες για κάθε εργασία. Με έδρα την Καλιφόρνια, είναι θυγατρική της Microsoft από το 2018.

Το GitHub προσφέρει τις βασικές υπηρεσίες του δωρεάν. Οι πιο προηγμένες επαγγελματικές και επιχειρηματικές υπηρεσίες του είναι εμπορικές. Οι δωρεάν λογαριασμοί GitHub χρησιμοποιούνται συνήθως για τη φιλοξενία έργων ανοιχτού κώδικα. Από τον Ιανουάριο του 2019, το GitHub προσφέρει απεριόριστα ιδιωτικά αποθετήρια σε όλα τα σχέδια, συμπεριλαμβανομένων των δωρεάν λογαριασμών, αλλά επιτρέπει μόνο έως και τρεις συνεργάτες ανά αποθετήριο ([repository](#)) δωρεάν. Από τις 15 Απριλίου 2020, το δωρεάν

πρόγραμμα επιτρέπει απεριόριστους συνεργάτες, αλλά περιορίζει τα ιδιωτικά αποθετήρια σε 2.000 λεπτά από τις ενέργειες του GitHub ανά μήνα. Από τον Ιανουάριο του 2020, το GitHub αναφέρει ότι έχει πάνω από 40 εκατομμύρια χρήστες και περισσότερα από 190 εκατομμύρια αποθετήρια (συμπεριλαμβανομένων τουλάχιστον 28 εκατομμυρίων δημόσιων αποθετηρίων), καθιστώντας το μεγαλύτερο πλήθος πηγαίου κώδικα στον κόσμο.

[5] Stack Overflow: Το Stack Overflow, είναι ένας ιστότοπος ερωτήσεων και απαντήσεων για επαγγελματίες και ενθουσιώδεις προγραμματιστές. Είναι ένας ιδιωτικός ιστότοπος, ο εμβληματικός ιστότοπος του [Stack Exchange Network](#), που δημιουργήθηκε το 2008 από τους Jeff Atwood και Joel Spolsky. Περιλαμβάνει ερωτήσεις και απαντήσεις σε ένα ευρύ φάσμα θεμάτων στον προγραμματισμό υπολογιστών. Δημιουργήθηκε για να είναι μια πιο ανοιχτή εναλλακτική λύση σε προηγούμενους ιστότοπους ερωτήσεων και απαντήσεων, όπως το [Experts-Exchange](#). Το όνομα για τον ιστότοπο επιλέχθηκε με ψηφοφορία τον Απρίλιο του 2008 από τους αναγνώστες του Coding Horror, του δημοφιλούς blog προγραμματισμού της Atwood. Ο ιστότοπος χρησιμεύει ως πλατφόρμα για τους χρήστες να κάνουν ερωτήσεις και να απαντούν σε ερωτήσεις και, μέσω της συμμετοχής και της ενεργούς συμμετοχής, να ψηφίζουν ερωτήσεις και απαντήσεις πάνω ή κάτω και να επεξεργάζονται ερωτήσεις και απαντήσεις με τρόπο παρόμοιο με το [wiki](#) ή το [Reddit](#). Οι χρήστες του Stack Overflow μπορούν να κερδίσουν πόντους φήμης και "σήματα". Για παράδειγμα, σε ένα άτομο απονέμονται 10 πόντοι φήμης για τη λήψη ψηφοφορίας "πάνω" σε μια ερώτηση ή απάντηση σε μια ερώτηση, και μπορεί να λάβει σήματα για τις αξιόλογες συνεισφορές του, κάτι που αντιπροσωπεύει ένα [gamification](#) του παραδοσιακού ιστότοπου Q&A. Οι χρήστες ξεκλειδώνουν νέα προνόμια με αύξηση της φήμης, όπως η δυνατότητα ψήφου, σχολιασμού και ακόμη και επεξεργασίας αναρτήσεων άλλων ατόμων. Όλο το περιεχόμενο που δημιουργείται από τον χρήστη διαθέτει άδεια χρήσης με άδεια [Creative Commons Attribute-ShareAlike](#), έκδοση 2.5, 3.0 ή 4.0, ανάλογα με την ημερομηνία συνεισφοράς του περιεχομένου. Το κλείσιμο ερωτήσεων είναι μια κύρια διαφοροποίηση από άλλους ιστότοπους Q&A όπως το Yahoo! Απαντήσεις και τρόπος αποφυγής ερωτήσεων χαμηλής ποιότητας. Ο μηχανισμός αναθεωρήθηκε το 2013. Οι ερωτήσεις που επεξεργάστηκαν αφού τέθηκαν "σε αναμονή" εμφανίζονται τώρα σε μια ουρά κριτικής. Ο Jeff Atwood δήλωσε το 2010 ότι οι διπλές ερωτήσεις δεν θεωρούνται πρόβλημα, αλλά μάλλον αποτελούν πλεονέκτημα εάν τέτοιες πρόσθετες ερωτήσεις οδηγούν σε επιπλέον επισκεψιμότητα στον ιστότοπο πολλαπλασιάζοντας τις σχετικές επισκέψεις λέξεων-κλειδιών στις μηχανές αναζήτησης. Από τον Ιανουάριο του 2019, το Stack Overflow έχει περισσότερους από 10 εκατομμύρια εγγεγραμμένους χρήστες και ξεπέρασε τα 16 εκατομμύρια ερωτήσεις στα μέσα του 2018. Με βάση τον τύπο των ετικετών που ανατέθηκαν σε ερωτήσεις, τα κορυφαία οκτώ θέματα που συζητήθηκαν στον ιστότοπο είναι: JavaScript, Java, C #, PHP , Android, Python, jQuery και HTML. Το Stack Overflow διαθέτει επίσης μια ενότητα Jobs για να βοηθήσει τους προγραμματιστές να βρουν την επόμενη ευκαιρία τους. Για τους εργοδότες, το Stack Overflow παρέχει εργαλεία για την επωνυμία της επιχείρησής τους, τη διαφήμιση των ανοιγμάτων τους στον ιστότοπο και την προέλευση των υποψηφίων από τη βάση δεδομένων του Stack Overflow για προγραμματιστές που είναι ανοιχτοί σε επαφή.

[6] Entity-Relationship model (E-R): Ένα μοντέλο οντοτήτων-συσχετίσεων (ή μοντέλο ER), περιγράφει αλληλένδετα πράγματα ενδιαφέροντος σε έναν συγκεκριμένο τομέα γνώσης. Ένα βασικό μοντέλο ER αποτελείται από τύπους οντοτήτων (πού ταξινομούν τα πράγματα που ενδιαφέρουν) και καθορίζει σχέσεις που μπορούν να υπάρχουν μεταξύ οντοτήτων (περιπτώσεις αυτών των τύπων οντοτήτων). Στη μηχανική λογισμικού, ένα μοντέλο ER συνήθως διαμορφώνεται για να αντιπροσωπεύει πράγματα που πρέπει να θυμάται μια επιχείρηση προκειμένου να εκτελεί επιχειρηματικές διαδικασίες. Κατά συνέπεια, το μοντέλο

ER γίνεται ένα αφηρημένο μοντέλο δεδομένων, το οποίο ορίζει μια δομή δεδομένων ή πληροφοριών που μπορεί να εφαρμοστεί σε μια βάση δεδομένων, συνήθως μια σχεσιακή βάση δεδομένων. Μοντελοποίηση οντοτήτων-σχέσεων αναπτύχθηκε για βάση δεδομένων και σχεδιασμό από τον Peter Chen και δημοσιεύθηκε σε ένα έγγραφο του 1976, με παραλλαγές της ιδέας που υπήρχαν προηγουμένως. Ορισμένα μοντέλα ER παρουσιάζουν υπερ-οντότητες και οντότητες που συνδέονται με σχέσεις γενίκευσης-εξειδίκευσης και ένα μοντέλο ER μπορεί να χρησιμοποιηθεί επίσης στην προδιαγραφή οντολογιών για συγκεκριμένους τομείς.