

Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης



Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Μηχανικών Πληροφορικής Τ.Ε.



Πτυχιακή εργασία

Τίτλος :Ναυτιλιακές Εταιρίες

Καριώτης Δήμος (ΑΜ:2163)

Επιβλέπων καθηγητής : Παπαδάκης Νικόλαος

ΗΡΑΚΛΕΙΟ

2015

Ευχαριστίες

Μετά από αρκετά χρόνια στη πόλη του Ηρακλείου, όπου εδρεύει το τμήμα μου, ήρθε η στιγμή της παρουσίασης της πτυχιακής μου εργασίας και η αποφοίτησής μου. Κατά τη διάρκεια των σπουδών μου, αποκόμισα πολλά πράγματα από το περιβάλλον, τους συμφοιτητές μου, τους καθηγητές και τους κατοίκους του Ηρακλείου. Θα ήθελα να ευχαριστήσω κατ' αρχάς, τους γονείς μου και τις αδερφές μου, που με στήριξαν σε πολλές δύσκολες στιγμές κατά τη διάρκεια των φοιτητικών μου σπουδών, την οικονομική αλλά και ψυχολογική βοήθεια που είχα όλα αυτά τα χρόνια. Επίσης, να ευχαριστήσω τον καθηγητή μου κ. Νίκο Παπαδάκη, ο οποίος μου εμπιστεύτηκε το παραπάνω θέμα, παρά το μεγάλο φόρτο εργασίας του.. Τέλος, ένα μεγάλο ευχαριστώ στη γυναίκα μου που με στήριξε και με βοήθησε να φέρω εις πέρας την εργασία μου.

Abstract

The purpose of my thesis is the creation of a web application, in which you can book a short-sea shipping ticket. Users can login either as a simple user, either as a travel agent and either as an administrator. Each of these users has specific capabilities and limitations.

For the implementation of application used the following tools and programming languages. Xampp, a tool very useful for creating local host (local server), the languages PHP, HTML, JavaScript. Also, technologies content formatting CSS to make the application accessible to the user. MySQL is management system source relational databases open source software, which are queries to the database.

Σύνοψη

Σκοπός της πτυχιακής είναι η δημιουργία μιας διαδικτυακής εφαρμογής, με την οποία μπορεί κάποιος να κάνει κράτηση ενός ακτοπλοϊκού εισιτηρίου. Ο χρήστης θα μπορεί να κάνει login είτε ως απλός χρήστης, είτε ως ταξιδιωτικός πράκτορας, είτε ως administrator. Κάθε ένας από τους παραπάνω χρήστες έχει συγκεκριμένες δυνατότητες και περιορισμούς.

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκαν τα ακόλουθα εργαλεία και γλώσσες προγραμματισμού. Xampp, ένα εργαλείο πολύ χρήσιμο για τη δημιουργία local host (τοπικού server), οι γλώσσες PHP, HTML, JavaScript. Επίσης, τεχνολογίες μορφοποίησης περιεχομένου CSS για να γίνει η εφαρμογή προσιτή προς το χρήστη. Η MySQL, είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων ανοιχτού κώδικα, με την οποία γίνονται τα queries προς τη βάση δεδομένων.

Πίνακας περιεχομένων

Ευχαριστίες	i
Abstract	ii
Σύνοψη	iii
Πίνακας περιεχομένων	iv
Πίνακας Εικόνων.....	vi
Λίστα Πινάκων.....	ix
1. ΕΙΣΑΓΩΓΗ	1
1.1 Περίληψη.....	1
1.2 Κίνητρο για τη διεξαγωγή της εργασίας	1
1.3 Σκοπός και στόχοι της εργασίας.....	1
1.4 Δομή εργασίας.....	2
2. ΜΕΘΟΔΟΛΟΓΙΑ ΥΛΟΠΟΙΗΣΗΣ	3
2.1 XAMPP	3
2.2 PHP.....	3
2.3 ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ	3
2.4 SQL	3
2.5 HTML.....	4
2.6 JAVASCRIPT	4
2.7 CSS.....	4
2.8 MYSQL.....	4
3. STATE OF THE ART.....	5
3.1 XAMPP.....	5
3.1.1 Τι είναι το Xampp	5
3.1.2 Τι περιλαμβάνει.....	6
3.2 PHP.....	7
3.2.1 Τι είναι η PHP	7
3.2.2 Ιστορική Αναδρομή.....	7
3.2.3 Πλεονεκτήματα & Μειονεκτήματα της PHP	8
3.2.4 Μειονεκτήματα.....	8
3.2.5 Σύνταξη της PHP.....	9
3.2.6 Μεταβλητές	10
3.2.7 Cookies.....	10
3.2.8 Sessions	10
3.2.9 PHP-MySql	11
3.3 ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ.....	11
3.3.1 Οι Πληροφορίες και τα Δεδομένα.....	12
3.3.2 Οργάνωση Αρχείων.....	13
3.3.3 Τι είναι Βάση Δεδομένων και Ιστορικά Στοιχεία.....	13
3.3.4 Συστήματα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ-DBMS)	14
3.3.5 Διαχειριστής Βάσεων Δεδομένων	15
3.3.6 Μοντέλα Δεδομένων	15
3.3.7 Μοντέλο Οντοτήτων-Συσχετίσεων	16
3.3.8 Σχεσιακό Μοντέλο	16
3.3.9 Ασφάλεια Βάσεων Δεδομένων.....	17
3.4 SQL.....	18
3.4.1 Ιστορία της SQL.....	18
3.4.2 Τι είναι η SQL.....	18

3.4.3	Εντολές και Τρόπος Χρήσης της Sql	19
3.5	PHPMYADMIN.....	21
3.5.1	Ιστορία του phpMyAdmin.....	21
3.5.2	Τι είναι το phpMyAdmin.....	22
3.5.3	Web Interface	22
3.5.4	Δημιουργία βάσης δεδομένων.....	23
3.6	MYSQL.....	25
3.6.1	Ιστορία.....	25
3.6.2	Πλεονεκτήματα	25
3.6.3	MySql-Php.....	26
3.7	HTML.....	27
3.7.1	Ιστορία και Εκδόσεις.....	28
3.7.2	Ετικέτες	28
3.7.3	XHTML.....	29
3.7.4	HTML vs XHTML.....	30
3.8	JAVASCRIPT	32
3.8.1	Ιστορία.....	34
3.8.2	Μοντέλο Εκτέλεσης	34
3.8.3	Javascript vs Java	35
3.8	CSS.....	36
3.8.1	Τι είναι τα CSS.....	36
3.8.2	Διαφορές CSS-HTML.....	36
3.8.2	Πως λειτουργούν	37
4.	ΘΕΩΡΗΤΙΚΟ ΜΕΡΟΣ ΤΗΣ ΠΤΥΧΙΑΚΗΣ.....	40
4.1	Ανάλυση του προβλήματος.....	40
4.2	Σχεδιασμός Υλοποίησης	40
5.	ΠΡΑΚΤΙΚΟ ΜΕΡΟΣ ΤΗΣ ΠΤΥΧΙΑΚΗΣ	41
5.1	Εγκατάσταση απαραίτητου λογισμικού	41
5.1.1	Εγκατάσταση του Xampp.....	41
5.1.2	Ρυθμίσεις Xampp	45
5.2	Σχεδίαση Βάσης Δεδομένων	48
5.3	Στατικές σελίδες.....	52
5.3.1	Contact.php.....	52
5.4	Δυναμικές σελίδες.....	53
5.4.1	Σελίδες Χρήστη.....	55
5.4.2	Σελίδες Administrator	64
6	Αποτελέσματα	82
6.1	Συμπεράσματα.....	82
6.2	Μελλοντική Εργασία και Επεκτάσεις	82
	Βιβλιογραφία.....	83

Πίνακας Εικόνων

- Εικόνα 1: Δομή εργασίας
- Εικόνα 2: Λογότυπο του Xampp
- Εικόνα 3: Εργαλεία του Xampp
- Εικόνα 4: Λογότυπο της Php
- Εικόνα 5: Server-side language
- Εικόνα 6: Προσπέλαση PHP σελίδων
- Εικόνα 7: Ακολουθιακό Διάγραμμα
- Εικόνα 8: Πληροφορίες και Δεδομένα
- Εικόνα 9: Η αρχιτεκτονική ANSI ενός ΣΔΒΔ
- Εικόνα 10: Διάγραμμα οντοτήτων συσχετίσεων
- Εικόνα 11: Διάγραμμα σχεσιακού μοντέλου
- Εικόνα 12: Λογότυπο της Sql
- Εικόνα 13: Λογότυπο του PhpMyAdmin
- Εικόνα 14: Web Interface-1 phpmyadmin
- Εικόνα 15: Web Interface-2 phpMyAdmin
- Εικόνα 16: Web Interface-3 phpMyAdmin
- Εικόνα 17: Web Interface-4 phpMyAdmin
- Εικόνα 18: Web Interface-5 phpMyAdmin
- Εικόνα 19: Λογότυπο της MySql
- Εικόνα 20: Λογότυπο HTML5
- Εικόνα 21: Διαφορές HTML-XHTML
- Εικόνα 22: Λογότυπο του CSS
- Εικόνα 23: Εγκατάσταση λογισμικού
- Εικόνα 24: Download Xampp
- Εικόνα 25: Save Xampp file
- Εικόνα 26: Setup Xampp 1
- Εικόνα 27: Setup Xampp 2
- Εικόνα 28: Setup Xampp 3
- Εικόνα 29: Setup Xampp 4
- Εικόνα 30: Setup Xampp 5
- Εικόνα 31: Finish Xampp installation
- Εικόνα 32: Xampp control panel 1
- Εικόνα 33: Xampp control panel 2
- Εικόνα 34: Xampp Admin Apache 1
- Εικόνα 35: Xampp Admin Apache 2
- Εικόνα 36: Xampp Admin Apache 3
- Εικόνα 37: Xampp Admin Apache 4
- Εικόνα 38: Βάση δεδομένων εφαρμογής
- Εικόνα 39: member-Create and Insert Table
- Εικόνα 40: tblcity-Create and Insert Table
- Εικόνα 41: tblcompany-Create and Insert Table
- Εικόνα 42: tblemployee-Create and Insert Table
- Εικόνα 43: tblreservation-Create and Insert Table
- Εικόνα 44: tbltransportation-Create and Insert Table

Εικόνα 45: tblroute-Create and Insert Table
Εικόνα 46: Contact.php
Εικόνα 47: Πρόσβαση στη βάση δεδομένων
Εικόνα 48: catalogdbinfo
Εικόνα 49: Session_start
Εικόνα 50: Clock
Εικόνα 51: Index.php
Εικόνα 51: City.php
Εικόνα 52: Cities.php
Εικόνα 53: Δρομολόγια
Εικόνα 54: Search route
Εικόνα 55: Routeresults_php
Εικόνα 56: Ticketsreservation_php
Εικόνα 57: Loginmember.php
Εικόνα 58: Εγγραφή Μέλους
Εικόνα 59: Διαθέσιμα Εισιτήρια
Εικόνα 60: Υπηρεσίες Μέλους
Εικόνα 61: Κρατήσεις εισιτηρίων
Εικόνα 62: Ακύρωση κράτησης 1
Εικόνα 63: Ακύρωση κράτησης 2
Εικόνα 64: Ακύρωση κράτησης 3
Εικόνα 65: Έλεγχος ακύρωσης εισιτηρίου
Εικόνα 66: Τροποποίηση στοιχείων
Εικόνα 67: Ταξιδιωτικός Πράκτορας
Εικόνα 68: Loginadmin.php
Εικόνα 69: Ταυτοποίηση Διαχειριστή
Εικόνα 70: Ταυτοποίηση Διαχειριστή 2
Εικόνα 71: Πρόσβαση ή Αποτροπή
Εικόνα 72: Admin/admin.php
Εικόνα 73: Admins Link
Εικόνα 74: Προσθήκη νέας πόλης
Εικόνα 75: Google maps
Εικόνα 76: Τροποποίηση πόλης
Εικόνα 77: Τροποποίηση-Διαγραφή Links
Εικόνα 78: Τροποποίηση πόλης 2
Εικόνα 79: Προσθήκη πλοίου
Εικόνα 80: Τροποποίηση πλοίου 1
Εικόνα 81: Τροποποίηση πλοίου 2
Εικόνα 82: Προσθήκη δρομολογίου
Εικόνα 83: Διαθεσιμότητα πλοίου 1
Εικόνα 83: Διαθεσιμότητα πλοίου 2
Εικόνα 84: Τροποποίηση δρομολογίου
Εικόνα 85: Προσθήκη υπαλλήλου
Εικόνα 86: Τροποποίηση υπαλλήλου 1
Εικόνα 87: Τροποποίηση υπαλλήλου 2
Εικόνα 88: Διαγραφή υπαλλήλου
Εικόνα 89: Μη διαγραφή πλοιάρχου

- Εικόνα 90: Προσθήκη ναυτιλιακής εταιρίας
- Εικόνα 91: Τροποποίηση ναυτιλιακής εταιρίας 1
- Εικόνα 92: Τροποποίηση ναυτιλιακής εταιρίας 2
- Εικόνα 93: Προσθήκη ταξιδιωτικού πράκτορα
- Εικόνα 94: Τροποποίηση ταξιδιωτικού πράκτορα 1
- Εικόνα 95: Τροποποίηση ταξιδιωτικού πράκτορα 2
- Εικόνα 96: Συγκεντρωτική κατάσταση δρομολογίων
- Εικόνα 97: Διπλοκράτηση
- Εικόνα 98: Επιβάρυνση ακύρωσης
- Εικόνα 99: Πόλεις σε αλφαβητική σειρά

Λίστα Πινάκων

- Πίνακας 1: Σχετικές μεθοδολογίες υλοποίησης
- Πίνακας 2: Παράδειγμα-1 Php
- Πίνακας 3: Παράδειγμα-2 Php
- Πίνακας 4: Παράδειγμα-3 Php
- Πίνακας 5: Παράδειγμα-4 Php
- Πίνακας 6: Εντολή Select
- Πίνακας 7: Εντολή Insert
- Πίνακας 8: Εντολή Update
- Πίνακας 9: Εντολή Delete
- Πίνακας 10: Εντολή Create Table
- Πίνακας 11: Εντολή Alter Table(1)
- Πίνακας 12: Εντολή Alter Table(2)
- Πίνακας 13: Εντολή Alter Table(3)
- Πίνακας 14: Εντολή Drop Table
- Πίνακας 15: Παράδειγμα-1 MySql
- Πίνακας 16: Παράδειγμα-2 MySql
- Πίνακας 17: Παράδειγμα-3 MySql
- Πίνακας 18: Παράδειγμα δομής HTML
- Πίνακας 19: Παράδειγμα δομής XHTML
- Πίνακας 20: Ετικέτες της HTML
- Πίνακας 21: Παράδειγμα Javascript
- Πίνακας 22: Java vs Javascript
- Πίνακας 23: External Style Sheet
- Πίνακας 24: Internal Style Sheet
- Πίνακας 25: Inline Style Sheet
- Πίνακας 26: Βασικές ιδιότητες
- Πίνακας 27: CSS Επιλογείς
- Πίνακας 28: Πίνακας member
- Πίνακας 29: Πίνακας tblcity
- Πίνακας 30: Πίνακας tblcompany
- Πίνακας 31: Πίνακας tblemployee
- Πίνακας 32: Πίνακας tblreservation
- Πίνακας 33: Πίνακας tbltransportation
- Πίνακας 34: Πίνακας tblroute

1. ΕΙΣΑΓΩΓΗ

1.1 Περίληψη

Ο σκοπός της εργασίας είναι ο σχεδιασμός, η υλοποίηση και η παρουσίαση μιας διαδικτυακής εφαρμογής κρατήσεων ακτοπλοϊκών εισιτηρίων. Ο επισκέπτης έχει τη δυνατότητα να παρακολουθήσει τα διαθέσιμα δρομολόγια, το μέσο με το οποίο πραγματοποιείται η διαδρομή, την ώρα αναχώρησης, την ώρα άφιξης και την τιμή του εισιτηρίου. Για να μπορέσει να κάνει την κράτηση που επιθυμεί, θα πρέπει να κάνει εγγραφή ως Μέλος και στη συνέχεια να συνδεθεί με τα στοιχεία του, username και password.

Για τις απαιτήσεις της πτυχιακής έχουμε δημιουργήσει τρεις διαφορετικούς τύπους χρηστών, μη συμπεριλαμβανομένου τον απλό επισκέπτη που μπορεί να δει μόνο τα διαθέσιμα δρομολόγια. *Μέλος*, ο οποίος μπορεί να δει τα διαθέσιμα δρομολόγια και να κάνει κράτηση ή ακύρωση ενός ή περισσότερων εισιτηρίων *Ταξιδιωτικός Πράκτορας* ο οποίος θα μπορεί να κάνει κρατήσεις στο όνομα κάποιου πελάτη και ακύρωση μόνο των δικών του εισιτηρίων και όχι άλλων πρακτόρων *Διαχειριστής*, ο οποίος θα κάνει την προσθαφαίρεση δρομολογίων, πλοίων, τοποθεσίας, υπαλλήλου, ταξιδιωτικού πράκτορα και εκτύπωση των δρομολογίων.

Η εφαρμογή διέπεται από κάποιους κανόνες ακεραιότητας οι οποίοι είναι το σημαντικό κομμάτι της πτυχιακής εργασίας. Σαν κανόνες ακεραιότητας έχουν εξεταστεί τα εξής σημεία:

- Δεν μπορεί κάποιος πελάτης να κάνει κράτηση σε 2 ή περισσότερα εισιτήρια, των οποίων οι ώρες αναχώρησης συμπίπτουν.
- Δεν μπορεί ένα πλοίο να εκτελεί δύο ή περισσότερα δρομολόγια, των οποίων οι ώρες αναχώρησης είναι ίδιες
- Δεν μπορεί ο ίδιος πλοίαρχος να βρίσκεται σε δύο ή περισσότερα ταξίδια που έχουν τις ίδιες ώρες αναχώρησης
- Σε περίπτωση ακύρωσης μίας κράτησης από κάποιον πελάτη, η οποία γίνεται σε λιγότερο από 24 ώρες πριν την εκκίνηση του δρομολογίου, θα υπάρχει ποινή επιστροφής των χρημάτων του, που θα φτάνει το 10% της τιμής του εισιτηρίου.

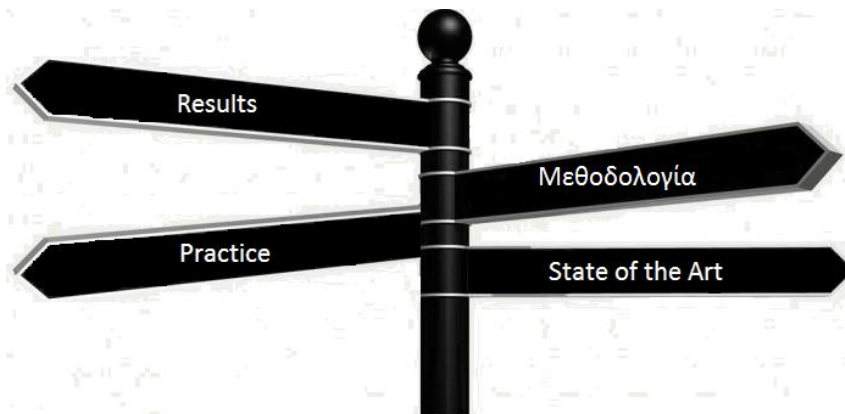
1.2 Κίνητρο για τη διεξαγωγή της εργασίας

Κίνητρο για τη διεξαγωγή της εργασίας αποτέλεσε ο τρόπος κατασκευής μιας δυναμικής διαδικτυακής εφαρμογής που παρέχει τη δυνατότητα στους χρήστες να μπορούν να κλείσουν ακτοπλοϊκά εισιτήρια για τους προορισμούς που επιθυμούν. Κατ' επέκταση από τη μεριά του προγραμματιστή, να μπορέσω να αποκτήσω περισσότερες γνώσεις σε θέματα βάσεων δεδομένων και την λειτουργία της γλώσσας προγραμματισμού PHP.

1.3 Σκοπός και στόχοι της εργασίας

Σκοπός της παρούσας εργασίας είναι η υλοποίηση μιας εφαρμογής που θα επιτρέπει σε οποιονδήποτε άνθρωπο να μπορεί να βρει δρομολόγια ακτοπλοϊκών εισιτηρίων, από και προς διάφορα λιμάνια της Ελλάδας και της Ιταλίας και να μπορεί να κάνει κράτηση εισιτηρίου. Από τη πλευρά μου, ως το άτομο που υλοποιεί την εργασία αυτή, η εξοικείωση με την PHP θα με παροτρύνει να εισέλθω στον τρόπο προγραμματισμού διαδικτυακών εφαρμογών και ιστοσελίδων, αφού τη σημερινή εποχή το Internet είναι αναπόσπαστο κομμάτι από την καθημερινότητα των περισσότερων ανθρώπων.

1.4 Δομή εργασίας



Εικόνα 1: Δομή εργασίας

Η δομή της πτυχιακής εργασίας είναι η παρακάτω:

- **Κεφάλαιο 1: Εισαγωγή**
Στο πρώτο κεφάλαιο γίνεται αναφορά σχετικά με ότι αφορά τα κίνητρα και το σκοπό της πτυχιακής εργασίας.
- **Κεφάλαιο 2:Μεθοδολογία Υλοποίησης**
Στο δεύτερο κεφάλαιο κάνουμε μία εισαγωγή στη μεθοδολογία και τον τρόπο με τον οποίο πορευόμαστε για να υλοποιήσουμε το project. Γίνεται μία περιληπτική αναφορά στα εργαλεία και τις τεχνικές που ακολουθήθηκαν.
- **Κεφάλαιο 3:State of the Art**
Στο τρίτο κεφάλαιο πραγματοποιείται το σχέδιο δράσης της εργασίας. Αναλύονται περιληπτικά τα εργαλεία και οι γλώσσες προγραμματισμού που χρησιμοποιήθηκαν. Εκτενή περιγραφή γίνεται για την PHP και τις βάσεις δεδομένων, γιατί είναι τα κύρια εργαλεία για την παρούσα πτυχιακή.
- **Κεφάλαιο 4:Θεωρητικό μέρος πτυχιακής**
Στο τέταρτο κεφάλαιο αναφέρουμε τις απαιτήσεις του συστήματος, γίνεται η ανάλυση του προβλήματος και προγραμματίζεται ο σχεδιασμός υλοποίησης.
- **Κεφάλαιο 5:Πρακτικό μέρος πτυχιακής**
Στο πέμπτο κεφάλαιο, αρχικά θα γίνει αναφορά με την εγκατάσταση των απαραίτητων εργαλείων που θα χρησιμοποιηθούν και στη συνέχεια θα γίνει λεπτομερή αναφορά του πρακτικού μέρους της εργασίας με εικόνες.
- **Κεφάλαιο 6:Αποτελέσματα**
Στο έκτο κεφάλαιο παρουσιάζονται τα αποτελέσματα της εφαρμογής, τα συμπεράσματα που βγήκαν και η μελλοντική επέκταση της εφαρμογής.

2. ΜΕΘΟΔΟΛΟΓΙΑ ΥΛΟΠΟΙΗΣΗΣ

Παρακάτω θα αναλύσουμε τους τρόπους και τα εργαλεία που χρησιμοποιήθηκαν για την σχεδίαση, υλοποίηση και κατασκευή της διαδικτυακής μας εφαρμογής. Ως επί το πλείστον, αυτά είναι open source εργαλεία, τα οποία συνδέονται μεταξύ τους και μας δίνουν ένα πολύ αξιόλογο αποτέλεσμα. Παραδείγματος χάρη, το Xampp μπορεί να διαχειριστεί πολύ οργανωμένα και με εύκολο τρόπο τις open source δυνατότητες που μας παρέχουν η MySql και η PHP. Βασικός παράγοντας για να υλοποιηθεί η εφαρμογή, είναι η εκμάθηση βάσεων δεδομένων και συγκεκριμένα η SQL. Αναλυτικότερα:

2.1 XAMPP

Θα χρησιμοποιήσουμε το Xampp ως μία πλατφόρμα για τη σχεδίαση διαδικτυακών εφαρμογών και ιστοσελίδων, το οποίο περιέχει τον εξυπηρετητή Apache HTTP, την βάση δεδομένων MySql, PHP. Το μικρό μέγεθός του και η φορητότητά του, το κάνει ιδανικό εργαλείο για σπουδαστές και για δοκιμές εφαρμογών σε PHP και MySql.

2.2 PHP

Η PHP είναι μία αντικειμενοστραφής γλώσσα προγραμματισμού σχεδιασμένη ειδικά για το διαδίκτυο. Σκοπός της είναι να δώσει στους web developers τη δυνατότητα να δημιουργούν δυναμικού περιεχομένου ιστοσελίδες. Το δυνατό της σημείο είναι ότι μπορεί να υποστηρίξει μία ευρεία γκάμα από βάσεις δεδομένων, όπως MySql, Oracle, Solid, Sybase, Unixdbm, PostgreSQL. Σημαντικό της χαρακτηριστικό επίσης είναι, ότι χρησιμοποιεί πρωτόκολλα όπως HTTP, POP3, IMAP, SMTP, NNTP.

2.3 ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

Μία βάση δεδομένων είναι ένα ολοκληρωμένο σύστημα που αποτελείται από δεδομένα και το κατάλληλο λογισμικό, τα οποία με τη συνεισφορά του hardware βοηθούν στην ενημέρωση και την πληροφόρηση των χρηστών. Ένα πρόγραμμα που διαχειρίζεται βάσεις δεδομένων ονομάζεται Σύστημα Διαχείρισης Βάσεων Δεδομένων (DBMS). Με αυτό μπορούμε να αποθηκεύσουμε, να τροποποιήσουμε, να προσθέσουμε ή να διαγράψουμε τα δεδομένα που έχουμε αποθηκεύσει. Περαιτέρω ανάλυση για αυτό θα γίνει σε επόμενο κεφάλαιο.

2.4 SQL

Η SQL (Structure Query Language) είναι η δομημένη γλώσσα ερωτημάτων που μας δίνει πρόσβαση σε βάσεις δεδομένων και αποτελεί ένα standard του ANSI. Οι εντολές της χρησιμοποιούνται για να κάνουμε ανάκτηση και ενημέρωση των δεδομένων σε μία βάση δεδομένων. Συνεργάζεται με διάφορα προγράμματα βάσεων δεδομένων, μερικά από αυτά είναι η Access, η Oracle, η Sybase, η Informix, η Microsoft SQL Server και πολλά άλλα.

2.5 HTML

Η HTML δεν είναι απόλυτα μία γλώσσα προγραμματισμού, αφού δεν χρησιμοποιεί συνθήκες ελέγχου ή επαναλήψεων. Είναι όμως πολύ σημαντικό εργαλείο για τη δημιουργία μίας ιστοσελίδας. Μας βοηθάει να τοποθετούμε τα κείμενα, τις εικόνες, τα web στο site μας, τη μορφοποίηση και τον τρόπο με τον οποίο θα κάνουμε την προβολή του.

2.6 JAVASCRIPT

Με τη JavaScript μπορούμε να δημιουργήσουμε σενάρια που να εκτελούν αυτόματες λειτουργίες. Στην δική μας περίπτωση για παράδειγμα, χρησιμοποιούμε JavaScript για τον έλεγχο του πλήθους των διαθέσιμων εισιτηρίων και διαθέσιμων οχημάτων που μπορούν να επιβιβαστούν σε ένα πλοίο.

2.7 CSS

Για τη δημιουργία του template χρησιμοποιήθηκε η τεχνική των Cascade Style Sheet (CSS). Η CSS χρησιμοποιείται για τον έλεγχο της εμφάνισης της εφαρμογής, η οποία είναι γραμμένη σε HTML. Το αποτέλεσμα ήταν το αρχείο style.css το οποίο χρησιμοποιείται από κάθε ιστοσελίδα για τον τρόπο με τον οποίο θα εμφανιστούν τα περιεχόμενα.

2.8 MYSQL

Η MySql θα μας βοηθήσει να διαχειριστούμε τη βάση δεδομένων. Συγκριτικά με άλλα συστήματα διαχείρισης βάσεων δεδομένων όπως Oracle, Solid, Sybase, Unixdbm, PostgreSQL, είναι πολύ γρήγορη, μπορούν να έχουν πρόσβαση πολλοί χρήστες ταυτόχρονα και διασφαλίζει πως μόνο οι πιστοποιημένοι χρήστες έχουν πρόσβαση σε αυτή. Για τις ανάγκες της πτυχιακής, επιλέχθηκε η MySql διότι είναι διαθέσιμη δωρεάν, έχει ευκολία στη χρήση και μπορεί να χρησιμοποιηθεί σε διαφορετικά συστήματα Unix, όπως και Microsoft Windows.

ΜΕΘΟΔΟΛΟΓΙΑ	ΒΙΒΛΙΟΓΡΑΦΙΑ
XAMPP	https://www.apachefriends.org/index.html
PHP	http://www.w3schools.com/php/default.asp
ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ	http://www.slideshare.net/gkhglpzg/ss-8565914
SQL	http://www.w3schools.com/sql/sql_intro.asp
HTML	http://www.wlearn.gr/index.php/html-html-32
JAVASCRIPT	http://www.w3schools.com/js/default.asp
CSS	http://www.w3schools.com/css/
MYSQL	http://www.mysql.com/

Πινάκας 1: Σχετικές μεθοδολογίες υλοποίησης

3. STATE OF THE ART

Ο 20ος αιώνας, ήταν για τον τομέα της πληροφορικής η χρυσή τομή για να εισέλθουμε σε ένα διαφορετικό τρόπο ζωής. Ειδικότερα, τα τέλη του 20ου αιώνα ήταν η περίοδος που κάτι νέο και πρωτόγνωρο εμφανίστηκε, το Διαδίκτυο. Για κάποιους ήταν κάτι το οποίο θα επέφερε ζημίες και καταστροφές αλλά για κάποιους άλλους ήταν ένας τρόπος να αλλάξουν την καθημερινότητά τους και να δούνε τον κόσμο μέσα από ένα άλλο πρίσμα. Καθώς το Internet αναπτυσσόταν όλο και περισσότερο, οι δυνατότητες που μας επιφύλασσε είναι πραγματικά τεράστιες και κατά μία έννοια μας γλυτώνουν από χρόνο και χρήμα. Διανύοντας τη δεύτερη δεκαετία του 21ου αιώνα, το Internet πλέον χρησιμοποιείται από το πλήθος του πληθυσμού της γης. Με ελάχιστες γνώσεις ηλεκτρονικού υπολογιστή, άνθρωποι κάθε ηλικίας μπορούν να κάνουν search στο google και να αναζητήσουν αυτό που επιθυμεί ο καθένας.

Αφού λοιπόν το Internet βοηθάει στην πιο γρήγορη εξυπηρέτηση των ανθρώπων που το χρησιμοποιούν, έτσι και εμείς θα δημιουργήσουμε κάτι το οποίο θα εξυπηρετεί γρήγορα και απλά τους ανθρώπους που θέλουν να ταξιδέψουν με πλοίο προς διάφορους προορισμούς. Αυτό θα είναι ένας γρήγορος τρόπος κράτησης ηλεκτρονικού ακτοπλοϊκού εισιτηρίου. Χωρίς άσκοπες μετακινήσεις σε τουριστικά πρακτορεία, αποφυγή κυκλοφοριακού συνωστισμού και πολύ γρήγορα, θα μπορεί κάποιος να κλείσει το ακτοπλοϊκό εισιτήριο που επιθυμεί.

3.1 XAMPP



Εικόνα 2: Λογότυπο του Xampp

3.1.1 Τι είναι το Xampp

Το Xampp είναι ένα ελεύθερο εργαλείο ανάπτυξης και δοκιμής ιστοσελίδων τοπικά στον υπολογιστή χωρίς να είναι απαραίτητη η σύνδεση στο διαδίκτυο. Περιλαμβάνει μία ελαφριά διανομή Apache, που το καθιστά εξαιρετικά εύκολο για τους προγραμματιστές να δημιουργήσουν έναν τοπικό server για τους σκοπούς της δοκιμής τους. Το ακρωνύμιό του είναι Cross-Platform (X), Apache (A), MySQL (M), PHP (P), Perl (P). Τα παραπάνω λογισμικά είναι ότι χρειάζεται κάποιος για να δημιουργήσει ένας web server. Δεδομένου ότι οι περισσότερες πραγματικές web server υλοποιήσεις, χρησιμοποιούν τα ίδια components με το XAMPP, κάνει τη μετάβαση από έναν τοπικό server δοκιμών σε ένα πραγματικό server να είναι εξαιρετικά εύκολη. Η ανάπτυξη μίας διαδικτυακής εφαρμογής, χρησιμοποιώντας το Xampp είναι εξαιρετικά εύκολη ακόμα και για αρχάριους χρήστες καθώς περιλαμβάνει τις PHP και MySQL, οι οποίες είναι εξίσου εύκολες να διδαχθούν.

3.1.2 Τι περιλαμβάνει



Εικόνα 3: Εργαλεία του Xampp

Το Xampp περιλαμβάνει 4 κύρια components:

1. Apache: Είναι μία πραγματική web server εφαρμογή που επεξεργάζεται και παρέχει περιεχόμενο web σε ένα υπολογιστή. Είναι ο πλέον δημοφιλής web server στο Internet και περιλαμβάνεται σχεδόν στο 50% όλων των ιστοσελίδων.

2. MySQL: Κάθε διαδικτυακή εφαρμογή, είτε απλή είτε πολύπλοκη, απαιτεί μια βάση δεδομένων για την αποθήκευση των δεδομένων. Η MySQL, η οποία είναι ανοικτού κώδικα, είναι το πιο δημοφιλές σύστημα διαχείρισης βάσεων δεδομένων. Κινεί τα πάντα, από προσωπικές ιστοσελίδες απλής χρήσης, μέχρι επαγγελματικές πλατφόρμες όπως η WordPress.

3. PHP: Είναι μια server-side scripting γλώσσα, η οποία χρησιμοποιείται σε κάποιες από τις πιο δημοφιλείς ιστοσελίδες στον κόσμο, συμπεριλαμβανομένων τη WordPress και το Facebook. Είναι open source, εύκολη στην εκμάθηση και λειτουργεί άψογα με τη MySQL, καθιστώντας την δημοφιλή επιλογή για τους web developers.

4. Perl: Perl είναι μία υψηλού επιπέδου, δυναμική γλώσσα προγραμματισμού που χρησιμοποιείται ευρέως στον προγραμματισμό διαδικτύου, system admin κ.α. Αν και λιγότερο δημοφιλής για ανάπτυξη web εφαρμογών, η Perl έχει εισέλθει σε ένα κομμάτι της αγοράς για δημιουργία εφαρμογών.

Διάφορες εκδόσεις του XAMPP μπορεί να έχουν πρόσθετα components για τη δημιουργία ενός πλήρη διακομιστή web, όπως το OpenSSL και το phpMyAdmin, το οποίο χρησιμοποιούμε για την υλοποίηση της πτυχιακής. Επίσης, μπορούμε να χειριστούμε τον localhost σαν έναν απομακρυσμένο υπολογιστή, χρησιμοποιώντας σύνδεση μέσω ενός FTP client. Χρησιμοποιώντας ένα πρόγραμμα όπως το Filezilla, έχουμε πολλά πλεονεκτήματα όταν εγκαθιστάτε ένα σύστημα διαχείρισης περιεχομένου(CMS) όπως το Joomla ή Wordpress. Είναι επίσης δυνατή η σύνδεση του localhost μέσω FTP με έναν HTML editor. Η προεπιλογή χρήστη FTP είναι "newuser", ο προεπιλεγμένος κωδικός πρόσβασης FTP είναι "xampp". Η προεπιλογή χρήστη MySQL είναι "root" ενώ δεν έχει password.

Βασικές διαφορές που το καθιστούν λειτουργικότερο από τους ανταγωνιστές του είναι το γεγονός ότι μπορεί να δουλέψει εξίσου καλά σε όλα τα λειτουργικά συστήματα, συμπεριλαμβανομένων των Windows, Linux, Unix, Mac. Το interface του είναι πιο απλό προς το χρήστη και κάνει εύκολη τη λειτουργία του.

3.2 PHP



Εικόνα 4: Λογότυπο της Php

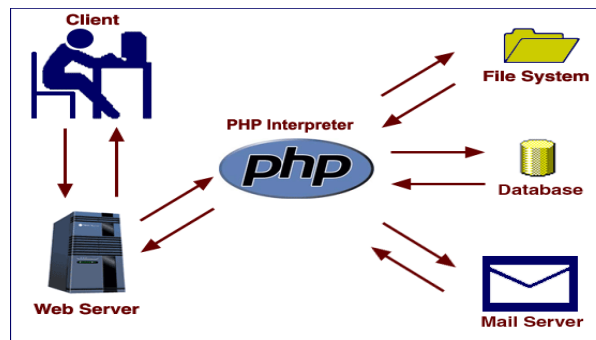
3.2.1 Τι είναι η PHP

Η PHP είναι πιθανός η πιο δημοφιλής scripting γλώσσα προγραμματισμού για το διαδίκτυο. Είναι αντικειμενοστραφής και χρησιμοποιείται για να ενισχύει τις ιστοσελίδες. Με την PHP μπορούμε να κάνουμε διάφορα πράγματα, όπως να δημιουργήσουμε σελίδες με login που χρειάζονται username και password, να ελέγξουμε πληροφορίες για μια φόρμα, δημιουργία forums, γκαλερί φωτογραφιών κ.α. Μια ιστοσελίδα γραμμένη σε PHP περνά από επεξεργασία από ένα διακομιστή π.χ. Apache, ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο που θα σταλεί στον τελικό χρήστη. Ένα αρχείο αποτελούμενο από κώδικα PHP θα πρέπει να έχει κατάληξη .php, .php4, .phtml κ.α. Ανταγωνιστικές προς αυτή γλώσσες προγραμματισμού είναι η ASP της Microsoft, η CFML της Allaire και η JSP της Sun. Το ακρωνύμιο της προέρχεται από το Hypertext Pre-processor. Αλλά με βάση αυτό θα έπρεπε να γράφεται HPP. Μία εναλλακτική εξήγηση είναι ότι τα αρχικά της προέρχονται από την παλαιότερη έκδοση του προγράμματος, η οποία ονομαζόταν Personal Home Page Tools.

3.2.2 Ιστορική Αναδρομή

Η ιδέα για τη δημιουργία της PHP ήρθε από τον Rasmus Lerdorf. Οι πρώτες versions της χρησιμοποιήθηκαν για να μπορεί κάποιος να ελέγχει αυτούς που μπαίνουν σε μία σελίδα. Η πρώτη έκδοση είχε το όνομα Personal Home Page Tools και δόθηκε για χρήση στο κοινό στις αρχές του 1995. Αποτελούνταν από μία απλοϊκή μηχανή αναζήτησης, η οποία καταλάβαινε ειδικές μακροεντολές και utilities, ένα guestbook, ένα μετρητή και κάποιο ακόμα υλικό. Ο αναλυτής ξαναγράφηκε το 1995 και πήρε την ονομασία PHP/F1 version 2. Το όνομα F1 το πήρε από ένα άλλο πακέτο που έγραψε ο Rasmus, το οποίο διερμήνευε τα δεδομένα από της φόρμες της HTML. Στη συνέχεια συνδύασε τα εργαλεία scripts της PHP με τον File Interpreter και πρόσθεσε υποστήριξη για mSql. Η PHP/F1 αναπτύχθηκε ραγδαία, καθώς πολλοί χρήστες άρχισαν να προσφέρουν κώδικα σε αυτή. Έως τα τέλη του 1996, η PHP/F1 χρησιμοποιούταν από 15.000 ιστοσελίδες παγκοσμίως, ενώ μέσα στο 1997 ο αριθμός αυτός είχε φτάσει τις 50.000. Την ίδια χρονιά, οι Andi Gutmans και Zeev Suraski ξαναέγραψαν τη γλώσσα από την αρχή, βασιζόμενοι στην C. Έτσι έφτασε η δημιουργία της PHP 3.0. Κατόπιν, οι δύο αυτοί άντρες έφτιαξαν την εταιρία Zend (από τα αρχικά των ονομάτων τους), η οποία συνεχίζει μέχρι σήμερα την ανάπτυξη και εξέλιξη της γλώσσας αυτής. Το 1998 ήρθε η ώρα της PHP 4.0 και τον Ιούλιο του 2004 διατέθηκε η PHP 5.0 ενώ αυτή τη στιγμή έχουν διατεθεί τα πρώτα snapshots της επερχόμενης PHP 6.0.

3.2.3 Πλεονεκτήματα & Μειονεκτήματα της PHP



Εικόνα 5: Server-side language

Σήμερα περισσότερες από 16.000.000 ιστοσελίδες, ποσοστό μεγαλύτερο από το 35% των ιστοσελίδων του διαδικτύου, χρησιμοποιούν scripts γραμμένα με PHP, ενώ το υπόλοιπο 65% είναι γραμμένο με διάφορες στατικές γλώσσες προγραμματισμού. Η μεγάλη στροφή προς την PHP έγινε επειδή έχει σημαντικά πλεονεκτήματα απέναντι στην HTML. Μερικά από τα πλεονεκτήματά της παρουσιάζονται παρακάτω,

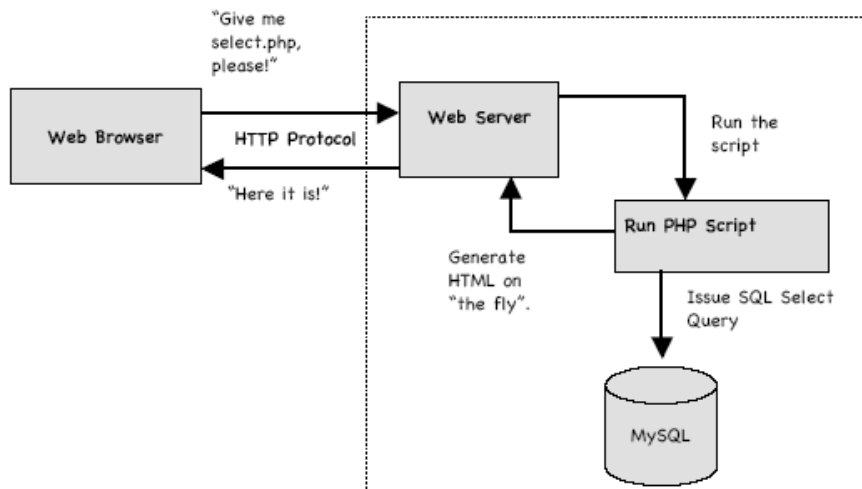
- Λειτουργεί σε περιβάλλον Windows, Linux, Mac, Risk κ.α.
- Η PHP είναι δυναμική, που σημαίνει ότι τα δεδομένα της ιστοσελίδας μπορούν να ανανεώνονται και να αλλάζουν με την παρέμβαση του χρήστη.
- Είναι ασφαλής για ιστοσελίδες που απαιτούν ασφάλεια στα δεδομένα όπως e-banking, air-banking, bind shopping, επεξεργασία βάσεων δεδομένων.
- Μπορεί να χρησιμοποιηθεί και για command line scripting (PHP CLI)
- Εύκολη στη εκμάθηση και τη χρήση της
- Είναι ανοιχτού κώδικα
- Παρέχει υποστήριξη για συνομιλία με άλλες υπηρεσίες χρησιμοποιώντας πρωτόκολλα HTTP, IMAP, POP3, SMTP, NNTP
- Παρέχει υποστήριξη σε μία ευρεία γκάμα από βάσεις δεδομένων, όπως MySql, Oracle, PostgreSQL, Solid, Sybase, Velocis, Unixdbm.
- Μπορούμε να δουλέψουμε είτε με αντικειμενοστραφή προγραμματισμό είτε όχι
- Συνεχής εξέλιξη και έχει πολλές έτοιμες συναρτήσεις
- Ίσως το σημαντικότερο πλεονέκτημά της είναι ότι είναι server-side, δηλαδή τρέχει στον server χωρίς ο χρήστης να βλέπει το αποτέλεσμα

3.2.4 Μειονεκτήματα

Αν και τα πλεονεκτήματά της υπερिशύουν, μερικά μειονεκτήματα που έχει είναι:

- Θέματα ασφαλείας, οποιοσδήποτε μπορεί να δει τον πηγαίο κώδικα
- Δεν είναι κατάλληλη για μεγάλες εφαρμογές διότι δεν είναι modular
- Ασθενές τύπος, μία ανεπιφύλακτη μετατροπή μπορεί να εκπλήξει τους ανυποψίαστους προγραμματιστές. Για παράδειγμα, οι συμβολοσειρές "1000" και "1e3" συγκρίνονται ως ίσες, διότι πετάει χωρίς προειδοποίηση του αριθμούς κινητής υποδιαστολής

3.2.5 Σύνταξη της PHP



Εικόνα 6: Προσπέλαση PHP σελίδων

Η PHP χρησιμοποιεί τα σύμβολα `<?php` και `?>` για να εισάγει ένα block script μέσα σε ένα έγγραφο html ή xhtml. Μπορούμε να χρησιμοποιήσουμε επίσης τα σύμβολα `<?>` για συντομία ή ακόμα και τα σύμβολα της ASP `<%>` και `%>`, αλλά δεν συνιστάτε διότι δεν υποστηρίζονται από την PHP 6. Για λόγους συμβατότητας χρησιμοποιούμε μόνο τον πρώτο συμβολισμό.

```
<?php
?>
```

- Εντολή echo

```
<?php
echo("Καλημέρα");
?>

<br />

<?php
echo("Καλησπέρα");
?>
```

Το παραπάνω παράδειγμα θα τυπώσει

```
Καλημέρα
Καλησπέρα
```

Πίνακας 2: Παράδειγμα Php

Κάθε εντολή τερματίζεται με το ελληνικό ερωτηματικό. Η PHP είναι γλώσσας case sensitive, το οποίο σημαίνει ότι γίνεται διάκριση ανάμεσα στα πεζά και στα κεφαλαία γράμματα. Όταν θέλουμε να συμπεριλάβουμε σχόλια, χρησιμοποιούμε το `//` για σχόλια σε μία γραμμή και το `/*` και `*/`, όταν βάζουμε σχόλια σε πολλές γραμμές.

3.2.6 Μεταβλητές

Βασικός κανόνας στην PHP είναι ότι στα ονόματα των μεταβλητών πρέπει να προηγείται ο χαρακτήρας \$. Έτσι, η PHP μπορεί να αναγνωρίσει τις μεταβλητές ανάμεσα στα κείμενα. Το όνομα κάθε μεταβλητής πρέπει να αρχίζει από γράμμα ή την κάτω παύλα “_”. Ανάλογα με την προσβασιμότητα που θέλουμε να έχουν οι μεταβλητές μας χωρίζονται σε local, global, static.

```
<?php
$text="Καλησπέρα";
print $text;
?>
```

Η εκτέλεση του παραπάνω κώδικα θα μας δώσει το ακόλουθο αποτέλεσμα

Καλησπέρα

Πίνακας: Παράδειγμα-2 PHP

3.2.7 Cookies

Τα cookies είναι μικρά αρχεία που αποθηκεύονται στο μηχάνημά του client και περιέχει δεδομένα σχετικά με την ταυτότητα και το προφίλ του χρήστη. Η PHP έχει τη δυνατότητα να δημιουργεί τέτοια αρχεία στα μηχανήματα του χρήστη σχετικά με τον ιστότοπο.

Για να δημιουργήσουμε ένα τέτοιο αρχείο, η εντολή που θα μας το επιτρέψει είναι η setcookie() και δέχεται μέχρι 6 παραμέτρους

Παράδειγμα,

```
setcookie(name, value, expire, path, domain, secure);
<?php
setcookie("visitor","obama","expire=time()+3600");
?>
```

Αυτό σημαίνει ότι name=visitor , value=obama και το cookie θα ζήσει για 3600 δευτερόλεπτα από τη στιγμή που θα δημιουργηθεί

3.2.8 Sessions

Είναι ένας μηχανισμός με τον οποίο μπορούμε να ελέγξουμε την ταυτότητα του επισκέπτη που επισκέπτεται την ιστοσελίδα. Με την PHP μπορούμε να δημιουργήσουμε sessions, με τα οποία ελέγχουμε την ταυτότητα των χρηστών της ιστοσελίδας και κατά συνέπεια την πρόσβαση που έχουν σε ευαίσθητες σελίδες. Χρησιμοποιούν την τεχνολογία των cookies. Κάθε επισκέπτης δέχεται ένα cookie που περιέχει έναν αριθμό (id), το οποίο είναι το αναγνωριστικό του. Με αυτόν τον τρόπο μπορούμε να ελέγξουμε εάν ο συγκεκριμένος χρήστης είναι σε session ή όχι. Για να μπορέσουμε να διαβάσουμε μία τιμή αναγνωριστικού χρησιμοποιούμε την εντολή session_id().

Ένα session ξεκινά με τη συνάρτηση session_start(). Οι μεταβλητές session είναι global και δηλώνονται με τον ακόλουθο τρόπο \$_SESSION

Παράδειγμα :

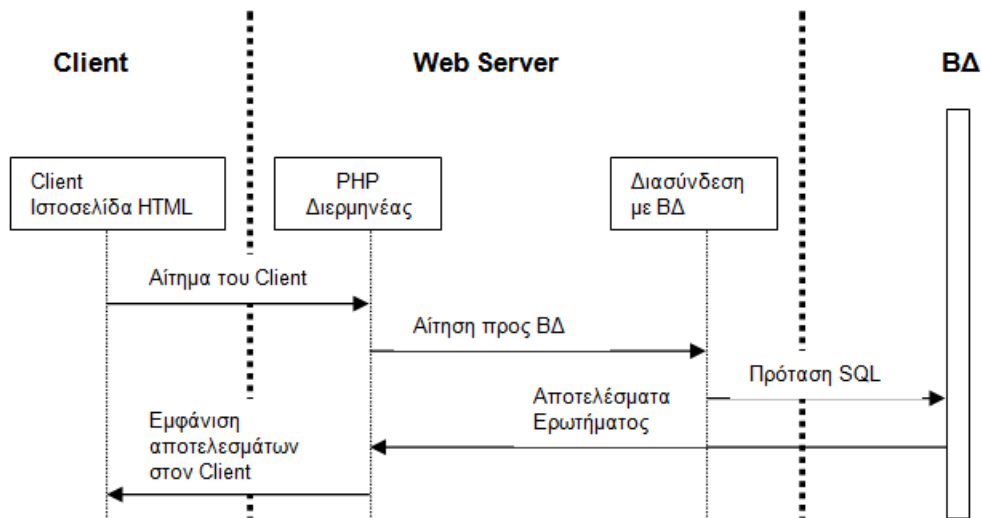
```
<?php
session_start();
$_SESSION['visitor']=session_id();
?>
```

Η εντολή με την οποία καταστρέφουμε ένα session είναι η `unset()`; και ο τρόπος που χρησιμοποιείται είναι `unset($_SESSION['visitor'])`; . Εναλλακτικός τρόπος διαγραφής ενός session είναι η εντολή `session_destroy()`; . Με την τελευταία, διαγράφονται όλα τα δεδομένα.

3.2.9 PHP-MySql

Ο συνδυασμός της γλώσσας PHP και της βάσης δεδομένων MySql έχει σαν βασικό πλεονέκτημα ότι υποστηρίζεται από σχεδόν όλες τις πλατφόρμες. Έχουν κοινά χαρακτηριστικά όπως είναι το γεγονός ότι ανήκουν και οι δύο στις εφαρμογές ανοικτού κώδικα και τα δικαιώματα χρήσης τους είναι δωρεάν. Εξαιτίας των κοινών αυτών χαρακτηριστικών, έχουν αναπτυχθεί web servers που υποστηρίζουν τα δύο αυτά λογισμικά και την άμεση συνδεσιμότητα μεταξύ τους.

Η βασική λειτουργία του συνδυασμού των δύο τεχνολογιών είναι η εξής: Ένας δυναμικός δικτυακός τόπος αποτελείται από PHP σελίδες. Η λειτουργικότητα που παρέχουν οι σελίδες αυτές στο χρήστη στηρίζεται στον εκτελέσιμο κώδικα που είναι ενσωματωμένος. Οι δυναμικές σελίδες PHP περιέχουν κώδικα ο οποίος εκτελείται στον server. Ο κώδικας αυτός εκτελεί ερωτήματα σε SQL τα οποία μεταβιβάζονται μέσω του ειδικού driver της MySql στη βάση MySql. Ανάλογα με την αίτηση του χρήστη μπορεί να γίνει μια καταχώριση, τροποποίηση ή διαγραφή δεδομένων στη Βάση. Επίσης ο χρήστης μπορεί να αιτηθεί την ανάκτηση κάποιας πληροφορίας. Η αίτηση μεταβιβάζεται στη Βάση Δεδομένων και τα αποτελέσματα επιστρέφουν στο web server. Στη συνέχεια τα δεδομένα χρησιμοποιούνται στη δημιουργία της σελίδας που τελικά αποστέλλεται στο χρήστη και του προβάλλει το περιεχόμενο που ζήτησε. Το περιεχόμενο παρουσιάζεται στο χρήστη από τον αντίστοιχο browser.



Εικόνα 7: Ακολουθιακό Διάγραμμα

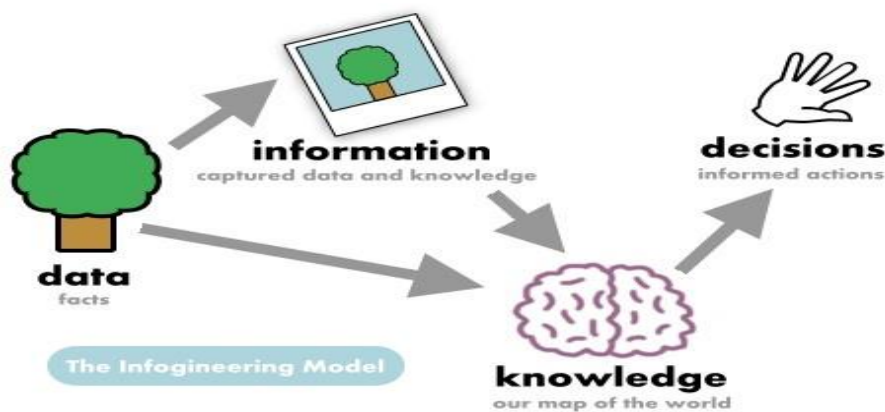
3.3 ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

Η ραγδαία ανάπτυξη της πληροφορικής τα τελευταία χρόνια έχει κάνει την πληροφορία να είναι ένα βασικό και αν μη τι άλλο, μη προβλέψιμο γεγονός για την ανάγκη της. Η πληροφορία έχει ωφελήσει όλους τους εργαζομένους αλλά και τους εκπαιδευόμενους ώστε να είναι πιο αποδοτικοί και παραγωγικοί άνθρωποι.

3.3.1 Οι Πληροφορίες και τα Δεδομένα

Ο όρος πληροφορία αναφέρεται σε γεγονότα, έννοιες που αποκτά κάποιος από την καθημερινή του επικοινωνία και θεωρείτε αποκτηθείσα γνώση, ενώ τα δεδομένα δεν είναι πολλές φορές δομημένα ή κατάλληλα επεξεργασμένα. Ο αυστηρός ορισμός από την επιτροπή ANSI των ΗΠΑ, για το τι είναι πληροφορία και δεδομένα είναι ο παρακάτω:

- *Πληροφορία* είναι η σημασία που δίνουμε σε ένα σύνολο από δεδομένα, τα οποία μπορούμε να επεξεργαστούμε βάσει προκαθορισμένων κανόνων και να βγάλουμε χρήσιμα συμπεράσματα. Με τις πληροφορίες περιορίζεται η αβεβαιότητα που έχουμε για διάφορα πράγματα και μας βοηθάνε για να λάβουμε σωστές αποφάσεις.
- *Δεδομένα* είναι μία παράσταση όπως γράμματα, αριθμοί, σύμβολα κ.α. στα οποία μπορούμε να δώσουμε κάποια έννοια.



Εικόνα 8: Πληροφορίες και Δεδομένα

Η παραπάνω εικόνα μας δείχνει τον τρόπο με τον οποίο λαμβάνονται αποφάσεις. Για να μπορέσει να γίνει αυτό, αρχικά θα πρέπει τα δεδομένα να διαχειριστούν και να επεξεργαστούν και να οργανωθούν κατάλληλα. Ένα παράδειγμα σωστής οργάνωσης δεδομένων είναι οι τηλεφωνικοί κατάλογοι των πόλεων. Τα ονόματα των πελατών είναι κατοχυρωμένα αλφαβητικά σύμφωνα με το επίθετο του κάθε συνδρομητή. Φανταστείτε ένα τηλεφωνικό κατάλογο του Ηρακλείου, ο οποίος θα περιείχε όλα τα τηλέφωνα των κατοίκων του νομού αλλά θα ήταν με τυχαία σειρά. Θα είχαμε τόσα πολλά σημαντικά δεδομένα αλλά θα μας ήταν εντελώς άχρηστος αφού θα χρειαζόμασταν πολλές ώρες ή και ημέρες για να μπορέσουμε να βρούμε τον τηλεφωνικό αριθμό που θέλουμε. Ένα άλλο παράδειγμα μη σωστής οργάνωσης δεδομένων είναι μία μεγάλη βιβλιοθήκη με χιλιάδες τόμους βιβλίων, η οποία δε θα είχε κάποιο σύστημα ταξινόμησης των βιβλίων της. Θα επικρατούσε ένα χάος στα ράφια και θα ήταν πραγματικά αδύνατο να μπορέσουμε να βρούμε το βιβλίο που θέλουμε να διαβάσουμε.

Χαρακτηριστικά παραδείγματα δεδομένων που απαιτούν σωστή οργάνωση και διαχείριση είναι:

- Τα στοιχεία πελατών, υπαλλήλων, προμηθευτών μιας εμπορικής επιχείρησης
- Τα στοιχεία ασθενών, γιατρών, υπαλλήλων και υλικών ενός νοσοκομείου
- Τα στοιχεία φοιτητών, καθηγητών, διοικητικού προσωπικού ενός πανεπιστημίου
- Τα στοιχεία υλικών μίας αποθήκης

3.3.2 Οργάνωση Αρχείων

Ο πιο διαδεδομένος τρόπος οργάνωσης δεδομένων είναι σε αρχεία εγγράφων. Για να γίνει κατανοητό, θα έχουμε σαν παράδειγμα αυτό μιας εμπορικής επιχείρησης. Θα πρέπει να δημιουργήσουμε καρτέλες για τους πελάτες με τα εξής στοιχεία:

- Κωδικός
- Επίθετο
- Όνομα Διεύθυνση
- ΤΚ
- Πόλη
- Τηλέφωνο
- ΑΦΜ

Το σύνολο των παραπάνω πεδίων, θα το σώσουμε και θα ονομάζεται Αρχείο. Η οργάνωση των αρχείων μπορεί να επιφέρουν τα εξής προβλήματα:

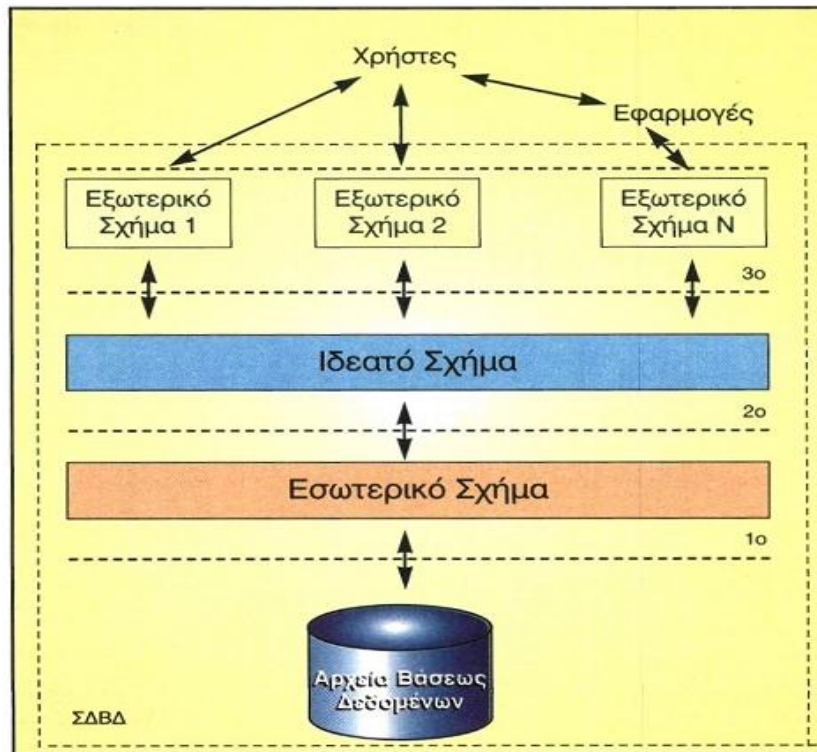
- Πλεονασμός δεδομένων: Επανάληψη των ίδιων δεδομένων σε αρχεία διαφορετικών εφαρμογών.
- Ασυνέπεια δεδομένων: Ίδια στοιχεία πελατών και στο αρχείο πελατών και στο αρχείο παραγγελιών. Όταν γίνει αλλαγή στοιχείων ενός πεδίου σε ένα από τα δύο αρχεία, δε θα φέρει αλλαγή στο άλλο.
- Αδυναμία μερισμού δεδομένων: Δημιουργεί καθυστέρηση στη λήψη αποφάσεων και στην εξυπηρέτηση χρηστών.
- Αδυναμία προτυποποίησης: Ανομοιομορφία και με διαφορετική οργάνωση δεδομένων στα αρχεία. Επιφέρει προβλήματα στη προσαρμογή των χρηστών καθώς και στην ανταλλαγή δεδομένων διαφορετικών συστημάτων.

3.3.3 Τι είναι Βάση Δεδομένων και Ιστορικά Στοιχεία

Βάση Δεδομένων είναι ένα σύνολο αρχείων με υψηλό βαθμό οργάνωσης τα οποία είναι συνδεδεμένα μεταξύ τους με λογικές σχέσεις, ώστε να μπορούν να τις χρησιμοποιούν πολλοί χρήστες και πολλές εφαρμογές την ίδια στιγμή.

Η ιστορία των Βάσεων Δεδομένων ξεκίνησε στα τέλη της δεκαετίας του 1960 και αφορμή για αυτό ήταν τα τραπεζικά δεδομένα. Ονομαζόντουσαν βάσεις δεδομένων CODASYL, από το όνομα της επιτροπής που της τυποποίησε. Εκείνη την εποχή πρωτοεμφανίστηκε το IMS που ήταν το πιο γνωστό σύστημα διαχείρισης ιεραρχικών βάσεων δεδομένων (hierarchical DBMS). Το IMS είναι προϊόν της εταιρίας IBM, το οποίο ακόμα και σήμερα έχει τις περισσότερες εγκαταστάσεις. Αυτά τα δύο μοντέλα χαρακτηρίζονται από το γεγονός ότι η διαχείριση των δεδομένων γίνεται σε χαμηλό επίπεδο με αποτέλεσμα οι εφαρμογές να σχετίζονται άμεσα από την τοποθεσία που έχουν αποθηκευτεί τα δεδομένα και ότι τα δεδομένα αυτά τα διαχειριζόμαστε ατομικά. Το 1970, ο E. F. Codd δημοσιεύει ένα άρθρο με τίτλο "A Relational Model of Data for Large Shared Data Banks", το οποίο έκανε την επανάσταση στο χώρο των βάσεων δεδομένων. Στα μέσα της δεκαετίας του 1980, όπου η Oracle είχε τα πρωτεία στις βάσεις δεδομένων, δημιουργήθηκε το πρώτο standard για μία γλώσσα διαχείρισης και επερώτησης σχεσιακών βάσεων δεδομένων, την Structured Query Language-SQL.

3.3.4 Συστήματα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ-DBMS)



Εικόνα 9: Η αρχιτεκτονική ANSI ενός ΣΔΒΔ

Είναι ένα λογισμικό που μεσολαβεί ανάμεσα στα αρχεία δεδομένων και στις εφαρμογές που χρησιμοποιούνται από τους χρήστες. Είναι δηλαδή ένα πρόγραμμα που διαχειρίζεται βάσεις δεδομένων και με αυτό μπορούμε να τροποποιήσουμε, αποθηκεύσουμε, προσθέσουμε, αφαιρέσουμε ή διαγράψουμε τα δεδομένα. Τα δεδομένα αυτά πρέπει να είναι:

- Ολοκληρωμένα (Integrated), δηλαδή να είναι αποθηκευμένα και οργανωμένα σε αρχεία όπου δεν υπάρχει κάποια επανάληψη ή πλεονασμός ίδιων στοιχείων
- Καταμερισμένα (Shared), δηλαδή να μπορούν πολλοί χρήστες μαζί να έχουν τα ίδια δεδομένα, την ίδια χρονική στιγμή.

Από τη μεριά του ηλεκτρονικού υπολογιστή έχουμε:

- Το λογισμικό (software), που είναι τα προγράμματα και τα εργαλεία με τα οποία επεξεργαζόμαστε τα δεδομένα
- Το υλικό (hardware), που είναι τα περιφερειακά μέρη του υπολογιστή όπως σκληροί δίσκοι, dvd, cloud και εκεί αποθηκεύονται τα δεδομένα μας
- Προγραμματιστές, είναι αυτοί που αναπτύσσουν τις εφαρμογές του ΣΒΔ σε κάποια από τις γνωστές γλώσσες προγραμματισμού
- Διαχειριστής δεδομένων, είναι αυτός που έχει τη διοικητική αρμοδιότητα και ευθύνη για την οργάνωση της βάσης δεδομένων και την απόδοση δικαιωμάτων πρόσβασης στους χρήστες
- Διαχειριστής βάσης δεδομένων, είναι αυτός που λαμβάνει οδηγίες από τον διαχειριστή δεδομένων και διαθέτει τις τεχνικές γνώσεις και αρμοδιότητες για τη σωστή και αποδοτική λειτουργία του ΣΔΒΔ
- Τελικοί χρήστες, είναι αυτοί που χρησιμοποιούν κάποια εφαρμογή για να παίρνουν στοιχεία από μια βάση δεδομένων, έχουν τις λιγότερες δυνατότητες επέμβασης στα στοιχεία της βάσης δεδομένων, χρησιμοποιούν ειδικούς κωδικούς πρόσβασης και το σύστημα τούς επιτρέπει ανάλογα πρόσβαση σε συγκεκριμένο κομμάτι της βάσης δεδομένων

Για να μπορέσουμε να πούμε ότι ένα Σύστημα Διαχείρισης Βάσεων Δεδομένων είναι αποδοτικό θα πρέπει να πληροί τις ακόλουθες προδιαγραφές,

- Να έχει ευκολία στην ανάπτυξη νέων εφαρμογών
- Να υποστηρίζεται από μηχανισμούς ασφαλείας
- Να έχει μείωση πλεοναζόντων δεδομένων
- Να υπάρχει υποστήριξη δοσοληψιών
- Να υπάρχει ανεξαρτησία των δεδομένων από τις εφαρμογές τόσο στο φυσικό επίπεδο όσο και στις παρεχόμενες όψεις

Η χρήση των ΣΔΒΔ πλεονεκτεί σε σχέση με την αποθήκευση σε απλά αρχεία:

- Στην ασφάλεια των δεδομένων
- Στην ταυτόχρονη πρόσβαση από πολλούς χρήστες ταυτόχρονα
- Στη διασφάλιση ακεραιότητας των δεδομένων
- Στη γρήγορη πρόσβαση στα δεδομένα
- Στον τρόπο αποθήκευσης των δεδομένων σε διαφορετικούς υπολογιστές
- Στην καταγραφή στοιχείων χρήσης
- Στη οργάνωση αποθήκευσης δεδομένων π.χ. χρήση δομών δέντρων
- Είναι εφοδιασμένα με μηχανισμούς ανάκτησης δεδομένων

3.3.5 Διαχειριστής Βάσεων Δεδομένων

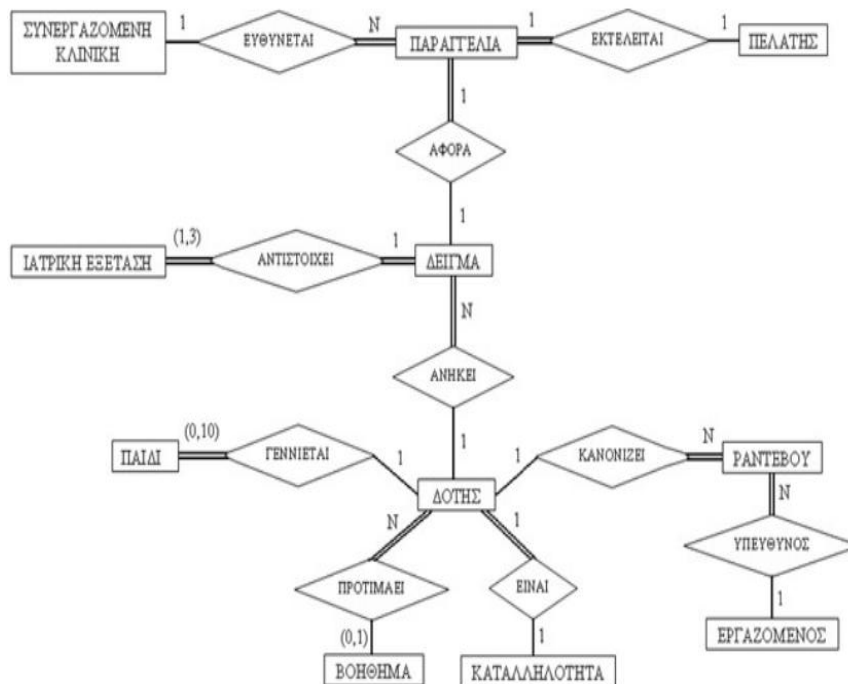
Είναι αυτός που πρέπει να εξετάσει την αποδοτικότητα, την αξιοπιστία, τη σωστή λειτουργία της βάσης δεδομένων και έχει τις εξής αρμοδιότητες,

- Πρέπει να καθορίσει την εννοιολογική σχεδίαση της βάσης δεδομένων εντοπίζοντας τις σημαντικές οντότητες και τις σχετικές με αυτές πληροφορίες που θα τηρούνται. Δημιουργεί το αντίστοιχο εννοιολογικό σχήμα, χρησιμοποιώντας την εννοιολογική DDL (γλώσσα ορισμού δεδομένων).
- Αποφασίζει για τη σχεδίαση της βάσης δεδομένων και δημιουργεί τον ορισμό αποθηκευτικής δομής, χρησιμοποιώντας την εσωτερική DDL. Πρέπει επίσης να ορίσει τη σχετική απεικόνιση μεταξύ του εσωτερικού και του εννοιολογικού σχήματος. Οι αντίστοιχες απεικονίσεις θα υπάρχουν και σε μορφή πηγαίου κώδικα και σε μορφή αντικειμένου κώδικα.
- Εξασφαλίζει ότι τα δεδομένα που χρειάζονται είναι διαθέσιμα. Ζωγραφίζει τα απαραίτητα εξωτερικά σχήματα, χρησιμοποιώντας την κατάλληλη DDL ώστε να οριστεί η απεικόνιση ανάμεσα σε οποιοδήποτε δεδομένο εξωτερικό σχήμα και στο εννοιολογικό σχήμα. Συγχρόνως παρέχει συμβουλές για την σχεδίαση των εφαρμογών, τεχνική εκπαίδευση, εντόπιση και επίλυση προβλημάτων.
- Ορίζει τους κανόνες ασφαλείας και ακεραιότητας μπορούν να θεωρηθούν μέρος του εννοιολογικού σχήματος.
- Ορίζει τις διαδικασίες για τη λήψη εφεδρικών αντιγράφων και την ανάκαμψη.

3.3.6 Μοντέλα Δεδομένων

Μοντέλο δεδομένων εννοούμε ένα σύνολο από έννοιες που περιγράφουν τον τρόπο με τον οποίο αυτά αναπαρίστανται, προσπελαύνονται, αποθηκεύονται. Κατηγοριοποιούνται με βάση τη μορφή των εννοιών που έχουν για την περιγραφή των δεδομένων και αυτά είναι τα *εννοιολογικά* μοντέλα, τα *φυσικά* μοντέλα και τα *λογικά* μοντέλα. Ένα παράδειγμα εννοιολογικού μοντέλου είναι το μοντέλο *οντοτήτων-συσχετίσεων*, το οποίο θα το παρουσιάσουμε παρακάτω ενώ κάποια λογικά μοντέλα (μοντέλα υλοποίησης) είναι το *Ιεραρχικό μοντέλο*, το *Δικτυωτό μοντέλο*, το *Σχεσιακό μοντέλο*.

3.3.7 Μοντέλο Οντοτήτων-Συσχετίσεων



Εικόνα 10: Διάγραμμα οντοτήτων συσχετίσεων

Το μοντέλο οντοτήτων-συσχετίσεων αναπτύχθηκε από τον P.Chen το 1976 ως ένας τρόπος επικοινωνίας μεταξύ χρηστών και σχεδιαστών. Βασίζεται στην αντίληψη ότι ο πραγματικός κόσμος αποτελείται από οντότητες (entities) με χαρακτηριστικά (attributes) και συσχετίσεις (relationships) μεταξύ τους. Σήμερα θεωρείται ένα από τα πιο σημαντικά εργαλεία κατασκευής εννοιολογικών μοντέλων βάσεων δεδομένων, σε περιβάλλοντα με ποικίλες απαιτήσεις, και αποτελεί βασική συνιστώσα μεθοδολογιών ανάπτυξης πληροφοριακών συστημάτων.

- **Οντότητα:** είναι κάθε κατάσταση, γεγονός, αντικείμενο ή έννοια που μπορούμε να διακρίνουμε ή να καταγράψουμε
- **Χαρακτηριστικά:** είναι τα συστατικά στοιχεία που περιγράφουν μία οντότητα
- **Συσχέτιση:** είναι μία σύνδεση μεταξύ των οντοτήτων, η οποία αναπαριστά μία αντίστοιχη σχέση των αντικειμένων στον πραγματικό κόσμο
- **Πληθικότητα:** είναι ο αριθμός των στιγμιότυπων μίας οντότητας που αντιστοιχίζεται με μία άλλη οντότητα άλλου τύπου
- **Κλειδί:** είναι το σύνολο των χαρακτηριστικών μίας οντότητας και ταυτοποιεί κάθε εγγραφή και τη διακρίνει από τις άλλες. Προσδιορίζει με μοναδικό τρόπο μία οντότητα

3.3.8 Σχεσιακό Μοντέλο

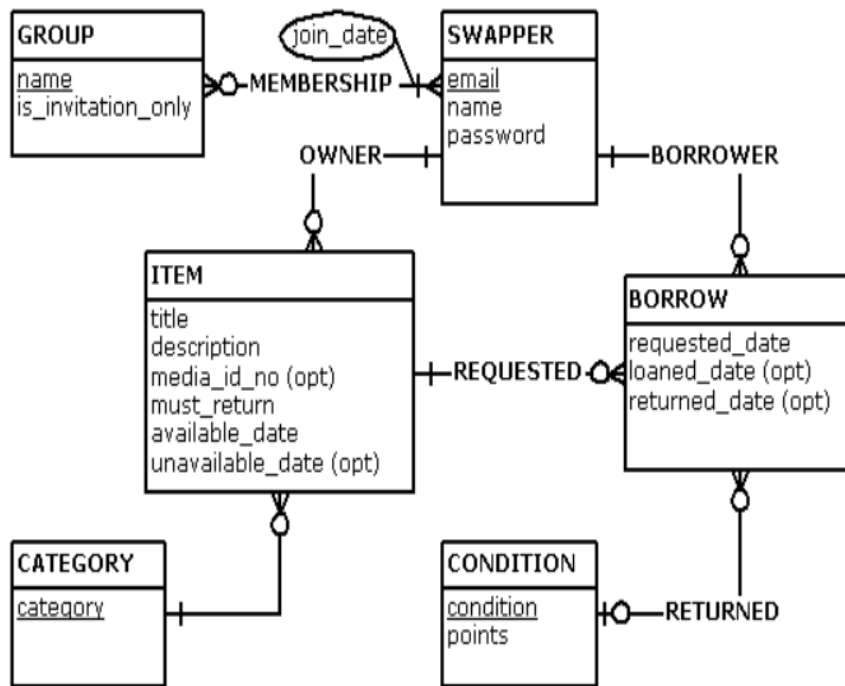
Το σχεσιακό μοντέλο (relational model) έχει επικρατήσει στην αναπαράσταση των δεδομένων καθώς διαθέτει σημαντικά πλεονεκτήματα ως προς το ιεραρχικό και το δικτυωτό και οι βάσεις δεδομένων που σχεδιάζονται σύμφωνα με αυτό αποκαλούνται σχεσιακές βάσεις δεδομένων (relational databases). Με τις σχεσιακές βάσεις δεδομένων έχουμε έναν σαφή, απλό και κατανοητό τρόπο για να αναπαραστήσουμε και να διαχειριστούμε τα δεδομένα μας. Μειονέκτημά τους είναι μόνο η ταχύτητα

υπολογισμών και ότι απαιτούν μεγάλο χώρο για αποθήκευση, αλλά μόνο όταν έχουμε να διαχειριστούμε πολύ μεγάλες βάσεις δεδομένων.

Οι βάσεις δεδομένων περιγράφονται με αυστηρές μαθηματικές έννοιες και ο χρήστης βλέπει τις οντότητες και τις συσχετίσεις με τη μορφή πινάκων (tables) και σχέσεων (relations) αντίστοιχα.

Ένας πίνακας (table) αποτελείται από γραμμές (rows) και στήλες (columns). Η κάθε στήλη του πίνακα χαρακτηρίζει κάποια ιδιότητα της οντότητας και αποκαλείται χαρακτηριστικό (attribute) ή πεδίο (field), ενώ η κάθε γραμμή του πίνακα περιέχει όλες τις πληροφορίες (στήλες) που αφορούν ένα στοιχείο της οντότητας και αποκαλείται πλειάδα (tuple) ή εγγραφή (record).

Κάθε πεδίο μπορεί να πάρει συγκεκριμένες τιμές, οι οποίες καθορίζονται από τον τύπο δεδομένων της ιδιότητας, όπως ονόματα ή αριθμοί ή και από αυτό που εκφράζει, όπως το ότι δεν μπορούμε να έχουμε αρνητικό βάρος ή αρνητικό ΑΦΜ. Το σύνολο των αποδεκτών τιμών μιας οντότητας αποκαλείται πεδίο ορισμού (domain).



Εικόνα 11: Διάγραμμα σχεσιακού μοντέλου

3.3.9 Ασφάλεια Βάσεων Δεδομένων

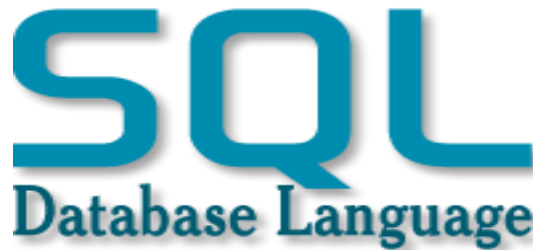
Η **ασφάλεια** (security) αναφέρεται στην προστασία δεδομένων από τη γνωστοποίηση, την αλλοίωση, ή την καταστροφή από μη εξουσιοδοτημένους χρήστες

Ακεραιότητα (integrity) σημαίνει να εξασφαλίζεται ότι αυτά που επιχειρούν να κάνουν οι χρήστες είναι σωστά και δεν θα πρέπει να μπορούν να τροποποιούν πράγματα που δεν πρέπει. π.χ., ένας φοιτητής δεν μπορεί να βάλει βαθμούς ή να τους τροποποιήσει

Μυστικότητα (secrecy) οι χρήστες θα πρέπει να μπορούν να δουν μόνο ότι πρέπει π.χ., ένας φοιτητής δεν θα πρέπει να μπορεί να δει τους βαθμούς άλλων φοιτητών

Διαθεσιμότητα (availability) οι χρήστες θα πρέπει να μπορούν να βλέπουν και να αλλάζουν, μόνο τα πράγματα που τους επιτρέπεται

3.4 SQL



Εικόνα 12: Λογότυπο της Sql

3.4.1 Ιστορία της SQL

Η δημιουργία της SQL (1969) βασίστηκε στην ιδέα του ερευνητή Ted Codd των εργαστηρίων της IBM, με αφορμή το σχεσιακό μοντέλο που εφαρμόστηκε από τον ίδιο. Αρχικά δημιουργήθηκε με σκοπό να παρέχει μία ημί-φυσική γλώσσα για το IBM System Relational database system. Το 1970, δύο επιστήμονες της IBM Almaden Donald D. Chamberlin και Raymond F. Boyce δημιούργησαν την πρώτη έκδοση της SQL η οποία ονομάστηκε SEQUEL. Το όνομα αυτό αλλάζει σε SQL λόγω του ότι το προηγούμενο ήταν κατοχυρωμένο από εταιρία κατασκευής αεροπλάνων. Το 1985, η IBM κάνει πατέντα την SQL. Οι κυριότερες διάλεκτοι είναι οι τυποποιημένες ANSI/ISO SQL. Πρόκειται για την σταθερή έκδοχή της γλώσσας όπως έχει οριστεί από δύο αναγνωρισμένους οργανισμούς. Υπήρξαν δυο πρότυπα: το SQL1 και το SQL2. Οι περισσότερες εφαρμογές προσπαθούν να μείνουν πιστές στην SQL2.

3.4.2 Τι είναι η SQL

Είναι μία γλώσσα για τις βάσεις δεδομένων και σχεδιάστηκε για τη διαχείριση δεδομένων σε συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων. Είναι μία σχετικά απλή γλώσσα και είναι υψηλού επιπέδου. Η Sql χρησιμοποιεί ειδικές γλώσσες προγραμματισμού, τις λεγόμενες γλώσσες ερωταπαντήσεων (query languages). Είναι γλώσσες μη διαδικαστικές, 4^{ης} γενιάς. Εμείς κάνουμε την ερώτηση και περιμένουμε την απάντηση.

Κάποια από τα πλεονεκτήματά της είναι:

- Η SQL είναι πολύ εύκολη στην εκμάθηση
- Η SQL αποτελεί μια στάνταρτ γλώσσα του ANSI (ANSI standard language)
- Η SQL μάς δίνει τη δυνατότητα να έχουμε πρόσβαση σε μια βάση δεδομένων
- Η SQL μπορεί να εκτελέσει ερωτήματα (queries) και να αναζητήσει πληροφορίες σε μια βάση δεδομένων
- Η SQL μπορεί να ανακτήσει δεδομένα από μια βάση δεδομένων
- Η SQL μπορεί να εισαγάγει νέες εγγραφές σε μια βάση δεδομένων
- Η SQL μπορεί να διαγράψει εγγραφές από μια βάση δεδομένων
- Η SQL μπορεί να ενημερώσει εγγραφές σε μια βάση δεδομένων

Η SQL αποτελείται από τις εντολές και τα ορίσματά τους, τις οποίες μπορούμε να χρησιμοποιήσουμε με συγκεκριμένους κανόνες σύνταξης για να πάρουμε τα αποτελέσματα που θέλουμε. Με την SQL μπορούμε να δημιουργήσουμε μια βάση δεδομένων και τους πίνακές της με τα αντίστοιχα πεδία, να καταχωρήσουμε, να τροποποιήσουμε και να διαγράψουμε τα δεδομένα αυτά, να αλλάξουμε τη δομή

των πινάκων με προσθήκη και διαγραφή πεδίων και να εμφανίσουμε. Η SQL έχει διάφορα τμήματα, τα πιο βασικά είναι τα παρακάτω:

Τη Γλώσσα Ορισμού Δεδομένων (DDL, Data Definition Language), η οποία περιέχει τις απαραίτητες εντολές για τον ορισμό και την τροποποίηση του σχεσιακού σχήματος, τη δημιουργία, την τροποποίηση και τη διαγραφή σχέσεων. Περιέχει ακόμη εντολές δημιουργίας και επεξεργασίας όψεων και ορισμού περιορισμών ακεραιότητας.

Τη Γλώσσα Χειρισμού Δεδομένων (DML, Data Manipulation Language), η οποία περιέχει τις απαραίτητες εντολές για την εμφάνιση δεδομένων καθώς και για την καταχώρηση, τροποποίηση και διαγραφή των εγγραφών (πλειάδων) μιας σχέσης.

Τέλος, περιέχει εντολές για τον ορισμό και την επεξεργασία συναλλαγών και εντολές για την ασφάλεια.

3.4.3 Εντολές και Τρόπος Χρήσης της Sql

Όπως δηλώνει και το όνομά της, η Sql είναι μία γλώσσα υποβολής και εκτέλεσης ερωτημάτων. Επίσης, περιλαμβάνει και μία σύνταξη για την ενημέρωση, την εισαγωγή και διαγραφή εγγραφών. Η DML, που αποτελεί κομμάτι της Sql χρησιμοποιεί τις παρακάτω εντολές:

- **SELECT:** προβάλλει τα δεδομένα από τις στήλες ενός πίνακα της βάσης δεδομένων. Η σύνταξη της εντολής φαίνεται στον παρακάτω πίνακα

select	A_1, A_2, A_n	←	Ονόματα πεδίων
from	R_1, R_2, R_m	←	Ονόματα πινάκων
where	P1;	←	Συνθήκη
group by	A1	←	Πεδία ομαδοποίησης
having	P2	←	Συνθήκη
order by	$A_1 \text{ asc/desc}$	←	Πεδίο ASC ή DESC

Πίνακας 6: Εντολή Select

Το select, αντιστοιχεί στην πράξη της προβολής της σχεσιακής άλγεβρας

Το from, στην πράξη του καρτεσιανού γινομένου

Το where, στη συνθήκη της πράξης

Το group by, μόνο σε περιπτώσεις που έχουν να κάνουν με ομαδοποίηση αποτελεσμάτων

Το having, περιορίζει τις γραμμές που επιστρέφονται από μία group by

Το order by, ταξινόμηση των εγγραφών κατά asc ή desc

- **INSERT:** εισαγωγή νέων δεδομένων σε μία βάση δεδομένων. Η σύνταξη της στον παρακάτω πίνακα:

insert into	R1	←	Εισαγωγή στον πίνακα
value	V1	←	Νέες τιμές στα πεδία

Πίνακας 7: Εντολή Insert

Η εντολή insert μπορεί να χρησιμοποιηθεί σε συνδυασμό με την select για να εισαχθούν πολλαπλές εγγραφές σε έναν πίνακα.

- **UPDATE:** ενημερώνει τα δεδομένα που βρίσκονται σε όλες τις γραμμές ενός πίνακα μιας βάσης δεδομένων

update	R1	←	Όνομα πίνακα
set	V1=new value	←	Όνομα στήλης
where	P	←	Συνθήκη

Πίνακας 8: Εντολή Update

- **DELETE:** διαγραφή δεδομένων από μία βάση δεδομένων

delete from	R	←	Όνομα γραμμής
where	P	←	Συνθήκη

Πίνακας 9: Εντολή Delete

Όταν λείπει το where τότε σβήνονται όλες οι εγγραφές του πίνακα

Διάφορες άλλες εντολές που ανήκουν στην DML είναι οι:

- **CREATE TABLE:** δημιουργεί έναν νέο πίνακα και η σύνταξή της φαίνεται παρακάτω

create table	R(A1D1,A2D2,...,AnDn)	←	Όνομα νέου πίνακα
<περιορισμός ακεραιότητας1>, <.....>, <περιορισμός ακεραιότητας2>			Ai= Ονόματα γνωρισμάτων Di= Τύποι πεδίων

Πίνακας 10: Εντολή Create Table

Οι περιορισμοί ακεραιότητας μπορούν να είναι της μορφής:

primary key $A_{j1}, A_{j2}, \dots, A_{jn}$, ορίζει το πρωτεύον κλειδί του πίνακα.

unique $A_{j1}, A_{j2}, \dots, A_{jn}$

check P έλεγχος μιας συνθήκης σε ένα πεδίο

foreign key (A_i) references A_i ορισμός ενός πεδίου ως ξένο κλειδί.

- **ALTER TABLE :** με αυτή την εντολή μπορούμε να μεταβάλλουμε τα πεδία ενός πίνακα. Η μορφή σύνταξης της παραπάνω εντολής μπορεί να είναι ενός των τριών τρόπων

alter table	R	←	Όνομα πίνακα
add	V type	←	Όνομα στήλης και τύπος

Πίνακας 11: Εντολή Alter Table(1)

alter table	R	← Όνομα πίνακα
drop	V	← Όνομα στήλης

Πίνακας 12: Εντολή Alter Table(2)

alter table	R	← Όνομα πίνακα
modify	V new type	← Όνομα στήλης και νέος τύπος

Πίνακας 13: Εντολή Alter Table(3)

H add προσθέτει μία νέα στήλη
 H drop διαγράφει μία στήλη
 H modify τροποποιεί μία στήλη

- *DROP TABLE* : για να σβήσουμε έναν πίνακα

drop table	R	← Όνομα πίνακα
------------	---	----------------

Πίνακας 14: Εντολή Drop Table

3.5 ΡΗΡΜΥADMIN



Εικόνα 13: Λογότυπο του PhpMyAdmin

3.5.1 Ιστορία του phpMyAdmin

Εμπνευστής του phpMyAdmin είναι ο Tobias Ratsiller, ιδρυτής της εταιρίας λογισμικού Maguma. Το 1998 άρχισε να εργάζεται σε front-end web εφαρμογές MySQL με την PHP, εμπνευσμένος από την MySQL-Webadmin. Το 2000, παρέδωσε το project phpAdsNew, του οποίου ήταν συντάκτης. Το 2001, προκειμένου να συντονίσουν τον αυξανόμενο αριθμό των επιδιορθώσεων, μια ομάδα από τρεις προγραμματιστές κατέγραψε το project phpMyAdmin στο SourceForge και ανέλαβαν την ανάπτυξη του. Η SourceForge φιλοξενεί τη λήψη του λογισμικού. Πριν από την έκδοση 4, η οποία χρησιμοποιεί Ajax εκτενώς για τη βελτίωση της εμπειρίας χρήσης, το λογισμικό χρησιμοποιείται σε HTML frames. Από τότε, η phpMyAdmin είχε γίνει μία από τις πιο δημοφιλείς εφαρμογές PHP και MySQL εργαλείων διαχείρισης, με μια μεγάλη κοινότητα χρηστών και συνεργατών.

3.5.2 Τι είναι το phpMyAdmin

Το phpMyAdmin είναι ένα δωρεάν λογισμικό ανοιχτού κώδικα, γραμμένο σε PHP και προορίζεται για τη διαχείριση βάσεων δεδομένων MySQL. Υποστηρίζει ένα ευρύ φάσμα εργασιών στα MySQL, MariaDB και Drizzle. Συχνά χρησιμοποιούμενες λειτουργίες (διαχείριση βάσεων δεδομένων, πίνακες, στήλες, σχέσεις, ευρετήρια, χρήστες, δικαιώματα, κ.λπ.) μπορούν να εκτελεστούν μέσω του user interface, ενώ έχουμε τη δυνατότητα να εκτελέσουμε άμεσα οποιαδήποτε δήλωση σε Sql.

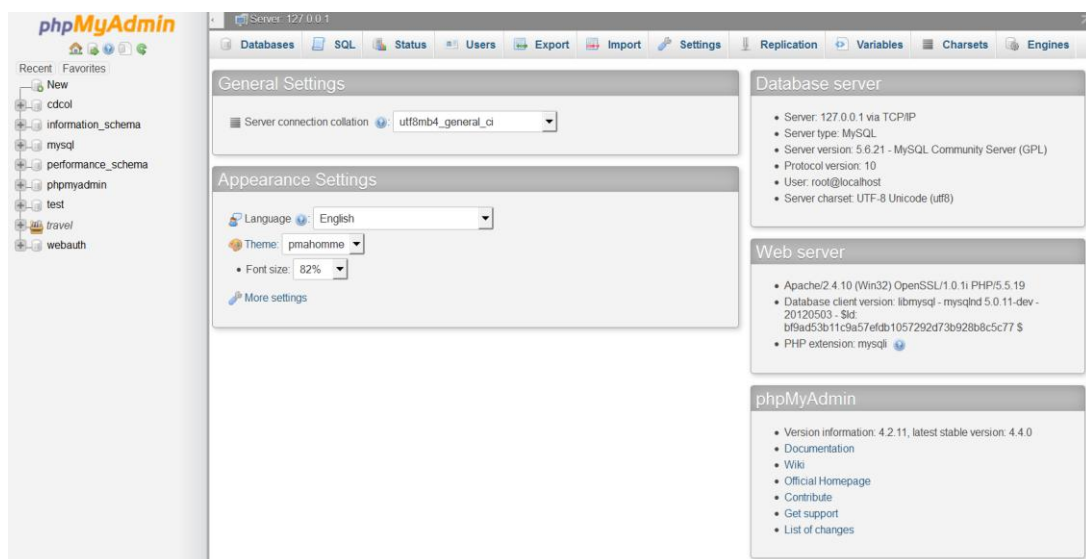
Έρχεται με ένα ευρύ φάσμα εγχειριδίου και χρήστες ευπρόσδεκτους να μας ενημερώσουν για τις σελίδες wiki να μοιράζονται ιδέες και howtos για διάφορες εργασίες. Το phpMyAdmin ομάδα θα προσπαθήσει να σας βοηθήσει αν αντιμετωπίζετε οποιοδήποτε πρόβλημα, μπορείτε να χρησιμοποιήσετε μια ποικιλία υποστήριξη κανάλια για να λάβετε βοήθεια. Για την ευκολία χρήσης του, είναι μεταφρασμένο σε 72 γλώσσες. Επιλέγη ως η καλύτερη PHP εφαρμογή σε διάφορα βραβεία και κέρδισε κάθε χρόνο το βραβείο "Best Tool or Utility for SysAdmins" από τη SourceForge.net Community.

Τα πλεονεκτήματα που συμπεριλαμβάνει είναι:

- Όμορφο web interface
- Διαχείριση βάσης δεδομένων MySQL
- Εισαγωγή δεδομένων από το αρχείο CSV και SQL
- Εξαγωγή δεδομένων σε διάφορες μορφές: CSV, SQL, XML, PDF, ISO, Word, LATEX κ.α.
- Διαχείριση πολλών διακομιστών
- Pdf γραφικά της βάσης δεδομένων
- Δημιουργία σύνθετων ερωτημάτων για την QBE
- Αναζήτηση βάσης δεδομένων σε παγκόσμιο πεδίο
- Ενεργή εποπτεία ερωτημάτων

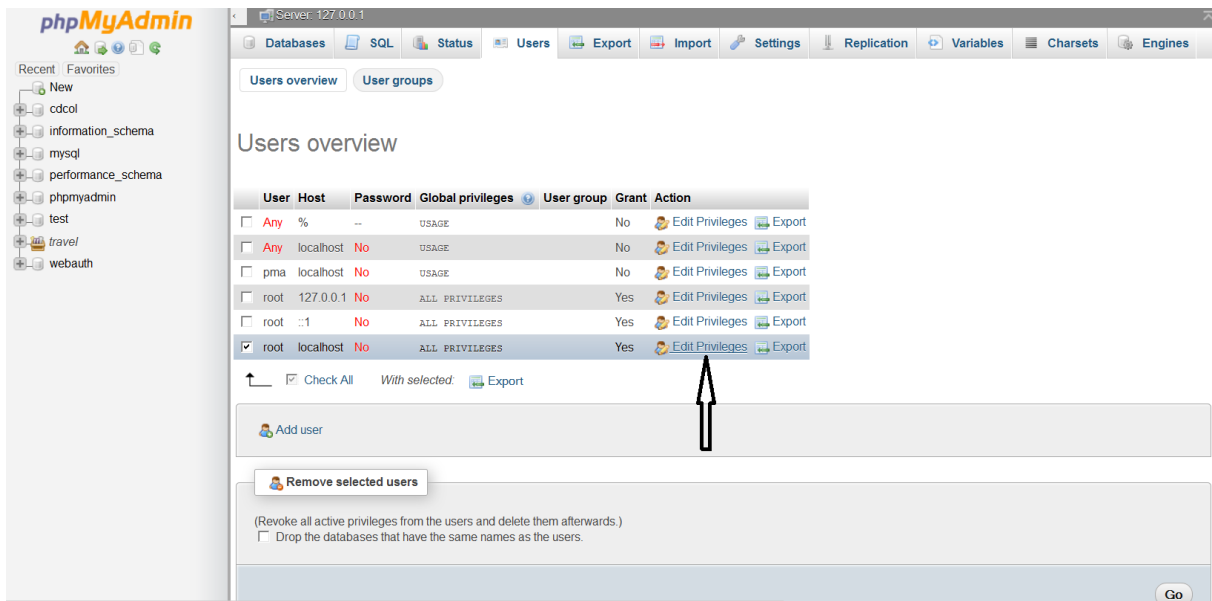
3.5.3 Web Interface

Για να εισέλθουμε στο web interface του phpMyAdmin, αρχικά πρέπει να ανοίξουμε το xampp και στη συνέχεια θα πρέπει να πληκτρολογήσουμε στον browser μας την διεύθυνση localhost/phpmyadmin .



Εικόνα 14: Web Interface-1 phpMyAdmin

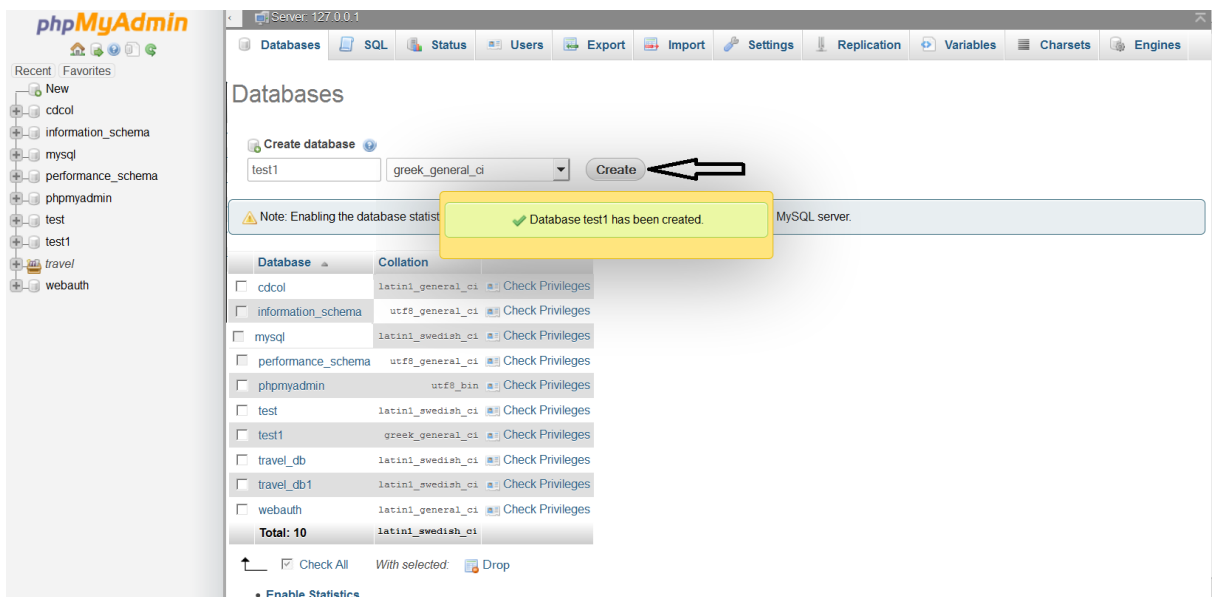
Στο αριστερό μέρος υπάρχουν οι βάσεις δεδομένων που έχουμε αποθηκεύσει. Η εφαρμογή από default της, δεν έχει κωδικό για τον root user. Για να βάλουμε κωδικό εισόδου στο phpMyAdmin, πρέπει να πάμε στην καρτέλα Users → Change password και να βάλουμε τον κωδικό που θέλουμε.



Εικόνα 15: Web Interface-2 phpMyAdmin

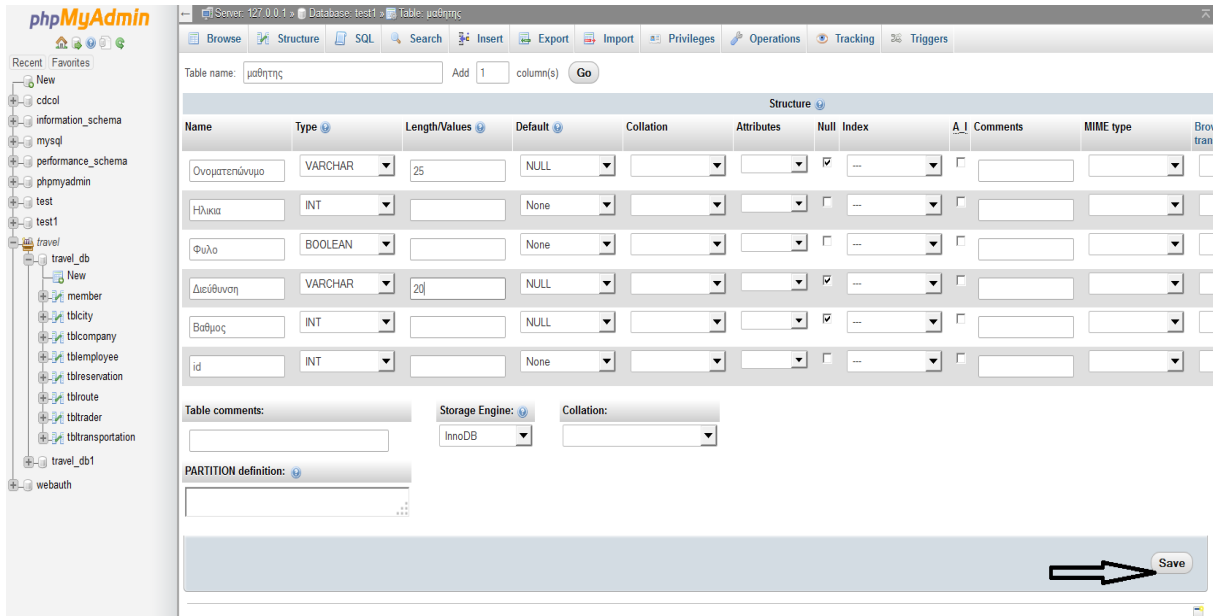
3.5.4 Δημιουργία βάσης δεδομένων

Η διαδικασία για να δημιουργήσουμε μία νέα βάση δεδομένων είναι πολύ εύκολη διαδικασία. Καθώς βρισκόμαστε στην κεντρική σελίδα του phpMyAdmin, πατάμε το κουμπί Databases. Ορίζουμε το νέο όνομα της βάσης και στο listbox επιλέγουμε την επιλογή greek_general_ci και πατάμε Create.



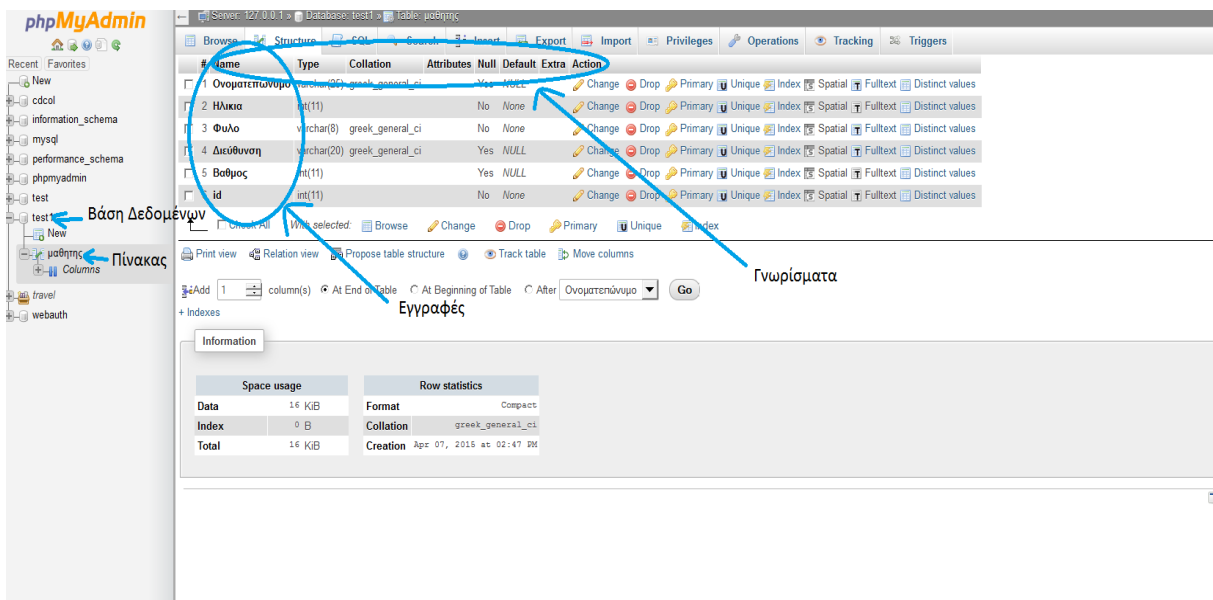
Εικόνα 16: Web Interface-3 phpMyAdmin

Για να φτιάξουμε μία εγγραφή, πατάμε πάνω στη βάση μας, δίνουμε το όνομα της εγγραφής (πίνακα) και επιλέγουμε τον αριθμό των χαρακτηριστικών που θα έχει ο πίνακάς μας. Δηλώνουμε τα χαρακτηριστικά και τον τύπο τους. Με αυτό τον τρόπο μπορούμε να δημιουργήσουμε πολλές εγγραφές με πολλά χαρακτηριστικά.



Εικόνα 17: Web Interface-4 phpMyAdmin

Η βάση δεδομένων είναι έτοιμη και έχει έναν πίνακα με κάποια πεδία



Εικόνα 18: Web Interface-5 phpMyAdmin

3.6 MYSQL

Η MySQL είναι αυτό που αποκαλούμε DBMS. (Database Management System – Σύστημα Διαχείρισης Βάσης Δεδομένων). Από το όνομα μπορούμε να συμπεράνουμε ότι ένα DBMS είναι ένα πρόγραμμα που διαχειρίζεται την δημιουργία, συντήρηση και χρήση μιας βάσης δεδομένων.



Εικόνα 19: Λογότυπο της MySql

3.6.1 Ιστορία

Η MySql δημιουργήθηκε από μία Σουηδική εταιρία, την MySql AB, της οποίας ιδρυτές ήταν οι David Axmark, Michael Widenius και Allan Larsson. Η πρώτη έκδοση της εμφανίστηκε στις 23 Μαΐου του 1995. Αρχικά δημιουργήθηκε για προσωπική χρήση στο χαμηλό της γλώσσας ISAM. Δημιούργησαν μία νέα διεπαφή Sql, ενώ κράτησαν το API της mSql. Με τη διατήρηση του συγκεκριμένου API, πολλοί προγραμματιστές ήταν σε θέση να χρησιμοποιούν την MySql αντί της mSql. Το 2008, η Sun Microsystems αγόρασε την MySql για το αστρονομικό ποσό των 1 δισεκατομμυρίων δολαρίων, ενώ το 2009 η Oracle σύναψε συμφωνία για την αγορά της Sun Microsystems. Το όνομά της προέρχεται από το όνομα της κόρης του συνιδρυτή Michael Widenius, My.

3.6.2 Πλεονεκτήματα

Ένα από τα μεγαλύτερα πλεονεκτήματα της MySql είναι η δυνατότητα που έχει για σύνδεση με διάφορες βάσεις δεδομένων. Οι βάσεις δεδομένων που υποστηρίζονται περιλαμβάνουν τις Adabas D, InterBase, PostgreSQL, dBase, FrontBase, SQLite, Empress, mSQL, Solid, FilePro (read-only), Direct MS-SQL, Sybase, Hyperwave, MySQL, Velocis, IBM, ODBC. Μεγάλοι ανταγωνιστές της είναι η PostgreSQL, Microsoft Sql και Oracle. Σε σύγκριση με τους παραπάνω έχει χαμηλό κόστος, εύκολη διαμόρφωση και είναι ανοιχτού κώδικα.

- Απόδοση: Η MySql είναι χωρίς αμφιβολία πολύ γρήγορη. Μπορείτε να δείτε την σελίδα δοκιμών και θα διαπιστώσετε ότι είναι γρηγορότερη από τους ανταγωνιστές της.
- Χαμηλό κόστος: Είναι διαθέσιμη δωρεάν, με άδεια ανοιχτού κώδικα. Η αγορά της εμπορικής άδειας είναι σαφώς μικρότερη των άλλων εταιριών.
- Ευκολία χρήσης: Οι περισσότερες καινούριες βάσεις δεδομένων χρησιμοποιούν την Sql. Σε περίπτωση που κάποιος έχει δουλέψει σε κάποιο άλλο σύστημα διαχείρισης βάσεων δεδομένων, θα είναι πολύ εύκολο να προσαρμοστεί.
- Κώδικας προέλευσης: Όπως και με την PHP, μπορούμε να έχουμε και να τροποποιήσουμε τον κώδικα προέλευσης της MySql.
- Μεταφερσιμότητα: Μπορεί να χρησιμοποιηθεί σε διαφορετικά λειτουργικά συστήματα, όπως Windows, Unix, Mac OS.
- Νέα έκδοση: Η νέα έκδοση 5 έχει έρθει με εντυπωσιακές λειτουργίες και είναι ικανή να στηρίζει μεγάλες εφαρμογές με υψηλή αξιοπιστία.

3.6.3 MySql-Php

Η php θεωρείται η καλύτερη γλώσσα για την δημιουργία δυναμικών εφαρμογών που επικοινωνούν με βάσεις δεδομένων, ένας βασικός λόγος για να την επιλέξουμε είναι ότι διαθέτει ενσωματωμένες συναρτήσεις που επικοινωνούν με μεγάλο αριθμό εμπορικών συστημάτων βάσεων δεδομένων. Το παρακάτω παράδειγμα παρουσιάζει τις συναρτήσεις, που χρησιμοποιούνται για τη σύνδεση με μια βάση δεδομένων στη MySql, τη δημιουργία επερωτήσεων και την ανάκτηση των αποτελεσμάτων για περαιτέρω επεξεργασία τους από την εφαρμογή

```
mysql_connect("server name", "user name", "password");
```

Πίνακας 15: Παράδειγμα-1 MySql

Σε περίπτωση αποτυχημένης σύνδεσης πρέπει να ειδοποιείτε ο χρήστης. Αυτό γίνεται με τον παρακάτω κώδικα

```
<?php
    $con = mysql_connect("local host", "test-user", "test-pass");
    if (!con)
    {
        die('Could not connect: ' . mysql_error());
    }
?>
```

Πίνακας 16: Παράδειγμα-2 MySql

Το παρακάτω παράδειγμα παρουσιάζει τις συναρτήσεις, που χρησιμοποιούνται για τη σύνδεση με μια βάση δεδομένων στη MySql, τη δημιουργία επερωτήσεων και την ανάκτηση των αποτελεσμάτων για περαιτέρω επεξεργασία τους από την εφαρμογή

```
<?php
    $con = mysql_connect("local host", "test-user", "test-pass");
    if (!con)
    {
        die('Could not connect: ' . mysql_error());
    }
    mysql_select_db("mydb" , $con);
    $result = mysql_query("SELECT * mytable" , $con);
    while($row = mysql_fetch_array($result , MYSQL_NUM))
    {
        //edit results
    }
    mysql_close($con);
?>
```

Πίνακας 17: Παράδειγμα-3 MySql

- `mysql_connect()`: χρησιμοποιείται για να γίνει η σύνδεση του MySQL Server και έχει σαν παραμέτρους : 'hostname' ή την IP του server, 'όνομα χρήστη', 'κωδικός ασφαλείας'. Η τιμή που επιστρέφει η συνάρτηση χρησιμοποιείται σαν παράμετρος στις επόμενες συναρτήσεις που θα καλέσουν τον MySQL Server.
- `mysql_error()`: επιστρέφει ένα μήνυμα σφάλματος σε περίπτωση αποτυχίας της προηγούμενης ενέργειας που πραγματοποιήθηκε.
- `mysql_select_db()`: επιλέγει μία συγκεκριμένη βάση δεδομένων, η οποία έχει ήδη δημιουργηθεί. Οι παράμετροι που περιλαμβάνει είναι οι εξής: 'όνομα βάσης δεδομένων', 'τιμή που επιστράφηκε από την `mysql_connect()`'.
- `mysql_query()`: χρησιμοποιείται για την αποστολή μίας ερώτησης προς τη βάση δεδομένων που ορίζει η δεύτερη παράμετρος (\$con). Η πρώτη παράμετρος είναι αυτή που θα εκτελεστεί.
- `mysql_fetch_array()`: επιτρέπει την ανάκτηση των εγγραφών, δηλαδή των rows, που επέστρεψε το προηγούμενο ερώτημα που έγινε προς τη βάση δεδομένων.
- `mysql_close()`: χρησιμοποιείται για να κλείσει η σύνδεση που ορίζει η παράμετρος \$con. Το τελευταίο δεν είναι απαραίτητο να ισχύει, αφού όταν τελειώσει το script τότε κλείνουν αυτόματα όλες οι συνδέσεις.

3.7 HTML



Εικόνα 20: Λογότυπο HTML5

Η HTML είναι το ακρωνύμιο των λέξεων HyperText Markup Language, δηλαδή γλώσσα χαρακτηρισμού υπερκειμένου και είναι υποσύνολο της γλώσσας SGML, που επινοήθηκε από την IBM προκειμένου να λυθεί το πρόβλημα της μη τυποποιημένης εμφάνισης κειμένων σε διάφορα υπολογιστικά συστήματα. Η HTML ορίζει ένα σύνολο κοινών μορφοποιήσεων για τις ιστοσελίδες, όπως τίτλοι (titles), επικεφαλίδες (headings), παράγραφοι (paragraphs), λίστες (lists) και πίνακες (tables). Ορίζει επίσης μορφοποίηση χαρακτήρων, όπως η έντονη γραφή. Οι browser, εκτός από τη δυνατότητα που έχουν να ανακτούν σελίδες από το internet, λειτουργούν και σαν μορφοποιητές για την HTML. Όταν ο browser προσπαθεί να διαβάσει μία ιστοσελίδα, στην πραγματικότητα, διερμηνεύει τα tags της HTML και μορφοποιεί το κείμενο και τις εικόνες στην οθόνη. Διαφορετικοί browsers μπορεί να παρουσιάζουν διαφορετικά τα στυλ των στοιχείων μιας ιστοσελίδας. Αυτό σημαίνει ότι οι ιστοσελίδες που δημιουργούμε με την HTML μπορεί να δείχνουν εντελώς διαφορετικές από λειτουργικό σε λειτουργικό και από browser σε browser. Δηλαδή, οι πραγματικές πληροφορίες και οι σύνδεσμοι που περιέχουν οι ιστοσελίδες δεν παραμετροποιούνται, αλλά η εμφάνιση των σελίδων θα είναι διαφορετική.

3.7.1 Ιστορία και Εκδόσεις

Η HTML βασίστηκε σε ένα σύστημα χρήσης και διαμοιρασμού εγγράφων για τους ερευνητές του CERN, το οποίο (σύστημα) επινόησε ο Τιμ Μπερνερς Λι το 1980. Το 1989, πρότεινε ένα σύστημα βασισμένο στο διαδίκτυο που θα χρησιμοποιούσε υπερκείμενο. Εν τέλει, έφτιαξε την προδιαγραφή της HTML και έγραψε τον browser και το λογισμικό έναν χρόνο αργότερα. Η πρώτη δημόσια διαθέσιμη περιγραφή της HTML ήταν ένα έγγραφο με το όνομα Ετικέτες HTML, το οποίο πρωτοαναφέρθηκε στο Διαδίκτυο από τον Μπέρνερς Λι στα τέλη του 1991. Μετά που τα πρόχειρα HTML και HTML+ έληξαν, στις αρχές του 1994, το IETF δημιούργησε μία ομάδα για την HTML, η οποία το 1995 ολοκλήρωσε την HTML 2.0, με την πρόθεση να αποτελέσει την πρώτη προδιαγραφή πάνω στην οποία θα βασίζονταν οι μελλοντικές υλοποιήσεις. Το 1997 δημοσιεύτηκε ως Σύσταση από τον W3C η έκδοση 3.2. Το 1997, δημοσιεύτηκε η HTML 4, ως Σύσταση. Παρείχε τρεις εκδοχές: την *Strict*, στην οποία απαγορεύεται η χρήση ξεπερασμένων στοιχείων, την *Transitional*, στην οποία επιτρέπονται τα ξεπερασμένα στοιχεία, την *Frameset*, στην οποία επιτρέπονται μόνο στοιχεία frame. Τον Ιανουάριο του 2008 δημοσιεύεται η HTML5 ως ένα Πρόχειρο Εργασίας.

3.7.2 Ετικέτες

Οι ετικέτες ελέγχουν την δομή και την μορφή του κειμένου της ιστοσελίδας. Επίσης παρέχουν πληροφορίες προς τον browser για την σελίδα που πρόκειται να εμφανίσουν, όπως ο τίτλος της σελίδας ή ο συγγραφέας της, κ.α. Οι ετικέτες γράφονται ανάμεσα στα σύμβολα < και >. Οι περισσότερες ετικέτες αποτελούνται από μια ετικέτα αρχής και μια ετικέτα τέλους και ανάμεσα σε αυτές υπάρχει το κείμενο που χαρακτηρίζεται από τις ετικέτες αυτές. Η ετικέτα τέλους περιέχει τον χαρακτήρα / πριν το όνομα της ετικέτας.

- <όνομα-ετικέτας> ... κείμενο ... </όνομα-ετικέτας>
- Κείμενο με έμφαση .

Υπάρχουν ορισμένες ετικέτες που δεν έχουν ετικέτες τέλους. Στις ετικέτες αυτές, πριν από το σύμβολο > τοποθετούμε τον χαρακτήρα /

- <όνομα-ετικέτας />
-

Ένα ζευγάρι ετικετών καθορίζει την μορφή ενός κειμένου ή μιας και μόνο λέξης. Για παράδειγμα αν ήθελα να γράψω στην γλώσσα HTML το κείμενο 'Μαθαίνω HTML' με έντονη γραφή, θα το έγραφα έτσι:

- Μαθαίνω HTML.

Μόλις συναντήσαμε την πρώτη HTML ετικέτα. Το γράμμα b είναι το αρχικό από την λέξη bold. Το ζευγάρι των ετικετών και , δηλώνει στον web browser, ότι το κείμενο που είναι γραμμένο μεταξύ των ετικετών αυτών θα εμφανιστεί με έντονα γράμματα. Αν ήθελα να γράψω την λέξη wlearn με πλάγιους χαρακτήρες θα έγραφα:

- <i>wlearn</i>

Το ζευγάρι των ετικετών <i> και </i>, λέει στον browser να εμφανίσει το κείμενο με πλάγιους χαρακτήρες. Το γράμμα i είναι το πρώτο γράμμα από την λέξη italics. Οι ετικέτες μέσα σε ένα HTML αρχείο μπορούν να είναι γραμμένες είτε με πεζά, είτε με κεφαλαία. Η ετικέτα είναι ίδια με τη ετικέτα . Μόνο πρέπει να προσέχουμε τις ετικέτες τέλους να είναι γραμμένες όπως η ετικέτες

αρχής. Δηλαδή αν μια ετικέτα αρχής είναι γραμμένη με πεζά τότε και η ετικέτα τέλους πρέπει να γραφεί με πεζά.

Λάθος τρόπος:

- `wlearn` ή `wlearn`

Σωστός τρόπος:

- `wlearn` ή `wlearn`

Ένα αρχείο HTML αρχίζει πάντα με την ετικέτα `<html>` και αποτελείται από δύο ενότητες. Η πρώτη είναι η κεφαλή HEAD και η δεύτερη το κυρίως περιεχόμενο BODY. Η ετικέτα `<html>` πληροφορεί τον browser ότι οι γραμμές που περικλείονται μέσα σε αυτές τις ετικέτες, είναι κώδικας γραμμένος σε HTML.

Η ετικέτα `<head>` αποτελεί τον πρόλογο της ιστοσελίδας και συνήθως περιλαμβάνει λίγες ετικέτες. Βασική ετικέτα της ενότητας HEAD είναι η `<title>` και ακολουθούν οι `<meta>`, `<link>` και `<script>`.

Η ετικέτα `<body>` περιλαμβάνει – όπως φαίνεται και από το όνομά της – το κύριο περιεχόμενο της ιστοσελίδας. Εδώ προσθέτουμε εικόνες, βίντεο και τη μορφοποίηση.

Παρακάτω δείχνουμε έναν κώδικα, γραμμένο σε HTML.

```
<html>

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

<body>
  <!-- Κυρίως Περιεχόμενο -->
</body>

</html>
```

Πίνακας 18: Παράδειγμα δομής HTML

Βασικές ετικέτες της Html που καθορίζουν επικεφαλίδες, παραγράφους και γραμμές είναι οι `
`, `<p>`, `<hr>`, `<h1>`. Για την διαμόρφωση του κειμένου, συχνά χρησιμοποιούμενες ετικέτες είναι οι ``, ``, `<big>`, ``, `<i>`, `<u>`, `<small>`, `<sub>`, `<sup>`, `<code>`, `<var>`, `<tt>`. Για τη θέση και τη στοίχιση του κειμένου χρησιμοποιούμε τις `<center>`, `<blockquote>`, `<p>`. Για τις λίστες ``, ``, `<dir>`, ``, `<dl>`, `<dt>`, `<dd>`.

3.7.3 XHTML

Αξίζει να σημειωθεί, πως για την υλοποίηση της εφαρμογής χρησιμοποιήθηκε η έκδοση XHTML. Το όνομά της προέρχεται από τις λέξεις eXtensible HyperText Markup Language, που σημαίνει εκτεταμένη γλώσσα σήμανσης υπερκειμένου. Εμείς απλά θα τη λέμε με το μικρό της όνομα, XHTML. Αποτελεί στάνταρντ του Web και η έκδοση 1.0, έγινε επίσημη Σύσταση του W3C το 2000. Η XHTML είναι παρόμοια με την HTML, αποτελεί μία αυστηρότερη έκδοση της HTML και σκοπεύει να αντικαταστήσει την προκάτοχό της.

Παρακάτω, αναφέρουμε ένα παράδειγμα κώδικα, δομημένο στη γλώσσα XHTML.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

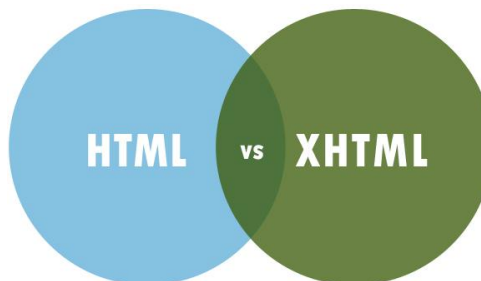
<head>
  <title></title>
</head>

<body>
<div id="wrapper">
  <div id="menu">
  </div>
</div>

</body>
</html>
```

Πίνακας 19: Παράδειγμα δομής XHTML

3.7.4 HTML vs XHTML



Εικόνα 21: Διαφορές HTML-XHTML

Οι διαφορές του είναι λίγες αλλά πολύ σημαντικές. Μπορούν να χωριστούν σε τρεις ενότητες.

Δομή του εγγράφου:

- Η XHTML DOCTYPE είναι υποχρεωτική
- Η ιδιότητα xmlns είναι υποχρεωτική στο <html>
- Οι ετικέτες <html>, <head>, <title> και <body> είναι υποχρεωτικές

Στοιχεία της XHTML:

- Τα στοιχεία της πρέπει να είναι σωστά φωλιασμένα
- Τα στοιχεία της πρέπει να είναι πάντα κλειστά
- Τα στοιχεία της πρέπει να είναι γραμμένα με πεζούς χαρακτήρες
- Τα έγγραφα της πρέπει να έχουν ένα root element

Ιδιότητες XHTML:

- Οι ιδιότητες των ονομάτων πρέπει να είναι γραμμένα με μικρούς χαρακτήρες
- Οι ιδιότητες των τιμών πρέπει να παραθετηθούν
- Η ιδιότητα ελαχιστοποίηση, απαγορεύεται

Ο παρακάτω πίνακας περιέχει όλες τις ετικέτες της HTML

<!-->	Σχόλια	<!DOCTYPE>	Πρότυπο
<a>	Link	<abbr>	Συντόμευση
<acronym>	Ακρωνύμιο	<address>	Διεύθυνση
<applet>	Εφαρμογή Java	<area>	Περιοχή σε μία εικόνα
<article>	Εξωτερικό περιεχόμενο	<aside>	Χωριστό κείμενο
<audio>	Εισαγωγή ήχου		Bold
<base>	URL	<basefont>	Μέγεθος κειμένου
<bdo>	Αλλαγή κατεύθυνσης	<big>	Μεγαλύτερα γράμματα
<blockquote>	Αριστερό περιθώριο	<body>	Σώμα σελίδας
 	Αλλαγή γραμμής	<button>	Εισαγωγή κουμπιού
<canvas>	Σχεδίαση γραφικών	<caption>	Λεζάντα πίνακα
<center>	Κεντράρισμα κειμένου	<cite>	Παραθέτει αναφορά
<code>	Εμφάνιση κώδικα	<col>	Ιδιότητες πίνακα
<colgroup>	Ομάδα στηλών	<command>	Κουμπί ελέγχου
<datalist>	Επιλογές πεδίου	<dd>	Περιγραφή λίστα ορισμ
	Διαγραμμένο κείμενο	<details>	Πληροφορίες
<dfn>	Υποδηλώνει όρο	<dir>	Δημιουργία λίστας
<div>	Ορίζει ένα τμήμα	<dl>	Εισάγει λίστα ορισμού
<dt>	Όρος λίστας ορισμών		Πλάγιοι,αχνοί χαρακτη
<embed>	Εξωτερική εφαρμογή	<fieldset>	Πλαίσιο
<figcaption>	Επικεφαλίδα στοιχείου	<figure>	Ομαδοποίηση
	Μέγεθος,χρώμα,γραμματ	<footer>	Υποσέλιδο
<form>	Ορίζει φόρμα σχολίων	<frame>	Πλαίσια
<frameset>	Σειτ πλαisiών	<h1>-<h6>	Επικεφαλίδες
<head>	Επικεφαλίδα	<header>	Επικεφαλίδα σελίδας
<hgroup>	Ομαδοποιεί επικεφαλίδε	<hr>	Οριζόντια γραμμή
<html>	Αρχικό tag	<i>	Italic
<iframe>	Ένθετο παράθυρο		Εισαγωγή εικόνας
<input>	Εισαγωγή πεδίο φόρμας	<ins>	Προστιθέμενο κείμενο
<keygen>	Ταυτοποίηση χρήστη	<kbd>	Υπόδειξη κειμένου
<label>	Κείμενο μηχανισμού φο	<legend>	Λεζάντα
	Προσθέτει γραμμές	<link>	Εξωτερικό αρχείο
<map>	Ορίζει χάρτη εγγράφου	<mark>	Ορίζει μαρκαρισμένο κείμ
<menu>	Ορίζει μία λίστα/μενού	<meta>	Στοιχεία search machine
<meter>	Ορίζει βαθμωτό μέτρησης	<nav>	Σύνδεσμος πλοήγησης
<noframe>	Ορίζει τμήμα noframe	<noscript>	Ορίζει τμήμα noscript
<object>	Ενσωματωμένο αντικείμεν		Αριθμημένη λίστα
<optgroup>	Ορίζει option group	<option>	Στοιχείο σε listbox
<output>	Αποτέλεσμα υπολογισμού	<p>	Αλλαγή παραγράφου
<param>	Παράμετρος αντικειμένου	<pre>	Εισαγωγή κειμένου
<progress>	Πρόοδος μίας εργασίας	<q>	Σύντομη παραπομπή
<rp>	Μη υποστηριξη ruby σχολ	<rt>	Υποστήριξη ruby σχόλιων
<ruby>	Ορίζει ruby σχόλια	<s>	Επισημανση μη ορθού κειμ

<samp>	Δειγματικός κωδ. υπολογι	<script>	Ορίζει ένα script
<section>	Ορίζει ενότητες	<select>	Ορίζει λίστα επιλογής
<small>	Μικρότερα γράμματα	<source>	Ορίζει πηγές audio,video
	Ορίζει τμήμα σε έγγραφο	<strike>	Διαγράμμιση κειμένου
	Ορίζει έντονο κείμενο	<style>	Ορίζει ένα στυλ
<sub>	Δείκτης	<summary>	Ορίζει τίτλο για 'Default'
<sup>	Εκθέτης κειμένου	<table>	Ορίζει έναν πίνακα
<tbody>	Ορίζει έναν table body	<td>	Ορίζει ένα κελί πίνακα
<textarea>	Ορίζει μία περιοχή κειμέν	<tfoot>	Ορίζει ένα fixed tabl foote
<th>	Ορίζει επικεφαλίδα πίνακ	<thead>	Ορίζει fixed tabl header
<time>	Ορίζει ώρα/ημερομηνία	<title>	Τίτλος σελίδας
<tr>	Ορίζει γραμμή πίνακα	<tt>	Ορίζει κείμενο teletype
<u>	Κείμενο με ορθ/φικά λάθη		Ορίζει μη αριθμημένη λιστ
<var>	Ορίζει μία μεταβλητή	<video>	Ορίζει βίντεο
<wbr>	Πρόσθεση γραμμής διαλείμματος		

Πίνακας 20: Ετικέτες της HTML

3.8 JAVASCRIPT

Η javascript είναι μια scripting γλώσσα η οποία έχει σχεδιαστεί και χρησιμοποιείται για να εισάγουμε την έννοια της διαδραστικότητας στις html σελίδες. Με την javascript μπορούμε να εκτελέσουμε κάποια πράγματα όταν συμβαίνει ένα γεγονός, για παράδειγμα όταν ο χρήστης κλικάρει σε ένα html στοιχείο, να εκτελείται κάποιο script και λαμβάνουμε τα αντίστοιχα αποτελέσματα. Αυτό ουσιαστικά σημαίνει, ότι με την javascript κάνουμε την εφαρμογή μας δυναμική. Έχουν δημιουργηθεί πανίσχυρες βιβλιοθήκες, οι οποίες διευκολύνουν την ανάπτυξη εφαρμογών σε web, πλατφόρμες και κινητά. Για αυτό το λόγο θα πρέπει να την προτιμούμαι σε σχέση με την Java, την C++ ή την Ruby. Σήμερα, όλοι οι browsers είναι συμβατοί με την JavaScript. Το γεγονός ότι η γλώσσα υποστηρίζει τόσο τον functional προγραμματισμό όσο και τον αντικειμενοστραφή (object oriented) την κάνει ακόμα πιο δημοφιλή.

- Είναι πολύ γρήγορη και χρήσιμη και επιτρέπει γρήγορη επικοινωνία ανάμεσα στον browser και τον server
- Χρησιμοποιείται με ενιαίο τρόπο στο διαδίκτυο, ανεξαρτήτως από την γλώσσα προγραμματισμού στον server (PHP, Python)
- Η JavaScript δουλεύει ακόμα και όταν ο browser είναι offline
- Η γλώσσα είναι συμβατή με όλα τα σύγχρονα PCs, tablets, phones
- Χρησιμοποιείται από τις μεγαλύτερες εταιρίες στον χώρο της τεχνολογίας, όπως οι Apple, Google και Microsoft

Τα τελευταία χρόνια υλοποιούνται πολλά ενδιαφέροντα λογισμικά που την υποστηρίζουν και η JavaScript έχει τραβήξει ακόμη περισσότερο τα βλέμματα. Έχει εδραιωθεί στο διαδίκτυο, ενώ παράλληλα έχει αρχίσει να επεκτείνεται και σε άλλους τομείς καθώς οι βιβλιοθήκες που έχουν υλοποιηθεί σε αυτήν την κάνουν ακόμα πιο ευέλικτη και πιο αναγνωρίσιμη. Κάποιες ενδιαφέρουσες βιβλιοθήκες που αναδεικνύουν την χρησιμότητα της γλώσσας αλλά και την συνεισφορά της στο web προγραμματισμό:

JQuery

Η JQuery είναι η βιβλιοθήκη της JavaScript που έχει χρησιμοποιηθεί όσο καμία άλλη όλα αυτά τα χρόνια. Η JQuery απλοποιεί τον προγραμματισμό καθώς με την βοήθεια της μπορούν να υλοποιηθούν tasks πολύ απλά, τα οποία με την συμβατική JavaScript θα ήταν ιδιαίτερα πολύπλοκα. Στα πλεονεκτήματά της συγκαταλέγεται και ο μεγάλος αριθμός δυνατοτήτων και λειτουργιών που έχει, καθώς και το μεγάλο open-source community που την υποστηρίζει. Θεωρείται από τις απαραίτητες βιβλιοθήκες για την υλοποίηση μια διαδικτυακής εφαρμογής, είτε αυτή είναι απλή, είτε όχι.

Node.JS

Η Node.js χρησιμοποιείται στις real-time εφαρμογές, στέλνοντας δεδομένα μέσω web sockets. Αυτό που την κάνει τόσο επαναστατική είναι ότι μετά από είκοσι χρόνια με χρήση μόνο απλών ερωτο-απαντήσεων τύπου request-response, πλέον μπορούν να υπάρξουν real-time εφαρμογές με επικοινωνία που ξεκινά και από τις δυο μεριές, του πελάτη και του server, οι οποίοι μπορούν να ανταλλάξουν δεδομένα ελεύθερα, χωρίς περιορισμούς. Αυτό έρχεται σε αντίθεση με τα όσα ίσχυαν μέχρι πριν λίγα χρόνια, όταν μόνο ο πελάτης μπορούσε να ξεκινήσει την επικοινωνία (Ajax). Επιπροσθέτως, η Node.js, βασίζεται στην ισχύουσα δομή του web (HTML, CSS και Javascript) και χρησιμοποιεί την διαδεδομένη θύρα 80.

Angular JS

Ένα ενδιαφέρον framework ανοιχτού λογισμικού της Google, αλλάζει τα δεδομένα στον τομέα στον διαδικτυακών εφαρμογών, καθώς έχει σχεδιαστεί με τους εξής στόχους:

- Αποδέσμευση του DOM από την λογική της εφαρμογής, βελτιώνοντας την δυνατότητα ελέγχου του κώδικα
- Ο έλεγχος της εφαρμογής έχει ίση σημασία με την διαδικασία υλοποίησης. Η δυσκολία των δοκιμών επηρεάζεται δραματικά από τον τρόπο που είναι δομημένος ο κώδικας
- Αποσύνδεση της πλευράς του πελάτη (browser) από την πλευρά του διακομιστή. Αυτό επιτρέπει το έργο να προχωρήσει παράλληλα και επιτρέπει την επαναχρησιμοποίηση κώδικα και στις δύο πλευρές.
- Οδηγός για τους προγραμματιστές σε όλη τη διαδικασία υλοποίησης μιας εφαρμογής: από το σχεδιασμό του user interface, στη δοκιμή.
- Ακολουθεί το μοτίβο MVC της μηχανικής λογισμικού και ενθαρρύνει τη χαλαρή σύνδεση μεταξύ παρουσίασης, δεδομένων και λογικής.
- Ένα μεγάλο μέρος της επιβάρυνσης για το backend μειώνεται, οδηγώντας σε πολύ ελαφρύτερες web εφαρμογές.

Γενικά, το AngularJS δένει την HTML (views) σε αντικείμενα JavaScript (models). Όταν συμβαίνει μια αλλαγή στα models, η σελίδα ανανεώνεται αυτόματα. Συμβαίνει επίσης και το αντίθετο, όταν δηλαδή ένα text field ενημερώνεται στο view, το AngularJS χειρίζεται την κατάσταση αυτόματα χωρίς να χρειάζεται να γίνει κάποια επιπλέον ενέργεια στην HTML, ή στην JQuery με την σύνδεση κάποιου event.

PhoneGap

Τελευταία, παρατηρείται μια κατακόρυφη αύξηση της χρήσης κινητών συσκευών και των εφαρμογών τους. Η JavaScript έχει αρχίσει να διεισδύει και σε αυτόν τον τομέα και μάλιστα με πολύ επιτυχημένα, πρώτα βήματα, όπως το PhoneGap (ή αλλιώς Apache Cordova).

Το PhoneGap είναι ένα framework για ανάπτυξη λογισμικού σε κινητές συσκευές που έχει πλέον αγοραστεί από την Adobe Systems και το οποίο δίνει την δυνατότητα στους προγραμματιστές, να δημιουργήσουν εφαρμογές για κινητές συσκευές με χρήση HTML5, CSS3 και JavaScript και όχι στις γλώσσες που υποστηρίζονται ανά συσκευή. Οι εφαρμογές που προκύπτουν είναι υβριδικές. Δηλαδή, δεν κάνουν χρήση των frameworks που ορίζει το λειτουργικό της κινητής συσκευής, αλλά ταυτόχρονα δεν αποτελούν καθαρά web-based εφαρμογές. Οι πλατφόρμες κινητών συσκευών, όπως το iOS, το Android και το Windows έχουν διαφορετικές απαιτήσεις, κανόνες και γλώσσες προγραμματισμού μεταξύ τους. Με την χρήση όμως του PhoneGap, η υλοποίηση πλέον μιας εφαρμογής γίνεται μια φορά για όλες τις πλατφόρμες.

Παρόλα αυτά, το PhoneGap έχει κάποια μειονεκτήματα, όπως ότι δεν μπορεί να χρησιμοποιηθεί για την υλοποίηση εφαρμογών με πολλά γραφικά (π.χ. παιχνίδι) αλλά και το γεγονός πως δεν περιέχει κάποια βάση έτοιμου user interface περιεχομένου, με αποτέλεσμα μερικές φορές η υλοποίηση να κρατάει περισσότερο.

Όμως, το πρώτο βήμα για την ενοποίηση όλων των πλατφόρμων που χρησιμοποιούνται στις κινητές συσκευές έγινε. Και έγινε με την βοήθεια της JavaScript.

Οι τελευταίες εξελίξεις στον χώρο του προγραμματισμού αναδεικνύουν κυρίαρχο του παιχνιδιού την JavaScript, η οποία γίνεται όλο και πιο δημοφιλής. Πλέον, έχουν δημιουργηθεί πανίσχυρες βιβλιοθήκες, οι οποίες διευκολύνουν την ανάπτυξη εφαρμογών σε web, πλατφόρμες και κινητά. Το καλύτερο όμως είναι, ότι η υλοποίησή τους θα μπορούσε να γίνει ενοποιημένα, γεγονός που βοηθάει τους προγραμματιστές στην δημιουργία νέων, καινοτόμων προγραμμάτων.

3.8.1 Ιστορία

Η γλώσσα προγραμματισμού JavaScript δημιουργήθηκε από τον Brendan Eich, υπάλληλο της εταιρείας Netscape με την ονομασία Mocha. Αργότερα μετονομάστηκε σε LiveScript αλλά τελικά πήρε την ονομασία JavaScript κυρίως επειδή η ανάπτυξή της επηρεάστηκε από τη γλώσσα Java. Όταν πρωτοκυκλοφόρησε στην αγορά, η επίσημη ονομασία της ήταν LiveScript και ήταν σε beta έκδοση τον Σεπτέμβριο του 1995. Η JavaScript απέκτησε μεγάλη επιτυχία ως γλώσσα client-side για εκτέλεση κώδικα σε ιστοσελίδες, και περιλήφθηκε σε διάφορα προγράμματα περιήγησης στο Web. Τον Νοέμβριο του 1996, η Netscape ανακοίνωσε ότι είχε υποβάλει τη γλώσσα JavaScript στο Ecma International -μια οργάνωση της τυποποίησης των γλωσσών προγραμματισμού- για εξέταση ως βιομηχανικό πρότυπο, και στη συνέχεια το έργο είχε ως αποτέλεσμα την τυποποιημένη μορφή που ονομάζεται ECMAScript.

Η JavaScript έχει γίνει μία από τις πιο δημοφιλείς γλώσσες προγραμματισμού ηλεκτρονικών υπολογιστών στο Web. Αρχικά, όμως, πολλοί επαγγελματίες προγραμματιστές υποτίμησαν τη γλώσσα διότι το κοινό της ήταν ερασιτέχνες συγγραφείς ιστοσελίδων και όχι επαγγελματίες προγραμματιστές. Με τη χρήση της τεχνολογίας Ajax, η JavaScript γλώσσα επέστρεψε στο προσκήνιο και έφερε πιο επαγγελματική προσοχή προγραμματισμού. Αποτέλεσμα ήταν ένα καινοτόμο αντίκτυπο στην εξάπλωση των παισιών και των βιβλιοθηκών, τη βελτίωση προγραμματισμού με JavaScript, καθώς και αυξημένη χρήση της JavaScript έξω από τα προγράμματα περιήγησης στο Web.

Τον Ιανουάριο του 2009, το έργο CommonJS ιδρύθηκε με στόχο τον καθορισμό ενός κοινού προτύπου βιβλιοθήκης κυρίως για την ανάπτυξη της JavaScript έξω από το πρόγραμμα περιήγησης και μέσα σε άλλες τεχνολογίες π.χ. server-side.

3.8.2 Μοντέλο Εκτέλεσης

Η γλώσσα βασίστηκε κατά πολύ στην C για τη σύνταξή της. Ο κώδικάς της ενσωματώνεται στον HTML κώδικα, ενώ τα scripts εκτελούνται χωρίς να έχει προηγηθεί μεταγλώττιση του κώδικα. Είναι μία πολύ βοηθητική γλώσσα για τους προγραμματιστές-σχεδιαστές διαδικτύου, είναι πολύ εύκολη στην εκμάθησή της. Τοποθετείται μέσα στην HTML και ανάμεσα στις ετικέτες `<script>` και `</script>`. Όταν υπάρχει κώδικάς της μέσα σε ένα HTML έγγραφο, τότε ο browser διαβάζει την HTML και μεταφράζει την Javascript. Ο κώδικάς της μπορεί να εκτελεσθεί αμέσως ή αργότερα όταν λάβει χώρα το συμβάν. Επίσης, προσφέρει στους σχεδιαστές ένα εργαλείο που έχει εντολές όπως οι `if`, `while`, `switch`, `for` κ.α. που δεν υπάρχουν στην HTML. Μπορεί να χρησιμοποιηθεί για επικύρωση δεδομένων σε φόρμες και να προσδιορίσουμε ζωντάνια στις ιστοσελίδες μας, όπως είναι για παράδειγμα:

- Να εμφανίσουμε κυλιόμενα μηνύματα στη γραμμή κατάστασης του browser
- Να κάνουμε υπολογισμούς με τα πεδία κειμένου μιας φόρμας
- Να δημιουργήσουμε εφέ κίνησης
- Να προσθέσουμε οριζόντια ή κάθετα μενού
- Να επικυρώσουμε τα περιεχόμενα των πεδίων μιας φόρμας
- Να εμφανίσουμε προειδοποιητικά μηνύματα στο χρήστη

Η εισαγωγή της Javascript γίνεται με το χαρακτηριστικό language ως εξής:

```
<script type = "text/javascript"> ή <script language = "Javascript">
```

Για να εμφανίσουμε κάποιο κείμενο σε μία σελίδα, το επιτυγχάνουμε με την εντολή `document.write`:
`document.write("Ναυτιλιακές εταιρίες")`

Παρακάτω φαίνεται ένα παράδειγμα Javascript μέσα σε ένα αρχείο HTML

```
<html>
<head>
<script type="text/javascript">
function message ()
{
    alert("Ναυτιλιακές εταιρίες")
}
</script>
</head>
<body onload="message ()">
</body>
</html>
```

Πίνακας 21: Παράδειγμα Javascript

Υπάρχουν δύο τρόποι για να εισάγουμε Javascript κώδικα μέσα σε ένα HTML αρχείο. Ο πρώτος είναι να τον εισάγουμε εσωτερικά στο αρχείο HTML και να γράψουμε javascript μέσα στην ετικέτα <body> ή <head>. Ο άλλος τρόπος είναι να γραφτεί σε ξεχωριστό αρχείο με κατάληξη .js. Τα εξωτερικά αρχεία δεν πρέπει να περιέχουν τις ετικέτες <script>, </script>.

3.8.3 Javascript vs Java

Η Javascript πολλές φορές συγχέεται με την Java, όμως είναι δύο διαφορετικές γλώσσες και δεν πρέπει να τις μπερδεύουμε. Μερικές από τις διαφορές τους αναφέρονται παρακάτω.

Java	Javascript
<ul style="list-style-type: none"> • Περίπλοκη στη χρήση • Απαιτεί το JDK • Τα προγράμματα μεταγλωττίζονται σε εκτελέσιμα αρχεία και ενσωματώνονται στις HTML σελίδες με την ετικέτα <APPLET>. Ο browser βλέπει την ετικέτα, φορτώνει και τρέχει το Java applet • Οι εφαρμογές είναι σχετικά αργές • Είναι μία πλήρης γλώσσα με πολλές δυνατότητες • Είναι προσανατολισμένη στο αντικείμενο και είναι μία καθαρά αντικειμενοστραφής γλώσσα • Έχει αυστηρούς κανόνες στη χρήση των μεταβλητών 	<ul style="list-style-type: none"> • Σχετικά εύκολη στη χρήση • Χρειάζεται μόνο ένας browser συμβατός με Javascript • Τα προγράμματα ενσωματώνονται στις HTML σελίδες με τη μορφή σεναρίου και δεν χρειάζονται μεταγλώττιση ή διερμηνευση • Εκτελείτε πολύ γρήγορα • Είναι κατάλληλη για πιο απλές εφαρμογές • Είναι βασισμένη στο αντικείμενο αλλά δεν είναι καθαρά αντικειμενοστραφής γλώσσα • Χαρακτηρίζεται από χαλαρότητα ως προς τις μεταβλητές της

Πίνακας 22: Java vs Javascript

3.8 CSS



Εικόνα 22: Λογότυπο του CSS

3.8.1 Τι είναι τα CSS

Το Cascading Style Sheets (διαδοχικά φύλλα στυλ), είναι ένα πολύ καλό εργαλείο για να μπορούμε να αλλάζουμε την εμφάνιση και διάταξη των ιστοσελίδων μας. Είναι μία απλή γλώσσα που μας βοηθάει να ορίζουμε με σαφήνεια και εύκολο τρόπο το πώς θα εμφανίζεται η μορφοποίηση της ιστοσελίδας. Με τη χρήση τους μπορούν να μας γλυτώσουν από πολύ χρόνο και κόπο και μας δίνουν τη δυνατότητα να σχεδιάζουμε τις ιστοσελίδες μας με μια εντελώς καινούργια φιλοσοφία. Η κατανόηση των CSS απαιτεί να υπάρχει κάποια βασική εμπειρία με την HTML.

Για να δουλέψουμε με τα CSS δεν χρειάζεται να χρησιμοποιήσουμε κάποιο πρόγραμμα δημιουργίας ιστοσελίδων, όπως είναι τα γνωστά FrontPage, DreamWeaver ή και το Word, καθώς θα σταθούν εμπόδιο στην κατανόηση των CSS. Αυτό που χρειαζόμαστε είναι ένας απλός επεξεργαστής κειμένου, όπως είναι το Σημειωματάριο (Notepad) των Windows.

3.8.2 Διαφορές CSS-HTML

Η HTML χρησιμοποιείται για να δομήσει το περιεχόμενο (content), ενώ τα CSS χρησιμοποιούνται για τη διαμόρφωση ή μορφοποίηση (formatting) του δομημένου περιεχομένου. Από το ξεκίνημά της, η HTML χρησιμοποιείτο μόνο για να μπορούμε να προσθέτουμε δομή στο κείμενο, όπως για να επισημαίνουμε ποια είναι κεφαλίδα (headline) ή ποια είναι παράγραφος (paragraph) με τα γνωστά tags (ετικέτες) της HTML, όπως είναι τα `<h1>` και `<p>`.

Καθώς όμως η δημοτικότητα του Web αύξανε συνεχώς, οι σχεδιαστές των ιστοσελίδων άρχισαν να ψάχνουν για επιπλέον δυνατότητες, όπως προσθήκη διάταξης (layout) στα έγγραφα. Για να γίνει αυτό, οι browser επινόησαν καινούργια HTML tags, όπως για παράδειγμα το ``, τα οποία διέφεραν από τα αρχικά HTML tags καθώς όριζαν τη διάταξη και όχι τη δομή μιας ιστοσελίδας.

Τα CSS είναι μια γλώσσα στυλ (style language) που ορίζουν τη διάταξη (layout) των HTML αρχείων. Τα CSS για παράδειγμα, έχουν να κάνουν με γραμματοσειρές (fonts), με χρώματα (colours), με περιθώρια (margins), με εικόνες φόντου (background images) και με πολλά άλλα. Με την HTML θα δυσκολευτούμε να αλλάξουμε τη διάταξη των ιστοσελίδων μας, αλλά τα CSS προσφέρουν πολλές επιλογές και είναι πολύ πιο συγκεκριμένα στις λεπτομέρειες. Επιπλέον, υποστηρίζονται απ' όλους τους browser. Έχουν μικρότερο μέγεθος αρχείου, δεδομένου ότι ο κάθε κανόνας μορφοποίησης γράφεται μόνο μία φορά και όχι κάθε φορά που χρησιμοποιείται. Η συντήρηση των ιστοσελίδων γίνεται ευκολότερη και μπορεί να ελέγχεται από ένα μόνο εξωτερικό αρχείο CSS. Έτσι, κάθε αλλαγή στο στυλ της ιστοσελίδας μπορεί να γίνεται με μια μοναδική αλλαγή σε αυτό το αρχείο, αντί για την επεξεργασία πολλών σημείων σε κάθε σελίδα που υπάρχει στην ιστοσελίδα μας. Επίσης, είναι το καλύτερο SEO (Search engine optimization), κάτι που σημαίνει πως οι μηχανές αναζήτησης δεν

μπερδεύονται ανάμεσα σε περιεχόμενο και τη μορφοποίηση του, αλλά έχουν πρόσβαση στο περιεχόμενο σκέτο, οπότε είναι πολύ ευκολότερο να το καταγράψουν και να το αρχειοθετήσουν. Όταν χρησιμοποιούμε εξωτερικό αρχείο CSS, ο browser την πρώτη φορά που θα φορτώσει κάποια σελίδα του site μας το αποθηκεύει στην cache, οπότε δεν χρειάζεται να το κατεβάσει ξανά κάθε φορά που κατεβάζει ο χρήστης του κάποια άλλη σελίδα του site μας.

Τα μειονεκτήματά που έχει αποφέρει είναι η ανάγκη εκμάθησης μίας ακόμα γλώσσας προγραμματισμού, ενώ δεν υποστηρίζονται από όλους τους browser το ίδιο καλά.

3.8.2 Πως λειτουργούν

Τα CSS στην HTML 4.0 ορίζουν το πώς εμφανίζονται τα στοιχεία αυτά, όπως ακριβώς το tag font και το attribute color στην HTML 3.2. Τα στυλ αποθηκεύονται συνήθως σε αρχεία που είναι εξωτερικά (external) στα HTML έγγραφα μας. Τα εξωτερικά φύλλα στυλ (external style sheets) μάς δίνουν τη δυνατότητα να αλλάξουμε την εμφάνιση και τη διάταξη όλων των σελίδων στο δικό μας Web site, με απλή επεξεργασία ενός μόνου CSS εγγράφου.

Τα CSS επιτρέπουν τον ορισμό των πληροφοριών στυλ με πολλούς τρόπους. Τα στυλ μπορούν να ορισθούν μέσα σ' ένα μόνο HTML στοιχείο, μέσα στο στοιχείο <head> μιας HTML σελίδας ή σε ένα εξωτερικό αρχείο CSS. Πολλά εξωτερικά CSS μπορούν να χρησιμοποιούνται μέσα απ' ένα μόνο HTML έγγραφο. Σε γενικές γραμμές μπορούμε να πούμε ότι όλα τα στυλ θα καταλήξουν (cascade) σ' ένα νέο εικονικό (virtual) CSS σύμφωνα με τους παρακάτω κανόνες, όπου ο μεγαλύτερος αριθμός έχει και την υψηλότερη προτεραιότητα. Επίσης παρουσιάζεται ένας κώδικας που δείχνει πως γράφεται σε κάθε μία από τις περιπτώσεις:

- Προεπιλογή του browser. Ο browser είναι αυτός που θα αποφασίσει πως θα απεικονίσει τα διάφορα στυλ που υπάρχουν στην ιστοσελίδα.
- Εξωτερικό Φύλλο Στυλ (External Style Sheet).

```
<html>
<head>
<title>Shipping Companies</title>
<meta content="text/html; charset=iso-8859-7" />
<link type="text/css" rel="stylesheet" href="/style.css"/>
</head>

<body>
.....
</body>
</html>
```

Πίνακας 23: External Style Sheet

- Εσωτερικό Φύλλο Στυλ (Internal Style Sheet), μέσα στο τμήμα header του εγγράφου.

```
<html>
<head>
<title>Shipping Companies</title>
<meta content="text/html; charset=iso-8859-7" />

<style type="text/css">
body {
    background-color: blue;
}
</style>
</head>

<body>
.....
</body>
</html>
```


Πίνακας 24: Internal Style Sheet

- Inline Style, μέσα στο HTML στοιχείο.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-trans
<html>
<head>
<meta content="text/html; charset=utf-8" />
<title>Shipping Companies</title>
</head>
<body>


Η εικόνα ul εισάγει μια μη αριθμημένη λίστα στην σελίδα μας. Το ul βγαίνει από το <em>Unordered List</em>.
Η εικόνα li προσθέτει γραμμές στην λίστα. Η ιδιότητα
<em style="FONT-SIZE: 12px; FONT-FAMILY: Arial, Helvetica, sans-serif; COLOR: #228B22; FONT-WEIGHT: bold;
TEXT-DECORATION: none;">type</em> της εικόνας
ul ορίζει το σύμβολο που μπαίνει μπροστά από κάθε γραμμή της λίστας. Οι τιμές που παίρνει η ιδιότητα
<em style="FONT-SIZE: 12px; FONT-FAMILY: Arial, Helvetica, sans-serif; COLOR: #228B22; FONT-WEIGHT: bold;
TEXT-DECORATION: none;">type</em> είναι οι εξής:
<em style="FONT-SIZE: 12px; FONT-FAMILY: Arial, Helvetica, sans-serif; COLOR: #FFA07A; FONT-WEIGHT: bold;
TEXT-DECORATION: none;">disc</em>,
<em style="FONT-SIZE: 12px; FONT-FAMILY: Arial, Helvetica, sans-serif; COLOR: #FFA07A; FONT-WEIGHT: bold;
TEXT-DECORATION: none;">circle</em>,
<em style="FONT-SIZE: 12px; FONT-FAMILY: Arial, Helvetica, sans-serif; COLOR: #FFA07A; FONT-WEIGHT: bold;
TEXT-DECORATION: none;">square</em>
</body>
</head>
</html>


```

Πίνακας 25: Inline Style Sheet

Μερικές από τις βασικές ιδιότητες που αφορούν τη μορφοποίηση κειμένου είναι

Color	Αφορά το χρώμα του κειμένου
Font-size	Αφορά το μέγεθος της γραμματοσειράς
Font-family	Ορίζει πλήθος γραμματοσειρών
Font-style	Αφορά το κείμενο όπου είναι πλάγιο
Font-weight	Αφορά αν το κείμενο θα είναι έντονο
Text-decoration	Εφέ (none, underline, line-through, blink)
Text-align	Στοιχηση του κειμένου
Font	Αφορά τις ιδιότητες του κειμένου
Background	Αφορά το φόντο
Border	Αφορά το περίγραμμα

Πίνακας 26: Βασικές ιδιότητες

Παρακάτω θα αναφέρουμε περιληπτικά μερικούς CSS selectors.

*	Οι ιδιότητες θα υιοθετηθούν σε κάθε στοιχείο της ιστοσελίδας
στοιχείο	Οι ιδιότητες θα υιοθετηθούν μόνο από το συγκεκριμένο στοιχείο
.όνομα_κλάσης	Οι ιδιότητες θα υιοθετηθούν στα στοιχεία της κλάσης «όνομα_κλάσης»
στοιχείο.όνομα_κλάσης	Εφαρμόζεται σε όσα στοιχεία αποτελούνται από το html tag <στοιχείο> και ανήκουν στην κλάση όνομα_κλάσης
#όνομα_id	Ο browser εφαρμόζει τις ιδιότητες που θα γράψουμε στο στοιχείο το οποίο περιλαμβάνει την ιδιότητα id="όνομα_id".
στοιχείο[attribute="value"]	Ο browser θα εφαρμόζει τις ιδιότητες που θα γράψουμε σε αυτόν σε κάθε στοιχείο με tag <στοιχείο> το οποίο επιπροσθέτως έχει την τιμή value στην html ιδιότητα attribute
επιλογέα1,επιλογέα2,επιλογέα3	Οι ιδιότητες που θα γράψουμε σε αυτόν τον κανόνα CSS, θα εφαρμοστούν σε κάθε στοιχείο που πληροί τις προϋποθέσεις επιλογής είτε για τον επιλογέα1, είτε για τον επιλογέα2, είτε για τον επιλογέα3 κοκ

Πίνακας 27: CSS Επιλογείς

4. ΘΕΩΡΗΤΙΚΟ ΜΕΡΟΣ ΤΗΣ ΠΤΥΧΙΑΚΗΣ

4.1 Ανάλυση του προβλήματος

Με τον όρο Πρόβλημα εννοείτε μία κατάσταση η οποία χρήζει αντιμετώπισης, απαιτεί λύση, η λύση της οποίας δεν είναι ούτε γνωστή ούτε προφανής. Στην παρούσα πτυχιακή το πρόβλημα είναι η δημιουργία μίας διαδικτυακής εφαρμογής και πιο συγκεκριμένα μία δυναμική διαδικτυακή εφαρμογή κρατήσεων ακτοπλοϊκών εισιτηρίων. Σε κάθε πρόβλημα υπάρχουν κάποια δεδομένα για την ύπαρξη της λύσης του. Δεδομένα είναι τα στοιχεία εκείνα του προβλήματος που θεωρούνται γνωστά και με βάση αυτά θα δώσουμε λύση στο πρόβλημα. Τα δεδομένα για την παρούσα πτυχιακή είναι:

- Δημιουργία μίας βάσης δεδομένων που θα κρατάει πληροφορίες για,
 - Τα δρομολόγια που θα πραγματοποιούνται
 - Τους πελάτες μας
 - Τις ναυτιλιακές εταιρίες
 - Τους χρήστες που θα κάνουν login
 - Τους υπαλλήλους των πλοίων
- Τήρηση των κανόνων ακεραιότητας
 - Ένα πλοίο δεν μπορεί να ταξιδεύει σε 2 προορισμούς ταυτόχρονα
 - Ένας πλοίαρχος δεν μπορεί να βρίσκεται σε 2 δρομολόγια με ίδιες ώρες αναχώρησης
 - Ένας επιβάτης δεν μπορεί να κλείσει 2 ή περισσότερα εισιτήρια για δρομολόγια των οποίων οι ώρες συμπίπτουν
- Πληροφορίες σχετικά με τους χρήστες
 - Ο Διαχειριστής μπορεί να κάνει προσθαφαίρεση εταιριών, δρομολογίων, πλοιαρχων, δρομολογίων, τοποθεσιών
 - Ο Ταξιδιωτικός Πράκτορας μπορεί να κάνει κράτηση και ακύρωση εισιτηρίου στο όνομα κάποιου πελάτη του αλλά όχι άλλων πρακτόρων
 - Ο Χρήστης-Πελάτης θα πρέπει να κάνει login με username και password και θα μπορεί να κλείσει εισιτήριο για τον εαυτό του

4.2 Σχεδιασμός Υλοποίησης

- Σχεδίαση βάσεων δεδομένων. Πως αυτές λειτουργούν και πως κατασκευάζονται. Λεπτομερή αναζήτηση σε βιβλία της σχολής και σε πολλές ιστοσελίδες
- Προγραμματισμός διαδικτύου με τις ακόλουθες γλώσσες προγραμματισμού PHP, HTML, JavaScript. Αναζήτηση πληροφοριών στο internet και στο forum της σχολής.
- Εκμάθηση της MySQL. Έρευνα στο διαδίκτυο.
- Σχεδίαση της διεπαφής χρήστη με CSS. Σχετική έρευνα στο διαδίκτυο.
- Ύπαρξη ενός server, στην προκειμένη περίπτωση θα χρησιμοποιηθεί ένας localhost και αυτό θα γίνει με το εργαλείο Xampp. Εκ των προτέρων γνωστό, λόγο διαχείρισής του παλαιότερα.

5. ΠΡΑΚΤΙΚΟ ΜΕΡΟΣ ΤΗΣ ΠΤΥΧΙΑΚΗΣ

Σε αυτό το κεφάλαιο θα αναλύσουμε τη διαδικασία του πρακτικού μέρους της πτυχιακής εργασίας. Θα αναφερθούμε στις διαδικασίες εγκατάστασης των προγραμμάτων και της υλοποίησης της βάσης δεδομένων.

5.1 Εγκατάσταση απαραίτητου λογισμικού



Εικόνα 23: Εγκατάσταση λογισμικού

5.1.1 Εγκατάσταση του Xampp

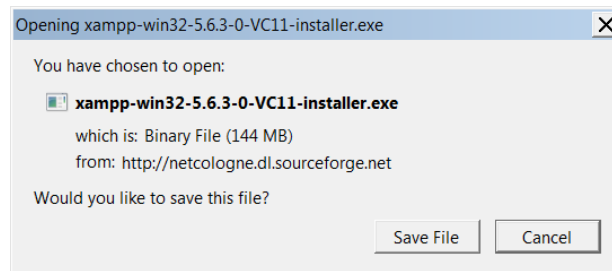
Για την εγκατάσταση του Xampp και των εργαλείων που θα συμπεριληφθούν με αυτό θα ανατρέξουμε στη σελίδα του κατασκευαστή του συγκεκριμένου προγράμματος που φιλοξενείται από την apachefriends. Ανάλογα με το λειτουργικό σύστημα που χρησιμοποιεί ο χρήστης, επιλέγουμε ποιο εργαλείο Xampp θα κατεβάσουμε.



The screenshot shows the Apache Friends website's XAMPP download page. At the top, there is a navigation menu with links for 'Download', 'Add-ons', 'Hosting', 'Community', and 'About'. A search bar is also present. The main heading is 'XAMPP Apache + MySQL + PHP + Perl'. Below this, there is a section titled 'What is XAMPP?' which describes XAMPP as a popular PHP development environment. To the right of this text is a video player showing the XAMPP logo. At the bottom, there is a large green 'Download' button with the text 'Click here for other versions'. Below this are three buttons for downloading XAMPP for different operating systems: Windows, Linux, and OS X, all version 5.6.3 (PHP 5.6.3).

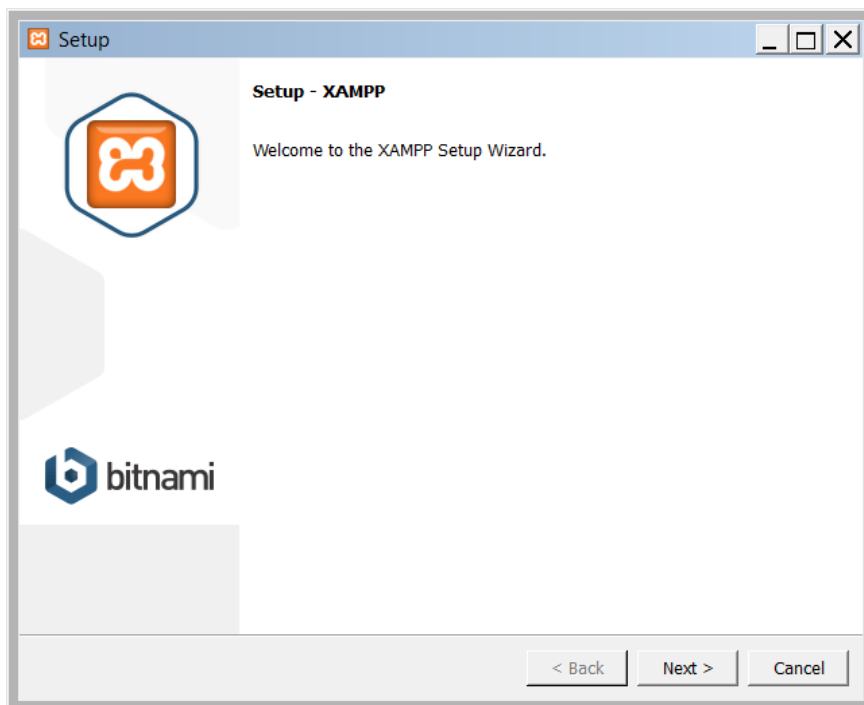
Εικόνα 24: Download Xampp

Σώζουμε το αρχείο που ανοίγει στο νέο παράθυρο.



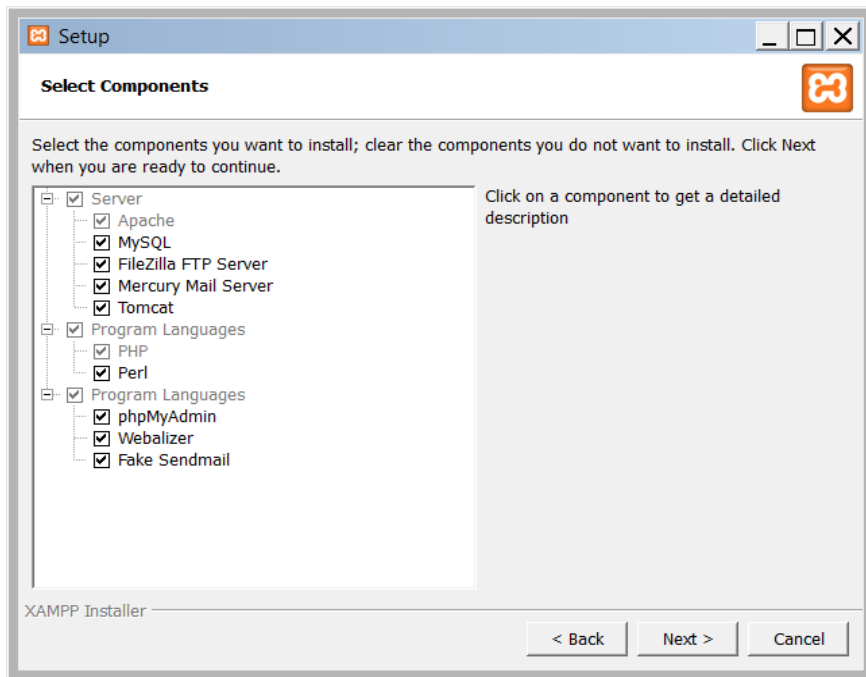
Εικόνα 25: Save Xampp file

Μόλις ολοκληρωθεί η λήψη, τρέχουμε το αρχείο.



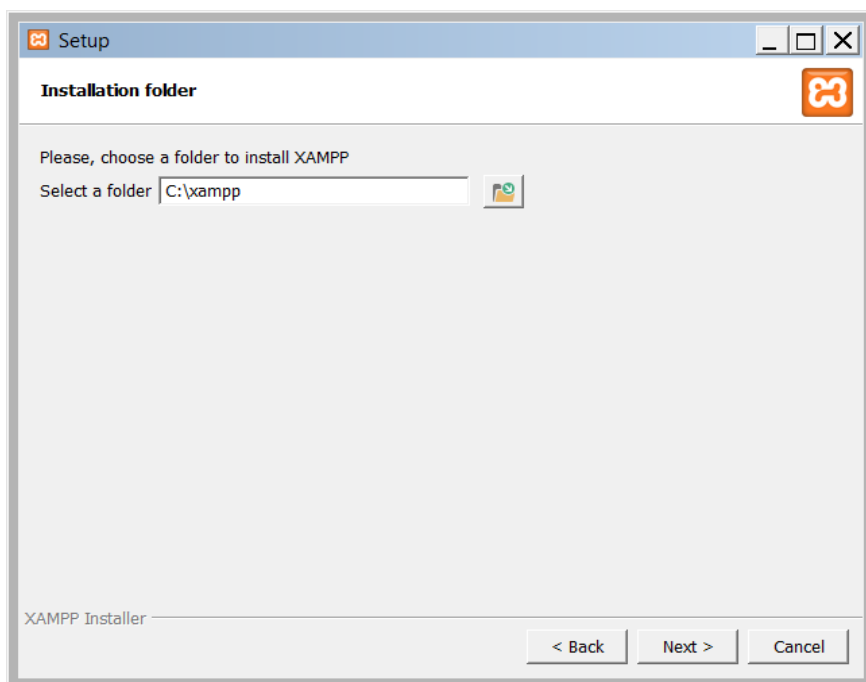
Εικόνα 26: Setup Xampp 1

Όπως βλέπουμε στην παρακάτω εικόνα, έχουμε τη δυνατότητα να εγκαταστήσουμε οποιοδήποτε από τα παρακάτω components μας ενδιαφέρουν. Ο Apache server και η PHP είναι μη διαχειρίσιμα πεδία. Δεν μπορούμε να τα ξετικάρουμε και ο λόγος είναι ότι το Xampp δημιουργήθηκε λόγω του ότι ο Apache server και η PHP συνεργάζονται εξαιρετικά και είναι δύο εργαλεία που ανήκουν στο Open Source λογισμικό.



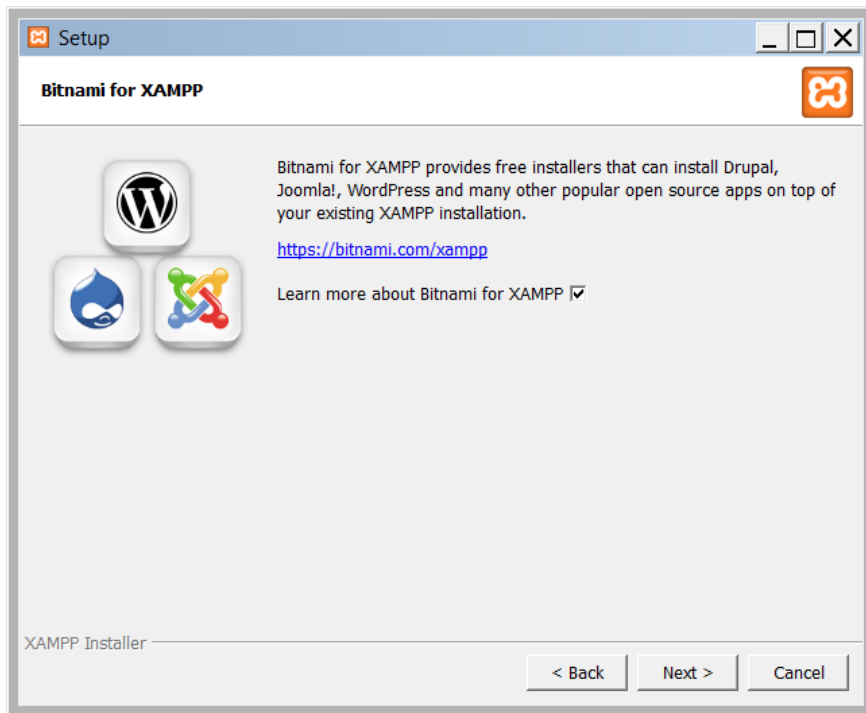
Εικόνα 27: Setup Xampp 2

Επιλέγουμε τον φάκελο που θα γίνει η εγκατάσταση του Xampp. Θεμιτό είναι να μην αλλάξουμε το default folder.



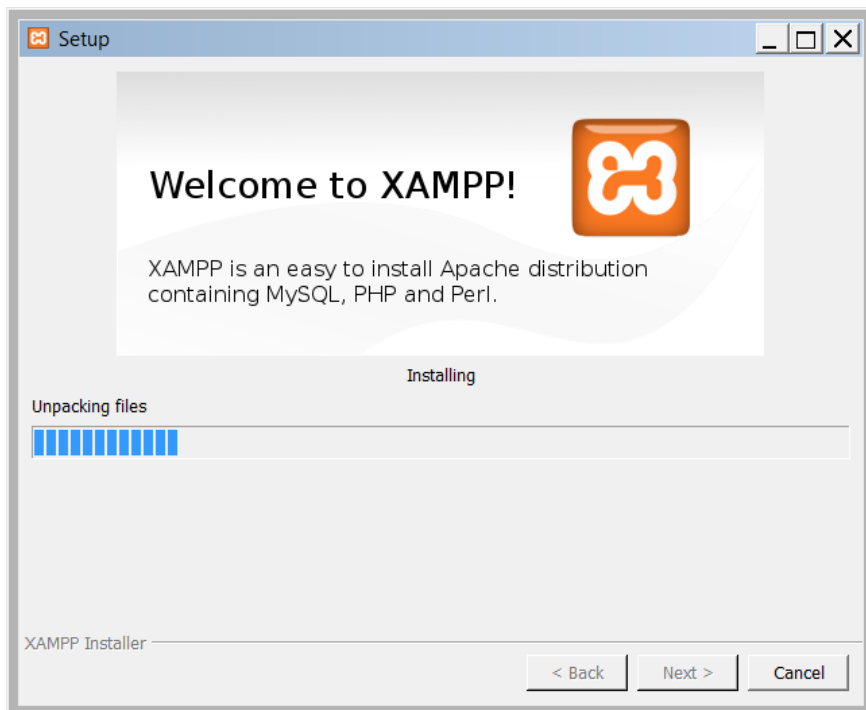
Εικόνα 28: Setup Xampp 3

Πατάμε next



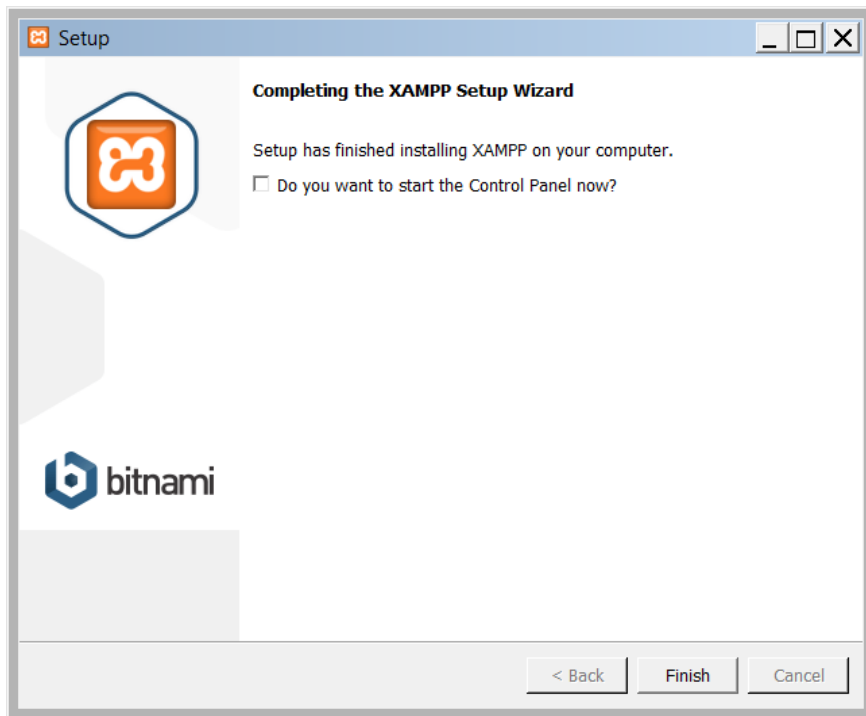
Εικόνα 29: Setup Xampp 4

Άρχισε η εγκατάσταση του Xampp στον υπολογιστή μας.



Εικόνα 30: Setup Xampp 5

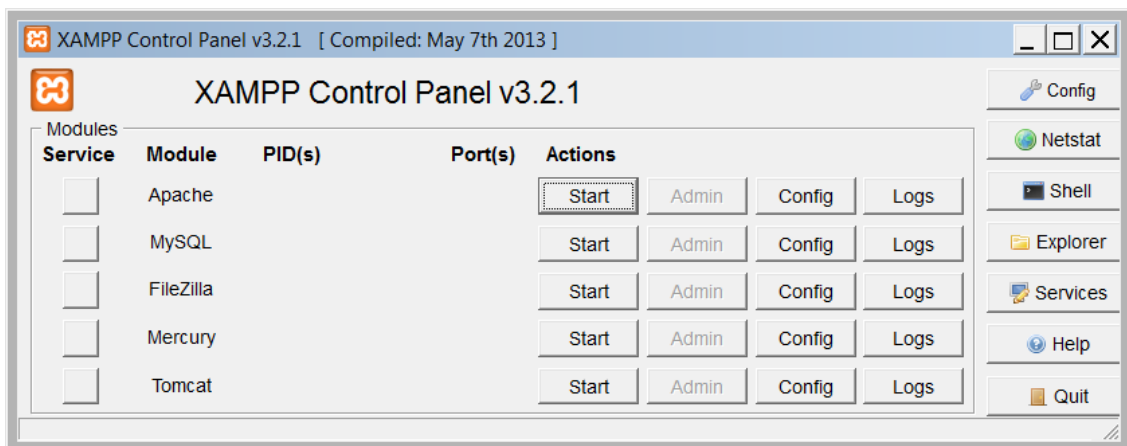
Πατάμε Finish για να ολοκληρωθεί η εγκατάσταση.



Εικόνα 31: Finish Xampp installation

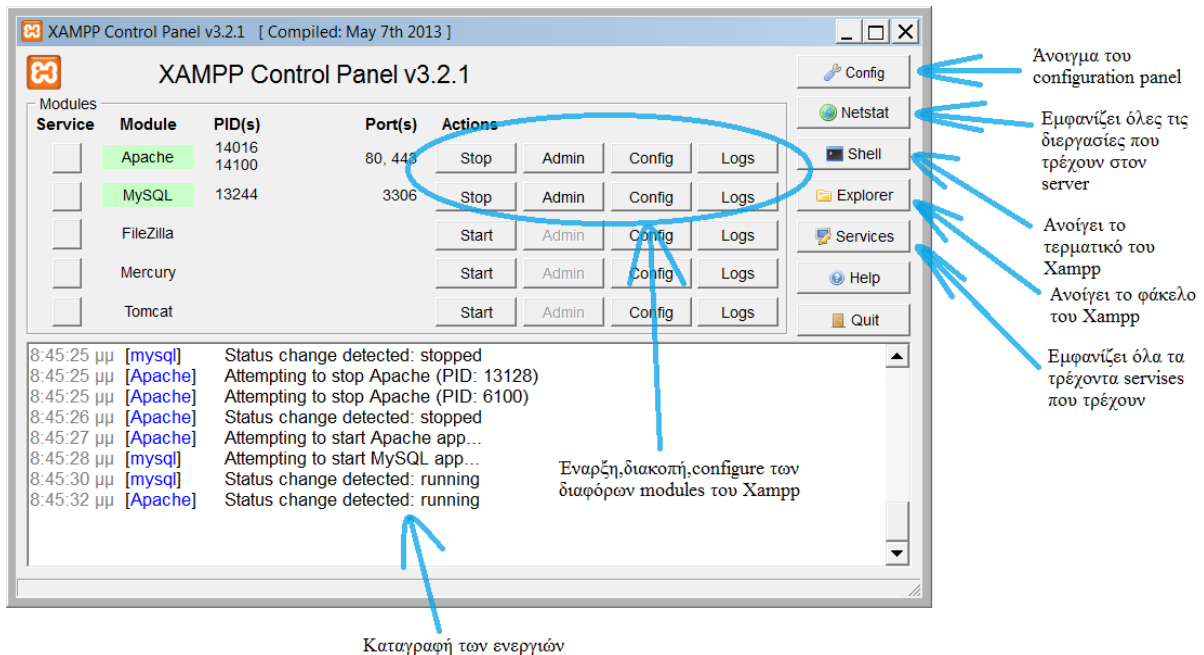
5.1.2 Ρυθμίσεις Xampp

Ανοίγουμε το Xampp, είτε από τη γραμμή εργασιών, είτε από το εικονίδιο που δημιουργήθηκε στην επιφάνια εργασίας μας. Σημαντική σημείωση που πρέπει να κάνω, για να λειτουργήσουν σωστά όλες οι λειτουργίες του Xampp θα πρέπει να έχουμε δικαιώματα διαχειριστή και όχι απλού χρήστη.



Εικόνα 32: Xampp control panel 1

Κατόπιν πατάμε το κουμπί Start του Apache και το κουμπί Start της MYSQL. Περιμένουμε έως ότου πρασινίσουν και τα δύο modules. Μόλις γίνει αυτό, ο τοπικός μας server είναι έτοιμος για χρήση.



Εικόνα 33: Xampp control panel 2

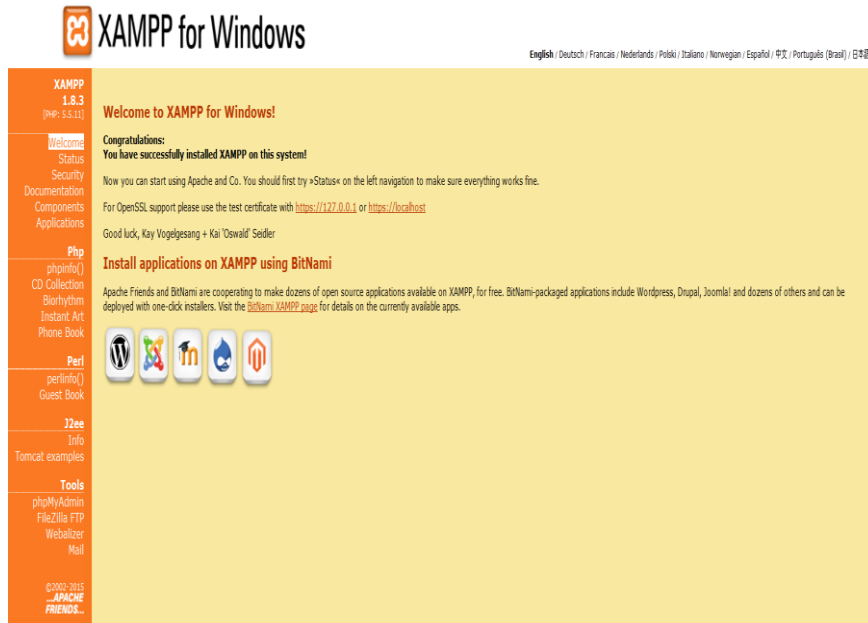
Επόμενο βήμα είναι να ανοίξουμε τον browser που χρησιμοποιούμε (επειδή ο firefox μου παρουσίαζε κάποια λάθη με τις γραμματισειρές, χρησιμοποίησα τον Internet Explorer) και να γράψουμε την διεύθυνση <http://localhost>. Κλικάρουμε την γλώσσα που επιθυμούμε και μεταφερόμαστε στην κεντρική σελίδα του Xampp.



[English](#) / [Deutsch](#) / [Francais](#) / [Nederlands](#) / [Polski](#) / [Italiano](#) / [Norwegian](#) / [Español](#) / [中文](#) / [Português \(Brasil\)](#) / [日本語](#)

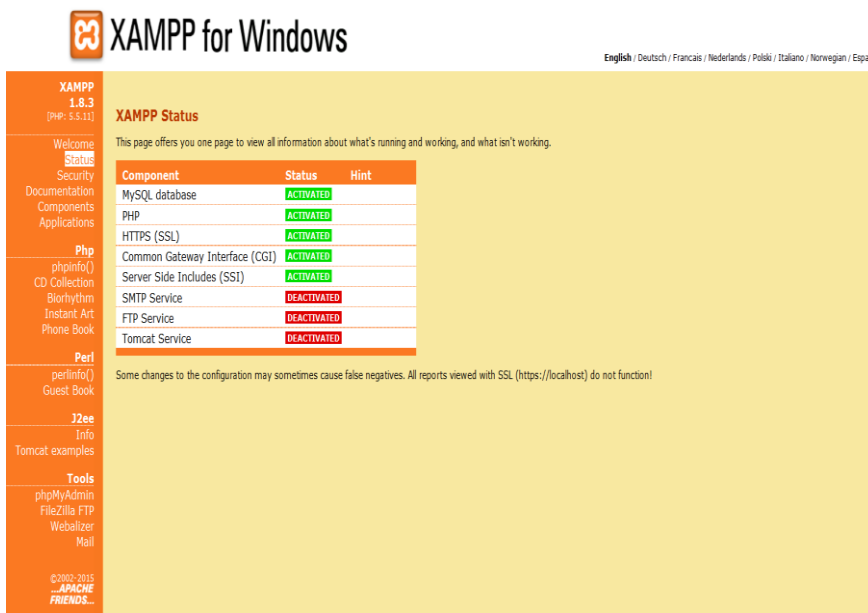
Εικόνα 34: Xampp Admin Apache 1

Εδώ μπορούμε να αναζητήσουμε πληροφορίες για τις εκδόσεις των λογισμικών που έχουμε εγκατεστημένους στον υπολογιστή μας, έγγραφα σχετικά με τη χρήση αυτών των λογισμικών και διάφορα ακόμα. Στην καρτέλα εργαλεία, βλέπουμε μία επιλογή που λέγεται phpMyAdmin. Αυτό είναι το εργαλείο που θα μας βοηθήσει να δημιουργήσουμε τη βάση δεδομένων μας (θα αναφερθούμε εκτενέστερα για αυτό παρακάτω).



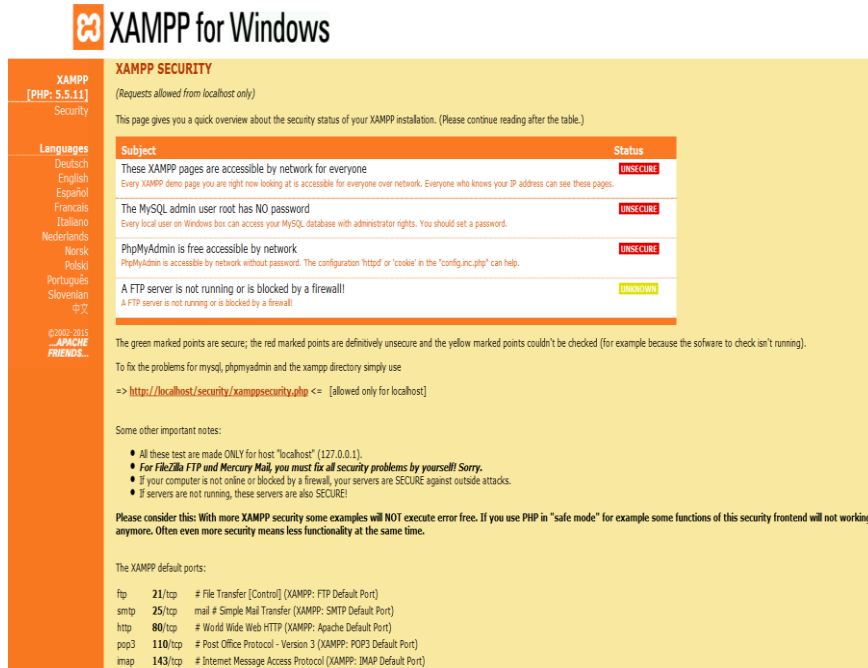
Εικόνα 35: Xampp Admin Apache 2

Σημαντική παρατήρηση ότι θα πρέπει να ελέγξουμε την καρτέλα Status, τα components να είναι activated, όπως παρουσιάζονται στην παρακάτω εικόνα.



Εικόνα 36: Xampp Admin Apache 3

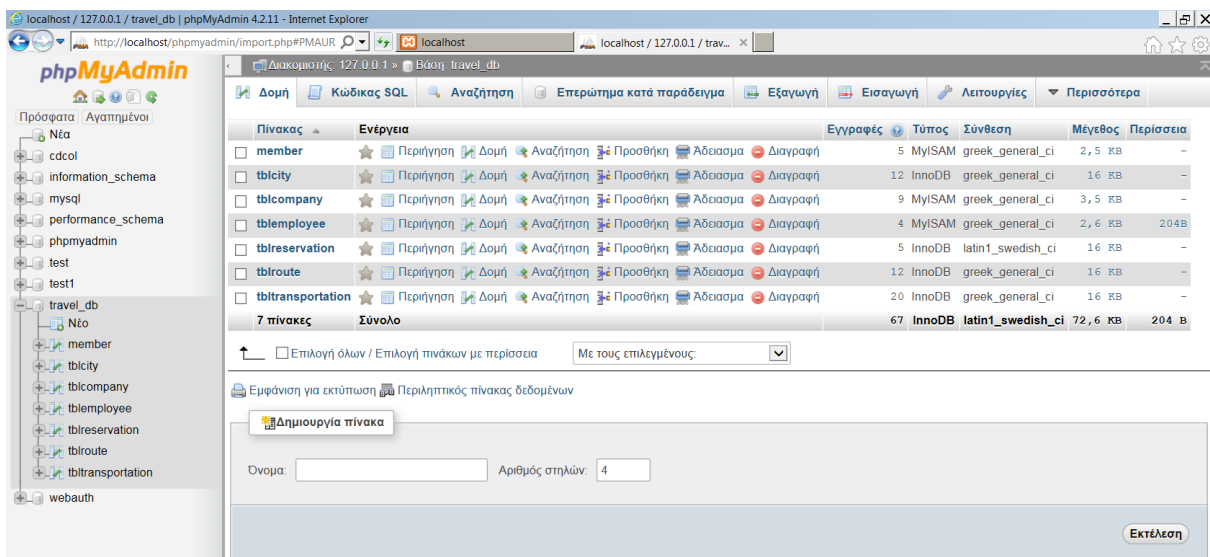
Στην καρτέλα Security, μπορούμε να δούμε κάποια πιθανά σφάλματα και κενά ασφαλείας, σχετικά με τις υπηρεσίες τρέχουν στον server μας. Στη δική μου περίπτωση μου αναφέρει ότι υπάρχουν κάποια μη ασφαλή σημεία, όπου μπορεί κάποιος να πάρει τον έλεγχο του μηχανήματός μου. Μου αναφέρει αναλυτικά ότι, 1) οποιοσδήποτε μπορεί να δει τις σελίδες που έχουμε φορτώσει στον server, αρκεί να γνωρίζει την IP μου, 2) πρέπει να βάλω κωδικό στη MySQL για τον administrator, 3) πρέπει να βάλω κωδικό για το phpMyAdmin και 4) με ενημερώνει ότι το FTP δεν τρέχει ή έχει μπλοκαριστεί.



Εικόνα 37: Xampp Admin Apache 4

5.2 Σχεδίαση Βάσης Δεδομένων

Όπως έχουμε αναφέρει η βάση δεδομένων είναι υλοποιημένη σε MySQL. Στην παρακάτω εικόνα φαίνεται η βάση δεδομένων, έτσι όπως δημιουργήθηκε με το phpMyAdmin.



Εικόνα 38: Βάση δεδομένων εφαρμογής

Τα στοιχεία που είναι απαραίτητα για την υλοποίηση του συστήματος αφορούν τις επιχειρήσεις (ναυτιλιακές εταιρίες, πλοία κα), τις πόλεις, τους προορισμούς και τα δρομολόγια μεταξύ αυτών. Επίσης αφορά και τους χρήστες του συστήματος. Όλα αυτά αποτελούν οντότητες του συστήματος και θα διατηρούνται σε μια σχεσιακή βάση δεδομένων. Με βάση την ανάλυση των δεδομένων, οι πληροφορίες καταχωρούνται σε αντίστοιχους πίνακες στη βάση δεδομένων της MySQL. Παρακάτω απεικονίζονται οι βασικοί πίνακες, ο κώδικας με τον οποίο μπορούμε να τους δημιουργήσουμε και τροποποιήσουμε (να προσθέσουμε τιμές στα διάφορα πεδία) έτσι όπως φαίνονται μέσα από το MySql Workbrench:

member (Μέλη)

Για κάθε έναν επισκέπτη που επιθυμεί να κάνει κράτηση υπάρχει ο πίνακας member, όπου ο επισκέπτης γίνεται εξουσιοδοτημένο μέλος και μπορεί να κάνει τις κρατήσεις εισιτηρίου και να τις διαχειριστεί. Επίσης, σε αυτόν τον πίνακα καταχωρούνται οι ταξιδιωτικοί πράκτορες και ο administrator. Παρατηρούμε ότι οι στήλες lname, email, address, phone, usertype είναι null, που σημαίνει ότι η τιμή δεν είναι γνωστή ή δεν υπάρχει τιμή.

Column Name	Data Type	Nullable	Default Value
id	INT(11)	NO	
pass	VARCHAR(250)	NO	
fname	VARCHAR(250)	NO	
lname	VARCHAR(250)	YES	
email	VARCHAR(250)	YES	
address	VARCHAR(250)	YES	
phone	VARCHAR(250)	YES	
usertype	VARCHAR(250)	YES	
username	VARCHAR(50)	NO	
cid	VARCHAR(10)	NO	

Πίνακας 28: Πίνακας member

```
CREATE TABLE IF NOT EXISTS `member` (
  `id` int(11) NOT NULL,
  `pass` varchar(250) NOT NULL,
  `fname` varchar(250) NOT NULL,
  `lname` varchar(250) DEFAULT NULL,
  `email` varchar(250) DEFAULT NULL,
  `address` varchar(250) DEFAULT NULL,
  `phone` varchar(250) DEFAULT NULL,
  `usertype` varchar(250) DEFAULT 'user',
  `username` varchar(50) NOT NULL,
  `cid` varchar(10) NOT NULL
) ENGINE=MyISAM AUTO_INCREMENT=24 DEFAULT CHARSET=greek;

INSERT INTO `member` (`id`, `pass`, `fname`, `lname`, `email`, `address`,
`phone`, `usertype`, `username`, `cid`) VALUES
(6, '12345', '', NULL, '', NULL, '', 'admintp', 'admin', ''),
(21, '12345', '', 'dafni', 'pozitronio85@hotmail.com', 'kalamonas', '6948589650',
'membertp', 'dafni', 'x124578'),
(22, '12345', '', 'Makarios Travel', 'makarios@makarios.gr', '0θωνα 23, Δράμα',
'2521094480', 'tradertp', 'Makarios', 'E345678'),
(20, '12345', '', 'dimos', 'pozitronio85@hotmail.com', 'kalamonas', '6948589650',
'membertp', 'pozitronio', 'x319191'),
(23, '12345', '', 'Tezas Tours', 'pozitronio85@hotmail.com', 'Βαρθολομαίου 12, Καλαμπάκι, Δράμα',
'6948589650', 'tradertp', 'Τέζας', 'P567890');
```

Εικόνα 39: member-Create and Insert Table

tblcity (Πόλη)

Στον πίνακα αυτό καταχωρούνται τα στοιχεία μιας πόλης όπως το όνομα, η περιγραφή και η γεωγραφική θέση.

```
CREATE TABLE IF NOT EXISTS `tblcity` (
  `cityid` int(11) NOT NULL,
  `cityname` varchar(250) NOT NULL,
  `priority` int(11) NOT NULL,
  `citybasic` int(11) NOT NULL DEFAULT '0',
  `citydetails` varchar(250) NOT NULL,
  `mapXY` varchar(250) NOT NULL,
  `cityzoom` int(11) NOT NULL
) ENGINE=InnoDB AUTO_INCREMENT=14 DEFAULT CHARSET=greek;
```

Column Name	Data Type	Nullable	Default Value
cityid	INT(11)	NO	
cityname	VARCHAR(250)	NO	
priority	INT(11)	NO	
citybasic	INT(11)	NO	0
citydetails	VARCHAR(250)	NO	
mapXY	VARCHAR(250)	NO	
cityzoom	INT(11)	NO	

Πίνακας 29: Πίνακας tblcity

```
INSERT INTO `tblcity` (`cityid`, `cityname`, `priority`, `citybasic`, `citydetails`, `mapXY`, `cityzoom`) VALUES
(1, 'Σκιάθος', 0, 1, 'Σκιάθος', '(39.46074698294581, 23.068495605468787)', 9),
(2, 'Ρόδος', 1, 1, 'Ρόδος', '(36.177740004867886, 27.023573730468787)', 0),
(3, 'Αλόνησος', 3, 1, 'Αλόνησος', '(39.13773569689827, 23.93881878662113)', 0),
(4, 'Χανιά', 4, 1, 'Χανιά', '(35.503112662908606, 24.12455639648441)', 0),
(5, 'Βόλος', 5, 1, 'Βόλος', '', 0),
(6, 'Χίος', 6, 1, 'Χίος', '(38.337298001113226, 26.43992504882816)', 0),
(7, 'Πειραιάς', 0, 1, 'Πειραιάς', '(37.26525959534969, 24.691725585937537)', 9),
(8, 'Σκόπελος', 0, 1, 'Σκόπελος', '(38.90616183374738, 23.66313061523441)', 9),
(10, 'Κω', 0, 1, 'Κω', '(36.87155295678894, 27.37650952148441)', 9),
(11, 'Ζάκυνθος', 0, 1, 'Ζάκυνθος', '(37.783819320347405, 20.873289916992224)', 9),
(12, 'Κέρκυρα', 0, 1, 'Κέρκυρα', '(39.56876935347532, 19.984083984375036)', 9),
(13, 'Ηράκλειο', 1, 1, 'Ηράκλειο', '(35.35462154695522, 25.016645143252834)', 9);
```

Εικόνα 40: tblcity-Create and Insert Table

tblcompany (Ναυτιλιακή εταιρία)

Στον πίνακα αυτό καταχωρούνται τα βασικά στοιχεία μιας επιχείρησης όπως επωνυμία, διεύθυνση κα.

```
CREATE TABLE IF NOT EXISTS `tblcompany` (
`id` int(11) NOT NULL,
`username` varchar(100) NOT NULL,
`url` varchar(200) NOT NULL,
`email` varchar(50) NOT NULL,
`phone` varchar(50) NOT NULL,
`description` text NOT NULL,
`photo` varchar(250) NOT NULL,
`title` varchar(250) NOT NULL,
`address` varchar(250) NOT NULL,
`pwd` varchar(20) NOT NULL
) ENGINE=MyISAM AUTO_INCREMENT=19 DEFAULT CHARSET=greek;
```



Πίνακας 30: Πίνακας tblcompany

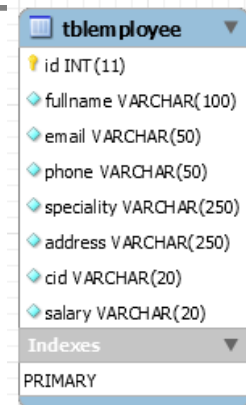
```
INSERT INTO `tblcompany` (`id`, `username`, `url`, `email`, `phone`, `description`, `photo`, `title`, `address`, `pwd`) VALUES
(1, 'Ανέκ', 'http://web.anek.gr/portal/page/portal/ANEK_prod', 'customerservice@anek.gr', '2322111225', 'Ανέκ', 'anek.jpg',
'Ανέκ Lines', 'Ποσειδώνος 10 Φαληρο, Πειραιας', ''),
(2, 'Blue Star', 'http://www.bluestarferries.gr/site/content.asp?loc=1', 'bluestarferries@bluestarferries.com', '2322111226',
'Blue Star', 'Blue-Star-Ferries.jpg', 'Blue Star Ferries', 'Ποσειδώνος 16 Φαληρο, Πειραιας', ''),
(12, 'Δωδεκάνησος', 'http://www.12ne.gr/el/', 'info@12ne.gr', '2241070590', 'Δωδεκάνησος', 'dodekanisos.jpg', 'Dodekanisow Seaways',
'Αυστραλίας 3, 85100 Ρόδος', ''),
(13, 'Fast Ferries', 'http://www.fastferries.com.gr/el/home-page', 'info@fastferries.com.gr', '2104182005', 'Fast Ferries',
'fast ferries.jpg', 'Fast Ferries A.E.', 'Σκουζέ 10, 18536, Πειραιάς', ''),
(14, 'Golden Star', 'http://goldenstarferries.gr/', 'general@goldenstarferries.gr', '212 222 4000', 'Golden Star',
'golden star ferries.jpg', 'Golden Star Ferries', 'Θήλωνος 133, 18536 Πειραιάς', ''),
(15, 'Hellenic', 'https://www.hellenicseaways.gr', 'customerservice@hsw.gr', '210 41 99 000', 'Hellenic', 'hellenic seaways.jpg',
'Hellenic Seaways', 'Αστρινγος 6, Πλατεία Καραϊσκάκη, 185 31 Πειραιάς', ''),
(16, 'Ionian', 'http://www.ionianferries.gr/', 'info@ionianferries.gr', '211 2215502', 'Ionian', 'ionian ferries.jpg',
'Ionian Ferries', 'Πειραιώς 1 (εντός στοάς) 105 52, Αθήνα', ''),
(17, 'Minoan', 'http://www.minoan.gr', 'info@minoan.gr', '2810 399800', 'Minoan', 'minoan.jpg', 'Minoan Lines',
'25ης Αυγούστου 17, 712 02 Ηράκλειο Κρήτης', ''),
(18, 'SuperFast', 'http://www.superfast.com', 'info.heraklion@attica-group.com', '210 8919700', 'SuperFast', 'superfast.jpg',
'SuperFast Ferries', 'Λεωφ. Συγγρού 123-125 και Τορβά 3', '');
```

Εικόνα 41: tblcompany-Create and Insert Table

tblemployee (Υπάλληλος ναυτιλιακής εταιρίας)

Στον πίνακα αυτό καταχωρούνται τα βασικά στοιχεία ενός υπαλλήλου όπως επωνυμία, διεύθυνση κα.

```
CREATE TABLE IF NOT EXISTS `tblemployee` (
`id` int(11) NOT NULL,
`fullname` varchar(100) NOT NULL,
`email` varchar(50) NOT NULL,
`phone` varchar(50) NOT NULL,
`speciality` varchar(250) NOT NULL,
`address` varchar(250) NOT NULL,
`cid` varchar(20) NOT NULL,
`salary` varchar(20) NOT NULL
) ENGINE=MyISAM AUTO_INCREMENT=20 DEFAULT CHARSET=greek;
```



Πίνακας 31: Πίνακας tblemployee

```
INSERT INTO `tblemployee` (`id`, `fullname`, `email`, `phone`, `speciality`, `address`, `cid`, `salary`) VALUES
(12, 'Καρασατεφάνης Παύλος', 'pozitronio85@hotmail.com', 'sdfgsdfg6948589650', 'Πλοίαρχος', 'Χαριτσίου 12', 'AA121213', '2000'),
(15, 'Μαχαϊρίδης Ιωάννης', 'pozitronio85@hotmail.com', '6948589650', 'Πλοίαρχος', 'Τούμπας 09', 'X344334', '1500'),
(16, 'Τζανής Απόστολος', 'pozitronio85@hotmail.com', '6948589650', 'Πλοίαρχος', 'Πειραιά 07', 'Σ304059', '1700'),
(19, 'Μανωλάς Αντρέας', 'pozitronio85@hotmail.com', '6948589650', 'Πλοίαρχος', 'Σαλαμίνας 26', 'M483828', '1600');
```

Εικόνα 42: tblemployee-Create and Insert Table

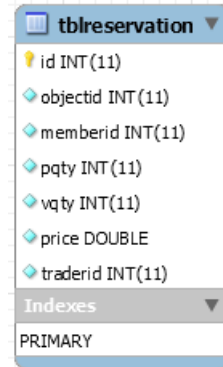
tbltblreservation. (Κρατήσεις εισιτηρίων)

Στον πίνακα αυτό αποθηκεύονται οι κρατήσεις εισιτηρίων και στοιχεία όπως η τιμή εισιτηρίου, ποιος έκανε την κράτηση, τόπος αναχώρησης, τόπος άφιξης κα.

```
CREATE TABLE IF NOT EXISTS `tblreservation` (
`id` int(11) NOT NULL,
`objectid` int(11) NOT NULL,
`memberid` int(11) NOT NULL,
`pqty` int(11) NOT NULL,
`vqty` int(11) NOT NULL,
`price` double NOT NULL,
`traderid` int(11) NOT NULL DEFAULT '0'
) ENGINE=InnoDB AUTO_INCREMENT=36 DEFAULT CHARSET=latin1;
```

```
INSERT INTO `tblreservation` (`id`, `objectid`, `memberid`, `pqty`, `vqty`, `price`, `traderid`) VALUES
(14, 11, 20, 1, 0, 50, 0),
(15, 16, 20, 1, 0, 28, 0),
(25, 21, 20, 1, 0, 60, 0),
(34, 20, 20, 1, 0, 50, 0),
(35, 20, 0, 20, 0, 1000, 22);
```

Εικόνα 43: tblreservation-Create and Insert Table



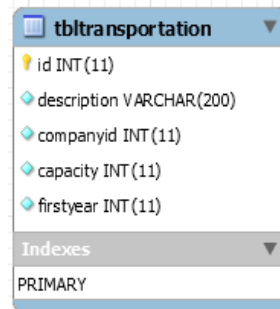
Πίνακας 32: Πίνακας tblreservation

tbltransportation (Κατηγορία μεταφορικού μέσου)

Στον πίνακα καταχωρούνται από το Διαχειριστή τα πλοία με τα διάφορα χαρακτηριστικά τους.

```
CREATE TABLE IF NOT EXISTS `tbltransportation` (
`id` int(11) NOT NULL,
`description` varchar(200) NOT NULL,
`companyid` int(11) NOT NULL,
`capacity` int(11) NOT NULL,
`firstyear` int(11) NOT NULL
) ENGINE=InnoDB AUTO_INCREMENT=25 DEFAULT CHARSET=greek;
```

```
INSERT INTO `tbltransportation` (`id`, `description`, `companyid`, `capacity`, `firstyear`) VALUES
(5, 'Κρήτη I', 1, 1200, 1985),
(6, 'Κρήτη II', 1, 1600, 1984),
(7, 'Πρέβελης', 1, 800, 1978),
(8, 'Blue Horizon', 2, 1600, 1989),
(9, 'Blue Star I', 2, 850, 1988),
(10, 'Διαγόρας', 2, 1400, 1974),
(11, 'Dodekanisos Express', 12, 800, 1990),
(12, 'Dodekanisos Pride', 12, 500, 1987),
(13, 'Θεολόγος', 13, 800, 1990),
(14, 'Αικατερίνη', 13, 900, 1991),
(15, 'SUPERFERRY II', 14, 1200, 1991),
(16, 'Highspeed 4', 15, 1500, 1998),
(17, 'Highspeed 5', 15, 1200, 1998),
(18, 'Flyingcat ', 15, 600, 2001),
(19, 'Ionian Star', 16, 600, 1978),
(20, 'Ιονίς', 16, 450, 1979),
(21, 'Knossos Palace', 17, 2500, 1998),
(22, 'Festos Palace', 17, 2500, 1994),
(23, 'SuperFast 1', 18, 1600, 2001),
(24, 'SuperFast XI', 18, 2000, 1999);
```



Πίνακας 33: Πίνακας tbltransportation

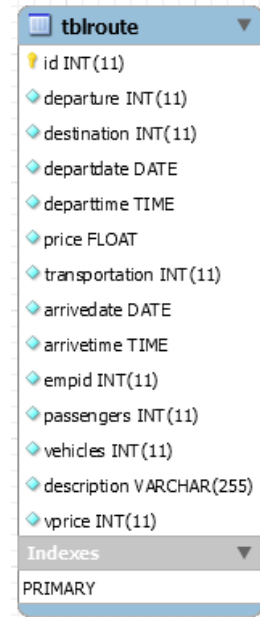
Εικόνα 44: tbltransportation-Create and Insert Table
tblroute (Δρομολόγιο)

Στον πίνακα καταχωρούνται από τον Διαχειριστή τα πιθανά δρομολόγια με κάθε μεταφορικό μέσο από ένα προορισμό (city) σε έναν άλλο, με τις ώρες και ημέρες αναχώρησης και άφιξης.

```
CREATE TABLE IF NOT EXISTS `tblroute` (
  `id` int(11) NOT NULL,
  `departure` int(11) NOT NULL,
  `destination` int(11) NOT NULL,
  `departdate` date NOT NULL,
  `departtime` time NOT NULL,
  `price` float NOT NULL,
  `transportation` int(11) NOT NULL,
  `arrivedate` date NOT NULL,
  `arrivetime` time NOT NULL,
  `empid` int(11) NOT NULL,
  `passengers` int(11) NOT NULL,
  `vehicles` int(11) NOT NULL,
  `description` varchar(255) NOT NULL,
  `vprice` int(11) NOT NULL
) ENGINE=InnoDB AUTO_INCREMENT=22 DEFAULT CHARSET=greek;
```

```
INSERT INTO `tblroute` (`id`, `departure`, `destination`, `departdate`,
`departtime`, `price`, `transportation`, `arrivedate`, `arrivetime`,
`empid`, `passengers`, `vehicles`, `description`, `vprice`) VALUES
(10, 13, 7, '2015-05-15', '21:00:00', 40, 24, '2015-05-16', '06:00:00', 12, 2000, 1000, '', 80),
(11, 3, 12, '2015-05-15', '22:00:00', 50, 8, '2015-05-16', '10:00:00', 19, 1000, 500, '', 100),
(12, 5, 4, '2015-05-16', '11:00:00', 60, 9, '2015-05-16', '23:00:00', 15, 500, 500, '', 120),
(13, 11, 12, '2015-05-17', '10:00:00', 25, 18, '2015-05-17', '17:00:00', 16, 600, 200, '', 60),
(14, 12, 4, '2015-05-18', '13:00:00', 45, 22, '2015-05-18', '23:30:00', 12, 700, 100, '', 90),
(15, 10, 2, '2015-05-18', '16:00:00', 30, 11, '2015-05-18', '20:00:00', 16, 800, 40, '', 60),
(16, 7, 6, '2015-05-19', '08:00:00', 28, 16, '2015-05-19', '13:00:00', 12, 900, 1000, '', 56),
(17, 2, 8, '2015-05-19', '11:00:00', 45, 12, '2015-05-19', '21:00:00', 19, 500, 400, '', 80),
(18, 1, 3, '2015-05-19', '17:00:00', 50, 14, '2015-05-19', '23:00:00', 15, 1500, 1000, '', 90),
(19, 8, 5, '2015-05-20', '08:00:00', 15, 18, '2015-05-20', '10:00:00', 15, 900, 520, '', 25),
(20, 4, 10, '2015-05-20', '11:00:00', 50, 5, '2015-05-20', '18:00:00', 16, 900, 400, '', 120),
(21, 6, 11, '2015-05-21', '13:00:00', 60, 20, '2015-05-22', '02:00:00', 16, 500, 120, '', 150);
```

Εικόνα 45: tblroute-Create and Insert Table



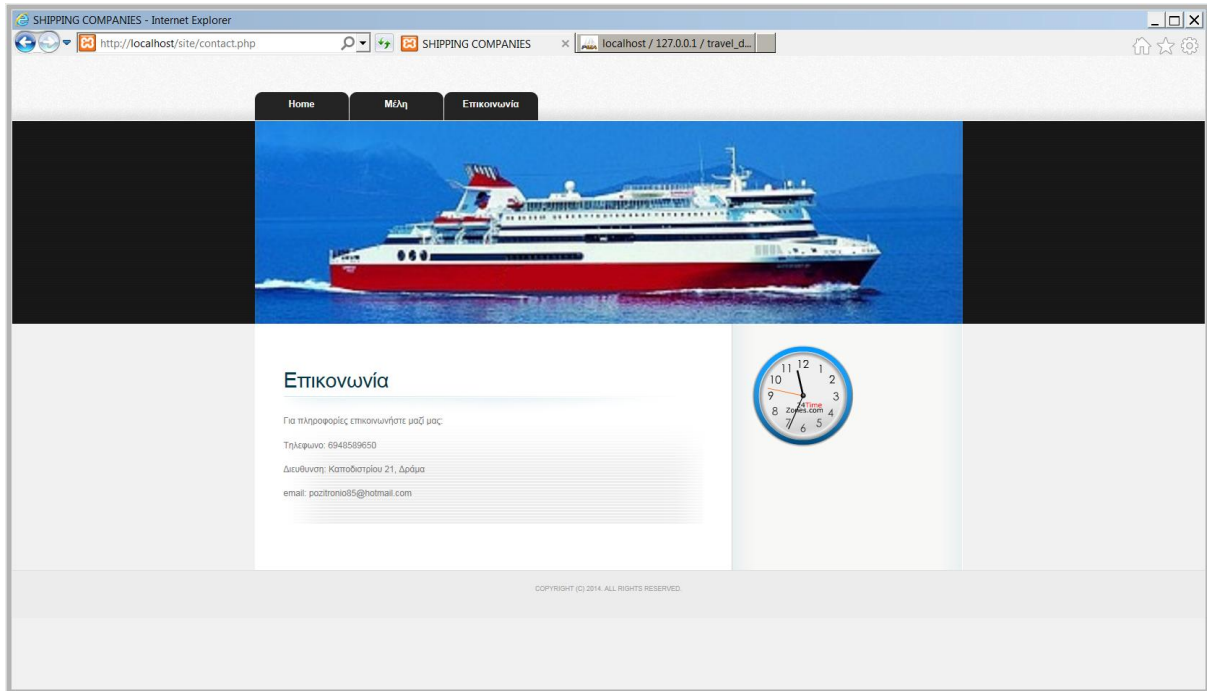
Πίνακας 34: Πίνακας tblroute

5.3 Στατικές σελίδες

Ο δικτυακός τόπος χωρίζεται σε δύο τμήματα: το στατικό και το δυναμικό. Στατική ονομάζεται η ιστοσελίδα της οποίας το περιεχόμενο μεταφέρεται στον χρήστη ακριβώς στην μορφή που είναι αποθηκευμένο στον εξυπηρετητή ιστοσελίδων (web server). Τα περιεχόμενα μιας στατικής ιστοσελίδας εμφανίζονται με την ίδια μορφή σε όλους του χρήστες, με την μορφή που είναι αποθηκευμένα στο σύστημα αρχείων του εξυπηρετητή ιστοσελίδων. Οι στατικές ιστοσελίδες είναι αποθηκευμένες συνήθως σε μορφή HTML και μεταφέρονται χρησιμοποιώντας το πρωτόκολλο HTTP. Με απλά λόγια, στατική ιστοσελίδα είναι εκείνη που το περιεχόμενό της είναι συγκεκριμένο και προκαθορισμένο. Δεν γίνονται πολλές αλλαγές και ανανέωση στο περιεχόμενό της. Η ιστοσελίδα, όμως, μπορεί να έχει διάφορα κινούμενα γραφικά. Είναι κατάλληλες, κυρίως, για την δημιουργία 'μόνιμων/στατικών παρουσιάσεων', όπου δεν υπάρχει συχνά η ανάγκη να τροποποιείται το περιεχόμενό τους.

5.3.1 Contact.php

Στη σελίδα αυτή παρουσιάζονται μόνο στατικά στοιχεία όπως είναι η διεύθυνση και τα στοιχεία επικοινωνίας του ιδιοκτήτη της εφαρμογής. Οι πληροφορίες αυτές είναι στατικές και δεν τροποποιούνται παρά μόνο με τη χρήση HTML και το ανέβασμα της νέας σελίδας.



Εικόνα 46: Contact.php

5.4 Δυναμικές σελίδες

Δυναμική λέγεται η ιστοσελίδα η οποία δημιουργείται δυναμικά την στιγμή της πρόσβασης σε αυτή ή την στιγμή που ο χρήστης αλληλεπιδρά με τον εξυπηρετητή ιστοσελίδων. Οι δυναμικές ιστοσελίδες θεωρούνται δομικό στοιχείο της νέας γενιάς του παγκόσμιου ιστού όπου η πληροφορία διαμοιράζεται σε πολλαπλές ιστοσελίδες. Το περιεχόμενό της είναι εύκολα προσαρμόσιμο και αλλάζει τη δεδομένη στιγμή που θέλουμε, κάνοντας όποιες προσθήκες, αφαιρέσεις, αλλαγές και διορθώσεις θέλουμε. Μπορεί να τη διαχειριστεί ο χρήστης πολύ εύκολα και μπορεί να επεκταθεί εύκολα, προσθέτοντας πάνω σε αυτήν ένα e-shop, δημιουργώντας μια νέα βάση δεδομένων και οτιδήποτε άλλο θέλουμε. Ουσιαστικά πρόκειται για μία εφαρμογή, ένα πρόγραμμα, στο οποίο είναι μεγαλύτερη η αλληλεπίδραση του χρήστη με την σελίδα. Στη παρούσα πτυχιακή, το σύνολο των σελίδων είναι δυναμικές.

Το δυναμικό περιεχόμενο των ιστοσελίδων έχει υλοποιηθεί με τη χρήση της πλατφόρμας PHP – MySQL. Τα περιεχόμενα των δυναμικών σελίδων προέρχονται κυρίως από αποτελέσματα ερωτημάτων προς τη βάση δεδομένων που έχει δημιουργηθεί σε MySQL. Για την πρόσβαση στη βάση υπάρχει ένα αρχείο php με το όνομα dbs.php. Εκεί περιγράφονται οι λεπτομέρειες πρόσβασης στη βάση:

```
<?php
/* το όνομα του server */
$host = "localhost";
/* το όνομα του χρήστη */
$user = "root";
/* τον κωδικό του χρήστη*/
$pwd = "";
/* το όνομα της βάσης δεδομένων */
$db = "travel_db";
?>
```


Εικόνα 47: Πρόσβαση στη βάση δεδομένων

Το παραπάνω αρχείο dbs.php, χρησιμοποιείται από αρχείο catalogdbinfo.php. Αυτό είναι υπεύθυνο για τη σύνδεση με τη βάση δεδομένων και ορίζει τις παραμέτρους οι οποίες αφορούν την κωδικοποίηση της γραμματοσειράς και της γλώσσας που χρησιμοποιήθηκαν από τη βάση δεδομένων.

```
<?php
/* εισαγωγή του dbs.php */
include 'dbs.php';

$link = mysql_connect($host, $user, $pwd);
/* έλεγχος για τυχόν πρόβλημα με τη βάση δεδομένων */
if (!$link)
{
/* η εντολή die διακόπτει τον κώδικα */
die ("Error: ". mysql_error());
}
else
{
/* επιλογή της βάσης δεδομένων */
$db = mysql_select_db($db);
/* έλεγχος σφάλματος και διακοπή*/
if (!$db)
{
die("Error: ".mysql_error());
}
}
/* ορισμός Ελληνικής γλώσσας */
mysql_query('set character set greek', $link);
mysql_query("SET NAMES 'greek'", $link);
?>
```

Εικόνα 48: catalogdbinfo

Στην αρχή κάθε σελίδας της εφαρμογής, υπάρχει ο παρακάτω κώδικας. Με αυτό το κομμάτι ξεκινάει το session (σύνοδος) και γίνεται χρήση του αρχείου catalogdbinfo.php

```
<?php
session_start();
include 'catalogdbinfo.php';
?>
```

Εικόνα 49: Session_start

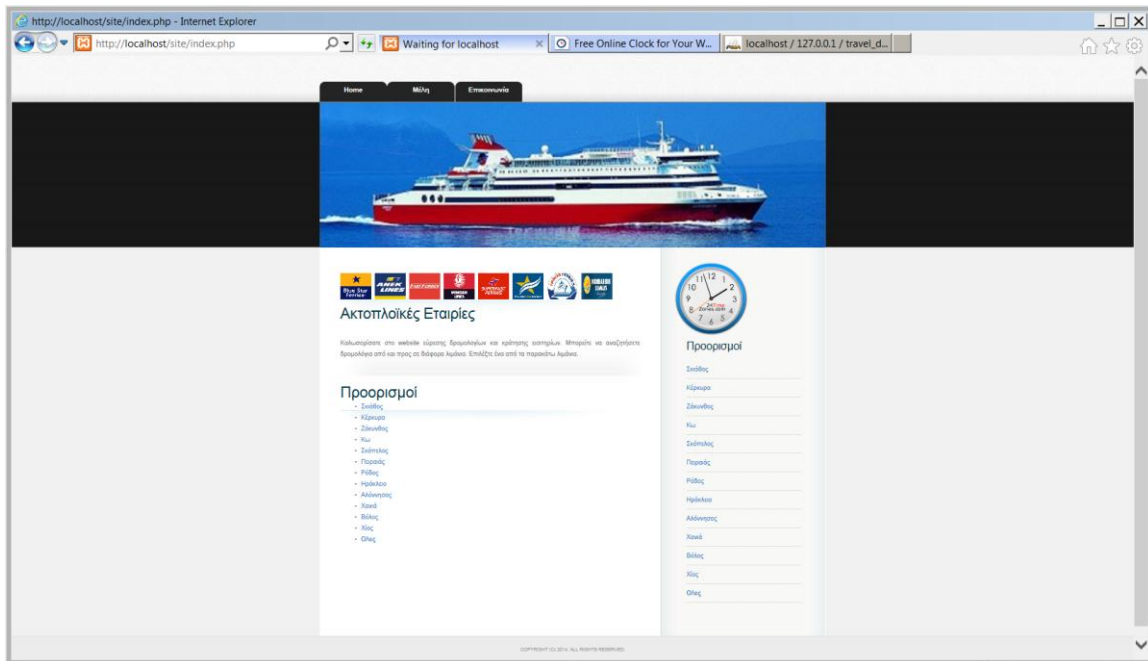
Σε όλες τις σελίδες της εφαρμογής συμπεριλαμβάνεται το ρολόι, το οποίο το έχω πάρει από την ιστοσελίδα <http://24timezones.com>. Είναι script code και αυτός φαίνεται παρακάτω:

```
<script src="http://24timezones.com/js/swfobject.js" language="javascript"></script>
<script src="http://24timezones.com/timescript/maindata.js.php?city=1925389" language="javascript"></script>
<table><tr><td><div id="flash_container_tt552532e560beb"></div><script type="text/javascript">
var flashMap = new SWFObject("http://24timezones.com/timescript/clock_final.swf", "main", "150",
"150", "7.0.22", "#FFFFFF", true)
flashMap.addParam("movie", "http://24timezones.com/timescript/clock_final.swf");
flashMap.addParam("quality", "high");
flashMap.addParam("wmode", "transparent");
flashMap.addParam("flashvars", "color=0099FF&logo=1&city=1925389");
flashMap.write("flash_container_tt552532e560beb");
</script></td></tr><tr><td style="text-align: right; font-weight: bold"><a href=
"http://24timezones.com/world_directory/time_in_athens.php" target="_blank" style="text-decoration: none"
title="accurate time Athens"></a></td></tr></table>
```

Εικόνα 50: Clock

5.4.1 Σελίδες Χρήστη

Η πρώτη σελίδα που ανοίγει στον χρήστη είναι η σελίδα index.php. Είναι η κεντρική σελίδα της εφαρμογής.



Εικόνα 51: Index.php

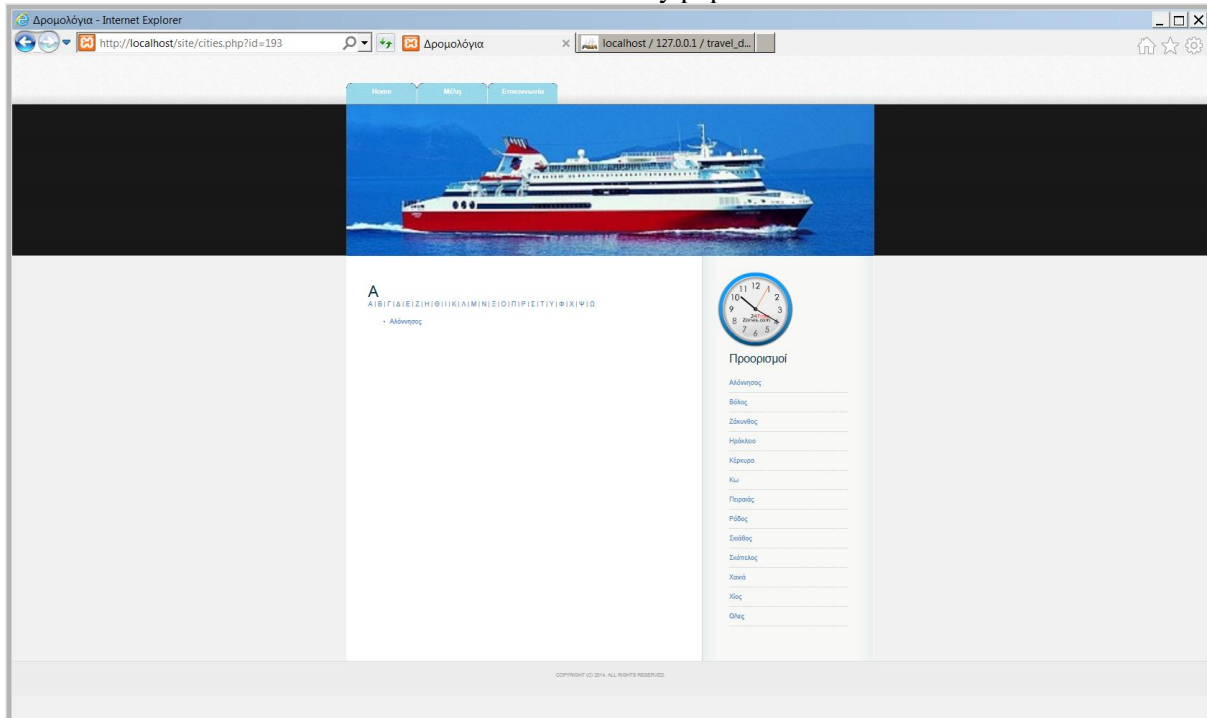
Υπάρχει ένα link προς τη σελίδα cities.php όπου εμφανίζονται αλφαβητικά όλες οι διαθέσιμες πόλεις

```
<li><b> <a href="cities.php?id=193" class="more">Όλες</a></b></li>
```

Ο παρακάτω κώδικας χρησιμοποιείται για να εμφανίσει όλες τις κύριες πόλεις και να δημιουργήσει ένα link προς τη σελίδα city.php στέλνοντας το id της πόλης ως παράμετρο.

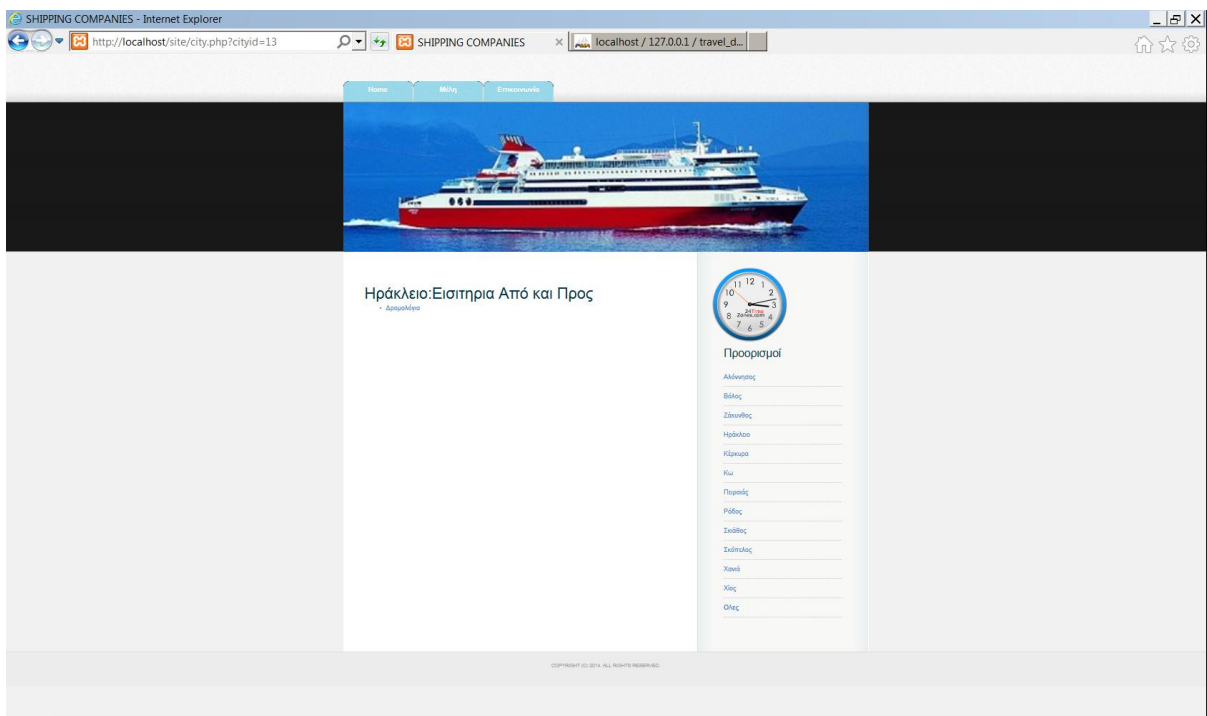
```
<h2>Προορισμοί</h2>
<ul class="sb_list">
<?php $cities=mysql_query("select * from tblcity order by cityname");
$i=0;
while($i < mysql_num_rows($cities))
{
?>
<li><a href="city.php?cityid=<?php
echo mysql_result($cities,$i,"cityid");?>"><?php
echo mysql_result($cities,$i,"cityname");?></a></li>
<?php
$i++;
}
?>
```

Εικόνα 51: City.php



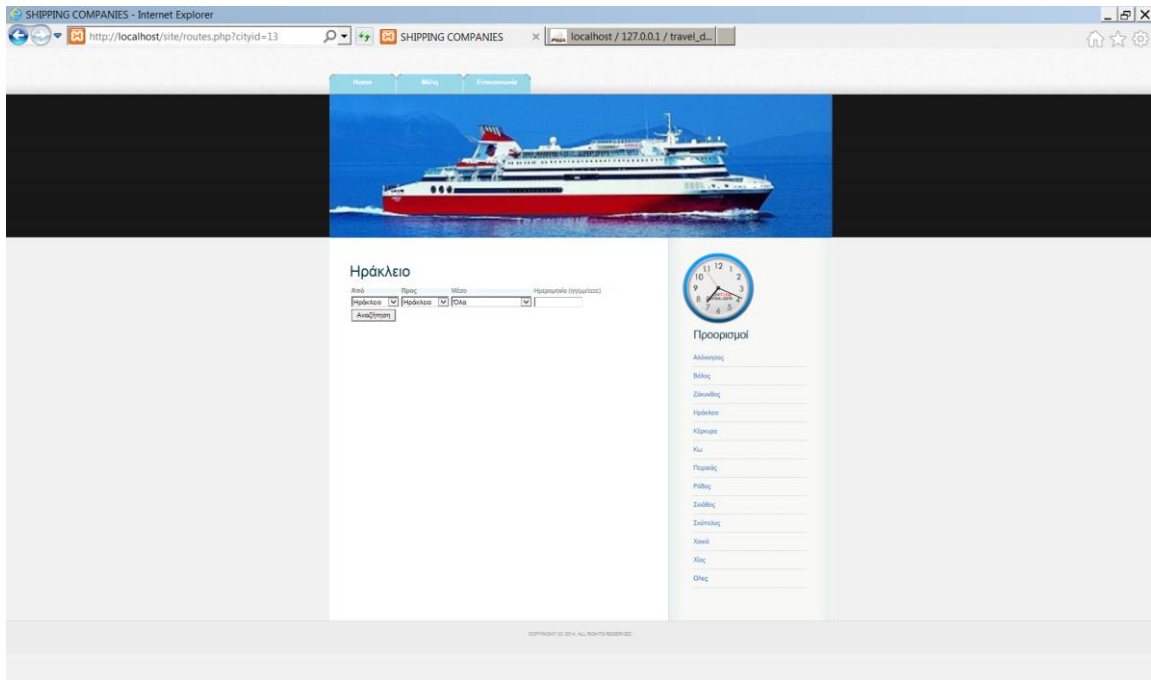
Εικόνα 52: Cities.php

Ο χρήστης, χωρίς να κάνει ακόμα εγγραφή ή login, μπορεί να ενημερωθεί σχετικά με τα δρομολόγια. Μπορεί να επιλέξει ανάμεσα στις διάφορες πόλεις και να δει αναλυτικά τα δρομολόγια από και προς τον προορισμό που επιθυμεί. Αυτό το επιτυγχάνει είτε από το κυρίως μενού είτε από το sidebar όπου υπάρχουν όλα τα διαθέσιμα λιμάνια. Καθώς επιλέξει τον τόπο αναχώρησης, προβάλλεται η επόμενη σελίδα,



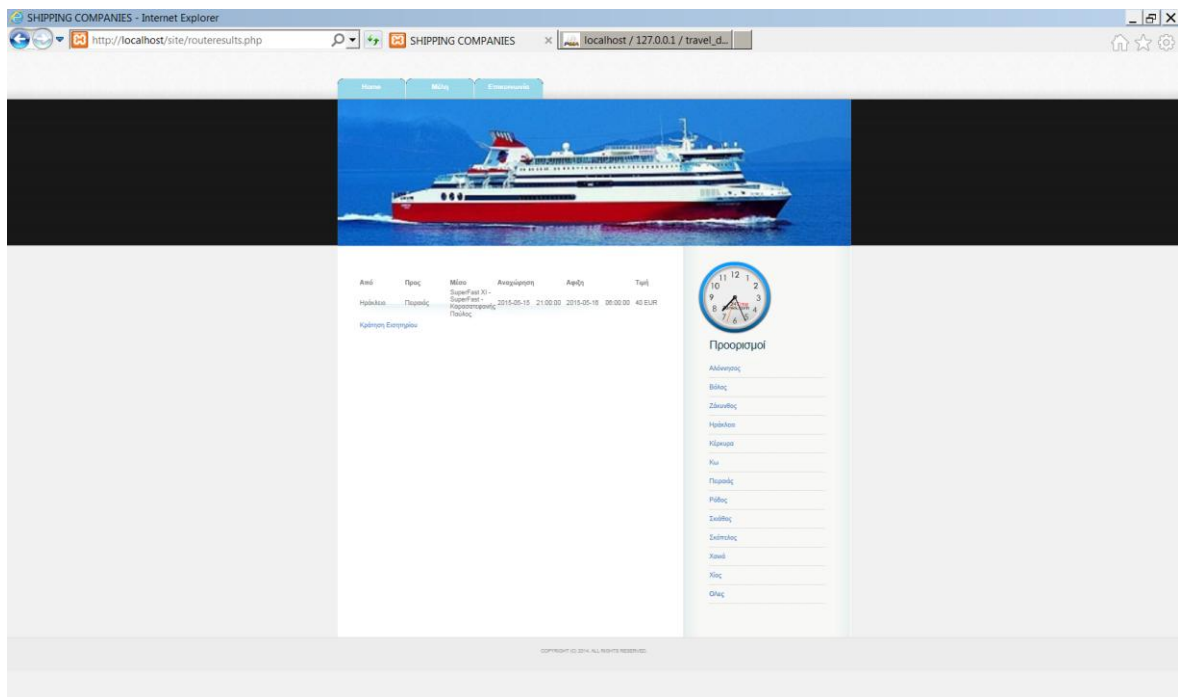
Εικόνα 53: Δρομολόγια

Κάνοντας κλικ στο 'Δρομολόγια', μεταφερόμαστε στην παρακάτω εικόνα από όπου επιλέγουμε προορισμό και ημερομηνία αναχώρησης. Επίσης, μας δίνεται η δυνατότητα να επιλέξουμε το μεταφορικό μέσο εφόσον το γνωρίζουμε.



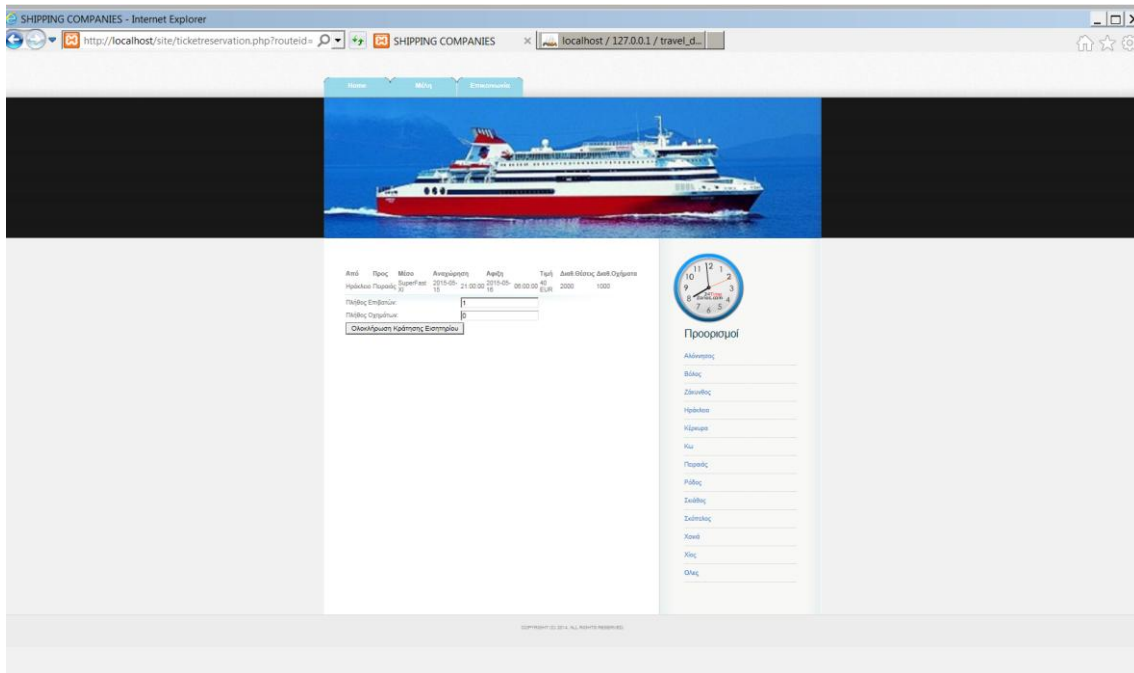
Εικόνα 54: Search route

Καθώς γίνει η αναζήτηση, μεταφερόμαστε στη σελίδα routeresults.php, όπου γίνεται η κράτηση του εισιτηρίου. Εδώ εμφανίζονται πληροφορίες σχετικά με το πλοίο που κάνει το ταξίδι, την ναυτιλιακή εταιρία, τον πλοίαρχο, ώρα και ημερομηνία αναχώρησης και άφιξης και τιμή εισιτηρίου.



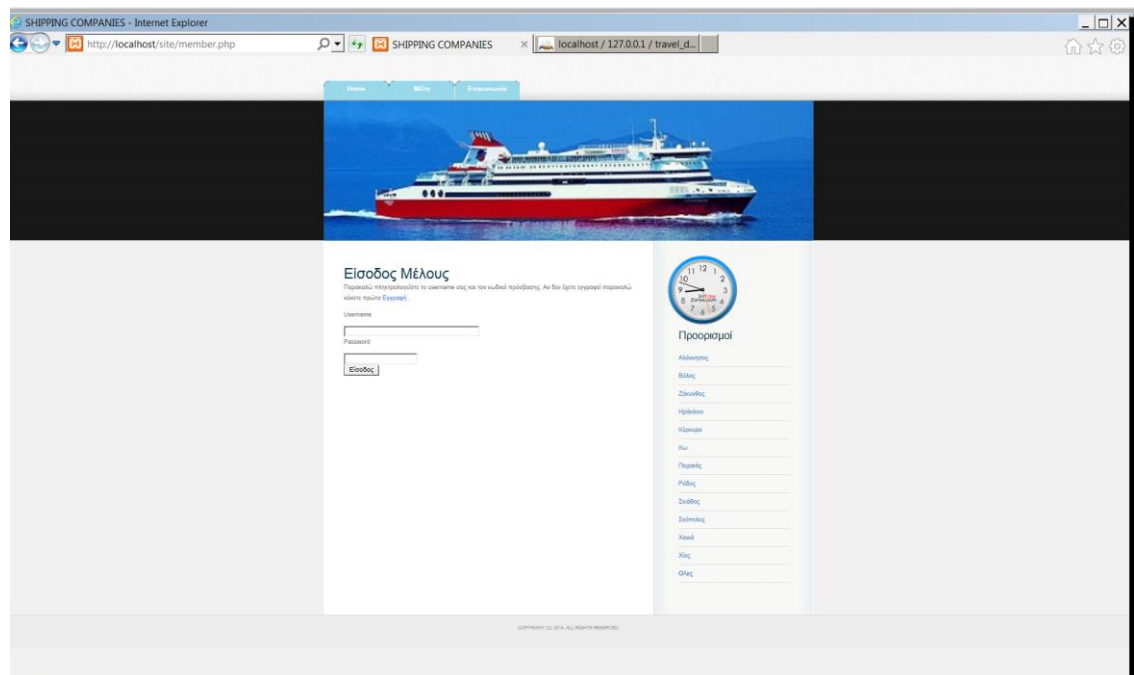
Εικόνα 55: Routeresults_php

Η επόμενη σελίδα, ticketreservation.php, μας δίνει τη δυνατότητα να επιλέξουμε το πλήθος των εισιτηρίων και το πλήθος των οχημάτων που θα κάνουμε κράτηση. Επίσης μας πληροφορεί για τις διαθέσιμες θέσεις και τα διαθέσιμα οχήματα.

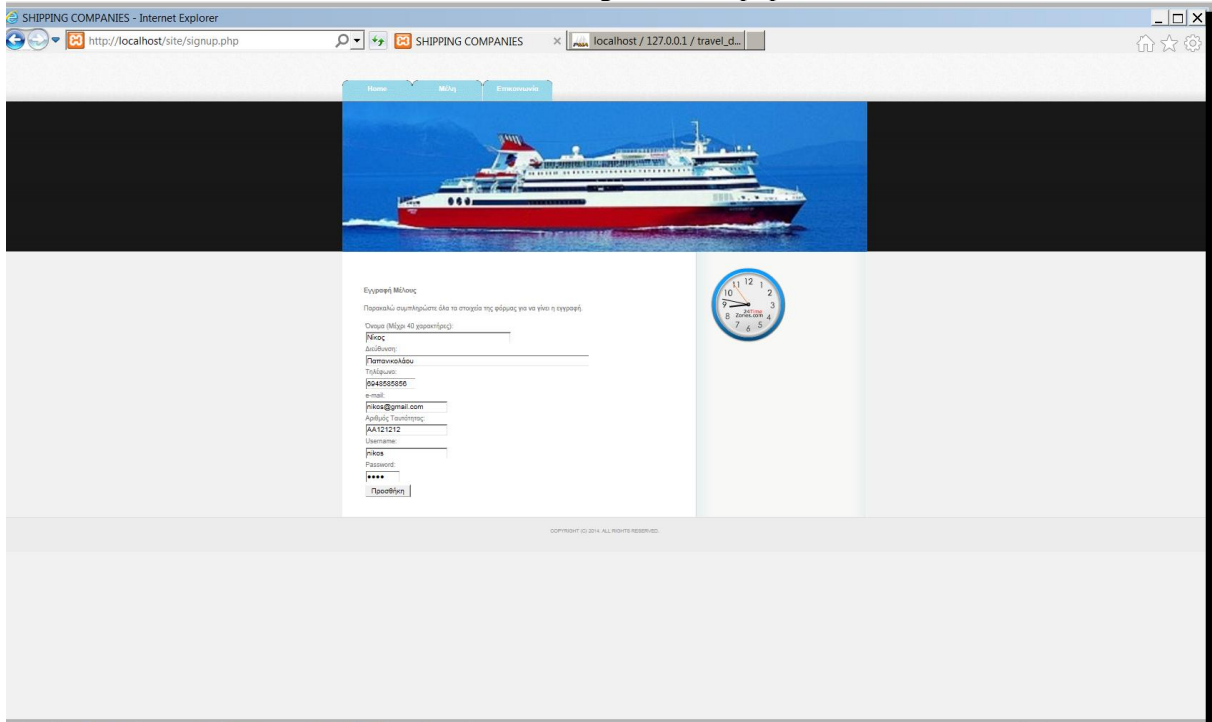


Εικόνα 56: Ticketsreservation_php

Διευκρίνιση, ο χρήστης μέλος δεν μπορεί να κάνει κράτηση σε περισσότερα από ένα εισιτήρια. Για να γίνει κράτηση σε περισσότερα από ένα εισιτήρια θα πρέπει να είναι εγγεγραμμένος ως Ταξιδιωτικός πράκτορας. Εάν πατήσουμε Ολοκλήρωση Κράτησης Εισιτηρίου δε θα πραγματοποιηθεί η κράτηση, καθώς δεν έχει γίνει login. Σε αυτή την περίπτωση, μεταφερόμαστε στην σελίδα member.php. Σε αυτό το σημείο πρέπει είτε να κάνουμε login, σε περίπτωση που έχει δημιουργηθεί λογαριασμός, είτε να δημιουργήσουμε νέο λογαριασμό.



Εικόνα 57: Loginmember.php



Εικόνα 58: Εγγραφή Μέλους

Το παρακάτω κομμάτι κώδικα που ακολουθεί είναι ένα javascript και είναι αυτό που κάνει τον έλεγχο ότι υπάρχουν διαθέσιμες θέσεις και διαθέσιμα οχήματα προς επιβίβαση. Επίσης, για να ολοκληρωθεί η κράτηση πρέπει να δοθεί ένας αριθμός κράτησης θέσης ή κράτησης οχήματος.

```
function checklog ()
{
    error="";
    if (document.reserveform.pqty.value == "")
        error+=" Παρακαλώ εισάγετε τιμή σε Πλήθος Επιβατών ";
    if (document.reserveform.vqty.value == "")
        error+=" Παρακαλώ εισάγετε τιμή σε Πλήθος Οχημάτων ";

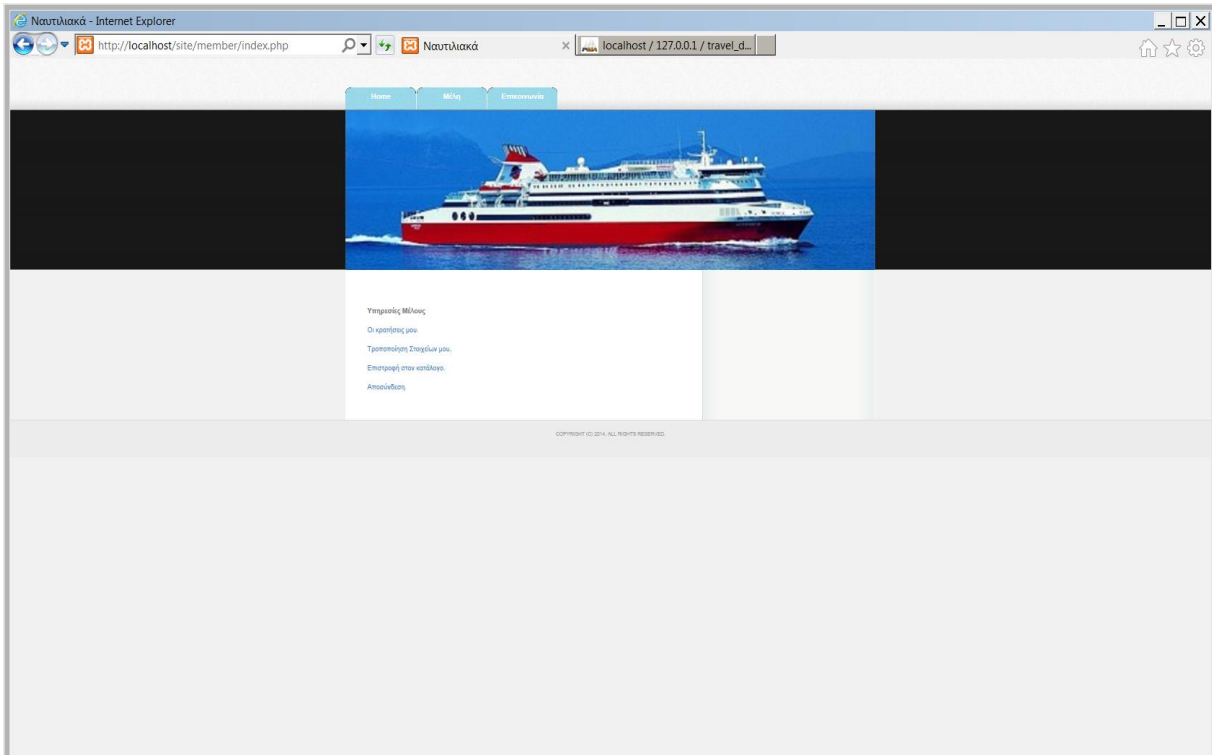
    if (document.reserveform.vqty.value >= document.reserveform.avveh.value)
        error+=" Δεν υπάρχουν τόσες θέσεις για οχήματα ";

    if (document.reserveform.pqty.value >= document.reserveform.avpass.value)
        error+=" Δεν υπάρχουν τόσες θέσεις για επιβάτες ";

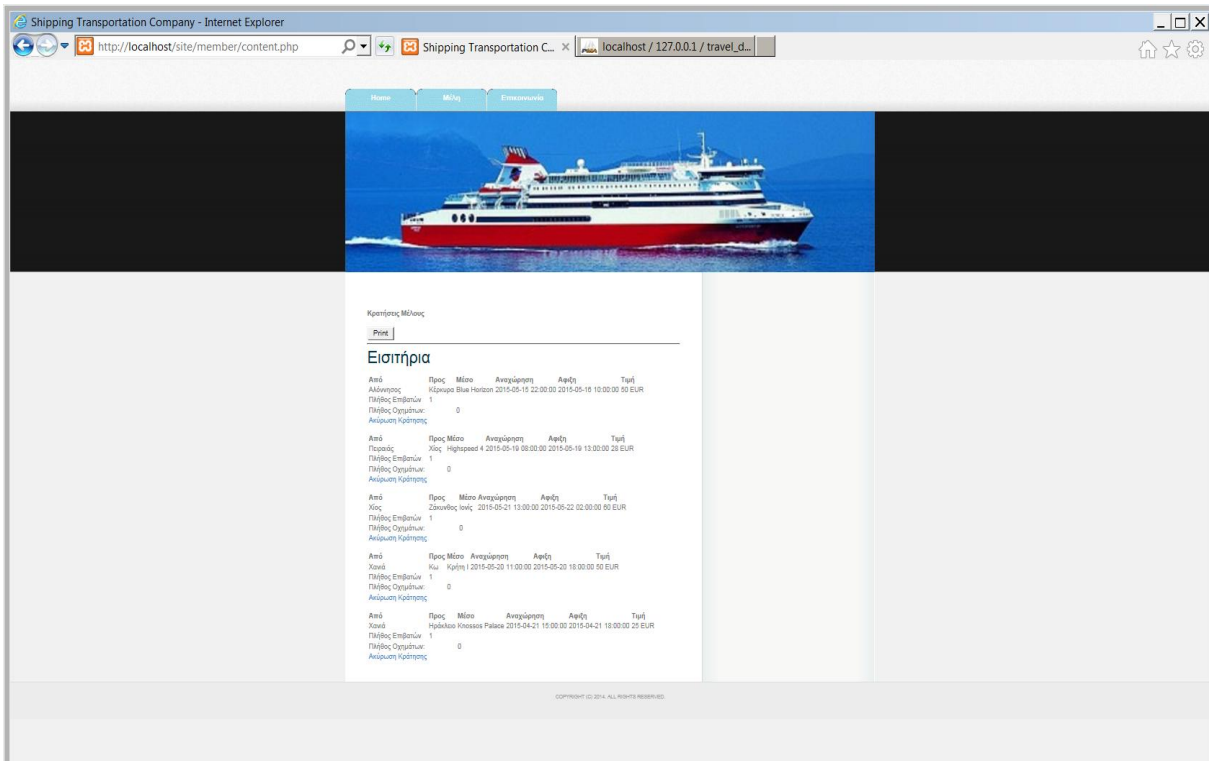
    if (error!="")
    {
        error=" "+error;
        alert(error);
        return false;
    }
    else
        return true;
}
```

Εικόνα 59: Διαθέσιμα Εισιτήρια

Αν ο παραπάνω έλεγχος είναι επιτυχής τότε η κράτηση προστίθεται στις κρατήσεις του χρήστη. Ο χρήστης πατώντας στην επιλογή Μέλη μπορεί να δει τις παλαιότερες κρατήσεις του και αν επιθυμεί να κάνει εκτύπωση της σελίδας με τα εισιτήρια που έχει κλείσει.

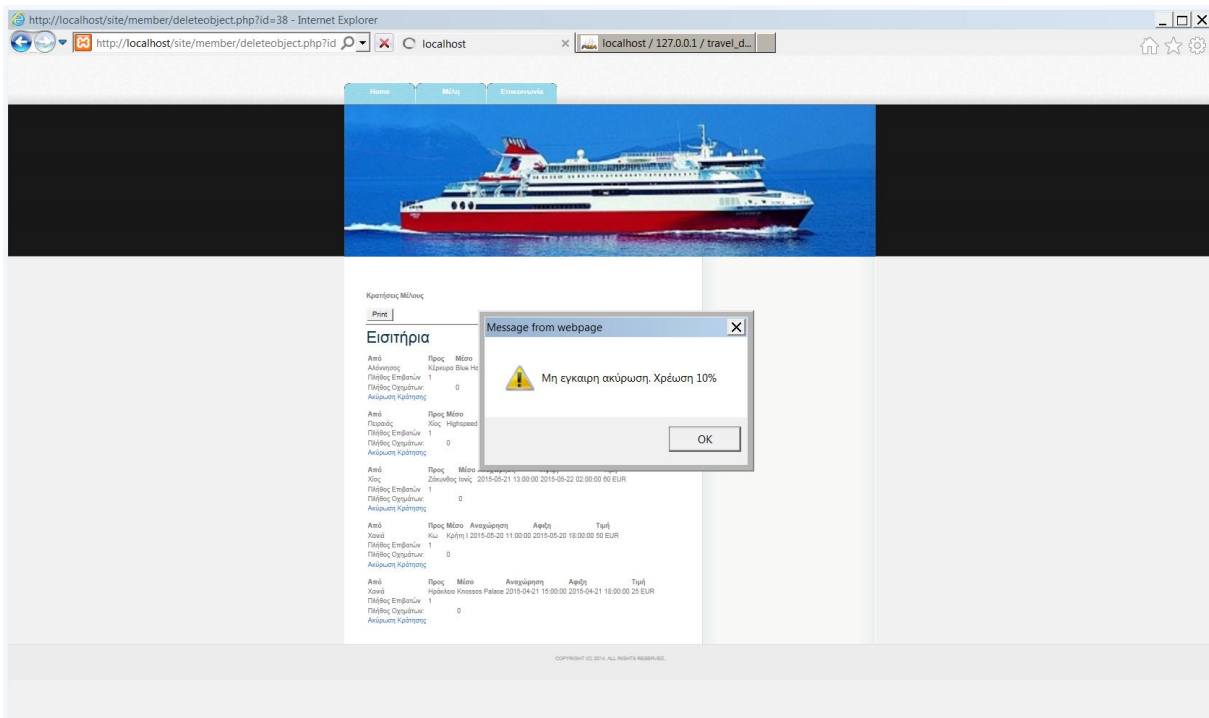


Εικόνα 60: Υπηρεσίες Μέλους



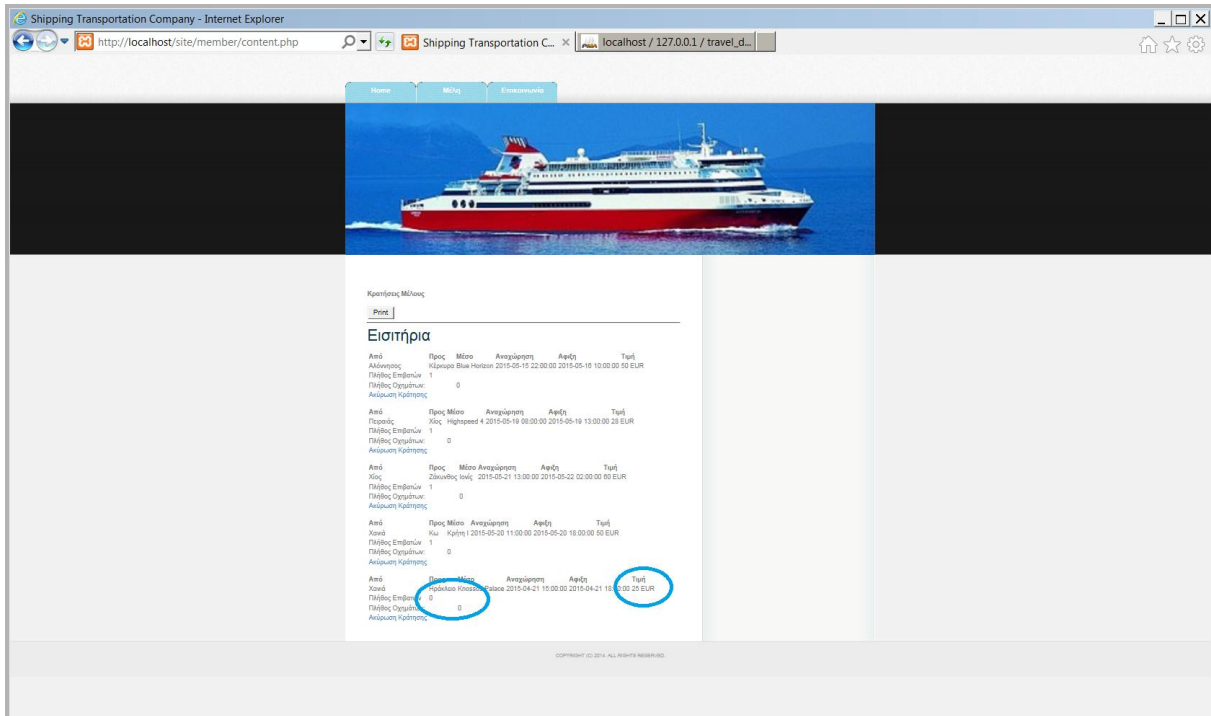
Εικόνα 61: Κρατήσεις εισιτηρίων

Η ακύρωση είναι πλήρης σε περίπτωση έγκαιρης διαγραφής αλλά αν η ακύρωση γίνει εντός 24ωρου από την αναχώρηση τότε ο χρήστης επιβαρύνεται με το 90% του ποσού.



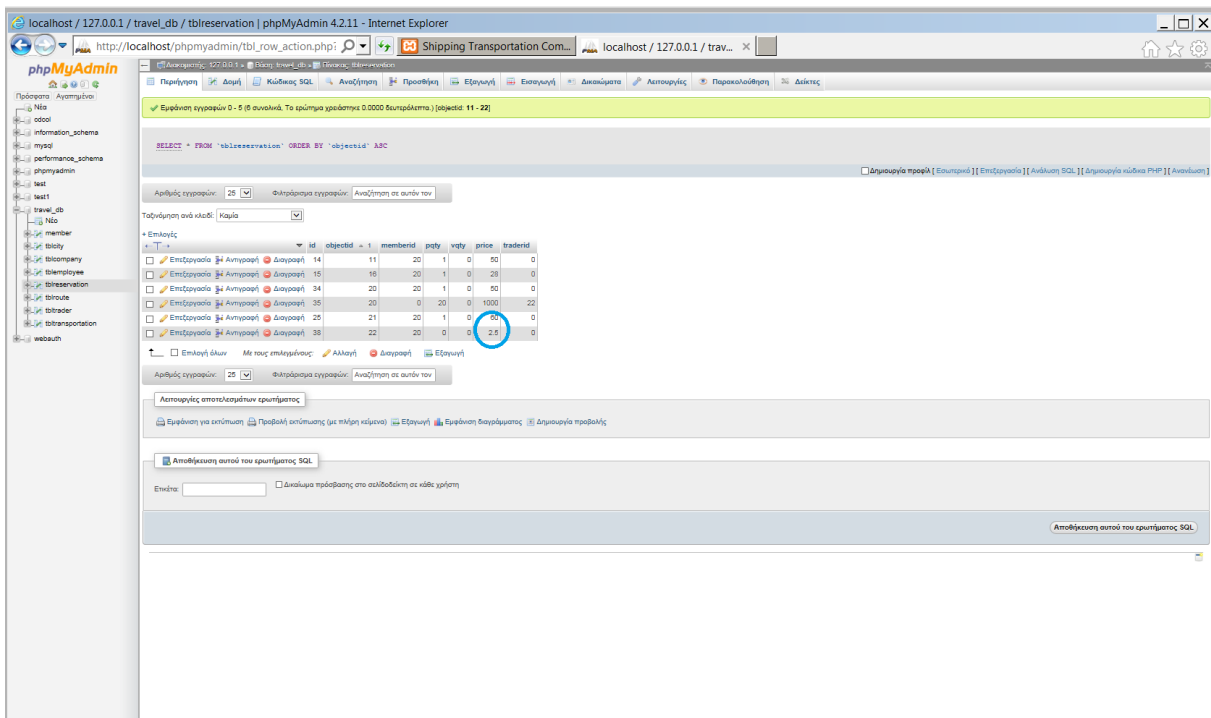
Εικόνα 62: Ακύρωση κράτησης 1

Παρατηρούμε πως το πλήθος επιβατών έχει μηδενιστεί για το συγκεκριμένο δρομολόγιο όπου έγινε η ακύρωση αλλά δεν έχει μειωθεί η τιμή εισιτηρίου.



Εικόνα 63: Ακύρωση κράτησης 2

Όμως, στο phpMyAdmin, στον πίνακα tbltransportation και στο πεδίο price, παρατηρούμε ότι έχει αλλάξει.



Εικόνα 64: Ακύρωση κράτησης 3

Ο παρακάτω κώδικας κάνει τον έλεγχο σε περίπτωση ακύρωσης εισιτηρίου που γίνει μέσα σε ένα εικοσιτετράωρο.

```

$dates=mysql_query("select tblroute.departdate , tblroute.departtime from tblreservation,tblroute
where tblreservation objectid=tblroute.id and tblreservation.id=".$sid);

$d1=mysql_result($dates,0,'tblroute.departdate')."T".mysql_result($dates,0,'tblroute.departtime');
$date1 = new DateTime($d1);
$date2 = new DateTime(date('Y-m-dTH:i:s'));

$diff = $date2->diff($date1);

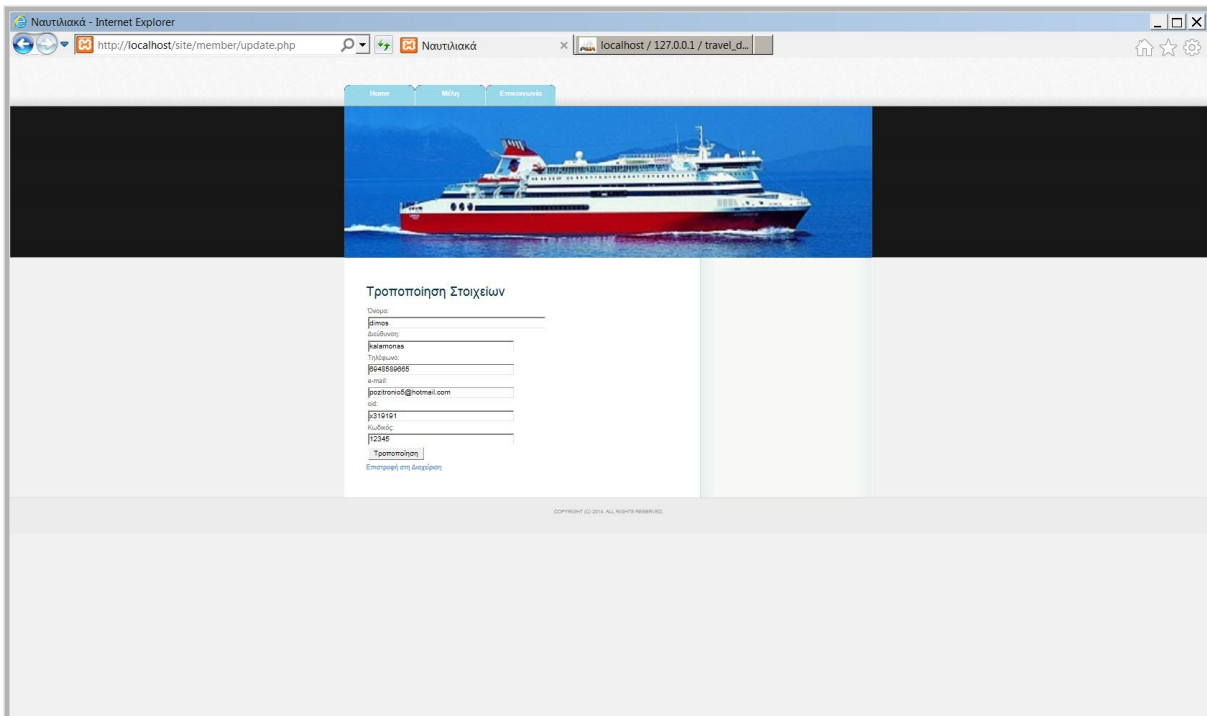
$hours = $diff->h;
$hours = $hours + ($diff->days*24);

if ($hours<24)
{
    echo "<script type='text/javascript'>alert('Μη ενκαίριον ακύρωση. Χρέωση 10%!');</script>";

mysql_query("update tblreservation set qty=0, vqty=0, price=0.1*price where id=".$sid);
}
else
{
mysql_query("delete from tblreservation where id=".$sid);
}
mysql_close($link);
    
```

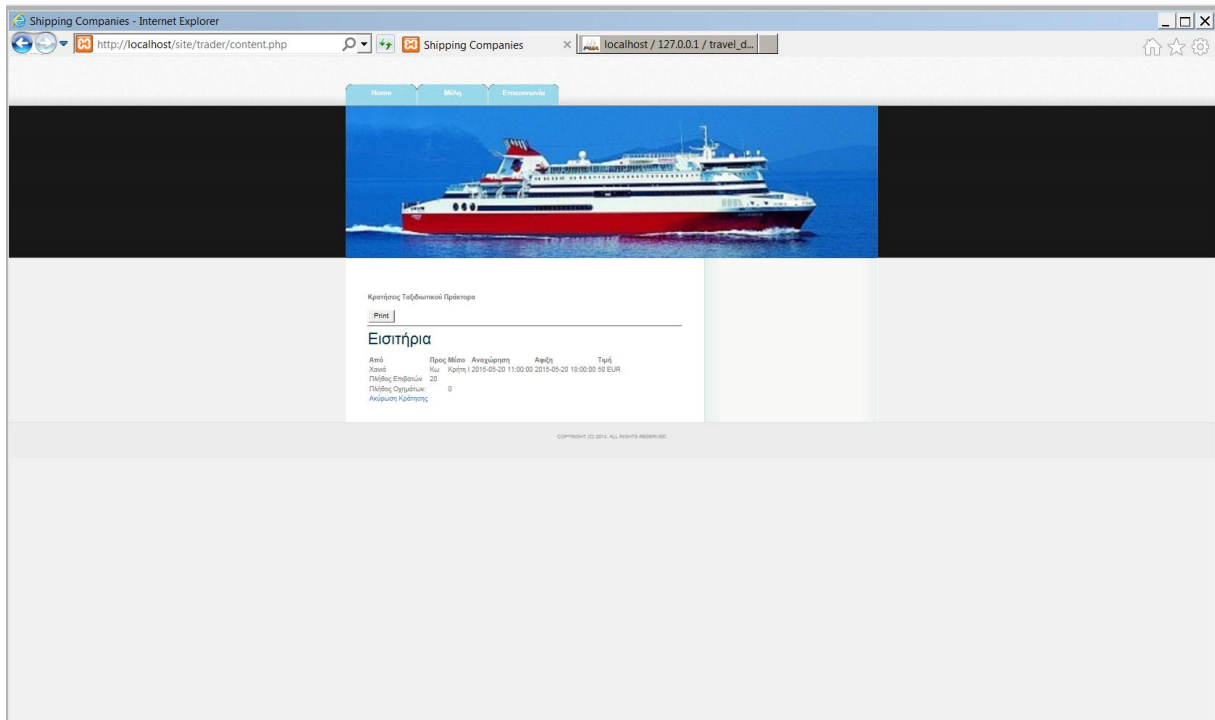
Εικόνα 65: Έλεγχος ακύρωσης εισιτηρίου

Επίσης μπορεί να επιστρέψει στην αρχική σελίδα ή να κάνει αποσύνδεση ή ακόμα να κάνει τροποποίηση των στοιχείων του.



Εικόνα 66: Τροποποίηση στοιχείων

Ο ταξιδιωτικός πράκτορας έχει τις ίδιες επιλογές με τον απλό χρήστη, συμπεριλαμβανομένου ότι μπορεί να κάνει μαζική κράτηση εισιτηρίων επιβατών και οχημάτων.



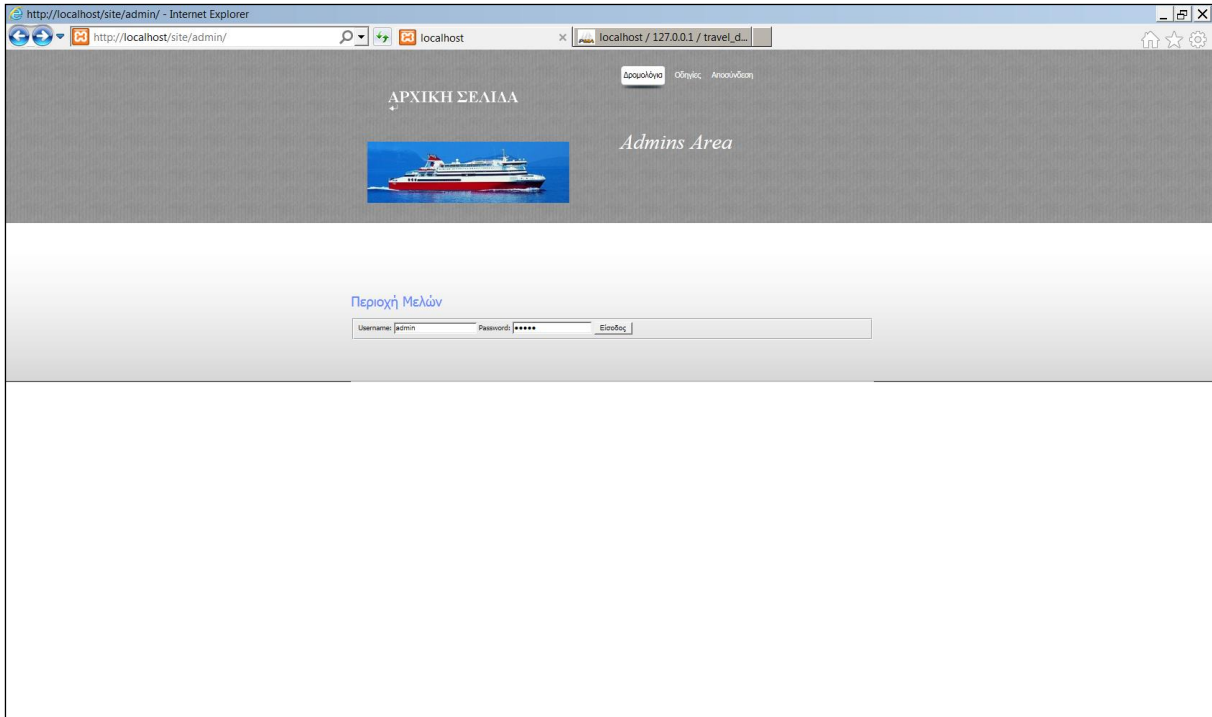
Εικόνα 67: Ταξιδιωτικός Πράκτορας

5.4.2 Σελίδες Administrator

Στην περιοχή αυτή ο διαχειριστής της εφαρμογής έχει τη δυνατότητα να επεξεργαστεί και να τροποποιήσει τις διάφορες παραμέτρους που του δίνονται από την εφαρμογή. Κάποιες από αυτές είναι η προσθήκη και τροποποίηση λιμανιών, караβιών, ναυτιλιακών εταιριών, ταξιδιωτικών πρακτόρων, δρομολογίων, υπαλλήλων. Για να εισέλθει στο σύστημα θα πρέπει να εισάγει το username και το password του διαχειριστή έτσι όπως αυτοί έχουν δημιουργηθεί από τον developer.

Ο διαχειριστής έχει την επιλογή να μεταφερθεί κατευθείαν στην αρχική σελίδα πατώντας στο μέρος όπου αναγράφεται 'ΑΡΧΙΚΗ ΣΕΛΙΔΑ', στη σελίδα με τα δρομολόγια πατώντας στο κουμπί 'ΔΡΟΜΟΛΟΓΙΑ', στη σελίδα σχετικά με οδηγίες περί αλλαγών της εφαρμογής πατώντας στο κουμπί 'ΟΔΗΓΙΕΣ'. Μπορεί επίσης να κάνει αποσύνδεση του λογαριασμού του για να υπάρχει ασφάλεια του συστήματος. Όλα τα παραπάνω αναφέρονται και για τις εσωτερικές σελίδες του διαχειριστή, καθώς όπως θα δούμε στη συνέχεια συμπεριλαμβάνονται από όλες τις σελίδες της εφαρμογής.

Για την είσοδο στη σελίδα `admin/index.php`, ο χρήστης θα πρέπει να εισάγει στη διεύθυνση του browser την εξής διεύθυνση: <http://localhost/site/admin>, με username admin και password 12345.



Εικόνα 68: Loginadmin.php

Για την ταυτοποίηση του διαχειριστή χρησιμοποιείται ο ακόλουθος κώδικας

```
<h2 class="title1"><span lang="el">Περιοχή Μελών</span></h2>
<form id="form1" method="post" action="loginadmin.php">
  <fieldset>
    <label for="inputtext1">Username:</label>
    <input id="inputtext1" type="text" name="inputtext1" value="" size="20" />
    <label for="inputtext2">Password:</label>
    <input id="inputtext2" type="password" name="inputtext2" value="" size="20" />
    <input id="inputsubmit1" type="submit" name="inputsubmit1" value="Είσοδος" />
  </fieldset>
</form>
```

Εικόνα 69: Ταυτοποίηση Διαχειριστή

Η παραπάνω ταυτοποίηση υλοποιείται στη σελίδα loginadmin.php. Λαμβάνει τις παραμέτρους username και password από την σελίδα admin/index.php και εκτελεί ένα query προς τον πίνακα member της βάσης δεδομένων. Ελέγχει τα δύο αυτά πεδία και εάν είναι έγκυρα και ελέγχει το είδος χρήστη. Εάν είναι τύπος administrator τότε τον ανακατευθύνει στη σελίδα διαχείρισης και δημιουργεί μία session μεταβλητή με όνομα currentuser και παίρνει τιμή administrator. Σε διαφορετική περίπτωση παίρνει την τιμή του email του εξουσιοδοτημένου χρήστη. Το παρακάτω κομμάτι κώδικα μας εξηγεί όσα ειπώθηκαν παραπάνω

```

<?php
session_start();
include 'catalogdbinfo.php';
$username='';
$password='';
$username = addslashes($_POST['inputtext1' ]);
$password=addslashes($_POST['inputtext2']);
$query=mysql_query("select * from member where username='".$username."'");
if (mysql_num_rows($query)==0)
{
mysql_close($link);
?>
<script language="javascript">
window.alert("Μη αποδεκτό Username");
window.location="../index.php";
</script>
<?php }
else if (mysql_result($query,0,"pass")!= $password)
{
mysql_close($link);
?>
<script language="javascript">
window.alert("Μη αποδεκτό Password");
window.location="../index.php";
</script>
<?php
}
else
{
$_SESSION['currentuser']=$username ;
if (mysql_result($query,0,"usertype")== "admintp")
{
$_SESSION['usertype']="admintp" ;
mysql_close($link);
?>
<script language="javascript">
window.alert("Ο Administrator συνδέθηκε επιτυχώς");
window.location="admin.php";
</script>
<?php
}
else
{
?>
<script language="javascript">
window.alert("Συνδεθήκατε επιτυχώς");
window.location="../index.php";
</script>
<?php }
}
?>

```

Εικόνα 70: Ταυτοποίηση Διαχειριστή 2

Για να επιτραπεί ή να αποτραπεί η πρόσβαση στη σελίδα χρησιμοποιείται ο κώδικας:

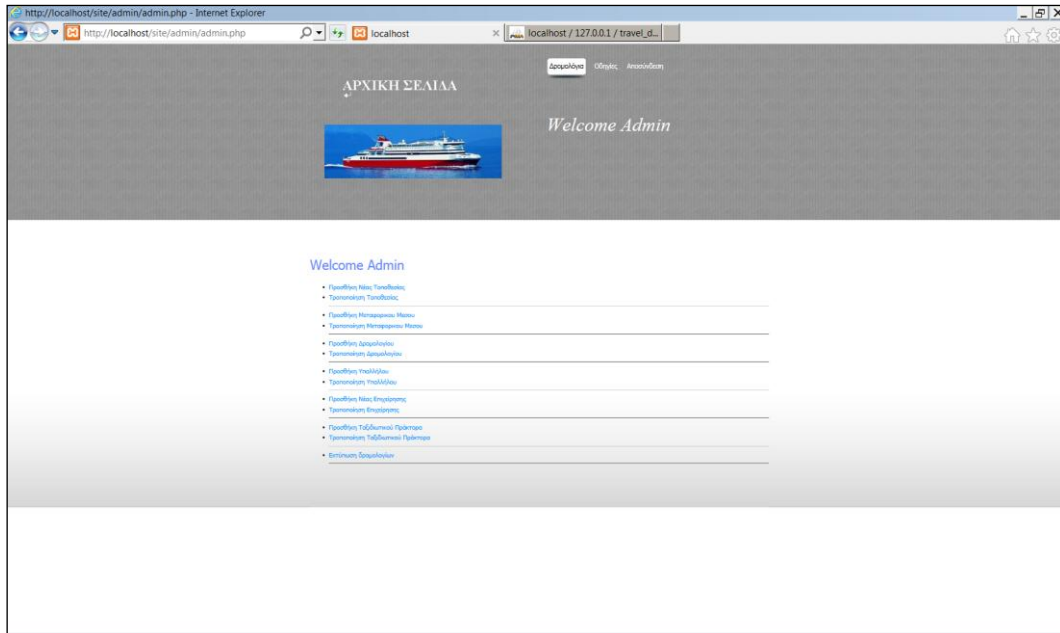
```

<?php
session_start();
if (!(isset($_SESSION['usertype']))) || ($_SESSION['usertype']!="admintp"){
header( 'Location: ../index.php');
}
include 'catalogdbinfo.php';
?>

```

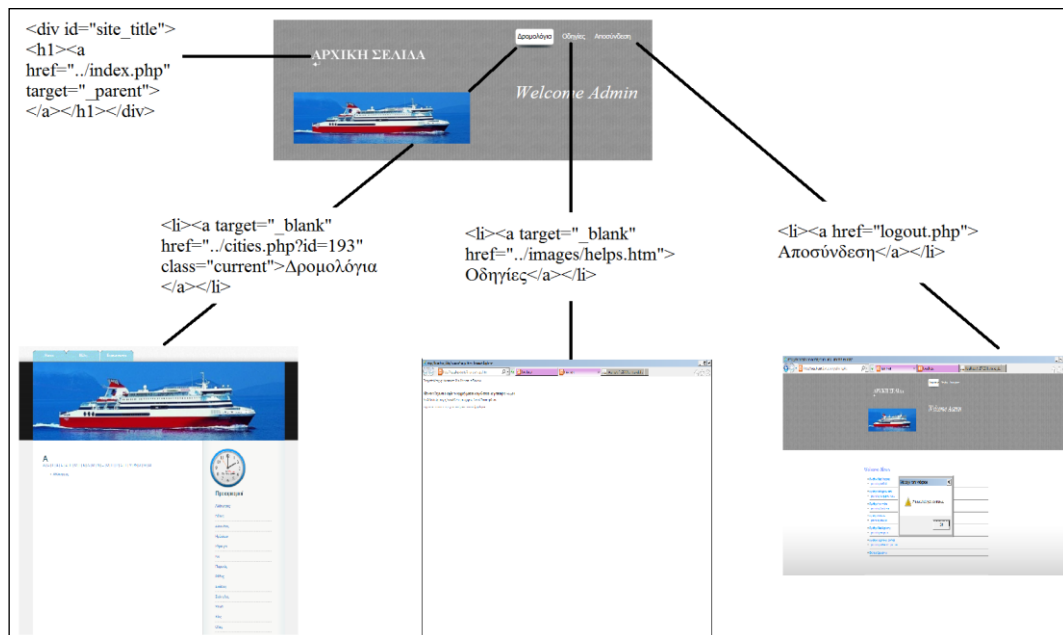
Εικόνα 71: Πρόσβαση ή Αποτροπή

Μετά την επιτυχή σύνδεση, μεταφερόμαστε στη σελίδα admin/admin.php.



Εικόνα 72: Admin/admin.php

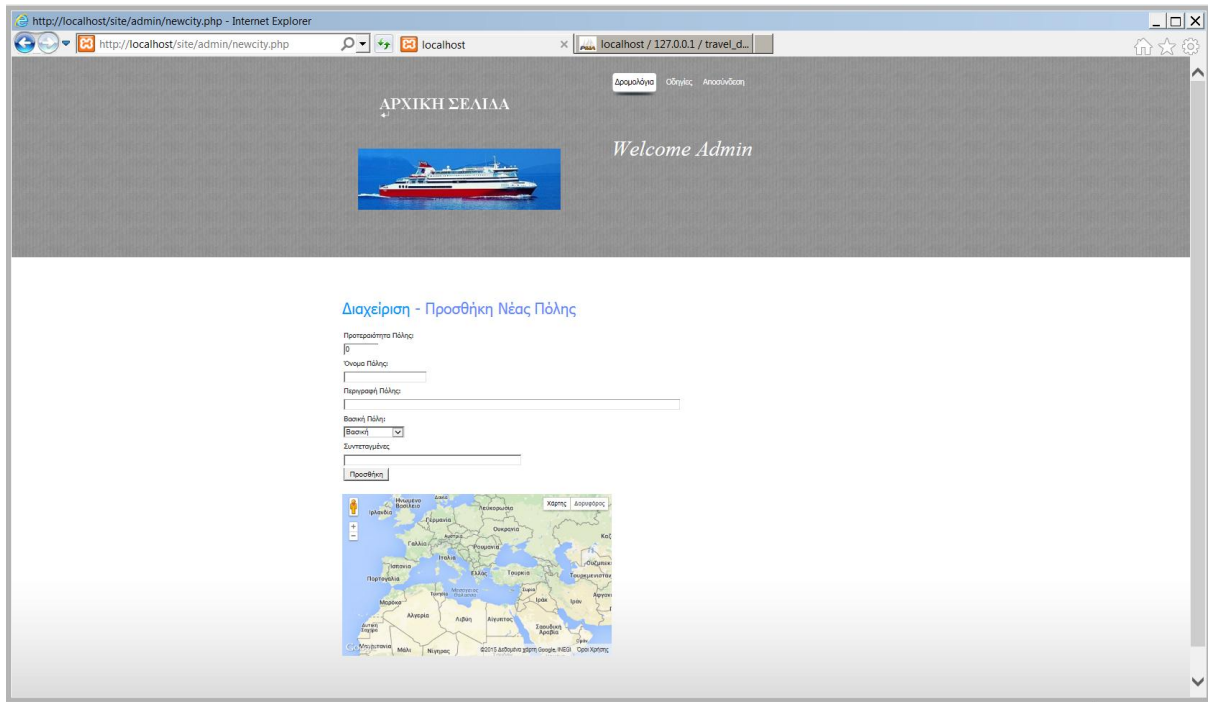
Στην παρακάτω εικόνα βλέπουμε τα link που μας οδηγούν στις αντίστοιχες σελίδες.



Εικόνα 73: Admins Link

Προσθήκη Νέας Τοποθεσίας

Σε αυτήν την σελίδα μπορούμε να προσθέσουμε μία νέα πόλη-λιμάνι. Το σημείο με το χάρτη, είναι φτιαγμένο με ένα javascript code, το οποίο μπορούμε να το λάβουμε από το google maps.



Εικόνα 74: Προσθήκη νέας πόλης

Ο κώδικας που χρησιμοποιεί το script φαίνεται παρακάτω

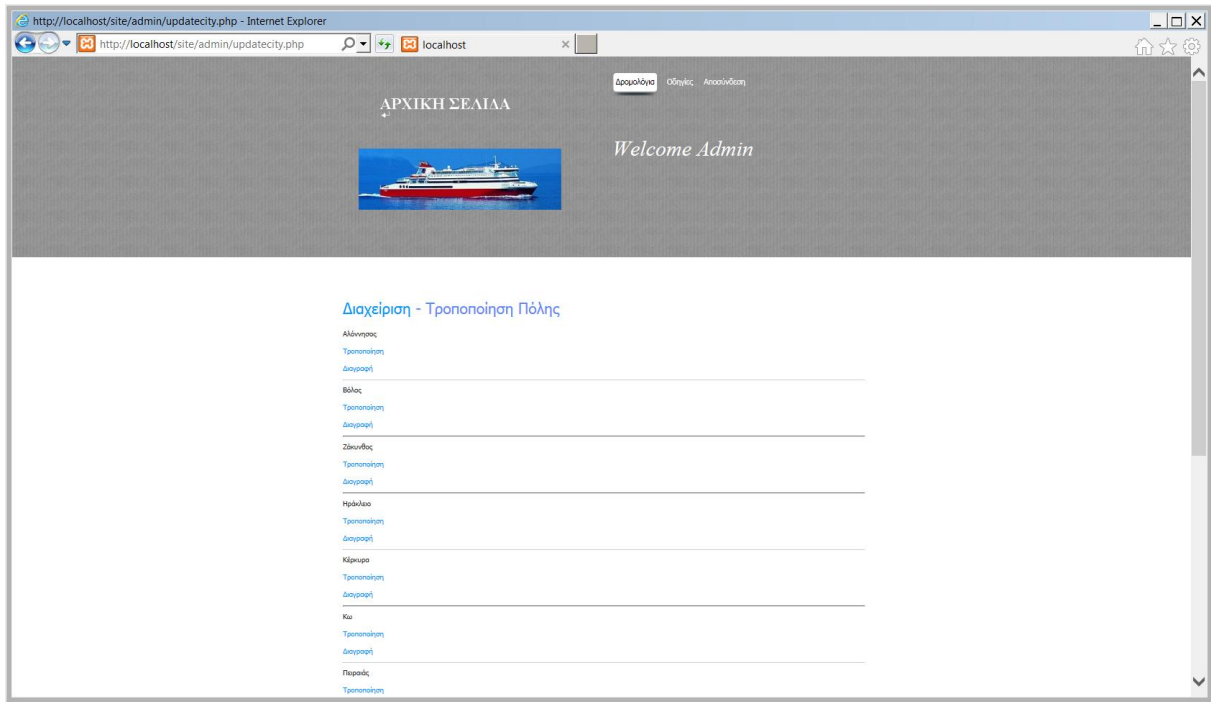
```
<script src="http://maps.google.com/maps?file=api&v=2&key=ABQIAAAAC
_mPwwhHzQIWXrkWZ6bSohSse_vYtllNt-Vk2RfqK4lP9fPV9BSbxjUe_ymjsTvnRNm8fsQJvVObxQ"
type="text/javascript"></script>
<script type="text/javascript">

function initialize() {
    var map = new GMap2(document.getElementById("map_canvas"));
    map.setCenter(new GLatLng(39.0618, 22.0605), 3);
    map.addControl(new GLargeMapControl());
    var mapControl = new GMapTypeControl();
    map.addControl(mapControl);
    GEvent.addListener(map, "click", function() {
        var center = map.getCenter();
        var photoxy=document.getElementById("photoXY");
        photoxy.value=center.toString();
    });
}
```

Εικόνα 75: Google maps

Τροποποίηση-Διαγραφή Τοποθεσίας

Βλέπουμε όλες τις διαθέσιμες πόλεις-λιμάνια και μπορούμε να κάνουμε τροποποίηση κάποιας ή να την διαγράψουμε.



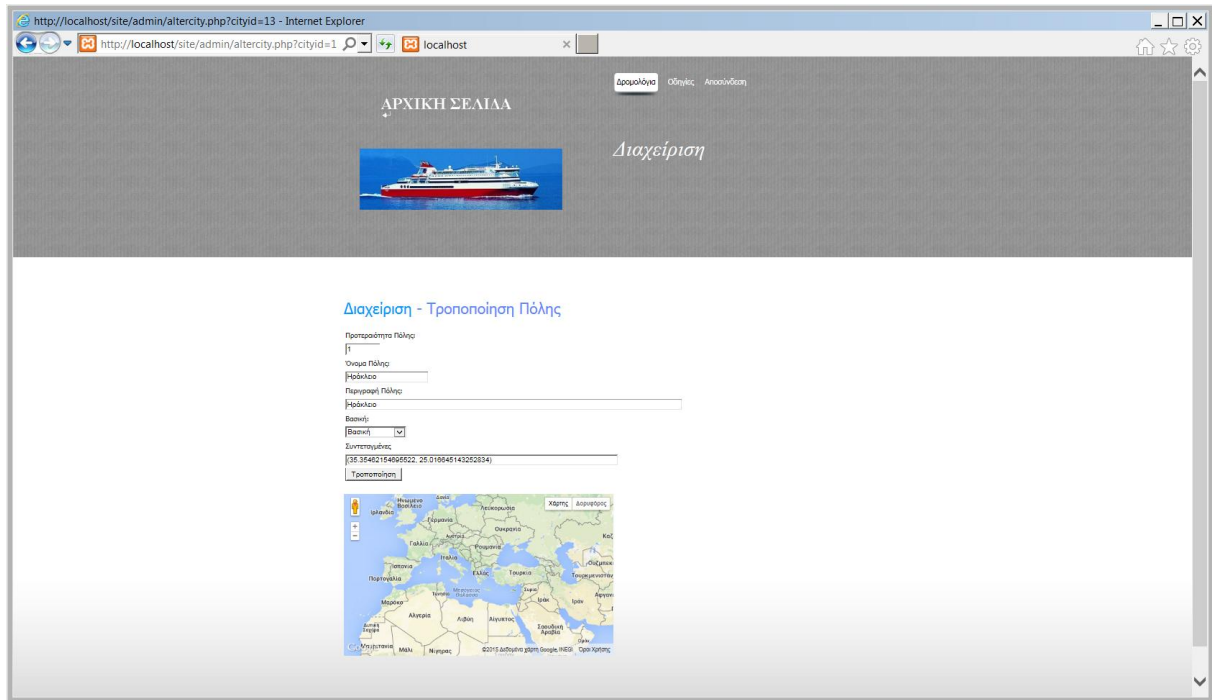
Εικόνα 76: Τροποποίηση πόλης

Ο παρακάτω κώδικας μας εμφανίζει τις πόλεις που έχουμε προσθέσει και τα links για την τροποποίηση και τη διαγραφή της κάθε μίας.

```

<?php $categories=mysql_query("select * from tblcity order by cityname");
    $i=0;
    while($i < mysql_num_rows($categories))
    {
        ?>
        <p><?php echo mysql_result($categories,$i,"cityname");?></p>
        <p><a href="altercity.php?cityid=<?php echo mysql_result($categories,$i,"cityid");
        ?>">Τροποποίηση</a></p>
        <p><a href="deletecity.php?cityid=<?php echo mysql_result($categories,$i,"cityid");
        ?>">Διαγραφή</a></p><hr/>
    <?php
        $i++;
    }
    ?>
    
```

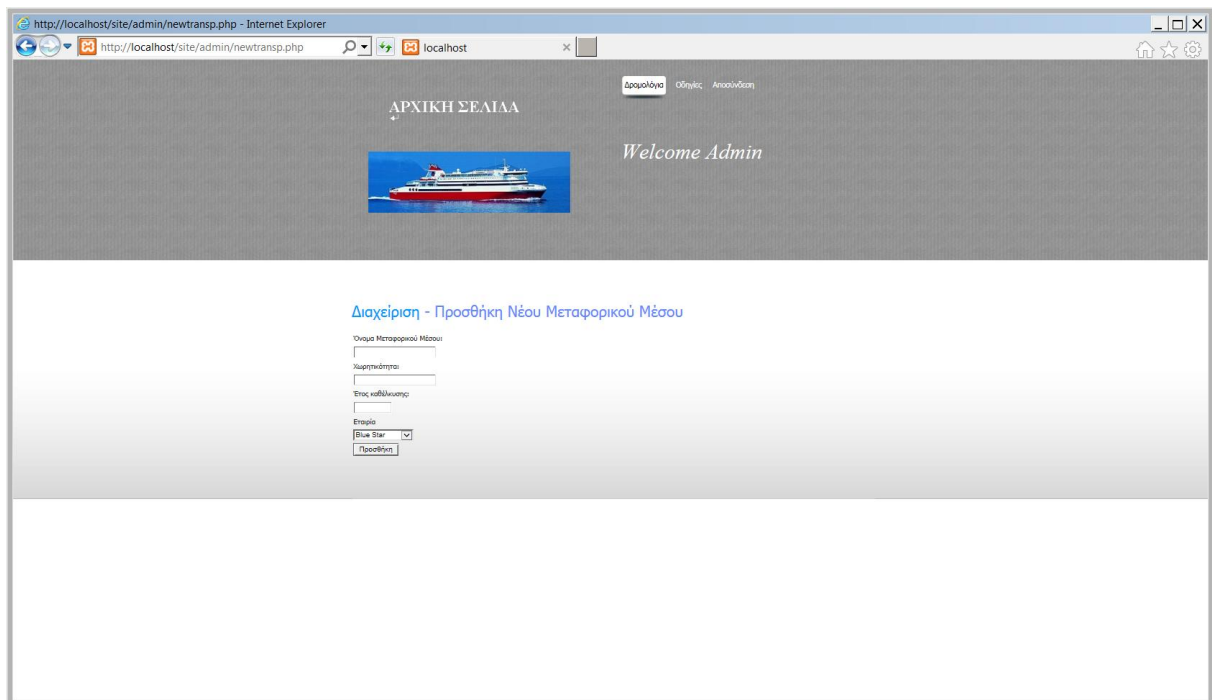
Εικόνα 77: Τροποποίηση-Διαγραφή Links



Εικόνα 78: Τροποποίηση πόλης 2

Προσθήκη Νέου Μεταφορικού Μέσου

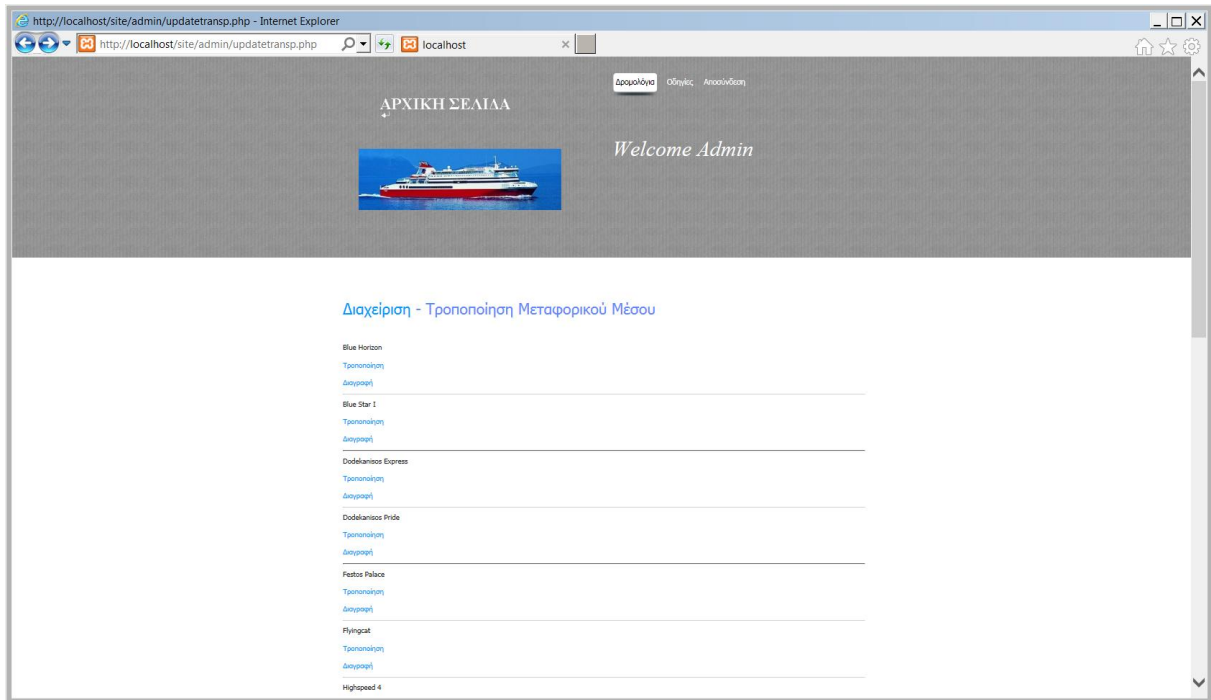
Στην παρακάτω σελίδα μπορούμε να προσθέσουμε ένα νέο πλοίο μιας υπάρχουσας ναυτιλιακής εταιρίας.



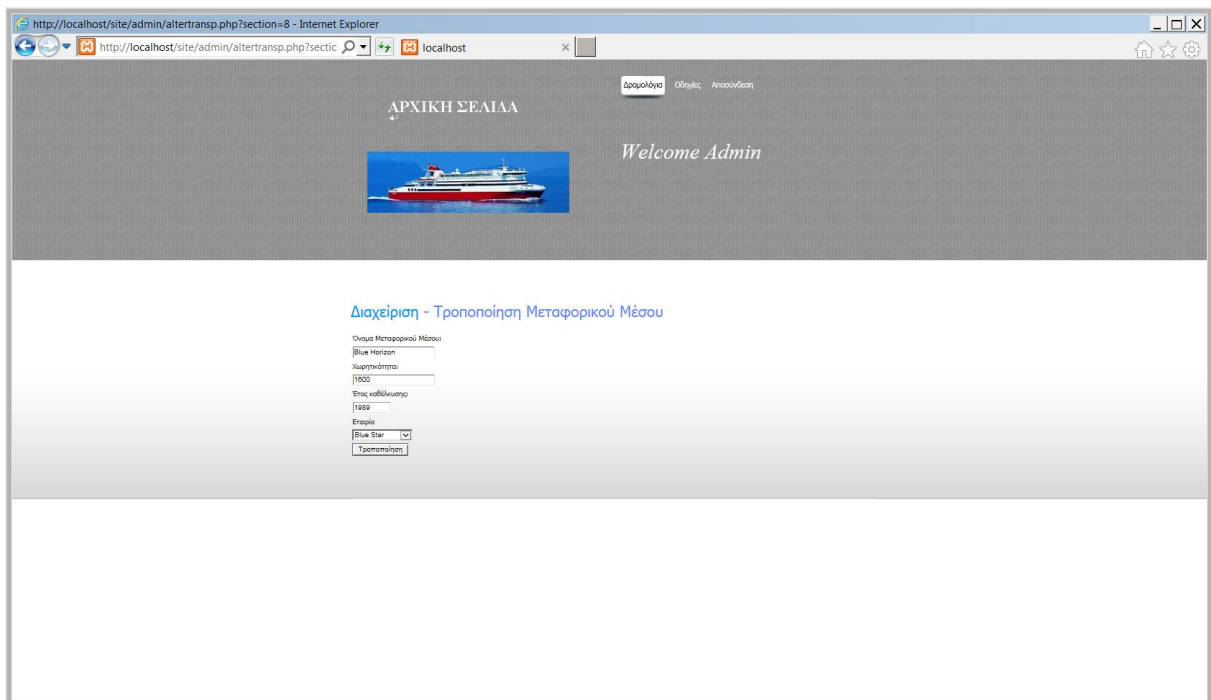
Εικόνα 79: Προσθήκη πλοίου

Τροποποίηση-Διαγραφή Μεταφορικού Μέσου

Μπορούμε να τροποποιήσουμε κάποιο πλοίο, όπως στην περίπτωση μίας νέας πόλης. Σε περίπτωση που προσθέσουμε ένα πλοίο με το ίδιο όνομα ενός πλοίου που έχουμε ήδη δημιουργήσει, τότε θα μας εμφανίσει ένα μήνυμα σφάλματος που θα μας ειδοποιεί ότι το πλοίο έχει ήδη συμπεριληφθεί.



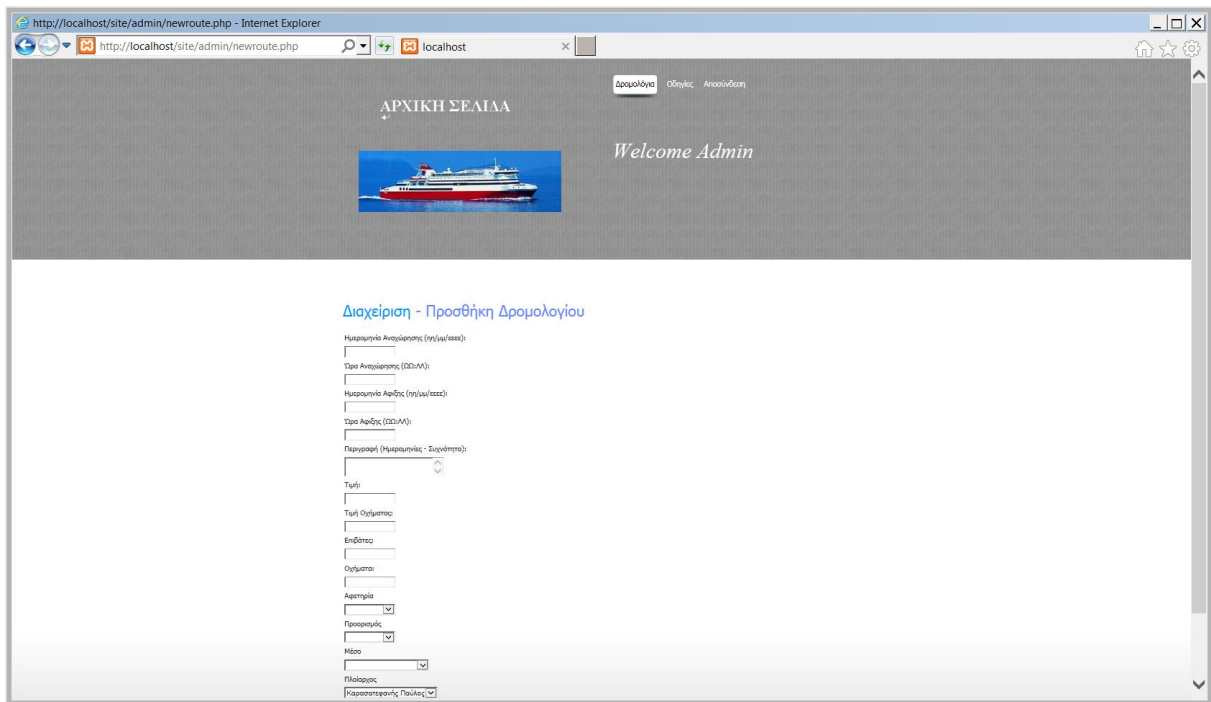
Εικόνα 80: Τροποποίηση πλοίου 1



Εικόνα 81: Τροποποίηση πλοίου 2

Προσθήκη Δρομολογίου

Σε αυτή τη σελίδα κάνουμε εισαγωγή ενός νέου δρομολογίου.



Εικόνα 82: Προσθήκη δρομολογίου

Χρησιμοποιώντας την ετικέτα <option> της HTML, δημιουργούμε τα listbox που είναι για την εισαγωγή ΑΦΕΤΗΡΙΑ, ΠΡΟΟΡΙΣΜΟΣ, ΜΕΣΟ, ΠΛΟΙΑΡΧΟΣ. Αυτό σαν κανόνας ότι δεν μπορεί να προστεθεί κάποιο από αυτά, το οποίο δεν υπάρχει στη βάση δεδομένων. Υπάρχουν όμως κάποιοι κανονισμοί σχετικά με την προσθήκη δρομολογίου και αυτοί είναι ο έλεγχος διαθεσιμότητας πλοίου δηλαδή ότι το πλοίο δεν εκτελεί κάποιο άλλο δρομολόγιο εκείνη την ώρα και ημερομηνία. Επίσης, γίνεται έλεγχος διαθεσιμότητας πλοίαρχου, δηλαδή δεν μπορούμε να προσθέσουμε έναν πλοίαρχο που είναι ήδη πλοίαρχος σε κάποιο άλλο δρομολόγιο. Τα παραπάνω υλοποιούνται με τον παρακάτω κώδικα:

```

$specs=mysql_query("select * from tblroute where transportation=".$transportation);
$i=0;
while($i<mysql_numrows($specs))
{echo mysql_result($specs,$i,'tblroute.departdate')." ".$departdate."
  ".mysql_result($specs,$i,'tblroute.arrivedate')." <br>";
if(($departdate<=mysql_result($specs,$i,'tblroute.arrivedate'))&&
($departdate>mysql_result($specs,$i,'tblroute.departdate')))
{
if(!(($departdate==mysql_result($specs,$i,'tblroute.arrivedate'))&&
(mysql_result($specs,$i,'tblroute.destination')==departure)&&
($departtime >mysql_result($specs,$i,'tblroute.arrivetime'))))
  $f=true;
}
if(($arrivedate<=mysql_result($specs,$i,'tblroute.arrivedate'))&&
($arrivedate>mysql_result($specs,$i,'tblroute.departdate')))
{
if(!(($arrivedate==mysql_result($specs,$i,'tblroute.departdate'))&&
(mysql_result($specs,$i,'tblroute.departure')==destination)&&
($arrivetime <mysql_result($specs,$i,'tblroute.departtime'))))
  $f=true;
}
}
}

```

Εικόνα 83: Διαθεσιμότητα πλοίου 1

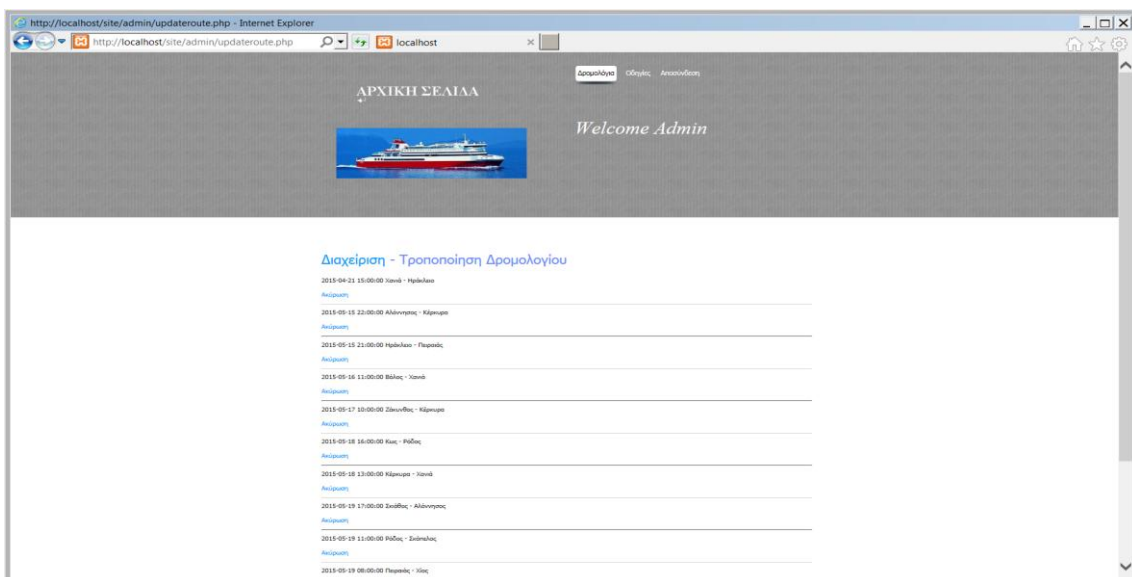
```

$specs=mysql_query("select * from tblroute where empid=".$sempid);
$i=0;
while($i<mysql_numrows($specs))
{
echo mysql_result($specs,$i,'tblroute.departdate')." ".$sempid." "
.mysql_result($specs,$i,'tblroute.arrivedate')."<br>";
if(($departdate<mysql_result($specs,$i,'tblroute.arrivedate'))&&
($departdate>mysql_result($specs,$i,'tblroute.departdate')))
{
if(!(($departdate==mysql_result($specs,$i,'tblroute.arrivedate'))&&
(mysql_result($specs,$i,'tblroute.destination')==departure)&&
($departtime >mysql_result($specs,$i,'tblroute.arrivetime'))))
    $f=true;
}
if(($arrivedate<mysql_result($specs,$i,'tblroute.arrivedate'))&&
($arrivedate>mysql_result($specs,$i,'tblroute.departdate')))
{
if(!(($arrivedate==mysql_result($specs,$i,'tblroute.departdate'))&&
(mysql_result($specs,$i,'tblroute.departure')==destination)&&
($arrivetime <mysql_result($specs,$i,'tblroute.departtime'))))
    $f=true;
}
}
if($f==true)
{
echo "<script type='text/javascript'>alert(' Το πλοίο ή ο πλοίαρχος
χρησιμοποιείται σε άλλο δρομολόγιο');</script>";
}
else
{
mysql_query("insert into tblroute(price,vprice,departure,destination,
description,departdate,arrivedate,departtime,arrivetime,transportation,
vehicles,passengers,empid) values (". $price.", ". $vprice.", "
.$departure.", ". $destination.", ". $description.", ". $departdate.", "
.$arrivedate.", ". $departtime.", ". $arrivetime.", "
.$transportation.", ". $vehicles.", ". $passengers.", ". $empid.")");
echo "<script type='text/javascript'>alert(' Η διαδρομή
προστέθηκε επιτυχώς');</script>";
}
}

```

Εικόνα 83: Διαθεσιμότητα πλοίου 2

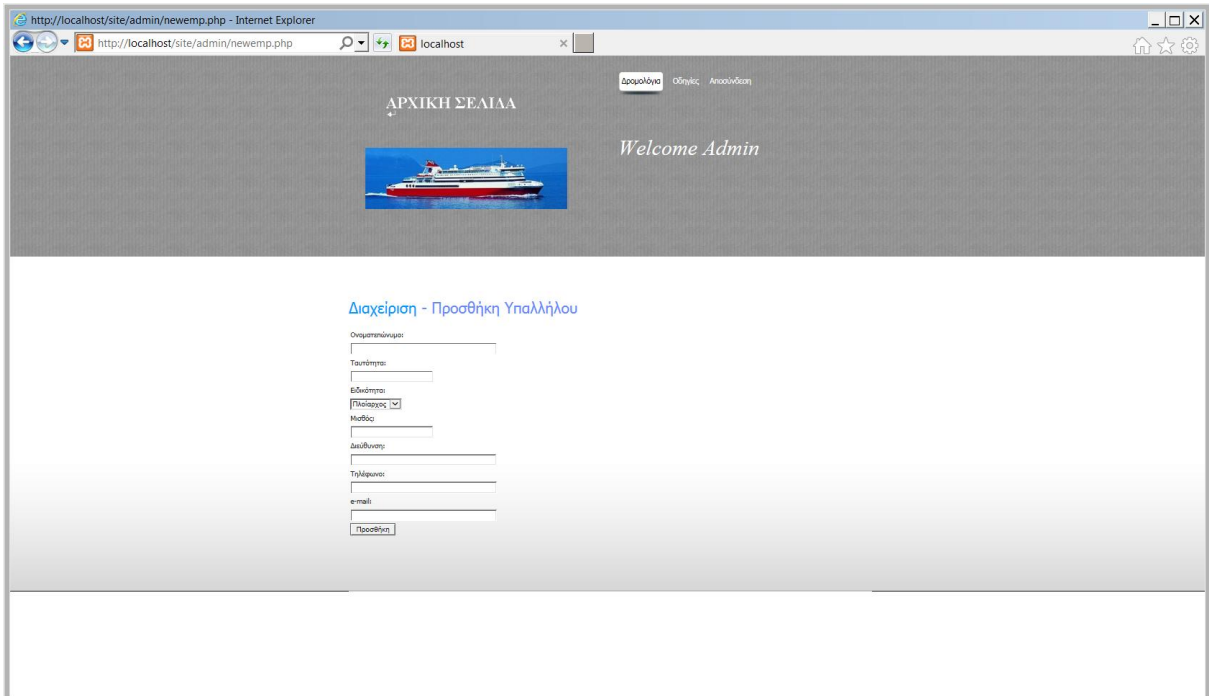
Τροποποίηση-Διαγραφή Δρομολογίου



Εικόνα 84: Τροποποίηση δρομολογίου

Προσθήκη Υπαλλήλου

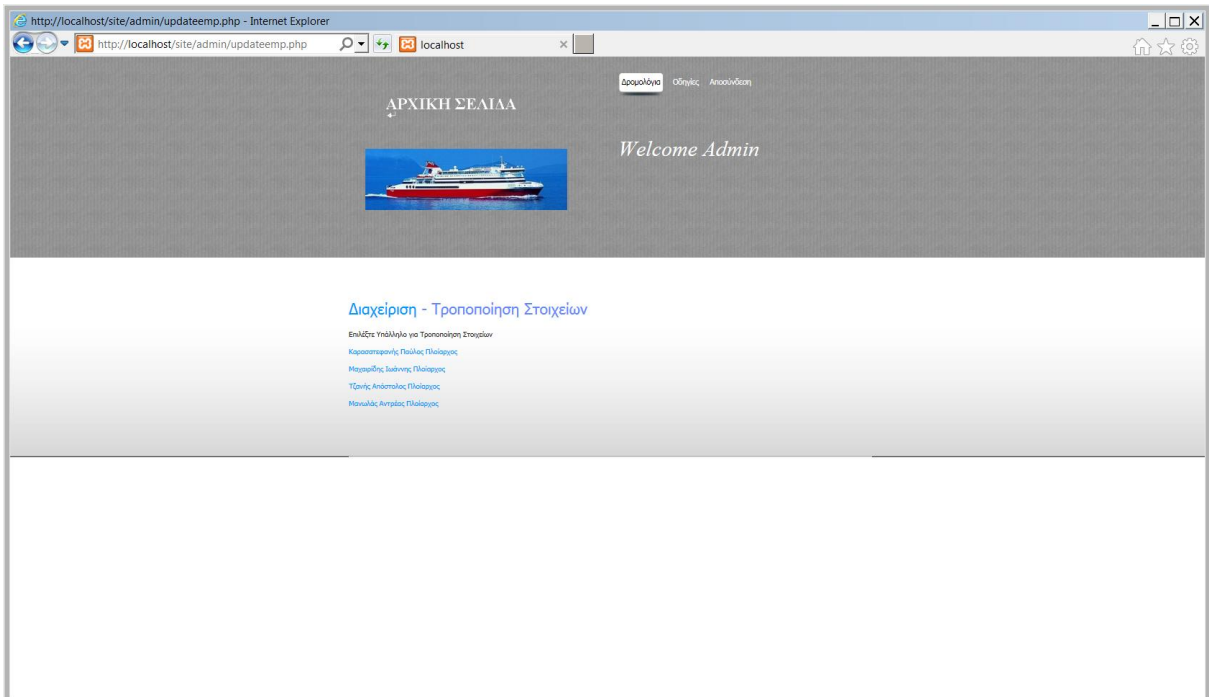
Εδώ προσθέτουμε έναν νέο υπάλληλο και μπορούμε να ορίσουμε εάν θα είναι πλοίαρχος ή απλός υπάλληλος με τη μέθοδο που χρησιμοποιήθηκε και για την προσθήκη δρομολογίου.



The screenshot shows a web browser window with the URL `http://localhost/site/admin/newemp.php`. The page header includes the text "ΑΡΧΙΚΗ ΣΕΛΙΔΑ" and "Welcome Admin" next to a ship image. The main content area is titled "Διαχείριση - Προσθήκη Υπαλλήλου" and contains a form with the following fields: "Ονοματεπώνυμο:", "Ταυτότητα:", "Ειδικότητα:" (with a dropdown menu showing "Πλοίαρχος"), "Μισθός:", "Διαθέσιμο:", "Τηλέφωνο:", "e-mail:", and a "Προσθήκη" button.

Εικόνα 85: Προσθήκη υπαλλήλου

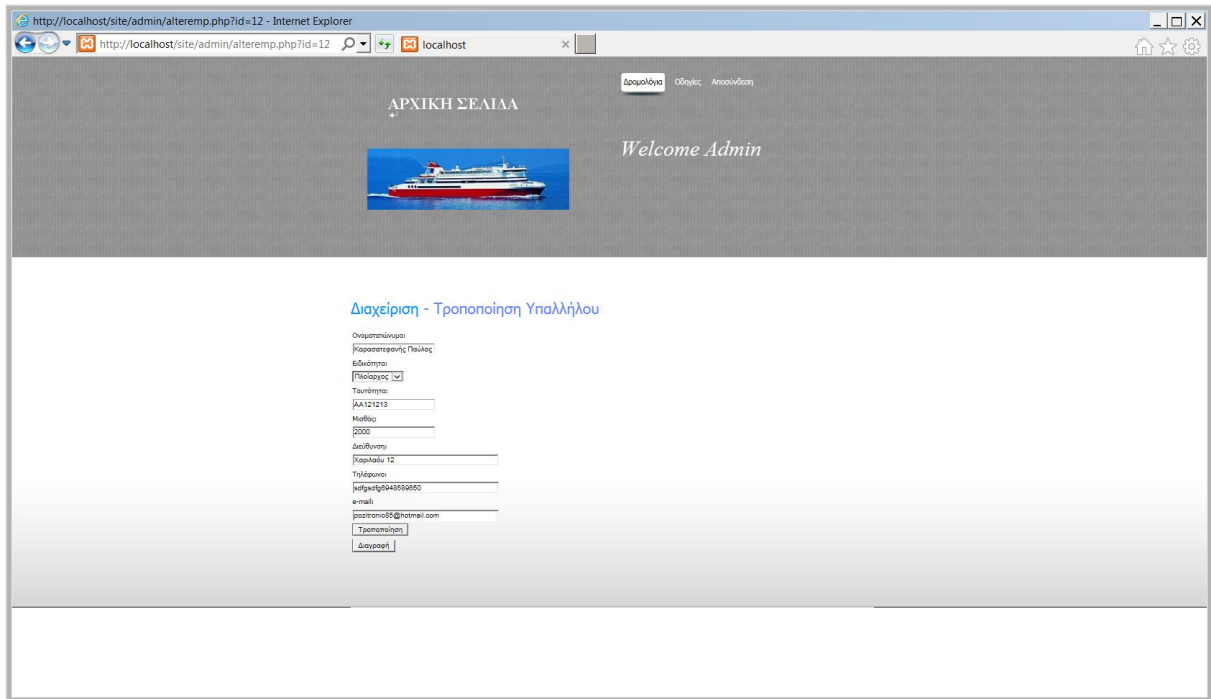
Τροποποίηση-Διαγραφή Υπαλλήλου



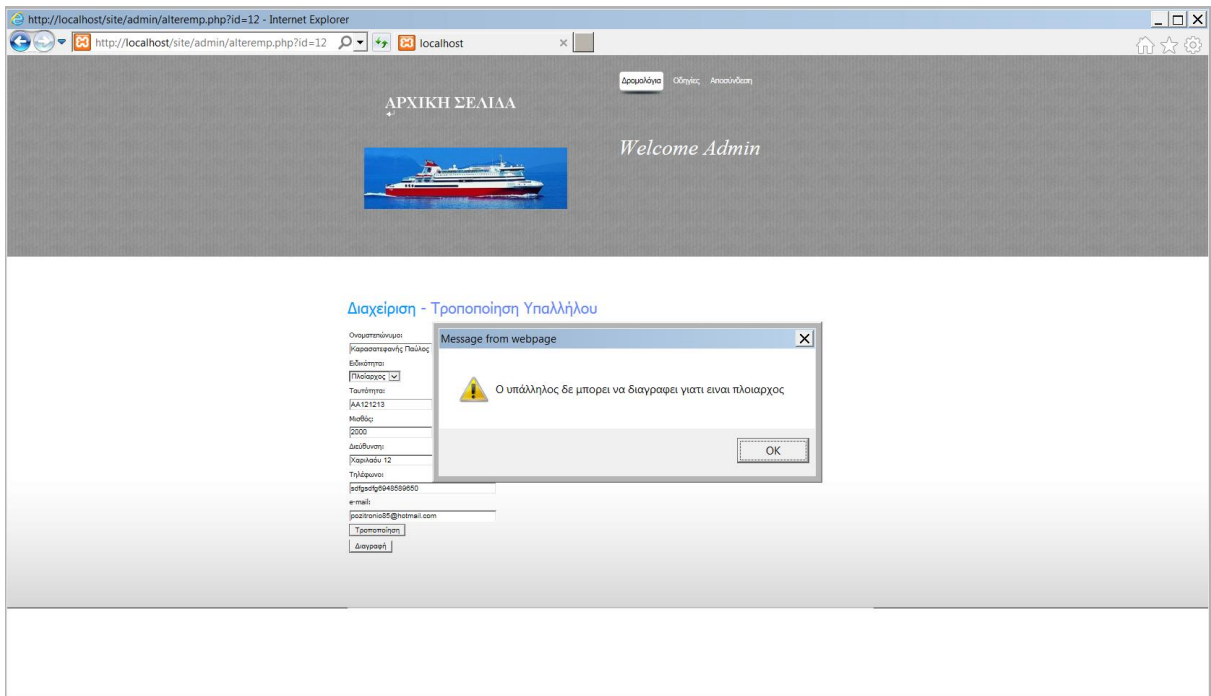
The screenshot shows a web browser window with the URL `http://localhost/site/admin/updateemp.php`. The page header is identical to the previous screenshot. The main content area is titled "Διαχείριση - Τροποποίηση Στοιχείων" and contains the text "Επιλέξτε το όνομα για τροποποίηση Στοιχείων" followed by a list of employee names: "Καραστεφανίλης Πλοίας Πλοίαρχος", "Μαρομπίλης Ιωάννης Πλοίαρχος", "Τζάνης Απόστολος Πλοίαρχος", and "Μουναλός Αντρέας Πλοίαρχος".

Εικόνα 86: Τροποποίηση υπαλλήλου 1

Ο κανόνας ακεραιότητας που διέπει αυτή τη σελίδα, είναι ότι δεν μπορεί να διαγραφεί ένας υπάλληλος σε περίπτωση που έχει οριστεί πλοίαρχος σε κάποιο άλλο δρομολόγιο.



Εικόνα 87: Τροποποίηση υπαλλήλου 2



Εικόνα 88: Διαγραφή υπαλλήλου

Ο παρακάτω κώδικας απαγορεύει την διαγραφή ενός πλοιάρχου που υπάρχει σε κάποιο δρομολόγιο.

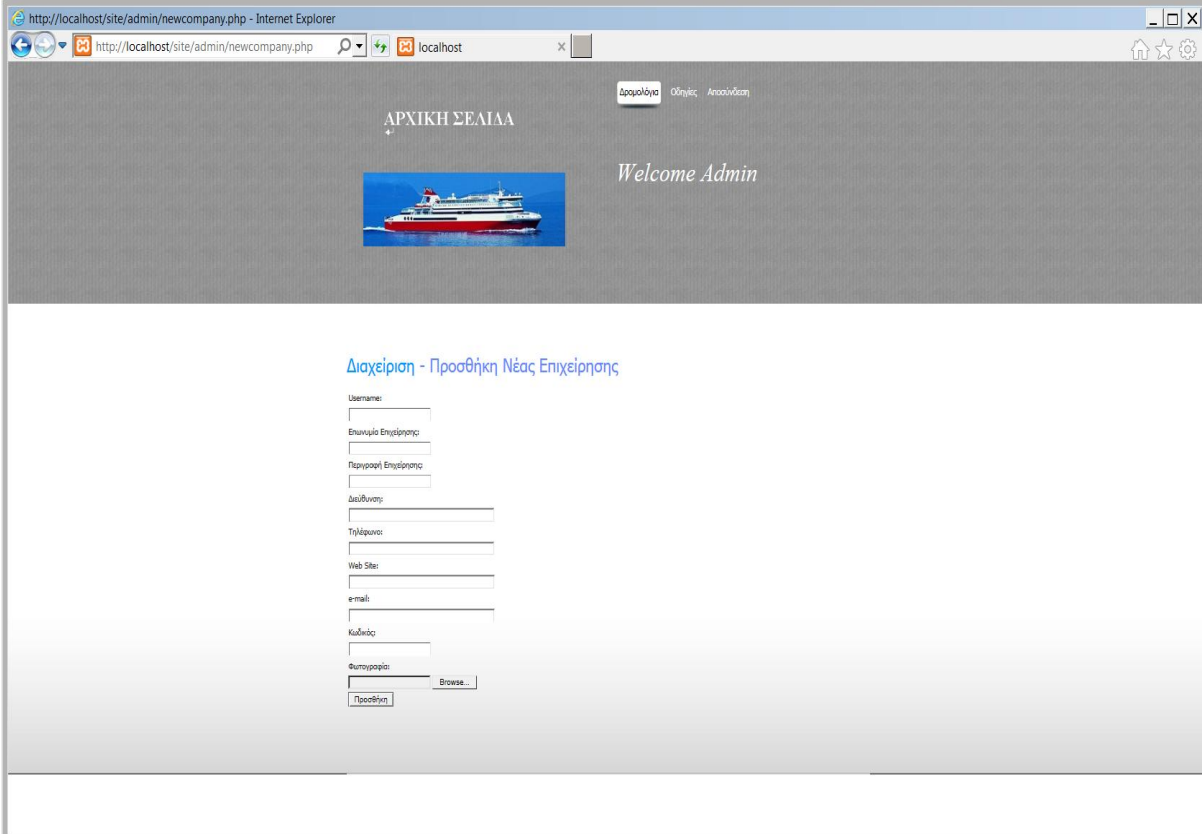
```

$cid = $_POST['id'];
include 'catalogdbinfo.php';
//an einai ploiarxos den diagrafetai
$specs=mysql_query("select speciality from tblemployee where id=".$cid);
$captain=mysql_query("select empid from tblroute where empid=".$cid);
if((mysql_result($specs,0,'speciality')=="Πλοίαρχος") &&
(mysql_numrows($captain)>0))
echo "<script type='text/javascript'>alert('Ο υπάλληλος δε μπορεί
να διαγραφεί γιατί είναι πλοίαρχος');</script>";
else
{
//allios diagrafetai
mysql_query("delete from tblemployee where id=".$cid);
echo "<script type='text/javascript'>alert(' Ο υπάλληλος διαγράφηκε επιτυχώς');</script>"; }
mysql_close($link);
?>

```

Εικόνα 89: Μη διαγραφή πλοιάρχου

Προσθήκη Ναυτιλιακής Εταιρίας



http://localhost/site/admin/newcompany.php - Internet Explorer

http://localhost/site/admin/newcompany.php localhost

ΑΡΧΙΚΗ ΣΕΛΙΔΑ

Welcome Admin

Διαχείριση - Προσθήκη Νέας Επιχείρησης

Username:

Επικοινωνία Επιχείρησης:

Περιγραφή Επιχείρησης:

Διεύθυνση:

Τηλέφωνο:

Web Site:

e-mail:

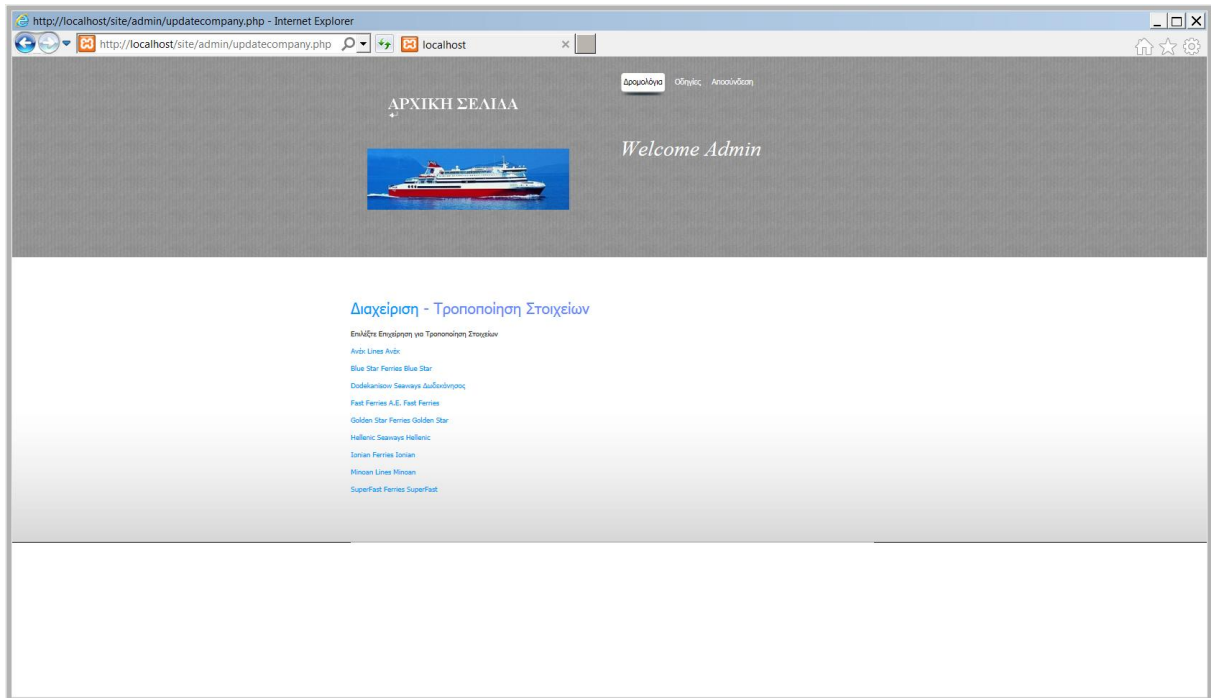
Κωδικός:

Όνομα Επιχείρησης: Browse...

Προσθήκη

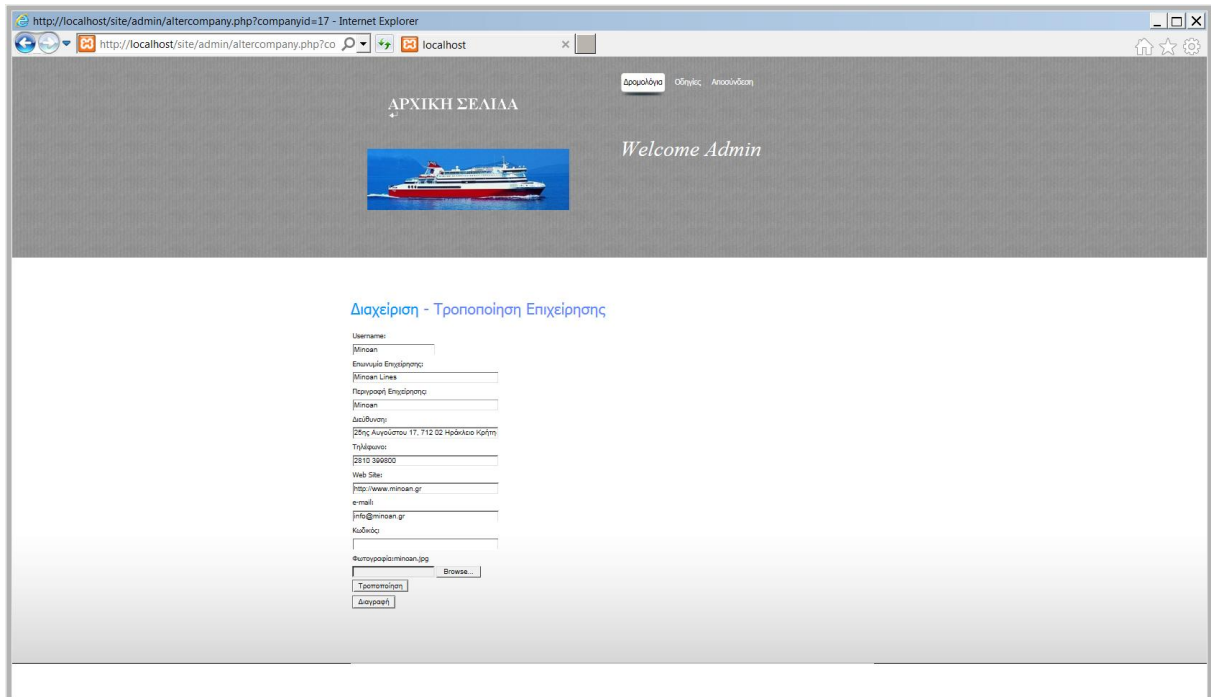
Εικόνα 90: Προσθήκη ναυτιλιακής εταιρίας

Τροποποίηση-Διαγραφή Ναυτιλιακής Εταιρίας



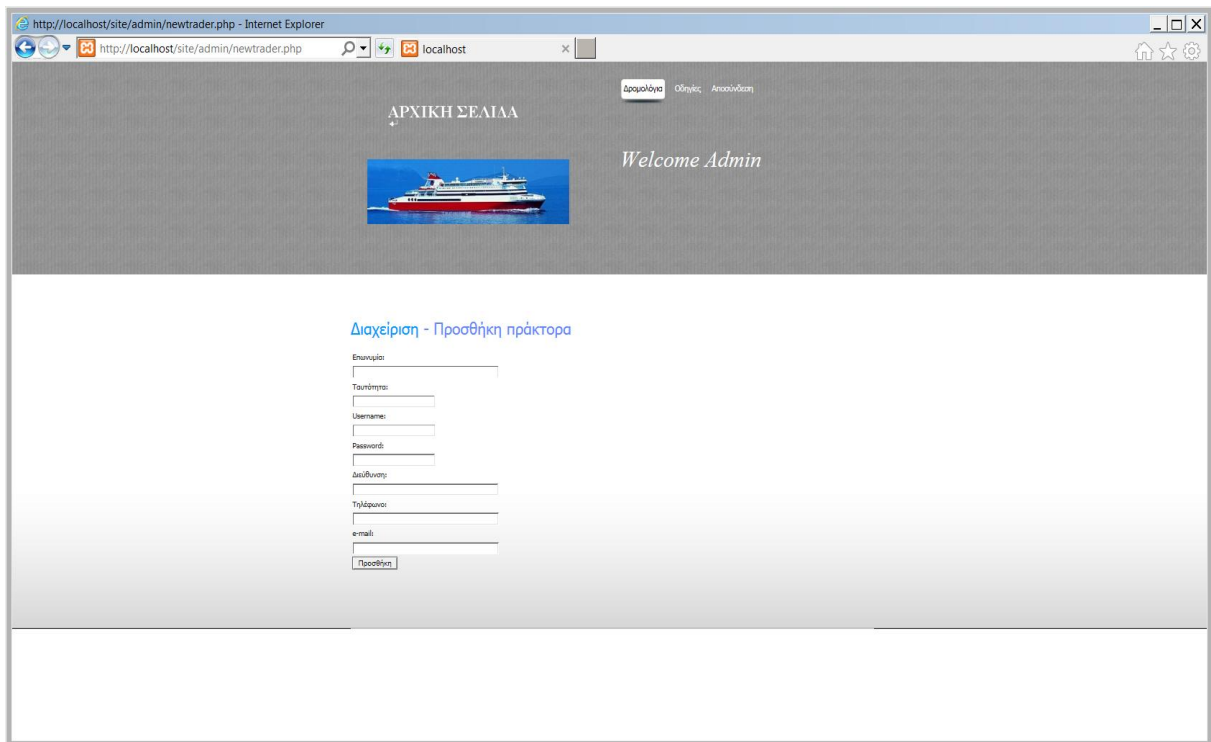
Εικόνα 91: Τροποποίηση ναυτιλιακής εταιρίας 1

Όπως στις περιπτώσεις του πλοιάρχου και των πλοίων, δεν μπορεί να διαγραφεί μία επιχείρηση, η οποία έχει ένα πλοίο, το οποίο είναι δεσμευμένο σε κάποιο δρομολόγιο.



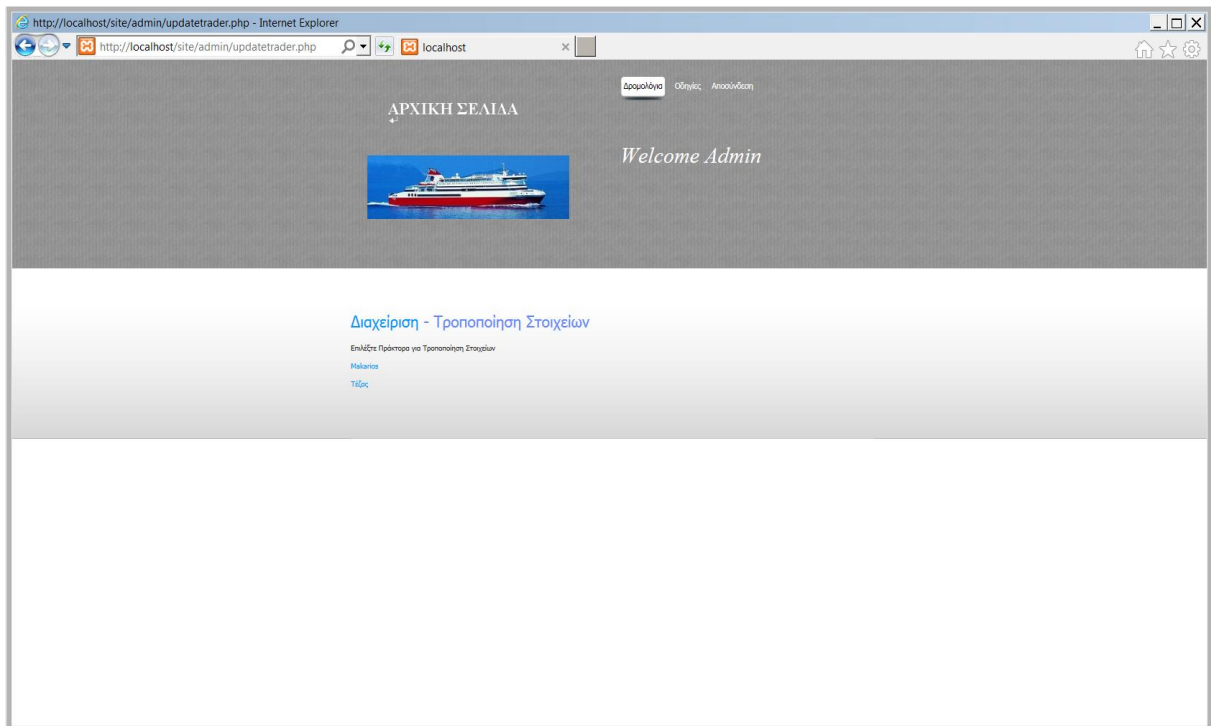
Εικόνα 92: Τροποποίηση ναυτιλιακής εταιρίας 2

Προσθήκη Ταξιδιωτικού Πράκτορα

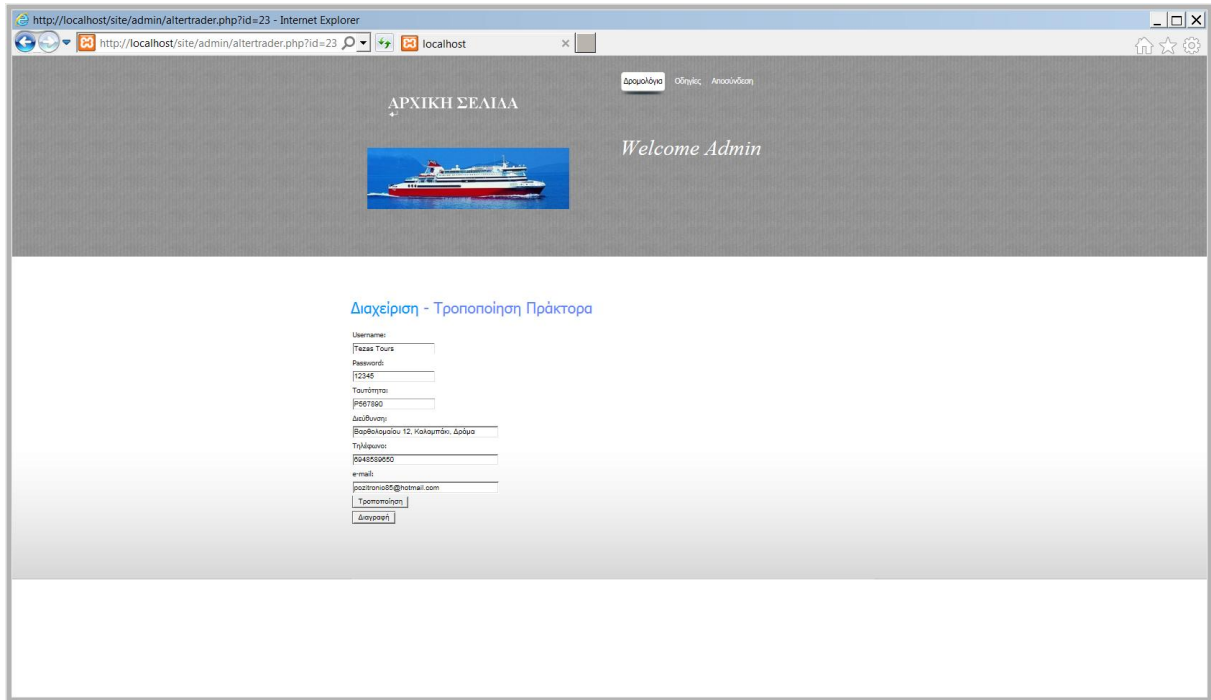


Εικόνα 93: Προσθήκη ταξιδιωτικού πράκτορα

Τροποποίηση-Διαγραφή Ταξιδιωτικού Πράκτορα



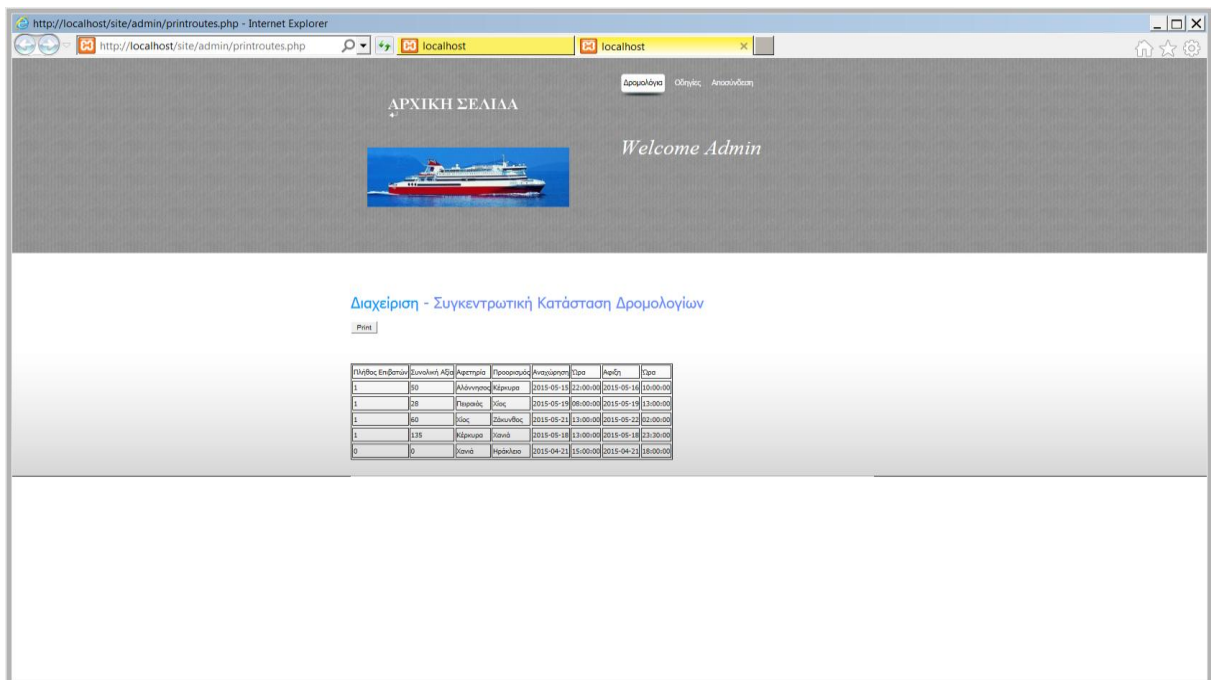
Εικόνα 94: Τροποποίηση ταξιδιωτικού πράκτορα 1



Εικόνα 95: Τροποποίηση ταξιδιωτικού πράκτορα 2

Συγκεντρωτική Κατάσταση Δρομολογίων

Μπορούμε να εμφανίσουμε όλα τα δρομολόγια σε έναν πίνακα και να εκτυπώσουμε την σελίδα. Εμφανίζονται πληροφορίες σχετικά με την Αφετηρία, Προορισμό, Ώρα αναχώρησης, Ώρα άφιξης, Πλήθος επιβατών που έκαναν κράτηση και τη συνολική αξία από όλους τους χρήστες.



Εικόνα 96: Συγκεντρωτική κατάσταση δρομολογίων

Παρακάτω εναποθέτονται κάποια σημαντικά κομμάτια από τον κώδικα.

- Ο χρήστης δεν μπορεί να κάνει κράτηση σε ένα δρομολόγιο, αφού πρώτα έχει κάνει μία άλλη κράτηση στις οποίες συμπίπτουν οι ώρες και ημερομηνίες αναχώρησης.

```

$reservation=mysql_query("select * from tblroute where id=".$objectid);
$departdate=mysql_result($reservation,0,'tblroute.departdate');
$arrivedate=mysql_result($reservation,0,'tblroute.arrivedate');
$destination=mysql_result($reservation,0,'tblroute.destination');
$departure=mysql_result($reservation,0,'tblroute.departure');
$specs=mysql_query("select * from tblroute.tblreservation where tblroute.id=tblreservation.objectid
and memberid=".$_SESSION['memberid']);
$i=0;
$f=false;
while($i<mysql_numrows($specs))
{
if(($departdate<=mysql_result($specs,$i,'tblroute.arrivedate'))&&($departdate>=mysql_result($specs,$i,
'tblroute.departdate')))
{
if(!((($departdate==mysql_result($specs,$i,'tblroute.arrivedate'))&&(mysql_result($specs,$i,
'tblroute.destination')==departure)&&($departtime >mysql_result($specs,$i,'tblroute.arrivetime'))))
    $f=true;
}
}
if(($arrivedate<=mysql_result($specs,$i,'tblroute.arrivedate'))&&($arrivedate>=mysql_result($specs,$i,
'tblroute.departdate')))
{
if(!((($arrivedate==mysql_result($specs,$i,'tblroute.departdate'))&&(mysql_result($specs,$i,
'tblroute.departure')==destination)&&($arrivetime <mysql_result($specs,$i,'tblroute.departtime'))))
    $f=true;
}
}
$i++;
}
if($f==true)
{
echo "<script type='text/javascript'>alert(' Υπάρχει κράτηση σε άλλο δρομολόγιο');</script>";
}
else
mysql_query("insert into tblreservation(objectid,memberid,qty,vqty,price) values(".$objectid.", "
.$_SESSION['memberid'].", ".$spqty.", ".$svqty.", ".$price.")");
echo "<script type='text/javascript'>alert(' Η κράτηση καταχωρήθηκε επιτυχώς');</script>";
}
else
{
mysql_query("insert into tblreservation(objectid,traderid,qty,vqty,price) values(".$objectid.", "
.$_SESSION['memberid'].", ".$spqty.", ".$svqty.", ".$price.")");
echo "<script type='text/javascript'>alert(' Η κράτηση καταχωρήθηκε επιτυχώς');</script>";
}

```

Εικόνα 97: Διπλοκράτηση

- Σε περίπτωση που γίνει ακύρωση εντός εικοσιτεσσέρων ωρών, τότε υπάρχει επιβάρυνση 10% επί του εισιτηρίου.

```

<?php
session_start();
if (!(isset($_SESSION['usertype']))) || ($_SESSION['usertype'] != "traveler"){
header('Location: ../index.php');
}

$id = $_GET['id'];
include 'catalogdbinfo.php';

$dates=mysql_query("select tblroute.departdate , tblroute.departtime from tblreservation,
tblroute where tblreservation.objectid=tblroute.id and tblreservation.id=".$id);

$d1=mysql_result($dates,0,'tblroute.departdate')."T".mysql_result($dates,0,'tblroute.departtime');
$date1 = new DateTime($d1);
$date2 = new DateTime(date('Y-m-dTH:i:s'));

$diff = $date2->diff($date1);

$hours = $diff->h;
$hours = $hours + ($diff->days*24);

if ($hours<24)
{
echo "<script type='text/javascript'>alert(' Μη ενκαίρια ακύρωση. Χρέωση 10%');</script>";
mysql_query("update tblreservation set qty=0, qty=0, price=0.1*price where id=".$id);
}
else
{
mysql_query("delete from tblreservation where id=".$id);
}
mysql_close($link);
?>

```

Εικόνα 98: Επιβάρυνση ακύρωσης

- Αυτό το κομμάτι εμφανίζει τα λιμάνια με αλφαβητική σειρά.

```

<h2><?php echo chr($id);?></h2>
<a href="cities.php?id=193">Α</a>
<nbsp<a href="cities.php?id=194">Β</a>
<nbsp<a href="cities.php?id=195">Γ</a>
<nbsp<a href="cities.php?id=196">Δ</a>
<nbsp<a href="cities.php?id=197">Ε</a>
<nbsp<a href="cities.php?id=198">Ζ</a>
<nbsp<a href="cities.php?id=199">Η</a>
<nbsp<a href="cities.php?id=200">Θ</a>
<nbsp<a href="cities.php?id=201">Ι</a>
<nbsp<a href="cities.php?id=202">Κ</a>
<nbsp<a href="cities.php?id=203">Λ</a>
<nbsp<a href="cities.php?id=204">Μ</a>
<nbsp<a href="cities.php?id=205">Ν</a>
<nbsp<a href="cities.php?id=206">Ξ</a>
<nbsp<a href="cities.php?id=207">Ο</a>
<nbsp<a href="cities.php?id=208">Π</a>
<nbsp<a href="cities.php?id=209">Ρ</a>
<nbsp<a href="cities.php?id=211">Σ</a>
<nbsp<a href="cities.php?id=212">Τ</a>
<nbsp<a href="cities.php?id=213">Υ</a>
<nbsp<a href="cities.php?id=214">Φ</a>
<nbsp<a href="cities.php?id=215">Χ</a>
<nbsp<a href="cities.php?id=216">Ψ</a>
<nbsp<a href="cities.php?id=217">Ω</a>
<br/><br/>
<ul class="sb_list">
<?php $cities=mysql_query("select * from tblcity where
cityname like '".chr($id)."%' order by cityname");
$i=0;
while($i < mysql_num_rows($cities))
{
?>
<li><a href="city.php?cityid=
<?php echo mysql_result($cities,$i,"cityid");?>
<?php echo mysql_result($cities,$i,"cityname");?></a></li>

<?php
    $i++;
}
?>

```

Εικόνα 99: Πόλεις σε αλφαβητική σειρά

6 Αποτελέσματα

Η εκπόνηση της πτυχιακής εργασίας πραγματοποιήθηκε. Το αποτέλεσμα ήταν μία ολοκληρωμένη πτυχιακή εργασία, με εφαρμογή όλων των απαιτήσεων του καθηγητή και μερικές ακόμα προσθήκες από εμένα, για να κάνω ένα σωστό και όμορφο interface. Αρχικά δυσκολεύτηκα αρκετά για να μπω στο κλίμα των διαφόρων γλωσσών προγραμματισμού που χρησιμοποιήθηκαν, αλλά στην πορεία ξεκαθάρισαν πολλά πράγματα. Οι γλώσσες προγραμματισμού που χρησιμοποιήθηκαν έχουν την ίδια λογική και ήταν πιο εύκολα αφομοιώσιμες.

Η πτυχιακή μπορεί να ωφελήσει κάποιους φοιτητές ώστε να κάνουν μία περαιτέρω ανάλυση και σχεδίαση της εφαρμογής αλλά και κάποιους επιχειρηματίες που βρίσκονται στον τουριστικό κλάδο και θέλουν μία εφαρμογή για την επιχείρησή τους. Βέβαια, είναι δύσκολο να προσαρμοστεί σε πολλούς αλλά θα είναι εύκολο να παραγοντοποιηθεί ώστε να την δουλέψουν αυτοί που επιθυμούν. Το σίγουρο είναι πως έχει βοηθήσει εμένα, να μάθω αρκετά πράγματα που ήταν άγνωστα μέχρι εχθές και μου δίνεται η δυνατότητα να ασχοληθώ με κάτι παρόμοιο στο μέλλον.

6.1 Συμπεράσματα

Καθώς η εργασία έφτασε στο τέλος της, τα συμπεράσματα που βγάζουμε από την όλη προσπάθεια είναι πολύ θετικά και αν μη τη άλλο, αναμενόμενα(ως προς το ποσοστό επιτυχίας). Οι απαιτήσεις που τέθηκαν από τον καθηγητή ήταν ξεκάθαρες, αν και κάποια σημεία υλοποιήθηκαν με αρκετή δυσκολία, και η εφαρμογή δουλεύει με όλες τις απαιτήσεις του θέματος. Για περισσότερα συμπεράσματα, ανατρέξτε στο κεφάλαιο 5, που ουσιαστικά είναι η υλοποίηση του project.

6.2 Μελλοντική Εργασία και Επεκτάσεις

Η εφαρμογή φτιάχτηκε με τέτοιο τρόπο ώστε να δοθεί η δυνατότητα επέκτασής της στο μέλλον. Είναι μία δυναμική εφαρμογή και μπορεί να γίνουν εύκολα αλλαγές μέσα στον κώδικα της. Μία πολύ σημαντική αλλαγή που μπορεί να γίνει είναι να αλλάξει το interface και να γίνει πιο σύγχρονο. Μπορεί να δημιουργηθεί ένας χάρτης, οπού με βάση αυτόν και με μόνο μερικά κλικ θα μπορεί ο χρήστης να κλείσει εισιτήρια, γρηγορότερα και χωρίς να προκύψει κάποιο λάθος βιασύνης. Επίσης, θα είναι όμορφο να αλλάξει ο τρόπος που επιλέγουμε ημερομηνίες αναχώρησης και να τοποθετηθεί ένα ημερολόγιο, όπως αυτά που χρησιμοποιούν οι αεροπορικές εταιρίες.

Καθώς πραγματοποιηθούν όλα τα παραπάνω, τότε στην εφαρμογή μπορεί να προστεθεί ένα σύστημα αγοράς εισιτηρίου μέσω πιστωτικής κάρτας, χρεωστικής, paypal κα. Αυτό απαιτεί χρόνο και πολύ εξάσκηση ώστε να επιτευχτεί. Εάν φτάσει μέχρι αυτό το σημείο, το τελικό βήμα προόδου που μπορεί να γίνει στην εφαρμογή, είναι να γίνει απευθείας σύνδεση με τη βάση δεδομένων των ακτοπλοϊκών εταιριών και να κλείνεται αμέσως το εισιτήριο.

Βιβλιογραφία

Xampp

<https://blog.udemy.com/xampp-tutorial/>
<http://www.slideshare.net/rodotheos/xampp>
<https://www.apachefriends.org/index.html>
<http://en.wikipedia.org/wiki/XAMPP>

PHP

<https://www.pinterest.com/pin/424605071090254004/>
<http://www.tutor.edu.gr/index.php/php>
<http://elearning.test4u.eu/el/getstarted/php/lesson/1/step/1>
http://www2.csd.uoc.gr/hy360/2014/project/php_mysql.pdf
<http://openspot.antithesis.gr/archives/8>
<http://dide.flo.sch.gr/Plinet/plinet.html>
<http://ti-einai.gr/php/>
<http://www.w3schools.com/php/default.asp>
<http://www.freestuff.gr/forums/viewtopic.php?t=19080>
<https://eclass.teicrete.gr/modules/document/document.php?course=TP327&openDir=/54ed92bd1r5H>

Βάσεις Δεδομένων

<https://eclass.teicrete.gr/courses/TP132/>
<http://slideplayer.gr/>
<http://www.infogineering.net/data-information-knowledge.htm>
<http://epoch.cs.berkeley.edu:8000/personal/jmh/cs286/lec1.htm>
<http://slideplayer.gr/slide/2921597/>
<http://www.tutor.edu.gr/index.php/databases>
<http://dide.flo.sch.gr/Plinet/plinet.html>
ftp://teiser.gr/pliforiki/Baseis_Dedomenon_I/A-Theory-slides.pdf
<http://www.cs.uoi.gr/~pitoura/courses/db/db11/slides/intro11.pdf>
E. F. Codd, "A Relational Model of
Data for Large Shared Data Banks",
Communications of the ACM, 13(6), 1970.

[Hell98] J. Hellerstein, Lecture Notes on Implementation of Database Systems, Univ. of California, College of Engineering, Department of EECS, Computer Science Division, CS286, Spring 1996. Available at:

Sql

<http://www.tutor.edu.gr/index.php/sqlintro>
<http://www.slideshare.net/neuromancer13/03-sql1?related=1>
<http://dide.flo.sch.gr/Plinet/plinet.html>
<http://www.freestuff.gr/tutorials/f/37/%C2%DC%F3%E5%E9%F2+%C4%E5%E4%EF%EC%DD%ED%F9%ED+%EA%E1%E9+SQL++%E3%E5%ED%E9%EA%DC>
<http://www.w3schools.com/sql/default.asp>
<http://dide.flo.sch.gr/Plinet/plinet.html>

HTML

<http://www.tutor.edu.gr/index.php/html>
<http://elearning.test4u.eu/el/getstarted/html/lesson/1/step/1>
<http://dide.flo.sch.gr/Plinet/plinet.html>
<http://www.freestuff.gr/tutorials/f/1/HTML+%EA%E1%E9+XHTML>
<http://www.w3schools.com/html/default.asp>

<http://el.wikipedia.org/wiki/HTML>
<http://www.wlearn.gr/index.php/html-html-32>
<http://www.wlearn.gr/index.php/home-xhtml>
<http://ti-einai.gr/html/>
<http://www.w3schools.com/html/default.asp>
<http://pacific.jour.auth.gr/html/>

Javascript

<http://www.tutor.edu.gr/index.php/javascript>
<http://elearning.test4u.eu/el/getstarted/javascript/lesson/1/step/1>
<http://dide.flo.sch.gr/Plinet/plinet.html>
<http://www.freestuff.gr/tutorials/f/14/JavaScript+%EA%E1%E9+Ajax>
<http://www.w3schools.com/js/default.asp>
<http://studentguru.gr/w/tutorials/a0-javascript-javascript-html>
<http://el.wikipedia.org/wiki/JavaScript>

CSS

<http://www.tutor.edu.gr/index.php/css>
<http://elearning.test4u.eu/el/getstarted/css/lesson/1/step/1>
<http://dide.flo.sch.gr/Plinet/plinet.html>
<http://www.freestuff.gr/tutorials/f/101/CSS+--+Cascading+Style+Sheets>
<http://www.w3schools.com/css/default.asp>
<http://www.wlearn.gr/index.php/home-css-83>

MySQL

http://webseminars.ee.auth.gr/presentations/11_MySQL_beginner.pdf
<http://elearning.test4u.eu/el/getstarted/mysql/lesson/1/step/1>
<http://en.wikipedia.org/wiki/MySQL>
<http://users.cs.uoi.gr/~itsoulos/ecommerce/php1.pdf>
<http://dide.flo.sch.gr/Plinet/plinet.html>
<http://www.freestuff.gr/tutorials/f/116/MySQL>
http://www.slideshare.net/neuromancer13/mysql-workbench-guide?next_slideshow=4
<http://www.google.gr/url?sa=t&rct=j&q=&esrc=s&source=web&cd=5&ved=0CD0QFjAE&url=http%3A%2F%2Fdide.flo.sch.gr%2FPlinet%2FTutorials%2FJavaScript-ProgrammingTechniques.pps&ei=-aYuVfyuKJb1aqbUgLGf&usq=AFQjCNFaBInoKYIbR4NSg2XhOQtsGXix5g&bvm=bv.90790515,d.d2s&cad=rja>

phpmyadmin

<http://docs.phpmyadmin.net/en/latest/intro.html>
http://www.phpmyadmin.net/home_page/
<http://en.wikipedia.org/wiki/PhpMyAdmin>