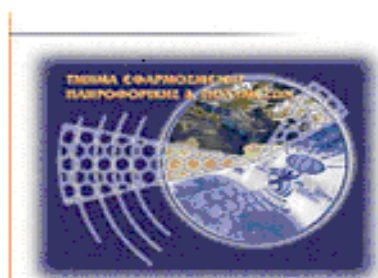




Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

**Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων**



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Τίτλος : On-Line Σύστημα για διαχείριση του τομέα ΕΥ
του ΓΤΘΕ (Γραμματεία)**

**Παντερής Βίκτωρας (Α.Μ.1700)
Σταθουδάκης Ιωάννης (Α.Μ.1941)**

Επιβλέπων καθηγητής : Παπαδάκης Νικόλαος

Ευχαριστίες

Με τη λήξη αυτού του ακαδημαϊκού κύκλου, θεωρούμε χρέος μας να ευχαριστήσουμε όλους όσους μας στήριξαν κατά τη διάρκεια της εκπόνησης της πτυχιακής εργασίας μας.

Οι θερμές μας ευχαριστίες απευθύνονται προς τον επιβλέποντα Καθηγητή κ. Νικόλαο Παπαδάκη για την αμέριστη συμπαράσταση, τις κατευθύνσεις που μας έδωσε σε σχετικές συναντήσεις, τις εποικοδομητικές ιδέες και τη γόνιμη κριτική και βοήθεια κατά την εκπόνηση αυτής της πτυχιακής εργασίας.

Θα θέλαμε επίσης να ευχαριστήσουμε, όλους τους καθηγητές μας στο τμήμα εφαρμοσμένης πληροφορικής & πολυμέσων, που με την μεθοδικότητά τους, συντέλεσαν τα μέγιστα στη διαδικασία της μάθησης.

Abstract

The purpose of this project is to create a dynamic web application for running and managing a university department at administration level. To implement this application we have used a variety of different softwares which where combined to achieve the final result. These softwares are freeware to use by anyone and are very easy to be found to the Internet. These softwares are: EasyPHP 1.8 and Adobe Dreamweaver CS4.

Σύνοψη

Σκοπός της εργασίας αυτής είναι η δημιουργία μιας δυναμικής διαδικτυακής εφαρμογής για την λειτουργία και διαχείριση μίας γραμματείας ενός τμήματος πανεπιστημίου σε επίπεδο διαχείρισης.

Για την υλοποίηση της εφαρμογής αυτής χρησιμοποιήθηκαν διάφορα λογισμικά τα οποία συνδυάστηκαν για την επίτευξη του τελικού αποτελέσματος. Τα λογισμικά αυτά είναι ελεύθερα προς χρήση στο ευρύ κοινό και βρίσκονται πολύ εύκολα στο διαδίκτυο. Τα λογισμικά αυτά είναι : EasyPHP 1.8 και Adobe Dreamweaver CS4.

Περιεχόμενα

Ευχαριστίες	2
Abstract	3
Σύνοψη	4
Κεφάλαιο 1	9
Εισαγωγή.....	9
1.1 Περίληψη.....	9
1.2 Κίνητρο για την Διεξαγωγή της Εργασίας.....	9
1.3 Σκοπός και Στόχοι της Εργασίας	10
1.4 Δομή Εργασίας.....	10
Κεφάλαιο 2.....	11
Μεθοδολογία Υλοποίησης	11
2.1 Στατικές και Δυναμικές Εφαρμογές.....	11
2.1.1 Στατικές Εφαρμογές.....	11
2.1.2 Δυναμικές Εφαρμογές.....	11
2.2 Web Σελίδες και HTML.....	12
2.2 Εργαλεία Δημιουργίας Δυναμικών Ιστοσελίδων.....	14
2.2.1 Joomla	14
2.2.2 Visual Basic.....	15
2.2.3 Drupal.....	17
2.2.4 XML.....	19
2.2.5 ASP.....	21
Κεφάλαιο 3.....	22
3.1 Εργαλεία Ανάπτυξης Δυναμικών Εφαρμογών.....	22
3.1.1 EasyPHP.....	22
3.1.2 Η γλώσσα PHP.....	23
3.1.3 Η Mysql.....	28
3.1.4 Apache Tomcat	30
3.1.5 Adobe Dreamweaver CS4.....	30
3.1.6 JavaScript	32
Κεφάλαιο 4.....	34
4.1 Ανάλυση Προβλήματος.....	34
4.2 Απαιτήσεις του Συστήματος	34
4.2.1 Βάσεις Δεδομένων	34
4.2.2 Τι παρέχει ένα Σύστημα Διαχείρισης Βάσης Δεδομένων.....	35
4.2.3 Το Μοντέλο Οντοτήτων-Συσχετίσεων.....	36
4.2.4 Χρήση μοντέλου οντοτήτων-συσχετίσεων	36
4.2.5 Σχεδίαση-ανάλυση μοντέλου οντοτήτων συσχετίσεων της εφαρμογής	37
4.3 Ανάλυση Πινάκων Βάσης Δεδομένων.....	42
4.3.1 Πίνακας Ανήκει.....	42
4.3.2 Πίνακας Διδάσκει.....	42
4.3.3 Πίνακας Φοιτητής	43
4.3.4 Πίνακας Καθηγητής	44
4.3.5 Πίνακας Μάθημα	45
4.3.6 Πίνακας Παίρνει.....	46
4.3.7 Πίνακας Προαπαιτούμενα.....	47
4.3.8 Πίνακας Πρόγραμμα Σπουδών.....	47
4.3.9 Εισαγωγή Κωδικών	48

4.3.10 Ο Πίνακας User	49
4.4 Η Εφαρμογή «Γραμματεία Πανεπιστημίου»	50
4.4.1 Σελίδα Εισόδου	50
4.4.2 Σελίδα Διαχείρισης.....	51
4.4.3 Σελίδα Βάσης Δεδομένων	51
4.4.4 Σελίδα Εγγραφής Φοιτητή	52
4.4.5 Σελίδα Εγγραφής Καθηγητή	54
4.4.6 Σελίδα Εισαγωγής Μαθήματος	55
4.4.7 Σελίδα Διαγραφής Φοιτητή	56
4.4.8 Σελίδα Διαγραφής Καθηγητή.....	58
4.4.9 Σελίδα Διαγραφής Μαθήματος	60
4.4.10 Σελίδα Προγράμματος Σπουδών	62
4.4.11 Σελίδα Αλλαγής Κωδικού	64
4.4.12 Σελίδα Δήλωσης Μαθημάτων.....	66
4.4.13 Σελίδα Εμφάνισης και εκτύπωσης αναλυτικής βαθμολογίας	68
Κεφάλαιο 5.....	70
5.1 Αποτελέσματα- Χρησιμότητα.....	70
Παράρτημα Α	71
Α.1 Ανάλυση Κώδικα SQL.....	71
Α.1.1 Ανάλυση Κώδικα Βάσης με όνομα Grammateia (Πίνακες 8)	71
Α.1.2 Ανάλυση Κώδικα Βάσης με όνομα Db Login (Πίνακες 1).....	74
Α.2 Ανάλυση Κώδικα PHP	75
Α.2.1 Αρχική Σελίδα Εισόδου.....	75
Α.2.2 Σελίδα Υποβολής και Ελέγχου Στοιχείων Χρήστη.....	76
Α.2.3 Σελίδα Εξόδου Εφαρμογής	77
Α.2.4 Σελίδα Εγγραφής Φοιτητή.....	77
Α.2.5 Σελίδα Εγγραφής Καθηγητή	83
Α.2.6 Σελίδα Εισαγωγής Μαθήματος	87
Α.2.7 Σελίδα Διαγραφής Φοιτητή	92
Α.2.8 Σελίδα Διαγραφής Καθηγητή.....	96
Α.2.9 Σελίδα Διαγραφής Μαθήματος	100
Α.2.10 Σελίδα Εμφάνισης Προγράμματος Σπουδών	104
Α.2.11 Σελίδα Αλλαγής Κωδικού Εισόδου.....	107
Α.2.12 Σελίδα Δήλωσης Μαθημάτων	111
Α.2.13 Σελίδα Εμφάνισης και εκτύπωσης αναλυτικής βαθμολογίας φοιτητή.....	117
Βιβλιογραφία.....	122

Πίνακας Εικόνων

Εικόνα 1 .Joomla Template.....	15
Εικόνα 2 Το λογισμικό Visual Basic	17
Εικόνα 3 Το λογότυπο Drupal.....	18
Εικόνα 4 Εφαρμογή Easy Php.....	22
Εικόνα 5 Σελίδα διαχείρισης της εφαρμογής EasyPHP	23
Εικόνα 6.Περιβάλλον διαχείρισης MySql.....	28
Εικόνα 7.Το λογισμικό Adobe DreamWeaver CS4.....	31
Εικόνα 8.Περιβάλλον Δημιουργίας Σελίδας Html.....	32
Εικόνα 9.Η Δομή της Βάσης ‘Γραμματεία’	39
Εικόνα 10.Περιήγηση Στον Πίνακα ‘Φοιτητής’	40
Εικόνα 11.Δομή του Πίνακα ‘Ανήκει’	42
Εικόνα 12.Δομή του Πίνακα ‘Διδάσκει’	43
Εικόνα 13.Δομή του Πίνακα ‘Φοιτητής’	43
Εικόνα 14.Δομή του Πίνακα ‘Καθηγητής’	44
Εικόνα 15.Δομή του Πίνακα ‘Καθηγητής’	44
Εικόνα 16.Δομή του Πίνακα ‘Μάθημα’	45
Εικόνα 17.Οι εγγραφές του Πίνακα ‘Μάθημα’	45
Εικόνα 18.Δομή του Πίνακα ‘Παίρνει’	46
Εικόνα 19.Δεδομένα του Πίνακα ‘Παίρνει’	46
Εικόνα 20.Δομή του Πίνακα Προσπαιτούμενα.....	47
Εικόνα 21.Δομή του Πίνακα ‘Πρόγραμμα Σπουδών’	48
Εικόνα 22.Η Βάση DbLogin	48
Εικόνα 23.Η Δομή του Πίνακα ‘User’	49
Εικόνα 24. Τα στοιχεία του Πίνακα User	49
Εικόνα 25.Σελίδα Κωδικού Εισόδου Χρήστη.....	50
Εικόνα 26 Αρχική Σελίδα Διαχείρισης	51
Εικόνα 27.Η αρχική σελίδα της Βάσης δεδομένων	51
Εικόνα 28.Φόρμα Εγγραφής Φοιτητή.....	52
Εικόνα 29 Σελίδα Επιβεβαίωσης Υποβολής Στοιχείων Εγγραφής.....	53
Εικόνα 30.Η Σελίδα Εγγραφής Στοιχείων Καθηγητή.....	54
Εικόνα 31.Σελίδα Επιβεβαίωσης Υποβολής Στοιχείων Εγγραφής Καθηγητή	54
Εικόνα 32. Σελίδα Εισαγωγής Στοιχείων Μαθήματος.....	55
Εικόνα 33.Σελίδα Υποβολής Στοιχείων Εγγραφής Μαθήματος.....	56
Εικόνα 34. Σελίδα Διαγραφής Στοιχείων Φοιτητή	57
Εικόνα 35.Έλεγχος Συμπλήρωσης Πεδίων Διαγραφής Φοιτητή	57
Εικόνα 36.Σελίδα Επιτυχούς Διαγραφής Φοιτητή	58
Εικόνα 37.Φόρμα διαγραφής στοιχείων Καθηγητή	58
Εικόνα 38 Μήνυμα ελέγχου ορθής εισαγωγής στοιχείων.....	59
Εικόνα 39.Μήνυμα επιτυχούς διαγραφής στοιχείων καθηγητή.....	59
Εικόνα 40.Σελίδα διαγραφής Μαθήματος	60
Εικόνα 41.Μήνυμα ελέγχου ορθής εισαγωγής στοιχείων.....	61
Εικόνα 42 Μήνυμα επιτυχούς διαγραφής στοιχείων μαθήματος.....	61
Εικόνα 43.Πρόγραμμα Σπουδών.....	62
Εικόνα 44.Πρόγραμμα Σπουδών.....	63

Εικόνα 45. Πρόγραμμα Σπουδών.....	63
Εικόνα 46. Αλλαγή Κωδικού Πρόσβασης	64
Εικόνα 47. Έλεγχος Πεδίων Αλλαγής Κωδικού	65
Εικόνα 48. Έλεγχος Μήκους Νέου Κωδικού.....	65
Εικόνα 49. Επιτυχής Αλλαγή Password.....	66
Εικόνα 50. Φόρμα Δήλωσης Μαθημάτων	67
Εικόνα 51. Σφάλμα SQL διπλότυπης εισαγωγής δεδομένων	67
Εικόνα 52. Εμφάνιση εκτύπωσης αναλυτικής βαθμολογίας.....	68
Εικόνα 53. Σελίδα εμφάνισης στοιχείων φοιτητή	68
Εικόνα 54. Σελίδα ελέγχου ορθότητας ΑΜ φοιτητή.....	69

Κεφάλαιο 1

Εισαγωγή

Η πτυχιακή εργασία (ΠΕ), όσον αφορά τον εκπαιδευτικό της χαρακτήρα, είναι ένα από τα πιο σημαντικά μέρη του πτυχίου. Προσομοιάζει το πραγματικό εργασιακό περιβάλλον στο οποίο θα δραστηριοποιηθούν οι φοιτητές και περιλαμβάνει την μεγαλύτερη συμμετοχή του καθηγητή από οποιοδήποτε άλλο μάθημα της σχολής. Εναπόκειται στον σπουδαστή να εκμεταλλευτεί στο έπακρο την ευκαιρία αυτή και να αποκομίσει τα μέγιστα δυνατά οφέλη σε σχέση με την ακαδημαϊκή και επαγγελματική του κατάρτιση.

1.1 Περίληψη

Ο σκοπός της πτυχιακής αυτής ήταν η μελέτη, σχεδίαση και ανάπτυξη ενός συστήματος για την κάλυψη των αναγκών μιας γραμματείας ενός τμήματος του πανεπιστημίου, για την μετατροπή των χειρόγραφων εγγράφων ,σε ηλεκτρονικά έγγραφα . Η εφαρμογή θα είναι δυναμική, κάτι που θα μας δίνει την δυνατότητα εύκολου και γρήγορου χειρισμού.

Οι τεχνολογίες που χρησιμοποιήθηκαν είναι HTML(γλώσσα σήμανσης υπερκειμένου), PHP (γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο), MYSQL (Σύστημα Διαχείρισης Σχεσιακών Βάσεων Δεδομένων), JAVASCRIPT (γλώσσα σεναρίων) και το EASYPHP (υποστηρίζει την εκτέλεση δυναμικών ιστοσελίδων).

Έτσι το σύστημα που αναπτύχθηκε , διευκολύνει την τόσο απαιτητική δουλειά μιας γραμματείας ,που καθημερινά έχει να κάνει με χιλιάδες φοιτητές που ζητάνε τόσο να γραφτούν στην σχολή , ή να πάρουν κάποια δήλωση μαθημάτων ή αναλυτική βαθμολογία . Επίσης ο υπάλληλος της γραμματείας μπορεί εύκολα και γρήγορα να δημιουργήσει κάποιο μάθημα ή να το διαγράψει, να εισάγει στην βάση δεδομένων ,τα στοιχεία ενός καθηγητή ή να τα διαγράψει, να κρατάει ενήμερο το πρόγραμμα σπουδών και να αλλάζει τον κωδικό εισόδου του για να υπάρχει και η απαραίτητη ασφάλεια στα στοιχεία των εγγεγραμμένων.

1.2 Κίνητρο για την Διεξαγωγή της Εργασίας

Η παρούσα εργασία δημιουργήθηκε με γνώμονα την καθημερινή κοπιαστική εργασία που χρειάζεται από μέρους μιας γραμματείας κάποιου πανεπιστημίου, έτσι ώστε να φέρνει εις πέρας τις συνήθειες εργασίες της ημέρας, δηλαδή την αποθήκευση , διαχείριση και ασφάλεια διαφόρων πληροφοριών και δεδομένων.

1.3 Σκοπός και Στόχοι της Εργασίας

Ο σκοπός υλοποίησης αυτής της εφαρμογής είναι η κάλυψη των αναγκών μια γραμματείας ενός τμήματος του πανεπιστημίου, για εγγραφή ή διαγραφή ενός φοιτητή ή ενός καθηγητή ή ενός μαθήματος, με στόχο την αποτελεσματικότητα, την ταχύτητα και την ευκολία. Επίσης μπορούμε να αποθηκεύουμε πληροφορίες ή να εκτυπώνουμε το πρόγραμμα σπουδών και την δήλωση μαθημάτων ενός φοιτητή. Παρέχεται επίσης και η απαραίτητη ασφάλεια κωδικού.

Στόχος μας είναι να δημιουργήσουμε μια διαδικτυακή εφαρμογή η οποία θα αναπτυχθεί και περαιτέρω, ώστε να κατανοήσουμε ορισμένες λειτουργίες των δυναμικών ιστοσελίδων και εφαρμογών και πως χτίζεται κομμάτι κομμάτι μία τέτοια εφαρμογή. Επίσης να κατανοήσουμε και να μάθουμε την λειτουργία ορισμένων αντιπροσωπευτικών λογισμικών, τα οποία θα μας βοηθήσουν στην κατασκευή της εφαρμογής αυτής.

1.4 Δομή Εργασίας

Στο 1^ο κεφάλαιο προσπαθούμε να εισάγουμε τον αναγνώστη για τους σκοπούς και τους στόχους της υλοποίησης αυτής της εφαρμογής. Στο 2^ο κεφάλαιο αναφερόμαστε αναλυτικότερα στις εφαρμογές που χρησιμοποιήσαμε για την μελέτη, σχεδίαση και ανάπτυξη ενός συστήματος. Στο 3^ο κεφάλαιο αναφερόμαστε στην βιβλιογραφική αναζήτηση της τεχνολογίας και στην υλοποίηση του σχεδίου δράσης. Στο 4^ο κεφάλαιο περνάμε στο κύριο μέρος της εργασίας μας όπου αναλύουμε το πρόβλημα σχεδιάζουμε και υλοποιούμε την λύση του. Στο 5^ο κεφάλαιο ασχολούμαστε με τα αποτελέσματα της εφαρμογής και ποια τα οφέλη που μπορούμε να αποκομίσουμε από αυτήν, τα συμπεράσματά μας και την μελλοντική ανάπτυξη της.

Κεφάλαιο 2

Μεθοδολογία Υλοποίησης

2.1 Στατικές και Δυναμικές Εφαρμογές

2.1.1 Στατικές Εφαρμογές

Οι στατικές εφαρμογές είναι ουσιαστικά απλά ηλεκτρονικά "έγγραφα". Μπορούν να περιέχουν κείμενα, φωτογραφίες, συνδέσμους (links), αρχεία για download, κινούμενα σχέδια (animated graphics). Είναι κατάλληλες κυρίως για την δημιουργία "μόνιμων/στατικών παρουσιάσεων", όπου δεν υπάρχει συχνά η ανάγκη να τροποποιείται το περιεχόμενό τους. Δηλαδή, θα μπορούσε να χρησιμοποιηθεί μια στατική εφαρμογή για την γενική παρουσίαση μιας επιχείρησης και των προϊόντων ή υπηρεσιών της.

Ο χαρακτηρισμός "στατική" ,δεν σημαίνει ότι τίποτα στην σελίδα δεν κουνιέται, αντιθέτως μπορεί να έχει κινούμενα γραφικά και άλλα μιχλιμπίδια. Σημαίνει ότι το περιεχόμενο της κάθε σελίδας είναι σταθερό και συγκεκριμένο. Πρόκειται δηλαδή για σταθερά ηλεκτρονικά έγγραφα. Το κύριο μειονέκτημά τους είναι ότι δεν είναι πρακτική/βολική η χρήση τους όταν χρειάζεται το περιεχόμενο της ιστοσελίδας να αλλάζει συχνά, π.χ. τουλάχιστον μία φορά την εβδομάδα να προστίθεται κάποιο άρθρο, ή να αλλάζουν τα προϊόντα κτλ. Αυτό συμβαίνει γιατί για να τροποποιηθεί μια στατική ιστοσελίδα, πρέπει να γίνουν επεμβάσεις απευθείας πάνω στην ιστοσελίδα, με κάποιο σχετικό πρόγραμμα (html editor είτε απλό είτε WYSIWYG), μια διαδικασία δηλαδή όπως το να σβήνεις και να γράφεις σε ένα τετράδιο, που όμως χρειάζεται μια σχετική εμπειρία.

2.1.2 Δυναμικές Εφαρμογές

Δυναμικές εφαρμογές με βάσεις δεδομένων. Οι δυναμικές εφαρμογές, μπορεί στην εμφάνιση, σε πολλές περιπτώσεις, να μην έχουν μεγάλη διαφορά με τις στατικές, όμως οι δυνατότητές είναι πολύ περισσότερες, από πολλές πλευρές, καθώς στην περίπτωση αυτή ουσιαστικά πρόκειται για μία εφαρμογή (πρόγραμμα), και όχι ένα απλό ηλεκτρονικό έγγραφο.

Συνήθως, οι δυναμικές εφαρμογές, χρησιμοποιούν κάποια βάση δεδομένων (database), όπου αποθηκεύουν πληροφορίες και απ' όπου αντλούν το περιεχόμενό τους, ανάλογα με το τι ζητάει ο χρήστης/επισκέπτης σε κάθε του "κλικ". Η χρήση των βάσεων δεδομένων, είναι αυτή που επιτρέπει την εύκολη προσθαφαίρεση περιεχομένου στις δυναμικές ιστοσελίδες, καθώς δεν απαιτείται να επεξεργάζεται κανείς κάθε φορά την ίδια την ιστοσελίδα, αλλά απλά να διαχειρίζεται έμμεσα το περιεχόμενο στην βάση δεδομένων και οι υπόλοιπες διαδικασίες γίνονται αυτοματοποιημένα από τον "μηχανισμό" της ιστοσελίδας.

Για την εύκολη **διαχείριση του περιεχομένου** σε μία δυναμική εφαρμογή, υπάρχει πρόσθετα στην ιστοσελίδα ένας εύχρηστος μηχανισμός "**CMS**" (Content Managment System), μέσω του οποίου η προσθαφαίρεση περιεχομένου μπορεί να γίνει ακόμη και από κάποιον αρχάριο. Φυσικά, η "περιοχή διαχείρισης" της ιστοσελίδας, προστατεύεται με κωδικό πρόσβασης (password), και δεν μπορούν να εισέλθουν σε αυτή οι επισκέπτες της ιστοσελίδας.

2.2 Web Σελίδες και HTML

Μία Web Σελίδα είναι ένα text αρχείο H/Y κατασκευασμένο με τέτοιο τρόπο που να μπορεί να παρουσιάζεται ολοκληρωμένο και στην επιθυμητή για τον δημιουργό του μορφή, με μία μόνο κλήση από τον χειριστή του κατάλληλου λογισμικού (δηλ. ενός web client). Κάθε Web σελίδα έχει την δική της διεύθυνση μέσα στο Internet. Αυτή η διεύθυνση είναι η διεύθυνση του κόμβου στον οποίο είναι αποθηκευμένα τα directories πέρα από το αρχικό directory που έχει ορίσει ο Web Server και το όνομά της. Για παράδειγμα, κάποια σελίδα βρίσκεται στο URL <http://www.eexi.gr/interbiz/index.html> Αν όμως δούμε τα directories του server θα την βρούμε στο `usr/local/sbin/httpd/html/interbiz/index.html` Το όνομα του αρχείου δεν είναι απαραίτητο να συμπεριληφθεί στο URL αν είναι `index.htm` ή `index.html` (κάθε αρχείο που περιέχει μια Web σελίδα πρέπει να έχει όνομα με επέκταση `htm` ή `html`) διότι αν δεν βάλουμε το όνομα του αρχείου, τότε ο browser θα ψάξει από μόνος του (by default) για ένα αρχείο με το όνομα `index` στο URL και path που του έχουμε δώσει.

Web Server ονομάζουμε το λογισμικό που τρέχει σε ένα κόμβο Internet και επιτρέπει σε άλλους υπολογιστές να αποκτούν αντίγραφα των Web σελίδων που είναι αποθηκευμένες σε αυτόν. Web Client ονομάζουμε το λογισμικό που τρέχει ένας H/Y και του επιτρέπει να "διαβάζει" Web σελίδες (από το σκληρό του δίσκο ή από το Internet αν είναι συνδεδεμένος με αυτό).

Τα αρχικά HTML προέρχονται από τις λέξεις HyperText Markup Language. Η `html` δεν είναι μια γλώσσα προγραμματισμού. Είναι μια γλώσσα σήμανσης (markup language), δηλαδή ένας ειδικός τρόπος γραφής κειμένου. Ο καθένας μπορεί να δημιουργήσει ένα αρχείο HTML χρησιμοποιώντας απλώς έναν επεξεργαστή κειμένου. Αποτελεί υποσύνολο της γλώσσας SGML (Standard Generalized Markup Language) που επινοήθηκε από την IBM προκειμένου να λυθεί το πρόβλημα της μη τυποποιημένης εμφάνισης κειμένων στα διάφορα υπολογιστικά συστήματα. Ο browser αναγνωρίζει αυτόν τον τρόπο γραφής και εκτελεί τις εντολές που περιέχονται σε αυτόν. Αξίζει να σημειωθεί ότι η `html` είναι η πρώτη και πιο διαδεδομένη γλώσσα περιγραφής της δομής μιας ιστοσελίδας. Η `html` χρησιμοποιεί τις ειδικές ετικέτες (τα tags) να δώσει τις απαραίτητες οδηγίες στον browser. Τα tags είναι εντολές που συνήθως ορίζουν την αρχή ή το τέλος μιας λειτουργίας. Τα tags βρίσκονται πάντα μεταξύ των συμβόλων `<` και `>`. Π.χ. `<BODY>` Οι οδηγίες είναι case insensitive, δεν επηρεάζονται από το αν έχουν γραφτεί με πεζά (μικρά) ή κεφαλαία. Ένα αρχείο HTML πρέπει να έχει κατάληξη `htm` ή `html`.

Για να μπορούν οι browser να ερμηνεύουν σχεδόν απόλυτα σωστά την `html` έχουν θεσπιστεί κάποιοι κανόνες. Αυτοί οι κανόνες είναι γνωστοί ως προδιαγραφές. Επομένως σχεδόν κάθε είδος υπολογιστή μπορεί να δείξει το ίδιο καλά μια ιστοσελίδα. Οι πρώτες προδιαγραφές ήταν η `html 2.0`. Πρόβλημα προέκυψε όταν η Microsoft και η Netscape πρόσθεσαν στην `html` τέτοιες δυνατότητες που στην αρχή τουλάχιστον ήταν συμβατές μόνο με συγκεκριμένους browser. Ακόμη και σήμερα υπάρχουν διαφορές στην απεικόνιση κάποιας σελίδας από διαφορετικούς browsers. Ιδιαίτερο είναι το πρόβλημα όταν η ιστοσελίδα, εκτός από "καθαρή" HTML περιλαμβάνει και εφαρμογές Javascript.

Σήμερα πολλοί είναι εκείνοι που δημιουργούν μια ιστοσελίδα σε κάποιο πρόγραμμα που επιτρέπει την δημιουργία χωρίς την συγγραφή κώδικα. Η κοινή άποψη πάνω στο θέμα όμως είναι ότι κάτι τέτοιο είναι αρνητικό επειδή ο δημιουργός δεν έχει τον απόλυτο έλεγχο του κώδικα με αποτέλεσμα πολλές φορές να υπάρχει οπτικό χάος στην προσπάθεια των browser να εμφανίσουν την ιστοσελίδα. Για το σκοπό αυτό έχει δημιουργηθεί ειδικό λογισμικό, που επιτρέπει το "στήσιμο" της σελίδας οπτικά, χωρίς τη συγγραφή κώδικα, δίνει όμως τη δυνατότητα παρέμβασης και στον κώδικα. Χαρακτηριστικό παράδειγμα το λογισμικό Dreamweaver της Adobe και το FrontPage της Microsoft.

Ονομάζουμε Δομή μιας Web Παρουσίασης ή ενός Web site τον τρόπο με τον οποίο είναι συνδεδεμένες (παραπέμπουν η μία στην άλλη) οι Web σελίδες που το απαρτίζουν. Η Δομή αυτή μπορεί να είναι Ιεραρχική (δηλ. όπως ένα δενδροειδές διάγραμμα με την αρχή του στην κορυφή), Γραμμική (όπως αυτή των σελίδων ενός βιβλίου όπου έχουμε μια ακολουθία με αρχή την πρώτη σελίδα και τέλος την τελευταία), Μικτή (που αποτελεί ένα συνδυασμό των δύο προηγούμενων) και Ακανόνιστη (χωρίς κεντρική δομή και εμφανή αρχή και τέλος).

Μία Web Σελίδα είναι ένα text αρχείο. Για τον λόγο αυτό μπορούμε να χρησιμοποιήσουμε οποιοδήποτε πρόγραμμα μας δίνει την δυνατότητα να παράγουμε text αρχεία. Αρχικά σας προτείνουμε να χρησιμοποιήσετε το notepad (σημειωματάριο) των Windows. Αν θέλετε να χρησιμοποιήσετε το Word for Windows μην ξεχάσετε πως το αρχείο πρέπει να σωθεί σε μορφή txt και όχι doc. Αρκετοί αναγνώστες θα έχουν ακούσει για προγράμματα όπως το Frontpage ή το Compose με τα οποία μπορεί κανείς να κατασκευάσει Web σελίδες. Πρόκειται για πολύ χρήσιμα εργαλεία αλλά με αυτά δεν μαθαίνει κανείς HTML. Σας προτείνουμε εντονότατα να μην τα χρησιμοποιήσετε ακόμη. Μόνο όταν μάθετε τι είναι και πως λειτουργεί η HTML θα μπορέσετε να τα χρησιμοποιήσετε χωρίς προβλήματα.

Βήμα 1

Η HTML κώδικας της πρώτης μας σελίδας θα έχει την παρακάτω μορφή:

<HTML>

Αυτή

είναι

η

πρώτη

μου

σελίδα.

</HTML>

Το μόνο που κάναμε ήταν να γράψουμε ένα κείμενο που να περικλείεται από την οδηγία <HTML>...</HTML> Κάθε Web σελίδα αρχίζει με την οδηγία (tag) <HTML> και τελειώνει με την εντολή </HTML>. Η πρώτη πληροφορεί τον Web client πως το αρχείο είναι γραμμένο σε γλώσσα HTML και η τελευταία πως ο HTML κώδικας τελειώσε.

Βήμα 2

Αποθηκεύουμε το αρχείο με όποιο όνομα θέλουμε αρκεί η επέκτασή του να είναι htm ή html. Π.χ. μπορούμε να το ονομάσουμε mypage.htm ή mypage.html (προσέξτε μήπως το πρόγραμμα με το οποίο γράψατε την σελίδα προσθέσει μετά από μόνο του μια δική του επέκταση μετά το htm ή html).

Βήμα 3

Ανοίγουμε το αρχείο (δηλαδή την πρώτη μας σελίδα) με τον browser που χρησιμοποιούμε. Για να το κάνουμε αυτό στο Netscape Communicator κάνουμε File - Open Page - Choose File και File - Open - Browse (Αρχείο - 'νοιγμα - Αναζήτηση) στον Internet Explorer. Αυτό που βλέπουμε είναι μια σελίδα που γράφει:

Αυτή είναι η πρώτη μου σελίδα.

2.2 Εργαλεία Δημιουργίας Δυναμικών Ιστοσελίδων

2.2.1 Joomla

Υπάρχουν διάφορα προγράμματα και εφαρμογές με τα οποία μπορούμε να κατασκευάσουμε μια ιστοσελίδα. Μερικά από αυτά είναι δωρεάν για ευρεία χρήση τα οποία χρησιμοποιούνται για ερασιτεχνική χρήση κυρίως και είναι πιο φιλικά στο χρήστη, δηλαδή δεν απαιτούν ιδιαίτερες προγραμματιστικές γνώσεις. Ένα από αυτά είναι το Joomla.

Γεμάτο δυνατότητες αλλά και ταυτόχρονα εξαιρετικά ευέλικτο και φιλικό, το Joomla! είναι η εφαρμογή που χρειαζόμαστε για να δημοσιεύσουμε στο διαδίκτυο μια προσωπική ιστοσελίδα, αλλά και έναν ολόκληρο εταιρικό δικτυακό τόπο. Είναι προσαρμόσιμο σε περιβάλλοντα επιχειρηματικής κλίμακας όπως τα intranets μεγάλων επιχειρήσεων ή οργανισμών. Οι δυνατότητες επέκτασής του είναι πρακτικά απεριόριστες.

Το Joomla εγκαθίσταται σε έναν κεντρικό υπολογιστή, τον web server. Ο χρήστης, εμείς δηλαδή, έχουμε πρόσβαση στο περιβάλλον διαχείρισης μέσω ενός browser, όπως είναι ο Internet Explorer ή ο Firefox.

Οι δυνατότητες χρήσης του Joomla περιορίζονται μόνον από τη δημιουργικότητά μας. Κάποιοι χρησιμοποιούν το Joomla για διασκέδαση, για τη κατασκευή μιας προσωπικής ή οικογενειακής ιστοσελίδας. Ενσωματώνοντας διάφορα, διαθέσιμα δωρεάν, πρόσθετα εργαλεία / επεκτάσεις, διευρύνονται οι δυνατότητες και λειτουργίες του Joomla, κάνοντάς το μια εφαρμογή αξιόπιστη για την ανάπτυξη σοβαρών εταιρικών δικτυακών τόπων. Το Joomla χρησιμοποιεί μια ισχυρή templating μηχανή που μας δίνει τη δυνατότητα να χρησιμοποιήσουμε το δικό μας, εξατομικευμένο template.

Για το σκοπό αυτό μπορούμε είτε να δημιουργήσουμε δικό μας template, είτε να κατεβάσουμε από το διαδίκτυο ένα από τα εκατοντάδες (περίπου 500) που διατίθενται δωρεάν, είτε να αγοράσουμε κάποιο από αυτά που πωλούνται, είτε να παραγγείλουμε την κατασκευή ενός σε έναν επαγγελματία. Από τη στιγμή που θα δημιουργηθεί το δικό μας template, δεν χρειάζονται παρά τρία κλικ, ώστε το περιεχόμενο του δικτυακού μας τόπου να προσαρμοστεί στη μορφή του.

Μπορούμε να προσθέσουμε περιεχόμενο στον ιστότοπο μας από οποιονδήποτε υπολογιστή διαθέτει σύνδεση στο διαδίκτυο. Πληκτρολογούμε το κείμενο, ανεβάζουμε τις φωτογραφίες και δημοσιεύουμε.

Χρησιμοποιούμε το Joomla για να συνεργαστούμε με όσους μπορούν να συμβάλλουν στην επεξεργασία του περιεχομένου. Δίνουμε τη δυνατότητα σε συνεργάτες μας να δημοσιεύσουν περιεχόμενο, στο πλαίσιο ασφαλώς των αρμοδιοτήτων τους. Είναι εύκολο. Μπορούν να το κάνουν από κάθε υπολογιστή με σύνδεση στο διαδίκτυο.

Δημοσιεύουμε απεριόριστες σελίδες, χωρίς να περιοριζόμαστε από το Joomla. Χωρίς κανένα πρόβλημα, μπορούμε να κάνουμε αναζητήσεις περιεχομένου και να τις αρχειοθετήσουμε. Προσθέτουμε forum, photo galleries, βιβλιοθήκες αρχείων, βιβλία επισκεπτών και φόρμες επικοινωνίας; λίγες από τις εκατοντάδες πρόσθετες εφαρμογές που κυκλοφορούν για το Joomla.



Εικόνα 1 .Joomla Template

2.2.2 Visual Basic

Η Visual Basic (VB) είναι γλώσσα προγραμματισμού τρίτης γενιάς, οδηγούμενη από συμβάντα (event driven) και έχει ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) από τη Microsoft για το μοντέλο προγραμματισμού COM. Η VB θεωρείται επίσης μία σχετικά εύκολη γλώσσα προγραμματισμού στην εκμάθηση και τη χρησιμοποίηση, λόγω των χαρακτηριστικών της, καθώς έχει Γραφικό Περιβάλλον Χρήστη και συγγένεια με την γλώσσα προγραμματισμού BASIC. Η Visual Basic προέρχεται από τη BASIC και επιτρέπει την ταχεία ανάπτυξη εφαρμογών (RAD) με Γραφικό Περιβάλλον Χρήστη (GUI), πρόσβαση σε βάσεις δεδομένων χρησιμοποιώντας αντικείμενα (Data Access Objects, Remote Data Objects, ή ActiveX Data Objects), και τη δημιουργία στοιχείων ελέγχου ActiveX και αντικειμένων. Οι γλώσσες προγραμματισμού τύπου "scripting", όπως η VBA και VBScript συντακτικά είναι παρόμοιες με τη Visual Basic, αλλά έχουν διαφορετικές επιδόσεις. Ένας προγραμματιστής μπορεί να ολοκληρώσει μια εφαρμογή χρησιμοποιώντας τα στοιχεία που παρέχονται με την Visual Basic. Προγράμματα γραμμένα σε Visual Basic μπορούν, επίσης, να χρησιμοποιήσουν το Windows API, αλλά κάτι τέτοιο απαιτεί δηλώσεις εξωτερικών συναρτήσεων. Η τελική έκδοση 6 βγήκε το 1998. Η εκτεταμένη υποστήριξη της Microsoft έληξε το Μάρτιο του 2008 και ορίστηκε διάδοχος της η Visual Basic.NET (γνωστή απλά ως Visual Basic)

Όπως και η γλώσσα προγραμματισμού BASIC, η Visual Basic έχει σχεδιαστεί για να είναι εύκολη στην εκμάθηση και το χειρισμό. Η γλώσσα δεν επιτρέπει στους προγραμματιστές να δημιουργήσουν μόνο απλές εφαρμογές GUI, αλλά μπορούν, επίσης, να αναπτύξουν πολύπλοκες εφαρμογές. Ο προγραμματισμός σε VB συνίσταται από τον οπτικό συνδυασμό στοιχείων ή ελέγχων σε μια φόρμα, τον προσδιορισμό χαρακτηριστικών και ενεργειών αυτών των στοιχείων και την σύνταξη επιπλέον γραμμών κώδικα για αυξημένη λειτουργικότητα. Καθώς υπάρχουν προεπιλεγμένα χαρακτηριστικά και ενέργειες για τα επιμέρους στοιχεία, μπορεί να δημιουργηθεί ένα απλό πρόγραμμα χωρίς ο προγραμματιστής να γράψει πολλές γραμμές κώδικα. Στις προηγούμενες εκδόσεις υπήρχαν προβλήματα επιδόσεων, αλλά με τους ταχύτερους υπολογιστές και τη μεταγλώττιση εγγενούς κώδικα αυτό παύει να είναι ένα τόσο σημαντικό ζήτημα.

Αν και τα προγράμματα μπορούν να μετατραπούν σε εγγενή εκτελέσιμο κώδικα από την έκδοση 5 και μετά, αυτά εξακολουθούν να απαιτούν την παρουσία των βιβλιοθηκών χρόνου εκτέλεσης (runtime) με μέγεθος περίπου 1 MB. Οι βιβλιοθήκες runtime υπάρχουν στα Windows 2000 και αργότερα, αλλά στις παλαιότερες εκδόσεις των Windows όπως τα 95/98/NT πρέπει να διανέμονται μαζί με το εκτελέσιμο αρχείο.

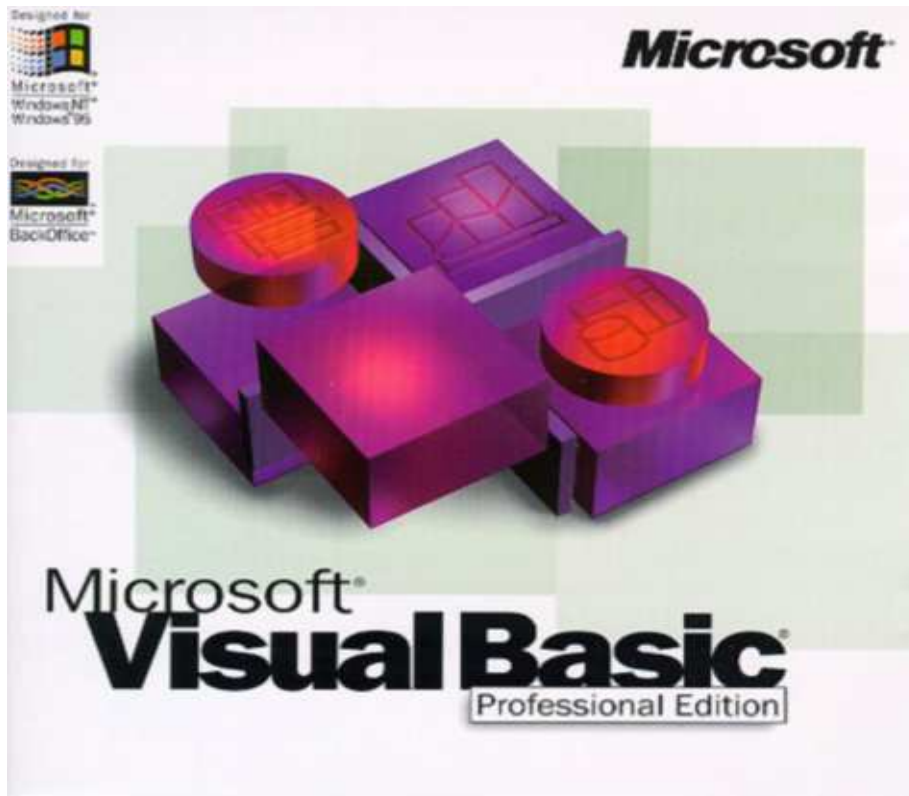
Οι φόρμες δημιουργούνται χρησιμοποιώντας τεχνικές "σύρε κι άσε" (drag-and-drop). Χρησιμοποιείται ένα εργαλείο για την τοποθέτηση στοιχείων ελέγχου (π.χ. πλαίσια κειμένου, κουμπιά, κλπ.) στη φόρμα (παράθυρο). Τα στοιχεία ελέγχου έχουν χαρακτηριστικά και χειριστές συμβάντων συνδεδεμένους με αυτά. Οι προεπιλεγμένες τιμές παρέχονται όταν δημιουργείται το στοιχείο ελέγχου, αλλά μπορούν να τροποποιηθούν από τον προγραμματιστή. Πολλές τιμές χαρακτηριστικών είναι δυνατό να τροποποιηθούν κατά το χρόνο εκτέλεσης από ενέργειες του χρήστη ή αλλαγές του περιβάλλοντος, παρέχοντας έτσι μια δυναμική εφαρμογή. Για παράδειγμα, μπορεί να εισαχθεί κώδικας στον χειριστή συμβάντων αλλαγής διαστάσεων της φόρμας, ώστε ένα στοιχείο

ελέγχου να παραμένει πάντα στο κέντρο της φόρμας ή να μεγαλώσει ώστε να την γεμίσει, κλπ. Με την προσθήκη κώδικα μέσα σε ένα χειριστή συμβάντων για το πάτημα των πλήκτρων σε ένα πλαίσιο κειμένου, το πρόγραμμα μπορεί αυτόματα να μετατρέψει το εισαγόμενο κείμενο σε κεφαλαία ή πεζά ή ακόμα και να εμποδίσει ορισμένους από τους χαρακτήρες να εμφανιστούν.

Με τη Visual Basic είναι δυνατή η δημιουργία εκτελέσιμων (EXE) αρχείων, στοιχείων ελέγχου ActiveX ή αρχείων DLL, αλλά χρησιμοποιείται κυρίως για την ανάπτυξη εφαρμογών για τα Windows και τη διασύνδεση συστημάτων βάσεων δεδομένων. Πλαίσια διαλόγου με λιγότερες λειτουργίες μπορούν να χρησιμοποιηθούν για pop-up δυνατότητες. Τα στοιχεία ελέγχου παρέχουν τις βασικές λειτουργίες της εφαρμογής, ενώ οι προγραμματιστές μπορούν να εισαγάγουν επιπλέον λογική μέσα στο κατάλληλο χειριστή γεγονότων. Για παράδειγμα, ένα πτυσσόμενο πλαίσιο θα εμφανίζει αυτόματα μια λίστα που θα επιτρέπει στο χρήστη να επιλέξει οποιοδήποτε στοιχείο. Ένας χειριστής γεγονότων καλείται όταν ένα αντικείμενο είναι επιλεγμένο

και στη συνέχεια μπορεί να εκτελεστεί πρόσθετος κώδικας που δημιουργείται από τον προγραμματιστή για να εκτελεστεί κάποια ενέργεια που βασίζεται στο στοιχείο που έχει επιλεγθεί.

Εναλλακτικά, ένα συστατικό της Visual Basic μπορεί να μην έχει Γραφικό Περιβάλλον Χρήστη, αλλά, αντ' αυτού, να παρέχει αντικείμενα ActiveX σε άλλα προγράμματα μέσω Component Object Model (COM). Αυτό επιτρέπει επεξεργασία στην πλευρά του διακομιστή (server-side processing) ή τη δημιουργία πρόσθετων μορφωμάτων (add-in module). Η γλώσσα έχει αυτόματη διαχείριση μνήμης με την τεχνική της συλλογής σκουπιδιών (garbage collection) χρησιμοποιώντας υπολογισμό αναφορών και έχει μια μεγάλη βιβλιοθήκη με βοηθητικά αντικείμενα καθώς και βασική αντικειμενοστραφή υποστήριξη. Από τα πιο κοινά στοιχεία που περιλαμβάνονται στο προεπιλεγμένο πρότυπο έργου, ο προγραμματιστής σπάνια χρειάζεται να καθορίσει πρόσθετες βιβλιοθήκες. Αντίθετα με πολλές άλλες γλώσσες προγραμματισμού η Visual Basic γενικά δεν διαχωρίζει τους πεζούς από τους κεφαλαίους χαρακτήρες, αν και θα μετατρέψει τις λέξεις-κλειδιά σε μία τυπική διαμόρφωση. Οι συγκρίσεις συμβολοσειρών διαχωρίζουν τα πεζά από τα κεφαλαία από προεπιλογή, αλλά μπορεί να αλλάξει αυτό, εφόσον το επιθυμείτε. Ο μεταγλωττιστής της Visual Basic είναι κοινός με τις άλλες γλώσσες του Visual Studio (C, C++), αλλά οι περιορισμοί στον IDE δεν επιτρέπουν τη δημιουργία ορισμένων στόχων (μοντέλα Windows DLL) και σε μοντέλα νημάτων.



Εικόνα 2 Το λογισμικό Visual Basic

2.2.3 Drupal

Το Drupal είναι ένα αρθρωτό σύστημα διαχείρισης περιεχομένου (Content Management System, CMS) ανοικτού/ελεύθερου λογισμικού, γραμμένο στη γλώσσα προγραμματισμού PHP. Το Drupal, όπως πολλά σύγχρονα CMS, επιτρέπει στο διαχειριστή συστήματος να οργανώνει το περιεχόμενο, να προσαρμόζει την παρουσίαση, να αυτοματοποιεί διαχειριστικές εργασίες και να διαχειρίζεται τους επισκέπτες του ιστοτόπου και αυτούς που συνεισφέρουν.

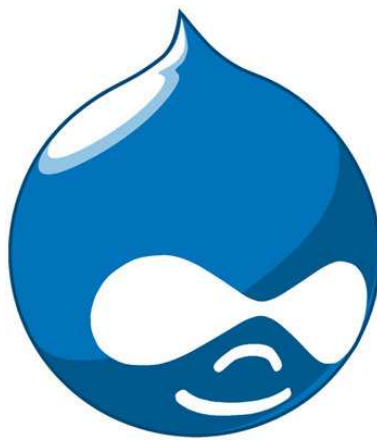
Παρόλο που υπάρχει μια πολύπλοκη προγραμματιστική διεπαφή, οι περισσότερες εργασίες μπορούν να γίνουν με λίγο ή και καθόλου προγραμματισμό. Το Drupal ορισμένες φορές περιγράφεται ως "υποδομή για εφαρμογές ιστού", καθώς οι δυνατότητές του προχωρούν παραπέρα από τη διαχείριση περιεχομένου, επιτρέποντας ένα μεγάλο εύρος υπηρεσιών και συναλλαγών.

Το Drupal μπορεί να εκτελεστεί σε διάφορες πλατφόρμες, συμπεριλαμβανομένων των λειτουργικών συστημάτων Windows, Mac OS X, Linux, FreeBSD, ή οποιασδήποτε πλατφόρμας που υποστηρίζει είτε το διακομιστή ιστοσελίδων Apache HTTP Server (έκδοση 1.3+), είτε το Internet Information Services (έκδοση IIS5+), καθώς επίσης και τη γλώσσα προγραμματισμού PHP (έκδοση 4.3.3+). Το Drupal απαιτεί μια βάση δεδομένων όπως η MySQL και η PostgreSQL για την αποθήκευση του περιεχομένου και των ρυθμίσεών του.

Αρχικά γραμμένο από τον Dries Buytaert ως σύστημα πίνακα ανακοινώσεων (BBS, bulletin board system), το Drupal μετατράπηκε σε εγχείρημα ανοικτού κώδικα το 2001. Drupal είναι η διατύπωση στην Αγγλική γλώσσα της Ολλανδικής λέξης "druppel", που σημαίνει "σταγόνα". Το όνομα πάρθηκε από τον ξεπερασμένο πλέον ιστοτόπο Drop.org, του οποίου ο κώδικας εξελίχθηκε στο Drupal. Ο Buytaert ήθελε να ονομάσει τον ιστοτόπο "dorp" (στα Ολλανδικά σημαίνει "χωριό", αναφερόμενος στη διάσταση της κοινότητας), αλλά έκανε ένα ορθογραφικό λάθος κατά τη διαδικασία ελέγχου του ονόματος χώρου (domain name) και τελικά σκέφτηκε ότι ακούγεται καλύτερα. Από το Μάιο του 2006 ως τον Απρίλιο του 2007, χρήστες κατέβασαν το Drupal από τον επίσημο ιστοτόπο περισσότερες από 600.000 φορές. Μια μεγάλη κοινότητα χρηστών λαμβάνει πλέον μέρος στη συνεχή εξέλιξη του Drupal.

Το Drupal έχει λάβει επαίνους από τους διαχειριστές ιστοσελίδων, σχεδιαστές και προγραμματιστές για τον αρθρωτό σχεδιασμό του, που παρέχει το βασικό του στρώμα, ή "πυρήνα", να παρέχει τα βασικά χαρακτηριστικά του Drupal στην προεπιλεγμένη εγκατάστασή του. Πρόσθετα χαρακτηριστικά λειτουργικότητας και παρουσίασης μπορούν να επεκταθούν στον πυρήνα με την πρόσθεση προσαρτώμενων μονάδων και θεματικών παραλλαγών.

Οι μονάδες του Drupal χρησιμοποιούνται για να "υπερβούν" τα ενσωματωμένα χαρακτηριστικά του πυρήνα, επεκτείνοντας έτσι ή και αντικαθιστώντας την εξ' ορισμού συμπεριφορά του Drupal, χωρίς την επέμβαση στον αυτούσιο κώδικα των αρχείων του πυρήνα του Drupal. Η δυνατότητα αυτή της τροποποίησης της λειτουργικότητας του πυρήνα έχει επίπτωση στην προσαρμοστικότητα του Drupal καθώς και στην ασφάλειά του, ειδικότερα σε θέματα ασφαλείας, όπως η έγχυση εντολών SQL (SQL injection). Προσαρμοσμένες θεματικές παραλλαγές, που μπορούν να προστεθούν χωρίς να επηρεάζουν τον πυρήνα του Drupal, χρησιμοποιούν προτυποποιημένες μορφές που μπορούν να δημιουργηθούν από μηχανές σχεδίασης θεματικών παραλλαγών τρίτων.



Εικόνα 3 Το λογότυπο Drupal

2.2.4 XML

Η XML (αγγλ. αρκτ. από το Extensible Markup Language) είναι μία γλώσσα σήμανσης, που περιέχει ένα σύνολο κανόνων για την ηλεκτρονική κωδικοποίηση κειμένων. Ορίζεται, κυρίως, στην προδιαγραφή XML 1.0 (XML 1.0 Specification), που δημιούργησε ο διεθνής οργανισμός προτύπων W3C (World Wide Web Consortium), αλλά και σε διάφορες άλλες σχετικές προδιαγραφές ανοιχτών προτύπων.

Η XML σχεδιάστηκε δίνοντας έμφαση στην απλότητα, τη γενικότητα και τη χρησιμότητα στο Διαδίκτυο. Είναι μία μορφοποίηση δεδομένων κειμένου, με ισχυρή υποστήριξη Unicode για όλες τις γλώσσες του κόσμου. Αν και η σχεδίαση της XML εστιάζει στα κείμενα, χρησιμοποιείται ευρέως για την αναπαράσταση αυθαίρετων δομών δεδομένων, που προκύπτουν για παράδειγμα στις υπηρεσίες ιστού.

Υπάρχει μία ποικιλία διεπαφών προγραμματισμού εφαρμογών, που μπορούν να χρησιμοποιούν οι προγραμματιστές, για να προσπελαίνουν δεδομένα XML, αλλά και διάφορα συστήματα σχημάτων XML, τα οποία είναι σχεδιασμένα για να βοηθούν στον ορισμό γλωσσών, που προκύπτουν από την XML. Έως το 2009, έχουν αναπτυχθεί εκατοντάδες γλώσσες που βασίζονται στην XML, συμπεριλαμβανομένων του RSS, του SOAP και της XHTML. Προεπιλεγμένες κωδικοποιήσεις βασισμένες στην XML, υπάρχουν για τις περισσότερες σουίτες εφαρμογών γραφείου, συμπεριλαμβανομένων του Microsoft Office (Office Open XML), του OpenOffice.org (OpenDocument) και του iWork της εταιρίας Apple.

Το περιεχόμενο αυτής της ενότητας, βασίζεται στην προδιαγραφή XML 1.0. Δεν αποτελεί μία πλήρη λίστα όλων των όρων που υπάρχουν στη γλώσσα XML. Είναι μία εισαγωγή στα βασικά στοιχεία, που συναντώνται στην καθημερινή της χρήση. Χαρακτήρας Unicode Εξ ορισμού, ένα κείμενο XML είναι μία ακολουθία χαρακτήρων. Σχεδόν κάθε χαρακτήρας Unicode μπορεί να εμφανίζεται σε ένα κείμενο XML.

Επεξεργαστής και Εφαρμογή

Είναι το λογισμικό που επεξεργάζεται ένα κείμενο XML. Είναι αναμενόμενο, ότι ένας επεξεργαστής δουλεύει για μία εφαρμογή. Υπάρχουν μερικές πολύ συγκεκριμένες απαιτήσεις, σχετικά με το τι μπορεί και τι δεν μπορεί να κάνει ένας επεξεργαστής XML, αλλά καμία, όσον αφορά στη συμπεριφορά της εφαρμογής. Ο επεξεργαστής (όπως ονοματίζεται από την προδιαγραφή), αναφέρεται συχνά, με τον αγγλικό όρο XML parser.

Σήμανση και Περιεχόμενο

Οι χαρακτήρες που απαρτίζουν ένα κείμενο XML, αποτελούν είτε τη σήμανση είτε το περιεχόμενό του. Η σήμανση και το περιεχόμενο, μπορούν να επισημανθούν και να διακριθούν, ύστερα από την εφαρμογή κάποιων απλών συντακτικών κανόνων. Όλα τα αλφαριθμητικά που συνιστούν τη σήμανση, είτε ξεκινούν με το χαρακτήρα "<" και καταλήγουν στο χαρακτήρα ">", είτε ξεκινούν με το χαρακτήρα "&" και καταλήγουν στο χαρακτήρα ";". Ακολουθίες χαρακτήρων που δε συνιστούν τη σήμανση, αποτελούν το περιεχόμενο ενός κειμένου XML.

Ετικέτα

Ένα στοιχείο σήμανσης που ξεκινά με το χαρακτήρα "<" και καταλήγει στο χαρακτήρα ">". Υπάρχουν τρία είδη ετικέτας: ετικέτες-αρχής, για παράδειγμα <section>, ετικέτες-τέλους, για παράδειγμα </section>, και ετικέτες-χωρίς-περιεχόμενο, για παράδειγμα <line-break/>.

Στοιχείο

Ένα λογικό απόσπασμα ενός κειμένου, που είτε ξεκινά με μία ετικέτα-αρχής και καταλήγει σε μία ετικέτα-τέλους, είτε αποτελείται μόνο από μία ετικέτα-χωρίς-περιεχόμενο. Οι χαρακτήρες που υπάρχουν, αν υπάρχουν, μεταξύ μιας ετικέτας-αρχής και μιας ετικέτας-τέλους, συνιστούν το περιεχόμενο του στοιχείου, το οποίο μπορεί να περιέχει σήμανση, συμπεριλαμβανομένων και άλλων στοιχείων, που ονομάζονται στοιχεία-παιδιά. Ένα παράδειγμα ενός στοιχείου είναι το <Greeting>Hello, world.</Greeting>. Ένα άλλο είναι το <line-break/>.

Χαρακτηριστικό

Ένα στοιχείο σήμανσης που αποτελείται από ένα ζευγάρι όνομα/τιμή, το οποίο υπάρχει μέσα σε μία ετικέτα-αρχής ή σε μία ετικέτα-χωρίς-περιεχόμενο. Στο παράδειγμα παρακάτω, το στοιχείο `img` έχει δύο χαρακτηριστικά, τα `src` και `alt`: . Ένα άλλο παράδειγμα θα ήταν το <step number="3">Connect A to B.</step>, όπου το όνομα του χαρακτηριστικού είναι "number" και η τιμή του είναι "3".

Δήλωση XML

Τα κείμενα XML μπορούν να αρχίζουν, με τη δήλωση κάποιων πληροφοριών σχετικών με αυτά, όπως στο ακόλουθο παράδειγμα:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Παράδειγμα

Το παρακάτω είναι ένα μικρό, αλλά πλήρες κείμενο XML, που κάνει χρήση όλων των παραπάνω εννοιών και στοιχείων.

```
<?xml version="1.0" encoding='UTF-8'?>
<painting>
  
  <caption>This is Raphael's "Foligno" Madonna, painted in
  <date>1511</date>-<date>1512</date>.</caption>
</painting>
```

Υπάρχουν πέντε στοιχεία σε αυτό το κείμενο του παραδείγματος: τα `painting`, `img`, `caption`, και δύο `date`. Τα στοιχεία `date`, είναι παιδιά του στοιχείου `caption`, το οποίο είναι παιδί του στοιχείου-ρίζας `painting`. Το στοιχείο `img` έχει δύο χαρακτηριστικά, τα `src` και `alt`.

Τα κείμενα XML αποτελούνται εξ ολοκλήρου από χαρακτήρες Unicode. Εκτός από ένα μικρό αριθμό, ειδικά εξαιρούμενων χαρακτήρων ελέγχου, κάθε χαρακτήρας που ορίζεται στο Unicode, μπορεί να εμφανίζεται στο περιεχόμενο ενός κειμένου XML. Το σύνολο των χαρακτήρων που μπορούν να εμφανίζονται στη σήμανση, αν και κάπως περιορισμένο, παραμένει μεγάλο. Η XML παρέχει κάποιες διευκολύνσεις για την ταυτοποίηση της κωδικοποίησης των χαρακτήρων Unicode που απαρτίζουν ένα κείμενο και για την απεικόνιση χαρακτήρων που, για τον έναν ή τον άλλο λόγο, δεν μπορούν να χρησιμοποιηθούν ευθέως.

2.2.5 ASP

Active Server Pages (ASP), γνωστή και ως κλασική ASP ή ASP Classic, ήταν η πρώτη γραφή της Microsoft server-side μηχανή, για δυναμικά παραγόμενες ιστοσελίδες. Αρχικά κυκλοφόρησε ως add-on για τον Internet Information Services (IIS) μέσω του Windows NT 4.0 Option Pack (για 1998), περιελήφθη στη συνέχεια ως δωρεάν στοιχείο του Windows Server (μετά την αρχική έκδοση του Windows 2000 Server). ASP.NET έχει εκτοπίσει την ASP.

ASP 2,0 παρέχει έξι ενσωματωμένα αντικείμενα: Εφαρμογή, ASPError, Αίτηση, Response, Server και Session. Το Session, για παράδειγμα, αποτελεί ένα cookie με βάση συνόδου που διατηρεί την κατάσταση των μεταβλητών από σελίδα σε σελίδα. Η υποστήριξη της Active Scripting μηχανής του Component Object Model (COM) επιτρέπει ASP ιστοσελίδων για πρόσβαση στις λειτουργίες όπου καταρτίζονται βιβλιοθήκες, όπως η DLLs.

Ιστοσελίδες με το. asp αρχείο επέκταση κάνουν χρήση της ASP, αν και ορισμένες ιστοσελίδες αποκρύπτουν την επιλογή της scripting γλώσσας για λόγους ασφαλείας (π.χ. που χρησιμοποιούν ακόμη και η πιο κοινή. htm ή. html παράταση). Σελίδες με τη χρήση επέκτασης aspx ανοίγουν με ASP.NET (με βάση την NET της Microsoft-πλαίσιο.), η οποία τα καθιστά πιο γρήγορα και πιο ισχυρή από ό, τι server-side scripting σε ASP, το οποίο ερμηνεύεται στο χρόνο εκτέλεσης, ωστόσο, ASP.NET σελίδες μπορεί να εξακολουθεί να περιλαμβάνουν ορισμένες δέσμες ενεργειών ASP. Η εισαγωγή του ASP.NET οδήγησε στη χρήση του όρου Classic ASP για την αρχική τεχνολογία.

Οι προγραμματιστές γράφουν τις περισσότερες σελίδες ASP με VBScript, αλλά και κάθε άλλη μηχανή Active Scripting μπορεί να επιλεγεί αντ 'αυτού με το @ οδηγία Γλώσσα ή το language="language" <script σύνταξη runat="server">. JScript (εφαρμογή της Microsoft για ECMAScript) είναι η άλλη γλώσσα που είναι συνήθως διαθέσιμη. PerlScript (παράγωγο της Perl) και άλλα είναι διαθέσιμα ως τρίτη δυνατότητα εγκατάστασης ως Active Scripting μηχανές.

Παράδειγμα ASP

```
<html>
<body>
<%
response.write("My first ASP script!")
%>
</body>
</html>
```

Θα εμφανίσει το παρακάτω στην οθόνη μας

My first ASP script!

Κεφάλαιο 3

3.1 Εργαλεία Ανάπτυξης Δυναμικών Εφαρμογών

3.1.1 EasyPHP

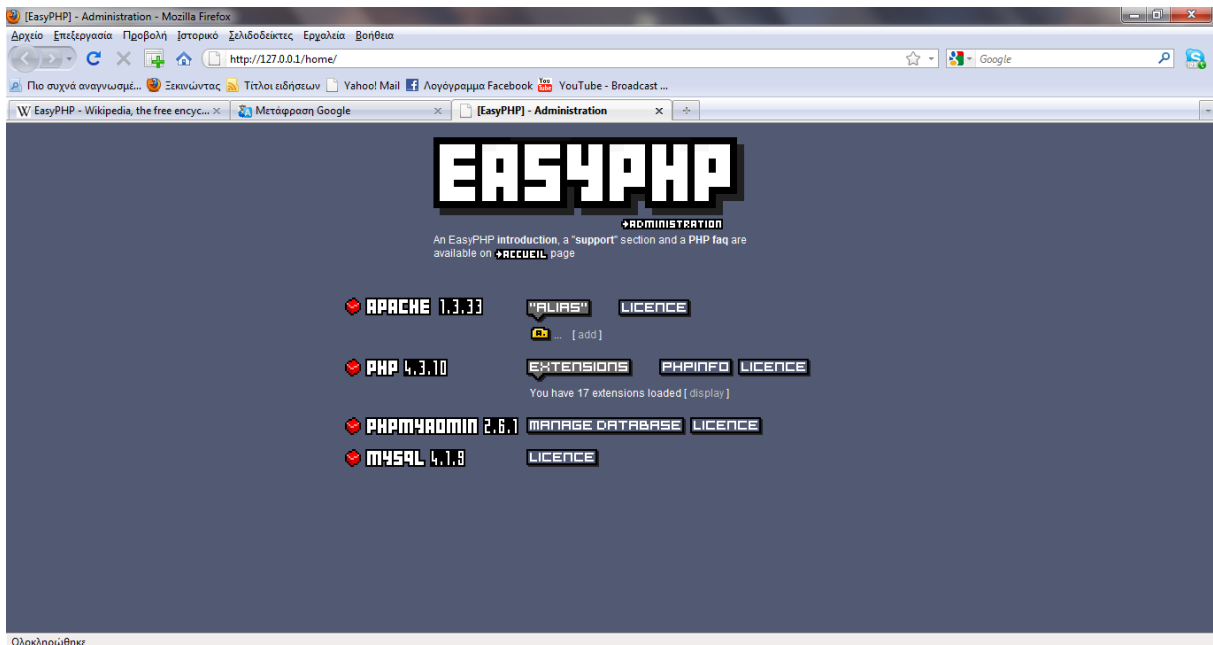
Ξεκινώντας το 1999, το EasyPHP ήταν το πρώτο λογισμικό WAMP διαθέσιμο. Το EasyPHP εγκαθιστά υπηρεσίες τύπου web server στον υπολογιστή, σε λειτουργικό Windows και επιτρέπει τη γρήγορη και εύκολη ανάπτυξη της PHP και MySQL σε localhost (γνωστό και ως 127.0.0.1). Το πακέτο περιλαμβάνει ένα διακομιστή Apache, βάση δεδομένων MySQL, καθώς και η επέκταση της PHP. Εγκαθιστούμε το πρόγραμμα στον υπολογιστή και στον φάκελο www τοποθετούμε όλα τα αρχεία php τα οποία θέλουμε να τρέξουμε και να δοκιμάσουμε στον browser μας. Τα αρχεία html τρέχουν και απευθείας από την επιφάνεια εργασίας.

Για την ανάπτυξη και διανομή διαδικτυακών εφαρμογών είναι απαραίτητο κάποιο εργαλείο, το οποίο να υποστηρίζει την εκτέλεση δυναμικών ιστοσελίδων. Δηλαδή, να διαθέτει έναν web server που να μπορεί να εκτελεί server side scripts καθώς και να υποστηρίζει τη χρήση βάσεων δεδομένων. Η εφαρμογή EasyPHP 1.8, η οποία χρησιμοποιήθηκε στην ανάπτυξη της εφαρμογής, αποτελείται από τα παρακάτω πακέτα λογισμικού:

1. Το διαδικτυακό Εξυπηρετητή (web server) Apache 1.3.33. Στον web server αποθηκεύονται όλα τα αρχεία (HTML, PHP, κ.α.) μίας ιστοσελίδας και είναι υπεύθυνος για την αλληλεπίδραση με τα προγράμματα περιήγησης των επισκεπτών.
2. Τη γλώσσα σεναρίων PHP 4.3.10. Έτσι ώστε να μπορεί να εκτελεί σενάρια στον εξυπηρετητή και να αποστέλλει τα αποτελέσματα στον επισκέπτη.
3. Το εργαλείο για την διαχείριση βάσεων δεδομένων phpMyAdmin 2.6.1. Με αυτή την εφαρμογή, δίνεται η δυνατότητα διαχείρισης των διάφορων βάσεων δεδομένων, με εύκολο τρόπο μέσω web browser και χωρίς την πληκτρολόγηση εντολών.
4. Τον εξυπηρετητή βάσεων δεδομένων MySQL 4.1.9.



Εικόνα 4 Εφαρμογή Easy Php



Εικόνα 5 Σελίδα διαχείρισης της εφαρμογής EasyPHP

3.1.2 Η γλώσσα PHP

Η PHP, όπου τα αρχικά σημαίνουν Hypertext PreProcessor, είναι μια γλώσσα συγγραφής σεναρίων (scripting language) που ενσωματώνεται μέσα στον κώδικα της HTML και εκτελείται στην πλευρά του server (server-side scripting). Ανταγωνιστικές της τεχνολογίας PHP είναι οι εξής γλώσσες προγραμματισμού : ASP (Active Server Pages) της εταιρείας Microsoft, CFML (ColdFusion Markup Language) της εταιρείας Allaire και JSP (JavaServer Pages) της εταιρείας Sun.

Το μεγαλύτερο μέρος της σύνταξής της, η PHP το έχει δανειστεί από την C, την Java και την Perl και διαθέτει και μερικά δικά της μοναδικά χαρακτηριστικά. Ο σκοπός της γλώσσας είναι να δώσει τη δυνατότητα στους web developers να δημιουργούν δυναμικά παραγόμενες ιστοσελίδες

Η PHP είναι μια γλώσσα προγραμματισμού ειδικά για την κατασκευή δυναμικών ιστοσελίδων. Με τον όρο δυναμική εννοείται μια ιστοσελίδα που αλλάζει αυτόματα, ανάλογα με τα στοιχεία του θεατή της. Στοιχεία όπως το λειτουργικό του σύστημα, η διεύθυνση IP του κ.ά. Η PHP χρησιμοποιείται όχι για την διακόσμηση μιας ιστοσελίδας αλλά για τον χειρισμό των λειτουργιών και εργασιών που θα διεκπεραιώνει. Συνεπώς, ο κώδικας που γράφεται για μια ιστοσελίδα σε γλώσσα PHP δεν γίνεται άμεσα αντιληπτός αλλά μετά από την επέμβαση του θεατή στην ιστοσελίδα.

Για να γίνει αυτό κατανοητό: η PHP χρησιμοποιείται ευρέως για τον χειρισμό ιστοσελίδων με δυνατότητες όπως η εγγραφή χρηστών (user registration), τα φόρουμ κ.ά. Λειτουργεί με την βοήθεια της HTML και πλέον και με την XHTML (νέα αναθεωρημένη έκδοση της HTML). Σε συνδυασμό και με την MySQL μπορεί να χρησιμοποιηθεί κάλλιστα για την διαχείριση δεδομένων μέσα σε βάσεις. Λόγω χάρη σε προηγούμενο παράδειγμα της εγγραφής των χρηστών η PHP μπορεί να αποθηκεύει τα ονόματα και τους κωδικούς χρηστών σε μια βάση δεδομένων.

Ακολουθεί ένα εισαγωγικό παράδειγμα :

```
<html>

  <head>

    <title> Παράδειγμα </title>

  </head>

  <body>

    <?php echo "Γεια σας, είμαι ένα script της PHP!"; ?>

  </body>

</html>
```

Προσέξτε πόσο διαφέρει από ένα CGI script που γράφεται σ' άλλες γλώσσες, όπως η Perl ή η C, όπου αντί να γράψουμε ένα πρόγραμμα με πολλές εντολές για να δημιουργήσουμε κώδικα HTML, γράφουμε ένα HTML script με κάποιον ενσωματωμένο κώδικα για να κάνει κάτι, όπως στην συγκεκριμένη περίπτωση να εμφανίσει κάποιο κείμενο (μήνυμα). Ο κώδικας της PHP περικλείεται με ειδικά tags αρχής και τέλους για να μπορούμε να εισερχόμαστε και να εξερχόμαστε από το PHP mode.

Αυτό που ξεχωρίζει την PHP από μια γλώσσα όπως η JavaScript, η οποία εκτελείται στην πλευρά του χρήστη (client-side), είναι ότι ο κώδικάς της εκτελείται στον server. Αν είχαμε σ' έναν server ένα script παρόμοιο με το παραπάνω, ο χρήστης (client) θα λάμβανε το αποτέλεσμα της εκτέλεσης αυτού του script, χωρίς να είναι σε θέση να γνωρίζει ποιος μπορεί να είναι ο αρχικός κώδικας.

Μπορούμε ακόμη να ρυθμίσουμε (configure) τον web server ώστε να επεξεργάζεται όλα τα HTML αρχεία με την PHP και τότε δεν θα υπάρχει πράγματι κανένας τρόπος να μάθουν οι χρήστες τον κώδικά μας.

Ένα σημαντικό στοιχείο το οποίο χαρακτηρίζει την php είναι η δημιουργία φορμών υποβολής στοιχείων ,το οποίο είναι χαρακτηριστικό παράδειγμα χρήσης php.Παρακάτω παραθέτουμε ένα παράδειγμα του πως μπορούμε να φτιάξουμε μια πλήρη τυπική φόρμα υποβολής στοιχείων καθώς και το τι βλέπουμε στην οθόνη μας.

First Name:

Last Name:

Gender:

Male:

Female:

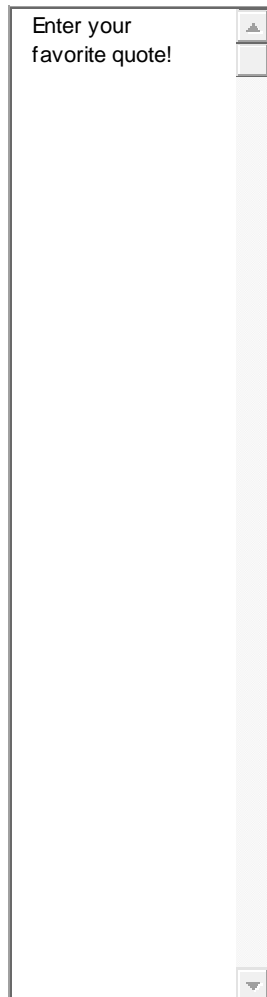
Favorite Food:

Steak:

Pizza:

Chicken:

Enter your favorite quote!



Select a Level of Education:

Select your favorite time of day:

Για να δημιουργήσουμε την παραπάνω φόρμα υποβολής στοιχείων θα πρέπει να γράψουμε τον παρακάτω κώδικα σε ένα αρχείο με κατάληξη php και να το τρέξουμε με ένα server side πρόγραμμα.

```
<?php
```

```
$Fname = $_POST["Fname"];
```

```
$Lname = $_POST["Lname"];
```

```

$gender = $_POST["gender"];
$food = $_POST["food"];
$quote = $_POST["quote"];
$education = $_POST["education"];
$TofD = $_POST["TofD"];
if (!isset($_POST['submit'])) { // if page is not submitted to itself echo the form
?>
<html>
<head>
<title>Personal INFO</title>
</head>
<body>
<form method="post" action="<?php echo $PHP_SELF;?>">
First Name:<input type="text" size="12" maxlength="12" name="Fname"><br />
Last Name:<input type="text" size="12" maxlength="36" name="Lname"><br />
Gender:<br />
Male:<input type="radio" value="Male" name="gender"><br />
Female:<input type="radio" value="Female" name="gender"><br />
Please choose type of residence:<br />
Steak:<input type="checkbox" value="Steak" name="food[]"><br />
Pizza:<input type="checkbox" value="Pizza" name="food[]"><br />
Chicken:<input type="checkbox" value="Chicken" name="food[]"><br />
<textarea rows="5" cols="20" name="quote" wrap="physical">Enter your favorite
quote!</textarea><br />
Select a Level of Education:<br />

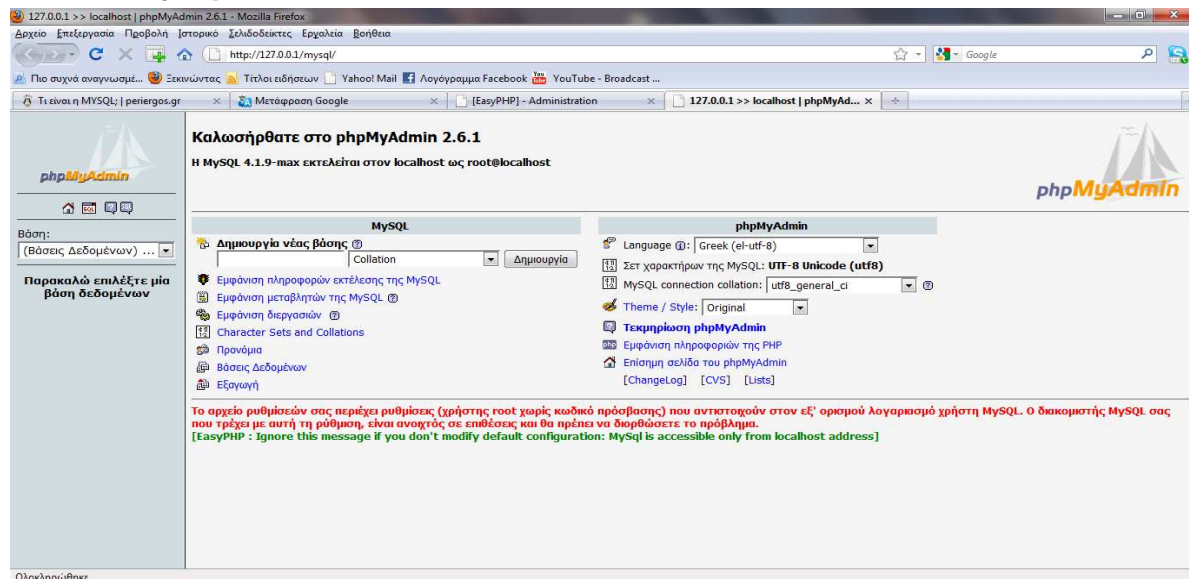
```

```

<select name="education">
<option value="Jr.High">Jr.High</option>
<option value="HighSchool">HighSchool</option>
<option value="College">College</option></select><br />
Select your favorite time of day:<br />
<select name="TofD" size="3">
<option value="Morning">Morning</option>
<option value="Day">Day</option>
<option value="Night">Night</option></select><br />
<input type="submit" value="submit" name="submit">
</form>
<?
} else {
echo "Hello, ".$Fname." ".$Lname."<br />";
echo "You are ".$gender.", and you like ";
foreach ($food as $f) {
echo $f."<br />";
}
echo "<i>".$quote."</i><br />";
echo "You're favorite time is ".$TofD.", and you passed ".$education."<br />";
}
?>

```

3.1.3 Η Mysql



Εικόνα 6.Περιβάλλον διαχείρισης MySQL

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακής βάση ανοικτού κώδικα όπως λέγεται (relational database management system - RDBMS) που χρησιμοποιεί την Structured Query Language (SQL), την πιο γνωστή γλώσσα για την προσθήκη, την πρόσβαση και την επεξεργασία δεδομένων σε μία Βάση Δεδομένων.

Ενα Σύστημα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ) (Database Management System - DBMS) αποτελείται από μία συλλογή σχετιζόμενων δεδομένων και ένα σύνολο προγραμμάτων για την προσπέλαση σε αυτά τα δεδομένα. Η συλλογή των δεδομένων ονομάζεται βάση δεδομένων (ΒΔ) (database). Μια Βάση Δεδομένων είναι μια σημαντική συλλογή δομημένων πληροφοριών που κρατούνται σε ένα μόνιμο αποθηκευτικό χώρο. Ο κύριος σκοπός ενός ΣΔΒΔ είναι να παρέχει ένα περιβάλλον για την εύκολη και αποδοτική αποθήκευση και ανάκληση μεγάλου όγκου πληροφορίας.

Με την MySQL μπορούμε να δημιουργήσουμε πίνακες στους οποίους μπορούμε να καταχωρούμε δεδομένα διαφόρων τύπων. Μπορούμε επίσης να δημιουργήσουμε συσχετίσεις μεταξύ των πινάκων ,να βάλουμε πλυθηκότητες, να γεμίσουμε τους πίνακες και τις συσχετίσεις με τα απαραίτητα στοιχεία, να θέσουμε τα πρωτεύοντα κλειδιά σε κάθε πίνακα, να κάνουμε διαφόρων τύπων ερωτήματα και να πάρουμε τις απαντήσεις που θέλουμε καθώς και τον απαραίτητο κώδικα mysql από όλες τις ενέργειες που εκτελούμε.

Όλα αυτά και πολλά άλλα μπορούμε να τα κάνουμε με την βοήθεια του προγράμματος EasyPHP,όπου μπορούμε να τα κάνουμε γραφικά και όχι σε περιβάλλον DOS με την χρήση εντολών, κάτι το οποίο είναι δύσκολο, χρονοβόρο και πιο επιρρεπές σε λάθη.

Η MySQL είναι πιο κατάλληλη για χρήση στο Internet γιατί:

- Παρέχει ευκολίες στο backup.
- Είναι ιδιαίτερα βελτιωμένη σε ταχύτητα για την ανάκτηση δεδομένων.
- Είναι συμβατή και μεταφέρσιμη σε διάφορες πλατφόρμες και για διάφορα εργαλεία ανάπτυξης.
- Είναι οικονομική.
- Είναι λογισμικό ανοιχτού κώδικα.

Στο κεφάλαιο αυτό θα παρουσιαστεί η γλώσσα SQL. Το ακρωνύμιο SQL προκύπτει από το Structured Query Language. Αρχικά η SQL ονομαζόταν SEQUEL από το Structured English Query Language (στο βιβλίο του Ullman, "Principles of Database Systems", που είναι ένα από τα πρώτα και βασικότερα βιβλία για Βάσεις Δεδομένων, θα τη βρείτε να αναφέρετε ως SEQUEL). Αρχικά η SQL υλοποιήθηκε για το System R που ήταν ένα πειραματικό σύστημα της IBM, αλλά σήμερα εξελίχθηκε και τυποποιήθηκε έτσι ώστε να αποτελεί πρότυπο γλώσσας ερωτήσεων. Έχουν υπάρξει ως πρότυπα οι SQL1, SQL2 και SQL3 σε συνεργασία ANSI και ISO. Η γλώσσα ερωτήσεων SQL δίνει δυνατότητες τόσο για αναζήτηση δεδομένων, όσο και για ορισμό δεδομένων, ενώ είναι δυνατή και η ενσωμάτωσή της μέσα σε γλώσσες προγραμματισμού (εμφωλιασμένη SQL). Είναι λοιπόν τόσο Γλώσσα Ορισμού Δεδομένων όσο και Γλώσσα Χειρισμού Δεδομένων. Παρακάτω θα δούμε τη χρήση της SQL ως γλώσσα χειρισμού δεδομένων, και ως γλώσσα ορισμού δεδομένων. Η γλώσσα ερωτήσεων QBE που παρουσιάζεται είναι μία γλώσσα που βρίσκεται πιο κοντά στη φιλοσοφία του σχεσιακού υπολογισμού από ότι στη φιλοσοφία της σχεσιακής άλγεβρας. Η διαφορά της με την SQL είναι ότι ο χρήστης δεν χρειάζεται να περιγράψει μία καθορισμένη ερώτηση, αλλά συμπληρώνει πίνακες με στοιχεία, δηλαδή ένα παράδειγμα, με βάση το οποίο επιλέγονται τα στοιχεία. QBE σημαίνει Query by Example, ακριβώς για να εκφράσει αυτή τη σχεδίαση των ερωτήσεων με βάση ένα παράδειγμα από έναν πίνακα. Η QBE υλοποιήθηκε από την IBM.

Στην ενότητα αυτή θα περιγράψουμε τις βασικές εντολές αναζήτησης στη γλώσσα ερωτήσεων SQL. Για να μπορέσουμε να γίνουμε πιο κατανοητοί θα χρησιμοποιήσουμε ένα απλό παράδειγμα και μερικά χαρακτηριστικά δεδομένα, ώστε να μπορούμε να έχουμε ένα κοινό πεδίο αναφοράς για την παρουσίαση των εντολών και για τις ασκήσεις. Σε επόμενο παράδειγμα, περιγράφονται οι πίνακες μίας απλής σχεσιακής Βάσης Δεδομένων, που θα τους χρησιμοποιήσουμε για τις ερωτήσεις σε όλο το κεφάλαιο, όταν θα παρουσιάσουμε τη γλώσσα ερωτήσεων QBE. Επίσης, σε επόμενο παράδειγμα δίνουμε μερικά χαρακτηριστικά δεδομένα τα οποία θα χρησιμοποιηθούν στις ασκήσεις και παραδείγματα.

Παράδειγμα :

Ας δούμε μία αρκετά απλή Βάση Δεδομένων: Έστω ότι οργανώνουμε σε Βάση Δεδομένων τα στοιχεία ενός Πανεπιστημίου. Βέβαια για λόγους απλότητας θα θεωρήσουμε ότι στη συγκεκριμένη Βάση Δεδομένων έχουμε το ελάχιστο δυνατό σύνολο πληροφορίας. Έτσι, έχουμε τους παρακάτω πίνακες (με έντονα γράμματα τα είναι τα πεδία που αποτελούν το κλειδί για κάθε πίνακα):

ΤΜΗΜΑ (ΟΝΟΜΑ, ΕΤΟΣ_ΙΔΡΥΣΗΣ)
ΚΑΘΗΓΗΤΗΣ(ΟΝΟΜΑ, ΕΠΩΝΥΜΟ, ΠΑΤΡΩΝΥΜΟ, ΒΑΘΜΙΔΑ, ΗΜ_ΓΕΝ)
ΦΟΙΤΗΤΗΣ(ΟΝΟΜΑ, ΕΠΩΝΥΜΟ, ΠΑΤΡΩΝΥΜΟ, ΑΜ, ΗΜ_ΓΕΝ)
ΜΑΘΗΜΑ(ΚΩΔΙΚΟΣ, ΟΝΟΜΑ, ΕΤΟΣ_ΔΙΔΑΣΚ)

Στους παραπάνω πίνακες θεωρήσαμε, για λόγους απλότητας ότι το όνομα κάθε τμήματος είναι μοναδικό, ότι κάθε μάθημα έχει ένα κωδικό που το χαρακτηρίζει με μοναδικό τρόπο, ότι κάθε φοιτητής έχει μοναδικό Αριθμό Μητρώου και ότι ο Καθηγητής μπορεί να έχει ως κλειδί το ονοματεπώνυμο. Ορίζουμε επίσης μερικές σχέσεις:

ΚΑΘ_ΤΜΗΜΑ (ΟΝΟΜΑ_Κ, ΕΠΩΝΥΜΟ_Κ, ΟΝΟΜΑ_Τ)
ΦΟΙΤ_ΤΜΗΜΑ(ΑΜ, ΟΝΟΜΑ)
ΦΟΙΤ_ΜΑΘ (ΑΜ, ΚΩΔΙΚΟΣ)
ΚΑΘ_ΜΑΘ(ΟΝΟΜΑ, ΕΠΩΝΥΜΟ, ΚΩΔΙΚΟΣ)

Από τις παραπάνω σχέσεις η ΚΑΘ_ΤΜΗΜΑ είναι σχέση N προς M , η ΦΟΙΤ_ΤΜΗΜΑ σχέση N προς 1 , η ΦΟΙΤ_ΜΑΘ σχέση N προς M και η ΚΑΘ_ΜΑΘ σχέση 1 προς N . (Κάνουμε την παραδοχή ότι κάθε μάθημα διδάσκεται μόνο από έναν Καθηγητή).

Εντολές SQL

Τώρα που έχουμε στη διάθεσή μας την περιγραφή των πινάκων μίας Βάσης Δεδομένων και μερικά χαρακτηριστικά δεδομένα, ας περάσουμε στην παρουσίαση των εντολών της SQL. Η πιο βασική εντολή της SQL είναι η εντολή αναζήτησης δεδομένων της μορφής:

```
SELECT [Πεδία]  
FROM [Πίνακες]  
WHERE [Συνθήκη]
```

Το WHERE μπορεί και να παραλείπεται, ενώ η λίστα με τα κατηγορήματα μπορεί να αντικαθίσταται με αστερίσκο (*) που σημαίνει (όλα τα πεδία). Η συνθήκη μπορεί να χρησιμοποιεί τους μαθηματικούς τελεστές >, <, <>, >=, <= και =, καθώς και τους λογικούς τελεστές AND, OR και NOT. Υπάρχουν και άλλοι τελεστές της SQL που μπορούν να εμφανίζονται στη συνθήκη, τους οποίους όμως θα παρουσιάσουμε αργότερα. Ας δούμε μερικά παραδείγματα εντολών σε SQL. Έστω ότι θέλουμε να βρούμε όλα τα στοιχεία όλων των φοιτητών. Η εντολή θα είναι

```
SELECT *  
FROM ΦΟΙΤΗΤΗΣ
```

Το αποτέλεσμα της εκτέλεσης αυτής της εντολής θα είναι να δοθεί ως απάντηση ο πλήρης πίνακας με όλα τα στοιχεία όλων των φοιτητών που περιέχονται στη Βάση Δεδομένων. Έστω τώρα ότι θέλουμε να δούμε μόνο το ονοματεπώνυμο των φοιτητών που έχουν γεννηθεί στις 20 Αυγούστου 1978. Η εντολή τότε θα είναι

```
SELECT ONOMA,ΕΠΩΝΥΜΟ  
FROM ΦΟΙΤΗΤΗΣ  
WHERE HM_GEN = '20-08-1978'
```

Το αποτέλεσμα της εκτέλεσης αυτής είναι ο πίνακας :

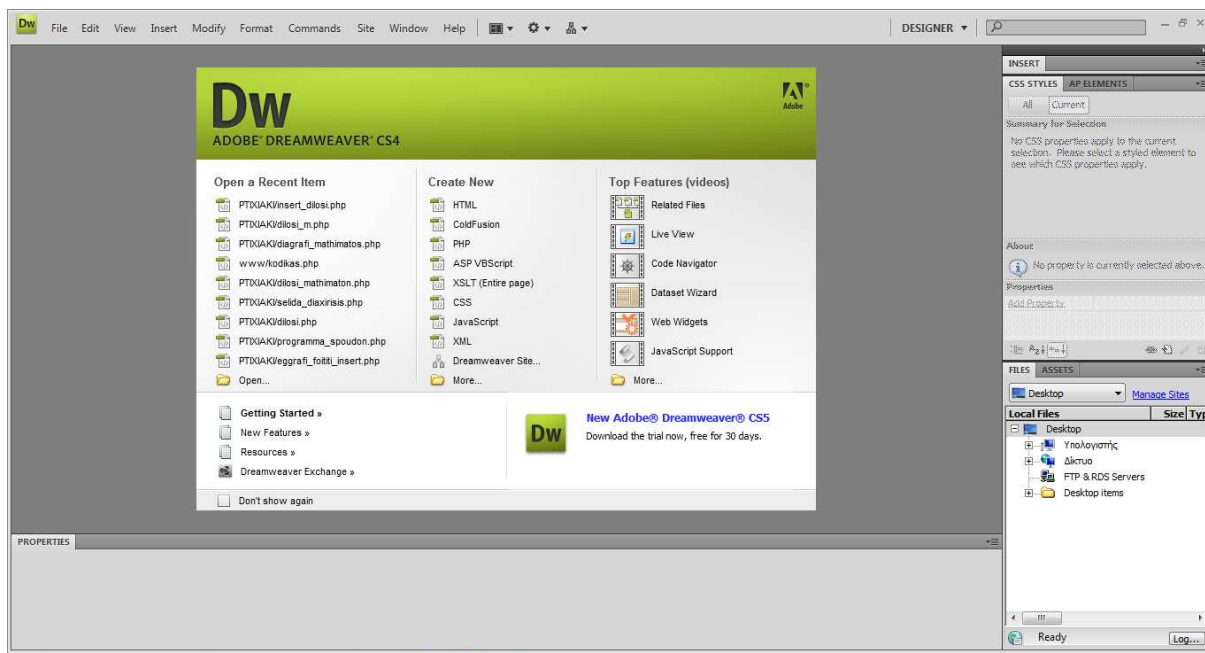
```
ONOMA  ΕΠΩΝΥΜΟ  
Νικόλαος  Βασιλείου  
Χρήστος  Χρήστου
```

3.1.4 Apache Tomcat

Apache Tomcat (ή Jakarta Tomcat ή απλά Tomcat) είναι ένα ανοιχτό λογισμικό servlet, δημιουργημένο και αναπτυγμένο από την Apache Software Foundation (ASF). Ο Tomcat υλοποιεί Java Servlet και JavaServer Pages (JSP) προδιαγραφές από τη Sun Microsystems και παρέχει ένα «καθαρό Java» web HTTP περιβάλλον server για Java κώδικα.

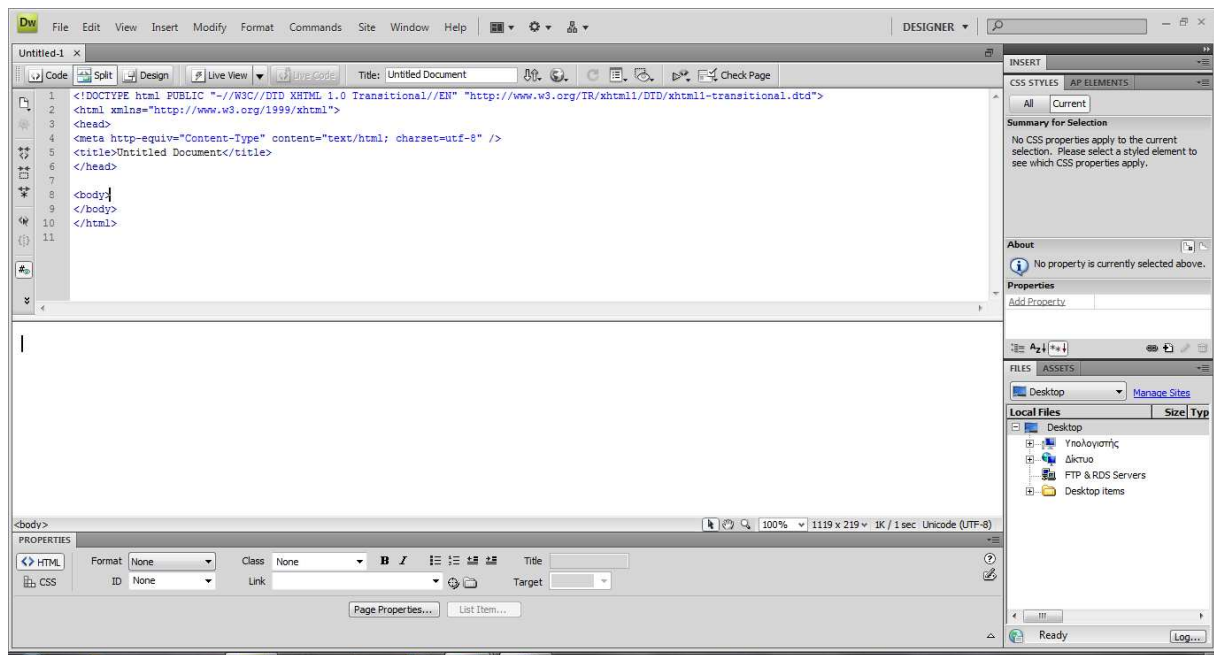
3.1.5 Adobe Dreamweaver CS4

Το Adobe DreamWeaver CS4 είναι μία έκδοση του λογισμικού DreamWeaver, το οποίο χρησιμοποιείται ευρέως στην κατασκευή ιστοσελίδων με την χρήση κώδικα Html και XHTML. Ο χρήστης μπορεί να δημιουργήσει αρχεία τύπου php.html κ.α. Μπορεί επίσης να δημιουργεί ένα περιβάλλον είτε με τη χρήση κώδικά είτε γραφικά.



Εικόνα 7. Το λογισμικό Adobe DreamWeaver CS4

Στην παρακάτω εικόνα φαίνεται αρχικά το περιβάλλον δημιουργίας ιστοσελίδας το οποίο είναι χωρισμένο σε δύο τομείς, έναν γραφικό και έναν για κώδικα. Στην Html μπορούμε να βάλουμε και php κώδικα αλλά επίσης μπορούμε να κάνουμε και το αντίθετο. Το Dreamweaver παρέχει αρκετά εργαλεία για τη δημιουργία και ανάπτυξη μίας δυναμικής ιστοσελίδας. Μερικά από αυτά είναι οι έτοιμες βιβλιοθήκες όπου μπορούμε να βρούμε έτοιμα κουμπιά, λίστες, φόρμες και τα συναφή που χρειάζεται μια φόρμα και πολλά άλλα. Μπορούμε να μορφοποιήσουμε την εφαρμογή μας με την χρήση css και να αποθηκεύσουμε την μορφοποίηση αυτή για μελλοντική χρήση. Όμως το Dreamweaver δεν μας υποδεικνύει τυχόν λάθη στο κώδικα που παράγουμε. Μπορούμε μόνο να τα εντοπίσουμε στον browser όπου ελέγχουμε την εφαρμογή μας αν λειτουργεί και τσεκάρουμε το αποτέλεσμα μας.



Εικόνα 8.Περιβάλλον Δημιουργίας Σελίδας Html

3.1.6 JavaScript

Η γλώσσα JavaScript είναι μια γλώσσα σεναρίων (scripting language) που βοηθά τον προγραμματιστή στην κατασκευή διαδραστικών και λειτουργικών εγγράφων στο Internet. Τα σενάρια γραμμένα σε JavaScript ενσωματώνονται στον κώδικα HTML και εκτελούνται από τον web browser του χρήστη (client-side processing). Για να μπορέσει ένας browser να εκτελέσει ένα σενάριο γραμμένο σε JavaScript θα πρέπει να διαθέτει έναν διερμηνευτή JavaScript (interpreter). Οι γνωστότεροι browsers Firefox, Internet Explorer, Opera ενσωματώνουν διερμηνευτή της JavaScript.

Μεταξύ των δυνατοτήτων που δίνει η Javascript είναι:

- Έλεγχος στο περιεχόμενο και στην εμφάνιση των ιστοσελίδων
- Έλεγχος της συμπεριφοράς και των λειτουργιών του browser
- Αλληλεπίδραση με φόρμες HTML
- Αλληλεπίδραση με το χρήστη με τη βοήθεια γεγονότων
- Ανάγνωση ή καταγραφή του Η/Υ του χρήστη μέσω cookies
- Δυνατότητα για εναλλαγή εικόνων

Από άποψη περιορισμών, η Javascript δεν επιτρέπει προσπέλαση ή χειρισμό αρχείων στον Η/Υ του χρήστη, για λόγους ασφαλείας. Επίσης, τα σενάρια που εκτελούνται στο web browser του χρήστη δεν μπορεί να ενημερώσει αρχεία που βρίσκονται στο web server. Πολλοί πιστεύουν ότι το JavaScript είναι Java λόγω της ομοιότητας του ονόματος. Αυτό, όμως, δεν είναι αλήθεια. Πιστεύουμε ότι θα έπαιρνε πολύ να καταλάβουμε όλες τις διαφορές - γι' αυτό απλώς απομνημονεύουμε ότι το Javascript δεν είναι Java.

Τι χρειάζεται ώστε να τρέξουν scripts γραμμένα σε JavaScript; Χρειαζόμαστε ένα browser που υποστηρίζει JavaScript - παραδείγματος χάριν το Netscape Navigator (από την έκδοση 2.0) ή το Microsoft Internet Explorer (MSIE εω συντομία - από την έκδοση 3.0). Εφόσον αυτοί οι δύο browsers είναι πολύ διαδεδομένοι, πολλοί χρήστες έχουν την δυνατότητα να τρέξουν scripts γραμμένα σε JavaScript.

Αυτό είναι βεβαίως ένας σημαντικός λόγος για να διαλέξουμε JavaScript για να εμπλουτίσουμε τις web- σελίδες μας. Βεβαίως χρειάζεται πρώτα να κατανοήσουμε βασικές λειτουργίες της HTML .

Ένα παράδειγμα κώδικα JavaScript το βλέπουμε παρακάτω :

```
<html>
<body>
<h1>My First Web Page</h1>
<script type="text/javascript">
document. write("<p>" + Date() + "</p>");
</script>
</body>
</html>
```

Ο παραπάνω κώδικας θα μας εμφανίσει στην οθόνη το παρακάτω :

My First Web Page

Sat Mar 26 2011 21:24:24 GMT+0200

Σε browsers που δεν υποστηρίζουν την JavaScript, θα εμφανιστεί το JavaScript, όπως το περιεχόμενο της σελίδας. Για να τους εμποδίσουμε να το κάνουν αυτό και ως μέρος της JavaScript προτύπου, η ετικέτα σχολίου HTML που θα πρέπει να χρησιμοποιηθεί για να "κρύψει" το JavaScript. Απλά προσθέτουμε ένα σχόλιο HTML ετικέτα (στο τέλος του σχολίου) μετά την τελευταία δήλωση της JavaScript, όπως αυτός <- - πριν από την πρώτη δήλωση JavaScript, και ένα!> ,όπως το παρακάτω παράδειγμα:

```
<html>
<body>
<script type="text/javascript">
<!--
document.getElementById("demo").innerHTML=Date();
//-->
</script>
</body>
</html>
```

Κεφάλαιο 4

Κύριο Μέρος της Εφαρμογής

4.1 Ανάλυση Προβλήματος

Μια δυναμική ιστοσελίδα ή εφαρμογή χρειάζεται κάποια εργαλεία για την ανάπτυξή της. Τα εργαλεία αυτά και την θεωρία τους, παρουσιάστηκαν στο κεφάλαιο 2 περιληπτικά. Σε αυτό το κεφάλαιο θα αναπτύξουμε την βασική θεωρία και δομή μίας βάσης δεδομένων, η οποία είναι απαραίτητη για την υλοποίηση μιας δυναμικής ιστοσελίδας καθώς χωρίς αυτήν δεν θα είχαμε δεδομένα προς αποθήκευση και επεξεργασία, κάτι το οποίο είναι βασικό στην πληροφορική και ειδικότερα στο διαδίκτυο, μιας και το τελευταίο βασίζεται στην συνεχή ροή και ανανέωση της πληροφορίας και των δεδομένων.

Χωρίς τις δυναμικές ιστοσελίδες το διαδίκτυο θα ήταν σαν μια εικόνα σταθερή την οποία θα βλέπαμε μεν αλλά δεν θα μπορούσαμε να επιδράσουμε πάνω τις, να αντλήσουμε πληροφορίες από αυτή ή να δώσουμε πληροφορίες.

Το πρόβλημα στην εφαρμογή αυτή είναι η δημιουργία μιας τέτοιας βάσης αποθήκευσης και επεξεργασίας δεδομένων και έπειτα η διασύνδεση της με το interface μας, δηλαδή το οπτικό μέρος της εργασίας και τις λειτουργίες του, τις οποίες και υλοποιούμε με την php και την JavaScript.

4.2 Απαιτήσεις του Συστήματος

4.2.1 Βάσεις Δεδομένων

Με τον όρο βάση δεδομένων εννοείται μία συλλογή από *συστηματικά οργανωμένα* (formatted) σχετιζόμενα δεδομένα. Ένας τηλεφωνικός κατάλογος, για παράδειγμα, θεωρείται βάση δεδομένων, καθώς αποθηκεύει και οργανώνει σχετιζόμενα τμήματα πληροφορίας, όπως είναι το όνομα και ο αριθμός τηλεφώνου. Ωστόσο, στον κόσμο των υπολογιστών, με τον όρο βάση δεδομένων αναφερόμαστε σε μια συλλογή σχετιζόμενων δεδομένων τμημάτων πληροφορίας ηλεκτρονικά αποθηκευμένων.

Πέρα από την εγγενή της ικανότητα να αποθηκεύει δεδομένα, η βάση δεδομένων παρέχει βάσει του σχεδιασμού και του τρόπου ιεράρχησης των δεδομένων της σε προγράμματα ή συλλογές προγραμμάτων, τα αποκαλούμενα συστήματα διαχείρισης περιεχομένου, τη δυνατότητα γρήγορης άντλησης και ανανέωσης των δεδομένων. Η ηλεκτρονική βάση δεδομένων χρησιμοποιεί ιδιαίτερου τύπου λογισμικό προκειμένου να οργανώσει την αποθήκευση των δεδομένων της. Το διακριτό αυτό λογισμικό είναι γνωστό ως Σύστημα διαχείρισης βάσης δεδομένων συντομευμένα (DBMS).

Σε μικρές Βάσεις Δεδομένων το μέσο αυτό συνήθως είναι ο σκληρός δίσκος του υπολογιστή, ενώ σε μεγάλες Βάσεις Δεδομένων χρησιμοποιούνται πολλαπλά επίπεδα αποθήκευσης που ποικίλουν από σκληρούς δίσκους διαφορετικής ταχύτητας και χωρητικότητας, juke boxes από CDs, μαγνητικές ταινίες κτλ.

Ο χειρισμός των δεδομένων στο υλικό του υπολογιστή γίνεται από το λογισμικό του Συστήματος Διαχείρισης Βάσεων Δεδομένων το οποίο ονομάζεται □Λογισμικό Αποθήκευσης και Ανάκλησης Δεδομένων□. Αυτό το λογισμικό αναλαμβάνει να μεταφέρει όλες τις εντολές του χρήστη προς το υλικό. Το λογισμικό αυτό συνεργάζεται με το άλλο τμήμα του Συστήματος Διαχείρισης Βάσεων Δεδομένων που είναι το □Λογισμικό Χειρισμού Ερωτήσεων□. Όλα τα Συστήματα Διαχείρισης Βάσεων Δεδομένων παρέχουν μία γλώσσα υποβολής ερωτήσεων προς τη Βάση Δεδομένων.

Αυτές οι ερωτήσεις επεξεργάζονται από το Λογισμικό Χειρισμού Ερωτήσεων και στη συνέχεια μετατρέπονται σε εντολές για ανάκληση συγκεκριμένων δεδομένων από το υλικό και μεταφέρονται στο Λογισμικό Αποθήκευσης και Ανάκλησης Δεδομένων. Ο χρήστης μίας Βάσης Δεδομένων δεν αντιλαμβάνεται τη διαδικασία, όπως την υλοποιεί το Σύστημα Διαχείρισης Βάσεων Δεδομένων. Συνήθως δεν γνωρίζει που βρίσκονται τα δεδομένα (δηλαδή σε ποιο υλικό είναι αποθηκευμένα). Ο χρήστης απλά εκτελεί λειτουργίες (εισαγωγή, διαγραφή, ή τροποποίηση) στα δεδομένα και υποβάλλει ερωτήσεις. Βλέπει μόνο τη Βάση Δεδομένων χωρίς να τον απασχολούν οι λειτουργίες του Συστήματος Διαχείρισης Βάσεων Δεδομένων. Τώρα που έχετε κατανοήσει τι είναι Σύστημα Διαχείρισης Βάσεων Δεδομένων πρέπει να σας πούμε ότι η χρήση του δεν είναι ο μόνος τρόπος για να δημιουργήσουμε μία Βάση Δεδομένων. Θεωρητικά κάποιος μπορεί να ετοιμάσει μία Βάση Δεδομένων υλοποιώντας όλο το λογισμικό που χρησιμοποιεί το Σύστημα Διαχείρισης Βάσεων Δεδομένων από την αρχή και κατασκευάζοντας εξειδικευμένο λογισμικό μόνο για τη συγκεκριμένη Βάση Δεδομένων. Αυτή η προσέγγιση θα είχε πολύ μεγαλύτερο κόστος και χρόνο υλοποίησης, αφού θα έπρεπε να υλοποιήσει από την αρχή όλο το λογισμικό που παρέχεται έτοιμο από ένα Σύστημα Διαχείρισης Βάσεων Δεδομένων. Αν και σε ορισμένες περιπτώσεις ακολουθείται αυτή η πρακτική, ο κανόνας είναι οι Βάσεις Δεδομένων να υλοποιούνται και να χρησιμοποιούνται μέσω ενός Συστήματος Διαχείρισης Βάσεων Δεδομένων.

4.2.2 Τι παρέχει ένα Σύστημα Διαχείρισης Βάσης Δεδομένων

Όπως σας αναφέραμε στην ενότητα μία Βάση Δεδομένων μπορεί να υλοποιηθεί και αυτόνομα χωρίς να χρησιμοποιηθεί ένα Σύστημα Διαχείρισης Βάσεων Δεδομένων. Όμως κατά κανόνα οι παροχές ενός Συστήματος Διαχείρισης Βάσεων Δεδομένων είναι αυτές που τελικά ωθούν τους χρήστες να το χρησιμοποιήσουν. Στην ενότητα αυτή θα περιγράψουμε τι παρέχει ένα Σύστημα Διαχείρισης Βάσεων Δεδομένων.

Όπως ήδη γνωρίζετε ένας χρήστης μίας Βάσης Δεδομένων θέλει αρχικά να σχεδιάσει τη Βάση Δεδομένων και να ορίσει τους τύπους δεδομένων που θα χρησιμοποιεί, μετά να δημιουργήσει τη Βάση Δεδομένων και στη συνέχεια να χρησιμοποιεί τη Βάση Δεδομένων ή να επιτρέπει και σε άλλους χρήστες να τη χρησιμοποιούν ταυτόχρονα με αυτόν. Η υλοποίηση του λογισμικού που πραγματοποιεί αυτές τις διαδικασίες δε χρειάζεται να γίνει, μια και όλα τα παραπάνω παρέχονται από το Σύστημα Διαχείρισης Βάσεων Δεδομένων. Επίσης το σύστημα παρέχει στον χρήστη ένα ενιαίο τρόπο για να ορίζει τις Βάσεις Δεδομένων του και μια φιλοσοφία που τον καθοδηγεί στις ενέργειές του.

Επειδή μία Βάση Δεδομένων μπορεί να χρησιμοποιείται από πολλούς χρήστες με διαφορετικές ανάγκες και γνώσεις, το Σύστημα Διαχείρισης Βάσεων Δεδομένων παρέχει τη δυνατότητα ορισμού διαφορετικών κατηγοριών χρηστών. Για κάθε κατηγορία χρηστών ορίζονται τα όρια εξουσιοδότησης που παρέχονται σε κάθε επίπεδο. Κάποια δεδομένα είναι πιθανόν να είναι εμπιστευτικά και να μην επιτρέπεται να έχουν πρόσβαση σε αυτά όλοι οι χρήστες. Επίσης, κάθε κατηγορία χρηστών θα πρέπει να έχει το δικαίωμα να επιτελεί διαφορετικές λειτουργίες από τις άλλες κατηγορίες. Κάποιοι χρήστες για παράδειγμα θα μπορούν μόνο να διαβάζουν δεδομένα, ενώ άλλοι θα μπορούν να εισάγουν ή να κάνουν αλλαγές στα δεδομένα. Εισαγωγή και τροποποίηση δεδομένων θα μπορεί να κάνει καθένας που θα θέλει να χρησιμοποιήσει τη Βάση Δεδομένων; Προφανώς όχι. Η δυνατότητα αυτή θα πρέπει να δίνεται μόνο στο εξουσιοδοτημένο προσωπικό, ενώ οι υπόλοιποι θα πρέπει να μπορούν μόνο να βλέπουν τα δεδομένα. Το Σύστημα Διαχείρισης Βάσεων Δεδομένων παρέχει διαφορετικά επίπεδα χρηστών ανάλογα με τις δυνατότητες και απαιτήσεις καθενός. Επίσης το σύστημα φροντίζει για την τήρηση των κανόνων ασφαλείας και ελέγχει την πρόσβαση σε αυτό και στις Βάσεις Δεδομένων και δεν επιτρέπει παραβάσεις των επιπέδων πρόσβασης. Τα επίπεδα

πρόσβασης δεν ισχύουν μόνο για τις Βάσεις Δεδομένων, αλλά και για το ίδιο το Σύστημα Διαχείρισης Βάσεων Δεδομένων. Αναλυτικά για τα επίπεδα χρηστών θα συζητήσουμε σε άλλη ενότητα. Το Σύστημα Διαχείρισης Βάσεων Δεδομένων παρέχει δυνατότητες για τήρηση αντιγράφων ασφαλείας καθώς και δυνατότητες ανάληψης της Βάσης Δεδομένων μετά από βλάβες. Έτσι εξασφαλίζει καλύτερη προστασία των δεδομένων.

Η ανάληψη (ή ανάνηψη) των δεδομένων είναι η διαδικασία κατά την οποία το Σύστημα Διαχείρισης Βάσεων Δεδομένων είτε αυτόματα (χωρίς τη συμβολή του διαχειριστή) είτε χειροκίνητα (με τη συμβολή του διαχειριστή) ανακτά τα δεδομένα που χάθηκαν έπειτα από μία βλάβη.

Σε μία Βάση Δεδομένων μπορούν να τεθούν περιορισμοί ορθότητας. Τέτοιοι περιορισμοί σχετίζονται με την οργάνωση των δεδομένων στη Βάση, ή τις τιμές που μπορούν να λάβουν κάποια δεδομένα. Για παράδειγμα σε Μετεωρολογική Βάση Δεδομένων που θα μπορούσαμε να ορίσουμε ότι η τιμή της θερμοκρασίας θα κυμαίνεται από -50 έως $+50$ βαθμούς. Με αυτόν τον τρόπο αν κάποιος εισήγαγε κατά λάθος την τιμή 177 αντί για 17 το Σύστημα Διαχείρισης Βάσεων Δεδομένων θα εντόπιζε το λάθος και δε θα δεχόταν την τιμή. (Θα αντιλαμβανόταν δηλαδή ότι η τιμή 177 δεν μπορεί να είναι πραγματική τιμή για θερμοκρασία.) Με αυτό τον τρόπο το σύστημα υλοποιεί ελέγχους και προστατεύει τη Βάση Δεδομένων από λαθεμένες εισαγωγές δεδομένων. Προσέξτε όμως: Ακόμα και με τον παραπάνω περιορισμό, κάποιος θα μπορούσε να κάνει λάθος και αντί για 17 να εισάγει 27 . Αυτό το λάθος δεν μπορεί να εντοπιστεί από το Σύστημα Διαχείρισης Βάσεων Δεδομένων, αλλά μόνο από τους χρήστες.

Το Σύστημα Διαχείρισης Βάσεων Δεδομένων πραγματοποιεί και ελέγχους για πλεονασμό δεδομένων. Υποθέστε ότι, αντί για Σύστημα Διαχείρισης Βάσεων Δεδομένων χρησιμοποιούσαμε μία Βάση Δεδομένων με καρτέλες και ότι για την τήρηση των αρχείων ήταν επιφορτισμένοι δύο υπάλληλοι. Από κάποιο λάθος λοιπόν έγραψαν και οι δύο καρτέλα για την Αθήνα – Ιανουάριος 1998. Εκτός από το πρόβλημα ότι θα είχαμε διπλά τα δεδομένα, θα μπορούσαν να προκύψουν μεγαλύτερα προβλήματα που σχετίζονται με την συμβατότητα των δεδομένων. Υποθέστε ότι εντοπίστηκε ένα λάθος στις μετρήσεις και η τιμή για την ημερομηνία 16 Ιανουαρίου πρέπει να αλλάξει και από 21 °C πρέπει να γίνει 18 °C. Εάν κάποιος αναλάβει να κάνει την αλλαγή θα την κάνει σε μία μόνο καρτέλα, ενώ η άλλη θα συνεχίζει να έχει την λάθος τιμή. Αυτό σημαίνει ότι η Βάση Δεδομένων θα περιέχει λάθος δεδομένα, παρά τις διορθώσεις. Στην περίπτωση του Συστήματος Διαχείρισης Βάσεων Δεδομένων οι πλεονασμοί ελέγχονται αυτόματα από το σύστημα και η συμβατότητα των δεδομένων προστατεύεται από το ίδιο το Σύστημα Διαχείρισης Βάσεων Δεδομένων απλοποιώντας την εργασία του χρήστη.

4.2.3 Το Μοντέλο Οντοτήτων-Συσχετίσεων

Το μοντέλο οντοτήτων-συσχετίσεων (*μοντέλο Ο/Σ – ER model*) είναι ένα αφαιρετικό ιδεατό μοντέλο δεδομένων, τα οποία έχουν καθορισμένη δομή. Στη μηχανική λογισμικού χρησιμοποιείται για να παρέχει ένα εννοιολογικό σχήμα κατά τη σχεδίαση βάσεων δεδομένων, ως μοντέλο δεδομένων ενός συστήματος και των απαιτήσεών του με top-down προσέγγιση. Ένα διάγραμμα που δημιουργείται με αυτή τη διαδικασία σχεδίασης καλείται διάγραμμα οντοτήτων-συσχετίσεων, ή *διάγραμμα Ο/Σ* ή *ΟΣΔ* εν συντομία. Προτάθηκε αρχικά το 1976 από τον Peter Chen, ωστόσο στη συνέχεια επινοήθηκαν πολλές παραλλαγές της διαδικασίας.

4.2.4 Χρήση μοντέλου οντοτήτων-συσχετίσεων

Χρησιμοποιείται στο πρώτο στάδιο σχεδίασης ενός συστήματος πληροφοριών, κατά την ανάλυση των απαιτήσεών του. Σκοπός του είναι να περιγράψει τις αναγκαίες πληροφορίες οι οποίες πρόκειται να αποθηκευτούν στη βάση δεδομένων ή τον τύπο τους.

Η μοντελοποίηση δεδομένων γίνεται για την περιγραφή των χρησιμοποιούμενων όρων και των σχέσεών τους σε έναν ορισμένο τομέα ενδιαφέροντος. Στην περίπτωση σχεδιασμού ενός συστήματος πληροφοριών, που στηρίζεται σε μια βάση δεδομένων, το εννοιολογικό μοντέλο δεδομένων χαρτογραφείται σε προχωρημένο στάδιο σε ένα λογικό μοντέλο δεδομένων, όπως το σχεσιακό μοντέλο δεδομένων. Το στάδιο αυτό ονομάζεται συνήθως στάδιο λογικού σχεδιασμού. Ύστερα, κατά τη διάρκεια του φυσικού σχεδιασμού το λογικό μοντέλο χαρτογραφείται σε κάποιο φυσικό μοντέλο. Ας σημειωθεί ότι ορισμένες φορές και οι δύο φάσεις αναφέρονται ως «φυσικός σχεδιασμός».

4.2.5 Σχεδίαση-ανάλυση μοντέλου οντοτήτων συσχετίσεων της εφαρμογής

Ανάλυση Σχεσιακού Μοντέλου

Η κάθε οντότητα έχει τα γνωρίσματά της τα οποία μπορεί να έχουν κάποια τιμή ή να είναι κενά. Κάθε οντότητα όμως έχει και ένα γνώρισμα κλειδί το οποίο καθορίζει το ποιό στοιχείο θέλουμε να επιλέξουμε κάθε φορά, κάτι δηλαδή σαν φίλτρο. Η συσχέτιση μεταξύ των οντοτήτων μάθημα και φοιτητής είναι το «παίρνει». Ένα μάθημα μπορούν να το πάρουν πολλοί φοιτητές (1,N), και ένας φοιτητής μπορεί να πάρει πολλά μαθήματα (1,N). Η συσχέτιση παίρνει έχει και αυτή κάποια γνωρίσματα, μερικά από τα οποία είναι ίδια με αυτά των οντοτήτων που συσχετίζεται. Αυτό γίνεται για να μπορεί να συνδεθεί η μία οντότητα με την άλλη, ώστε σε μια ερώτηση στην βάση να μπορούμε να πάρουμε το σωστό αποτέλεσμα με τα σωστά στοιχεία.

Η συσχέτιση μεταξύ των οντοτήτων μάθημα και πρόγραμμα σπουδών είναι το «ανήκει». Ένα μάθημα ανήκει σε πολλά προγράμματα σπουδών (1,N), και σε ένα πρόγραμμα σπουδών ανήκει σε πολλά μαθήματα (1,N).

Η συσχέτιση μεταξύ των οντοτήτων μάθημα και καθηγητής είναι το «διδάσκει». Ένα μάθημα έχει ένα καθηγητή (1,1), και ένας καθηγητής μπορεί να διδάξει πολλά μαθήματα (1,N). Τέλος η οντότητα μάθημα συνδέεται με την συσχέτιση προαπαιτούμενα. Ένα μάθημα μπορεί να έχει πολλά προαπαιτούμενα μαθήματα (1,N) και τα προαπαιτούμενα μαθήματα μπορεί να βρίσκονται σε πολλά μαθήματα (1,N).

Μετατροπή Μοντέλου Οντοτήτων-Συσχετίσεων Σε Σχεσιακό Μοντέλο

Ένα σύστημα διαχείρισης βάσης δεδομένων (ΣΔΒΔ) (*database management system (DBMS)*) αποτελείται από ένα σύνολο δεδομένων και προγράμματα πρόσβασης στα δεδομένα αυτά. Το σύνολο των δεδομένων καλείται βάση δεδομένων (*database*). Στόχος του ΣΔΒΔ είναι η εύκολη και γρήγορη χρήση και ανάκτηση των δεδομένων. Η διαχείριση των δεδομένων περιλαμβάνει:

- τον ορισμό δομών για τη αποθήκευση των δεδομένων
- τον ορισμό μεθόδων για τη διαχείριση των δεδομένων

Ο ορισμός της δομής της βάσης δεδομένων βασίζεται σε ένα μοντέλο δεδομένων το οποίο ορίζει τον τρόπο που περιγράφονται τα δεδομένα, οι σχέσεις τους, η σημασία τους και οι περιορισμοί πάνω στα δεδομένα αυτά.

Το σχεσιακό μοντέλο (*relational model*) δεδομένων παριστάνει δεδομένα και τις σχέσεις τους ως ένα σύνολο πινάκων. Κάθε πίνακας (*table*) αποτελείται από στήλες (*columns*) με μοναδικά ονόματα. Μια γραμμή (*row*) του πίνακα παριστάνει μια σχέση (*relationship*) ανάμεσα σε ένα σύνολο από τιμές. Ο πίνακας που ακολουθεί παριστάνει έναν τηλεφωνικό κατάλογο. Αποτελείται από δύο στήλες και πέντε γραμμές.

Όνομα	Τηλέφωνο
Γιώργος	32560
Μαρία	61359
Θανάσης	98756
Λίνα	78999
Πέτρος	12356

Η SQL (structured query language) αποτελεί σήμερα την πιο διαδεδομένη γλώσσα διαχείρισης σχεσιακών βάσεων δεδομένων. Η SQL παρέχει δυνατότητες για:

- τον ορισμό, τη διαγραφή και τη μεταβολή πινάκων και κλειδιών,
- τη σύνταξη ερωτήσεων (*queries*),
- την εισαγωγή, διαγραφή και μεταβολή στοιχείων,
- τον ορισμό όψεων (*views*) πάνω στα δεδομένα,
- τον ορισμό δικαιωμάτων πρόσβασης,
- τον έλεγχο της ακεραιότητας των στοιχείων,
- τον έλεγχο συναλλαγών (*transaction*)

Η SQL είναι ορισμένη ως διεθνές πρότυπο. Στις επόμενες ενότητες θα εξετάσουμε ένα υποσύνολο της SQL όπως υποστηρίζεται από την εγκατεστημένη στα εργαστήρια βάση δεδομένων Microsoft Access.

Στην περιγραφή της σύνταξης της SQL θα χρησιμοποιήσουμε τα παρακάτω σύμβολα:

[έκφραση]

η έκφραση εμφανίζεται προαιρετικά

έκφραση1 | έκφραση2

μπορεί να γραφεί η έκφραση1 ή η έκφραση2

έκφραση ...

η έκφραση μπορεί να επαναληφθεί

Βάσει της θεωρίας για να μετατρέψουμε το παρακάτω μοντέλο οντοτήτων-συσχετίσεων σε σχεσιακό θα πρέπει να γραφτούν οι οντότητες με κεφαλαία γράμματα και τα γνωρίσματα τους, με μικρά, μέσα στις παρενθέσεις καθώς και οι συσχετίσεις με τα δικά τους γνωρίσματα, όπως παρακάτω :

ΦΟΙΤΗΤΗΣ (ΑΜ, Όνομα, Επώνυμο, Όνομα Πατρός, Όνομα Μητρός, Α.Δ.Τ, Μ.Ο, Σειρά Επιτυχίας, Διεύθυνση Μόνιμη, Διεύθυνση Τοπική, Τηλ Οικίας, Κινητό Τηλ, Email)

ΠΑΙΡΝΕΙ (Κωδικός, ΑΜ, Βαθμός, Εξάμηνο, Όνομα Μαθήματος)

ΜΑΘΗΜΑ (Κωδικός, Εξάμηνο, Δ.Μ, Τύπος Μαθήματος, Ονομασία Μαθήματος)

ΠΡΟΑΠΑΙΤΟΥΜΕΝΑ (Κωδικός Προηγούμενου, Κωδικός Επόμενου)

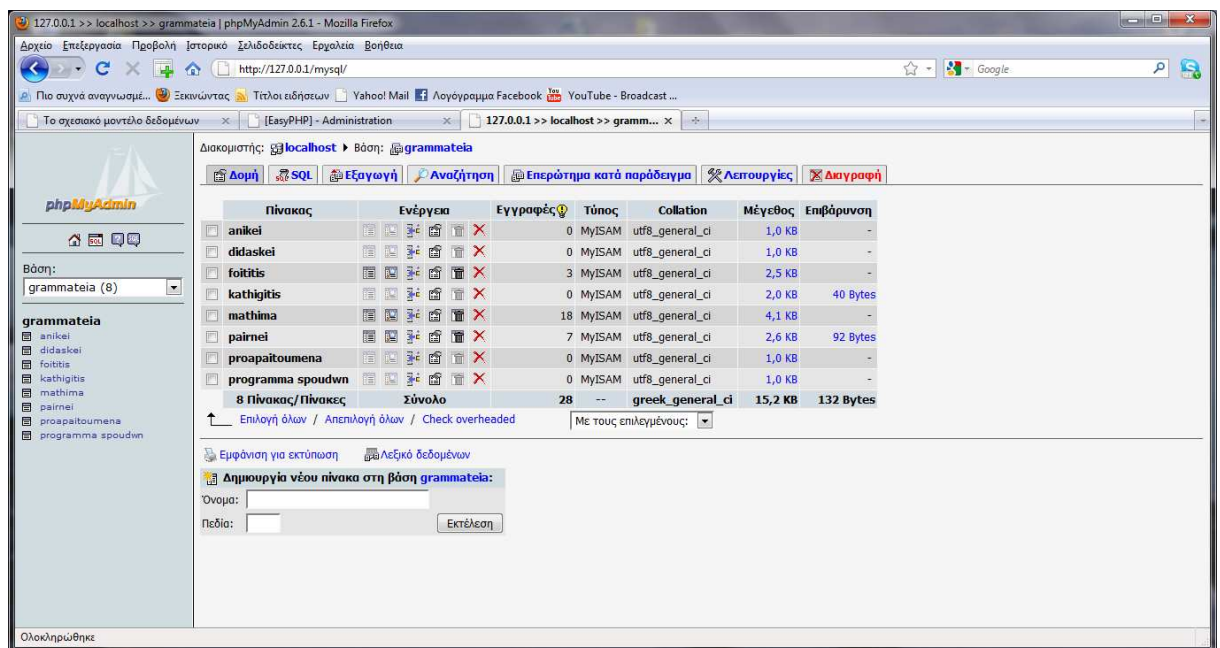
ΔΙΔΑΣΚΕΙ (Α.Δ.Τ)

ΑΝΗΚΕΙ (Εξάμηνο)

ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ (Κωδικός Μ, Κωδικός, Από-Έως)

ΚΑΘΗΓΗΤΗΣ (Όνομα, Επώνυμο, Α.Δ.Τ, Τηλ Οικίας, Βαθμίδα, Τηλ Γραφείου)

Μετατρέποντας το μοντέλο οντοτήτων συσχετίσεων σε σχεσιακό, μας βοηθά περισσότερο στο να στήσουμε την βάση δεδομένων μας στο πρόγραμμά μας, να δημιουργήσουμε δηλαδή τους κατάλληλους πίνακες με τα γνώρισμα τους και τα απαραίτητα ξένα και πρωτεύοντα κλειδιά τα οποία θα πραγματοποιήσουν την διασύνδεση των πινάκων με τις συσχετίσεις και τις πλυθηκότητες, όπως έχουμε κάνει στο παρακάτω διάγραμμα.

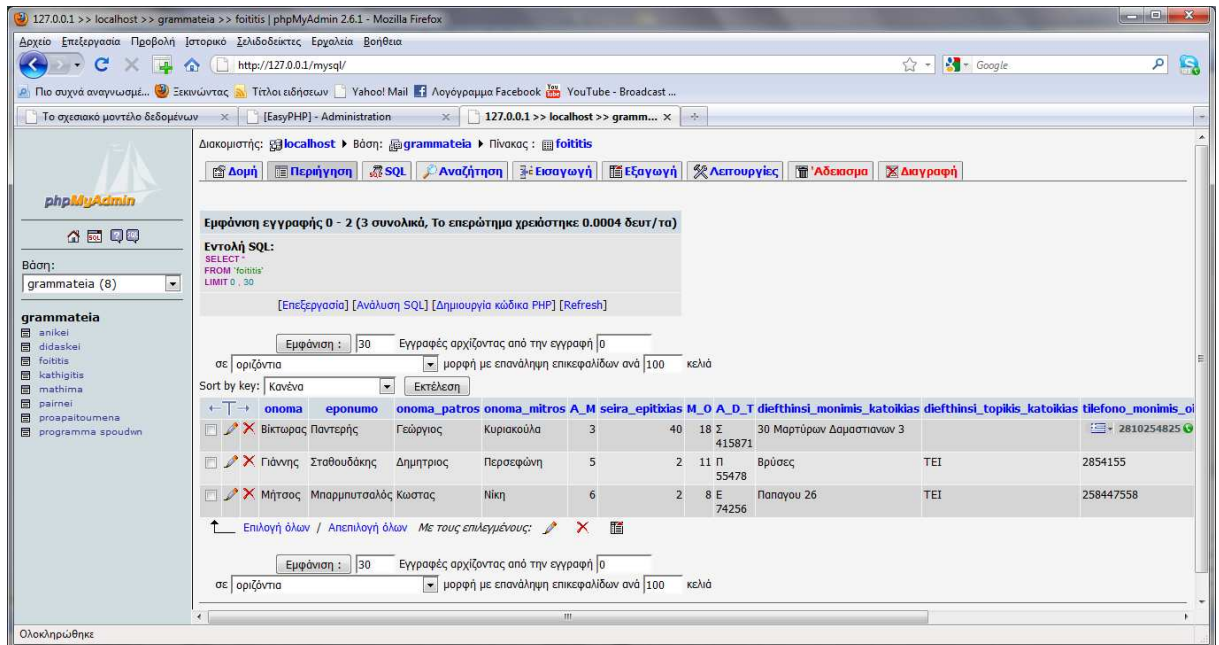


Εικόνα 9.Η Δομή της Βάσης "Γραμματεία"

Στην παραπάνω εικόνα βλέπουμε την δομή την βάσης δεδομένων πάνω στην οποία είναι στημένη η εφαρμογή μας. Υπάρχει και μία άλλη βάση η οποία είναι για τους κωδικούς εισόδου την οποία και θα αναλύσουμε παρακάτω. Βλέπουμε λοιπόν στο πρόγραμμά μας κάποια χαρακτηριστικά τα οποία εμφανίζονται για κάθε οντότητα που έχουμε εισάγει. Για παράδειγμα, η οντότητα φοιτητής, έχει κάποιες εγγραφές σαν πίνακας, τις οποίες μπορούμε να διαχειριστούμε όπως θέλουμε, δηλαδή να τις μετατρέψουμε, ή να τις σβήσουμε ή να αλλάξουμε όποια χαρακτηριστικά του πίνακα θέλουμε.

Φαίνεται επίσης το ποσό των εγγραφών σε κάθε πίνακα καθώς και ο τύπος κωδικοποίησης των χαρακτήρων που εισάγονται στον πίνακα. Αναγράφεται το μέγεθος που καταλαμβάνει ο πίνακας στην βάση κάθε φορά που μετατρέπεται.

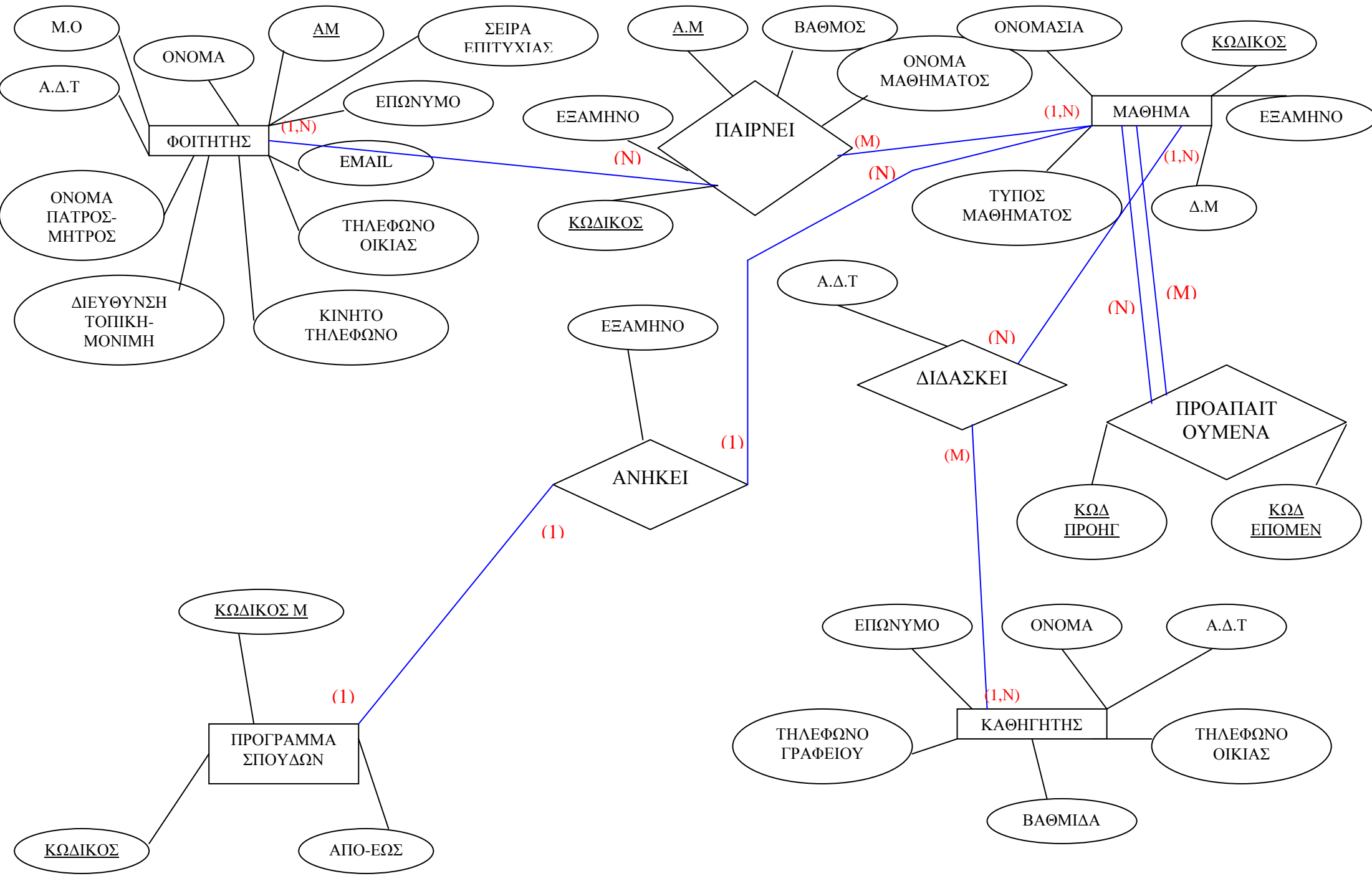
Εάν κάνουμε κλικ στο κουτάκι που είναι για παράδειγμα δίπλα από τον πίνακα φοιτητής ,το οποίο γράφει περιήγηση, θα εμφανιστεί αυτό που βλέπουμε στη παρακάτω εικόνα



Εικόνα 10.Περιήγηση Στον Πίνακα “Φοιτητής”

Εδώ λοιπόν παρατηρούμε ότι το πρόγραμμά μας εμφανίζει πάντα σε ένα παράθυρο τον κώδικα SQL που εκτελέστηκε κατά την ενέργεια που πραγματοποιήσαμε, πατώντας δηλαδή το κουμπί της περιήγησης.

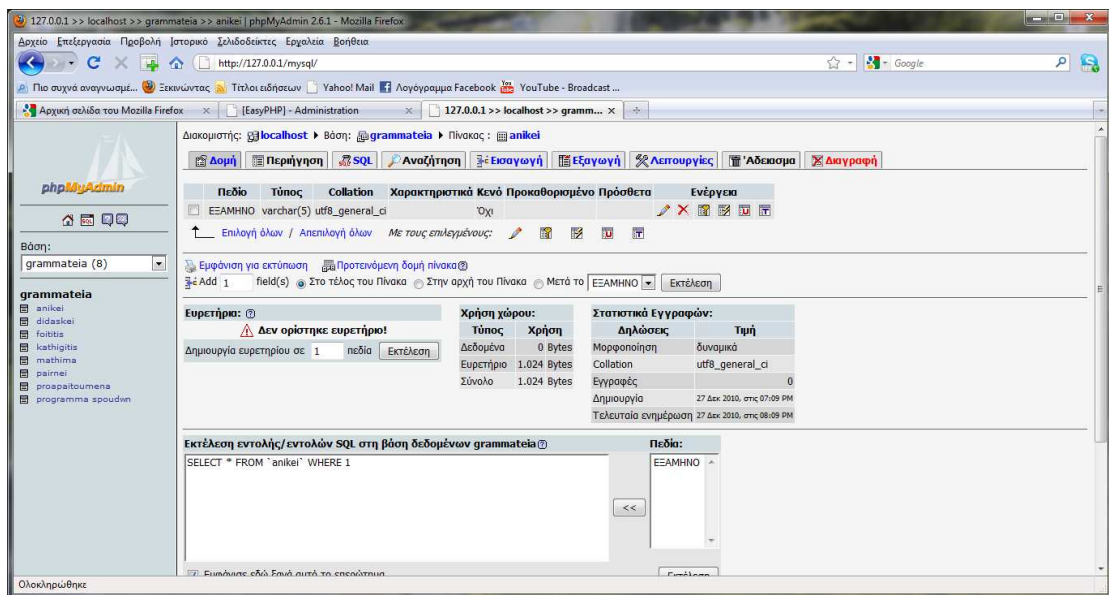
Επιπλέον βλέπουμε τις εκάστοτε εγγραφές στον πίνακα μας, οι οποίες εμφανίζονται με όλα τα γνωρίσματα τους είτε είναι κενά, είτε έχουν κάποια τιμή. Τις εγγραφές αυτές μπορούμε να τις τροποποιήσουμε, να αλλάξουμε δηλαδή τις τιμές τους ή ακόμα και να τις διαγράψουμε μία μία ή όλες μαζί.



4.3 Ανάλυση Πινάκων Βάσης Δεδομένων

4.3.1 Πίνακας Ανήκει

Σε αυτό το κομμάτι θα δείξουμε το γραφικό κομμάτι της βάσης μας, δηλαδή θα δείξουμε κάποιες φωτογραφίες από το γραφικό μέρος της εφαρμογής μας, η οποία υλοποιήθηκε με το πρόγραμμα EasyPHP 1.8 το οποίο και αναφέραμε στο κεφάλαιο 3. Οι πίνακες οι οποίοι δημιουργήθηκαν στο πρόγραμμα, φτιάχτηκαν βάσει του σχεδιαγράμματος του σχεσιακού μοντέλου δεδομένων το οποίο προήλθε από το διάγραμμα οντοτήτων-συσχετίσεων και οποίο αναφέραμε προηγουμένως.

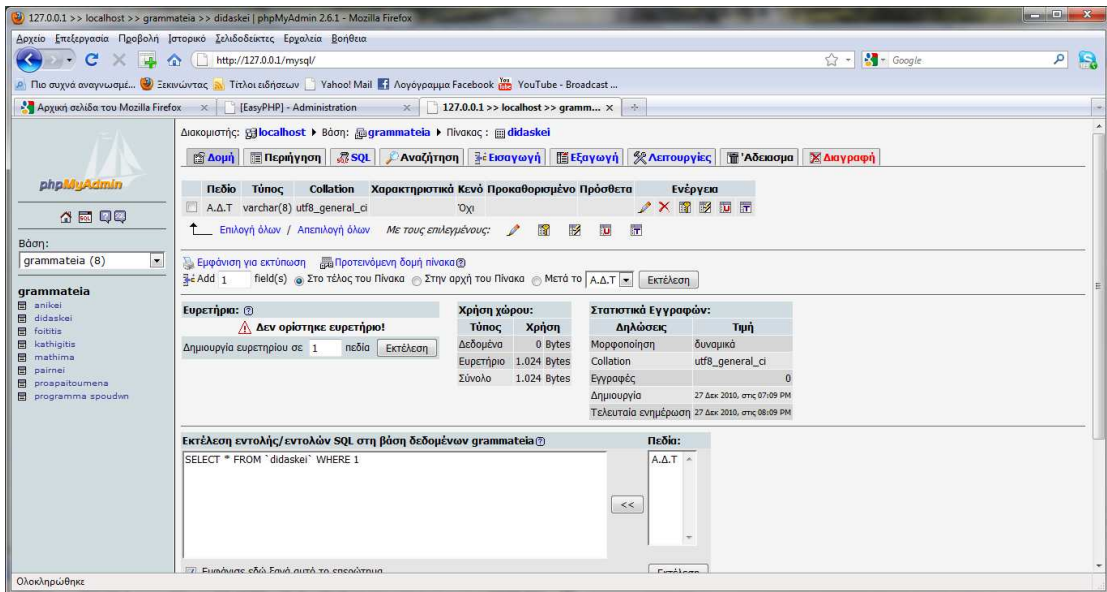


Εικόνα 11. Δομή του Πίνακα "Ανήκει"

Στην παραπάνω εικόνα βλέπουμε την δομή του πίνακα "Ανήκει", ο οποίος υπάρχει στην βάση "Γραμματεία". Παρατηρούμε ότι αποτελείται από ένα γνώρισμα. Υπάρχει επίσης και ένα πεδίο όπου μπορούμε να εκτελέσουμε ερωτήματα SQL και να πάρουμε τα επιθυμητά αποτελέσματα.

4.3.2 Πίνακας Διδάσκει

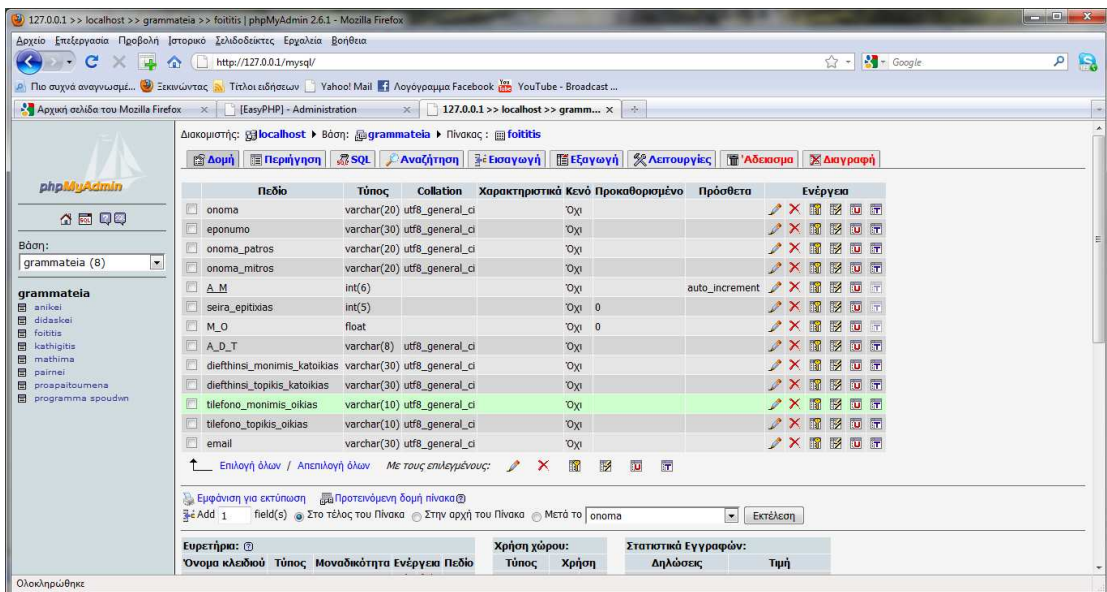
Στην επόμενη εικόνα παρακάτω βλέπουμε τη δομή του πίνακα "Διδάσκει". Αυτός ο πίνακας αποτελείται από το ένα γνώρισμα με τις καθορισμένες ρυθμίσεις που του έχουμε βάλει κατά την δημιουργία του.



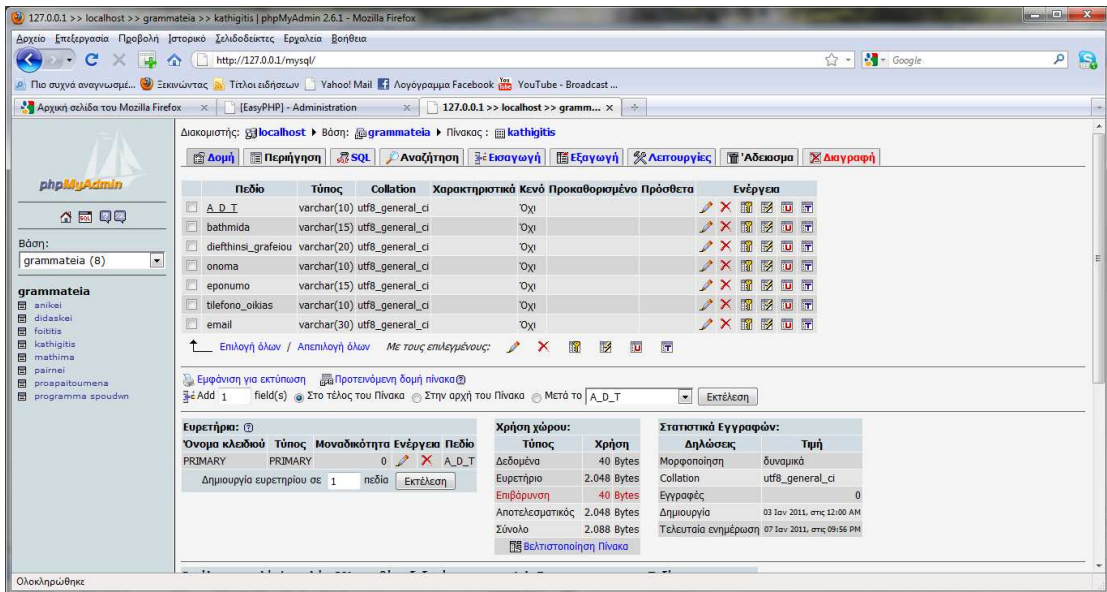
Εικόνα 12.Δομή του Πίνακα "Διδάσκει"

4.3.3 Πίνακας Φοιτητής

Στην επόμενη εικόνα έχουμε την δομή του πίνακα "Φοιτητής", η οποία έχει τα γνωρίσματα της με τις σχετικές τους ρυθμίσεις και το απαραίτητο πρωτεύων κλειδί. Βλέπουμε ότι έχουμε θέσει το γνώρισμα "AM" στη ρύθμιση αυτόματης αύξησης, δηλαδή κάθε φορά που εισάγεται μία τιμή στον πίνακα μας, αυτόματα αυξάνεται η τιμή του γνωρίσματος μας. Επίσης το γνώρισμα αυτό έχει τεθεί ως πρωτεύων κλειδί. Τις εγγραφές του πίνακα τις παραθέσαμε σε προηγούμενη φωτογραφία.



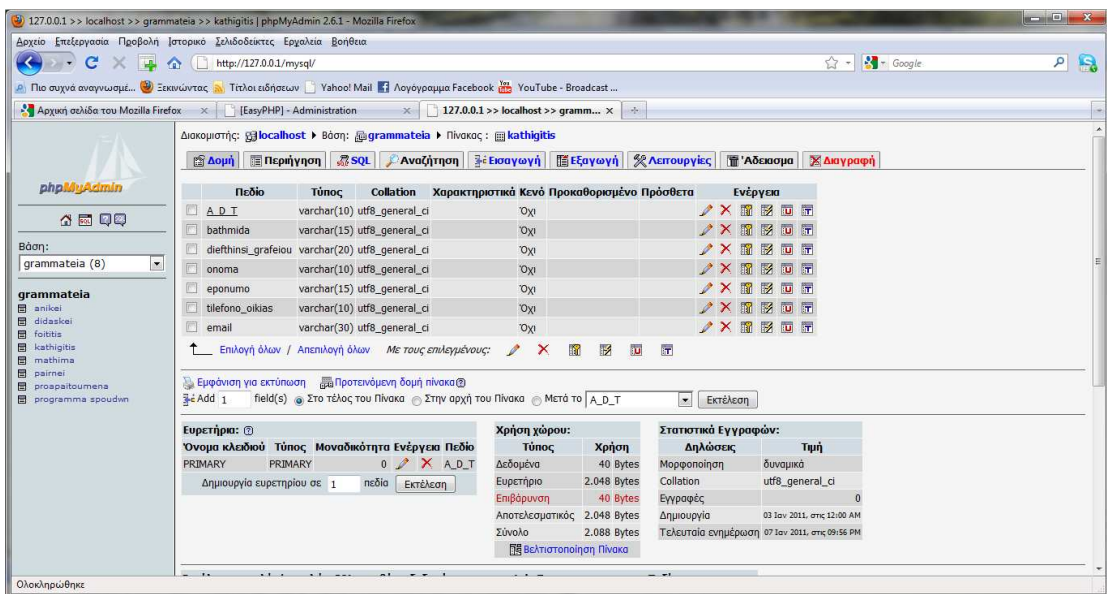
Εικόνα 13.Δομή του Πίνακα "Φοιτητής"



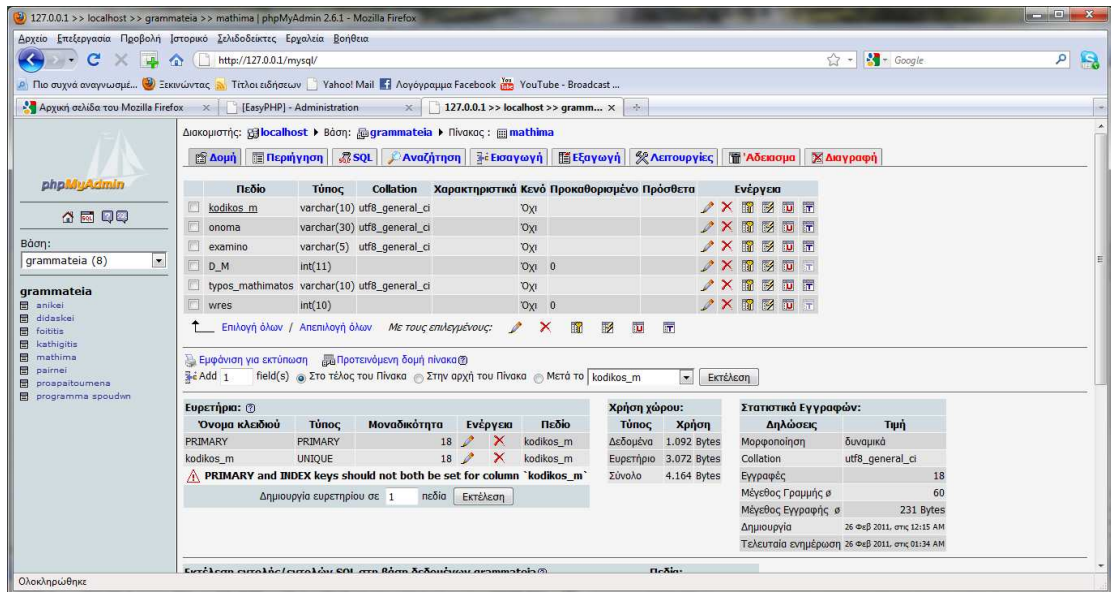
Εικόνα 14. Δομή του Πίνακα "Καθηγητής"

4.3.4 Πίνακας Καθηγητής

Η παρακάτω εικόνα μας δείχνει την δομή του πίνακα "Καθηγητής". Εδώ έχουμε θέσει ως πρωτεύων κλειδί τον αριθμό ταυτότητας. Παρατηρούμε και εδώ πως εμφανίζονται τα γνωρίσματα μας με τις ρυθμίσεις τους. Μπορούμε πάντα να εκτελέσουμε οποιαδήποτε ενέργεια θέλουμε στον πίνακα μας, να επέμβουμε δηλαδή και να αλλάξουμε τις ρυθμίσεις



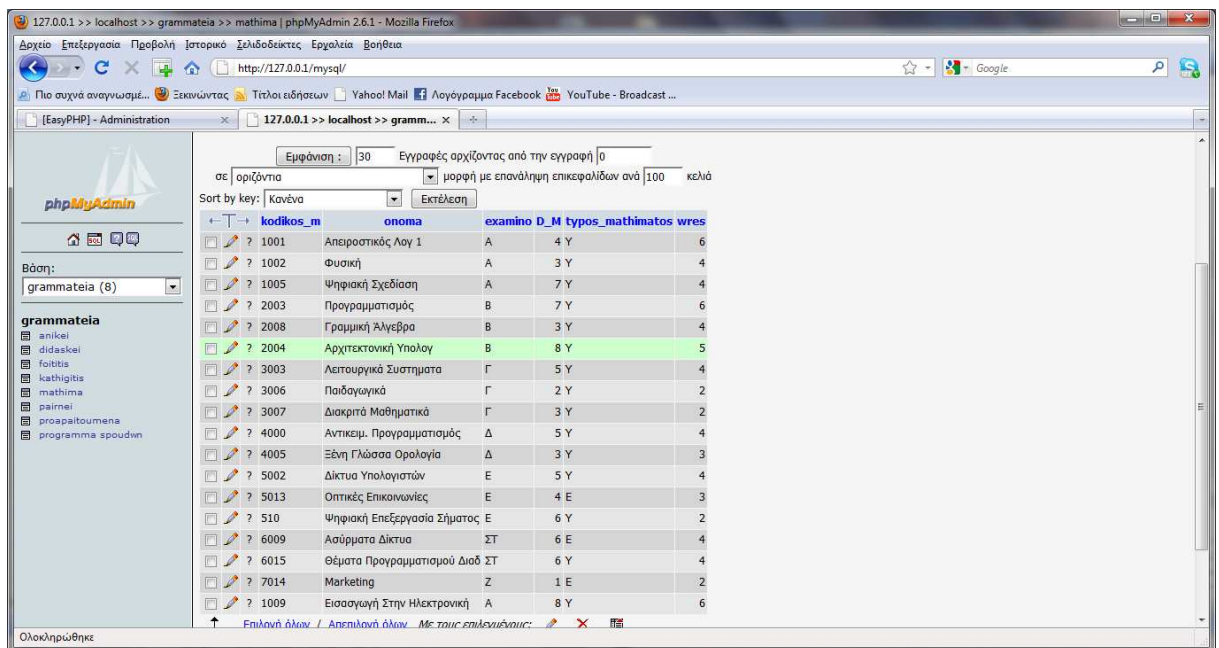
Εικόνα 15. Δομή του Πίνακα "Καθηγητής"



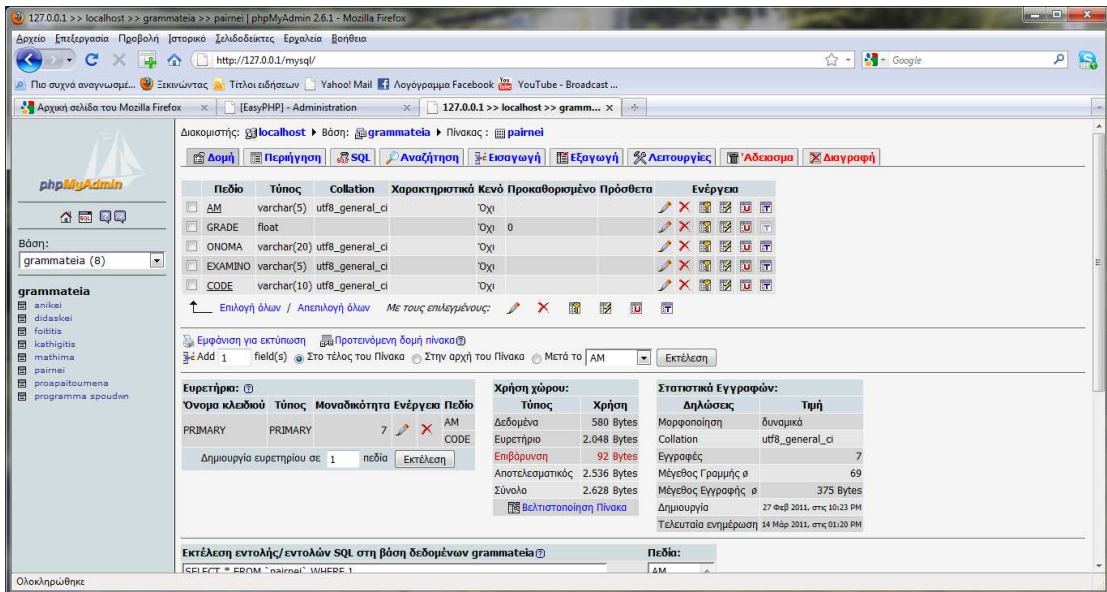
Εικόνα 16. Δομή του Πίνακα "Μάθημα"

4.3.5 Πίνακας Μάθημα

Στον παραπάνω πίνακα έχουμε ως πρωτεύων κλειδί τον κωδικό μαθήματος. Βλέπουμε επίσης ότι έχει οριστεί ως ευρετήριο το πρωτεύων κλειδί του πίνακα μας. Στο ευρετήριο βλέπουμε ότι έχουμε 18 εγγραφές στον πίνακα μας τις οποίες και βλέπουμε παρακάτω στην επόμενη εικόνα. Αυτές οι εγγραφές χρησιμοποιούνται όπως και θα δείξουμε παρακάτω στην εμφάνιση του προγράμματος σπουδών στην εφαρμογή μας.



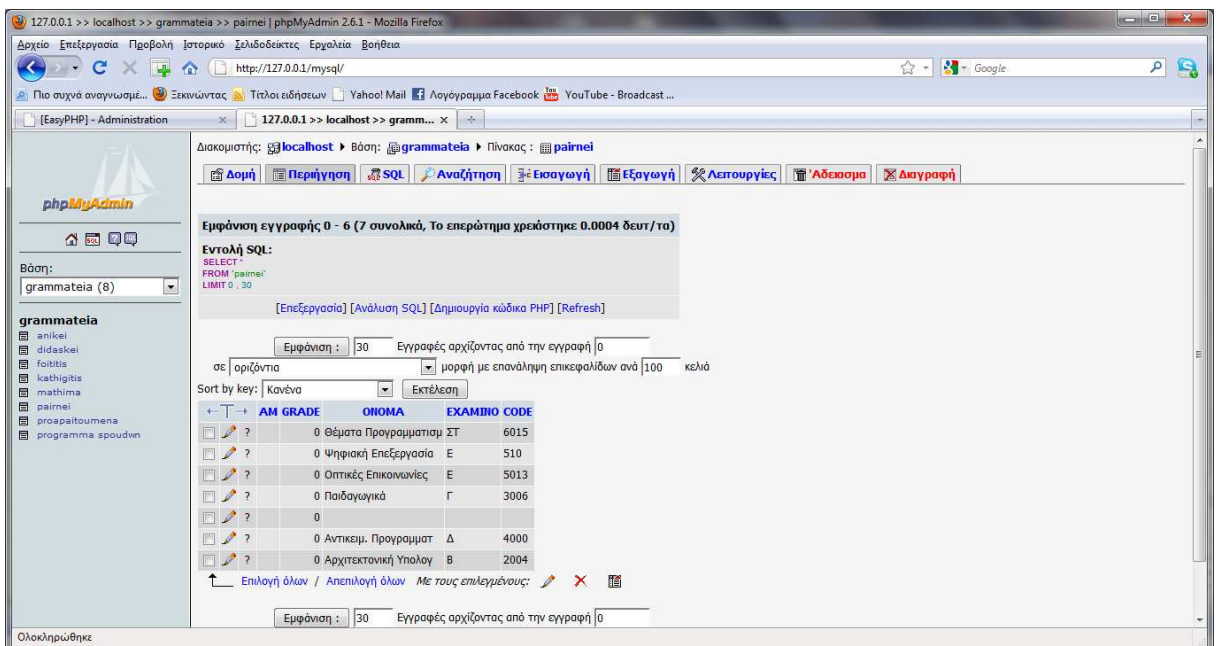
Εικόνα 17. Οι εγγραφές του Πίνακα "Μάθημα"



Εικόνα 18.Δομή του Πίνακα "Ψάρνει"

4.3.6 Πίνακας Ψάρνει

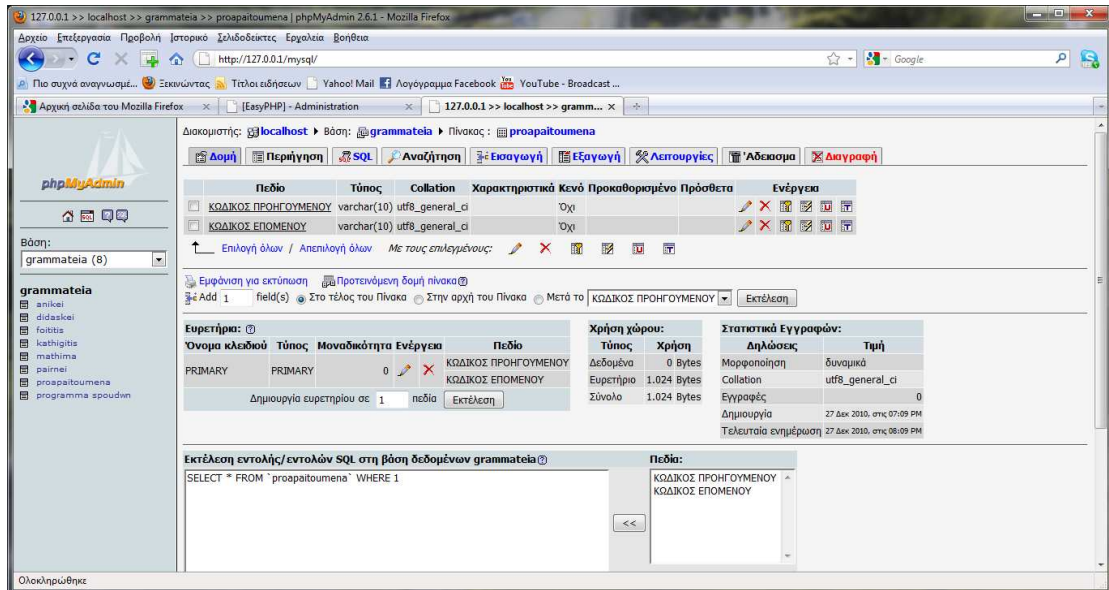
Στον πίνακα αυτόν, ο οποίος είναι συσχετίση στο διάγραμμα οντοτήτων-συσχετίσεων, έχουμε ως κλειδιά το AM το οποίο προέρχεται από την οντότητα-πίνακα φοιτητής. Αυτό το κάνουμε για να πετύχουμε την πολλαπλότητα στη συσχετίση η οποία είναι 1-N, δηλαδή ένα προς πολλά. Επίσης το άλλο κλειδί που έχουμε βάλει είναι ο κωδικός, ο οποίος προέρχεται από τον πίνακα μάθημα και χρησιμοποιείται για τον ίδιο σκοπό όπως περιγράψαμε προηγουμένως. Στην επόμενη εικόνα παραθέτουμε τα δεδομένα τα οποία είναι εισηγμένα στον πίνακα μας.



Εικόνα 19.Δεδομένα του Πίνακα "Ψάρνει"

4.3.7 Πίνακας Προαπαιτούμενα

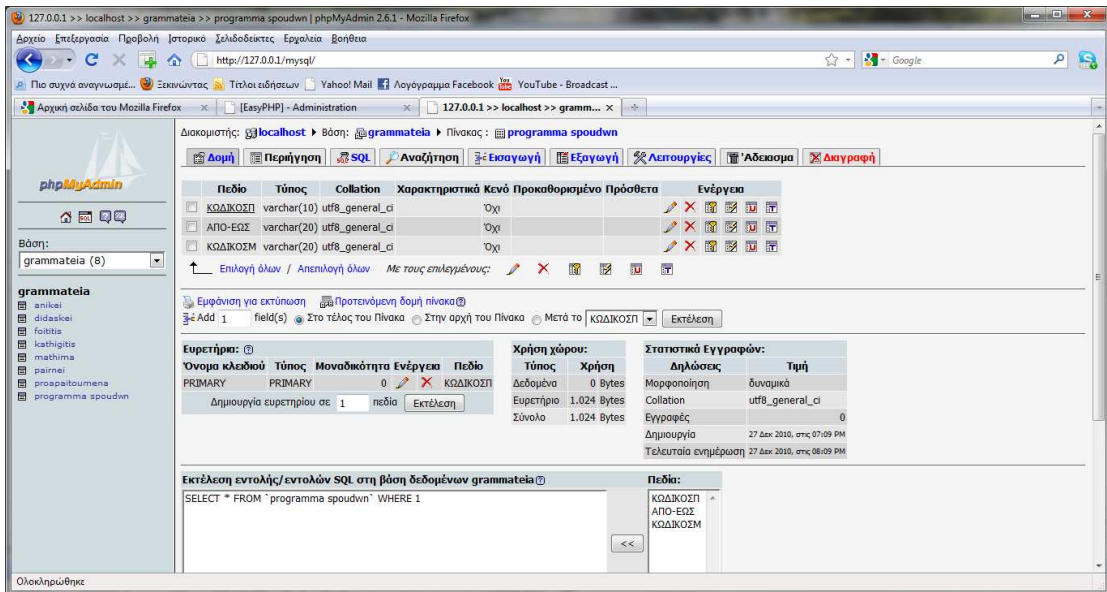
Στην επόμενη εικόνα έχουμε τον πίνακα προαπαιτούμενα ο οποίος έχει δύο κλειδιά και είναι αναδρομικός. Χρησιμοποιείται δηλαδή για τον έλεγχο κάποιου μαθήματος αν έχει προαπαιτούμενα ή όχι για να προχωρήσει η εγγραφή του εκάστοτε φοιτητή. Τα κλειδιά που έχει είναι ο κωδικός προηγούμενου και ο κωδικός επόμενου μαθήματος.



Εικόνα 20. Δομή του Πίνακα Προαπαιτούμενα

4.3.8 Πίνακας Πρόγραμμα Σπουδών

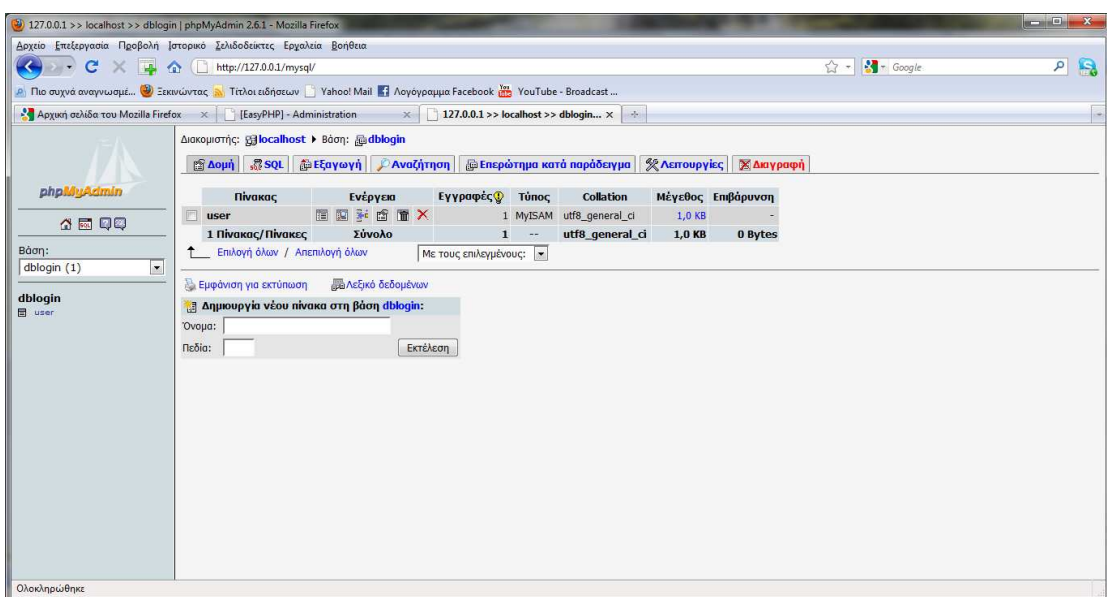
Τέλος έχουμε τον πίνακα Πρόγραμμα Σπουδών ο οποίος έχει ως πρωτεύον κλειδί τον κωδικό του προγράμματος σπουδών. Σε αυτόν τον πίνακα περνάμε τα μαθήματα που έχει το εκάστοτε πρόγραμμα σπουδών και τα οποία μπορεί να ανήκουν σε κάποιο παλαιότερο πρόγραμμα σπουδών ή να ανήκουν σε κάποιο τωρινό πρόγραμμα σπουδών.



Εικόνα 21.Δομή του Πίνακα "Πρόγραμμα Σπουδών"

4.3.9 Εισαγωγή Κωδικών

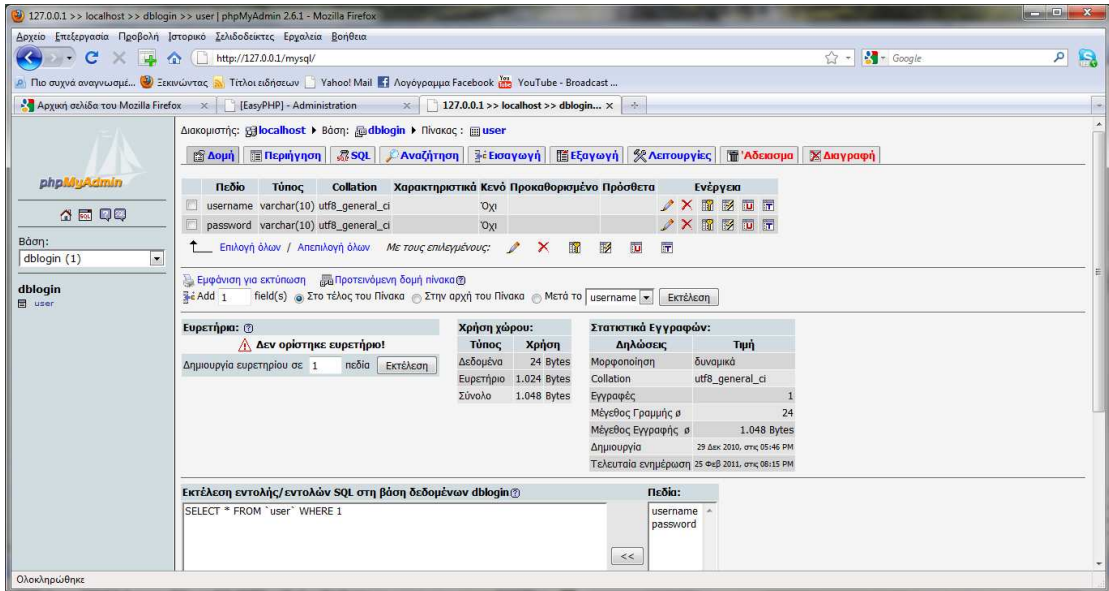
Για να έχουμε όμως ασφάλεια στην εφαρμογή μας θα πρέπει να υπάρχει κάποιο πεδίο όπου θα εισάγεται κάποιος κωδικός, ο οποίος θα μας επιτρέπει την ασφαλή είσοδο και την ακεραιότητα των δεδομένων καθώς και την ασφαλή διαχείριση τους. Η διαχείριση και η ταυτοποίηση του κωδικού αυτού γίνεται με την χρήση μιας άλλης βάσης δεδομένων. Αυτό το κάνουμε για μεγαλύτερη ασφάλεια και γι'αυτό επιλέγουμε να δημιουργήσουμε μία ξεχωριστή βάση δεδομένων με όνομα dblogin. Στην επόμενη εικόνα βλέπουμε το γραφικό περιβάλλον της βάσης μας.



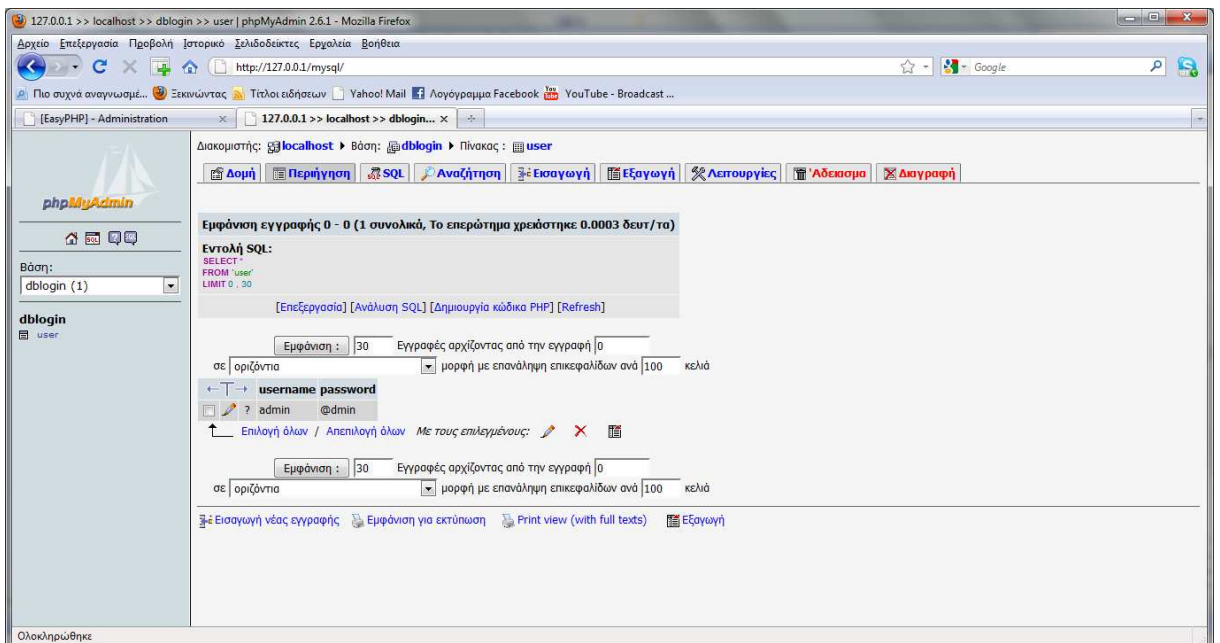
Εικόνα 22.Η Βάση DbLogin

4.3.10 Ο Πίνακας User

Η βάση μας περιέχει ένα πίνακα ο οποίος ονομάζεται user και του οποίου πίνακα βλέπουμε στιγμιότυπο παρακάτω και στην επόμενη εικόνα βλέπουμε τις τιμές που έχουμε δώσει για password και username.



Εικόνα 23. Η Δομή του Πίνακα "User"



Εικόνα 24. Τα στοιχεία του Πίνακα User

Ο πίνακας αυτός έχει δύο πεδία ,ένα για το όνομα χρήστη και ένα για τον κωδικό. Μπορούμε να βάλουμε ότι κωδικό και όνομα χρήστη θέλουμε και επιπλέον για μεγαλύτερη

ασφάλεια θέτουμε την επιλογή Md5 στο On , η οποία κωδικοποιεί το password που βάζουμε για μεγαλύτερη ασφάλεια και το εμφανίζει με κωδικοποιημένους χαρακτήρες.

Σε επόμενη ενότητα (παράρτημα) θα δείξουμε και τον παραγόμενο κώδικα SQL για κάθε ένα από αυτούς τους πίνακες.

4.4 Η Εφαρμογή «Γραμματεία Πανεπιστημίου»

4.4.1 Σελίδα Εισόδου

Σε αυτή την ενότητα θα παραθέσουμε φωτογραφίες-στιγμιότυπα από την εφαρμογή μας, η οποία βρίσκεται εφαρμογή σε γραμματείες πανεπιστημίων και τεχνολογικών ιδρυμάτων .Η εφαρμογή αυτή λειτουργεί από την μεριά του διαχειριστή, δηλαδή από την μεριά του υπαλλήλου μιας γραμματείας και η δουλειά της είναι να αποθηκεύει δεδομένα και στοιχεία για τους νεοεγγραφείς φοιτητές αλλά και για τους παλαιότερους φοιτητές, να αποθηκεύει και να διαγράφει στοιχεία κάποιου μαθήματος καθώς και να αποθηκεύει ή να διαγράφει στοιχεία ενός καθηγητή του τμήματος. Παρέχει επίσης ασφάλεια καθώς πρέπει να εισαχθεί όνομα χρήστη και κωδικός εισόδου αρχικά για να εισέλθει κάποιος στην εφαρμογή.



On-Line Σύστημα Διαχείρισης ΕΥ ΓΤΘΕ-Γραμματεία

Είσοδος Διαχείρισης	
Όνομα Χρήστη:	<input type="text"/>
Κωδικός Πρόσβασης:	<input type="text"/>
<input type="button" value="Είσοδος"/>	<input type="button" value="Επαναφορά"/>

Ολοκληρώθηκε

Εικόνα 25.Σελίδα Κωδικού Εισόδου Χρήστη

Αρχικά τρέχουμε το πρόγραμμα EasyPHP 1.8 και κατόπιν πηγαίνουμε στην ip 127.0.0.1 και έπειτα γράφουμε δίπλα της μετά από το χαρακτήρα / το όνομα του αρχείου που θέλουμε και πατάμε enter.Στην προκειμένη περίπτωση βάζουμε το όνομα του αρχείου μας το οποίο είναι selida_diaxirisis.php.Τα αρχεία php τα οποία τρέχουμε στη συγκεκριμένη εφαρμογή τα τοποθετούμε στον φάκελο www που υπάρχει στα προγράμματα εκεί που εγκαταστάθηκε το EasyPHP.

Στην εικόνα 25 παραπάνω βλέπουμε πάνω στην μέση της σελίδας μας ένα ψηφιακό ρολόι το οποίο έχει υλοποιηθεί με την χρήση της γλώσσας JavaScript .Κατόπιν έχουμε την επικεφαλίδα μας και έπειτα την φόρμα υποβολής username και password.Αφού εισάγουμε τα απαραίτητα στοιχεία πατούμε το κουμπί είσοδος για να γίνει ο έλεγχος των στοιχείων και αν είναι σωστά να εισέλθουμε στην εφαρμογή μας. Αν πατήσουμε το κουμπί επαναφορά τότε καθαρίζουν και τα δύο πεδία και φαίνονται όπως τώρα.

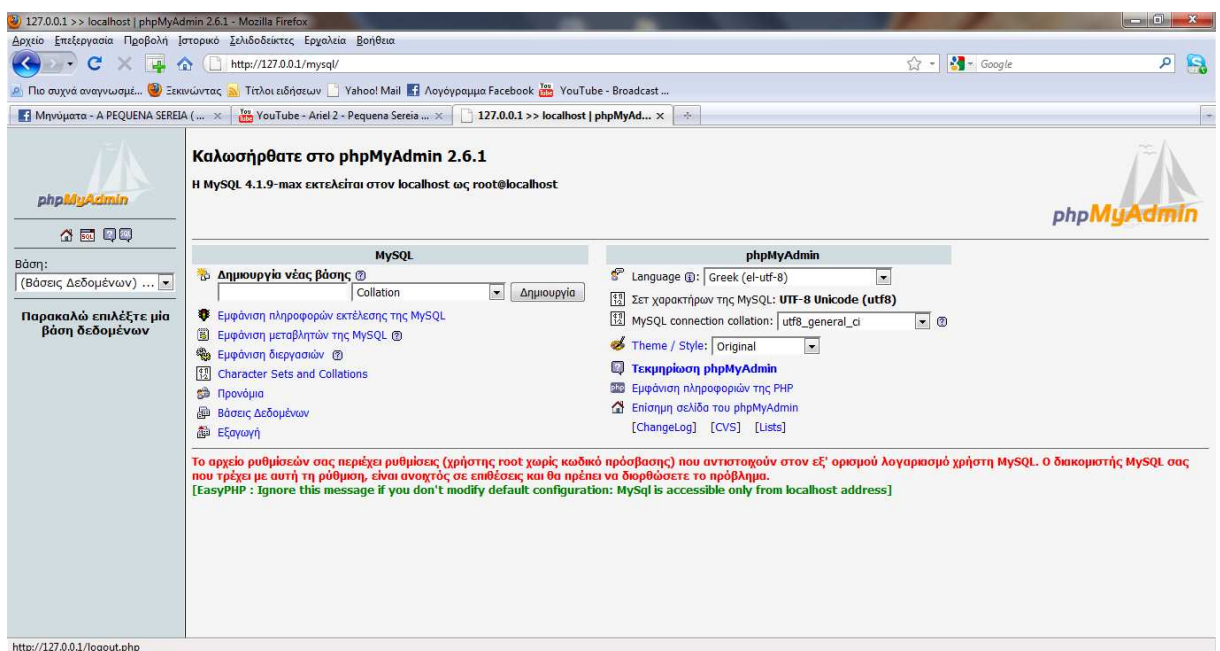
4.4.2 Σελίδα Διαχείρισης



Εικόνα 26 Αρχική Σελίδα Διαχείρισης

Σε αυτή τη σελίδα έχουμε όπως πάντα το ψηφιακό ρολόι ,τον τίτλο της εκάστοτε σελίδας και από κάτω υπάρχουν κάποια κουμπιά τα οποία εκτελούν κάποιες ενέργειες το καθένα ξεχωριστά. Την ενέργεια που εκτελεί το κάθε κουμπί την αναγράφει. Από κάτω από τα κουμπιά έχουμε το link της εξόδου .το οποίο μας επαναφέρει στην προηγούμενη σελίδα όπου εισάγουμε των κωδικό εισόδου μας και το από κάτω link, μας μεταφέρει στην εφαρμογή του προγράμματος όπου φτιάξαμε την βάση μας την οποία βλέπουμε στην εικόνα 27.

4.4.3 Σελίδα Βάσης Δεδομένων



Εικόνα 27.Η αρχική σελίδα της Βάσης δεδομένων

4.4.4 Σελίδα Εγγραφής Φοιτητή

Προχωρώντας την ανάλυση της εφαρμογής μας θα πατήσουμε το πρώτο κουμπί το οποίο αναγράφει «Εγγραφή Φοιτητή» και φυσικά όπως καταλαβαίνουμε αφορά την αποθήκευση των αρχικών στοιχείων του εκάστοτε νεοεγγραφείς φοιτητή στο τμήμα. Στην εικόνα 28 λοιπόν παρακάτω έχουμε την φόρμα την οποία καλούμαστε να συμπληρώσουμε. Η συμπλήρωση της φόρμας δεν είναι υποχρεωτική, δηλαδή μπορούμε να εισάγουμε στοιχεία σε επιλεγμένα πεδία και όχι να τα γεμίσουμε όλα αναγκαστικά.

The screenshot shows a web browser window with the title 'ΕΓΓΡΑΦΗ ΦΟΙΤΗΤΗ - Mozilla Firefox'. The address bar contains 'http://127.0.0.1/eggrafi_foititi.php'. The page content is titled 'Σελίδα Εγγραφής Φοιτητή' and features a registration form with the following fields and values:

Εγγραφή Φοιτητή	
Όνομα :	Βίκτωρος
Επίθετο :	Παντελής
Όνομα Πατρός :	Γεώργιος
Όνομα Μητρός :	Κυριακούλα
Σειρά Επιτυχίας :	τριτος
M.O. :	15
A.Δ.Τ. :	
Διεύθυνση μόν. κατοικίας :	
Διεύθυνση προσ. κατοικίας :	
Τηλέφωνο Μόνιμης Οικίας :	
Τηλέφωνο Προσωρινής Οικίας :	
Email :	

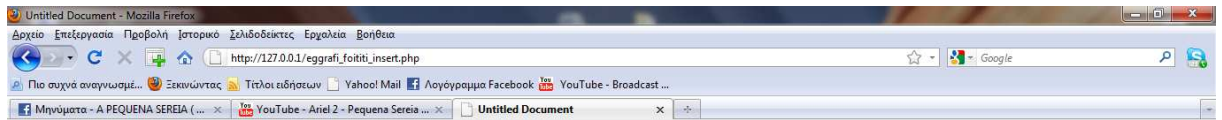
Below the form, there are two buttons: 'Εγγραφή' and 'Επιστροφή'. A red error message 'Input must be numeric' is displayed next to the M.O. field. At the bottom of the page, there is a link: 'Επιστροφή Στην Σελίδα Διαχείρισης'.

Εικόνα 28.Φόρμα Εγγραφής Φοιτητή

Παρατηρούμε ότι στην φόρμα μας στην εικόνα 28, τα πεδία όπου πρέπει να συμπληρωθούν με κάποιο νούμερο και όχι με γράμματα ή με γράμματα και νούμερα ανάμεικτα ,όπως πχ το πεδίο του Α.Δ.Τ ,δηλαδή του αριθμού δελτίου ταυτότητας, το οποίο περιέχει και γράμμα και αριθμούς, δεν αφήνουν τον χρήστη να εισάγει τίποτε άλλο παρά μόνο αριθμούς. Αν δεν βάλουμε κάποιο αριθμό, δεν μπορούμε να συνεχίσουμε την συμπλήρωση ή την υποβολή και αποθήκευση της φόρμας μας.

Αυτό το κάνουμε για να μην γίνονται τυχόν λάθη κατά την εισαγωγή των στοιχείων καθώς σε μία γραμματεία ο όγκος των στοιχείων και των εγγραφών είναι τέτοιος που δεν επιτρέπει να γίνονται πολλοί επανέλεγχοι για τυχόν λάθη ,κάτι το οποίο θα ήταν υπερβολικά χρονοβόρο και πολύ δύσκολο.

Αφού λοιπόν συμπληρώσουμε τα πεδία της επιλογής μας πατάμε το κουμπί της εγγραφής το οποίο και βλέπουμε στο τέλος της φόρμας για να αποσταλούν τα δεδομένα στην βάση μας και να αποθηκευτούν. Παρακάτω θα παραθέσουμε στιγμιότυπο με την σελίδα που εμφανίζεται μετά που θα πατήσουμε το κουμπί της εγγραφής. Δίπλα από αυτό το κουμπί, υπάρχει ένα άλλο το οποίο αν το πατήσουμε τότε αυτόματα η φόρμα μας επανέρχεται στην αρχική της μορφή, δηλαδή με όλα τα πεδία της κενά.

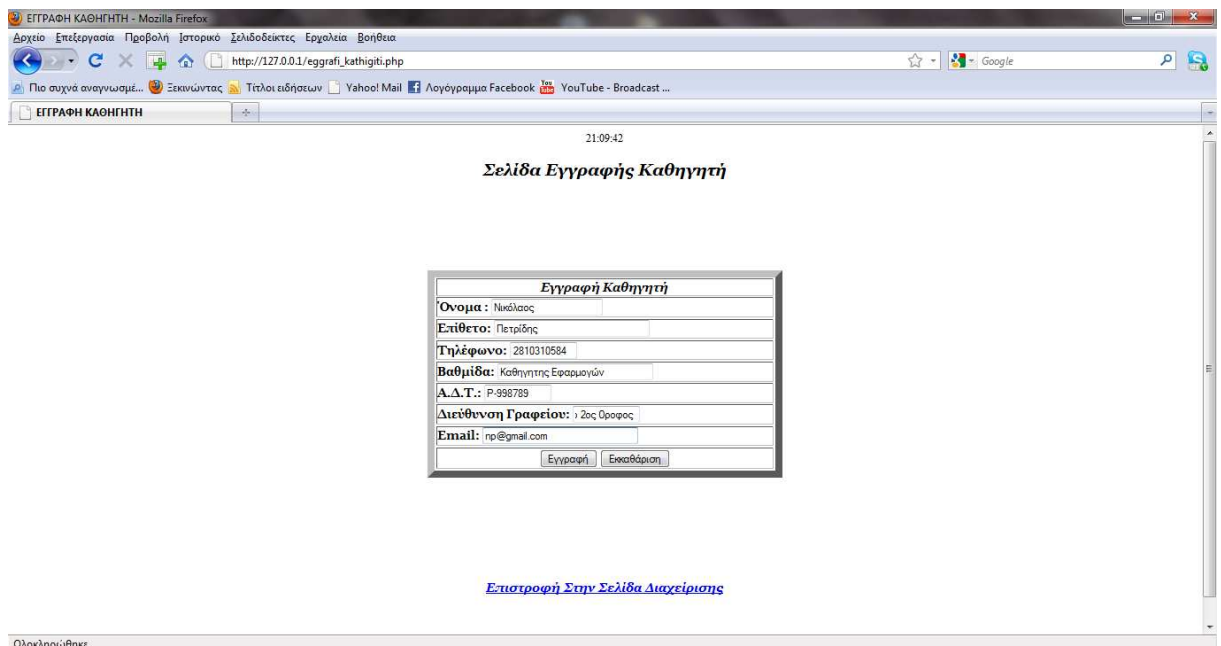


Ολοκληρώθηκε

Εικόνα 29 Σελίδα Επιβεβαίωσης Υποβολής Στοιχείων Εγγραφής

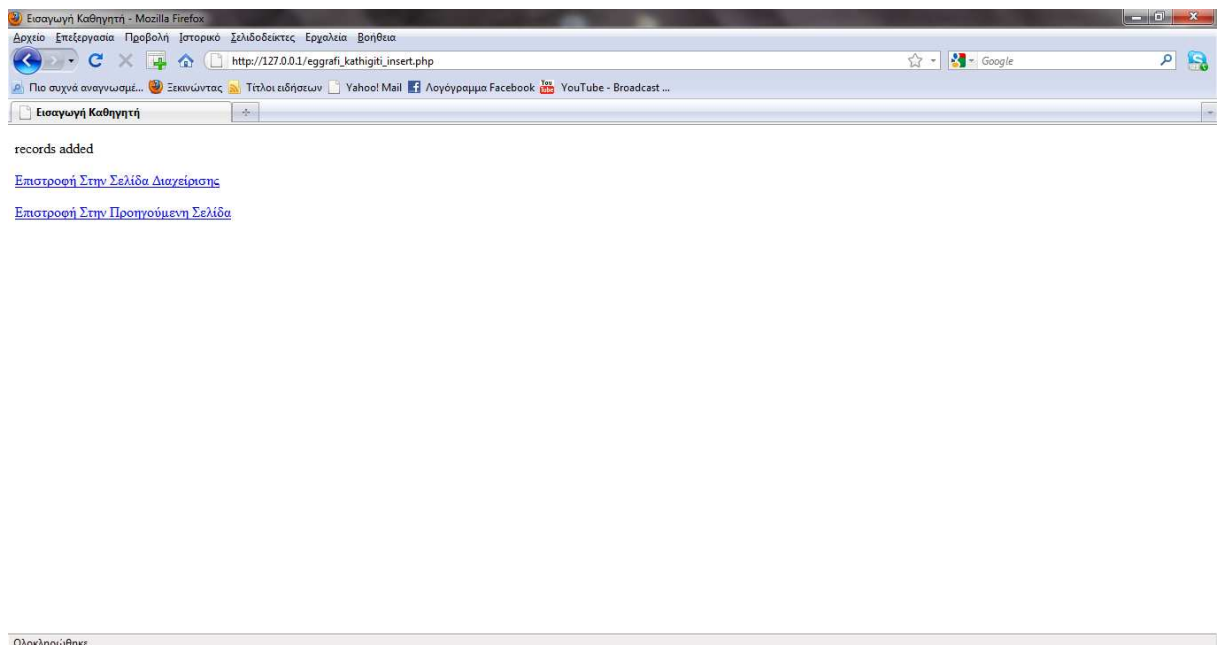
Στην φόρμα εγγραφής μας υπάρχει επίσης ένα link κάτω από την φόρμα, το οποίο μας παραπέμπει στο να επιστρέψουμε στην αρχική σελίδα διαχείρισης την οποία περιγράψαμε προηγουμένως .Στην εικόνα 29 όπως είπαμε και προηγουμένως βλέπουμε ότι μόλις πατήσαμε την εγγραφή, μας εμφάνισε το μήνυμα ότι η εγγραφή ήταν επιτυχής και από κάτω από το μήνυμα αυτό έχουμε δυο links, ένα το οποίο μας γυρίζει στην αρχική σελίδα διαχείρισης και ένα το οποίο μας γυρίζει στην προηγούμενη σελίδα .Όπως θα παρατηρήσουμε, η εικόνα 29 αποτελεί ξεχωριστό αρχείο php ,δηλαδή με το πάτημα του κουμπιού στη σελίδα εγγραφής πήγαμε αυτόματα σε έν άλλο αρχείο ονόματι eggrafi_foititi_insert το οποίο και είναι αρμόδιο για τον έλεγχο και την εισαγωγή των στοιχείων στη βάση. Τον κώδικα php και mysql θα τον αναλύσουμε στο παράρτημα όπου και θα δείξουμε όλο τον κώδικα της εφαρμογής.

4.4.5 Σελίδα Εγγραφής Καθηγητή



Εικόνα 30. Η Σελίδα Εγγραφής Στοιχείων Καθηγητή

Στην εικόνα 30 έχουμε τη σελίδα εγγραφής στοιχείων ενός καθηγητή. Τα πεδία που συμπληρώνουμε σε αυτή τη φόρμα είναι αυτά που χρειάζονται να γνωρίζει μία γραμματεία για να μπορεί να επικοινωνήσει με τον εκάστοτε καθηγητή καθώς και να μπορεί να επικοινωνήσει κάποιος φοιτητής. Στην αρχή της σελίδας υπάρχει το καθορισμένο ψηφιακό ρολόι και στο τέλος της φόρμας τα απαραίτητα κουμπιά για εγγραφή ή εκκαθάριση της φόρμας. Το αρχείο αυτό δεν έχει έλεγχο για αριθμητικά δεδομένα καθώς όπως παρατηρούμε δεν χρειάζεται.



Εικόνα 31. Σελίδα Επιβεβαίωσης Υποβολής Στοιχείων Εγγραφής Καθηγητή

Το όνομα του αρχείου που εκτελείται στην εικόνα 30 είναι eggrafi_kathigiti.php και εκτελώντας το , δηλαδή να πατήσουμε το κουμπί της εγγραφής μας πηγαίνει σε μία άλλη σελίδα την οποία βλέπουμε στην εικόνα 31 από πάνω .Αυτό το αρχείο έχει όνομα eggrafi_kathigiti_insert.php και κάνει την εισαγωγή των στοιχείων στη βάση μας. Μας επιστρέφει ένα μήνυμα επιτυχίας ή αποτυχίας αν δεν μπορεί να συνδεθεί με την βάση μας, για οποιοδήποτε λόγο και κάτω από το μήνυμα τα δύο κλασσικά Links που αναφέραμε και στην εγγραφή φοιτητή.

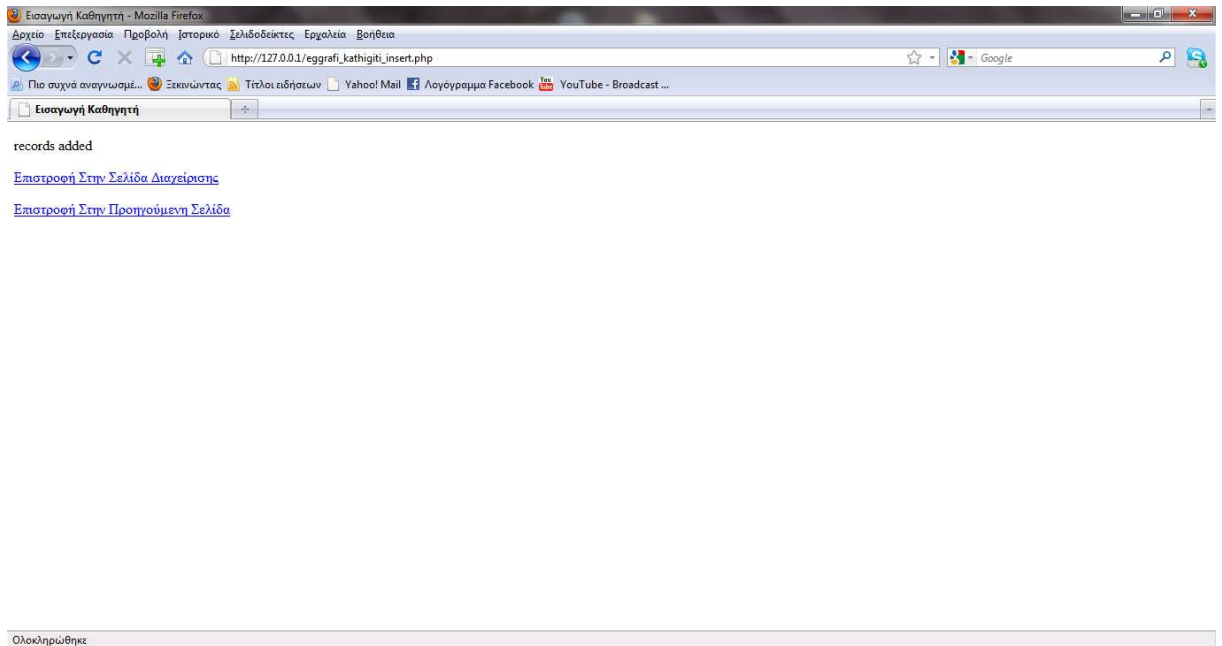
4.4.6 Σελίδα Εισαγωγής Μαθήματος

Εισαγωγή Μαθήματος	
ID :	15556
Τίτλος :	Μαθηματικά
Ώρες :	εξί
<i>Input must be numeric</i>	
Εξάμηνο :	Α
Τύπος :	Υ
Δ.Μ. :	4
<input type="button" value="Εισαγωγή"/> <input type="button" value="Εισαθάρηση"/>	

[Επιστροφή Στην Σελίδα Διαχείρισης](#)

Εικόνα 32. Σελίδα Εισαγωγής Στοιχείων Μαθήματος

Συνεχίζοντας την ανάλυση της εφαρμογής μας ,σειρά έχει το κουμπί εγγραφής μαθήματος. Το κουμπί αυτό τρέχει ένα αρχείο με όνομα eggrafi_mathimatos.php το οποίο και μας εμφανίζει τη σελίδα που βλέπουμε στην εικόνα 32.Σε αυτή τη φόρμα έχουμε βάλει τον περιορισμό της συμπλήρωσης ορισμένων πεδίων μόνο με νούμερα καθότι επιβάλλεται. Αφού συμπληρώσουμε τα πεδία μας και πατήσουμε το κουμπί της εγγραφής θα εμφανιστεί η κλασσική σελίδα υποβολής και αποθήκευσης των στοιχείων μας που βλέπουμε παρακάτω (εικόνα 33).



Εικόνα 33.Σελίδα Υποβολής Στοιχείων Εγγραφής Μαθήματος

Ολοκληρώσαμε λοιπόν το κομμάτι της εισαγωγής στοιχείων-δεδομένων στην βάση δεδομένων μας με τις τρεις φόρμες που δείξαμε προηγουμένως. Παρακάτω θα αναλύσουμε τα κουμπιά και τις φόρμες με τις οποίες θα μπορούμε να διαγράψουμε τα στοιχεία που εισαγάγαμε προηγουμένως.

4.4.7 Σελίδα Διαγραφής Φοιτητή

Στην εικόνα 34 παρακάτω έχουμε την φόρμα με την συμπλήρωση της οποίας θα μπορούμε να διαγράψουμε τα στοιχεία κάποιου εγγεγραμμένου φοιτητή στην βάση δεδομένων της γραμματείας. Για να διαγράψουμε λοιπόν τα στοιχεία του φοιτητή που θέλουμε, δεν χρειάζεται να έχουμε μία φόρμα στο μέγεθος αυτής που εισάγαμε τα στοιχεία του. Τώρα πλέον αφού έχουμε τα στοιχεία και αφού έχουμε ορίσει κάποιο κλειδί στον πίνακα εγγραφής των στοιχείων του φοιτητή, μπορούμε απλά να εισάγουμε το AM , το οποίο είναι και το κλειδί μας ,γιατί είναι μοναδικό για κάθε φοιτητή και έτσι μπορούμε να ξεχωρίσουμε κάποιον ,ακόμα και αν έχει τα ίδια στοιχεία με κάποιον άλλο φοιτητή.

Γι'αυτό λοιπόν βάζουμε τρία πεδία τα οποία πρέπει να συμπληρώσει ο χρήστης στην φόρμα μας εκ των οποίων μπορεί να συμπληρώσει τα δύο ,το AM το οποίο είναι απαραίτητο και το επώνυμο του φοιτητή. Το πεδίο του ονόματος είναι προαιρετικό. Στην εικόνα 34 συμπληρώνουμε μόνο το όνομα και πατάμε το κουμπί της διαγραφής .Εμφανίζεται όμως η σελίδα στην εικόνα 35 η οποία μας λέει ότι πρέπει να εισάγουμε τα απαραίτητα στοιχεία και ότι δεν έχει γίνει η διαγραφή. Το ίδιο θα γίνει και αν συμπληρώσουμε και το πεδίο του επωνύμου .Και αυτό γιατί δεν έχουμε συμπληρώσει το AM του φοιτητή. Όταν συμπληρωθούν τα πεδία τότε πατώντας την διαγραφή εμφανίζεται η σελίδα στην εικόνα 36 όπου ειδοποιούμαστε για την επιτυχή διαγραφή στην βάση μας και από κάτω τα σχετικά links τα οποία μας πηγαίνουν είτε στην προηγούμενη σελίδα είτε στην αρχική μας σελίδα.



21:36:10

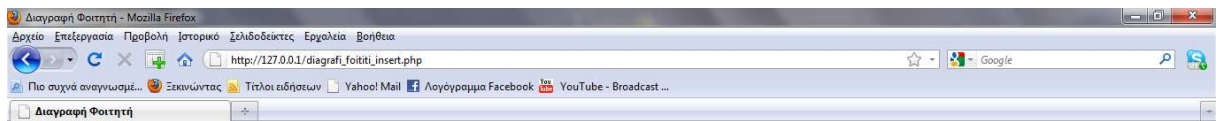
Σελίδα Διαγραφής Φοιτητή

Διαγραφή Φοιτητή	
Όνομα :	Βίκτωρας
Επίθετο :	
A.M.	
<input type="button" value="Διαγραφή"/> <input type="button" value="Εκκαθάριση"/>	

[Επιστροφή Στην Σελίδα Διαχείρισης](#)

Ολοκληρώθηκε

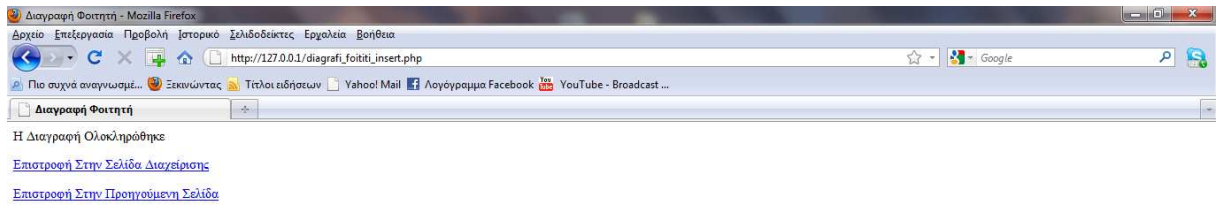
Εικόνα 34. Σελίδα Διαγραφής Στοιχείων Φοιτητή



Εισάγετε όλα τα στοιχεία

Ολοκληρώθηκε

Εικόνα 35. Έλεγχος Συμπλήρωσης Πεδίων Διαγραφής Φοιτητή

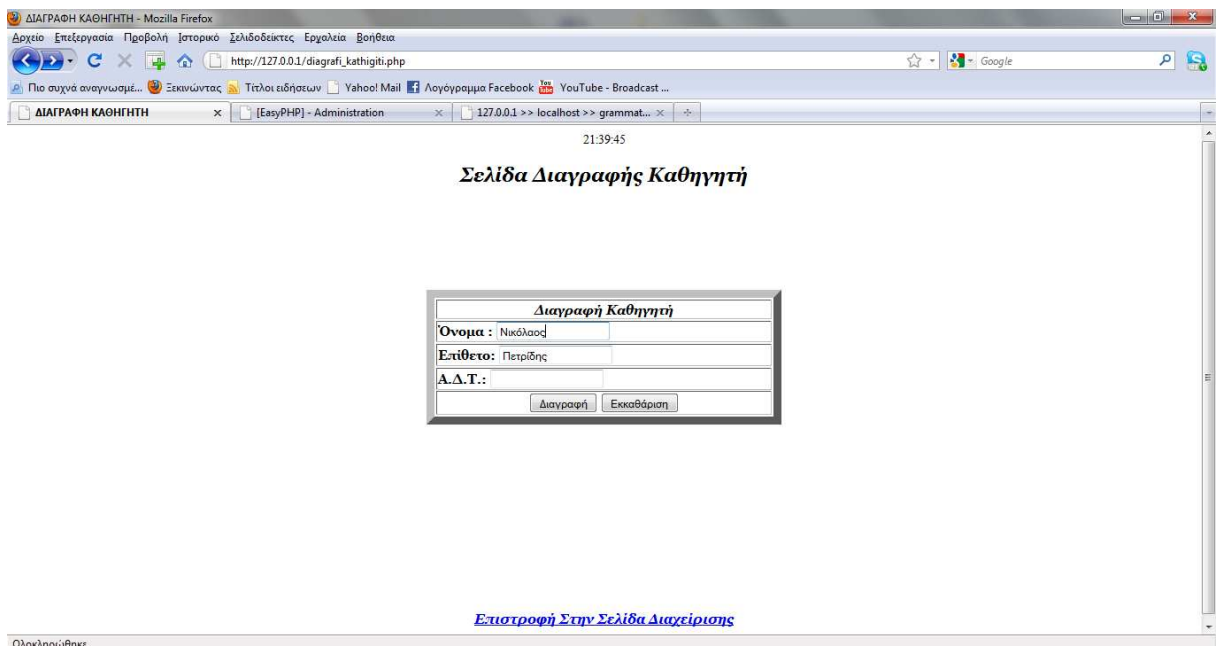


Ολοκληρώθηκε

Εικόνα 36.Σελίδα Επιτυχούς Διαγραφής Φοιτητή

4.4.8 Σελίδα Διαγραφής Καθηγητή

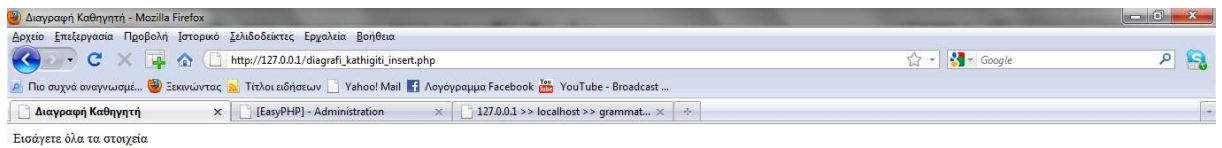
Συνεχίζοντας την ανάλυση της εφαρμογής μας στην παρακάτω εικόνα 37 δείχνουμε την φόρμα διαγραφής στοιχείων κάποιου καθηγητή. Σε αυτή τη φόρμα συναντάμε τα ίδια συστατικά και την ίδια φιλοσοφία με τις προηγούμενες φόρμες διαγραφής όπως αυτής του φοιτητή που αναλύσαμε προηγουμένως. Και σε αυτή τη φόρμα διαγραφής, πρέπει να συμπληρωθούν όλα τα στοιχεία προαιρετικά μεν αλλά πάντα απαραίτητα δε. Αν όμως για οποιοδήποτε λόγο δεν θέλουμε να συμπληρώσουμε όλα τα πεδία, αυτά που πρέπει να συμπληρώσουμε οπωσδήποτε είναι το επώνυμο και ο αριθμός ταυτότητας, ο οποίος είναι και το κλειδί στην βάση μας.



Ολοκληρώθηκε

Εικόνα 37.Φόρμα διαγραφής στοιχείων Καθηγητή

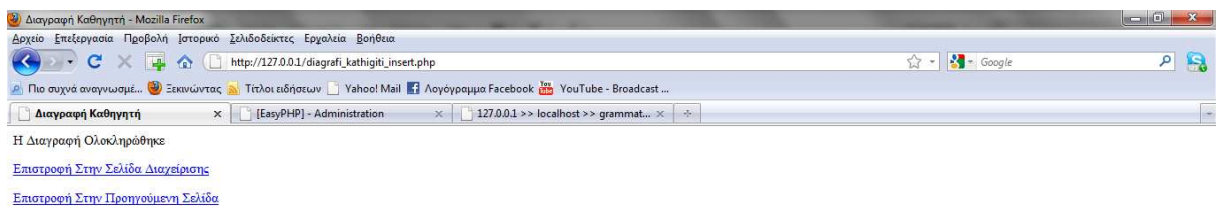
Αν δεν εισάγουμε όλα τα στοιχεία, τότε εμφανίζεται η παρακάτω σελίδα με το σχετικό μήνυμα προτροπής ,να εισάγουμε δηλαδή όλα τα απαραίτητα στοιχεία και κάνοντας το αυτό εννοούμε πάντα να συμπληρώσουμε όλα τα πεδία.



Ολοκληρώθηκε

Εικόνα 38 Μήνυμα ελέγχου ορθής εισαγωγής στοιχείων

Φυσικά αν εισάγουμε τα σωστά στοιχεία στα σωστά πεδία ,τότε εμφανίζεται η επόμενη εικόνα 39 που μας ενημερώνει ότι η διαγραφή μας εκτελέστηκε επιτυχώς από την βάση μας. Κάτω από το μήνυμα που εμφανίζεται, έχουμε και τα link τα οποία μας οδηγούν είτε ένα βήμα πίσω για περεταίρω διαγραφή κάποιου άλλου καθηγητή, είτε στην αρχική σελίδα διαχείρισης για άλλες ενέργειες, ή για έξοδο από την εφαρμογή.



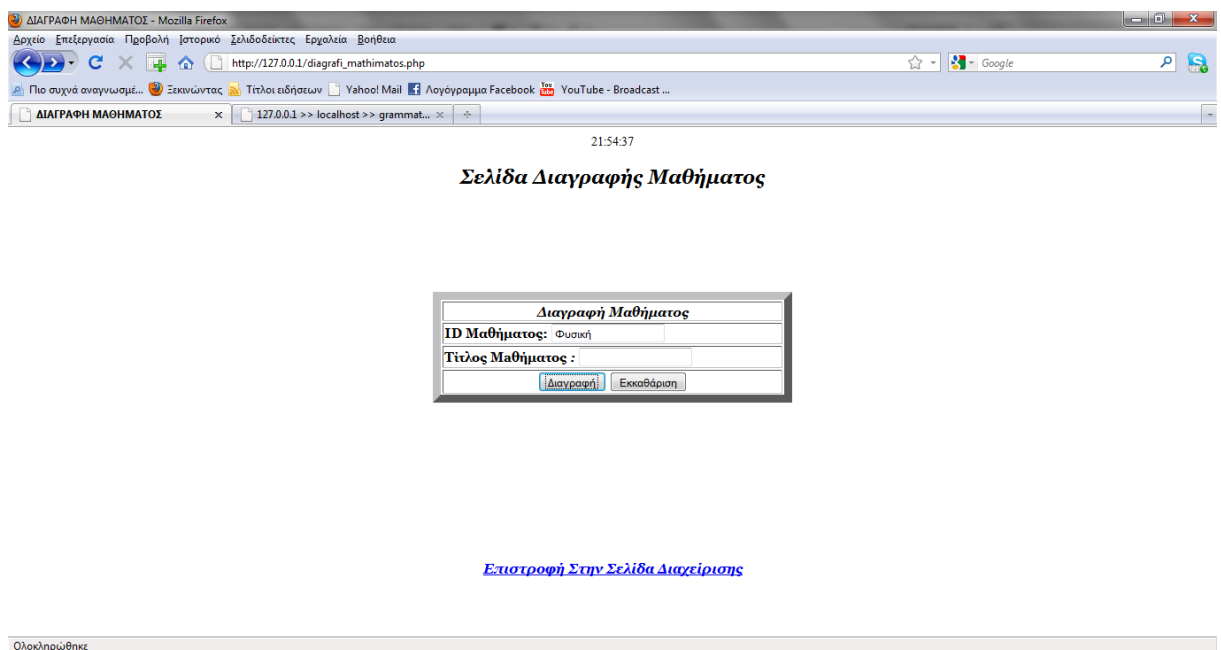
Ολοκληρώθηκε

Εικόνα 39.Μήνυμα επιτυχούς διαγραφής στοιχείων καθηγητή

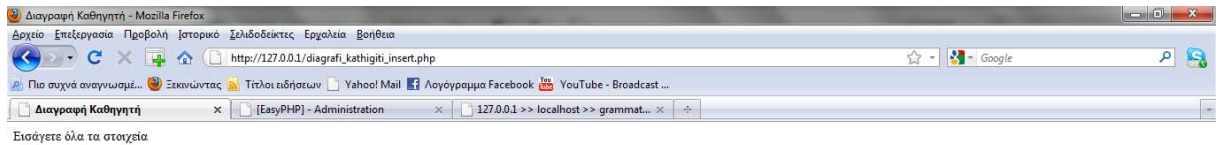
4.4.9 Σελίδα Διαγραφής Μαθήματος

Συνεχίζοντας, στις επόμενες εικόνες παραθέτουμε το επόμενο κομμάτι της εφαρμογής το οποίο αναφέρεται στην διαγραφή κάποιου μαθήματος. Διαγράφοντας ένα μάθημα αυτό αυτόματα διαγράφεται και από το πρόγραμμα σπουδών, εφόσον το έχουμε ρυθμίσει έτσι να φαίνεται και εκεί όποιο μάθημα εισάγεται κάθε φορά. Αυτό το κομμάτι θα το αναλύσουμε ξανά όταν εξηγήσουμε το κουμπί του προγράμματος σπουδών.

Σε αυτή τη φόρμα έχουμε δύο πεδία προς συμπλήρωση, ένα με το όνομα του προς διαγραφή μαθήματος και ένα με τον εκάστοτε κωδικό του. Στον πίνακα της βάσης μας ο κωδικός του μαθήματος είναι το πρωτεύων κλειδί, οπότε και το χρησιμοποιούμε σαν πεδίο στο να διαγράψουμε το μάθημα που θέλουμε. Πρέπει όμως εδώ να συμπληρώσουμε και τα δύο πεδία τα οποία είναι υποχρεωτικά σε αντίθεση με τις προηγούμενες φόρμες για να προχωρήσουμε στην διαγραφή του μαθήματος. Έτσι λοιπόν στην εικόνα 41 βλέπουμε πάλι το σχετικό μήνυμα το οποίο μας ενημερώνει ότι πρέπει να εισάγουμε όλα τα στοιχεία προκειμένου να προχωρήσουμε στην διαγραφή του μαθήματος που επιθυμούμε.



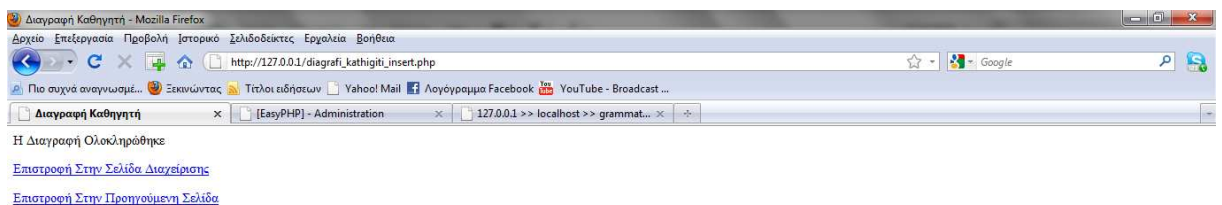
Εικόνα 40.Σελίδα διαγραφής Μαθήματος



Ολοκληρώθηκε

Εικόνα 41.Μήνυμα ελέγχου ορθής εισαγωγής στοιχείων

Στην εικόνα 42 έχουμε την σελίδα, η οποία εμφανίζεται με την ορθή εισαγωγή των στοιχείων του μαθήματος προς διαγραφή. Εμφανίζεται πάλι το σχετικό μήνυμα για την επιτυχή διαγραφή του μαθήματος και από κάτω όπως πάντα τα δύο links τα οποία μας καθοδηγούν αναλόγως.



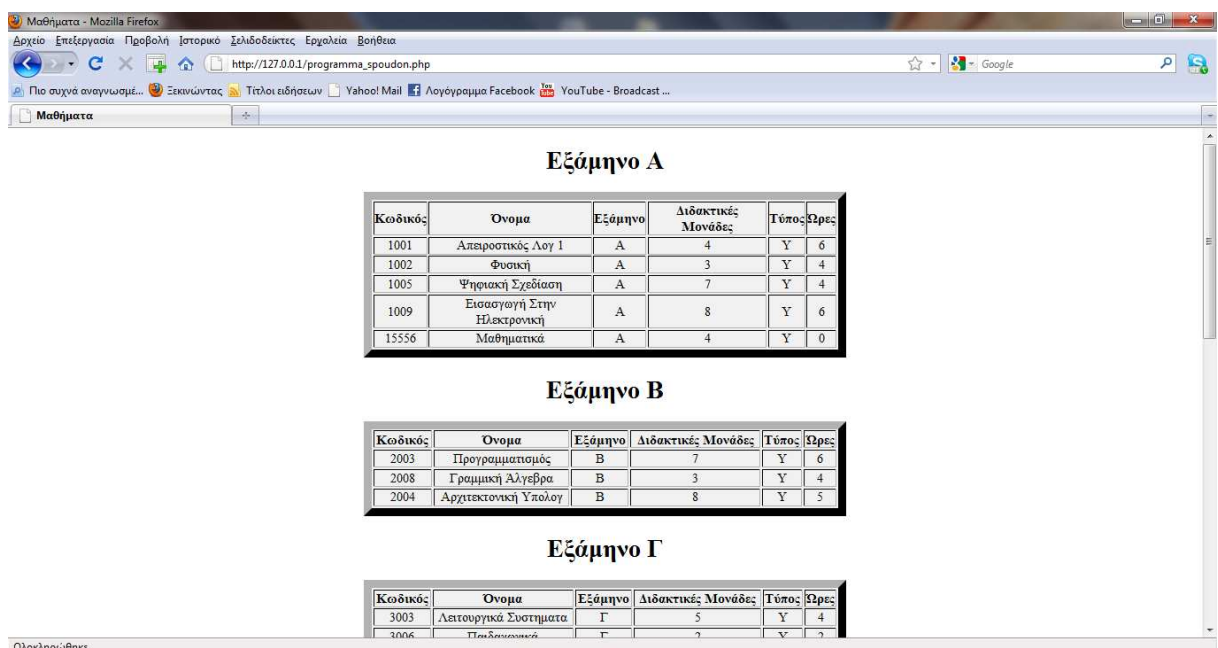
Ολοκληρώθηκε

Εικόνα 42 Μήνυμα επιτυχούς διαγραφής στοιχείων μαθήματος

4.4.10 Σελίδα Προγράμματος Σπουδών

Το επόμενο κουμπί της εφαρμογής μας, μας εμφανίζει το πρόγραμμα σπουδών της εκάστοτε σχολής. Το πρόγραμμα σπουδών ανανεώνεται δυναμικά κάθε φορά. Αυτό το επιτυγχάνουμε με την εισαγωγή και την διαγραφή μαθημάτων από τις φόρμες που αναλύσαμε παραπάνω. Δηλαδή όποτε εισάγουμε εμείς ένα μάθημα στην φόρμα εισαγωγής μαθήματος τότε αυτό αυτόματα κατατάσσεται στο πρόγραμμα σπουδών μας με βάση το εξάμηνο. Για παράδειγμα, αν εισάγουμε το μάθημα Φυσική με κωδικό 1002 ,εξάμηνο Α, τότε αυτό αυτόματα θα καταταχθεί στο πρόγραμμα σπουδών στο Α εξάμηνο. Όταν δε διαγράψουμε ένα μάθημα δεν χρειαζόμαστε το εξάμηνο αφού ζητάμε τον κωδικό ο οποίος είναι μοναδικός για κάθε μάθημα και έτσι δεν μπορούμε να κάνουμε λάθος στην διαγραφή. Γι'αυτό το λόγο και δεν χρησιμοποιούμε στην φόρμα της διαγραφής μαθήματος το εξάμηνο.

Στις παρακάτω εικόνες βλέπουμε όλο το πρόγραμμα σπουδών πως εμφανίζεται κάνοντας scroll down στην οθόνη του υπολογιστή μας. Έχουμε περάσει κάποια μαθήματα ενδεικτικά για το σκοπό αυτό.



The screenshot shows a web browser window with the URL http://127.0.0.1/programme_spondon.php. The page displays three tables of courses, one for each semester (A, B, and Gamma). Each table has columns for Course Code, Name, Semester, Credit Units, Type, and Hours.

Εξάμηνο Α

Κωδικός	Όνομα	Εξάμηνο	Διδακτικές Μονάδες	Τύπος	Ώρες
1001	Αλγεβρικός Λογ 1	A	4	Υ	6
1002	Φυσική	A	3	Υ	4
1005	Ψηφιακή Σχεδίαση	A	7	Υ	4
1009	Εισαγωγή Στην Ηλεκτρονική	A	8	Υ	6
15556	Μαθηματικά	A	4	Υ	0

Εξάμηνο Β

Κωδικός	Όνομα	Εξάμηνο	Διδακτικές Μονάδες	Τύπος	Ώρες
2003	Προγραμματισμός	B	7	Υ	6
2008	Γραμμική Άλγεβρα	B	3	Υ	4
2004	Αρχιτεκτονική Υπόλογ	B	8	Υ	5

Εξάμηνο Γ

Κωδικός	Όνομα	Εξάμηνο	Διδακτικές Μονάδες	Τύπος	Ώρες
3003	Λειτουργικά Συστήματα	Γ	5	Υ	4
3006	Παράδειγμα	Γ	?	Υ	?

Εικόνα 43.Πρόγραμμα Σπουδών

Μαθήματα - Mozilla Firefox

Δοχείο Επεξεργασία Προβολή Ιστορικό Σελιδοδείκτες Εργαλεία Βοήθεια

http://127.0.0.1/programma_spoudon.php

Πιο συχνά αναγνωριμέ... Σελκινώντας Τίτλοι ειδήσεων Yahoo! Mail Λογόγραμμα Facebook YouTube - Broadcast ...

Μαθήματα

Εξάμηνο Γ

Κωδικός	Όνομα	Εξάμηνο	Διδακτικές Μονάδες	Τύπος	Ωρες
3003	Λειτουργικά Συστήματα	Γ	5	Υ	4
3006	Παιδαγωγικά	Γ	2	Υ	2
3007	Διακριτά Μαθηματικά	Γ	3	Υ	2

Εξάμηνο Δ

Κωδικός	Όνομα	Εξάμηνο	Διδακτικές Μονάδες	Τύπος	Ωρες
4000	Αντικεμ. Προγραμματισμός	Δ	5	Υ	4
4005	Ξένη Γλώσσα Ορολογία	Δ	3	Υ	3

Εξάμηνο Ε

Κωδικός	Όνομα	Εξάμηνο	Διδακτικές Μονάδες	Τύπος	Ωρες
5002	Δίκτυα Υπολογιστών	Ε	5	Υ	4
5013	Οπτικές Επικοινωνίες	Ε	4	Ε	3
510	Ψηφιακή Επεξεργασία Σημάτων	Ε	6	Υ	2

Ολοκληρώθηκε

Εικόνα 44. Πρόγραμμα Σπουδών

Μαθήματα - Mozilla Firefox

Δοχείο Επεξεργασία Προβολή Ιστορικό Σελιδοδείκτες Εργαλεία Βοήθεια

http://127.0.0.1/programma_spoudon.php

Πιο συχνά αναγνωριμέ... Σελκινώντας Τίτλοι ειδήσεων Yahoo! Mail Λογόγραμμα Facebook YouTube - Broadcast ...

Μαθήματα

Εξάμηνο Ε

Κωδικός	Όνομα	Εξάμηνο	Διδακτικές Μονάδες	Τύπος	Ωρες
5002	Δίκτυα Υπολογιστών	Ε	5	Υ	4
5013	Οπτικές Επικοινωνίες	Ε	4	Ε	3
510	Ψηφιακή Επεξεργασία Σημάτων	Ε	6	Υ	2

Εξάμηνο ΣΤ

Κωδικός	Όνομα	Εξάμηνο	Διδακτικές Μονάδες	Τύπος	Ωρες
6009	Ασύρματα Δίκτυα	ΣΤ	6	Ε	4
6015	Θέματα Προγραμματισμού Διαδ	ΣΤ	6	Υ	4

Εξάμηνο Ζ

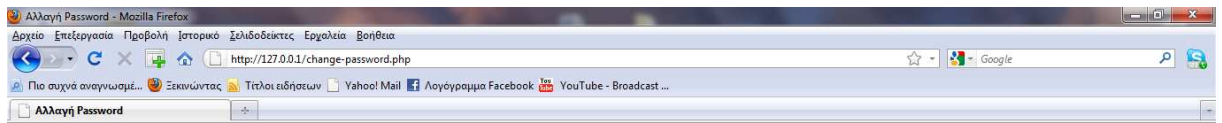
Κωδικός	Όνομα	Εξάμηνο	Διδακτικές Μονάδες	Τύπος	Ωρες
7014	Marketing	Ζ	1	Ε	2

[Επιστροφή Στην Σελίδα Αναγείρισης](#)

Ολοκληρώθηκε

Εικόνα 45. Πρόγραμμα Σπουδών

4.4.11 Σελίδα Αλλαγής Κωδικού



Σελίδα αλλαγής κωδικού πρόσβασης

Αλλαγή κωδικού πρόσβασης	
Παλιός κωδικός	*****
Νέος κωδικός	*****
Επιβεβαίωση κωδικού	
<input type="button" value="Αλλαγή κωδικού"/> <input type="button" value="Επιστροφή"/>	

[Επιστροφή Στην Σελίδα Διαχείρισης](#)

Ολοκληρώθηκε

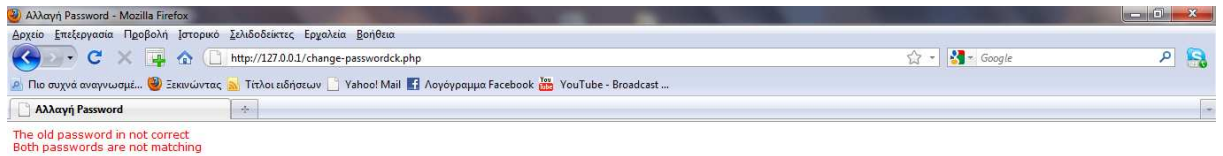
Εικόνα 46.Αλλαγή Κωδικού Πρόσβασης

Στην εικόνα 46 παραπάνω έχουμε την φόρμα αλλαγής του κωδικού πρόσβασης τον οποίο εισάγουμε στην αρχή της εφαρμογής για να εισέλθουμε στο κύριο μενού. Η φόρμα μας αποτελείται από τρία πεδία, το πεδίο του παλαιού κωδικού στο οποίο βάζουμε τον παλιό κωδικό που έχουμε. Στο επόμενο πεδίο εισάγουμε τον νέο κωδικό που επιθυμούμε και στο τρίτο πεδίο επαληθεύουμε τον νέο κωδικό που επιθυμούμε.

Σε αυτή την φόρμα απαιτείται ο έλεγχος όλων των πεδίων για πλήρη ασφάλεια. Έτσι αν αφήσουμε ένα πεδίο ασυμπλήρωτο δεν γίνεται η αλλαγή που επιθυμούμε. Επίσης εάν κάποιο από τα στοιχεία που συμπληρώσαμε στα πεδία δεν είναι σωστά ,πάλι δεν γίνεται η αλλαγή του κωδικού. Για να αλλάξουμε τον κωδικό ,πρέπει ο νέος κωδικός να έχει ψηφία όχι μικρότερα από 3 κα όχι μεγαλύτερα από 8.Αυτό ισχύει και για τα δύο πεδία που τοποθετούμε τον νέο κωδικό και την επαλήθευση του από κάτω

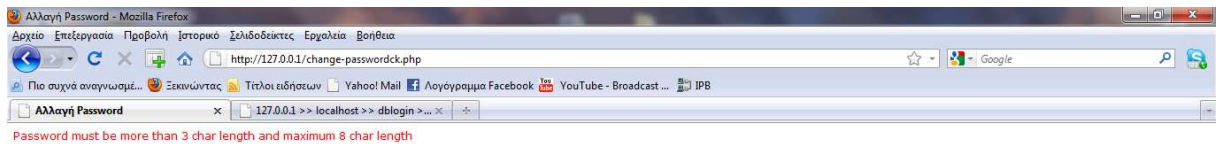
Στην εικόνα 47 παρακάτω βλέπουμε το μήνυμα το οποίο εμφανίζεται όταν δεν τηρείται κάποια από τις παραπάνω προϋποθέσεις για την ορθή αλλαγή κωδικού .Η φόρμα αυτή δεν αλλάζει το username του χρήστη .Για μεγαλύτερη ασφάλεια κωδικοποιούμε τον κωδικό με κωδικοποίηση MD5 στην βάση δεδομένων μας. Επίσης αποθηκεύουμε το username και το password σε πίνακα σε ξεχωριστή βάση δεδομένων. Και αυτό είναι θέμα ασφάλειας και καλύτερης διαχείρισης της βάσης μας. Με αυτόν τον τρόπο δεν μπερδεύουμε τα δεδομένα τα οποία αποθηκεύουμε με τον κωδικό εισόδου ο οποίος είναι και η ασφάλεια μας και για τα δεδομένα που έχουμε εισάγει.

Ανάλογα το σφάλμα που ανιχνεύεται από τον κώδικα μας εμφανίζεται και το κατάλληλο μήνυμα στην οθόνη μας. Έτσι βλέπουμε ότι μας ενημερώνει το πρόγραμμά μας ότι ο παλιός κωδικός που βάλαμε στο πρώτο πεδίο δεν είναι ο σωστός και επίσης ότι δεν είναι ίδιος ο νέος κωδικός που εισάγαμε με την επιβεβαίωση του. Στην εικόνα 48 βλέπουμε τον έλεγχο που πραγματοποιείται όταν ο παλιός κωδικός μεν είναι σωστός αλλά ο νέος κωδικός και ο έλεγχος του είναι μικρότεροι από 3 χαρακτήρες. Σε κάθε έλεγχο υπάρχει πάντα ένα κουμπί όπως παρατηρούμε στις παρακάτω εικόνες το οποίο χρησιμοποιείται για να προσπαθήσει ο χρήστης να εισάγει ξανά τα στοιχεία που θέλει. Επίσης υπάρχει και το κλασσικό link το οποίο μας παραπέμπει στη σελίδα διαχείρισης για περεταίρω ενέργειες.



Ολοκληρώθηκε

Εικόνα 47. Έλεγχος Πεδίων Αλλαγής Κωδικού



Ολοκληρώθηκε

Εικόνα 48. Έλεγχος Μήκους Νέου Κωδικού

Εάν βέβαια συμπληρώσουμε σωστά όλα τα πεδία τότε εμφανίζεται η παρακάτω σελίδα ελέγχου ,η οποία μας ενημερώνει ότι έχει γίνει επιτυχής αλλαγή του κωδικού πρόσβασης και μας ενημερώνει με μήνυμα να αλλάζουμε ταχτικά τον κωδικό εισόδου για να έχουμε μεγαλύτερη ασφάλεια στα δεδομένα μας και στο πρόγραμμά μας. Επίσης υπάρχει πάντα και το Link που μας πηγαίνει στο αρχικό μενού διαχείρισης του προγράμματος μας.



Εικόνα 49. Επιτυχής Αλλαγή Password

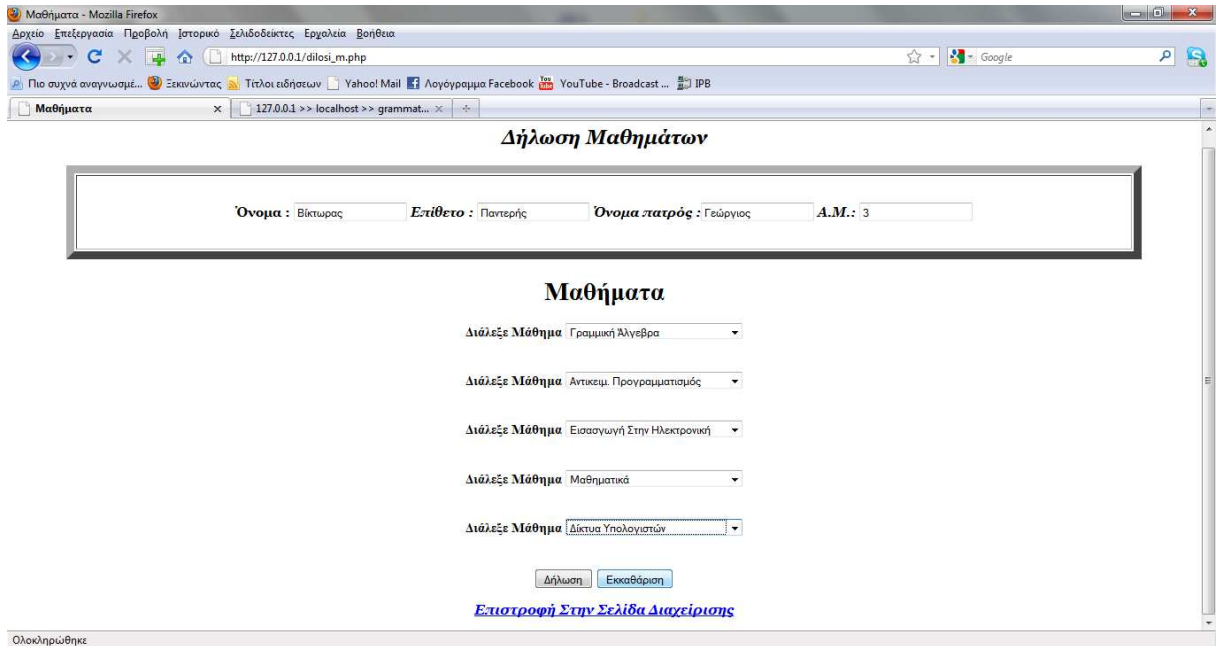
4.4.12 Σελίδα Δήλωσης Μαθημάτων

Στη συνέχεια έχουμε ένα άλλο κουμπί, το οποίο αναγράφει Δήλωση Μαθημάτων. Πατώντας το εμφανίζεται η παρακάτω σελίδα, την οποία βλέπουμε στην εικόνα 50. σε αυτή την φόρμα έχουμε διάφορα πεδία που πρέπει να συμπληρώσουμε. Αρχικά στο πάνω μέρος της σελίδας μας έχουμε την επικεφαλίδα και συνεχίζοντας από κάτω υπάρχει ένας πίνακας όπου πρέπει να συμπληρώσουμε τα στοιχεία του φοιτητή που θέλει να κάνει την δήλωση.

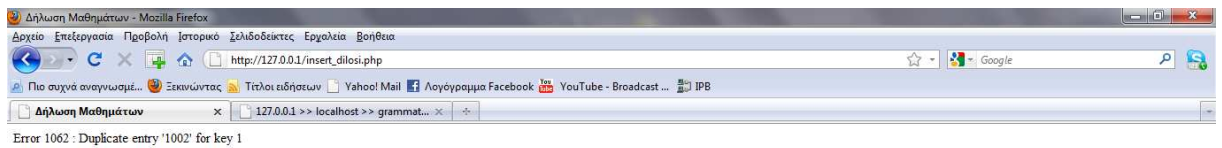
Πρέπει να συμπληρώσουμε όλα τα πεδία, ειδικότερα αυτό του ΑΜ διότι χωρίς αυτό δεν θα γνωρίζουμε ποιος φοιτητής έχει δηλώσει τι, κάθε φορά. Γιατί στον πίνακα παίρνει της βάσης μας εκτελούμε αναζήτηση κάθε φορά με βάση το ΑΜ του εκάστοτε φοιτητή. Αν λοιπόν δεν δώσουμε το ΑΜ του φοιτητή δεν θα είμαστε σε θέση να γνωρίζουμε ποιος έχει δηλώσει τα μαθήματα που γράφονται στον πίνακά μας.

Έπειτα αφού συμπληρώσουμε τα πεδία που θέλουμε, θα πρέπει να διαλέξουμε μαθήματα από τις λίστες που υπάρχουν παρακάτω. Υπάρχουν συνολικά 5 λίστες, κάτι το οποίο είναι τυπικό για την εργασία μας. Μπορούμε να βάλουμε όσες λίστες θέλουμε για να δηλώνουμε μαθήματα.

Οι λίστες αυτές ανανεώνονται δυναμικά, δηλαδή αυτόματα κάθε φορά που ο χρήστης εισάγει ή διαγράφει ένα μάθημα από τα κουμπιά εισαγωγής και διαγραφής μαθήματος. Έτσι ο χρήστης δεν έχει παρά να επιλέξει τα μαθήματα που θέλει και έπειτα να πατήσει το κουμπί που αναγράφει Δήλωση. Εάν δηλωθεί κάποιο μάθημα δύο φορές τότε εμφανίζεται σφάλμα στην οθόνη μας το οποίο προέρχεται από την βάση δεδομένων μας και πιο συγκεκριμένα από την sql, το οποίο σφάλμα το βλέπουμε στην εικόνα 51.



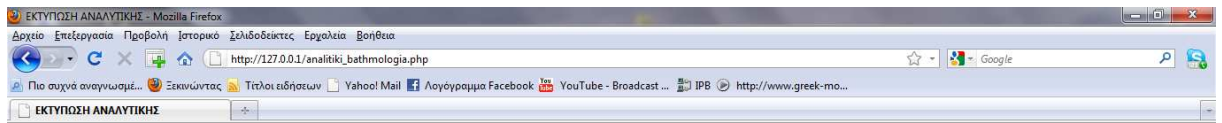
Εικόνα 50. Φόρμα Δήλωσης Μαθημάτων



Ολοκληρώθηκε

Εικόνα 51. Σφάλμα SQL διπλότυπης εισαγωγής δεδομένων

4.4.13 Σελίδα Εμφάνισης και εκτύπωσης αναλυτικής βαθμολογίας



19:48:32

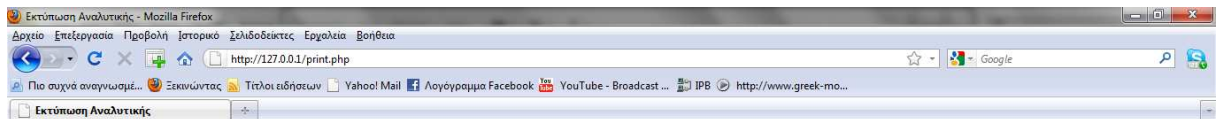
Σελίδα Εκτύπωσης Αναλυτικής Βαθμολογίας

Όνομα : Panteris
Επώνυμο : Victoras
Α.Μ. : :
<input type="button" value="Προβολή"/> <input type="button" value="Εκκαθάριση"/>

[Επιστροφή Στην Σελίδα Διαχείρισης](#)

Ολοκληρώθηκε

Εικόνα 52.Εμφάνιση εκτύπωσης αναλυτικής βαθμολογίας



Κωδικός Μαθήματος	Βαθμός Μαθήματος	Όνομα Μαθήματος	Εξάμηνο
1001	0	Αλγεβρικός Λογ 1	A
1005	0	Ψηφιακή Σχεδίαση	A

[Επιστροφή Στην Σελίδα Διαχείρισης](#)

[Επιστροφή Στην Προηγούμενη Σελίδα](#)

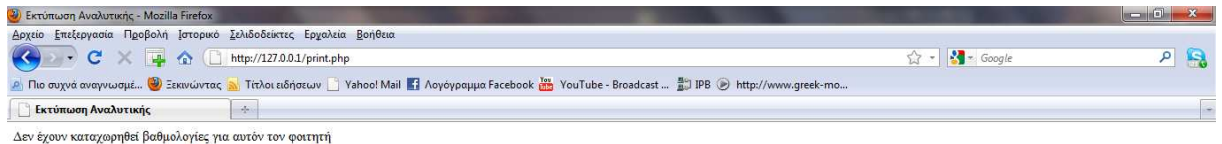
Ολοκληρώθηκε

Εικόνα 53. Σελίδα εμφάνισης στοιχείων φοιτητή

Στις εικόνες 52 και 53 στην προηγούμενη σελίδα, βλέπουμε την φόρμα στην οποία υποβάλλουμε το ΑΜ του φοιτητή καθώς και το ονοματεπώνυμο του για να μας εμφανίσει σε ένα πίνακα τα μαθήματα τα οποία έχει δηλωμένα και να μπορέσουμε μετά να τα εκτυπώσουμε. Σε αυτή την φόρμα το μόνο πεδίο που πραγματικά θέλουμε να είναι σωστό είναι το πεδίο του ΑΜ που πρέπει να είναι οπωσδήποτε συμπληρωμένο.

Στην εικόνα 52 διακρίνουμε λοιπόν τρία πεδία εκ των οποίων το τρίτο είναι στην ουσία αυτό που πρέπει να συμπληρωθεί για να πάρουμε το αποτέλεσμα μας. Στο πάνω μέρος της οθόνης μας έχουμε το ψηφιακό ρολόι και από κάτω τον τίτλο της σελίδας μας .Μετά τα πεδία έχουμε τα κουμπιά της υποβολής και της εκκαθάρισης.

Στην εικόνα 53 έχουμε τον πίνακα με τα μαθήματα που έχει δηλώσει ο φοιτητής και συγκεκριμένα βλέπουμε τέσσερις στήλες. Η πρώτη είναι ο κωδικός του μαθήματος, η δεύτερη είναι ο βαθμός που έχει πάρει στο συγκεκριμένο μάθημα, η τρίτη είναι το όνομα του μαθήματος και η τέταρτη είναι το εξάμηνο στο οποίο ανήκει το μάθημα. Τέλος έχουμε τα κλασσικά links τα οποία μας οδηγούν είτε στην προηγούμενη φόρμα είτε στην αρχική σελίδα διαχείρισης.



Ολοκληρώθηκε

Εικόνα 54. Σελίδα ελέγχου ορθότητας ΑΜ φοιτητή

Στην εικόνα 54 βλέπουμε το μήνυμα που εμφανίζεται όταν δεν υπάρχει το ΑΜ την βάση ή όταν δεν είναι το σωστό ΑΜ ή όταν δεν βάλουμε καθόλου ΑΜ στο πεδίο που ζητείται.

Κεφάλαιο 5

Σε αυτό το κεφάλαιο το οποίο είναι και το τελικό μας, θα εκθέσουμε τα συμπεράσματα και την άποψη μας για την λειτουργία της εφαρμογής μας και για την περαιτέρω ανάπτυξη της και την χρήση που μπορεί να βρει σε κάποια ιδρύματα .Θα αναλύσουμε επίσης τα αποτελέσματα που προκύπτουν από την εφαρμογή που αναλύσαμε στο 4^ο κεφάλαιο

5.1 Αποτελέσματα- Χρησιμότητα

Η εφαρμογή μας έχει σαν αποτέλεσμα την ευκολότερη διαχείριση του φόρτου και όγκου εργασίας που διαχειρίζεται μία γραμματεία κάποιου τεχνολογικού ιδρύματος είτε πανεπιστημίου. Σαν εφαρμογή μπορεί να είναι φτιαγμένη για γραμματεία, όμως γενικά η δομή της με κάποιες αλλαγές και κάποιες προσθήκες μπορεί να χρησιμοποιηθεί πέρα από τα προαναφερθέντα.

Μια γραμματεία διαχειρίζεται καθημερινά ένα όγκο δεδομένων τα οποία καλείται να αποθηκεύσει σε κάποιο χώρο ,να τα διαχειριστεί και να παράγει ένα αποτέλεσμα .Με την παρούσα εφαρμογή ελέγχουμε καλύτερα τα δεδομένα τα οποία αποθηκεύουμε και τα διαχειριζόμαστε ευκολότερα. Επίσης έχουμε οικονομία χώρου καθώς αποθηκεύουμε τα δεδομένα σε κάποιο σκληρό δίσκο και όχι σε φακέλους σε κάποιο ντουλάπι το οποίο κάποια στιγμή θα γεμίσει και θα χρειαστεί ένα καινούργιο.

Βέβαια η εφαρμογή μας δεν είναι αυτό που λέμε πλήρης, δηλαδή να είναι ολοκληρωμένη .Λέγοντας ολοκληρωμένη εννοούμε να υπάρχει και η πλευρά του φοιτητή, δηλαδή να μπορεί και ο φοιτητής να μπαίνει στην εφαρμογή με δικό του κωδικό τον οποίο εάν θέλει να μπορεί να αλλάζει, να διαλέγει μόνος του μαθήματα και να εγγράφεται στο εξάμηνο του σε προκαθορισμένες βέβαια πάντα ημερομηνίες και να μπορεί να βλέπει την ηλεκτρονική του καρτέλα με τα μαθήματα που έχει περάσει και τους βαθμούς που έχει περάσει το εκάστοτε μάθημα. Όλες αυτές τις λειτουργίες και όσες χρειάζονται ακόμα θα μπορούσαν να αποτελέσουν ένα δεύτερο κομμάτι πτυχιακής εργασίας καθώς απαιτείται η ίδια δουλειά που είχε η εφαρμογή μας για να φτάσει στο παρόν στάδιο .Θα μπορούσε επίσης για περαιτέρω ανάπτυξη και τελειοποίηση να διαμορφώναμε το interface της παρούσας εφαρμογής όταν τελειώσουμε από το κομμάτι του κώδικα ,να το κάνουμε δηλαδή πιο φιλικό και πιο εμπορικό προς τον χρήστη. Να βάλουμε δηλαδή κάποια χρώματα, ένα template πιο φιλικό και πιο ευχάριστο και να οργανώναμε τα κουμπιά σε άλλη διάταξη. Και αυτό το κομμάτι βέβαια χρειάζεται την δική του δουλειά και το οποίο θα μπορούσε να αποτελέσει ξεχωριστό κομμάτι εργασίας.

Με το πέρας αυτής της εργασίας έχουμε κατανοήσει αρκετά πράγματα,. Δουλεύοντας πάνω στο χτίσιμο της εφαρμογής κατανοήσαμε πως λειτουργεί μία εφαρμογή φτιαγμένη με την γλώσσα php,πως φτιάχνουμε τα κουμπιά τα οποία μας πηγαίνουν από ην μία σελίδα στην άλλη, πως φτιάχνουμε τους υπερσυνδέσμους. Κατανοήσαμε επίσης πως φτιάχνεται μια βάση δεδομένων η οποία αλληλεπιδρά με τον κώδικα στην σελίδα μας και συνεργάζεται αρμονικά, αυτό που λέμε με πιο απλά λόγια δυναμική εφαρμογή. Μελετήσαμε και εκπονήσαμε το στήσιμο μίας βάσης από την αρχή ,δηλαδή από το διάγραμμα οντοτήτων-συσχετίσεων, μέχρι την πλήρη υλοποίηση σε γραφικό περιβάλλον με το πρόγραμμα Easy Php το οποίο και αναλύσαμε σε προηγούμενο κεφάλαιο. Είδαμε την σύνδεση της σελίδα μας με την βάση δεδομένων την οποία υλοποιήσαμε με την χρήση της γλώσσας php και το οποίο θα αναλύσουμε στο παράρτημα όπου και θα εξηγήσουμε και όλο τον κώδικα της εφαρμογής μας καθώς και τον παραγόμενο sql κώδικα από την βάση δεδομένων, μιας και όλες τις ενέργειες τις κάναμε γραφικά και θα άξιζε τον κόπο να εξηγήσουμε ορισμένα πράγματα. Αποκομίσαμε γενικές γνώσεις πάνω στην κατασκευή μιας δυναμικής ιστοσελίδας με προοπτική για περαιτέρω ανάπτυξη και εκπαίδευση πάνω στην κατασκευή ιστοσελίδων.

Παράρτημα Α

Στο παράρτημα αυτό θα παραθέσουμε και θα αναλύσουμε εκτενέστερα τον παραγόμενο κώδικα SQL αρχικά και έπειτα τον κώδικα PHP που χρησιμοποιήσαμε σε κάθε σελίδα.

A.1 Ανάλυση Κώδικα SQL

A.1.1 Ανάλυση Κώδικα Βάσης με όνομα Grammateia (Πίνακες 8)

Βάση 'Γραμματεία'

```
CREATE DATABASE `grammateia1` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
```

Με την γραμμή αυτή δημιουργούμε την βάση δεδομένων μας με το όνομα που επιθυμούμε και την κωδικοποίηση που θέλουμε.

Πίνακας 'Ανήκει'

```
CREATE TABLE `anikei` (  
  `ΕΞΑΜΗΝΟ` VARCHAR( 5 ) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL  
  ) CHARACTER SET utf8 COLLATE utf8_general_ci;
```

Με την πρώτη γραμμή δημιουργούμε τον πίνακα με το όνομα που θέλουμε και με τις δύο επόμενες γραμμές δίνουμε όνομα στα πεδία που θέλουμε και θέτουμε την κωδικοποίηση του κάθε πεδίου ξεχωριστά (σε όλα τα πεδία βάζουμε το utf_8_general_ci).

Πίνακας 'Διδάσκει'

```
CREATE TABLE `didaskei` (  
  `Α.Δ.Τ` VARCHAR( 8 ) CHARACTER SET ucs2 COLLATE ucs2_general_ci NOT NULL  
  ) CHARACTER SET utf8 COLLATE utf8_general_ci;
```

Και εδώ ισχύει γενικά το ίδιο με την δημιουργία του προηγούμενου πίνακα.

Πίνακας 'Φοιτητής'

```
CREATE TABLE `foititis` (  
  `ονομα` VARCHAR( 20 ) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL ,  
  `εponumo` VARCHAR( 30 ) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL ,
```

```

`onoma_patros` VARCHAR( 20 ) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL

`,`onoma_mitros` VARCHAR( 20 ) CHARACTER SET utf8 COLLATE utf8_general_ci NOT
NULL ,
`A_M` INT( 6 ) NOT NULL AUTO_INCREMENT ,
`seira_epitixias` INT( 5 ) NOT NULL ,
`M_O` FLOAT( 5 ) NOT NULL ,
`A_D_T` VARCHAR( 8 ) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL ,
`diefthinsi_monimis_katoikias` VARCHAR( 30 ) CHARACTER SET utf8 COLLATE
utf8_general_ci NOT NULL ,
`diefthinsi_topikis_katoikias` VARCHAR( 30 ) CHARACTER SET utf8 COLLATE
utf8_general_ci NOT NULL ,
`telefono_monimis_katoikias` VARCHAR( 10 ) CHARACTER SET utf8 COLLATE
utf8_general_ci NOT NULL ,
`telefono_topikis_katoikias` VARCHAR( 10 ) CHARACTER SET utf8 COLLATE utf8_general_ci
NOT NULL ,
`email` VARCHAR( 30 ) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL ,
PRIMARY KEY ( `A_M` )
) CHARACTER SET utf8 COLLATE utf8_general_ci;

```

Οι εντολές και εδώ είναι ίδιες με την δημιουργία των προηγούμενων πινάκων, δηλαδή με την πρώτη γραμμή δημιουργούμε τον πίνακα με το όνομα που θέλουμε, με τις επόμενες εντολές δίνουμε όνομα στα πεδία και την κωδικοποίηση που θέλουμε. Στην προτελευταία γραμμή βλέπουμε την εντολή με την οποία θέτουμε το πρωτεύων κλειδί του πίνακα μας και στην τελευταία γραμμή δίνουμε την γενική κωδικοποίηση του πίνακα μας. Αν θέλουμε δεν δίνουμε κωδικοποίηση σε κάθε πεδίο και δίνουμε την κωδικοποίηση στο τέλος για όλο τον πίνακα.

Πίνακας 'Καθηγητής'

```

CREATE TABLE `kathigitis` (
`A_D_T` VARCHAR( 10 ) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL ,
`bathmida` VARCHAR( 15 ) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL ,
`diefthinsi_grafeiou` VARCHAR( 20 ) CHARACTER SET utf8 COLLATE utf8_general_ci NOT
NULL ,
`onoma` VARCHAR( 10 ) CHARACTER SET ucs2 COLLATE ucs2_general_ci NOT NULL ,
`eponumo` VARCHAR( 15 ) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL ,
`telefono_oikias` VARCHAR( 10 ) CHARACTER SET utf8 COLLATE utf8_general_ci NOT
NULL ,
`email` VARCHAR( 30 ) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL ,
PRIMARY KEY ( `A_D_T` )
) CHARACTER SET utf8 COLLATE utf8_general_ci;

```

Και εδώ ισχύει γενικά το ίδιο με την δημιουργία του προηγούμενου πίνακα.

Πίνακας 'Μάθημα'

```
CREATE TABLE `mathima` (  
  `kodikos_m` VARCHAR( 10 ) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL ,  
  `onoma` VARCHAR( 30 ) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL ,  
  `examino` VARCHAR( 5 ) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL ,  
  `D_M` INT( 11 ) NOT NULL ,  
  `typos_mathimatos` VARCHAR( 10 ) CHARACTER SET utf8 COLLATE utf8_general_ci NOT  
  NULL ,  
  `wres` INT( 10 ) NOT NULL ,  
  PRIMARY KEY ( `kodikos_m` )  
 ) CHARACTER SET utf8 COLLATE utf8_general_ci;
```

Και εδώ ισχύει γενικά το ίδιο με την δημιουργία του προηγούμενου πίνακα.

Πίνακας 'Παίρνει '

```
CREATE TABLE `pairnei` (  
  `A_M` VARCHAR( 5 ) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL ,  
  `GRADE` FLOAT( 5 ) NOT NULL ,  
  `ONOMA` VARCHAR( 20 ) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL ,  
  `EXAMINO` VARCHAR( 5 ) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL ,  
  `CODE` VARCHAR( 10 ) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL  
 ) CHARACTER SET utf8 COLLATE utf8_general_ci;
```

Και εδώ ισχύει γενικά το ίδιο με την δημιουργία του προηγούμενου πίνακα.

Πίνακας 'Προαπαιτούμενα'

```
CREATE TABLE `proapaitoumena` (  
  `KWDIKOS_PROHGOUMENOU` VARCHAR( 10 ) CHARACTER SET utf8 COLLATE  
  utf8_general_ci NOT NULL ,  
  `KWDIKOS_EPOMENOU` VARCHAR( 10 ) CHARACTER SET utf8 COLLATE utf8_general_ci  
  NOT NULL ,  
  PRIMARY KEY ( `KWDIKOS_PROHGOUMENOU` , `KWDIKOS_EPOMENOU` )  
 ) CHARACTER SET utf8 COLLATE utf8_general_ci;
```

Και εδώ ισχύει γενικά το ίδιο με την δημιουργία του προηγούμενου πίνακα.

Πίνακας 'Πρόγραμμα Σπουδών'

```
CREATE TABLE `programma spoudwn` (  
  `ΚΩΔΙΚΟΣΠ` VARCHAR( 10 ) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL ,  
  `ΑΠΟ-ΕΩΣ` VARCHAR( 20 ) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL ,  
  `ΚΩΔΙΚΟΣΜ` VARCHAR( 20 ) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL ,  
  PRIMARY KEY ( `ΚΩΔΙΚΟΣΠ` )  
 ) CHARACTER SET utf8 COLLATE utf8_general_ci;
```

Και εδώ ισχύει γενικά το ίδιο με την δημιουργία του προηγούμενου πίνακα.

A.1.2 Ανάλυση Κώδικα Βάσης με όνομα Db Login (Πίνακες 1)

Βάση 'Db Login'

```
CREATE DATABASE `dblogin1` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
```

Πάλι δημιουργούμε την βάση μας με το επιθυμητό μας όνομα και θέτουμε την κωδικοποίηση μας όπως εμείς θέλουμε.

Πίνακας 'User'

```
CREATE TABLE `user` (  
  `username` VARCHAR( 10 ) CHARACTER SET ucs2 COLLATE ucs2_general_ci NOT NULL ,  
  `password` VARCHAR( 10 ) CHARACTER SET ucs2 COLLATE ucs2_general_ci NOT NULL  
 ) CHARACTER SET utf8 COLLATE utf8_general_ci;
```

Και εδώ ισχύει γενικά το ίδιο με την δημιουργία των προηγούμενων πινάκων και δεν θέτουμε κάποιο πρωτεύων κλειδί.

A.2 Ανάλυση Κώδικα PHP

A.2.1 Αρχική Σελίδα Εισόδου

Κώδικας 'Loginprog.php'

```
<head>
<META HTTP-EQUIV="Content-Type" content="text/html; charset=iso-8859-7">
<title>ΕΙΣΟΔΟΣ</title>
</head>
<html xmlns="http://www.w3.org/1999/xhtml">
<div align="center"><body>
<div align="left">
  <script type="text/javascript">
function startTime()
{var today =new Date();
var h=today.getHours();
var m=today.getMinutes();
var s=today.getSeconds();
m=checkTime(m);
s=checkTime(s);
document.getElementById('txt').innerHTML=h+":"+m+":"+s;
t=setTimeout('startTime()',500);
}
function checkTime(i)
{if (i<10)
{i="0"+i;
} return i;
} </script>
  </head>
  <body onload="startTime()">
</div>
<div id="txt"></div>

  <h2 align="center"><em><font face="Georgia, Times New Roman, Times,
serif"><strong>ΓΡΑΜΜΑΤΕΙΑ ΠΑΝΕΠΙΣΤΗΜΙΟΥ </strong></font></em></h2>
  <p align="center">&nbsp;</p>
  <p align="center">&nbsp;</p>
  <table border="10" align="center" TD width="30%" height="100">
  <form method="POST" action="loginproc.php">
  <tr><th><strong><font face="Georgia, Times New Roman, Times, serif">Είσοδος
  Διαχείρισης</font></strong></th></tr>
  <tr><td align="center"><strong><font face="Georgia, Times New Roman, Times,
serif">Όνομα Χρήστη:
    <input type="text" name="username" size="20">
  </font></strong></td>
  </tr>
  <tr><td align="center"><strong><font face="Georgia, Times New Roman, Times,
serif">Κωδικός Πρόσβασης:
    <input type="password" name="password" size="20">
```

```

</font></strong></td>
</tr>
<tr><td align="center"> <table border="1" align="center">
  <tr>
    <td ><strong><font face="Georgia, Times New Roman, Times, serif">
      <input type="submit" value="Είσοδος " name="B6">
    </font></strong></td>
    <td ><strong><font face="Georgia, Times New Roman, Times, serif">
      <input type="reset" value="Επαναφορά" name="B7">
    </font></strong></td>
  </table> </td></tr>

</form>
</table>
<div align="center">
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>

  <p><font size="4" face="Georgia, Times New Roman, Times,
serif"><strong></strong></font></p>
</div></body></html>

```

Καθώς εισερχόμαστε στην εφαρμογή μας ,το πρώτο κομμάτι κώδικα που συναντάμε είναι αυτό που μας εμφανίζει την φόρμα εισαγωγής στοιχείων , δηλαδή username και password.Αυτό επιτυγχάνεται με τον παραπάνω κώδικα. Αναλύοντας τον, στην αρχή δηλώνουμε τον τίτλο της σελίδας μας και μετά φτιάχνουμε την συνάρτηση σε JavaScript η οποία μας εμφανίζει το ψηφιακό ρολόι στο πάνω μέρος της σελίδας μας. Την συνάρτηση αυτή την δηλώνουμε στο κομμάτι του head. Στο body δίνουμε τον τίτλο της σελίδας μας ο οποίος εμφανίζεται στην μέση της οθόνης. Μετά δημιουργούμε την φόρμα υποβολής στοιχείων η οποία είναι αυτό που βλέπουμε και γράφουμε το username και password και υποβάλλεται στον επόμενο κώδικα ο οποίος εκτελεί τον έλεγχο των στοιχείων.Αφού φτιάξουμε την φόρμα και τα κουμπιά κλείνουμε το body και γενικά το html έγγραφο.

A.2.2 Σελίδα Υποβολής και Ελέγχου Στοιχείων Χρήστη

```

<?
error_reporting(E_ALL & ~E_NOTICE);
session_start();

$server = "localhost";
$username = "root";
$password = "";
$db_name = "dblogin";

$db = mysql_connect($server,$username,$password) or DIE("Connection to database failed,
perhaps the service is down !!");
mysql_query("set names 'greek'", $db);
mysql_query("set character set 'greek'", $db);
mysql_select_db($db_name) or DIE("Database name not available !!");
$username=$_POST['username'];

```

```

$password=$_POST['password'];

    $login = mysql_query("select * from user where (username = " . $_POST['username'] . ")
and (password = " . ($_POST['password']) . ")",$db);
    $rowcount = mysql_num_rows($login);
    if ($rowcount == 1) {
    $_SESSION['username'] = $_POST['username'];
    header("Location: selida_diaxisirisis.php");
    }
    else
    {
    echo "Άρθόαδâ ËÛèìð Óðïé÷âβά";
    //header("Location: loginpage.php");

    }
?>

```

Σκοπός του κώδικα αυτού είναι να ελέγχει τα στοιχεία username και password και αν είναι σωστά να εισέρχεται ο χρήστης στην εφαρμογή αλλιώς να εμφανίζει μήνυμα λάθους. Δημιουργούμε λοιπόν μία session και μετά αποθηκεύουμε σε κάποιες μεταβλητές το όνομα του server μας καθώς και το username του και το όνομα της βάσης μας και τον κωδικό εισόδου, αν υπάρχει, σε αυτήν. Μετά με ένα ερώτημα sql εκτελούμε την σύνδεση με την βάση μας και σε περίπτωση που δεν γίνεται μας πετάει μήνυμα λάθους. Κατόπιν παίρνουμε από την βάση μας το username και το password που είναι αποθηκευμένα και τα βάζουμε σε δύο μεταβλητές. Τέλος με μία if ελέγχουμε αν είναι σωστά και αν ναι τότε μας πάει στην αρχική σελίδα διαχείρισης αλλιώς μας πετάει μήνυμα λάθους.

A.2.3 Σελίδα Εξόδου Εφαρμογής

```

<? session_start();
unset($_SESSION['username']);
session_destroy();
header("Location: loginprog.php");
?>

```

Σε αυτό το κομμάτι του κώδικα δημιουργούμε μία session, ο σκοπός της οποίας είναι να κρατά δεδομένα από το πέρασμα από μια σελίδα, σε μία άλλη. Εδώ λοιπόν κρατάμε τα username και password και καθώς εξερχόμαστε της εφαρμογής τότε καταστρέφουμε την session αυτή και πηγαίνουμε στην αρχική σελίδα εισαγωγής username και password.

A.2.4 Σελίδα Εγγραφής Φοιτητή

Κώδικας 'eggrafi_foititi.php'

```

<?php
$conn=mysql_connect("localhost","root",""); // (host, username, password)
mysql_query("set names 'greek'", $conn);

```

```
//mysql_query("set character set 'greek'", $conn);
$db = mysql_select_db('grammateia',$conn);
```

```
?>
```

```
<head>
<META HTTP-EQUIV="Content-Type" content="text/html; charset=iso-8859-7">
<title>ΕΓΓΡΑΦΗ ΦΟΙΤΗΤΗ</title>
</head>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
```

```
<body></head><html xmlns="http://www.w3.org/1999/xhtml">
<div align="center"><body>
<div align="left">
  <script type ="text/javascript" >
function startTime()
{var today =new Date();
var h=today.getHours();
var m=today.getMinutes();
var s=today.getSeconds();
m=checkTime(m);
s=checkTime(s);
document.getElementById('txt').innerHTML=h+":"+m+":"+s;
t=setTimeout('startTime()',500);
}
function checkTime(i)
{if (i<10)
{i="0"+i;
} return i;
}
</script>
```

```
<script type="text/javascript">
function isNumber(obj)
{
if(document.getElementById('err'))
{
obj.parentNode.removeChild(document.getElementById('err'));
}
setTimeout(function(){
if(isNaN(obj.value))
{
obj.select();
var p=document.createElement('p');
p.style.color="red";
p.style.fontStyle='italic';
p.style.margin="0px";
p.style.padding="0px";
p.id='err';
p.appendChild(document.createTextNode('Input must be numeric'));
obj.parentNode.appendChild(p);
}
}
```

```

},1);
}

function CheckRequiredFields() {

var errormessage = new String();

if(WithoutContent(document.foititis.name.value))
    { errormessage += "\n\nΣυμπληρώστε το πεδίο του Ονόματος."; }

if(WithoutContent(document.foititis.surname.value))
    { errormessage += "\n\nΣυμπληρώστε το πεδίο του Επιθέτου."; }

if(WithoutContent(document.foititis.f_name.value))
    { errormessage += "\n\nΣυμπληρώστε το πεδίο του Ονόματος Πατρός ."; }

if(WithoutContent(document.foititis.m_name.value))
    { errormessage += "\n\nΣυμπληρώστε το πεδίο του Ονόματος Μητρός."; }

if(WithoutContent(document.foititis.id.value))
    { errormessage += "\n\nΣυμπληρώστε το πεδίο του Αριθμου Ταυτότητας."; }

if(WithoutContent(document.foititis.p_address.value))
    { errormessage += "\n\nΣυμπληρώστε το πεδίο της Μόνιμης Διεύθυνσης."; }

if(WithoutContent(document.foititis.address.value))
    { errormessage += "\n\nΣυμπληρώστε το πεδίο της Προσωρινής Διεύθυνσης."; }

if(WithoutContent(document.foititis.email.value))
    { errormessage += "\n\nΣυμπληρώστε το πεδίο του Email."; }

if(errormessage.length > 0) {
    alert('NOTE:' + errormessage);
    return false;
}
return true;
}

function WithoutContent(ss) {
if(ss.length > 0) { return false; }
return true;
}

</script>

</head>
<body onload="startTime()">
</div>
<div id="txt"></div>
<div align="center">
<h2 align="center"><em><font face="Georgia, Times New Roman, Times,
serif"><strong>Σελίδα Εγγραφής Φοιτητή</strong></font></em></h2>
<p align="center">&nbsp;</p>

```

```

</div>
<p align="center"><table border="10" align="center" TD width="30%" height="100">
  <form method="POST" name="foititis" action="eggrafi_foititi_insert.php"
onSubmit="return CheckRequiredFields()">
  <tr>
    <th><strong><font face="Georgia, Times New Roman, Times, serif"><em><font
face="Georgia, Times New Roman, Times, serif"><strong>Εγγραφή
Φοιτητή</strong></font></em></font></strong></th>
    </tr>
    <tr><td align="left"><strong><font face="Georgia, Times New Roman, Times,
serif">Όνομα :</strong></font></strong></td>
      <input type="text" name="name" size="20">
    </font></strong></td>
    </tr>
    <tr><td align="left"><strong><font face="Georgia, Times New Roman, Times,
serif">Επίθετο :
      <input type="text" name="surname" size="20">
    </font></strong></td>
    </tr> <tr>
      <td align="left"><strong><font face="Georgia, Times New Roman, Times,
serif"><em>Όνομα Πατρός :</em>
      <input type="text" name="f_name" size="20">
    </font></strong></td>
    </tr> <tr>
      <td align="left"><strong><font face="Georgia, Times New Roman, Times,
serif"><em>Όνομα Μητρός :</em>
      <input type="text" name="m_name" size="20">
    </font></strong></td>
    </tr> <tr>
      <td align="left"><strong><font face="Georgia, Times New Roman, Times, serif">Σειρά
Επιτυχίας :
      <input type="text" name="row" size="20" onblur="isNumber(this);" >
    </font></strong></td>
    </tr> <tr>
      <td align="left"><strong><font face="Georgia, Times New Roman, Times, serif">Μ.Ο :
      <input type="text" name="mo" size="20" onblur="isNumber(this);" >
    </font></strong></td>
    </tr> <tr>
      <td align="left"><strong><font face="Georgia, Times New Roman, Times,
serif">Α.Δ.Τ. :
      <input type="text" name="id" size="20">
    </font></strong></td>
    </tr> <tr>
      <td align="left"><strong><font face="Georgia, Times New Roman, Times,
serif">Διεύθυνση μόν. κατοικίας:
      <input type="text" name="p_address" size="20">
    </font></strong></td>
    </tr><tr>
      <td align="left"><strong><font face="Georgia, Times New Roman, Times,
serif">Διεύθυνση προσ. κατοικίας:
      <input type="text" name="address" size="20">
    </font></strong></td>
    </tr> <tr>
      <td align="left"><strong><font face="Georgia, Times New Roman, Times,
serif"><em>Τηλέφωνο Μόνιμης Οικίας :</em>

```



```

        <input type="text" name="p_phone" size="20" onblur="isNumber(this);">
    </font></strong></td>
</tr> <tr>
    <td align="left"><strong><font face="Georgia, Times New Roman, Times,
serif"><em>Τηλέφωνο Προσωπικής Οικίας :</em>
        <input type="text" name="phone" size="20" onblur="isNumber(this);">
    </font></strong></td>
</tr><tr>
    <td align="left"><strong><font face="Georgia, Times New Roman, Times,
serif"><em>Email :</em>
        <input type="text" name="email" size="20">
    </font></strong></td>
</tr><tr>
    <!-- <td align="center"><strong><font face="Georgia, Times New Roman, Times,
serif"><em>Α.Μ :</em>
        <input type="text" name="am" size="20" onblur="isNumber(this);">
    </font></strong></td>
</tr>-->
<tr><td align="center"><strong><font face="Georgia, Times New Roman, Times, serif">
    <input type="submit" value="Εγγραφή" name="B62" >
        <input type="reset" value="Εκκαθάριση" name="B72">
    </font></strong></td>
</tr></form>

</table>&nbsp;</p>
<p align="center">&nbsp;</p>
<p>&nbsp;</p>
<p align="center"><strong><em><font size="4" face="Georgia, Times New Roman, Times,
serif"><a href="selida_diaxirisis.php">Επιστροφή Στην Σελίδα
Διαχείρισης</a></font></em></strong></p>
<strong><em><font size="4" face="Georgia, Times New Roman, Times, serif">
</center>
</font></em></strong>
</body>
</html>

```

Στο πρώτο κομμάτι κώδικα το οποίο περικλείεται από τα <? ?> τα οποία μας παραθέτουν σε ύπαρξη κώδικα php, εκτελούμε την σύνδεση με την βάση δεδομένων μας και θέτουμε την κωδικοποίηση των χαρακτήρων στα ελληνικά τα οποία και χρειαζόμαστε.

Έπειτα στο κομμάτι head του κώδικα έχουμε βάλει κάποια κομμάτια κώδικα JavaScript. Το πρώτο κομμάτι κώδικα αφορά την δημιουργία και εμφάνιση του ψηφιακού ρολογιού το οποίο βρίσκεται στο πάνω μέρος της κάθε σελίδας μας. Το άλλο κομμάτι κώδικα αφορά τον έλεγχο των αριθμητικών πεδίων που συμπληρώνουμε στις φόρμες μας, δηλαδή όποτε βάλουμε γράμμα σε ένα πεδίο το οποίο απαιτεί αριθμό, τότε μας εμφανίζει κάποιο μήνυμα και δεν μας αφήνει να προχωρήσουμε παρακάτω στη συμπλήρωση των υπολοίπων πεδίων.

Στο κομμάτι του body αρχικά βάζουμε το όνομα της καρτέλας μας πάνω αριστερά και έπειτα δημιουργούμε την φόρμα μας την οποία και αποστέλλουμε όταν πατάμε το κουμπί της υποβολής στο αρχείο που θα κάνει την διεργασία και τον έλεγχο εγκυρότητας της φόρμας μας. Κάνουμε την στοίχιση των κουμπιών μέσα στον πίνακα και διαμορφώνουμε το πλάτος και μήκος του. Έπειτα δημιουργούμε τα κουμπιά και τους δίνουμε τα ονόματα τα οποία εμφανίζονται καθώς και τα ονόματα τα οποία χρησιμοποιούμε στην επόμενη σελίδα για να κάνουμε ανάκτηση δεδομένων που έχουμε εισάγει. Τέλος φτιάχνουμε τα links τα οποία εμφανίζονται στο τέλος κάθε σελίδας τα οποία είναι υπογραμμισμένα και έχουν μπλε χρώμα.

Κώδικας 'eggrafi_foititi_insert.php'

```
<?php
$conn = mysql_connect("localhost","root","");
mysql_query("set names 'greek'", $conn);
$db = mysql_select_db('grammateia',$conn);

?>
```

```
<!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>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
</head>
```

```
<body>
<p>
<?
$name=$_POST['name'];
$surname=$_POST['surname'];
$f_name=$_POST['f_name'];
$m_name=$_POST['m_name'];
$row=$_POST['row'];
$mo=$_POST['mo'];
$id=$_POST['id'];
$p_address=$_POST['p_address'];
$address=$_POST['address'];
$p_phone=$_POST['p_phone'];
$phone=$_POST['phone'];
$email=$_POST['email'];
//$am=$_POST['am'];

if (!$conn)
{
die('Could not connect: ' . mysql_error());
}
```

```
$sql="INSERT INTO foititis
(onoma,eponumo,onoma_patros,onoma_mitros,seira_epitixias,M_O,A_D_T,diefthinsi_monimis_kat
oikias,diefthinsi_topikis_katoikias,tilefono_monimis_oikias,tilefono_topikis_oikias,email) VALUES
('$name','$surname','$f_name','$m_name','$row','$mo','$id','$p_address','$address','$p_phone','$phon
e','$email)";
```

```
if (!mysql_query($sql,$conn))
{
die('Error: ' . mysql_error());
}
echo "records added";
```

```
mysql_close($conn)
```

```
?>
```

```
</p>
```

```
<p><a href="selida_diaxirisis.php">Επιστροφή Στην Σελίδα Διαχείρισης</a></p>
```

```
<p><a href="eggrafi_foititi.php">Επιστροφή Στην Προηγούμενη Σελίδα</a></p>
```

```
</center>
```

```
</body>
```

```
</html>
```

Στο πρώτο κομμάτι κώδικα php εκτελούμε πάλι την σύνδεση με την βάση μας και στο κομμάτι head απλά έχουμε τα στοιχεία τα οποία πιστοποιούν την σελίδα μας σύμφωνα με τα standard του οργανισμού W3C. Έπειτα στο υπόλοιπο κομμάτι κώδικα php τοποθετούμε τα στοιχεία τα οποία έβαλε ο χρήστης στην φόρμα και τα αποθηκεύουμε σε διάφορες μεταβλητές με όνομα τέτοιο που να ξέρουμε τι είναι η κάθε μία. Κατόπιν με μία if εκτελούμε ένα έλεγχο για το αν είναι σωστή η σύνδεση μας στην βάση αλλιώς μας βγάζει λάθος. Μετά με μία εντολή insert αποθηκεύουμε στην βάση μας τα δεδομένα που έχει εισάγει ο χρήστης. Εκτελούμε κάποιους ελέγχους για την αποθήκευση των δεδομένων μας και αν είναι όλα εντάξει, τότε εμφανίζεται και το κατάλληλο μήνυμα. Κλείνουμε την σύνδεση μας και φτιάχνουμε και τα Links.

A.2.5 Σελίδα Εγγραφής Καθηγητή

Κώδικας 'eggragi_kathigiti.php'

```
<head>
```

```
<META HTTP-EQUIV="Content-Type" content="text/html; charset=iso-8859-7">
```

```
<title>ΔΙΑΓΡΑΦΗ ΜΑΘΗΜΑΤΟΣ</title>
```

```
</head>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
```

```
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<html>
```

```
<html>
```

```
<head>
```

```
<META HTTP-EQUIV="Content-Type" content="text/html; charset= iso-8859-7">
```

```

<title>AeaanaoS IaeSiaoio</title>
<meta name="GENERATOR" content="Arachnophilia 4.0">
<meta name="FORMATTER" content="Arachnophilia 4.0">
</head>

<body></head><html xmlns="http://www.w3.org/1999/xhtml">
<div align="center"><body>
<div align="left">
  <script type="text/javascript">
function startTime()
{var today =new Date();
var h=today.getHours();
var m=today.getMinutes();
var s=today.getSeconds();
m=checkTime(m);
s=checkTime(s);
document.getElementById('txt').innerHTML=h+":"+m+": "+s;
t=setTimeout('startTime()',500);
}
function checkTime(i)
{if (i<10)
{i="0"+i;
} return i;
}

function CheckRequiredFields() {

var errorMessage = new String();

if(WithoutContent(document.dmathima.name.value))
  { errorMessage += "\n\nΣυμπληρώστε το πεδίο του Ονόματος Μαθήματος."; }

if(WithoutContent(document.dmathima.id.value))
  { errorMessage += "\n\nΣυμπληρώστε το πεδίο του Κωδικού Μαθήματος."; }

if(errorMessage.length > 0) {
  alert('NOTE:' + errorMessage);
  return false;
}
return true;
}

function WithoutContent(ss) {
if(ss.length > 0) { return false; }
return true;
}

</script>
</head>
<body onload="startTime()">
</div>

```

```

<div id ="txt"></div>

<h2 align="center"><em><font face="Georgia, Times New Roman, Times,
serif"><strong>Σελίδα Διαγραφής Μαθήματος </strong></font></em></h2>
<p align="center">&nbsp;</p>
<p align="center">&nbsp;</p>
<p align="center">&nbsp;</p>
<p align="center"><table border="10" align="center" TD width="30%" height="100">
<form method="POST" name="dmathima" action="diagrafi_mathimatos_insert.php"
onSubmit="return CheckRequiredFields()">
<tr>
<th><strong><font face="Georgia, Times New Roman, Times, serif"><em>Διαγραφή
Μαθήματος </em></font></strong></th>
</tr>
<tr><td align="left"><strong><font face="Georgia, Times New Roman, Times, serif">ID
Μαθήματος:
<input type="text" name="id" size="20">
</font></strong></td></tr>
<tr><td align="left"><strong><font face="Georgia, Times New Roman, Times,
serif">Τίτλος Μαθήματος<em> :</em>
<input type="text" name="name" size="20">
</font></strong></td></tr>
<tr><td align="center"><strong><font face="Georgia, Times New Roman, Times, serif">
<input type="submit" value="Διαγραφή" name="B62">
<input type="reset" value="Εκκαθάριση" name="B72">
</font></strong></td>
</tr>
</form>
</table>
</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p align="center"><font color="#00CC99" size="5" face="Arial, Helvetica, sans-serif">
</font><font color="#00CC99" size="5" face="Arial, Helvetica, sans-serif">

</font></p>
<p align="center"><strong><font size="4" face="Georgia, Times New Roman, Times,
serif"><em><a href="selida_diaxisis.php">Επιστροφή Στην Σελίδα
Διαχείρισης</a></em></font></strong></p>
</body>
</html>

```

Και εδώ όπως και στο πρώτο κομμάτι κώδικα που παραθέσαμε (αυτό δηλαδή της εγγραφής φοιτητή), έχουμε στην αρχή τα κομμάτια κώδικα JavaScript τα οποία αφορούν την εμφάνιση και λειτουργία του ρολογιού της εφαρμογή μας και τον έλεγχο των πεδίων που απαιτούν αριθμητική εισαγωγή δεδομένων. Έπειτα στο body φτιάχνουμε την φόρμα μας και δίνουμε τον προορισμό των δεδομένων προς επεξεργασία, ο οποίος είναι μία άλλη σελίδα της οποίας τον κώδικα θα περιγράψουμε παρακάτω. Φτιάχνουμε τον πίνακα με τα κουμπιά του ,τα στοιχίζουμε και τα διαμορφώνουμε κατάλληλα. Τέλος φτιάχνουμε τα links τα οποία βρίσκονται στο κάτω μέρος της σελίδας μας.

Κώδικας 'eggragi_kathigiti_insert.php'

```
<?php
$conn = mysql_connect("localhost","root","");
mysql_query("set names 'greek'", $conn);
$db = mysql_select_db('grammateia',$conn);

?>
```

```
<!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>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Εισαγωγή Καθηγητή</title>
</head>
```

```
<body>
<p>
<?
$name=$_POST['name'];
$surname=$_POST['surname'];
$grade=$_POST['grade'];
$id=$_POST['id'];
$address=$_POST['address'];
$phone=$_POST['phone'];
$email=$_POST['email'];
```

```
if (!$conn)
{
die('Could not connect: ' . mysql_error());
}
```

```
$sql="INSERT INTO kathigitis
(onoma,eponumo,tilefono_oikias,bathmida,A_D_T,diefthinsi_grafeiou,email) VALUES
('$name','$surname','$phone','$grade','$id','$address','$email)";
```

```
if (!mysql_query($sql,$conn))
{
die('Error: ' . mysql_error());
}
echo "records added";
```

```
mysql_close($conn)
?>
```

```
</p>
<p><a href="selida_diaxirisis.php">Επιστροφή Στην Σελίδα Διαχείρισης</a></p>
<p><a href="eggrafi_kathigiti.php">Επιστροφή Στην Προηγούμενη Σελίδα</a></p>
```

```
</center>
</body>
</html>
```

Σε αυτό το κομμάτι κώδικα αρχικά πάλι στο πρώτο κομμάτι της php συνδεόμαστε με την βάση δεδομένων μας και καθορίζουμε την κωδικοποίηση των χαρακτήρων. Μετά ακολουθεί το καθορισμένο κομμάτι το οποίο αναφέρεται σε αναγνώριση της σελίδας μας από τον browser και το οποίο καθορίζεται από τον οργανισμό W3C. Έπειτα αποθηκεύουμε σε τοπικές μεταβλητές τα δεδομένα τα οποία πήραμε με την μέθοδο post από την φόρμα του προηγούμενου κώδικα. Μετά εκτελούμε ένα έλεγχο για την σύνδεση με την βάση δεδομένων μας και αν είναι αποτυχημένος τότε εμφανίζει σφάλμα στην οθόνη μας. Μετά με ένα απλό ερώτημα sql εισάγουμε τα δεδομένα μας και τα αποθηκεύουμε στον πίνακα που θέλουμε και στις μεταβλητές που επιθυμούμε. Κατόπιν εκτελούμε κάποιους ελέγχους και αν ισχύει η σύνδεση με την βάση μας ,τότε εμφανίζεται και το κατάλληλο μήνυμα και εκτελείται η εγγραφή των δεδομένων στη βάση μας. Κλείνουμε την σύνδεση με την βάση μας και φτιάχνουμε τα links τα οποία εμφανίζονται πάνω αριστερά.

A.2.6 Σελίδα Εισαγωγής Μαθήματος

Κώδικας 'eisagwi_mathimatos.php'

```
<?php
$conn=mysql_connect("localhost","root",""); // (host, username, password)
mysql_query("set names 'greek'", $conn);
//mysql_query("set character set 'greek'", $conn);
$db = mysql_select_db('grammateia',$conn);
?>

<head>
<META HTTP-EQUIV="Content-Type" content="text/html; charset=iso-8859-7">
<title>ΕΙΣΑΓΩΓΗ ΜΑΘΗΜΑΤΟΣ</title>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
</head><body><html xmlns="http://www.w3.org/1999/xhtml">
<div align="center"><body>
<div align="left">

<script type ="text/javascript" >
function startTime()
{var today =new Date();
var h=today.getHours();
var m=today.getMinutes();
```

```

var s=today.getSeconds();
m=checkTime(m);
s=checkTime(s);
document.getElementById('txt').innerHTML=h+": "+m+": "+s;
t=setTimeout('startTime()',500);
}
function checkTime(i)
{ if (i<10)
{i="0"+i;
} return i;
} </script>

```

```

<script type="text/javascript">
function isNumber(obj)
{
if(document.getElementById('err'))
{
obj.parentNode.removeChild(document.getElementById('err'));
}
setTimeout(function(){
if(isNaN(obj.value))
{
obj.select();
var p=document.createElement('p');
p.style.color="red";
p.style.fontStyle='italic';
p.style.margin="0px";
p.style.padding="0px";
p.id='err';
p.appendChild(document.createTextNode('Input must be numeric'));
obj.parentNode.appendChild(p);
}
},1);
}

```

```

function CheckRequiredFields() {

var errormessage = new String();

if(WithoutContent(document.mathima.name.value))
    { errormessage += "\n\nΣυμπληρώστε το πεδίο του Μαθήματος."; }

if(WithoutContent(document.mathima.semester.value))
    { errormessage += "\n\nΣυμπληρώστε το πεδίο του Εξαμήνου."; }

if(WithoutContent(document.mathima.type.value))
    { errormessage += "\n\nΣυμπληρώστε το πεδίο του τύπου Μαθήματος."; }

if(errormessage.length > 0) {
    alert('NOTE:' + errormessage);
    return false;
}
}

```



```

return true;
}

function WithoutContent(ss) {
if(ss.length > 0) { return false; }
return true;
}

</script>

</head>
<body onload="startTime()">
</div>
<div id ="txt"></div>

<h2 align="center"><em><font face="Georgia, Times New Roman, Times,
serif"><strong>Σελίδα Εισαγωγής Μαθήματος </strong></font></em></h2>
<p align="center">&nbsp;</p>
<p align="center">&nbsp;</p>
<p align="center"><table border="10" align="center" TD width="30%" height="100">
<form method="POST" name="mathima" action="eisagwgi_mathimatos_insert.php"
onSubmit="return CheckRequiredFields()">
<tr>
<th><strong><font face="Georgia, Times New Roman, Times, serif"><em>Εισαγωγή
Μαθήματος </em></font></strong></th>
</tr>
<tr><td align="center"><strong><font face="Georgia, Times New Roman, Times,
serif">ID :
<input type="text" name="id" size="20" onblur="isNumber(this);">
</font></strong></td></tr>
<tr><td align="center"><strong><font face="Georgia, Times New Roman, Times,
serif">Τίτλος <em> :</em>
<input type="text" name="name" size="20">
</font></strong></td></tr><tr>
<td align="center"><strong><font face="Georgia, Times New Roman, Times,
serif">Ωρες :<em>
<input type="text" name="hours" size="20" onblur="isNumber(this);">
</em></font></strong></td>
</tr><tr>
<td align="center"><strong><font face="Georgia, Times New Roman, Times,
serif">Εξάμηνο :<em>
<input type="text" name="semester" size="20">
</em></font></strong></td>
</tr><tr>
<td align="center"><strong><font face="Georgia, Times New Roman, Times,
serif">Τύπος :<em>
<input type="text" name="type" size="20">
</em></font></strong></td>
</tr><tr>
<td align="center"><strong><font face="Georgia, Times New Roman, Times,
serif">Δ.Μ. :
<em>
<input type="text" name="units" size="20" onblur="isNumber(this);">

```

```

        </em></font></strong></td>
    </tr>
    <tr><td align="center"><strong><font face="Georgia, Times New Roman, Times, serif">
        <input type="submit" value="Εισαγωγή" name="B62">
        <input type="reset" value="Εκκαθάριση" name="B72">
        </font></strong></td>
    </tr>

</form>
</table>&nbsp;</p>
<p align="center">&nbsp;</p>
<p align="center">&nbsp;</p>
<p align="center">&nbsp;</p>
<p align="center"><em><strong><font size="4" face="Georgia, Times New Roman, Times, serif"><a href="selida_diaxirisis.php">Επιστροφή Στην Σελίδα Διαχείρισης</a></font></strong></em></p>
<p align="center">&nbsp;</p>

</body>
</html>

```

Και εδώ όπως και στο πρώτο κομμάτι κώδικα που παραθέσαμε (αυτό δηλαδή της εγγραφής φοιτητή), έχουμε στην αρχή τα κομμάτια κώδικα JavaScript τα οποία αφορούν την εμφάνιση και λειτουργία του ρολογιού της εφαρμογή μας και τον έλεγχο των πεδίων που απαιτούν αριθμητική εισαγωγή δεδομένων. Έπειτα στο body φτιάχνουμε την φόρμα μας με τα πεδία και δίνουμε τον προορισμό των δεδομένων προς επεξεργασία, ο οποίος είναι μία άλλη σελίδα της οποίας τον κώδικα θα περιγράψουμε παρακάτω. Φτιάχνουμε τον πίνακα με τα κουμπιά του ,τα στοιχίζουμε και τα διαμορφώνουμε κατάλληλα. Στα πεδία που χρειάζεται αριθμητική εισαγωγή δεδομένων καλούμε την συνάρτηση της JavaScript για να εκτελέσει τον έλεγχο. Στα κουμπιά τα οποία είναι για την υποβολή και την εκκαθάριση δίνουμε τον τύπο του κάθε κουμπιού ,δηλαδή την ενέργεια που θα εκτελέσει. Τέλος φτιάχνουμε τα links τα οποία βρίσκονται στο κάτω μέρος της σελίδας μας και φυσικά ορίζουμε και τον τύπο γραμματοσειράς.

Κώδικας 'eisagwi_mathimatos_insert.php'

```

<?php
$conn = mysql_connect("localhost","root","");
mysql_query("set names 'greek'", $conn);
$db = mysql_select_db('grammateia',$conn);

?>

<!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>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

```

```

<title>Εισαγωγή Μαθήματος</title>
</head>

<body>

<p>
  <?
  $id=$_POST['id'];
  $name=$_POST['name'];
  $hours=$_POST['hours'];
  $semester=$_POST['semester'];
  $type=$_POST['type'];
  $units=$_POST['units'];

  if (!$conn)
  {
    die('Could not connect: ' . mysql_error());
  }

  $sql="INSERT INTO mathima (onoma,kodikos_m,examino,D_M,typos_mathimatos,wres)
VALUES ('$name','$id','$semester','$units','$type','$hours)";

  if (!mysql_query($sql,$conn))
  {
    die('Error: ' . mysql_error());
  }
  echo "records added";

  mysql_close($conn)
?>

</p>
<p><a href="selida_diaxirisis.php">Επιστροφή Στην Σελίδα Διαχείρισης</a></p>
<p><a href="eisagwgi_mathimatos.php">Επιστροφή Στην Προηγούμενη Σελίδα</a></p>
</center>
</body>
</html>

```

Το πρώτο κομμάτι κώδικα το οποίο περιλαμβάνεται από τα tags της php είναι για την σύνδεση της σελίδας μας με την βάση δεδομένων μας. Δίνουμε δηλαδή το όνομα του server μας και το όνομα της βάσης μας και θέτουμε την κωδικοποίηση των χαρακτήρων στα ελληνικά. Έπειτα υπάρχει ένα κομμάτι το οποίο το προσθέτει μόνο του το πρόγραμμα Dreamweaver το οποίο είναι για την αναγνώριση του εγγράφου από τους browsers και ελέγχεται αν πληροί όλα τα standards που έχουν τεθεί. Στο κομμάτι head δίνουμε την κωδικοποίηση και τον τίτλο που θα φέρει στην καρτέλα του browser η σελίδα μας.

Στο κομμάτι του body το οποίο μας ενδιαφέρει περισσότερο καθώς εκεί τοποθετούμε το κομμάτι του κώδικα μας το οποίο κάνει τις κυριότερες εργασίες στην σελίδα μας, αρχικά τοποθετούμε σε κάποιες μεταβλητές τα δεδομένα τα οποία έγραψε ο χρήστης στην φόρμα υποβολής στοιχείων. Μετά εκτελούμε έναν έλεγχο για την σύνδεση με την βάση μας και αν βρεθεί εντάξει τότε προχωράμε, αλλιώς μας εμφανίζει μήνυμα λάθους σύνδεσης και πρέπει να ελέγξουμε τον κώδικα για τυχόν λάθη.

Παρακάτω με ένα sql ερώτημα βάζουμε τα δεδομένα που έχουμε αποθηκεύσει στις παραπάνω μεταβλητές, στην βάση μας.. Με ένα έλεγχο πάλι αν υπάρχει λάθος στο κώδικα μας εμφανίζει σφάλμα και δεν γίνεται εισαγωγή στοιχείων ,αλλιώς εμφανίζεται ένα μήνυμα ότι οι εγγραφές είναι επιτυχείς. Κλείνουμε την σύνδεση την οποία ανοίξαμε στην αρχή του κώδικα με την βάση μας και αφού κλείσουμε και τα tags της php δημιουργούμε τα γνωστά μας Links τα οποία είτε μας πηγαίνουν στην αρχική σελίδα είτε μια σελίδα πίσω στην εκάστοτε φόρμα υποβολής στοιχείων.

A.2.7 Σελίδα Διαγραφής Φοιτητή

Κώδικας 'diagrafi_foititi.php'

```
<?php
$conn=mysql_connect("localhost","root",""); // (host, username, password)
mysql_query("set names 'greek'", $conn);
//mysql_query("set character set 'greek'", $conn);
$db = mysql_select_db('grammateia',$conn);

?>

<head>
<META HTTP-EQUIV="Content-Type" content="text/html; charset=iso-8859-7">
<title>ΔΙΑΓΡΑΦΗ ΦΟΙΤΗΤΗ</title>
</head>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>

<body></head><html xmlns="http://www.w3.org/1999/xhtml">
<div align="center"><body>
<div align="left">
  <script type ="text/javascript" >
function startTime()
{var today =new Date();
var h=today.getHours();
var m=today.getMinutes();
var s=today.getSeconds();
m=checkTime(m);
s=checkTime(s);
document.getElementById('txt').innerHTML=h+": "+m+": "+s;
t=setTimeout('startTime()',500);
}
function checkTime(i)
{if (i<10)
{i="0"+i;
} return i;
} </script>
```

```

<script type="text/javascript">
function isNumber(obj)
{
if(document.getElementById('err'))
{
obj.parentNode.removeChild(document.getElementById('err'));
}
setTimeout(function(){
if(isNaN(obj.value))
{
obj.select();
var p=document.createElement('p');
p.style.color="red";
p.style.fontStyle='italic';
p.style.margin="0px";
p.style.padding="0px";
p.id='err';
p.appendChild(document.createTextNode('Input must be numeric'));
obj.parentNode.appendChild(p);
}
},1);
}

function CheckRequiredFields() {

var errormessage = new String();

if(WithoutContent(document.dfoititis.name.value))
    { errormessage += "\n\nΣυμπληρώστε το πεδίο του Ονόματος."; }

if(WithoutContent(document.dfoititis.surname.value))
    { errormessage += "\n\nΣυμπληρώστε το πεδίο του Επιθέτου."; }

if(WithoutContent(document.dfoititis.am.value))
    { errormessage += "\n\nΣυμπληρώστε το πεδίο του AM."; }

if(errormessage.length > 0) {
    alert('NOTE:' + errormessage);
    return false;
}
return true;
}

function WithoutContent(ss) {
if(ss.length > 0) { return false; }
return true;
}

</script>

</head>
<body onload="startTime()">

```

```

</div>
<div id ="txt"></div>
<div align="center">
  <h2 align="center"><em><font face="Georgia, Times New Roman, Times,
serif"><strong>Σελίδα Διαγραφής Φοιτητή</strong></font></em></h2>
  <p align="center">&nbsp;</p>
  <p align="center">&nbsp;</p>
</div>
<p align="center"><table border="10" align="center" TD width="30%" height="100">
  <form method="POST" name="dfoititis" action="diagrafi_foititi_insert.php"
onSubmit="return CheckRequiredFields()" >
  <tr>
    <th><strong><font face="Georgia, Times New Roman, Times, serif"><em><font
face="Georgia, Times New Roman, Times, serif"><strong>Διαγραφή
Φοιτητή</strong></font></em></font></strong></th>
  </tr>
  <tr><td align="left"><strong><font face="Georgia, Times New Roman, Times,
serif">Όνομα :
  <input type="text" name="name" size="20">
  </font></strong></td>
  </tr>
  <tr><td align="left"><strong><font face="Georgia, Times New Roman, Times,
serif">Επίθετο :
  <input type="text" name="surname" size="20">
  </font></strong></td>
  </tr> <tr>
    <td align="left"><strong><font face="Georgia, Times New Roman, Times, serif">A.M.
  <input type="text" name="am" size="20" onblur="isNumber(this);">
  </font></strong></td>
  </tr>
  <tr><td align="center"><strong><font face="Georgia, Times New Roman, Times, serif">
  <input type="submit" value="Διαγραφή" name="B62">
  <input type="reset" value="Εκκαθάριση" name="B72">
  </font></strong></td>
  </tr>

</form>
</table>
</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p align="center"><strong><em><font size="4" face="Georgia, Times New Roman, Times,
serif"><a href="selida_diaxirisis.php">Επιστροφή Στην Σελίδα
Διαχείρισης</a></font></em></strong></p>
<strong><em><font size="4" face="Georgia, Times New Roman, Times, serif">
</center>
</font></em></strong>
</body></html>

```

Και εδώ όπως και στο πρώτο κομμάτι κώδικα που παραθέσαμε (αυτό δηλαδή της εγγραφής φοιτητή), έχουμε στην αρχή τα κομμάτια κώδικα JavaScript τα οποία αφορούν την εμφάνιση και λειτουργία του ρολογιού της εφαρμογή μας και τον έλεγχο των πεδίων που απαιτούν αριθμητική εισαγωγή δεδομένων. Έπειτα στο body φτιάχνουμε την φόρμα μας με τα πεδία και δίνουμε τον προορισμό των δεδομένων προς επεξεργασία, ο οποίος είναι μία άλλη σελίδα της οποίας τον κώδικα θα περιγράψουμε παρακάτω. Φτιάχνουμε τον πίνακα με τα κουμπιά του ,τα στοιχίζουμε και τα διαμορφώνουμε κατάλληλα. Στα πεδία που χρειάζεται αριθμητική εισαγωγή δεδομένων καλούμε την συνάρτηση της JavaScript για να εκτελέσει τον έλεγχο. Στα κουμπιά τα οποία είναι για την υποβολή και την εκκαθάριση δίνουμε τον τύπο του κάθε κουμπιού ,δηλαδή την ενέργεια που θα εκτελέσει. Τέλος φτιάχνουμε τα links τα οποία βρίσκονται στο κάτω μέρος της σελίδας μας και φυσικά ορίζουμε και τον τύπο γραμματοσειράς.

Κώδικας 'diagrafi_foititi_insert.php'

```
<!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>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Διαγραφή Φοιτητή</title>
</head>

<?

error_reporting(E_ALL & ~E_NOTICE);
$name=$_POST['name'];
$surname=$_POST['surname'];
$am=$_POST['am'];

$conn=mysql_connect("localhost","root","");
mysql_query("SET NAMES 'utf8'", $conn);
mysql_query("set character set 'greek'", $conn);
mysql_select_db("grammateia",$conn);

$qer=mysql_query("select A_M,onoma,eponumo FROM foititis where A_M=$am and
onoma=$name and eponumo=$surname");

if($am=="" || $surname==""){
echo "Εισάγετε όλα τα στοιχεία";
return false;
}
else{

$query =mysql_query("delete from foititis where A_M='$am' and eponumo='$surname'
");
```

```

echo "Η Διαγραφή Ολοκληρώθηκε";

}

?>

<p><a href="selida_diaxirisis.php">Επιστροφή Στην Σελίδα Διαχείρισης</a></p>
<p><a href="diagrafi-foititi.php">Επιστροφή Στην Προηγούμενη Σελίδα</a></p>
</center>

<body>
</body>
</html>

```

Το πρώτο κομμάτι κώδικα το οποίο περιλαμβάνεται από τα tags της php είναι για την σύνδεση της σελίδας μας με την βάση δεδομένων μας. Δίνουμε δηλαδή το όνομα του server μας και το όνομα της βάσης μας και θέτουμε την κωδικοποίηση των χαρακτήρων στα ελληνικά. Έπειτα υπάρχει ένα κομμάτι το οποίο το προσθέτει μόνο του το πρόγραμμα Dreamweaver το οποίο είναι για την αναγνώριση του εγγράφου από τους browsers και ελέγχεται αν πληροί όλα τα standards που έχουν τεθεί. Στο κομμάτι head δίνουμε την κωδικοποίηση και τον τίτλο που θα φέρει στην καρτέλα του browser η σελίδα μας. Στο κομμάτι του body το οποίο μας ενδιαφέρει περισσότερο καθώς εκεί τοποθετούμε το κομμάτι του κώδικα μας το οποίο κάνει τις κυριότερες εργασίες στην σελίδα μας, αρχικά τοποθετούμε σε κάποιες μεταβλητές τα δεδομένα τα οποία έγραψε ο χρήστης στην φόρμα υποβολής στοιχείων.

Μετά εκτελούμε έναν έλεγχο για την σύνδεση με την βάση μας και αν βρεθεί εντάξει τότε προχωράμε, αλλιώς μας εμφανίζει μήνυμα λάθους σύνδεσης και πρέπει να ελέγξουμε τον κώδικα για τυχόν λάθη. Παρακάτω με μία if ελέγχουμε αν είναι και τα δύο πεδία, αυτά του AM και του επωνύμου κενά και αν είναι τότε δεν προχωρά η διαγραφή αλλιώς με ένα sql ερώτημα (delete) διαγράφουμε τα δεδομένα που έχουμε αποθηκεύσει στην βάση μας αφού πρώτα τα έχουμε επιλέξει πιο πάνω με την εντολή select. Με ένα έλεγχο πάλι αν υπάρχει λάθος στο κώδικα μας εμφανίζει σφάλμα και δεν γίνεται εισαγωγή στοιχείων, αλλιώς εμφανίζεται ένα μήνυμα ότι οι εγγραφές είναι επιτυχείς. Κλείνουμε την σύνδεση την οποία ανοίξαμε στην αρχή του κώδικα με την βάση μας και αφού κλείσουμε και τα tags της php δημιουργούμε τα γνωστά μας Links τα οποία είτε μας πηγαίνουν στην αρχική σελίδα είτε μια σελίδα πίσω στην εκάστοτε φόρμα υποβολής στοιχείων.

A.2.8 Σελίδα Διαγραφής Καθηγητή

Κώδικας 'diagrafi_kathigiti.php'

```

<head>
<META HTTP-EQUIV="Content-Type" content="text/html; charset=iso-8859-7">
<title>ΔΙΑΓΡΑΦΗ ΚΑΘΗΓΗΤΗ</title>
</head>
<body></head><html xmlns="http://www.w3.org/1999/xhtml">
<div align="center"><body>
<div align="left">
<script type="text/javascript">

```



```

function startTime()
{ var today =new Date();
var h=today.getHours();
var m=today.getMinutes();
var s=today.getSeconds();
m=checkTime(m);
s=checkTime(s);
document.getElementById('txt').innerHTML=h+":"+m+": "+s;
t=setTimeout('startTime()',500);
}
function checkTime(i)
{if (i<10)
{i="0"+i;
} return i;
}

function CheckRequiredFields() {

var errormessage = new String();

if(WithoutContent(document.dkathigitis.name.value))
    { errormessage += "\n\nΣυμπληρώστε το πεδίο του Ονόματος."; }

if(WithoutContent(document.dkathigitis.surname.value))
    { errormessage += "\n\nΣυμπληρώστε το πεδίο του Επιθέτου."; }

if(WithoutContent(document.dkathigitis.id.value))
    { errormessage += "\n\nΣυμπληρώστε το πεδίο του ΑΜ."; }

if(errormessage.length > 0) {
    alert('NOTE:' + errormessage);
    return false;
}
return true;
}

function WithoutContent(ss) {
if(ss.length > 0) { return false; }
return true;
}

```

```

</script>
</head>
<body onload="startTime()">
</div>
<div id ="txt"></div>

```

```

<h2 align="center"><strong><em><font face="Georgia, Times New Roman, Times,
serif">Σελίδα Διαγραφής Καθηγητή</font></em></strong></h2>
<p align="center">&nbsp;</p>
<p align="center">&nbsp;</p>
<p align="center">&nbsp;</p>
<table border="10" align="center" TD width="30%" height="100">

```

```

        <form method="POST" name="dkathigitis" action="diagrafi_kathigiti_insert.php"
onSubmit="return CheckRequiredFields()">
        <tr>
        <th><font face="Georgia, Times New Roman, Times, serif"><strong><em>Διαγραφή
Καθηγητή</em></strong></font></th>
        </tr>
        <tr>
        <td align="left"><font face="Georgia, Times New Roman, Times,
serif"><strong>Όνομα :
        <input type="text" name="name" size="20">
        </strong></font></td>
        </tr>
        <tr>
        <td align="left"><font face="Georgia, Times New Roman, Times,
serif"><strong>Επίθετο:
        <input type="text" name="surname" size="20">
        </strong></font></td>
        </tr>
        <tr>
        <td align="left"><font face="Georgia, Times New Roman, Times,
serif"><strong>Α.Δ.Τ.:
        <input type="text" name="id" size="20">
        </strong></font></td>
        </tr>
        <tr>
        <td align="center"><font face="Georgia, Times New Roman, Times,
serif"><strong><em>
        <input type="submit" value="Διαγραφή" name="B62">
        <input type="reset" value="Εκκαθάριση" name="B72">
        </em></strong></font></td>
        </tr>
</form>
</table>
<p align="center">&nbsp;</p>
<p align="center">&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p align="center">&nbsp;</p>
<p align="center"><em><strong><font size="4" face="Georgia, Times New Roman, Times,
serif"><a href="selida_diaxirisis.php">Επιστροφή Στην Σελίδα
Διαχείρισης</a></font></strong></em></p>
<div align="center"></div></body></html>

```

Εδώ όπως και στο πρώτο κομμάτι κώδικα που παραθέσαμε (αυτό δηλαδή της εγγραφής φοιτητή), έχουμε στην αρχή τα κομμάτια κώδικα JavaScript τα οποία αφορούν την εμφάνιση και λειτουργία του ρολογιού της εφαρμογή μας και τον έλεγχο των πεδίων που απαιτούν αριθμητική εισαγωγή δεδομένων. Έπειτα στο body φτιάχνουμε την φόρμα μας με τα πεδία και δίνουμε τον προορισμό των δεδομένων προς επεξεργασία, ο οποίος είναι μία άλλη σελίδα της οποίας τον κώδικα θα περιγράψουμε παρακάτω. Φτιάχνουμε τον πίνακα με τα κουμπιά του ,τα στοιχίζουμε και τα διαμορφώνουμε κατάλληλα. Στα πεδία που χρειάζεται αριθμητική εισαγωγή δεδομένων καλούμε την συνάρτηση της JavaScript για να εκτελέσει τον έλεγχο. Στα κουμπιά τα οποία είναι για την υποβολή και την εκκαθάριση δίνουμε τον τύπο του κάθε κουμπιού ,δηλαδή την ενέργεια που θα εκτελέσει. Τέλος φτιάχνουμε τα links τα οποία βρίσκονται στο κάτω μέρος της σελίδας μας και φυσικά ορίζουμε και τον τύπο γραμματοσειράς.

Κώδικας 'diagrafi_kathigiti_insert.php'

```
<!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>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Διαγραφή Καθηγητή</title>
</head>

<?

error_reporting(E_ALL & ~E_NOTICE);
$name=$_POST['name'];
$id=$_POST['id'];
$surname=$_POST['surname'];

$conn=mysql_connect("localhost","root","");
mysql_query("set names 'utf8'", $conn);
mysql_query("set character set 'greek'", $conn);
mysql_select_db("grammateia",$conn);

$query=mysql_query("SELECT A_D_T,onoma,eponumo FROM kathigitis WHERE
A_D_T=$id and onoma=$name and eponumo=$surname ");

if($surname=="" || $id==""){
echo "Εισάγετε όλα τα στοιχεία";
return false;
}
else{

$query =mysql_query("DELETE from kathigitis where onoma='$name' and
eponumo='$surname' ");
```

```

echo "Η Διαγραφή Ολοκληρώθηκε";

}

?>

<p><a href="selida_diaxirisis.php">Επιστροφή Στην Σελίδα Διαχείρισης</a></p>
<p><a href="diagrafi_kathigiti.php">Επιστροφή Στην Προηγούμενη Σελίδα</a></p>
</center>

<body>
</body>
</html>

```

Το πρώτο κομμάτι κώδικα το οποίο περιλαμβάνεται από τα tags της php είναι για την σύνδεση της σελίδας μας με την βάση δεδομένων μας. Δίνουμε δηλαδή το όνομα του server μας και το όνομα της βάσης μας και θέτουμε την κωδικοποίηση των χαρακτήρων στα ελληνικά. Έπειτα υπάρχει ένα κομμάτι το οποίο το προσθέτει μόνο του το πρόγραμμα Dreamweaver το οποίο είναι για την αναγνώριση του εγγράφου από τους browsers και ελέγχεται αν πληροί όλα τα standards που έχουν τεθεί. Στο κομμάτι head δίνουμε την κωδικοποίηση και τον τίτλο που θα φέρει στην καρτέλα του browser η σελίδα μας. Στο κομμάτι του body το οποίο μας ενδιαφέρει περισσότερο καθώς εκεί τοποθετούμε το κομμάτι του κώδικα μας το οποίο κάνει τις κυριότερες εργασίες στην σελίδα μας, αρχικά τοποθετούμε σε κάποιες μεταβλητές τα δεδομένα τα οποία έγραψε ο χρήστης στην φόρμα υποβολής στοιχείων. Μετά εκτελούμε έναν έλεγχο για την σύνδεση με την βάση μας και αν βρεθεί εντάξει τότε προχωράμε, αλλιώς μας εμφανίζει μήνυμα λάθους σύνδεσης και πρέπει να ελέγξουμε τον κώδικα για τυχόν λάθη. Παρακάτω με μία if ελέγχουμε αν είναι και τα δύο πεδία, αυτά του Α.Δ.Τ και του επωνύμου κενά και αν είναι τότε δεν προχωρά η διαγραφή αλλιώς με ένα sql ερώτημα (delete) διαγράφουμε τα δεδομένα που έχουμε αποθηκεύσει στην βάση μας αφού πρώτα τα έχουμε επιλέξει πιο πάνω με την εντολή select. Με ένα έλεγχο πάλι αν υπάρχει λάθος στο κώδικα μας εμφανίζει σφάλμα και δεν γίνεται εισαγωγή στοιχείων ,αλλιώς εμφανίζεται ένα μήνυμα ότι οι εγγραφές είναι επιτυχείς. Κλείνουμε την σύνδεση την οποία ανοίξαμε στην αρχή του κώδικα με την βάση μας και αφού κλείσουμε και τα tags της php δημιουργούμε τα γνωστά μας Links τα οποία είτε μας πηγαίνουν στην αρχική σελίδα είτε μια σελίδα πίσω στην εκάστοτε φόρμα υποβολής στοιχείων.

A.2.9 Σελίδα Διαγραφής Μαθήματος

Κώδικας 'diagrafi_mathimatos.php'

```

<head>
<META HTTP-EQUIV="Content-Type" content="text/html; charset=iso-8859-7">
<title>ΔΙΑΓΡΑΦΗ ΜΑΘΗΜΑΤΟΣ</title>
</head>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

```

```
<html>
```

```
<html>
```

```
<head>
```

```
<META HTTP-EQUIV="Content-Type" content="text/html; charset= iso-8859-7">
```

```
<title>AeaanaoS IaeSiaoio</title>
```

```
<meta name="GENERATOR" content="Arachnophilia 4.0">
```

```
<meta name="FORMATTER" content="Arachnophilia 4.0">
```

```
</head>
```

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

```
<div align="center"><body>
```

```
<div align="left">
```

```
<script type="text/javascript" >
```

```
function startTime()
```

```
{ var today =new Date();
```

```
var h=today.getHours();
```

```
var m=today.getMinutes();
```

```
var s=today.getSeconds();
```

```
m=checkTime(m);
```

```
s=checkTime(s);
```

```
document.getElementById('txt').innerHTML=h+":"+m+":"+s;
```

```
t=setTimeout('startTime()',500);
```

```
}
```

```
function checkTime(i)
```

```
{ if (i<10)
```

```
{ i="0"+i;
```

```
} return i;
```

```
}
```

```
function CheckRequiredFields() {
```

```
var errormessage = new String();
```

```
if(WithoutContent(document.dmathima.name.value))
```

```
{ errormessage += "\n\nΣυμπληρώστε το πεδίο του Ονόματος Μαθήματος."; }
```

```
if(WithoutContent(document.dmathima.id.value))
```

```
{ errormessage += "\n\nΣυμπληρώστε το πεδίο του Κωδικού Μαθήματος."; }
```

```
if(errormessage.length > 0) {
```

```
alert('NOTE:' + errormessage);
```

```
return false;
```

```
}
```

```
return true;
```

```
}
```

```
function WithoutContent(ss) {
```

```
if(ss.length > 0) { return false; }
```

```
return true;
```

```
}
```

```

</script>
</head>
<body onload="startTime()">
</div>
<div id ="txt"></div>

<h2 align="center"><em><font face="Georgia, Times New Roman, Times,
serif"><strong>Σελίδα Διαγραφής Μαθήματος </strong></font></em></h2>
<p align="center">&nbsp;</p>
<p align="center">&nbsp;</p>
<p align="center">&nbsp;</p>
<p align="center"><table border="10" align="center" TD width="30%" height="100">
<form method="POST" name="dmathima" action="diagrafi_mathimatos_insert.php"
onSubmit="return CheckRequiredFields()">
<tr>
<th><strong><font face="Georgia, Times New Roman, Times, serif"><em>Διαγραφή
Μαθήματος </em></font></strong></th>
<tr>
<tr><td align="left"><strong><font face="Georgia, Times New Roman, Times, serif">ID
Μαθήματος:
<input type="text" name="id" size="20">
</font></strong></td></tr>
<tr><td align="left"><strong><font face="Georgia, Times New Roman, Times,
serif">Τίτλος Μαθήματος<em> :</em>
<input type="text" name="name" size="20">
</font></strong></td></tr>
<tr><td align="center"><strong><font face="Georgia, Times New Roman, Times, serif">
<input type="submit" value="Διαγραφή" name="B62">
<input type="reset" value="Εκκαθάριση" name="B72">
</font></strong></td>
</tr>
</form>
</table>
</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p align="center"><font color="#00CC99" size="5" face="Arial, Helvetica, sans-serif">
</font><font color="#00CC99" size="5" face="Arial, Helvetica, sans-serif">

</font></p>
<p align="center"><strong><font size="4" face="Georgia, Times New Roman, Times,
serif"><em><a href="selida_diaxirisis.php">Επιστροφή Στην Σελίδα
Διαχείρισης</a></em></font></strong></p>
</body>

</html>

```

Στην συνέχεια όπως και στο πρώτο κομμάτι κώδικα που παραθέσαμε (αυτό δηλαδή της εγγραφής φοιτητή), έχουμε στην αρχή τα κομμάτια κώδικα JavaScript τα οποία αφορούν την εμφάνιση και λειτουργία του ρολογιού της εφαρμογή μας και τον έλεγχο των πεδίων που απαιτούν αριθμητική εισαγωγή δεδομένων. Έπειτα στο body φτιάχνουμε την φόρμα μας με τα πεδία και δίνουμε τον προορισμό των δεδομένων προς επεξεργασία, ο οποίος είναι μία άλλη σελίδα της οποίας τον κώδικα θα περιγράψουμε παρακάτω. Φτιάχνουμε τον πίνακα με τα κουμπιά του ,τα στοιχίζουμε και τα διαμορφώνουμε κατάλληλα. Στα πεδία που χρειάζεται αριθμητική εισαγωγή δεδομένων καλούμε την συνάρτηση της JavaScript για να εκτελέσει τον έλεγχο. Στα κουμπιά τα οποία είναι για την υποβολή και την εκκαθάριση δίνουμε τον τύπο του κάθε κουμπιού ,δηλαδή την ενέργεια που θα εκτελέσει. Τέλος φτιάχνουμε τα links τα οποία βρίσκονται στο κάτω μέρος της σελίδας μας και φυσικά ορίζουμε και τον τύπο γραμματοσειράς.

Κώδικας 'diagrafi_mathimatos_insert.php'

```
<!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>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Διαγραφή Μαθήματος</title>
</head>

<?

error_reporting(E_ALL & ~E_NOTICE);
$name=$_POST['name'];
$id=$_POST['id'];

$conn=mysql_connect("localhost","root","");
mysql_query("set names 'utf8'", $conn);
mysql_query("set character set 'utf8'", $conn);
mysql_select_db("grammateia",$conn);

$query=mysql_query("select kodikos_m,onoma FROM mathima where kodikos_m=$id and
name=$name ");

if($id=="" || $name==""){
echo "Εισάγετε όλα τα στοιχεία";
return false;
}
else{

$query =mysql_query("delete from mathima where kodikos_m='$id' and onoma='$name'
");
```

```

echo "Η Διαγραφή Ολοκληρώθηκε";

}

?>

<p><a href="selida_diaxirisis.php">Επιστροφή Στην Σελίδα Διαχείρισης</a></p>
<p><a href="diagrafi_mathimatos.php">Επιστροφή Στην Προηγούμενη Σελίδα</a></p>
</center>

<body>

</body>
</html>

```

Το πρώτο κομμάτι κώδικα το οποίο περιλαμβάνεται από τα tags της php είναι για την σύνδεση της σελίδας μας με την βάση δεδομένων μας. Δίνουμε δηλαδή το όνομα του server μας και το όνομα της βάσης μας και θέτουμε την κωδικοποίηση των χαρακτήρων στα ελληνικά. Έπειτα υπάρχει ένα κομμάτι το οποίο το προσθέτει μόνο του το πρόγραμμα Dreamweaver το οποίο είναι για την αναγνώριση του εγγράφου από τους browsers και ελέγχεται αν πληροί όλα τα standards που έχουν τεθεί. Στο κομμάτι head δίνουμε την κωδικοποίηση και τον τίτλο που θα φέρει στην καρτέλα του browser η σελίδα μας. Στο κομμάτι του body το οποίο μας ενδιαφέρει περισσότερο καθώς εκεί τοποθετούμε το κομμάτι του κώδικα μας το οποίο κάνει τις κυριότερες εργασίες στην σελίδα μας, αρχικά τοποθετούμε σε κάποιες μεταβλητές τα δεδομένα τα οποία έγραψε ο χρήστης στην φόρμα υποβολής στοιχείων. Μετά εκτελούμε έναν έλεγχο για την σύνδεση με την βάση μας και αν βρεθεί εντάξει τότε προχωράμε, αλλιώς μας εμφανίζει μήνυμα λάθους σύνδεσης και πρέπει να ελέγξουμε τον κώδικα για τυχόν λάθη.

Παρακάτω με μία if ελέγχουμε αν είναι και τα δύο πεδία, αυτά του Α.Δ.Τ και του ονόματος κενά και αν είναι τότε δεν προχωρά η διαγραφή αλλιώς με ένα sql ερώτημα (delete) διαγράφουμε τα δεδομένα που έχουμε αποθηκεύσει στην βάση μας αφού πρώτα τα έχουμε επιλέξει πιο πάνω με την εντολή select. Με ένα έλεγχο πάλι αν υπάρχει λάθος στο κώδικα μας εμφανίζει σφάλμα και δεν γίνεται εισαγωγή στοιχείων, αλλιώς εμφανίζεται ένα μήνυμα ότι οι εγγραφές είναι επιτυχείς. Κλείνουμε την σύνδεση την οποία ανοίξαμε στην αρχή του κώδικα με την βάση μας και αφού κλείσουμε και τα tags της php δημιουργούμε τα γνωστά μας Links τα οποία είτε μας πηγαίνουν στην αρχική σελίδα είτε μια σελίδα πίσω στην εκάστοτε φόρμα υποβολής στοιχείων.

A.2.10 Σελίδα Εμφάνισης Προγράμματος Σπουδών

Κώδικας 'programa_spoudon.php'

```

<!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">

<html>
<head>

```



```

<title>Μαθήματα</title>

<meta http-equiv="Content-Type" content="text/html"; charset="UTF-8"/>
</head>
<body>
<?php

    function showerror( ){
        die("Error " . mysql_errno( ) . " : " . mysql_error( ));
    }

// Show the wines in an HTML <table>
function displayCourses($result,$examino){
    echo "<h1 align=center> Εξάμηνο $examino </h1>\n";

    // Start a table, with column headers
    echo "\n<table border='10' cellspacing='2' cellpadding='0' height='80' width='600'
align=center bgcolor='#f1f1f1'
        font face='verdana, arial, helvetica' size='2'>\n<tr>\n" .
        "\n\t<th>Κωδικός</th>" .
        "\n\t<th>Όνομα</th>" .
        "\n\t<th>Εξάμηνο</th>" .
        "\n\t<th>Διδακτικές Μονάδες</th>" .
        "\n\t\t<th>Τύπος</th>" .
        "\n\t\t<th>Ωρες</th>" .
        "\n</tr>";

    // Until there are no rows in the result set,
    // fetch a row into the $row array and ...
    while ($row = @ mysql_fetch_row($result)){
        // ... start a TABLE row ...
        echo "\n<tr>";

        // ... and print out each of the attributes
        // in that row as a separate TD (Table Data).
        foreach($row as $data)
            echo "\n\t<td align=center> $data </td>";

        // Finish the row
        echo "\n</tr>";
    }

    // Then, finish the table
    echo "\n</table>\n";
}

// Connect to the MySQL server
if (!$connection = @ mysql_connect("127.0.0.1","root",""))
    die("Cannot connect");

if (!(mysql_select_db("grammateia", $connection)))
    showerror( );

    mysql_query("SET CHARACTER SET 'utf8'", $connection);
    mysql_query("SET NAMES 'utf8'", $connection);

```

```

    $arr = array("A","B","Γ","Δ","Ε","ΣΤ","Ζ");

    foreach ($arr as $examino) {
        $query = "SELECT * FROM mathima where examino='$examino'";

        // Run the query on the connection
        if (!$result = @ mysql_query ($query, $connection))
            showerror( );

        // Display the results
        displayCourses($result,$examino);
    }
    // Close the connection
    if (!(mysql_close($connection)))
        showerror( );
?>

<p align="center" > <em><strong><p><a href="selida_diaxisis.php">Επιστροφή Στην
Σελίδα Διαχείρισης</a></p></strong></em></p>
</body>
</html>

```

Στο κομμάτι του κώδικα του προγράμματος σπουδών έχουμε αρχικά το ίδιο κομμάτι όπως σε κάθε σελίδα το οποίο αναφέρεται στα standards του οργανισμού W3C. Έπειτα στο κομμάτι head δίνουμε το όνομα της καρτέλας που εμφανίζεται στον browser. Στο body έχουμε τον κύριο κώδικα που εκτελεί όλες τις διεργασίες στην σελίδα μας. Δημιουργούμε μία συνάρτηση την οποία χρησιμοποιούμε για την εμφάνιση τυχόν λάθους στον κώδικα κατά την εκτέλεση του. Μετά έχουμε άλλη συνάρτηση η οποία μας εμφανίζει τα εξάμηνα και τα στοιχίζει στην μέση της σελίδας. Κατόπιν με την echo δημιουργούμε και εμφανίζουμε τα στοιχεία και τον πίνακα που βλέπουμε όταν πατάμε το κουμπί του προγράμματος σπουδών. Με μία while εμφανίζουμε τους πίνακες τους οποίους βλέπουμε στην οθόνη μας.

Έπειτα κάνουμε την σύνδεση με την βάση μας και χρησιμοποιούμε και την συνάρτηση ελέγχου λαθών που φτιάξαμε αρχικά, έτσι ώστε να ελέγχουμε αν η σύνδεση είναι εντάξει κάθε φορά που εκτελούνται οι εντολές. Θέτουμε την κωδικοποίηση των χαρακτήρων ούτως ώστε να εμφανίζονται στα ελληνικά τα περιεχόμενα των πινάκων. Με μία foreach παίρνουμε κάθε φορά τα δεδομένα από την βάση μας και τα εμφανίζουμε στους πίνακες μας βάσει του εξαμήνου του οποίου κάθε φορά δηλώνουμε στην εγγραφή μαθήματος. Επιπλέον κάθε φορά που βάζουμε ή διαγράφουμε ένα μάθημα από τα κουμπιά εισαγωγής και διαγραφής μαθήματος, τότε αυτόματα το εκάστοτε μάθημα εμφανίζεται στο πρόγραμμα σπουδών. Τέλος κλείνουμε την σύνδεση με την βάση μας και έξω από την php δημιουργούμε τα σχετικά links. Επίσης πάνω από κάθε κομμάτι κώδικα με τα // τα οποία εισάγουν σχόλια στην php γράφουμε στα αγγλικά με δύο λέξεις τι κάνει το κάθε κομμάτι κώδικα.

A.2.11 Σελίδα Αλλαγής Κωδικού Εισόδου

Κώδικας 'change_password.php'

```
<?php
session_start();
mysql_connect("localhost", "root", "")or die("cannot connect");
mysql_select_db("dblogin")
?>

<!doctype html public "-//w3c//dtd html 3.2//en">
<META HTTP-EQUIV="Content-Type" content="text/html; charset=windows-1253">
<html>

<style type="text/css">
<!--
.vic {
    font-family: Georgia, Times New Roman, Times, serif;
}
-->
</style>
<head>
<title>Αλλαγή Password</title>

<meta name="GENERATOR" content="Arachnophilia 4.0">
<meta name="FORMATTER" content="Arachnophilia 4.0">

</head>

<body>

<h2 align="center">Σελίδα αλλαγής κωδικού πρόσβασης</h2>
<p><br>
<br><br>
<?
// check the login details of the user and stop execution if not logged in
//require "check.php";

echo "<form action='change-passwordck.php' method=post><input type=hidden name=todo
value=change-password>

<table border='10' cellspacing='0' cellpadding='0' align=center>
<tr bgcolor=#f1f1f1' > <td colspan='2' align='center'><font face='verdana, arial, helvetica'
size='2' align='center'>&nbsp;<b>Αλλαγή κωδικού πρόσβασης</b> </font></td> </tr>

<tr bgcolor=#ffffff' > <td ><font face='verdana, arial, helvetica' size='2' align='center'>
&nbsp;<b>Παλαιός κωδικός
</font></td> <td align='center'><font face='verdana, arial, helvetica' size='2' >
```

```

<input type = 'password' class = 'bginput' name = 'pass' ></font></td></tr>

<tr bgcolor = '#ffffff' > <td ><font face = 'verdana, arial, helvetica' size = '2' align = 'center'>
&nbsp;   Νέος κωδικός
</font></td> <td align = 'center'><font face = 'verdana, arial, helvetica' size = '2' >
<input type = 'password' class = 'bginput' name = 'password' ></font></td></tr>

<tr bgcolor = '#f1f1f1' > <td ><font face = 'verdana, arial, helvetica' size = '2' align = 'center'>
&nbsp;   Επιβεβαίωση κωδικού
</font></td> <td align = 'center'><font face = 'verdana, arial, helvetica' size = '2' >
<input type = 'password' class = 'bginput' name = 'password2' ></font></td></tr>

<tr bgcolor = '#ffffff' > <td colspan = 2 align = center><input type = submit value = 'Αλλαγή
κωδικού'><input type = reset value = 'Επαναφορά'></font></td></tr>

";

echo "</table>";

//require "bottom.php";

?>
</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p align = "center" class = "vic"> <em><strong><a href = "selida_diaxirisis.php">Επιστροφή
Στην Σελίδα Διαχείρισης</a></strong></em></p>
</body>

</html>

```

Στο πρώτο κομμάτι αυτού του κώδικα, το οποίο περικλείεται από τα tags της php εκτελούμε την σύνδεση με την βάση δεδομένων μας όπως και στις προηγούμενες σελίδες που εξηγήσαμε προηγουμένως. Μετά την σύνδεση μας με την βάση δεδομένων, δίνουμε τον τίτλο της σελίδας μας στον browser και μετά κλείνουμε το κομμάτι head. Μετά στο body με την χρήση της php φτιάχνουμε μία φόρμα στην οποία βάζουμε τα πεδία με το όνομα που θέλουμε και την οποία υποβάλλουμε μέσω του κώδικα παρακάτω.

Κώδικας 'change_passwordck.php'

```

<?php
mysql_connect("localhost", "root", "")or die("cannot connect");
mysql_select_db("dblogin")
?>

<META HTTP-EQUIV="Content-Type" content="text/html; charset=windows-1253">

```

```

<!doctype html public "-//w3c//dtd html 3.2//en">

<html>

<head>
<title>Αλλαγή Password</title>

</head>

<body>
<?
// check the login details of the user and stop execution if not logged in
//require "check.php";
//////////Collect the form data //////////

$pass=$_POST['pass'];
$todo=$_POST['todo'];
$password=$_POST['password'];
$password2=$_POST['password2'];
//////////

if(isset($todo) and $todo=="change-password"){
$password=mysql_real_escape_string($password);

//Setting flags for checking
$status = "OK";
$msg="";

$res=mysql_query("SELECT password FROM user where username='admin'");
$newarray=mysql_fetch_array($res);

if( $pass != $newarray['password'] ){
    $msg=$msg."The old password in not correct<BR>";
    $status="NOTOK";
}

if ( strlen($password) < 3 or strlen($password) > 8 ){
    $msg=$msg."Password must be more than 3 char length and maximum 8 char length<BR>";
    $status= "NOTOK";}

if ( $password <> $password2 ){
    $msg=$msg."Both passwords are not matching<BR>";
    $status= "NOTOK";}

if($status<>"OK"){
    echo "<font face='Verdana' size='2' color=red>$msg</font><br><center><input
type='button' value='Retry' onClick='history.go(-1)'></center>";
    }else{ // if all validations are passed.

```

```

if(mysql_query("update user set password='". ($_POST['password']) . "'")){
    echo "<font face='Verdana' size='2' ><center>Thanks <br> Your password changed
successfully. Please keep changing your password for better security</font></center>";
    }else{echo "<font face='Verdana' size='2' color=red><center>Sorry <br> Failed to change
password Contact Site Admin</font></center>";
    }
    }
    }
//require "bottom.php";

?>
<p align="center" class="vic"> <em><strong><a href="selida_diaxirisis.php">Επιστροφή
Στην Σελίδα Διαχείρισης</a></strong></em></p>

</body>

</html>

```

Στο πρώτο κομμάτι αυτού του κώδικα, το οποίο περικλείεται από τα tags της php εκτελούμε την σύνδεση με την βάση δεδομένων μας όπως και στις προηγούμενες σελίδες που εξηγήσαμε προηγουμένως. Μετά την σύνδεση μας με την βάση δεδομένων, δίνουμε τον τίτλο της σελίδας μας στον browser και μετά κλείνουμε το κομμάτι head. Στο body αρχικά αποθηκεύουμε σε κάποιες μεταβλητές το παλιό password και το καινούργιο με την επιβεβαίωση του που πρέπει να είναι ίδιο με το προηγούμενο. Επίσης αποθηκεύουμε σε μία μεταβλητή το όνομα της φόρμας και μετά με μία if ελέγχουμε αν ισχύει αυτή η φόρμα αυτή ,δηλαδή τα στοιχεία της είναι ορθά. Έπειτα με ένα sql ερώτημα ελέγχουμε αν υπάρχουν κάποιοι ειδικοί χαρακτήρες στον κωδικό που δίνουμε και να τους αποφύγουμε. Μετά αποθηκεύουμε σε μία μεταβλητή με το ίδιο όνομα απαλλαγμένη από τους ειδικούς χαρακτήρες. Θέτουμε κάποιες σημαίες τις οποίες αργότερα θα χρησιμοποιήσουμε για τους ελέγχους. Επιλέγουμε το παλιό password από την βάση δεδομένων μας και το συγκρίνουμε με αυτό που βάζει ο χρήστης στη σελίδα. Αν δεν είναι σωστό εμφανίζεται σε άλλη σελίδα μήνυμα ότι είναι λάθος και πρέπει να ξαναπροσπαθήσουμε. Αυτός είναι ο πρώτος και κυριότερος έλεγχος πριν πάμε παρακάτω. Μετά ελέγχουμε το μήκος των εισαγόμενων νέων κωδικών το οποίο πρέπει να είναι μεγαλύτερο από 3 χαρακτήρες και μικρότερο από 8. Πάλι ελέγχουμε βάση της σημαίας και αναλόγως τι τιμή της δίνουμε εκτελείται και η ανάλογη πράξη. Παρακάτω ελέγχουμε αν είναι ίδια και τα δύο passwords που έχει βάλει ο χρήστης. Τέλος τοποθετούμε ένα κουμπί το οποίο μας πηγαίνει πάντα πίσω στην φόρμα αλλαγής κωδικού κάθε φορά που δεν περνάμε κάποιο από τους παραπάνω ελέγχους. Αν είναι όλα εντάξει τότε εμφανίζεται σχετικό μήνυμα και εκτελείται η αλλαγή του κωδικού στην βάση δεδομένων. Μετά τα tags της php βάζουμε ένα Link το οποίο μας πηγαίνει στην αρχική σελίδα.

A.2.12 Σελίδα Δήλωσης Μαθημάτων

Κώδικας 'dilos_i_m.php'

```
<!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">

<html>
<head>
  <title>Μαθήματα</title>

<meta http-equiv="Content-Type" content="text/html"; charset="UTF-8"/>

</head>

<body>
<div align="center"><body>
<div align="left">
  <script type="text/javascript">
function startTime()
{ var today =new Date();
var h=today.getHours();
var m=today.getMinutes();
var s=today.getSeconds();
m=checkTime(m);
s=checkTime(s);
document.getElementById('txt').innerHTML=h+":"+m+":"+s;
t=setTimeout('startTime()',500);
}
function checkTime(i)
{ if (i<10)
{ i="0"+i;
} return i;
} </script>

<script type="text/javascript">
function isNumber(obj)
{
if(document.getElementById('err'))
{
obj.parentNode.removeChild(document.getElementById('err'));
}
setTimeout(function(){
if(isNaN(obj.value))
{
obj.select();
var p=document.createElement('p');
p.style.color="red";
```

```

p.style.fontStyle='italic';
p.style.margin="0px";
p.style.padding="0px";
p.id='err';
p.appendChild(document.createTextNode('Input must be numeric'));
obj.parentNode.appendChild(p);
}
},1);
}

```

```

function CheckRequiredFields() {

```

```

var errormessage = new String();

```

```

if(WithoutContent(document.dilosi.fname.value))
    { errormessage += "\n\nΣυμπληρώστε το πεδίο του Ονόματος."; }

```

```

if(WithoutContent(document.dilosi.lname.value))
    { errormessage += "\n\nΣυμπληρώστε το πεδίο του Επιθέτου."; }

```

```

if(WithoutContent(document.dilosi.f_name.value))
    { errormessage += "\n\nΣυμπληρώστε το πεδίο του Ονόματος Πατρός."; }

```

```

if(WithoutContent(document.dilosi.am.value))
    { errormessage += "\n\nΣυμπληρώστε το πεδίο του AM."; }

```

```

if(errormessage.length > 0) {
    alert('NOTE:' + errormessage);
    return false;
}

```

```

return true;
}

```

```

function WithoutContent(ss) {
if(ss.length > 0) { return false; }
return true;
}

```

```

</script>

```

```

<body onload="startTime()">
<div align="center">
<h2 align="center" class="style1"><font face="Georgia, Times New Roman, Times,
serif"><em><strong>Δήλωση Μαθημάτων</strong></em></font></h2>
<p align="center" class="style1">
<div align="center">
<form method="POST" name="dilosi" action="insert_dilosi.php" onsubmit="return
CheckRequiredFields()">
<table width="91%" height="116" border="10" align="center" TD>
<tr>
<td align="center"><p><strong><font face="Georgia, Times New Roman, Times,
serif">Όνομα :
<input type="text" name="fname" size="20">
</font><em><font face="Georgia, Times New Roman, Times, serif">Επίθετο :

```



```

        <input name='lname' type='text' class='bginput' >
        <strong><em>Όνομα πατρός :<input name='f_name' type='text' class='bginput' >
        </em></strong></font></em></strong><strong><em><strong><em><font
face="Georgia, Times New Roman, Times, serif"> A.M.:
        </font><strong><em><font face="Georgia, Times New Roman, Times, serif">
        <input name='am' type='text' class='bginput' >
        </font></em></strong><font face="Georgia, Times New Roman, Times, serif">
</font></em></strong></em><font face="Georgia, Times New Roman, Times, serif">
        </font></strong></p></td>
    </tr>

</table>
</p>
</div>

<?php
function insert(){
}
//connect to databse or show the error
function showerror(){
    die("Error " . mysql_errno() . " : " . mysql_error());
}

    if (!$connection = @ mysql_connect("127.0.0.1","root",""))
        die("Cannot connect");

    if (!(mysql_select_db("grammateia", $connection)))
        showerror();
    // mysql_query("SET CHARACTER SET 'utf8'", $connection);
mysql_query("SET NAMES 'utf8'", $connection);

    $query = "select onoma from mathima";

    echo "<div class='courses'> ";
    echo "<h1> Μαθήματα </h1>";
// Start a table, with column headers
    $i = 0;
    $arr = array();
    for($i = 0; $i < 5; $i++){
        if (!$result = @ mysql_query ($query, $connection))
            showerror();
        echo "<br><strong>Διάλεξε Μάθημα</strong>
        <select name='lesson$i'>";
        echo "<option value=" selected='selected'>--Διάλεξε Μάθημα--
</option>";

        // Until there are no rows in the result set,
        // fetch a row into the $row array and ...
        while ($row = @ mysql_fetch_row($result)){

            // ... and print out each of the attributes
            // in that row as a separate TD (Table Data).
            foreach($row as $data){
                echo " <option>$data </option>";
            }
        }
    }
}

```

```

        // $arr[$i] = lesson.options[lesson.selectedIndex];
    }
}
echo "</select>\n\n";
echo "<br><br><br>";

} //end for
// insert();
echo "</div>";
?>
<input type="submit" value="Δήλωση" name="B62" >
    <input type="reset" value="Εκκαθάριση" name="B72">
</form>

<p align="center"><strong><em><strong><font size="4" face="Georgia, Times New
Roman, Times, serif"><a href="selida_diaxirisis.php">Επιστροφή Στην Σελίδα
Διαχείρισης</a></font></strong></em></strong></p>
</div></body>
</html>

```

Αρχικά στο κομμάτι head δίνουμε τον τίτλο της καρτέλας μας και έπειτα στο body φτιάχνουμε αρχικά την φόρμα την οποία και θα υποβάλουμε μέσω του επόμενου κώδικα που θα αναλύσουμε παρακάτω. Αφού δώσουμε τα ονόματα που θέλουμε και στοιχίσουμε τα πεδία που εισάγουμε τα στοιχεία του φοιτητή προχωρούμε στην δημιουργία μίας συνάρτησης η οποία θα χρησιμοποιηθεί για τον έλεγχο με την σύνδεση με την βάση δεδομένων μας. Αν αποτύχει τότε μας πετάει σχετικό μήνυμα και σταματά την σύνδεση. Θέτουμε την κωδικοποίηση των χαρακτήρων στα ελληνικά και έπειτα με ένα ερώτημα sql παίρνουμε το όνομα από το πρώτο μάθημα που είναι αποθηκευμένο και στο τοποθετούμε σε μία μεταβλητή. Έπειτα αφού φτιάξουμε την επικεφαλίδα μας με μία for loop και με ερωτήματα sql μέσα στην for δημιουργούμε τις drop down lists και κάθε φορά παίρνουμε το επόμενο μάθημα το οποίο είναι αποθηκευμένο στον πίνακα μάθημα και το βάζουμε μέσα στις λίστες μέχρι να τελειώσουν. Τέλος φτιάχνουμε τα κουμπιά της υποβολής και της εκκαθάρισης και τέλος τα γνωστά μας links.

Κώδικας 'insert_dilos.php'

```

<!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>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Δήλωση Μαθημάτων</title>
</head>
<?php

function showerror() {
    die("Error " . mysql_errno() . " : " . mysql_error());
}
// Connect to the MySQL server
if (!$connection = @mysql_connect("127.0.0.1","root",""))
    die("Cannot connect");

```

```

if (!(mysql_select_db("grammateia", $connection)))
    showerror( );

    mysql_query("SET CHARACTER SET 'utf8'", $connection);
mysql_query("SET NAMES 'utf8'", $connection);

// am tou foithth
$am = $_POST['am'];

$mathima0 = $_POST['lesson0'];
$mathima1 = $_POST['lesson1'];
$mathima2 = $_POST['lesson2'];
$mathima3 = $_POST['lesson3'];
$mathima4 = $_POST['lesson4'];
/*
for($i = 0 ; $i < 5; $i++){
    // make $mathima0, etc
    $temp = "$"."mathima" ".$i;
    echo $temp;
    echo "<br>";
    $temp = mysql_query("SELECT examino FROM mathima WHERE
onoma='$temp'", $connection);
        echo $temp[0];
        ///

    $temp2= mysql_fetch_row($temp);
    echo $temp2[0];
    echo "<br>";

    // kwdikos tou mathimatos
    $code = mysql_query ("SELECT KODIKOS_M FROM mathima WHERE
ONOMA='$temp'", $connection);
    $kode = mysql_fetch_row($code);

    echo $kode[0];
    echo "<br>";
    $query0 = "INSERT INTO PAIRNEI
VALUES('$am', '$temp', '$temp2[0]', '$kode[0]') ";

    if (!($result = @ mysql_query ($query0, $connection)))
        showerror( );

break;
}
*/

// examino tou mathimatos
$examino0 = mysql_query("SELECT examino FROM mathima WHERE
onoma='$mathima0'", $connection);
if (mysql_num_rows($examino0) != 0)
    $semester0= mysql_fetch_row($examino0);

$examino1 = mysql_query("SELECT examino FROM mathima WHERE
onoma='$mathima1'", $connection);
if (mysql_num_rows($examino1) != 0)

```

```

$semester1= mysql_fetch_row($examino1);

$examino2 = mysql_query("SELECT examino FROM mathima WHERE
onoma='$mathima2'", $connection);
if (mysql_num_rows($examino2) != 0)
$semester2= mysql_fetch_row($examino2);

$examino3 = mysql_query("SELECT examino FROM mathima WHERE
onoma='$mathima3'", $connection);
if (mysql_num_rows($examino3) != 0)
$semester3= mysql_fetch_row($examino3);

$examino4 = mysql_query("SELECT examino FROM mathima WHERE
onoma='$mathima4'", $connection);
if (mysql_num_rows($examino4) != 0)
$semester4= mysql_fetch_row($examino4);

// kwdikos tou mathimatos
$code0 = mysql_query ("SELECT KODIKOS_M FROM mathima WHERE
ONOMA='$mathima0'", $connection);
if (mysql_num_rows($code0) != 0)
$kode0 = mysql_fetch_row($code0);

$code1 = mysql_query ("SELECT KODIKOS_M FROM mathima WHERE
ONOMA='$mathima1'", $connection);
if (mysql_num_rows($code1) != 0)
$kode1 = mysql_fetch_row($code1);

$code2 = mysql_query ("SELECT KODIKOS_M FROM mathima WHERE
ONOMA='$mathima2'", $connection);
if (mysql_num_rows($code2) != 0)
$kode2 = mysql_fetch_row($code2);

$code3 = mysql_query ("SELECT KODIKOS_M FROM mathima WHERE
ONOMA='$mathima3'", $connection);
if (mysql_num_rows($code3) != 0)
$kode3 = mysql_fetch_row($code3);

$code4 = mysql_query ("SELECT KODIKOS_M FROM mathima WHERE
ONOMA='$mathima4'", $connection);
if (mysql_num_rows($code4) != 0)
$kode4 = mysql_fetch_row($code4);

if ((mysql_num_rows($code0) != 0) && (mysql_num_rows($examino0) != 0) )
$query0 = "INSERT INTO PAIRNEI
VALUES('$sam', '$mathima0', '$semester0[0]', '$kode0[0]') ";
if ((mysql_num_rows($code1) != 0) && (mysql_num_rows($examino1) != 0) )
$query1 = "INSERT INTO PAIRNEI
VALUES('$sam', '$mathima1', '$semester1[0]', '$kode1[0]') ";
if ((mysql_num_rows($code2) != 0) && (mysql_num_rows($examino2) != 0) )
$query2 = "INSERT INTO PAIRNEI
VALUES('$sam', '$mathima2', '$semester2[0]', '$kode2[0]') ";
if ((mysql_num_rows($code3) != 0) && (mysql_num_rows($examino3) != 0) )
$query3 = "INSERT INTO PAIRNEI
VALUES('$sam', '$mathima3', '$semester3[0]', '$kode3[0]') ";

```

```

        if ((mysql_num_rows($code4) != 0) && (mysql_num_rows($examino4) != 0) )
        $query4 = "INSERT INTO PAIRNEI
VALUES('$am','$mathima4','$semester4[0]','$kode4[0]') ";

// Run the query on the connection
        if (!$result = @ mysql_query ($query0, $connection))
                ;//showerror( );
        if (!$result = @ mysql_query ($query1, $connection))
                ;// showerror( );
        if (!$result = @ mysql_query ($query2, $connection))
                ;// showerror( );
        if (!$result = @ mysql_query ($query3, $connection))
                ;// showerror( );
        if (!$result = @ mysql_query ($query4, $connection))
                ;// showerror( );

?>
<div>
<p align="center"><strong><em><strong><font size="4" face="Georgia, Times New
Roman, Times, serif"><a href="selida_diaxirisis.php">Επιστροφή Στην Σελίδα
Διαχείρισης</a></font></strong></em></strong></p></div>
<body>
</body>
</html>

```

Και εδώ αρχικά δημιουργούμε τον τίτλο της καρτέλας μας και έπειτα φτιάχνουμε μία συνάρτηση για τον έλεγχο της σύνδεσής με την βάση δεδομένων μας και μετά εκτελούμε την σύνδεση με την βάση μας και χρησιμοποιούμε και την προηγούμενη συνάρτηση για τον έλεγχο της σύνδεσης. Μετά αποθηκεύουμε σε μεταβλητές το ΑΜ του φοιτητή και τα μαθήματα που δηλώνει ο χρήστης. Κατόπιν με sql ερωτήματα παίρνουμε από την βάση δεδομένων μας βάσει των μαθημάτων και των εξαμήνων που αποθηκεύσαμε στις προηγούμενες μεταβλητές τοποθετούμε τα εξάμηνα των μαθημάτων σε κάποιες μεταβλητές. Το ίδιο κάνουμε και για τον κωδικό των μαθημάτων και μετά με κάποια insert into sql ερωτήματα τοποθετούμε στον πίνακα παίρνει τα μαθήματα που δηλώνουμε κάθε φορά. Μετά εκτελούμε κάποιους ελέγχους για την εισαγωγή των δεδομένων μας στον πίνακα και τέλος φτιάχνουμε τα links μας.

A.2.13 Σελίδα Εμφάνισης και εκτύπωσης αναλυτικής βαθμολογίας φοιτητή

Κώδικας ‘‘ analitiki_bathmologia.php’’

```

<head>
<META HTTP-EQUIV="Content-Type" content="text/html; charset=iso-8859-7">
<title>ΕΚΤΥΠΩΣΗ ΑΝΑΛΥΤΙΚΗΣ</title>
</head>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

```

```

"http://www.w3.org/TR/html4/loose.dtd">
<html>

<html>

<head>
<META HTTP-EQUIV="Content-Type" content="text/html; charset= iso-8859-7">
<title>Καρτέλλα Αναλυτικής Κατάστασης</title>
<meta name="GENERATOR" content="Arachnophilia 4.0">
<meta name="FORMATTER" content="Arachnophilia 4.0">
</head>

<body></head><html xmlns="http://www.w3.org/1999/xhtml">
<div align="center"><body>
<div align="left">
  <script type="text/javascript">
function startTime()
{ var today =new Date();
var h=today.getHours();
var m=today.getMinutes();
var s=today.getSeconds();
m=checkTime(m);
s=checkTime(s);
document.getElementById('txt').innerHTML=h+":"+m+":"+s;
t=setTimeout('startTime()',500);
}
function checkTime(i)
{if (i<10)
{i="0"+i;
} return i;
}

function CheckRequiredFields() {

var errormessage = new String();

if(WithoutContent(document.analitiki.name.value))
  { errormessage += "\n\nΣυμπληρώστε το πεδίο του Ονόματος."; }

if(WithoutContent(document.analitiki.surname.value))
  { errormessage += "\n\nΣυμπληρώστε το πεδίο του Επιθέτου."; }

if(WithoutContent(document.analitiki.am.value))
  { errormessage += "\n\nΣυμπληρώστε το πεδίο του AM."; }

if(errormessage.length > 0) {
  alert('NOTE:' + errormessage);
  return false;
}
return true;
}

function WithoutContent(ss) {
if(ss.length > 0) { return false; }

```

```
return true;
}
```

```
</script>
</head>
<body onload="startTime()">
</div>
<div id="txt"></div>
```

```
<h2 align="center"><em><font face="Georgia, Times New Roman, Times, serif"><strong>Σελίδα
Εκτύπωσης Αναλυτικής Βαθμολογίας </strong>
</font></em></h2>
<p align="center">&nbsp;</p>
<p align="center">&nbsp;</p>
<p align="center">&nbsp;</p>
<p align="center"><table border="10" align="center" TD width="30%" height="100">
<form method="POST" name="analitiki" action="print.php" onSubmit="return
CheckRequiredFields()">
<tr>
<tr><td align="center"><strong><font face="Georgia, Times New Roman, Times, serif">Όνομα :
<input type="text" name="name" size="20">
</font></strong></td></tr>
<tr><td align="center"><strong><font face="Georgia, Times New Roman, Times, serif">Επώνυμο :
<input type="text" name="surname" size="20">
</font></strong></td></tr>
<tr><td align="center"><strong><font face="Georgia, Times New Roman, Times, serif">Α.Μ
:<em> :</em>
<input type="text" name="am" size="20">
</font></strong></td></tr>
<tr><td align="center"><strong><font face="Georgia, Times New Roman, Times, serif">
<input type="submit" value="Προβολή" name="B62">
<input type="reset" value="Εκκαθάριση" name="B72">
</font></strong></td>
</tr>
</form>
</table>
</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p align="center"><font color="#00CC99" size="5" face="Arial, Helvetica, sans-serif">
</font><font color="#00CC99" size="5" face="Arial, Helvetica, sans-serif">
</font></p>
<p align="center"><strong><font size="4" face="Georgia, Times New Roman, Times, serif"><em><a
href="selida_diaxirisis.php">Επιστροφή Στην Σελίδα Διαχείρισης</a></em></font></strong></p>
</body>
</html>
```

Αρχικά στο κομμάτι head δίνουμε τον τίτλο της καρτέλας μας και έπειτα στο body φτιάχνουμε αρχικά την φόρμα την οποία και θα υποβάλουμε μέσω του επόμενου κώδικα που θα αναλύσουμε παρακάτω. Αφού δώσουμε τα ονόματα που θέλουμε και στοιχίσουμε τα πεδία που εισάγουμε τα στοιχεία του φοιτητή προχωρούμε στην δημιουργία μίας συνάρτησης η οποία θα χρησιμοποιηθεί για τον έλεγχο με την σύνδεση με την βάση δεδομένων μας. Αν αποτύχει τότε μας πετάει σχετικό μήνυμα και σταματά την σύνδεση. Θέτουμε την κωδικοποίηση των χαρακτήρων στα ελληνικά. Πιο πάνω έχουμε μία συνάρτηση φτιαγμένη με την JavaScript η οποία μας δημιουργεί και εμφανίζει το ψηφιακό ρολόι που βλέπουμε στο πάνω μέρος της σελίδας μας και γενικότερα σε όλες τις σελίδες. Τέλος φτιάχνουμε τα κουμπιά της υποβολής και της εκκαθάρισης και τα γνωστά μας links.

Κώδικας 'print.php'

```
<!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>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Εκτύπωση Αναλυτικής </title>
</head>

<body>

<?php
//connect to database or show the error
function showerror(){
    die("Error " . mysql_errno() . " : " . mysql_error());
}

if (!$connection = @ mysql_connect("127.0.0.1","root",""))
    die("Cannot connect");

    if (!(mysql_select_db("grammateia", $connection)))
        showerror();

    mysql_query("SET CHARACTER SET 'utf8',$connection);
    mysql_query("SET NAMES 'utf8',$connection);

$name=$_POST['name'];
$surname=$_POST['surname'];
$am=$_POST['am'];

$result=mysql_query("SELECT * FROM pairnei WHERE AM='$am',$connection);// or
die(mysql_error());

if (mysql_num_rows($result) == 0) {
    echo "Δεν έχουν καταχωρηθεί βαθμολογίες για αυτόν τον φοιτητή";
    exit;
}
```



```

//$kodikos0= mysql_fetch_array($kodikos);
Print "<table border cellpadding=3>";
Print "<tr>
<th>Κωδικός Μαθήματος</th>
<th>Βαθμός Μαθήματος</th>
<th>Όνομα Μαθήματος</th>
<th>Εξάμηνο</th>
</tr>";
while ($row = mysql_fetch_assoc($result)) {
    Print "<tr>";
        Print "<td>".$row["CODE"]. "</td> ";
        Print "<td>".$row["GRADE"]. "</td> ";
        Print "<td>".$row["ONOMA"]. "</td> ";
        Print "<td>".$row["EXAMINO"]. "</td></tr> ";
}
Print "</table>";

//$kodikos0= mysql_fetch_row($kodikos);
//Print "kodikos=".$kodikos0['code'];
/*
Print "<table border cellpadding=3>";
while($kodikos0 = mysql_fetch_array( $kodikos ))
{
Print "<tr>";
Print "<th>AM:</th> <td>".$kodikos0['am'] . "</td> ";
// Print "<th>Pet:</th> <td>".$kodikos0['pet'] . " </td></tr>";
} */

//if (!(mysql_close($connection))
//      showerror( );
?>
<p align="center">&nbsp;</p>
<p align="center">&nbsp;</p>
<p align="center">&nbsp;</p>

<p><a href="selida_diaxirisis.php">Επιστροφή Στην Σελίδα Διαχείρισης</a></p>
<p><a href="analitiki_bathmologia.php">Επιστροφή Στην Προηγούμενη Σελίδα</a></p>

</body>
</html>

```

Και εδώ αρχικά δημιουργούμε τον τίτλο της καρτέλας μας και έπειτα φτιάχνουμε μία συνάρτηση για τον έλεγχο της σύνδεσής με την βάση δεδομένων μας και μετά εκτελούμε την σύνδεση με την βάση μας και χρησιμοποιούμε και την προηγούμενη συνάρτηση για τον έλεγχο της σύνδεσης. Μετά αποθηκεύουμε σε μεταβλητές το AM του φοιτητή ,το όνομα και το επώνυμο του. Κατόπιν με sql ερώτημα παίρνουμε από την βάση δεδομένων μας το AM του φοιτητή που έχουμε βάλει στην φόρμα της σελίδας. Μετά με μία if ελέγχουμε αν είναι το AM καταχωρημένο στην βάση μας ,αλλιώς εμφανίζουμε το κατάλληλο μήνυμα στον χρήστη. Έπειτα δημιουργούμε τον πίνακα που εμφανίζεται στην σελίδα μετά που πατάμε την υποβολή και όπου εκεί θα βάλουμε τα στοιχεία των μαθημάτων που θέλουμε να εκτυπώσουμε. Με μία while παίρνουμε και εμφανίζουμε τα αποτελέσματα στον πίνακα. Μετά εκτελούμε κάποιους ελέγχους για την εισαγωγή των δεδομένων μας στον πίνακα και τέλος φτιάχνουμε τα links μας.

Βιβλιογραφία

- [1] PHP 5 Power Programming By [Andi Gutmans](#), [Stig Bakken](#), [Derick Rethans](#) ,Published Oct 27, 2004 by [Prentice Hall](#). Part of the [Bruce Perens' Open Source Series](#) series.
- [2] PHP and MySQL Web Development (4th Edition) [Paperback] [Luke Welling](#) (Author), [Laura Thomson](#) (Author)
- [3] <http://www.php.net/>
- [4] <http://www.w3schools.com/php/default.asp>
- [5] <http://en.wikipedia.org>
- [6] <http://www.phpbuilder.com/welcomead/>
- [7] <http://www.tizag.com/phpT/>
- [8] www.google.com
- [9] http://dev.mysql.com/tech-resources/articles/mysql_intro.html
- [10] <http://www.mysqltutorial.org/>
- [11] <http://www.adobe.com/>
- [12] <http://www.easyphp.org/>