

Ελληνικό Μεσογειακό Πανεπιστήμιο

Σχολή Μηχανικών

**Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών
Υπολογιστών.**



Πτυχιακή Εργασία

Πληροφοριακό Σύστημα για ασφαλής Social Network,για
εκπαιδευτικούς σκοπούς.

Του φοιτητή
Κρανιωτακή Γιώργου
ΑΜ3385

Επιβλέπων καθηγητης
κ.Παπαδάκης Νίκος

Ηράκλειο 2020

Πρόλογος

Μεγάλο κεφάλαιο για την πρόοδο του κάθε φοιτητή στις σχολές μας, είναι η βοήθεια που παρέχει το γνωστό σε όλους μας πλέον E-class. Το e-class, είναι μια πλατφόρμα, όπου καθηγητές και φοιτητές έχουν μια διαρκή επικοινωνία μέσω αυτού για ζητήματα όπως εργασίες, βαθμολογίες, σημειώσεις, ανακοινώσεις ακόμα και άμεση επικοινωνία αυτού κάθε αυτού μέσω Live chatting.

Μέσω e-class μπορούν φυσικά και οι μεταπτυχιακοί φοιτητές να απασχοληθούν είτε μέσω εργασιών είτε μέσω επικοινωνίας με σπουδαστές.

Σε αυτή την πτυχιακή δημιούργησα ένα απλό ιστότοπο του e-class όπου θα μπορούμε να εισερχόμαστε στο σύστημα, να διαλέγουμε μαθήματα και να βλέπουμε τις ενημερώσεις.

Abstract

A big chapter for the progress of every student in our schools, is the help that the well-known to all of us now E-class. The e-class is a platform where teachers and students have a constant communication through it on issues such as homework , ratings, notes, announcements and even direct communication of each one through Live chatting.

Through e-class, of course, postgraduate students can also be employed either through assignments or through communication with students.

In this dissertation I created a simple e-class website where we can log in, select courses and view updates.

Περίληψη

Για την δημιουργία του ιστότοπου e-class, χρειάστηκαν βασικές γνώσεις στις γλώσσες προγραμματισμού, στις βάσεις δεδομένων αλλά και σε κάποια απαραίτητα προγραμματιστικά εργαλεία.

Στο πρώτο κεφάλαιο θα αναλύσουμε τι είναι οι γλώσσες προγραμματισμού και πώς χωρίζονται. Ακόμα θα αναλύσουμε και κάποιες βασικές έννοιες των βάσεων δεδομένων.

Στο δεύτερο κεφάλαιο θα δούμε τις γλώσσες που χρησιμοποιήσαμε για την δημιουργία της εφαρμογής e-class και τι μπορούμε να προγραμματίσουμε μέσω αυτών.

Ακόμα, στο τρίτο κεφάλαιο θα αναλύσουμε τα προγράμματα που χρησιμοποιήθηκαν και κάποιες βασικές χρήσεις.

Τέλος, στο τελευταίο κεφάλαιο θα δούμε τον κώδικα που χρησιμοποιήσαμε και θα αναλυθεί.

Μερικές λέξεις κλειδιά: Προγραμματισμός, e-class, Ίντερνετ, Διαχείριση, SQL

<u>Πρόλογος</u>	2
<u>Κεφάλαιο 1</u>	7
<u>Γλώσσες Προγραμματισμού</u>	8
<u>Διαδικασία Ανάπτυξης Λογισμικού</u>	9
<u>Ανάλυση Απαιτήσεων</u>	10
<u>Σχεδιασμός Λογισμικού</u>	10
<u>Αποσφαλμάτωση</u>	10
<u>Έλεγχος Λογισμικού</u>	10
<u>Συντήρηση Κώδικα</u>	11
<u>Πρότυπα Προγραμματισμού</u>	11
<u>Προστακτικός Προγραμματισμός</u>	11
<u>Συναρτησιακός Προγραμματισμός</u>	11-12
<u>Λογικός Προγραμματισμός</u>	13
<u>Αντικειμενοστραφής Προγραμματισμός</u>	13
<u>Ενθυλάκωση</u>	13
<u>Αφαίρεση Δεδομένων</u>	13
<u>Κληρονομικότητα</u>	13-14
<u>Πολυμορφισμός</u>	15
<u>Βάσεις Δεδομένων</u>	15
<u>Μοντέλα Βάσεων Δεδομένων</u>	16
<u>Δικτυωτό μοντέλο βάσεων δεδομένων</u>	17
<u>Σχεδιακό μοντέλο βάσεων δεδομένων</u>	17
<u>Κεφάλαιο 2</u>	19
<u>Πλατφόρμα Eclass</u>	19
<u>Γλώσσες Προγραμματισμού που μπορούν να χρησιμοποιηθούν</u>	20
<u>Net Framework</u>	20
<u>Γλώσσες Προγραμματισμού</u>	21
<u>Γλώσσα Προγραμματισμού C</u>	21
<u>Χαρακτηριστικά C</u>	22
<u>HTML</u>	23
<u>Σώμα κώδικα HTML</u>	24
<u>Δημιουργία Υπερσυνδέσεων</u>	24
<u>CSS</u>	25
<u>Δυνατότητες Css</u>	25
<u>Στοιχεία Css</u>	25
<u>Βάση Δεδομένων και Queries</u>	26
<u>Javascript</u>	29

<u>Κεφάλαιο 3</u>	31
<u>Προγραμματιστικά Εργαλεία</u>	31
<u>Εργαλεία που χρησιμοποιήθηκαν για το Eclass</u>	
.....	32
<u>Mysql/PhpMyadmin</u>	32
<u>Apache HTTP</u>	32
<u>PHP</u>	32-33
<u>Visual Studio Code</u>	33-43
<u>Κεφάλαιο 4</u>	44
<u>Περιγραφή κώδικα στο Eclass</u>	
.....	41
<u>Τροποποίηση του Eclass για Desktop/Laptop/Tablet/Mobiles</u>	
.....	45
<u>Παρουσίαση και επεξήγηση του κώδικα Eclass</u>	47

Κεφάλαιο 1

Εισαγωγή στις γλώσσες προγραμματισμού και βάσεις δεδομένων.

Γλώσσες προγραμματισμού

Οι γλώσσες προγραμματισμού είναι τυπικές γλώσσες εμπλουτισμένες με σημασιολογία που χρησιμοποιούνται για τον έλεγχο της συμπεριφοράς των ηλεκτρονικών υπολογιστών. Όλες οι γλώσσες προγραμματισμού έχουν κάποια θεμελιώδη δομικά στοιχεία για την περιγραφή των δεδομένων και των διεργασιών ή των μετασχηματισμών που εφαρμόζονται σε αυτά και ορίζονται χρησιμοποιώντας συντακτικούς και σημασιολογικούς κανόνες για τον προσδιορισμό της δομής και της σημασίας τους αντίστοιχα. Οι κανόνες αυτοί περιγράφονται σε μια προδιαγραφή γνωστή και ως Programming Language Specification.

Τα κυριότερα δομικά στοιχεία των γλωσσών προγραμματισμού είναι το συντακτικό, η σημασιολογία και η κύρια βιβλιοθήκη τους. Το συντακτικό μιας γλώσσας περιγράφει τους πιθανούς συνδυασμούς των συμβόλων της γλώσσας που στοιχειοθετούν ορθές φράσεις σε ένα πρόγραμμα. Το νόημα ενός συγκεκριμένου συντακτικώς ορθού συνδυασμού λέξεων καθορίζεται στην σημασιολογία. Η κύρια βιβλιοθήκη μιας γλώσσας, που συνήθως αποτελεί και μέρος της προδιαγραφής της, είναι ένα σύνολο κώδικα για την τέλεση βασικών λειτουργιών όπως το διάβασμα και το γράψιμο δεδομένων ή παρέχει την υλοποίηση βασικών αλγορίθμων και δομών δεδομένων.

Οι περισσότερες γλώσσες προγραμματισμού είναι καθαρά μέσω κειμένων, δηλαδή χρησιμοποιούν σειρές κειμένου που περιλαμβάνουν λέξεις αριθμούς και σημεία στίξης. Υπάρχουν ωστόσο μερικές γλώσσες που είναι από την φύση τους πιο οπτικές, αφού χρησιμοποιούν εικονικές σχέσεις μεταξύ συμβόλων για να καθορίσουν ένα πρόβλημα.

Οι γλώσσες προγραμματισμού δεν είναι εύκολο να κατηγοριοποιηθούν πρωταρχικά, καθώς συνδυάζουν τα στοιχεία τους με νέες ιδέες, ορισμένες εκ των οποίων μπορεί να μην είναι ευδιάκριτες στις φαινομενικά όμοιες τους. Διακρίσεις στις γλώσσες όμως μπορεί να γίνει ανάλογα με τον προγραμματιστικό τρόπο που ακολουθούν και ενδεχομένων με το πεδίο χρήσης για το οποίο προορίζονται. Ακόμα οι γλώσσες προγραμματισμού χωρίζονται σε επίπεδα, όπου στο χαμηλότερο επίπεδο βρίσκεται η μοναδική γλώσσα που μπορεί να κατανοήσει ένας επεξεργαστής, το σύστημα εντολών που εκτελείται κατευθείαν από αυτόν. Αυτό είναι το χαμηλότερο επίπεδο γλώσσας και ονομάζεται γλώσσα μηχανής. Το επίπεδο των υπόλοιπων γλωσσών εξαρτάται από το πόσο μοιάζουν με το κώδικα μηχανής.

Τα προγράμματα σε γλώσσες χαμηλού επιπέδου έχουν το πλεονέκτημα ότι μπορούν να εκμεταλλευτούν όλα τα χαρακτηριστικά που μπορεί να έχει η αρχιτεκτονική του συγκεκριμένου επεξεργαστή που μπορούν να τρέχουν. Κατά συνέπεια μπορούν να είναι πολύ αποδοτικά, αφού χρησιμοποιούν τον επεξεργαστή με το βέλτιστο τρόπο. Η δημιουργία τέτοιων προγραμμάτων απαιτεί αρκετό χρόνο τόσο για την εκμάθηση του συντακτικού τους, όσο και για την ανάπτυξη τους, καθώς και για την πλήρη κατανόηση της λειτουργίας του επεξεργαστή. Για το λόγο αυτό ο προγραμματισμός σε χαμηλό επίπεδο γίνεται για πολύ μικρά προγράμματα ή για τμήματα κώδικα που επιτελούν πολύ σημαντικές εργασίες και είναι απαραίτητο να τρέχουν όσο αποδοτικότερα γίνεται.

Οι γλώσσες υψηλού επιπέδου αναπαριστούν αφηρημένα τις λεπτομέρειες του υπολογιστή, χρησιμοποιούν εκτενώς φυσική γλώσσα, είναι ευκολότερες στη χρήση και συνεπώς επιτρέπουν τη γρήγορη δημιουργία μεγάλων προγραμμάτων. Το τελικό πρόγραμμα μπορεί να μην είναι όσο αποδοτικό όσο το αντίστοιχο σε γλώσσα χαμηλού επιπέδου, ωστόσο το κέρδος στο χρόνο ανάπτυξης υπερκεράζει την απόδοση των τελικών προγραμμάτων. Υπάρχουν δύο κύρια μοντέλα εκτέλεσης των υψηλών γλωσσών προγραμματισμού : α) οι διερμηνευμένες γλώσσες (interpreted languages) οι οποίες διαβάζονται από ένα ειδικό πρόγραμμα που καλείται διερμηνευτής (interpreter), μετατρέπονται σε γλώσσα μηχανής από αυτό και εκτελούνται άμεσα χωρίς φάση μεταγλώττισης (πχ. η PHP και η Python) και β) οι μεταγλωττισμένες γλώσσες (compiled languages) που πριν την εκτέλεση τους μετατρέπονται σε μια εκτελέσιμη μορφή: είτε κώδικα μηχανής (πχ. η C και η C++) είτε ενδιάμεσης μορφής που χρειάζεται να διερμηνευτεί για να εκτελεστεί (πχ. η java και η C#). Υπάρχουν επίσης και συνδυασμοί των παραπάνω μοντέλων όπως της java, που συνδυάζουν τη μεταγλώττιση με τη διερμηνεία, χρησιμοποιώντας ταυτόχρονα και τη μέθοδο just-in-time compilation για τη βελτίωση της ταχύτητας της διερμηνείας.

Σημαντικές γλώσσες στον προγραμματισμό τέλος είναι και οι γλώσσες σήμανσης (markup languages). Μια γλώσσα σήμανσης είναι ένα μοντέρνο σύστημα προσθήκης σημάνσεων σε κείμενο με τέτοιο τρόπο ώστε να είναι συντακτικά ευδιάκριτες από το κείμενο στο οποίο προστίθενται. Οι γλώσσες σήμανσης δεν είναι γλώσσες προγραμματισμού ωστόσο χρησιμοποιούνται ευρέως σε μια μεγάλη γκάμα εφαρμογών εξαιτίας της περιγραφικής δύναμης τους. Οι σημάνσεις που μπορούν να προστεθούν σε ένα κείμενο, μπορούν να είναι α) παρουσιαστικές (presentational markup) δηλαδή να χρησιμοποιούνται για να προσδιορίσουν το παρουσιαστικό του κειμένου στο οποίο αναφέρονται (πχ. η HTML), β) διαδικαστικές (procedural markup) για να παρέχουν οδηγίες σε άλλα προγράμματα για το πώς να επεξεργάζονται το κείμενο που περιέχουν (πχ. η TEX) και γ) περιγραφικές (descriptive markup) που απλώς παρέχουν μια ετικέτα στο κείμενο στο οποίο αναφέρονται.

Διαδικασία ανάπτυξης λογισμικού

Ένας απλός ορισμός που αποδίδει ικανοποιητικά την έννοια του προγραμματισμού υπολογιστών είναι η ικανότητα της επικοινωνίας του προγραμματιστή με τον υπολογιστή σε μια αμοιβαίως αντιληπτή γλώσσα προγραμματισμού για την πραγματοποίηση χρήσιμων εργασιών. Ο προγραμματισμός ωστόσο αποτελεί μόνο ένα κομμάτι της ανάπτυξης λογισμικού. Κατά συνέπεια ένας προγραμματιστής είναι γενικά χρήσιμο να γνωρίζει τα στάδια που ακολουθούνται στην ανάπτυξη έργων λογισμικού. Η ανάπτυξη λογισμικού περιλαμβάνει τις διαδικασίες της ανάλυσης απαιτήσεων, του σχεδιασμού, της γραφής, της δοκιμής, της αποσφαλμάτωσης και της διατήρησης του πηγαίου κώδικα των προγραμμάτων γραμμένων σε μια γλώσσα ή και περισσότερες. Ένας προγραμματιστής είναι καλό να κατανοεί τις μεθόδους και τις πρακτικές που εφαρμόζονται σε κάθε μια από τις παραπάνω διαδικασίες, δίνοντας έμφαση κατ' εξοχήν στη γραφή κώδικα. Ωστόσο ο προγραμματιστής δεν περιορίζεται μόνο στις παραπάνω διαδικασίες.

Συχνά απαιτείται και η εξειδίκευση σε πολλαπλές διαφορετικές πρακτικές ή θεωρητικές έννοιες, όπως οι αλγόριθμοι και η λογική .

Ανάλυση απαιτήσεων

Η επιτυχία οποιουδήποτε έργου λογισμικού εξαρτάται σε μεγάλο βαθμό από τον αρχικό εντοπισμό των απαιτήσεων του υπό ανάπτυξη συστήματος τόσο μέσα από τη λήψη των ζητούμενων των πελατών, όσο και από την ανάδειξη των διαφόρων υπο-προβλημάτων που μπορεί να προκύψουν. Το πρώτο βήμα στην ανάπτυξη ενός έργου λογισμικού είναι η ανάλυση απαιτήσεων (requirement analysis). Το αποτέλεσμα του πρώτου αυτού βήματος είναι προδιαγραφές τεχνολογικές, επιχειρηματικές, λειτουργικές, κόστους και χρονικής διάρκειας.

Σχεδιασμός λογισμικού

Η διαδικασία της γραφής κώδικα, είναι η μετατροπή του σχεδιασμού και γενικότερα της λογικής του λογισμικού σε κώδικα μιας γλώσσας προγραμματισμού. Η γραφή κώδικα απαιτεί πλήρη γνώση της χρησιμοποιούμενης γλώσσας προγραμματισμού καθώς η ελλιπής γνώση μπορεί να οδηγήσει σε άσκοπο γράψιμο κώδικα που υπάρχει ήδη ή την επιλογή λανθασμένων μονοπατιών υλοποίησης. Σε μεγάλα έργα προγραμματισμού υιοθετούνται συχνά ένα σύνολο προτύπων, κατευθύνσεων ή συμβάσεων γραφής με σκοπό τη βελτίωση της ποιότητας, της αναγνωσιμότητας και της ευκολίας της συντήρησης του κώδικα. Τα πρότυπα αυτά περιλαμβάνουν κανόνες ονομασίας και οργάνωσης των αρχείων ή των λέξεων στον κώδικα, τη δόμηση του κώδικα σε μπλοκ με εσοχές για να τονίζουν το επίπεδο ένθεσης του ή τη μορφοποίηση των διαφόρων εντολών. Τα πρότυπα κωδικογράφησης GNU, το στυλ του πυρήνα BSD και οι συμβάσεις κώδικα στη Java.

Αποσφαλμάτωση

Η γραφή κώδικα είναι μια ανθρώπινη διαδικασία και κατά συνέπεια είναι υποκείμενη σε σφάλματα που μπορεί να οφείλονται σε παραλήψεις ή απρόβλεπτους παράγοντες. Η αποσφαλμάτωση (debugging) είναι η μεθοδική διαδικασία της εύρεσης και του περιορισμού των σφαλμάτων (bugs) και των ελαττωμάτων στον κώδικα ενός προγράμματος. Γενικά αποτελείται από τα βήματα της αναπαραγωγής του προβλήματος, της απομόνωσης του σημείου που εμφανίζεται το σφάλμα, της αναγνώρισης της αιτίας που το προκαλεί, της διόρθωσης του σφάλματος και της επιβεβαίωσης της απαλοιφής του προβλήματος. Η αποσφαλμάτωση είναι μια αρκετά επίπονη διαδικασία και για το λόγο αυτό αναπτύχθηκαν ειδικά προγράμματα για το σκοπό αυτό : οι αποσφαλματωτές (debuggers).

Έλεγχος λογισμικού

Στη διαδικασία του ελέγχου λογισμικού βεβαιώνεται ότι το αποτέλεσμα της συγγραφής πληροί τις προδιαγραφές όπως αυτές ορίστηκαν στην ανάλυση και το σχεδιασμό, ότι δουλεύει όπως αναμενόταν και ότι μπορεί να εγκατασταθεί για εφαρμογή με τα ίδια χαρακτηριστικά. Υπάρχουν διάφορες μέθοδοι ελέγχου που μπορούν να γίνουν σε διαφορετικά επίπεδα, ανάλογα με την πρόσβαση στο κάθε τμήμα κώδικα. Ο έλεγχος μπορεί να φτάσει στα θεμελιώδη τμήματα του κώδικα ή να αντιμετωπίσει την εφαρμογή στην ολότητα της. Ο έλεγχος μπορεί να

πραγματοποιείται και για μεμονωμένα τμήματα κώδικα κατά την ανάπτυξη του αλλά παραδοσιακά γίνεται μετά την ολοκλήρωση της ανάπτυξης του.

Συντήρηση κώδικα

Ακόμα και μετά την αποσφαλμάτωση και τον έλεγχο του λογισμικού μπορούν να εντοπιστούν προβλήματα στο σχεδιασμό ή την υλοποίηση ή ακόμα και να προστεθούν νέες απαιτήσεις. Επειδή η αλλαγή είναι έμφυτη στην ανάπτυξη λογισμικού, η ανάγκη του ελέγχου της οργάνωσης και της εξέλιξης της ανάπτυξης του κώδικα είναι επιτακτική, ιδιαίτερα για τα έργα μεγάλης κλίμακας. Η διαδικασία της συντήρησης κώδικα (code maintenance) είναι ουσιαστικά ο έλεγχος εκδόσεων (revision control) του πηγαίου κώδικα και περιλαμβάνει τον έλεγχο της εξέλιξης του μέσα στο χρόνο, κρατώντας ένα χρονικό για το σκεπτικό πίσω από τις υφιστάμενες αλλαγές. Υπάρχει μια πληθώρα συστημάτων ελέγχου εκδόσεων (revision control systems) όπως τα CVS και Subversion (SVN) που χρησιμοποιούνται κατά κόρον σε έργα ανοιχτού κώδικα και τον Team Foundation Server για έργα στην πλατφόρμα NET.

Οι αρχές προγραμματισμού είναι ανεξάρτητες της γλώσσας που χρησιμοποιείται. Κάθε γλώσσα έχει τα δυνατά και αδύναμα σημεία της, καθιστώντας έτσι την ανάπτυξη κάποιων προγραμμάτων ευκολότερη ή δυσκολότερη σε αυτές. Ένας πεπειραμένος προγραμματιστής θα βρει σχετικά εύκολη η μετάβαση από μια γλώσσα σε μια άλλη καθώς το ουσιαστικότερο πράγμα που πρέπει να κάνει είναι να εντοπίσει πως υλοποιούνται οι προγραμματιστικές αρχές στη νέα γλώσσα. Τα περισσότερα μαθήματα προγραμματισμού επικεντρώνονται στο γράψιμο των προγραμμάτων από το μηδέν. Ωστόσο αρκετά μεγάλο ποσοστό της ανάπτυξης λογισμικού καταναλώνεται αφού έχει γραφεί ο κώδικας και πιο συγκεκριμένα στην ανάγνωση, την κατανόηση και την τροποποίηση του αρχικού κώδικα.

Η εκμάθηση της ανάγνωσης κώδικα είναι επίσης μια απαραίτητη ικανότητα για έναν προγραμματιστή ιδιαίτερα σήμερα, στην εποχή της επαναχρησιμοποίησης κώδικα και της συνεργατικής ανάπτυξης σε απαιτητικά εταιρικά περιβάλλοντα. Το λογισμικό ανοιχτού κώδικα παρέχει πρόσβαση σε εκατομμύρια γραμμές κώδικα για ανάγνωση, κριτική, τη δυνατότητα βελτίωσης από τον πεπειραμένο χρήστη και εκμάθησης για τον αρχάριο. Για το λόγο αυτό είναι καλή πρακτική η ανάγνωση ποιοτικού κώδικα που έχει γραφτεί για λογισμικό ανοιχτού κώδικα, καθώς έχει συζητηθεί, συγκεκριμενοποιηθεί, γενικευτεί και τεκμηριωθεί από αρκετούς και κυρίως έμπειρους προγραμματιστές. Η ανάγνωση κώδικα επομένως είναι πιθανό να ανεβάσει το αισθητικό κριτήριο του προγραμματιστή για τον κώδικα που παράγει, όπως ακριβώς η ανάγνωση ποιοτικού πεζού λόγου εμπλουτίζει το λεξιλόγιο, ενεργοποιεί τη φαντασία και διευρύνει το νου του αναγνώστη.

Πρότυπα Προγραμματισμού

Για να ξεκινήσει κανείς τη μελέτη της C# και του αντικειμενοστραφούς προγραμματισμού τον οποίο αυτή ακολουθεί, ¹²πρέπει να κατανοήσει τη γενικότερη φιλοσοφία και τα χαρακτηριστικά του. Ταυτόχρονα θα ήταν σκόπιμο να τον συγκρίνει με τα κυριότερα προγραμματιστικά πρότυπα που υπάρχουν.

Ένα προγραμματιστικό πρότυπο

είναι μια προσέγγιση για τον προγραμματισμό υπολογιστών που βασίζεται σε κάποια μαθηματική θεωρία ή ένα αυστηρό σύνολο κανόνων. Κάθε πρότυπο περιέχει ένα σύνολο εννοιών που το καθιστούν ιδανικό για την επίλυση συγκεκριμένου είδους προβλημάτων. Για παράδειγμα, ο αντικειμενοστραφής προγραμματισμός είναι ιδανικός για προβλήματα με ένα μεγάλο πλήθος αφαιρέσεων δεδομένων, οργανωμένων σε μια ιεραρχία ενώ ο λογικός προγραμματισμός είναι ιδανικός για το μετασχηματισμό πολύπλοκων συμβολικών δομών σύμφωνα με κανόνες λογικής. Υπάρχουν περίπου 27 διαφορετικά προγραμματιστικά πρότυπα, αρκετά από τα οποία ωστόσο έχουν κοινά στοιχεία μεταξύ τους.

Τα προγραμματιστικά πρότυπα δεν πρέπει να συγχέονται με τις τεχνικές προγραμματισμού οι οποίες σχετίζονται με τις αλγοριθμικές ιδέες για την επίλυση προβλημάτων κάποιας συγκεκριμένης τάξης, ούτε και με το στυλ προγραμματισμού, δηλαδή την αποτύπωση της προσωπικής έκφρασης του προγραμματιστή μέσα στον κώδικα.

Η επίλυση ενός προγραμματιστικού προβλήματος απαιτεί σωστές προσεγγίσεις. Ακόμα και τα πιο απλά προβλήματα ενδέχεται να χρειάζονται διαφορετική προσέγγιση σε διάφορα μέρη τους. Για το λόγο αυτό, είναι ιδανικό μια γλώσσα προγραμματισμού να υποστηρίζει περισσότερα από ένα προγραμματιστικά πρότυπα. Κατά πολλούς είναι ατυχές το γεγονός ότι δημοφιλείς γλώσσες όπως η Java ή η C++ υποστηρίζουν ένα ή δύο πρότυπα μόνο. Η υποστήριξη πολλαπλών προτύπων σε μια γλώσσα επιτρέπει στους προγραμματιστές να επιλέξουν τις σωστές έννοιες όταν τις χρειάζονται άμεσα. Οι γλώσσες που υποστηρίζουν πολλαπλά πρότυπα συχνά καλούνται «πολυπρότυπες». Είναι δυνατόν ωστόσο να υλοποιηθεί, εν μέρει ή και πλήρως, ένα πρότυπο σε μια γλώσσα που δεν το υποστηρίζει, αφού αυτό έχει κατανοηθεί πλήρως. Για παράδειγμα ο συναρτησιακός προγραμματισμός είναι δυνατόν να υποστηριχθεί στην C#.

Προστακτικός προγραμματισμός

Στον προστακτικό προγραμματισμό ένα πρόβλημα επιλύεται υπολογιστικά μέσω προτάσεων που αλλάζουν την κατάσταση του προγράμματος. Τα προστακτικά προγράμματα ορίζουν μια ακολουθία εντολών προς εκτέλεση από τον υπολογιστή. Εδώ ο προγραμματιστής γράφει κώδικα που περιγράφει τα ακριβή βήματα που πρέπει να ακολουθήσει ένα πρόγραμμα για να πετύχει το στόχο του. Επομένως, ο κυριότερος στόχος του προστακτικού προγραμματισμού είναι να δεχθεί δεδομένα σαν είσοδο, να τα αλλάξει μέσω ενημερώσεων στη μνήμη και να παράγει έξοδο βασισμένη σε αυτά τα συνεχώς αλλασώμενα δεδομένα. Κάθε πρόγραμμα έχει μια αρχική κατάσταση, μια λίστα υπολογιστικών βημάτων που πρέπει να ολοκληρώσει σε μια σειρά που ελέγχεται από τις δομές ελέγχου και ένα σημείο εξόδου. Οι συνηθέστερες εντολές που προσφέρονται από τις προστακτικές γλώσσες προγραμματισμού αφορούν την ανάθεση τιμών και την είσοδο/έξοδο. Οι γνωστότερες γλώσσες που ακολουθούν αυτό το παράδειγμα είναι η C, η Basic, η Pascal και η Fortran.

Συναρτησιακός Προγραμματισμός

Στο συναρτησιακό προ¹³γραμματισμό ο στόχος του προγράμματος είναι ελαφρώς διαφορετικός και αυτό έχει επίπτωση στον τρόπο προγραμματισμού : δίνεται έμφαση στη μορφή της πληροφορίας που είναι επιθυμητή και στους μετασχηματισμούς που χρειάζονται για να

φτάσει το πρόγραμμα σε αυτή. Το επίκεντρο εδώ είναι οι κλήσεις συναρτήσεων. Ο κυριότερος τρόπος αλλαγής των δεδομένων είναι οι συναρτήσεις και το πέρασμα των παραμέτρων.

Τα δεδομένα δεν αλλάζουν στο συναρτησιακό προγραμματισμό και οι νέες τιμές δημιουργούνται εκ νέου από παλιότερες. Κατά συνέπεια δεν υπάρχει η κλασική κατάσταση προγράμματος. Επίσης η διαφορά μεταξύ του προγράμματος και των δεδομένων εξαφανίζεται. Μια συνάρτηση είναι δεδομένα όπως κάθε άλλο στοιχείο δεδομένων. Για παράδειγμα οι ακέραιοι αριθμοί και οι συναρτήσεις είναι το ίδιο. Τα στατικά δεδομένα μπορεί να υπάρχουν αλλά παίζουν μικρό ρόλο στο συναρτησιακό προγραμματισμό.

Εδώ η στοιβία αποκτά μεγαλύτερη σημασία καθώς η περισσότερη δουλειά πραγματοποιείται με την κλήση συναρτήσεων. Στις συναρτησιακές γλώσσες λαμβάνεται μεγαλύτερη μέριμνα στην απόκρυψη του σωρού από τον προγραμματιστή έτσι ώστε να μην είναι διαθέσιμος για επεμβάσεις. Τα δεδομένα είναι δυναμικά αλλά μόλις δημιουργούνται στο σωρό, αν ακολουθείται συναρτησιακό μοντέλο αυστηρά δεν μπορούν να αλλάξουν. Ωστόσο αρκετές γλώσσες παρακάμπτουν αυτόν τον κανόνα για να αντιμετωπίσουν θέματα απόδοσης. Μερικές γλώσσες που ακολουθούν αυτό το παράδειγμα είναι η Clojure, η Erlang, η Haskell και F#.

Λογικός Προγραμματισμός

Ο λογικός προγραμματισμός είναι αρκετά διαφορετικός από τα προηγούμενα πρότυπα. Εδώ ο προγραμματιστής δεν γράφει ένα πρόγραμμα με την παραδοσιακή του έννοια αλλά ένα σύνολο από λογικούς κανόνες που πρέπει να ισχύουν. Από αυτό το σύνολο το πρόγραμμα προσπαθεί να απαντήσει ερωτήσεις με ένα ναι ή όχι. Κατά συνέπεια, η σειρά δήλωσης των κανόνων επηρεάζει την έξοδο του προγράμματος. Στο λογικό προγραμματισμό υπάρχει η έννοια του σωρού καθώς οι κανόνες αποτιμώνται κατά την εκτέλεση του προγράμματος και αυτή προϋποθέτει την δυναμική προσθήκη ή αφαίρεση κανόνων κατά την εκτέλεση.

Ο λογικός προγραμματισμός είναι ιδιαίτερα χρήσιμος στο πεδίο της Τεχνητής Νοημοσύνης όπου μπορεί να χρησιμοποιηθεί για την αναπαράσταση γνώσης με τη μορφή ενός λογικού προγράμματος. Η γνωστότερη γλώσσα στο λογικό προγραμματισμό είναι η Prolog.

Αντικειμενοστραφής Προγραμματισμός

Ο αντικειμενοστραφής προγραμματισμός καθιερώθηκε τη δεκαετία του 1990, η γλώσσα που συνεισέφερε αρκετά σε αυτό ήταν η Java, αντικαθιστώντας σε μεγάλο βαθμό το παραδοσιακό πρότυπο του προστακτικού προγραμματισμού. Εδώ το βασικό στοιχείο του προγράμματος είναι το αντικείμενο που συνδυάζει ιδιότητες και διαδικασίες που επενεργούν σε αυτές αλλάζοντας τα δεδομένα τους. Κάθε αντικείμενο αποτελεί δεσμευμένο κομμάτι της μνήμης (και συγκεκριμένα του σωρού) και καλείται και στιγμιότυπο, ενός σύνθετου τύπου δεδομένων που καλείται κλάση. Η κλάση είναι μια αυτοτελής και αφαιρετική αναπαράσταση κάποιας κατηγορίας αντικειμένων, είτε φυσικών αντικειμένων του προγραμματικού κόσμου είτε νοητών, εννοιολογικών αντικειμένων και στην ουσία αποτελεί την προδιαγραφή των δεδομένων και των διαδικασιών που επιδρούν πάνω σε αυτά. 14

Αν κανείς παρατηρήσει πως υλοποιούνται οι διάφορες εργασίες στον πραγματικό κόσμο, θα παρατηρήσει ότι αλληλοεπιδρά σε έναν κόσμο που στρέφεται γύρω από αντικείμενα. Αν θέλει κάποιος να πάει σε ένα μέρος για παράδειγμα μπορεί να χρησιμοποιήσει το αυτοκίνητο του. Το

αντικείμενο αυτοκίνητο αποτελείται από άλλα αντικείμενα που αλληλοεπιδρούν μεταξύ τους και κάνουν εφικτή τη μεταφορά μας. Ο οδηγός απομονώνεται από τις διαδικασίες αυτές και το πώς αλληλοεπιδρούν για να ξεκινήσει το αυτοκίνητο. Το μόνο που περιμένει είναι το αυτοκίνητο να ξεκινήσει και να διανύσει την διαδρομή που ο ίδιος επιθυμεί. Παρόμοια, αποκρύπτεται από τους τελικούς χρήστες ενός λογισμικού η λογική που χρειάζεται για να υλοποιηθεί μια διεργασία. Για παράδειγμα, όταν εκτυπώνει ένας χρήστης μια σελίδα από ένα πρόγραμμα επεξεργασίας κειμένου, ξεκινάει μια διαδικασία πατώντας το κουμπί της εκτύπωσης και χωρίς να ξέρει τι γίνεται εσωτερικά στο πρόγραμμα, περιμένει να δει μια εκτυπωμένη σελίδα ή ένα μήνυμα αποτυχίας.

Οι αιτίες που ώθησαν στην ανάπτυξη του αντικειμενοστραφούς προγραμματισμού είναι η ευκολία συντήρησης, οργάνωσης, χειρισμού και επαναχρησιμοποίησης κώδικα μεγάλων και πολύπλοκων εφαρμογών και επικράτησε επειδή μπορούσε να ανταπεξέλθει σε προγράμματα πολύ μεγαλύτερου όγκου και πολυπλοκότητας. Σε αυτό συνετέλεσαν τα ακόλουθα χαρακτηριστικά του αντικειμενοστραφούς προγραμματισμού :

Ενθυλάκωση

Η ιδιότητα της ενθυλάκωσης αναφέρεται στην απόκρυψη των δεδομένων και της υλοποίησης ενός αντικειμένου και είναι γνωστή και ως απόκρυψη δεδομένων. Τα δεδομένα και ο κώδικας όταν ενθυλακώνονται αποκρύπτονται από την εξωτερική πρόσβαση. Όταν ένας εξωτερικός παρατηρητής βλέπει ένα αντικείμενο, μπορεί να το χρησιμοποιήσει μόνο μέσω της εξωτερικής διασύνδεσης του (interface), δηλαδή ότι είναι διαθέσιμο προς χρήση, ενώ οι εσωτερικές λεπτομέρειες είναι κρυμμένες και δεν μπορούν να χρησιμοποιηθούν. Έτσι τα ενθυλακωμένα δεδομένα δεν μπορούν να αλλάξουν και πιο συγκεκριμένα να ενημερωθούν άμεσα από εξωτερικά αντικείμενα.

Αφαίρεση δεδομένων

Η αφαίρεση δεδομένων είναι η αρχή που απαιτεί τα προγράμματα να μην λαμβάνουν υπόψη συγκεκριμένες υλοποιήσεις ή εσωτερικές αναπαραστάσεις αλλά να έχουν μια γενική αφηρημένη ιδέα των αντικειμένων που χρησιμοποιούν. Στον αντικειμενοστραφή προγραμματισμό κάθε αντικείμενο μπορεί να ζητηθεί να παρέχει δεδομένα ή υπηρεσίες λαμβάνοντας μηνύματα/αιτήσεις. Τον αιτούντα δεν τον απασχολεί πώς το αντικείμενο θα παράγει τα δεδομένα αλλά τα δεδομένα καθαυτά. Το πώς δηλαδή η υλοποίηση ή αλλιώς η δημιουργία της πληροφορίας είναι μια εσωτερική για τον αντικείμενο υπόθεση. Επομένως δίνεται έμφαση στο τι αντί για το πώς κατά την αλληλεπίδραση μεταξύ δύο αντικειμένων. Η αφαίρεση δεδομένων προσπαθεί να μειώσει τις λεπτομέρειες και να τις ομαδοποιήσει σε επίπεδα μιας υλοποίησης έτσι ώστε ο προγραμματιστής να επικεντρωθεί στις έννοιες που τον ενδιαφέρουν για την επίλυση ενός προβλήματος. Ταυτόχρονα ο προγραμματιστής δεν απασχολείται με την εκμάθηση συγκεκριμένων υλοποιήσεων, αλλά επικεντρώνεται στην επίλυση του δεδομένου προβλήματος, γνωρίζοντας μόνο ότι χρειάζεται. Ένα σύστημα μπορεί να έχει πολλαπλά επίπεδα αφαίρεσης, σε καθένα από τα οποία φανερώνονται διαφορετικές λεπτομέρειες.

Κληρονομικότητα

Τα αντικείμενα τείνουν να δομούνται βασισμένα σε άλλα αντικείμενα. Όταν δημιουργείται ένα νέο αντικείμενο (μέσα από μια κλάση), ουσιαστικά ορίζονται οι ιδιότητες που το κάνουν

ξεχωριστό από τα υπόλοιπα αντικείμενα. Εξαιτίας της ομοιότητας πολλών αντικειμένων υπάρχει ένας μηχανισμός για τη μεταφορά των ιδιοτήτων κάποιων αντικειμένων σε άλλα αντικείμενα έτσι ώστε να αποφεύγεται ο άσκοπος ορισμός των χαρακτηριστικών σε κάθε αντικείμενο ξεχωριστά, αλλά και για να υπάρχει μια νοητή συγγένεια μεταξύ των αντικειμένων. Αυτός ο μηχανισμός καλείται κληρονομικότητα. Η κληρονομικότητα επιτρέπει στους προγραμματιστές την επαναχρησιμοποίηση των ορισμών δομών που έχουν οριστεί στο παρελθόν, μειώνοντας έτσι την εργασία κατά την ανάπτυξη προγραμμάτων.

Πολυμορφισμός

Ο όρος πολυμορφισμός στην κυριολεξία αναφέρεται στην ιδιότητα του να έχει κάτι πολλαπλές μορφές. Στις γλώσσες προγραμματισμού τις περισσότερες φορές αναφέρεται στην ιδιότητα των μεθόδων να μπορούν να δέχονται και/ή να επιστρέφουν τιμές διαφορετικού τύπου. Για παράδειγμα μια μέθοδος που δέχεται ένα μόνο όρισμα είναι πολυμορφική αν μπορεί να δεχτεί παραμέτρους διαφορετικού τύπου. Ο πολυμορφισμός είναι διάχυτος στις αντικειμενοστραφείς γλώσσες προγραμματισμού και συνδέεται αρκετά με την κληρονομικότητα.

Βάσεις Δεδομένων

Η αλματώδης ανάπτυξη της επιστήμης της πληροφορικής και των επικοινωνιών τα τελευταία χρόνια έχει καταστήσει την πληροφορία ως ένα από τα βασικότερα και πολυτιμότερα αγαθά. Για την διαχείριση αυτού του τεράστιου όγκου πληροφοριών χρησιμοποιούμε συστήματα βάσεων δεδομένων που μας επιτρέπουν την αποθήκευση την επεξεργασία αλλά και την αποδοτική εκμετάλλευση της πληροφορίας.

Μια βάση δεδομένων είναι ένα σύνολο αρχείων με υψηλό βαθμό οργάνωσης τα οποία είναι συνδεδεμένα μεταξύ τους με λογικές σχέσεις, έτσι ώστε να μπορούν να χρησιμοποιούνται από πολλές εφαρμογές και από πολλούς χρήστες ταυτόχρονα. Τα δεδομένα που υπάρχουν στις βάσεις δεδομένων πρέπει να είναι :

- Ολοκληρωμένα, δηλαδή τα δεδομένα πρέπει να είναι αποθηκευμένα σε ομοιόμορφα οργανωμένα σύνολα αρχείων όπου δεν πρέπει να υπάρχει επανάληψη ή πλεονασμός των ίδιων στοιχείων.
- Καταμεριζόμενα. Να μπορούν δηλαδή περισσότεροι του ενός χρήστες να βλέπουν και να μοιράζονται τα ίδια δεδομένα την ίδια χρονική στιγμή.

Υπάρχει ένα ειδικό λογισμικό το ¹⁶οποίο μεσολαβεί ανάμεσα στα αρχεία δεδομένων και τις εφαρμογές που χρησιμοποιούν οι χρήστες και αποκαλείται Σύστημα Διαχείρισης Βάσης Δεδομένων ή DBMS (Data Base Management System). Το DBMS είναι στην ουσία ένα σύνολο

από προγράμματα και υπορουτίνες που έχουν να κάνουν με τον χειρισμό της βάσης δεδομένων, όσον αφορά τη δημιουργία, τροποποίηση, διαγραφή στοιχείων, με ελέγχους ασφάλειας κ.λπ.

Οι χρήστες των εφαρμογών αντλούν τα στοιχεία που τους ενδιαφέρουν από τη βάση δεδομένων χωρίς να είναι σε θέση να γνωρίζουν με ποιο τρόπο είναι οργανωμένα τα δεδομένα σ αυτήν. Το DBMS παίζει τον ρόλο του μεσάζοντα ανάμεσα στον χρήστη και τη βάση δεδομένων και μόνο μέσω του DBMS μπορεί ο χρήστης να αντλήσει πληροφορίες από τη βάση δεδομένων.

Ένα Σύστημα Βάσης Δεδομένων αποτελείται από το υλικό, το λογισμικό τη βάση δεδομένων και τους χρήστες. Είναι δηλαδή ένα σύστημα με το οποίο μπορούμε να αποθηκεύσουμε και να αξιοποιήσουμε δεδομένα με τη βοήθεια ηλεκτρονικού υπολογιστή. Αναλυτικά :

- Το υλικό αποτελείται όπως είναι γνωστό από τους ηλεκτρονικούς υπολογιστές, τα περιφερειακά, τους σκληρούς δίσκους κ.α. , όπου είναι αποθηκευμένα τα αρχεία της βάσης δεδομένων αλλά και τα προγράμματα που χρησιμοποιούνται για την επεξεργασία τους.
- Το λογισμικό είναι τα προγράμματα που χρησιμοποιούνται για την επεξεργασία των δεδομένων της βάσης.
- Η βάση δεδομένων αποτελείται από το σύνολο των αρχείων όπου είναι αποθηκευμένα τα δεδομένα του συστήματος. Τα στοιχεία αυτά μπορεί να βρίσκονται αποθηκευμένα σ' έναν φυσικό υπολογιστή αλλά και σε περισσότερους. Όμως στον χρήστη δίνεται η εντύπωση ότι βρίσκονται συγκεντρωμένα στο ίδιο υπολογιστή. Τα δεδομένα των αρχείων αυτών είναι ενοποιημένα και μερισμένα. Ο κάθε χρήστης έχει διαφορετικά δικαιώματα και βλέπει διαφορετικό κομμάτι της βάσης δεδομένων, ανάλογα με τον σκοπό για τον οποίο συνδέεται.
- Οι χρήστες μια βάσης δεδομένων χωρίζονται στις εξής κατηγορίες :
- Τελικοί χρήστες. Χρησιμοποιούν κάποια εφαρμογή για να παίρνουν στοιχεία από μια βάση δεδομένων, έχουν τις λιγότερες δυνατότητες επέμβασης στα στοιχεία της βάσης δεδομένων, χρησιμοποιούν ειδικούς κωδικούς πρόσβασης και το σύστημα τους επιτρέπει ανάλογη πρόσβαση σε συγκεκριμένο κομμάτι της βάσης δεδομένων.
- Προγραμματιστές εφαρμογών. Αναπτύσσουν τις εφαρμογές του DBMS σε κάποια από τις γνωστές γλώσσες προγραμματισμού.
- Διαχειριστής δεδομένων. Έχει τη διοικητική αρμοδιότητα και ευθύνη για την οργάνωση της βάσης δεδομένων και την απόδοση δικαιωμάτων πρόσβασης στους χρήστες.
- Διαχειριστής βάσης δεδομένων. Λαμβάνει οδηγίες από τον διαχειριστή δεδομένων και είναι αυτός που διαθέτει τις τεχνικές γνώσεις και αρμοδιότητες για τη σωστή και αποδοτική λειτουργία του DBMS.

Μοντέλα βάσεων δεδομένων

Υπάρχουν τρία βασικά μοντέλα που έχουν επικρατήσει στις βάσεις δεδομένων, το ιεραρχικό, το δικτυωτό και το σχεσιακό, και τα οποία αναπτύχθηκαν με βάση αντίστοιχες δομές. Το

ιεραρχικό μοντέλο Το ιεραρχικό μοντέλο έχει μια ιεραρχική δομή που θυμίζει δένδρο. Οι οντότητες μοιάζουν με απολήξεις από κλαδιά δένδρων και τοποθετούνται σε επίπεδα ιεραρχίας. Τα κλαδιά παριστάνουν τις συσχετίσεις ανάμεσα στις οντότητες. Από μια οντότητα που βρίσκεται σ' ένα ανώτερο επίπεδο εκκινούν πολλά κλαδιά, καθένα από τα οποία καταλήγει σε μια οντότητα που βρίσκεται σ' ένα χαμηλότερο επίπεδο. Αλλά, σε κάθε οντότητα που βρίσκεται σ' ένα χαμηλότερο επίπεδο αντιστοιχεί μια και μόνο μια οντότητα που βρίσκεται σ' ένα ανώτερο επίπεδο. Το μοντέλο αυτό ήταν το πρώτο που εμφανίσθηκε αλλά σήμερα θεωρείται δύσχρηστο και ξεπερασμένο.

Το δικτυωτό μοντέλο βάσεων δεδομένων

Και στο δικτυωτό μοντέλο, τα στοιχεία τοποθετούνται σε ένα επίπεδο ιεραρχίας, αλλά κάθε στοιχείο μπορεί να συσχετισθεί με πολλά στοιχεία είτε σε ένα κατώτερο ή σ' ένα ανώτερο επίπεδο.

Το σχεσιακό μοντέλο βάσεων δεδομένων

Το σχεσιακό μοντέλο έχει επικρατήσει σήμερα στην αναπαράσταση των δεδομένων καθώς διαθέτει σημαντικά πλεονεκτήματα ως προς τα άλλα δύο και οι βάσεις δεδομένων που σχεδιάζονται σύμφωνα μ' αυτό αποκαλούνται σχεσιακές. Με τις σχεσιακές βάσεις δεδομένων διαθέτουμε έναν σαφή, απλό και εύκολα κατανοητό τρόπο για να μπορέσουμε να αναπαραστήσουμε και να διαχειριστούμε τα δεδομένα μας. Υστερούν μόνο σε ταχύτητα υπολογισμών και σε χώρο αποθήκευσης, αλλά μόνο όταν έχουμε να κάνουμε με πολύ μεγάλες βάσεις δεδομένων. Στο μοντέλο αυτό οι βάσεις δεδομένων περιγράφονται με αυστηρές μαθηματικές έννοιες και ο χρήστης βλέπει τις οντότητες και τις συσχετίσεις με την μορφή πινάκων και σχέσεων αντίστοιχα. Ένας πίνακας αποτελείται από γραμμές και στήλες, όπου τοποθετούμε τα στοιχεία σε οριζόντια και κάθετη μορφή. Η κάθε στήλη του πίνακα χαρακτηρίζει κάποια ιδιότητα της οντότητας και αποκαλείται χαρακτηριστικό ή πεδίο, ενώ η κάθε γραμμή του πίνακα περιέχει όλες τις πληροφορίες που αφορούν ένα στοιχείο της οντότητας και αποκαλείται πλειάδα ή εγγραφή. Κάθε πεδίο του πίνακα μπορεί να πάρει ορισμένες μόνο τιμές, οι οποίες μπορεί να καθορίζονται από τον τύπο δεδομένων της ιδιότητας, όπως ονόματα ή αριθμοί για παράδειγμα, ή και από αυτό που εκφράζει, όπως το ότι δεν μπορούμε να έχουμε αρνητικό βάρος ή αρνητικό ΑΦΜ, για παράδειγμα. Το σύνολο των αποδεκτών τιμών μια οντότητας αποκαλείται πεδίο ορισμού. Όπως είναι εύκολα κατανοητό, η βασικότερη εργασία που έχουμε να κάνουμε κατά τον σχεδιασμό μια σχεσιακής βάσης δεδομένων είναι να ορίσουμε τους πίνακες που θα χρησιμοποιήσουμε καθώς και τα πεδία που θα περιέχει ο καθένας απ' αυτούς. Η διαδικασία αυτή αποκαλείται κατασκευή του σχήματος μιας βάσης δεδομένων.

18

Οι κανόνες που πρέπει να ακολουθήσουμε πιστά κατά τον σχεδιασμό μιας σχεσιακής βάσης δεδομένων είναι οι εξής :

- Η κάθε οντότητα πρέπει να παριστάνεται ως ένας ξεχωριστός πίνακας

- Η κάθε στήλη του πίνακα αντιστοιχεί σε μια ιδιότητα της οντότητας.
- Η κάθε γραμμή του πίνακα αντιστοιχεί σε μια εμφάνιση της οντότητας.
- Η κάθε γραμμή πρέπει να είναι μοναδική, δηλαδή αποκλείεται να υπάρχουν δύο ή και περισσότερες γραμμές που να περιέχουν τα ίδια ακριβώς στοιχεία.
- Η σειρά εμφάνισης των γραμμών δεν έχει καμία σημασία.
- Η κάθε στήλη έχει μια δική της μοναδική ονομασία.
- Οι τιμές που ανήκουν στην ίδια στήλη πρέπει να είναι του ίδιου τύπου δηλαδή ή όλες αριθμοί ή όλες αλφαριθμητικές.
- Η στήλη που αποτελεί το πρωτεύον κλειδί μιας οντότητας, δεν πρέπει να είναι ποτέ κενή. Αποκλείεται να υπάρχουν δύο ή και περισσότερες γραμμές που να περιέχουν την ίδια τιμή στο πρωτεύον κλειδί.
- Το πρωτεύον κλειδί μιας οντότητας αποκαλείται ξένο κλειδί σε μια άλλη οντότητα, με την οποία υπάρχει συσχετισμός.
- Μπορεί να υπάρχουν πολλές γραμμές που να έχουν την ίδια τιμή στο ξένο κλειδί.

Οι γλώσσες προγραμματισμού και οι βάσεις δεδομένων αποτελούν ένα μεγάλο κομμάτι της πτυχιακής μου εργασίας. Αφού κατανοήσαμε τις βασικές έννοιες και τους τρόπους λειτουργίας τους, στο επόμενο κεφάλαιο θα μιλήσουμε συγκεκριμένα για το ποιες γλώσσες προγραμματισμού χρειαστήκαμε για να δημιουργήσουμε το eclass,τι βάση δεδομένων χρησιμοποιήσαμε και άλλα σημαντικά εργαλεία που χρειάστηκαν για την υλοποίηση της πτυχιακής.

Κεφάλαιο 2

Πλατφόρμα E-class

\

Ένα site σαν το e-class αποτελείται από διάφορα στοιχεία. Πρώτο στοιχείο είναι η πρόσοψη του site αυτό δηλαδή που εμφανίζεται στο πρόγραμμα πλοήγησης(browser) όταν πληκτρολογήσουμε την διεύθυνση του ιστότοπου. Το e-class χρησιμοποιεί επίσης μια βάση δεδομένων, η οποία δεν είναι ορατή, ώστε να αντλεί πληροφορίες για τον χρήστη όταν αυτός συνδέεται στο site και να αποθηκεύει τυχόν τροποποιήσεις που κάνει στο προφίλ του. Για κάθε χρήστη, είτε αυτός είναι φοιτητής είτε είναι καθηγητής είτε είναι ο διαχειριστής του ιστότοπου, υπάρχει ένα ξεχωριστό προφίλ στην βάση δεδομένων. Επίσης υπάρχει ένας διακομιστής-εξυπηρετητής του οποίου η δουλειά είναι να παράγει τις ιστοσελίδες και να τις αποστέλλει στο πρόγραμμα πλοήγησης.

Για την δημιουργία site υπάρχουν διαθέσιμα στο ίντερνετ διάφορα εργαλεία που σου επιτρέπουν να δημιουργήσεις ιστότοπους για προγράμματα πλοήγησης υπολογιστών αλλά και για κινητά τηλέφωνα. Για την ολοκλήρωση της πτυχιακής εργασίας επέλεξα να χρησιμοποιήσω την εφαρμογή Visual Studio της Microsoft. Σαν διακομιστή-εξυπηρετητή χρησιμοποίησα τον Apache, η βάση δεδομένων είναι MySql και για την OnLine διαχείριση της βάσης δεδομένων χρησιμοποίησα το phpMyAdmin . Αυτά είναι τα εργαλεία που χρησιμοποίησαν και οι προγραμματιστές που δημιούργησαν το e-class.

Γλώσσες προγραμματισμού που μπορούν να χρησιμοποιηθούν για την δημιουργία του ιστότοπου e-class.

NET Framework και C#

Το .NET Framework είναι ένα πλαίσιο λογισμικού που προορίζεται για την πλατφόρμα των Windows. Αποτελείται από μια μεγάλη βιβλιοθήκη κλάσεων και υποστηρίζει μια πλειάδα γλωσσών προγραμματισμού με την δυνατότητα η μια να μπορεί να χρησιμοποιηθεί από την άλλη. Τα προγράμματα που γράφονται για το .NET Framework εκτελούνται σε ένα περιβάλλον εκτέλεσης γνωστό ως Common Language Runtime (CLR), ενός ειδικού λογισμικού, σχεδιασμένου να υποστηρίζει την εκτέλεση προγραμμάτων και την απρόσκοπτη συνεργασία με το λειτουργικό σύστημα. Το CLR περιέχει μια εικονική μηχανή (virtual machine) που διαχειρίζεται την εκτέλεση ενός προγράμματος και παρέχει μια σειρά σημαντικών υπηρεσιών όπως ασφάλεια, διαχείριση μνήμης και διαχείριση εξαιρέσεων. Τα προγράμματα που γράφονται για το .NET χρησιμοποιούν τη βιβλιοθήκη κλάσεων του .NET, η οποία δίνει πρόσβαση στο βάσεις δεδομένων, η κρυπτογραφία, η ανάπτυξη Web εφαρμογών και οι δικτυακές επικοινωνίες παρέχονται μέσω του Application Programming Interface (API) του .NET και μπορούν να συνδυαστούν με κώδικα από του προγραμματιστές για τη δημιουργία ολοκληρωμένων εφαρμογών. Όσο ένα πρόγραμμα ²¹περιορίζεται στη χρήση της βιβλιοθήκης κλάσεων του .NET, μπορεί να τρέχει οπουδήποτε υπάρχει εγκατεστημένο το περιβάλλον εκτέλεσης του .NET.περιβάλλον εκτέλεσης του.

Γλώσσες προγραμματισμού

Οι γλώσσες προγραμματισμού είναι θεωρητικά εκτός την έννοιας μια πλατφόρμας λογισμικού, ωστόσο η σημασία τους στο .NET Framework είναι αρκετά μεγάλη. Το .NET υποστηρίζει τη χρήση πολλαπλών γλωσσών για τη δημιουργία εφαρμογών και βιβλιοθηκών κλάσεων. Από την πρώτη έκδοση του .NET, τόσο η Microsoft όσο και τρίτοι πάροχοι προσέφεραν ένα πλούσιο σύνολο γλωσσών συμβατών με το περιβάλλον εκτέλεσης. Οι κύριες γλώσσες που παρέχονται από τη Microsoft σαν κομμάτι του .NET Framework SDK ωστόσο είναι τέσσερις :

- Η Visual Basic .NET : Μια γλώσσα με την οποία εξοικειώνονται αμέσως όσοι έχουν γράψει κλασικά προγράμματα Visual Basic στο παρελθόν και που τώρα στοχεύει στο .NET Framework με πλήρη υποστήριξη αντικειμενοστρέφιας.
- Η C# : Η γλώσσα που αναπτύχθηκε παράλληλα με τη δημιουργία του περιβάλλοντος εκτέλεσης του .NET και την πρώτη έκδοση των βιβλιοθηκών κλάσεων. Βασίζεται στις αρχές του αντικειμενοστραφούς προγραμματισμού και η φύση της είναι πολύ πρακτική.
- Η C++/CLI: αρχικά γνωστή ως Managed Extension for C++, επιτρέπει την εκτέλεση C++ κώδικα πάνω στο περιβάλλον εκτέλεσης του .NET παρέχοντας ένα πολύ καλό εργαλείο για την αντιμετώπιση δια λειτουργικών προβλημάτων.
- Η F# : η τελευταία προσθήκη στον πυρήνα γλωσσών του .NET που στοχεύει στην επιστημονική υπολογιστική και συνδυάζει τις αρχές του συναρτησιακού προγραμματισμού με τον πλούτο της βιβλιοθήκης κλάσεων του .NET Εκτός των επίσημων γλωσσών της Microsoft, διάφορες άλλες γλώσσες έχουν εισαχθεί στο .NET Framework. Παραδείγματα αποτελούν η COBOL, η Smalltalk, η Delphi, η APL, η Pascal και άλλες.

Το τοπίο των γλωσσών προγραμματισμού είναι πάντα ρευστό και η ενοποιημένη υποδομή του .NET παρέχει έναν ασφαλή τρόπο για την εξυπηρέτηση των αναγκών των διαφορετικών αυτών γλωσσών, δίνοντας τους τη δυνατότητα να χρησιμοποιήσουν βιβλιοθήκες μεταξύ τους και να χρησιμοποιήσουν τα έτοιμα εργαλεία. Όλα αυτά επιτρέπουν στους προγραμματιστές με διαφορετικές προτιμήσεις σε γλώσσες προγραμματισμού να δουλέψουν μαζί και να επιλέξουν τα σωστά εργαλεία για κάθε έργο. Για παράδειγμα, ενώ η C# είναι μια πολύ καλή γλώσσα γενικού σκοπού για τη δημιουργία εταιρικών εφαρμογών, η F# είναι πιο ελκυστική για την ανάπτυξη επιστημονικών εφαρμογών.

Η γλώσσα προγραμματισμού C#

Η C# είναι κατά κύριο λόγο μια αντικειμενοστραφής γλώσσα προγραμματισμού που ενσωματώνει ωστόσο μερικά χαρακτηριστικά διαφόρων προγραμματιστικών προτύπων. Αναπτύχθηκε στη Microsoft, από μια ομάδα κάτω από την ηγεσία του Anders Hejberg, σαν μέρος του .NET Framework. Η C# είναι μια συνεχώς εξελισσόμενη γλώσσα και με κάθε νέα της

έκδοση προστίθενται νέα χαρακτηριστικά και συντακτικό, με μόνιμο στόχο να κάνει τα απλά πράγματα εύκολα, τα δύσκολα πράγματα εφικτά και άσχημα πράγματα δύσκολα. Ακολουθεί μια σύντομη επεξήγηση των χαρακτηριστικών της C# .

Χαρακτηριστικά της C#

Απλότητα

Η C# είναι μια απλή γλώσσα που μπορεί να χρησιμοποιηθεί χωρίς εντατική εκμάθηση, ενώ ταυτόχρονα είναι εναρμονισμένη με σύγχρονες προγραμματιστικές πρακτικές. Οι θεμελιώδεις αρχές της γλώσσας μπορούν να κατανοηθούν γρήγορα κάτι που σημαίνει ότι οι προγραμματιστές θα είναι παραγωγικοί σε σύντομο χρονικό διάστημα. Η C# έχει σχεδιαστεί έτσι ώστε να μειώνεται η πιθανότητα πρόκλησης λαθών από την πολυπλοκότητα του κώδικα, αφού τη μειώνει σε μεγάλο βαθμό με το απλουστευμένο συντακτικό της και την οργάνωση κώδικά της.

Αντικειμενοστρέφεια

Η C# από τα θεμέλια της σχεδιάστηκε να είναι αντικειμενοστραφής. Ο αντικειμενοστραφής προγραμματισμός επικράτησε σαν προγραμματιστικό πρότυπο την προηγούμενη δεκαετία και παραμένει στις πρώτες προτιμήσεις των προγραμματιστών. Οι ανάγκες για κατανεμημένα συστήματα πελάτη-εξυπηρέτηση συμπίπτουν με την ενθουσία και την ανταλλαγή μηνυμάτων που είναι βασικά χαρακτηριστικά του αντικειμενοστρεφούς προγραμματισμού. Κατά πολλούς ειδικούς στις γλώσσες προγραμματισμού, η επιτυχής λειτουργία των προγραμματιστικών συστημάτων σε δικτυακά περιβάλλοντα αυξανόμενης πολυπλοκότητας βασίζεται στην αντικειμενοστρέφεια. Η C# παρέχει μια ξεκάθαρη και αποδοτική αντικειμενοστραφή πλατφόρμα παρέχοντας στους προγραμματιστές μια συλλογή βιβλιοθηκών δοκιμασμένων αντικειμένων που παρέχουν λειτουργικότητα που ποικίλει από απλούς τύπους δεδομένων, σε διεπαφές εισόδου/εξόδου ή δικτυακές και εργαλεία για τη δημιουργία παραθυρικών εφαρμογών. Αυτές οι βιβλιοθήκες μπορούν να προσαρμοστούν στις ανάγκες του προγραμματιστή. Επιπρόσθετα η C# υποστηρίζει και τον προγραμματισμό βασισμένο σε components ο οποίος επιτρέπει τον προσδιορισμό αυτόνομων μονάδων λειτουργικότητας που είναι απομονωμένα και τεκμηριωμένα, παρουσιάζοντας ένα μοντέλο με ιδιότητες, μεθόδους, events και μεταδεδομένα για το component. Η C# υποστηρίζει αυτά τα χαρακτηριστικά άμεσα κάνοντας έτσι τη διαδικασία δημιουργίας και χρήσης των components πολύ εύκολη.

HTML

Προέλευση

Το 1980, ο φυσικός Τιμ Μπέρνερς Λι, ο οποίος εργαζόταν στο Cern, επινόησε το ENQUIRE, ένα σύστημα χρήσης και διαμοιρασμού εγγράφων για τους ερευνητές του CERN και κατασκεύασε ένα πρωτότυπό του. Αργότερα, το 1989, πρότεινε ένα σύστημα βασισμένο στο διαδίκτυο, το οποίο θα χρησιμοποιούσε υπερκείμενο. Έτσι, έφτιαξε την προδιαγραφή της HTML και έγραψε το πρόγραμμα πλοήγησης και το λογισμικό του εξυπηρετητή.

Η HTML (ακρωνύμιο του αγγλικού Hyper Text Markup Language) είναι η βασική γλώσσα σήμανσης που χρησιμοποιούμε για την δημιουργία ιστοσελίδων, και τα στοιχεία της αποτελούν τα δομικά στοιχεία των ιστοσελίδων. Τα στοιχεία HTML αποτελούνται από ετικέτες που λειτουργούν ανά ζεύγη οι οποίες περικλείονται μέσα σε σύμβολα, «μεγαλύτερο από» και «μικρότερο από», για παράδειγμα

ΚΑΙ

. Ανάμεσα στις ετικέτες, υπάρχει η δυνατότητα να τοποθετήσουμε κείμενο, πίνακες, εικόνες κλπ. Ο σκοπός των προγραμμάτων πλοήγησης είναι να διαβάσουν τα έγγραφα HTML και να τα συνθέτουν σε σελίδες που μπορεί κανείς να τα διαβάσει ή να τα ακούσει. Το πρόγραμμα πλοήγησης δεν εμφανίζει τις ετικέτες HTML, αλλά τις χρησιμοποιεί για να ερμηνεύσει το περιεχόμενο της σελίδας. Τα στοιχεία της HTML χρησιμοποιούνται για να κτίσουν όλους τους ιστότοπους. Η HTML επιτρέπει την ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα στη σελίδα, και μπορεί να χρησιμοποιηθεί για να εμφανίσει δια δραστικές φόρμες. Παρέχει τις μεθόδους δημιουργίας δομημένων εγγράφων (εγγράφων που αποτελούνται από το περιεχόμενο που μεταφέρουν και από τον κώδικα μορφοποίησης του περιεχομένου) καθορίζοντας δομικά σημαντικά στοιχεία για το κείμενο, όπως κεφαλίδες, παραγράφους, λίστες, συνδέσμους, παραθέσεις και άλλα. Μπορούν επίσης να ενσωματώνονται με σενάρια εντολών σε γλώσσες όπως η JavaScript, τα οποία επηρεάζουν τη συμπεριφορά των ιστοσελίδων HTML.

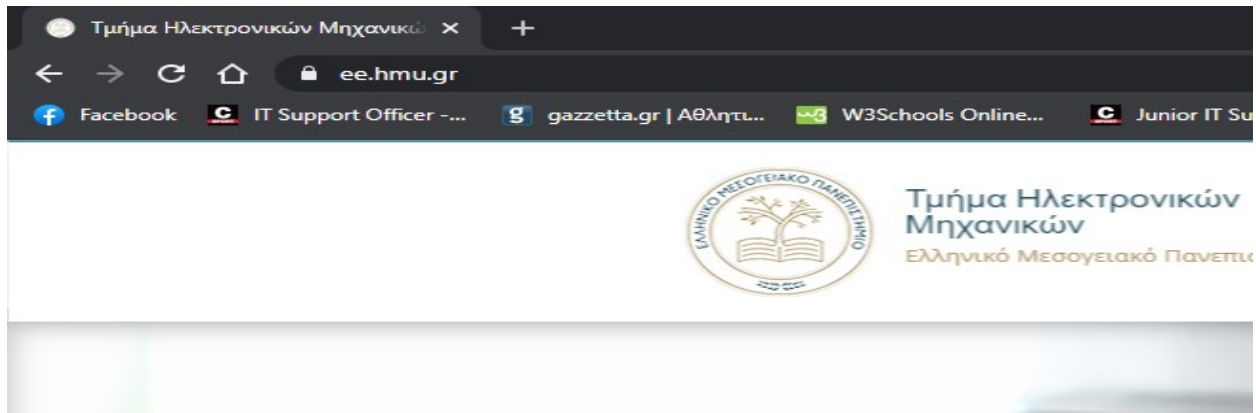
Βασικές γνώσεις της HTML που χρειάστηκαν για την δημιουργία του class

Όπως είδαμε στην εισαγωγή για την γλώσσα HTML είναι μια γλώσσα που χρησιμοποιείται για την δημιουργία ιστοσελίδων. Βασικά στοιχεία αυτής της γλώσσας είναι οι ετικέτες που είναι πάντα ζευγάρι. Η ετικέτα έναρξης και η ετικέτα λήξης, για παράδειγμα ετικέτα έναρξης και ετικέτα λήξης

24
.Τα HTML αρχεία αποτελούνται πάντα από δύο κομμάτια. Το κεφάλι () και το σώμα ().

Το κεφάλι του κώδικα HTML Το κεφάλι περιέχει πληροφορίες για το HTML αρχείο μας όπως τον τίτλο. Ο τίτλος είναι αυτό που βλέπουμε στην σελίδα του προγράμματος πλοήγησης όπως

φαίνεται στην παρακάτω φωτογραφία.



Εικ1(Ο τίτλος της ιστοσελίδας του ΤΕΙ που εμφανίζεται στο πρόγραμμα πλοήγησης).

Η παρακάτω εικόνα μας δείχνει τον κώδικα που γράφτηκε ώστε να εμφανίζεται αυτός ο τίτλος στο πρόγραμμα πλοήγησης.

```
<!DOCTYPE html>
<html lang="el" class="js">
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <link rel="pingback" href="https://ee.hmu.gr/xmlrpc.php">
    <script type="text/javascript" async src="https://www.google-analytics.com/analytics.js">
    </script>
    <script type="text/javascript">
      document.documentElement.className = 'js';
    </script>
    <script>...</script>
    <title>Τμήμα Ηλεκτρονικών Μηχανικών | Ελληνικό Μεσογειακό Πανεπιστήμιο</title>
```

Εικ2.(Ο κώδικας HTML της ιστοσελίδα του ΤΕΙ που μας δίνει αυτόν τον τίτλο).

Το σώμα του κώδικα HTML

Στο σώμα είναι το κομμάτι που θα βάλουμε το περιεχόμενο της ιστοσελίδας, όπως κείμενα, εικόνες και συνδέσμους. Το περιεχόμενο του σώματος είναι αυτό που βλέπουμε στην ιστοσελίδα μας.

Δημιουργία υπερσυνδέσεων

Πολλές φορές κομμάτια του κειμένου στα site που διαβάζουμε είναι συνδέσεις σε άλλες ιστοσελίδες. Το παρακάτω παράδειγμα μας δείχνει τον τρόπο για να κάνουμε μια τέτοια σύνδεση.

Για την δημιουργία υπερσυνδέσεων ο κώδικας είναι ο εξής . Αρχικά ανοίγουμε μια ετικέτα, μέσα στο σώμα του κώδικα html, [. Στην συνέχεια γράφουμε το κείμενο ή την λέξη με την οποία θέλουμε να εμφανίζεται αυτός ο υπερσύνδεσμος στην ιστοσελίδα μας και κλείνουμε την ετικέτα . Παράδειγμα](#)

Css

Με τον όρο CSS αναφερόμαστε σε μια γλώσσα μορφοποίησης σελίδων HTML. Για παράδειγμα, ένα φύλλο CSS μπορεί να ρυθμίσει την εμφάνιση των γραμματοσειρών, τα χρώματα, τα περιθώρια, τις γραμμές, το ύψος, το πλάτος, τις εικόνες φόντου μιας ιστοσελίδας και πολλά άλλα πράγματα. _

Διαφορά μεταξύ CSS και HTML Οι γλώσσες HTML και XHTML μπορεί να χρησιμοποιούνται για φτιάξουμε δομή στο περιεχόμενο και να ορίσουμε την διάταξη των αντικειμένων μιας ιστοσελίδας. Οι εντολές και τα φύλλα CSS προσφέρουν μορφοποίηση σε ήδη δομημένο περιεχόμενο, έχουν περισσότερες επιλογές εμφάνισης και καλύτερη ακρίβεια. Σήμερα τα CSS φύλλα υποστηρίζονται από όλα τα προγράμματα πλοήγησης. Αρχικά η γλώσσα HTML χρησιμοποιήθηκε μόνο για να προσδώσει δομή στο κείμενο. Ο προγραμματιστής θα μπορούσε να βάλει σήμανση σ' ένα κείμενο του με την χρήση των HTML tags, όριζε τον τίτλο με

και με

μια παράγραφο κλπ. Αυτό οδήγησε σε μια κατάσταση όπου οι αρχικές ετικέτες δομής, χρησιμοποιήθηκαν περισσότερο στις ιστοσελίδες προκειμένου να προσδιοριστεί η εμφάνιση του ιστότοπου παρά για να προσδεθεί δομή στο κείμενο. Επίσης υπήρχαν πολλές περιπτώσεις που ετικέτες διάταξης δεν υποστηριζόντουσαν από όλα τα προγράμματα πλοήγησης. Η γλώσσα CSS εφευρέθηκε για να διορθωθεί αυτή η κατάσταση, παρέχοντας στους σχεδιαστές ιστοσελίδων δυνατότητες οι οποίες να υποστηρίζονται από όλους τους browser. Ακόμη, ο διαχωρισμός της μορφής παρουσίασης των εγγράφων από το περιεχόμενο των εγγράφων, κάνει την συντήρηση του δικτυακού τύπου να είναι πολύ πιο εύκολη.

Δυνατότητες της CSS

Η CSS ήταν μια επανάσταση στον κόσμο του σχεδιασμού ιστοσελίδων. Παραθέτουμε κάποιες από τις δυνατότητες που μας δίνονται μέσω της CSS Ένα μόνο φύλλο CSS μπορεί να προσφέρει μορφοποίησης σε πολλά HTML και XHTML έγγραφα. Κάνει τον έλεγχο της διάταξης μιας ιστοσελίδας με μεγαλύτερη ακρίβεια. Παρουσιάζουν διαφορετική διάταξη σε διαφορετικούς τύπους μέσων πχ σε οθόνη, σε εκτύπωση, κλπ. Χρησιμοποιούν πολλές νεότερες, προηγμένες και εξελιγμένες τεχνικές. Στη συνέχεια θα δούμε πως μπορούμε να χρησιμοποιήσουμε την CSS ώστε να μορφοποιήσουμε την ιστοσελίδα μας, όπως για παράδειγμα να αλλάξουμε τα χρώματα το μέγεθος των γραμματοσειρών, το πλάτος και το χρώματα των γραμμών κ.α. Για παράδειγμα, σε μια σελίδα html οι απαιτήσεις μορφοποίησης μπορεί να είναι ως εξής : να καθορίσετε ότι το

φόντο της σελίδας είναι ένα ‘κρεμ’ χρώμα, το περιεχόμενο όλων των στοιχείων των διαφόρων παραγράφων () να εμφανίζεται με γκρι χρώμα και με γραμματοσειρά Arial, και όλοι οι τίτλοι πρώτου επιπέδου () θα πρέπει να είναι με κόκκινο χρώμα και γραμμένοι με την γραμματοσειρά Times New ROMAN. Για να πετύχουμε τέτοιες απαιτήσεις θα πρέπει να ορίσουμε ανάλογους κανόνες κάθε φορά για κάθε στοιχείο που θέλουμε να μορφοποιήσουμε.

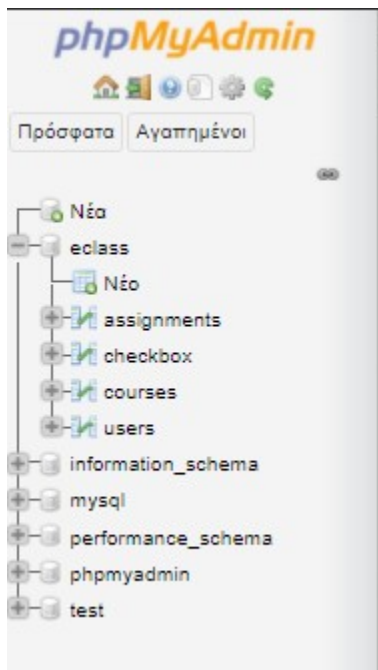
Στοιχεία CSS

Για αυτό είναι απαραίτητο να γνωρίζουμε τα παρακάτω :

- Τι κάνει ένας κανόνας CSS
- Πώς τοποθετούμε CSS κανόνες μέσα σε μια ιστοσελίδα ή πώς συνδέουμε την ιστοσελίδα με ένα εξωτερικό έγγραφο CSS
- Πώς ελέγχονται οι ιδιότητες παρουσίασης και οι αντίστοιχες τιμές τους των διαφόρων στοιχείων
- Πώς ελέγχεται η παρουσίαση του κειμένου με τη χρήση CSS
- Πώς τα φύλλα CSS βασίζονται σε ένα μοντέλο κουτί, και πώς ορίζονται διαφορετικές ιδιότητες για αυτά τα κουτιά. Στη συνέχεια θα δούμε κάποια βασικά παραδείγματα της CSS και των δυνατοτήτων που μας προσφέρει για την μορφοποίηση της ιστοσελίδα μας.

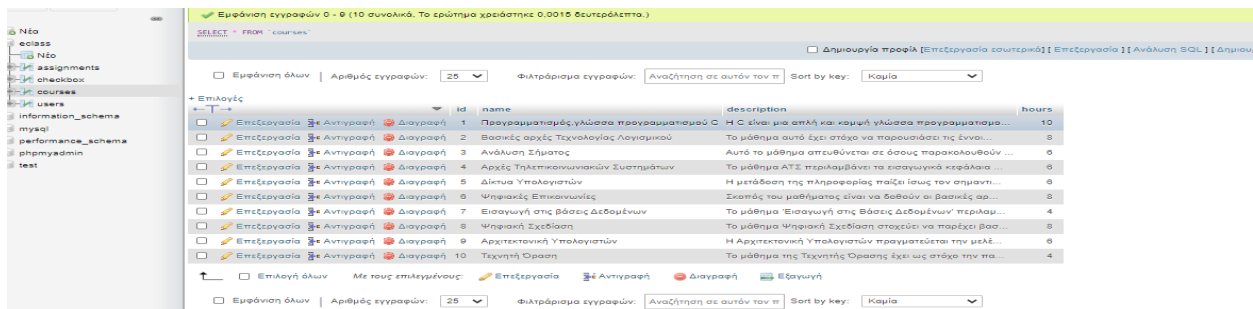
Βάση δεδομένων και Queries

Τα site που απαιτούν από τους χρήστες να δημιουργήσουν προσωπικό προφίλ χρησιμοποιούν, όπως αναφέραμε και στο προηγούμενο κεφάλαιο, βάσεις δεδομένων για την αποθήκευση όλων αυτών των πληροφοριών. Οι πληροφορίες στις βάσεις δεδομένων αποθηκεύονται σε πίνακες. Για την δημιουργία του eclass για κινητά χρησιμοποιήσαμε MySql βάση δεδομένων. Στην εικόνα παρακάτω βλέπουμε την βάση δεδομένων που τρέχει πίσω από το eclass.



Εικ3.(Το Eclass στην βάση δεδομένων μας).

Στην παραπάνω εικόνα βλέπουμε τους πίνακες μέσα στην βάση δεδομένων μας. Αυτοί οι πίνακες περιέχουν όλες τις πληροφορίες του site μας. Τα μαθήματα τις λεπτομέρειες των μαθημάτων, τους φοιτητές, πληροφορίες για τους φοιτητές κ.α.



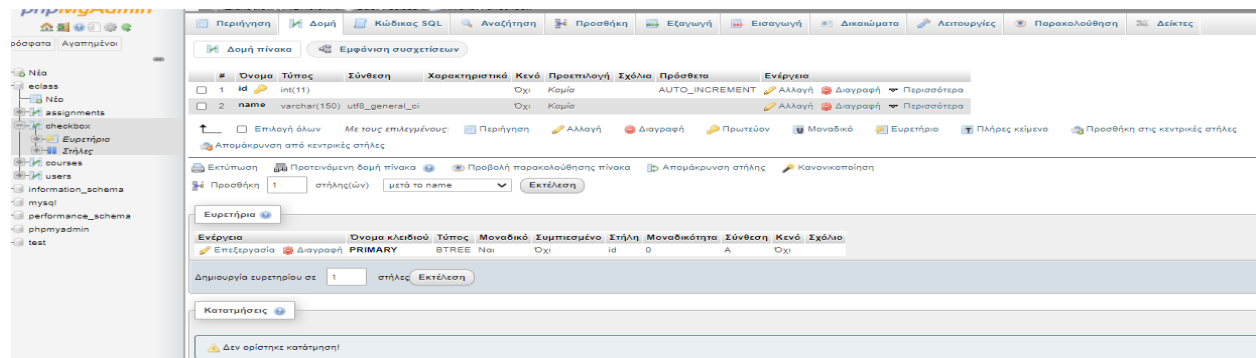
Εικ4.(Το Eclass ενότητα Courses, αποθηκευμένο).

Τα αιτήματα που γίνονται προς την βάση δεδομένων για συγκεκριμένες πληροφορίες ονομάζονται Queries. Πολλές φορές οι ερωτήσεις που κάνουμε προς τις βάσεις δεδομένων απαιτούν τον συνδυασμό πινάκων. Για παράδειγμα στο eclass οι φοιτητές μπορούν να δουν σε ποια μαθήματα είναι εγγεγραμμένοι. Αυτό για να το δούνε χρειάζεται ο συνδυασμός πινάκων. Την ίδια δυνατότητα θα έχουν και οι φοιτητές που επισκέπτονται το eclass από το κινητό τους. Για να γίνει εφικτό αυτό το αίτημα query που στείλαμε στην βάση δεδομένων είναι το εξής :

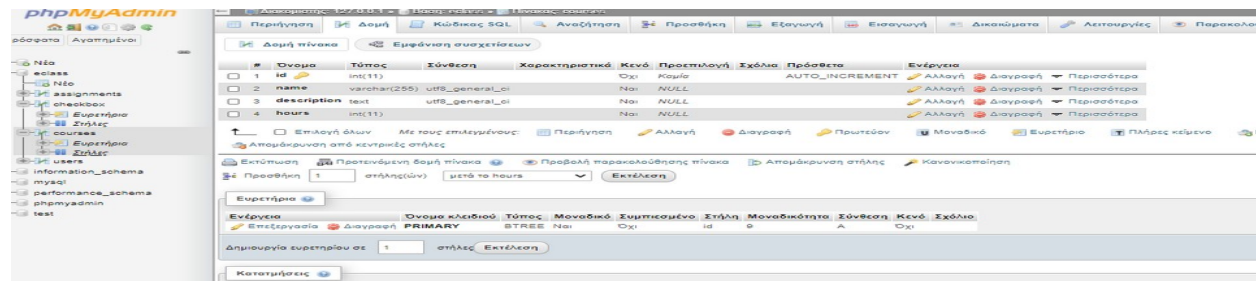
```
SELECT intitule, cours.cours_id FROM `course` join courses_user on cours.cours_id = courses_user.courses_id where user_id = 2
```

Μέσα στη βάση δεδομένων υπάρχει ένας πίνακας με την ονομασία cours(εικ 13) ο οποίος έχει όλα τα μαθήματα της σχολής μας. Μέσα σ αυτό το πίνακα υπάρχει μια στήλη στην οποία έχουμε αποθήκευση τα μαθήματα με την ονομασία τους την intitule. Έχουμε δώσει επίσης και μια ταυτότητα σε κάθε μάθημα και αυτή η ταυτότητα βρίσκεται στη στήλη cours_id. Αφού πάρουμε αυτές τις πληροφορίες από αυτόν το πίνακα του ζητάμε να τις συνδυάσει με τις πληροφορίες που θα πάρουμε από ένα άλλο πίνακα με την ονομασία courses_user(εικ 4) όπου εκεί έχουμε αποθήκευση τις ταυτότητες (id) των χρηστών αλλά και τις ταυτότητες των μαθημάτων (courses_id) και φαίνεται ποιοι χρήστες παρακολουθούν το κάθε μάθημα .

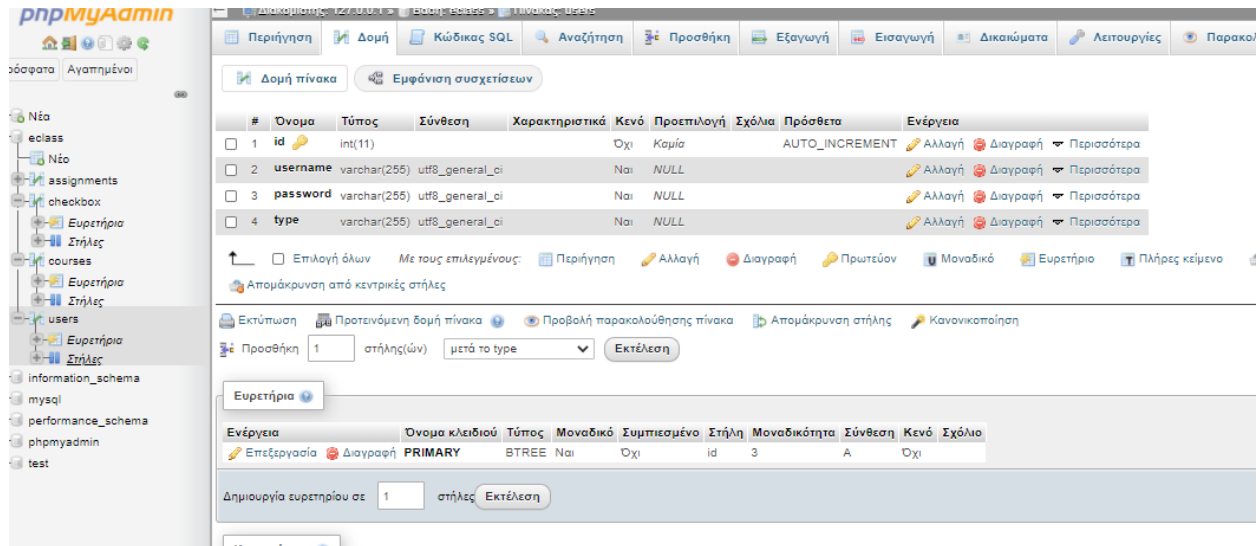
Στις παρακάτω φωτογραφίες (Εικόνες 5-8) δείχνω τα κλειδιά καθενός απο την βάση που χρησιμοποιήσα.



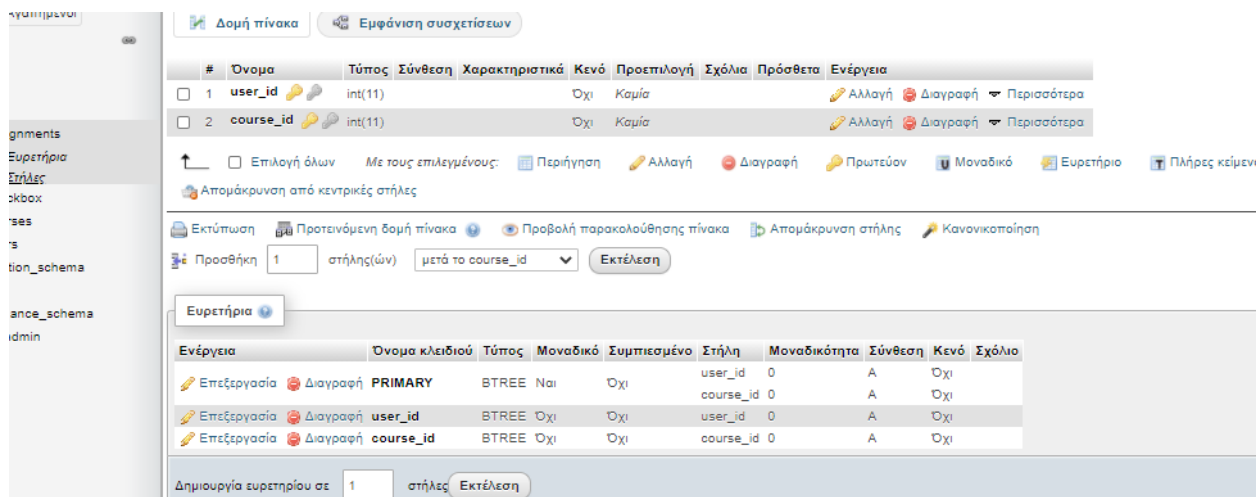
Εικ5.(Το κλειδί Checkbox).



Εικ6.(Το κλειδί courses).



Εικ7.(Το Eclass κλειδί users).



Εικ8.(Το κλειδί assignments).

Τα queries πολλές φορές μπορεί να είναι μια δύσκολη και σύνθετη διαδικασία. Ο μόνος τρόπος να κάνουμε πιο εύκολη αυτή τη διαδικασία είναι κατά την δημιουργία της βάσης δεδομένων. Ο προγραμματιστής καλείται να έχει στο νου του συνέχεια ότι μια βάση δεδομένων πρέπει να είναι καλά οργανωμένη και να είναι εύκολη η πρόσβαση σε βασικές πληροφορίες. Η html η CSS και η C# είναι οι τρεις γλώσσες που χρειαστήκαμε για την δημιουργία του eclass για κινητά. Το eclass το δημιουργήσαμε στο περιβάλλον του Visual Studio και χρησιμοποιήσαμε και κάποια άλλα βασικά εργαλεία.

Στο παρακάτω κεφάλαιο θα δούμε τα εργαλεία που χρησιμοποιήθηκαν και το περιβάλλον στο οποίο δημιουργήθηκε το eclass. 30

JavaScript

Μια άλλη γλώσσα που χρησιμοποιήσαμε για την δημιουργία του `eclass` είναι η JavaScript. Η JavaScript είναι μια δυναμική γλώσσα προγραμματισμού των ηλεκτρονικών υπολογιστών. Χρησιμοποιείται συχνά για την δημιουργία διαδραστικών ιστοσελίδων. Χρησιμοποιείται επίσης για την δημιουργία παιχνιδιών αλλά και για την δημιουργία εφαρμογών της επιφάνειας εργασίας και κινητών. Η JavaScript χαρακτηρίζεται ως μια πρωτότυπη γλώσσα σεναρίου (scripting) με δυναμικούς τύπους και πολλά χαρακτηριστικά από τις γλώσσες πρώτης κατηγορίας. Αυτά τα χαρακτηριστικά της δίνουν την δυνατότητα να υποστηρίζει διάφορα στυλ προγραμματισμού όπως αντικειμενοστραφή(object-oriented), επιτακτικό(imperative), και λειτουργικό(functional).

Εκτός από τις ομοιότητες στην ονομασία, στην σύνταξη και σε κάποιες βασικές βιβλιοθήκες, η JavaScript έχει πολύ διαφορετική σημασιολογία με την Java και είναι δύο εντελώς διαφορετικές γλώσσες. Στην πραγματικότητα ή σύνταξη της JavaScript προέρχεται από την C. Η JavaScript είναι μια ερμηνευόμενη γλώσσα (interpreted language), δηλαδή τα σενάρια (scripts) εκτελούνται χωρίς να προηγηθεί μεταγλώττιση του κώδικα. Η JavaScript υποστηρίζεται από όλα τα δημοφιλή προγράμματα πλοήγησης, όπως ο Internet Explorer, ο Google Chrome, ο Firefox κ.λπ. . Τρόποι εισαγωγής της JavaScript Υπάρχουν δύο τρόποι να εισάγουμε τον κώδικα JavaScript μέσα σε ένα Html αρχείο, εσωτερικά (Internal) και εξωτερικά (External).

1.Εσωτερικά Εσωτερικά του html αρχείου υπάρχουν δύο θέσεις που μπορούμε να γράψουμε τον JavaScript κώδικα. Η μια θέση είναι μέσα στο σώμα του html αρχείου δηλαδή στο

js Code.

Κεφάλαιο 3

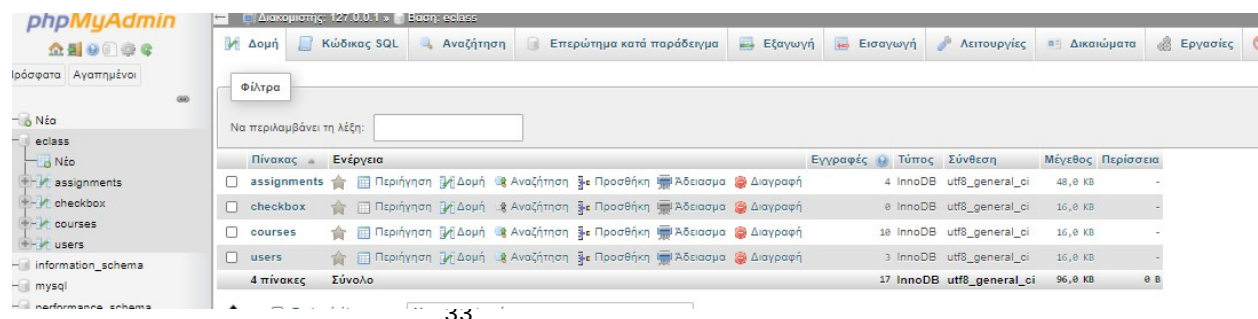
Προγραμματιστικά Εργαλεία

32

Εργαλεία που χρησιμοποιήθηκαν για την δημιουργία του Eclass

MySQL/phpMyAdmin

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων. Το πρόγραμμα τρέχει έναν εξυπηρετητή παρέχοντας πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων. Η MySQL είναι δημοφιλής βάση δεδομένων για διαδικτυακά προγράμματα και ιστοσελίδες. Χρησιμοποιείται από τις πιο διαδεδομένες διαδικτυακές υπηρεσίες, όπως το Flickr, το YouTube, η Wikipedia, το Google, το Facebook και το Twitter. Για την διαχείριση της MySQL μέσω ενός προγράμματος πλοήγησης χρησιμοποιήσαμε το phpMyAdmin ένα εργαλείο που διατίθεται δωρεάν στο ίντερνετ. Με το phpMyAdmin μπορούμε να κάνουμε διάφορες εργασίες όπως, δημιουργία τροποποίηση ή διαγραφή δεδομένων από τη βάση, δημιουργία πινάκων πεδίων ή σειρών. Διαχείριση των χρηστών και των δικαιωμάτων τους.



Εικ9.(Ολόκληρο το πρόγραμμα phpMyAdmin όπως το έχω αποθηκευμένο).

Apache HTTP

Μεσογειακό Πανεπιστήμιο Κρήτης
Κρανιωτακης Γιωργος

Ο apache είναι ένας εξυπηρετητής του διαδικτύου. Όταν ένας χρήστης επισκέπτεται μια ιστοσελίδα το πρόγραμμα πλοήγησης επικοινωνεί με ένα διακομιστή ο οποίος παράγει την επιθυμητή ιστοσελίδα και τις αποστέλλει πίσω στο πρόγραμμα πλοήγησης. Ένας από τους λόγους που οι προγραμματιστές χρησιμοποιούν συχνά τον apache είναι γιατί λειτουργεί σε διάφορες πλατφόρμες όπως τα Windows, Mac OS, Linux και Unix. Επίσης χρησιμοποιείται και σε τοπικά δίκτυα σαν διακομιστής συνεργαζόμενος με συστήματα διαχείρισης βάσης δεδομένων όπως MySql και Oracle .

PHP

Η **PHP** (PHP: Hypertext Preprocessor) είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. APACHE), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που είτε θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML ή θα επεξεργασθεί τις εισόδους δίχως να προβάλλει την έξοδο στο χρήστη, αλλά θα τις μεταβιβάσει σε κάποιο άλλο PHP script.

Ένα αρχείο με κώδικα PHP θα πρέπει να έχει την κατάλληλη επέκταση(π.χ. *.php, *.php4, *.phtml κ.ά.). Η ενσωμάτωση κώδικα σε ένα αρχείο επέκτασης .html δεν θα λειτουργήσει και θα εμφανίσει στον browser τον κώδικα χωρίς καμία επεξεργασία, εκτός αν έχει γίνει η κατάλληλη ρύθμιση στα MIME types του server. Επίσης ακόμη κι όταν ένα αρχείο έχει την επέκταση .php, θα πρέπει ο server να είναι ρυθμισμένος για να επεξεργάζεται και να μεταγλωττίζει τον κώδικα PHP σε HTML που καταλαβαίνει το πρόγραμμα πελάτη.

Ο συνδυασμός Linux/Apache/MySQL/PHP, που είναι η πιο δημοφιλής πλατφόρμα εκτέλεσης ιστοσελίδων είναι γνωστός και με το ακρωνύμιο LAMP. Παρόμοια, ο συνδυασμός */Apache/MySQL/PHP ονομάζεται *AMP, όπου το πρώτο αρχικό αντιστοιχεί στην πλατφόρμα, στην οποία εγκαθίστανται ο Apache, η MySQL και η PHP.

Visual Studio Code

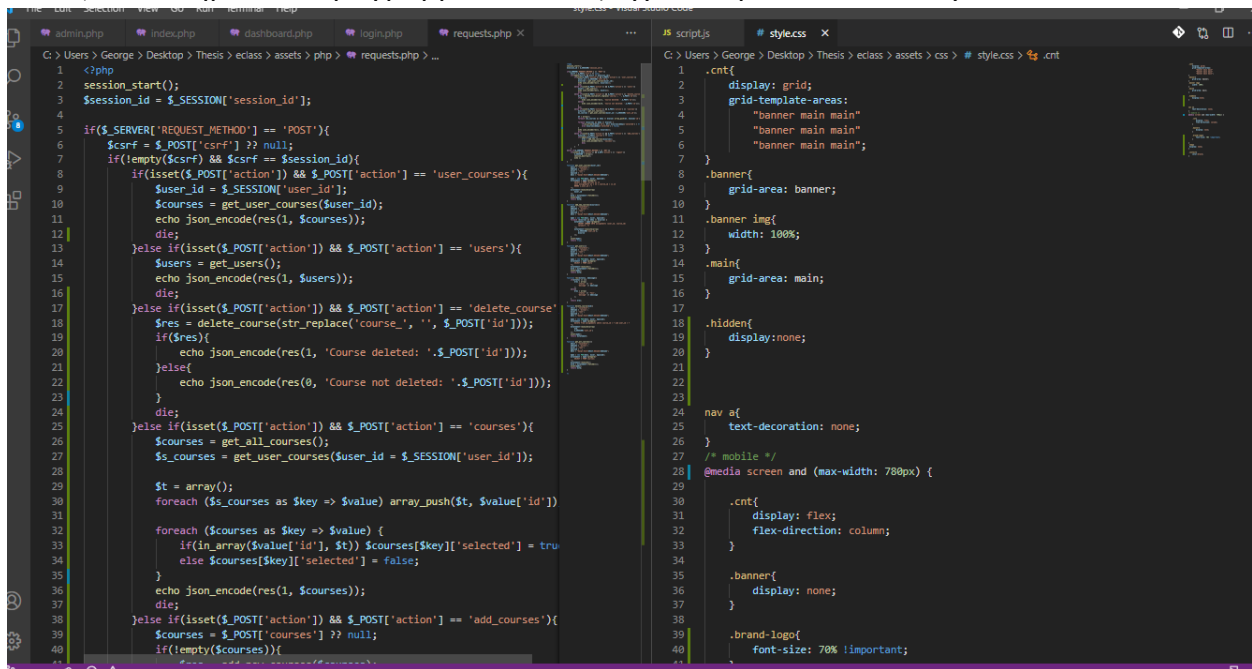
Το πρόγραμμα αυτό είναι ένα δωρεάν source-code editor, όπου δημιουργήθηκε από την Microsoft για χρήση μέσω Windows, Linux, MacOS. Υπάρχουν διαθέσιμες επιλογές στο συγκεκριμένο editor όπως η σύνταξη να φαίνεται με χρώμα, το debugging, snippet, code refactoring και συνεργασία με GIT. Οι χρήστες μπορούν να αλλάξουν θέματα, συντομεύσεις από το πληκτρολόγιο, προτιμήσεις και επέκτασεις όπου μπορούν να προσθέσουν οτι χρειάζεται ο κάθε χρήστης. Το Visual Studio Code, προέρχεται από το δωρεάν λογισμικό της Microsoft- VsCode-.

Η δυσκολία στην αρχική εγκατάσταση, φαίνεται ότι αποθαρρύνει πολλούς χρήστες στο να ξεκινήσουν να γράφουν Java στον VS Code. Ιδιαίτερα οι αρχάριοι χρήστες, αντιμετωπίζουν πολλά εμπόδια και συχνά τα παρατάνε για πιο εύχρηστες εναλλακτικές. Και δεν αποτελεί

έκπληξη, καθώς στο τεχνικό του κομμάτι, πρόκειται για μια πολύ περίπλοκη διαδικασία που συμπεριλαμβάνει την εγκατάσταση πολλών δυαδικών και επεκτάσεων του VS Code.

Η ομάδα του VS Code παρατήρησε το πρόβλημα και δεν άργησε να λάβει δράση. Πρόσφατα, η Microsoft κυκλοφόρησε έναν installer που αναλαμβάνει όλη αυτή τη διαδικασία για χάρη του χρήστη. Αρχικά βλέπει αν υπάρχει ήδη εγκατεστημένο ένα Java Development Kit (JDK). Αν όχι, εγκαθιστά έναν δυαδικό από το AdoptOpenJDK (που σπονσοράρει η Microsoft), ένα VS Code εφόσον χρειάζεται, και το Java Extension Pack. Το AdoptOpenJDK, το οποίο αποτελεί ουσιαστικά μια vendor-neutral εναλλακτική στο JDK της Oracle, είναι τώρα η προτεινόμενη Java διανομή της Microsoft για χρήστες που εγκαθιστούν την επέκταση Java του VS Code. Προς το παρόν, ο installer είναι διαθέσιμος μόνο για Windows, αλλά η ομάδα του VS Code σχεδιάζει να επεκτείνει την διαθεσιμότητά του μόλις φανεί το ενδιαφέρον από την κοινότητα των developers.

Εικ10.(Ολόκληρο το πρόγραμμα Eclass, σχεδιασμένο στο VSC).



The image shows a screenshot of the Visual Studio Code editor with two files open. The left pane shows a PHP file named 'requests.php' with the following code:

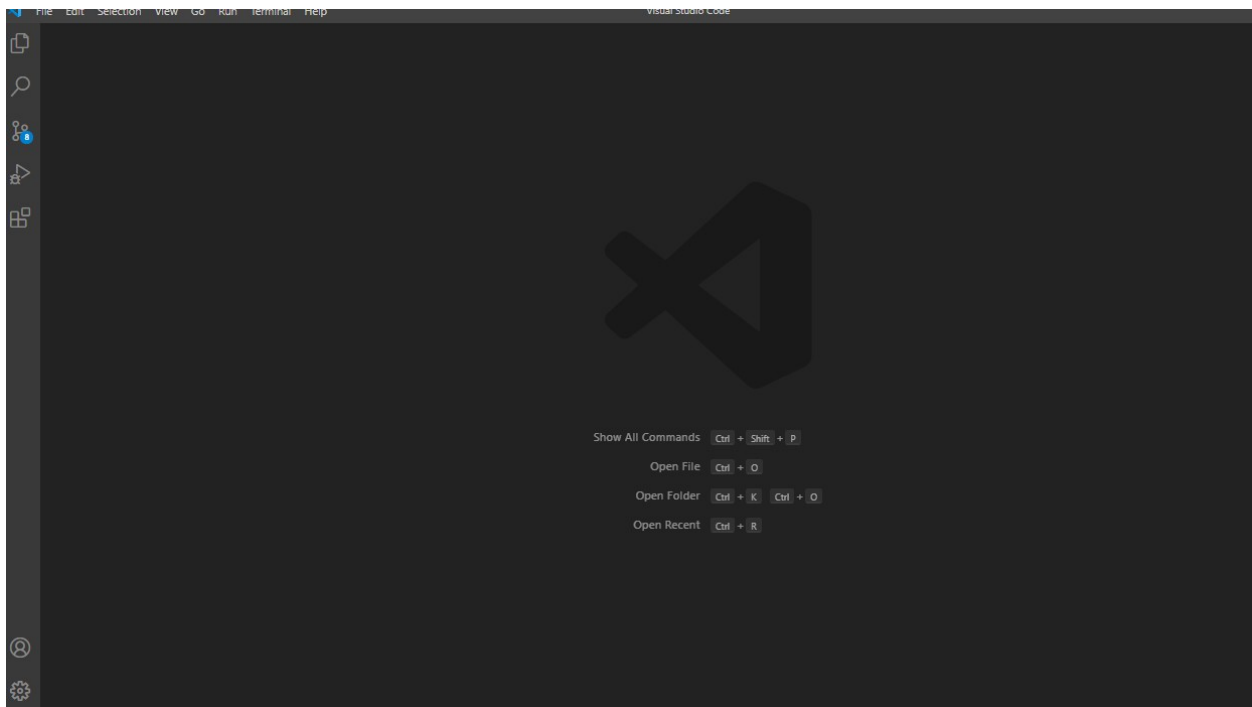
```
1 <?php
2 session_start();
3 $session_id = $_SESSION['session_id'];
4
5 if($_SERVER['REQUEST_METHOD'] == 'POST'){
6     $csrf = $_POST['csrf'] ?? null;
7     if(!empty($csrf) && $csrf == $session_id){
8         if(isset($_POST['action']) && $_POST['action'] == 'user_courses'){
9             $user_id = $_SESSION['user_id'];
10            $courses = get_user_courses($user_id);
11            echo json_encode(res(1, $courses));
12        }
13        }else if(isset($_POST['action']) && $_POST['action'] == 'users'){
14            $users = get_users();
15            echo json_encode(res(1, $users));
16        }
17        }else if(isset($_POST['action']) && $_POST['action'] == 'delete_course'){
18            $res = delete_course(str_replace('course_', '', $_POST['id']));
19            if($res){
20                echo json_encode(res(1, 'Course deleted: '.$_POST['id']));
21            }else{
22                echo json_encode(res(0, 'Course not deleted: '.$_POST['id']));
23            }
24        }
25        }else if(isset($_POST['action']) && $_POST['action'] == 'courses'){
26            $courses = get_all_courses();
27            $s_courses = get_user_courses($user_id = $_SESSION['user_id']);
28
29            $t = array();
30            foreach ($s_courses as $key => $value) array_push($t, $value['id']);
31
32            foreach ($courses as $key => $value) {
33                if(in_array($value['id'], $t)) $courses[$key]['selected'] = true;
34                else $courses[$key]['selected'] = false;
35            }
36            echo json_encode(res(1, $courses));
37        }
38        }else if(isset($_POST['action']) && $_POST['action'] == 'add_courses'){
39            $courses = $_POST['courses'] ?? null;
40            if(!empty($courses)){
```

The right pane shows a CSS file named 'style.css' with the following code:

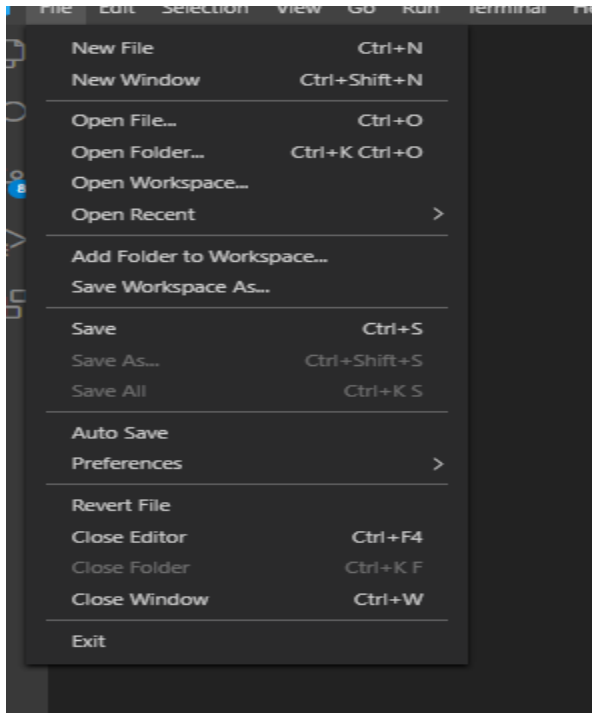
```
1 .cnt{
2     display: grid;
3     grid-template-areas:
4         "banner main main"
5         "banner main main";
6 }
7
8 .banner{
9     grid-area: banner;
10 }
11
12 .banner_img{
13     width: 100%;
14 }
15
16 .main{
17     grid-area: main;
18 }
19
20 .hidden{
21     display:none;
22 }
23
24 nav a{
25     text-decoration: none;
26 }
27 /* mobile */
28 @media screen and (max-width: 780px) {
29
30     .cnt{
31         display: flex;
32         flex-direction: column;
33     }
34
35     .banner{
36         display: none;
37     }
38
39     .brand-logo{
40         font-size: 70% !important;
41     }
```

Παρατηρείτε τα 5 ανοιχτά παραθύρα στην αριστερη πλευρα και τα 2 στην δεξια.Είναι διαμορφωμένα έτσι ώστε να με βοηθήσουν στην γλώσσα προγραμματισμου όπου έγγραφα.Αριστερά είναι η PHP και δεξιά η Javascript/Css.Γενικά το πρόγραμμα έχει πολλές

επιλογές για ευχρηστία του χρήστη και εμφάνιση για αυτόν που θα ασχοληθεί.Επιπλέον,παρακάτω θα παραθέσω πως είναι το VSC στην αρχική του μορφή.



Εικ11.(Αρχική μορφή VSC).



Εικ12.(Οι επιλογές μας στο κουμπί File στο VSC).

Παρακάτω θα εξηγησώ πως μπορούμε να προγραμματίσουμε στο πρόγραμμα καθώς θα επεξηγώ αυτό που έχω δημιουργήσει.Εργάστηκα εξ ολοκλήρου μέσω VSC και παρόλο που το επεξεργάστηκα πρώτη φορά,το βρήκα άνετο και ενδιαφέρον.Λιγη εξάσκηση χρειάζεται και είναι παρόμοιο με το Dreamweaver,για εμάς τους παλιούς που ξέρουμε.

Στην συνέχεια θα συνδέσουμε την ιστοσελίδα μας με την βάση δεδομένων που έχουμε δημιουργήσει μέσω PHPMyAdmin.Για να συνδέσουμε την σελίδα μας με την βάση δεδομένων MySQL που χρησιμοποιούμε θα χρειαστούμε να κάνουμε μια προσθήκη αναφοράς στο Project μας.Εφόσον προσθέσαμε την αναφορά μας στο Project είμαστε έτοιμοι να γράψουμε των κώδικα που θα μας χρειαστεί. Όπως έχουμε αναφέρει στα παραπάνω κεφάλαια η Html γλώσσα είναι η γλώσσα που χρησιμοποιούμε για να δημιουργήσουμε το γραφικό περιβάλλον του site μας. Η σύνδεση του site με την βάση δεδομένων θα γίνει γράφοντας τον κατάλληλο κώδικα.

Στην παρακάτω εικόνα βλέπουμε τον κώδικα που έκανε εφικτή την σύνδεση του site μας με την βάση δεδομένων.Λόγω των πολλών requests,είναι αρκετα μεγαλος και θα τον χωρίσω ανάλογα.

```
ex.php dashboard.php admin.php login.php X requests.php
Users > George > Desktop > Thesis > eclass > assets > php > login.php > ...
login.php

if($_SERVER['REQUEST_METHOD'] == 'POST'){
    $username = $_POST['username'] ?? null;
    $password = $_POST['password'] ?? null;

    $user = find_user($username);

    if(!empty($user)){
        if(!empty($user['password'])){
            $hash = $user['password'] ?? null;

            if(password_verify($password, $hash)){
                session_start();
                $csrf = rand(0,100000);
                $_SESSION['user_id'] = $user['id'];
                $_SESSION['session_id'] = $csrf;
                $_SESSION['type'] = $user['type'];
                $_SESSION['logged_in'] = true;
                header("Location: ../../dashboard.php");
            }else{
                session_start();
                $_SESSION = array();
                session_destroy();
                header("Location: ../../index.php?error=inc_pass");
            }
        }
    }else{
        header("Location: ../../index.php?error=no_user");
    }
}
}
```

Εικ13.(Ο κώδικας που

χρησιμοποιήσα για την βάση για το Login του προγράμματος στο login.php).

```
> Users > George > Desktop > Thesis > eclass > admin.php > ...
1 login.php
2 die; // REMOVE TO WORK
3 // filter_var('bob@example.com', FILTER_VALIDATE_EMAIL)
4
5 $username = '';
6 $password = '';
7
8 $password = password_hash($password, PASSWORD_DEFAULT);
9
10
11 $host = 'localhost';
12 $dbname = 'eclass';
13 $user = 'root';
14 $passwd = '';
15 $dsn = "mysql:host=$host;dbname=$dbname";
16
17 $pdo = new PDO($dsn, $user, $passwd);
18 $statement = $pdo->prepare("INSERT INTO users (username, password) values (?,
19 $statement->execute(array(
20     $username,
21     $password
22 ));
23 unset($pdo);
24
25 >
```

Εικ14.(Ο

κώδικας που χρησιμοποίησα ώστε να συνδεθώ στην βάση για το Login του προγράμματος στο admin.php όπου διαλέγω και γράφω εγώ ποιός χρήστης μπορεί να log(edit))

Μεσογετακο Πανεπιστημιο Κρητης
Κρανιωτακης Γιωργος

Αν παρατηρήσετε υπάρχουν αρκετές γραμμές κώδικα σχετικά με την βάση και την ένωση με την Sql. Αυτό συμβαίνει λόγω του ότι έχω δημιουργήσει για κάθε χρήστη, τα συγκεκριμένα μαθήματα που έχει διαλέξει και να αποθηκεύονται και μετά το logout. Ακόμα, έχω δημιουργήσει το checkbox, όπου χρειάζεται η “υπενθύμιση” στον κάθε χρήστη για τα μαθήματα του, καθώς και η επιλογή της διαγραφής για κάθε συγκεκριμένο μάθημα, για κάθε συγκεκριμένου χρήστη. Τέλος να σημειωθεί ότι και μετά το logout του χρήστη, παραμένουν τα μαθήματα όπως και στο παράθυρο με το checkbox, τα ήδη δηλωμένα μαθήματα δεν μπορείς να τα ξαναδηλώσεις.

Παρακάτω υπάρχει μια εικόνα με τον αρχικό κώδικα που χρησιμοποίησα για την ένωση. Θα προσπαθήσω να τον σπάσω σε εικόνες για καλύτερη επεξήγηση και αναφορά, παρόλο που φαίνεται εύκολος, δεν ήταν. Προσπάθησα να μειώσω τις γραμμές που θα χρειαζόταν κανονικά, αφού υπήρχαν πολλοί τρόποι για να δημιουργηθεί και νομίζω κατάφερα το βέλτιστο αποτέλεσμα σε αρκετά μειωμένες γραμμές κώδικα. Επισυνάπτω τις φωτογραφίες παρακάτω.

```
#!/php
session_start();
$_SESSION['session_id'];

if($_SERVER['REQUEST_METHOD'] == 'POST'){
    $csrf = $_POST['csrf'] ?? null;
    if(!empty($csrf) && $csrf == $_SESSION['session_id']){
        if(isset($_POST['action']) && $_POST['action'] == 'user_courses'){
            $user_id = $_SESSION['user_id'];
            $courses = get_user_courses($user_id);
            echo json_encode(res(1, $courses));
            die;
        }else if(isset($_POST['action']) && $_POST['action'] == 'users'){
            $users = get_users();
            echo json_encode(res(1, $users));
            die;
        }else if(isset($_POST['action']) && $_POST['action'] == 'delete_course'){
            $res = delete_course(str_replace('course_', '', $_POST['id']));
            if($res){
                echo json_encode(res(1, 'Course deleted: '.$_POST['id']));
            }else{
                echo json_encode(res(0, 'Course not deleted: '.$_POST['id']));
            }
            die;
        }else if(isset($_POST['action']) && $_POST['action'] == 'courses'){
            $courses = get_all_courses();
            $s_courses = get_user_courses($user_id = $_SESSION['user_id']);
```

Εικ15.(Ο

κώδικας που χρησιμοποίησα για τον σερβερ και την βάση για χρήστη,μαθήματα χρήστη,διαγραφή μαθημάτων χρήστη,μαθήματα).


```

function get_user_courses($user_id){
    $host = 'localhost';
    $dbname = 'eclass';
    $user = 'root';
    $passwd = '';
    $dsn = "mysql:host=$host;dbname=$dbname";

    $pdo = new PDO($dsn, $user, $passwd);
    $statement = $pdo->prepare("
        SELECT * FROM courses as cs
        join assignments as s on s.course_id = cs.id
        WHERE s.user_id = ?
    ");
    $statement->execute(array(
        $user_id
    ));
    $stm = $statement->fetchAll();
    unset($pdo);
    return $stm;
}

function add_new_courses($courses){
    $host = 'localhost';
    $dbname = 'eclass';
    $user = 'root';
    $passwd = '';
    $dsn = "mysql:host=$host;dbname=$dbname";

    $pdo = new PDO($dsn, $user, $passwd);
    foreach ($courses as $key => $course) {
        $statement = $pdo->prepare("
            INSERT IGNORE INTO assignments (user_id, course_id)
            values(?, ?)
        ");
        $statement->execute(array(
            $_SESSION['user_id'],
            $course
        ));
    }
    unset($pdo);
}

```

Εικ16.(Ο κώδικας που

χρηάζεται(function),ώστε να τραβήξω το κάθε μάθημα χρήστη και να προσθέσω κάθε καινούργιο μάθημα στην βάση δεδομένων μας).

Λόγω του ότι είναι αρκετά μεγάλος ο κώδικας στο σημείο αυτό(request.php),παρόμοιο κώδικα χρησιμοποίησα για να συντάξω,τον κάθε χρήστη ξεχωριστά,την διαγραφή μαθημάτων του κάθε χρήστη και φυσικά για την διαλογή όλων των διαθέσιμων μαθημάτων.Κοινός παρανομαστής όλων είναι η εμφάνιση του pdo,γνώστο και σαν PHP Data Objects,όπου χρειάζεται για την σύνδεση της PHP μας,με την βάση δεδομένων μας.Υπάρχει μια σύγκριση μεταξύ PDO και

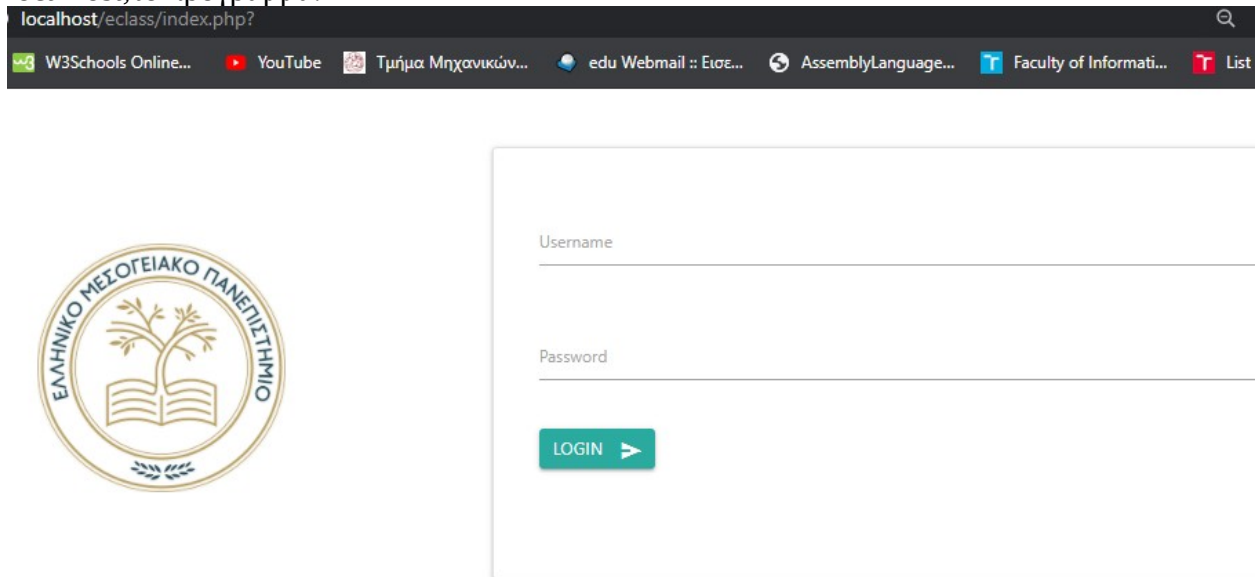
Mysqli, που είναι επίσης object oriented. Η βιβλιοθήκη PDO είναι αμιγώς αντικειμενοστραφής και επειδή το μέλλον της PHP είναι κοντά στις object oriented προσεγγίσεις, σίγουρα η χρήση της PDO είναι μια εγγυημένη λύση μακροπρόθεσμα. Επίσης διαθέτει δυνατότητα σύνδεσης με 12 διαφορετικά συστήματα διαχείρισης βάσεων δεδομένων εν αντίθεση με την Mysqli προσέγγιση όπου συνεργάζεται μόνο με την SQL. Η βιβλιοθήκη Mysqli όπου το I σημαίνει improved, διαθέτει και μια διαδικασιακή διεπαφή, οπότε κάνει αφενός πιο εύκολη την μετάβαση από παλαιού τύπου απλές Mysql συνδέσεις σε συνδέσεις Mysqli, αφετέρου δε η procedural διεπαφή απέχει από την φιλοσοφία του μέλλοντος, οπότε αργά η γρήγορα ενδεχομένως να χρειαστεί να αλλάξουμε τις όποιες προσεγγίσεις υλοποιήσαμε. Τόσο η PDO όσο και η Mysqli υποστηρίζουν prepared statements οι οποίες βοηθούν στην προστασία της εφαρμογής μας από sql injections και είναι ιδιαίτερα σημαντικές για την ασφάλεια της διαδικτυακής μας εφαρμογής.

```
1  $?php
2  die; // REMOVE TO WORK
3  // filter_var('bob@example.com', FILTER_VALIDATE_EMAIL)
4
5  $username = '';
6  $password = '';
7
8  $password = password_hash($password, PASSWORD_DEFAULT);
9
10
11 $host = 'localhost';
12 $dbname = 'eclass';
13 $user = 'root';
14 $passwd = '';
15 $dsn = "mysql:host=$host;dbname=$dbname";
16
17 $pdo = new PDO($dsn, $user, $passwd);
18 $statement = $pdo->prepare("INSERT INTO users (username, password) values (?,
19 $statement->execute(array(
20     $username,
21     $password
22 ));
23 unset($pdo);
24
25 ?>
```

Εικ17.(Ο

κώδικας που δημιούργησα για να μπορώ να κάνω edit τους χρήστες, ώστε να μπορούν να μπουν στο site μου).

Στην παρακάτω φωτογραφία φαίνεται το αποτέλεσμα που παίρνουμε όταν τρέξω στο localhost, το πρόγραμμα.



Εικ17(Το αποτέλεσμα του localhost).

Σαν επόμενο βήμα θα προσπαθήσουμε να εμφανίσουμε στο site μας ένα πίνακα με πληροφορίες από την βάση δεδομένων μας. Στις παρακάτω εικόνες ακολουθεί ο απαραίτητος κώδικας για την σύζευξη του ιστότοπου που δημιουργήσαμε με τις επιθυμητές πληροφορίες από την βάση δεδομένων που χρησιμοποιούμε.

```

❏ php

✓ if($_SERVER['REQUEST_METHOD'] == 'POST'){
  $username = $_POST['username'] ?? null;
  $password = $_POST['password'] ?? null;

  $user = find_user($username);

  if(!empty($user)){
    if(!empty($user['password'])){
      $phash = $user['password'] ?? null;

      if(password_verify($password, $phash)){
        session_start();
        $csrf = rand(0,100000);
        $_SESSION['user_id'] = $user['id'];
        $_SESSION['session_id'] = $csrf;
        $_SESSION['type'] = $user['type'];
        $_SESSION['logged_in'] = true;
        header("Location: ../../dashboard.php");
      }else{
        session_start();
        $_SESSION = array();
        session_destroy();
        header("Location: ../../index.php?error=inc_pass");
      }
    }
  }else{
    header("Location: ../../index.php?error=no_user");
  }
}

✓ function find_user($username){
  $host = 'localhost';
  $dbname = 'eclass';
  $user = 'root';
  $passwd = '';
  $dsn = "mysql:host=$host;dbname=$dbname";

```

Εικ18(Το αποτέλεσμα

του κώδικα για το Login του προτζεκτ μας,συνδεση βασης με πρόγραμμα επίσης).

Όπως και στο προηγούμενό Project που κάναμε πρέπει να δώσουμε τις απαραίτητες πληροφορίες στο site μας για να συνδεθεί με τη βάση δεδομένων. Το που βρίσκεται και πώς ονομάζεται η βάση δεδομένων, το όνομα και τον κωδικό του χρήστη με τα οποία θα γίνει εφικτή η σύνδεση με την βάση δεδομένων. Σε αυτό το Project στόχος μας είναι να συνδεθούμε στην βάση να πάρουμε κάποιες πληροφορίες από ένα πίνακα της επιλογής μας που βρίσκεται στη βάση δεδομένων και να τα εμφανίσουμε στον ιστότοπο που δημιουργούμε. Εφόσον συνδέσαμε τον ιστότοπο μας με την βάση δεδομένων περνάμε στο δεύτερο βήμα στην ανάκτηση δηλαδή των επιθυμητών πληροφοριών .

Αριθμός εγγράφων: 25 Φιλτράρισμα εγγράφων: Άνω

πιλοτές

	user_id	course_id
Επεξεργασία Αντιγραφή Διαγραφή	1	1
Επεξεργασία Αντιγραφή Διαγραφή	1	3
Επεξεργασία Αντιγραφή Διαγραφή	1	4
Επεξεργασία Αντιγραφή Διαγραφή	1	5
Επεξεργασία Αντιγραφή Διαγραφή	2	1
Επεξεργασία Αντιγραφή Διαγραφή	2	2

Επιλογή όλων Με τους επιλεγμένους: Επεξεργασία Αντιγραφή

Εμφάνιση όλων Αριθμός εγγράφων: 25 Φιλτράρισμα εγγράφων: Άνω

Εικ19(Το αποτέλεσμα στην βάση

δεδομένων μας σχετικά με τι μαθήματα έχει πάρει κάθε ξεχωριστός χρήστης).

	id	username	password	type
Επεξεργασία Αντιγραφή Διαγραφή	1	test	\$2y\$10\$Jl.JtVRBBdDGYIZnUwθgj.IT7hIlg7D3p1N1l/bRZvf...	student
Επεξεργασία Αντιγραφή Διαγραφή	2	test1	\$2y\$10\$M0tMox.phQ5DunBB.VShseojOUWjE/7YqHrvjpUj0SP...	NULL
Επεξεργασία Αντιγραφή Διαγραφή	3	test@test.com	\$2y\$10\$SId8QYnWcH/y1sB2Gwθ6/υθBZNRXZ9n8pyθ22L6Cm1B...	NULL

Εικ20(Οι χρήστες που έχω δημιουργήσει για να κάνουν Login στο σύστημα).

name	description	hours
Προγραμματισμός, γλώσσα προγραμματισμού C	Η C είναι μια απλή και κομψή γλώσσα προγραμματισμο...	10
Βασικές αρχές Τεχνολογίας Λογισμικού	Το μάθημα αυτό έχει στόχο να παρουσιάσει τις έννοι...	8
Ανάλυση Σήματος	Αυτό το μάθημα απευθύνεται σε όσους παρακολουθούν ...	6
Αρχές Τηλεπικοινωνιακών Συστημάτων	Το μάθημα ΑΤΣ περιλαμβάνει τα εισαγωγικά κεφάλαια ...	6
Δίκτυα Υπολογιστών	Η μετάδοση της πληροφορίας παίζει ίσως τον σημαντι...	6
Ψηφιακές Επικοινωνίες	Σκοπός του μαθήματος είναι να δοθούν οι βασικές αρ...	8
Εισαγωγή στις βάσεις Δεδομένων	Το μάθημα 'Εισαγωγή στις Βάσεις Δεδομένων' περιλαμ...	4
Ψηφιακή Σχεδίαση	Το μάθημα Ψηφιακή Σχεδίαση στοχεύει να παρέχει βασ...	8
Αρχιτεκτονική Υπολογιστών	Η Αρχιτεκτονική Υπολογιστών πραγματεύεται την μελέ...	6
Τεχνητή Όραση	Το μάθημα της Τεχνητής Όρασης έχει ως στόχο την πα...	4

Εικ21(Τα

Courses που μπορεί ο κάθε ξεχωριστός χρήστης να διαλέξει).

Τελειώνοντας με την θεωρία και τα παραδείγματα για τις γλώσσες και τα εργαλεία που χρησιμοποιήσαμε για την τροποποίηση του eclass στο επόμενο κεφάλαιο θα δούμε τον κώδικα που κρύβεται πίσω από το eclass και την εικόνα που θα δούμε αν το ανοίξουμε από την οθόνη του Desktop/Laptop/Tablet/Κινητό μας.

Κεφάλαιο 4

Περιγραφή κώδικα του E-class

Τροποποίηση του Eclass για Desktops/Laptops/Tablets/Mobiles

Η δημιουργία του eclass αποτελείται από δύο κομμάτια. Το πρώτο κομμάτι είναι η πρόσοψη του site, που για να την δημιουργήσουμε χρησιμοποιήσαμε την γλώσσα HTML. Για το καλλωπισμό του site (χρώματα επιλογή γραμματοσειράς κλπ.) χρησιμοποιούμε την CSS.

```
scripts # style.css x
: > xampp > htdocs > eclass > assets > css > # style.css > .cnt
1  |.cnt{
2     display: grid;
3     grid-template-areas:
4         "banner main main"
5         "banner main main"
6         "banner main main";
7  }
8  .banner{
9     grid-area: banner;
10 }
11 .banner img{
12     width: 100%;
13 }
14 .main{
15     grid-area: main;
16 }
17
18 .hidden{
19     display:none;
20 }
21
22
23
24 nav a{
25     text-decoration: none;
26 }
27 /* mobile */
28 @media screen and (max-width: 780px) {
29
30     .cnt{
31         display: flex;
32         flex-direction: column;
33     }
34
35     .banner{
36         display: none;
37     }
38
39     .brand-logo{
40         font-size: 70% !important;
```

Εικ22(Ο κώδικας του CSS σαν παράδειγμα, λόγω

του ότι είναι αρκετά μεγάλος.Απλός και κατανοητός).

Το δεύτερο κομμάτι αφορά την σύνδεση του site με την βάση δεδομένων, που χρησιμοποιούμε για να αποθηκεύουμε τους χρήστες και περεταίρω στοιχεία για αυτούς. Σε αυτό το σημείο έχω δημιουργήσει ορισμένες Functions τις οποίες καλώ μέσω του Script.Js, για κάθε ξεχωριστό πράγμα που τις χρειαζόμαστε. Για παράδειγμα, τα μαθήματα των χρηστών, τους χρήστες, τα μαθήματα που είναι προς επιλογή και ακόμα την δυνατότητα διαγραφής μαθήματος αν το επιθυμεί ο χρήστης φυσικά. Τώρα για την σύνδεση που χρειαζόμαστε για όλα αυτά παραθέτω ορισμένο κώδικα, διότι δεν γίνεται να ανεβάσω ολόκληρο το request που έχω χρησιμοποιήσει.

```
    }else if(isset($_POST['action']) && $_POST['action'] == 'courses'){
        $courses = get_all_courses();
        $s_courses = get_user_courses($user_id = $_SESSION['user_id']);

        $t = array();
        foreach ($s_courses as $key => $value) array_push($t, $value['id']);

        foreach ($courses as $key => $value) {
            if(in_array($value['id'], $t)) $courses[$key]['selected'] = true;
            else $courses[$key]['selected'] = false;
        }
        echo json_encode(res(1, $courses));
        die;
    }else if(isset($_POST['action']) && $_POST['action'] == 'add_courses'){
        $courses = $_POST['courses'] ?? null;
        if(!empty($courses)){
            $res = add_new_courses($courses);
            echo json_encode(res(1, 'success'));
            die;
        }
    }
}

}else if($_SERVER['REQUEST_METHOD'] == 'GET'){
    if(isset($_GET['action']) && $_GET['action'] == 'logout'){
        $_SESSION = array();
        session_destroy();
        echo 1;
    }
}

function get_user_courses($user_id){
    $host = 'localhost';
    $dbname = 'eclass';
    $user = 'root';
    $passwd = '';
    $dsn = "mysql:host=$host;dbname=$dbname";

    $pdo = new PDO($dsn, $user, $passwd);
    $statement = $pdo->prepare("
        SELECT * FROM courses as cs
```

Ετικ

23(Κώδικας όπου είναι απαραίτητος για συνδέσεις μεταξύ βάσης και προγράμματος).

Παρουσίαση και επεξήγηση του κώδικα του Eclass

Σε αυτό το κεφάλαιο θα δούμε μια τις σελίδες του eclass και τον κώδικα που τρέχει πίσω από τις σελίδες.

1)Index page

Για την διαμόρφωση του Eclass,χρειάστηκα ορισμένες βιβλιοθήκες για κάθε γλώσσα που χρησιμοποιήσα(PHP,JS).Σε αυτό το παράθυρο αποθήκευσα τις βιβλιοθήκες καθώς και ορισμένα πράγματα για την ομαλή λειτουργία του προγράμματος,όπως τις εικόνες,την διαμορφωση των εικόνων,Ακόμα,υπάρχει το κουμπί για την σύνδεση το πρόγραμμα,το μήνυμα error αν δεν υπάρχει ο χρήστης που έχω δηλώσει.Παραθέτω την φωτογραφία παρακάτω.

```
index.php X admin.php login.php dashboard.php requests.php
: > xampp > htdocs > eclass > index.php > html > head
1 <html>
2 <head>
3 <link rel="stylesheet" href="assets/libs/materialize/css/materialize.min.css">
4 <link rel="stylesheet" href="assets/css/style.css">
5 <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
6 <link rel="stylesheet" href="assets/libs/bootstrap/css/bootstrap.min.css">
7
8 <script defer src="assets/js/script.js"></script>
9 <script src="assets/libs/jquery/jquery-3.5.1.min.js"></script>
10 <script src="assets/libs/bootstrap/js/bootstrap.min.js"></script>
11 <script src="assets/libs/materialize/js/materialize.min.js"></script>
12 </head>
13 <body>
14 <div class="cnt my-5 mx-5">
15 <div class="banner m-auto">
16 
17
18
19
20
21 </div>
22 <div class="main card p-5">
23 <div class="row">
24 <form class="col s12" action="assets/php/login.php" method="post">
25 <div class="row">
26 <div class="input-field col s12">
27 <input id="username" type="text" name="username" class="validate">
28 <label for="username">Username</label>
29 </div>
30 </div>
31 <div class="row">
32 <div class="input-field col s12">
33 <input id="password" type="password" name="password" class="validate">
34 <label for="password">Password</label>
35 </div>
36 </div>
37 <div class="row">
```

Εικ24(Index.Php).

Παραπάνω βλέπουμε ακριβώς οτι είχα γράψει πριν την εικόνα.Ο απλός κώδικας του Intex βοηθάει να δηλώσω τις βιβλιοθήκες και τα απλά πράγματα της αρχικής σελίδας.Έχω τα unique id του προγράμματος,καθώς και οι κλάσεις των ξεχωριστών δειγμάτων που χρειάζομαι για να τρέξει η αρχική σελίδα και έπειτα.Υπαρχει η εικόνα-img-όπου την βλέπετε μετα το Login και το Button όπου το βρήκα έτοιμο στο Ίντερνετ και απλα το έκανα copy-paste(btn waves-effect).

2) Admin.PHP

```
amp > htdocs > eclass > admin.php > ...
<?php
die; // REMOVE TO WORK
// filter_var('bob@example.com', FILTER_VALIDATE_EMAIL)

$username = '';
$password = '';

$password = password_hash($password, PASSWORD_DEFAULT);

$host = 'localhost';
$dbname = 'eclass';
$user = 'root';
$passwd = '';
$dsn = "mysql:host=$host;dbname=$dbname";

$pdo = new PDO($dsn, $user, $passwd);
$stmt = $pdo->prepare("INSERT INTO users (username, password) values (?, ?)");
$stmt->execute(array(
    $username,
    $password
));
unset($pdo);
</pre>
```

Εικ24(Admin.Php).

Στην παραπάνω εικόνα βλέπουμε το αρχείο Admin,όπου δηλώνω τους χρήστες που θα μπορούν να κάνουν login και τους τοποθετώ στην βάση δεδομένων που έχω προσθέσει για αυτό τον σκοπό.


```

php_session_start();
if(!isset($_SESSION['logged_in']) || $_SESSION['logged_in'] != true){
    header("Location: index.php");
}
}
}
<html>
<head>
<link rel="stylesheet" href="assets/css/style.css">
<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
<!-- <link rel="stylesheet" href="assets/libs/bootstrap/css/bootstrap.min.css" -->
<link rel="stylesheet" href="assets/libs/materialize/css/materialize.min.css">
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/sweetalert2@10.3.5/dist/sweetalert2.min.css">

<script defer src="assets/js/script.js"></script>
<script src="assets/libs/jquery/jquery-3.5.1.min.js"></script>
<!-- <script src="assets/libs/bootstrap/js/bootstrap.min.js"></script> -->
<script src="assets/libs/materialize/js/materialize.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/sweetalert2@10.3.5/dist/sweetalert2.all.min.js"></script>
</head>
<body>
<div class="csrf" style="display:none;"><?php echo $_SESSION['session_id'];</div>
<nav>
<div class="nav-wrapper grey darken-4">
<a href="dashboard.php" class="brand-logo">Ελληνικό Μεσογειακό Πανεπιστήμιο </a>
<a href="#" data-target="mobile-demo" class="sidenav-trigger"><i class="material-icons">menu</i></a>
<ul class="right hide-on-med-and-down">
<li><a href="#" class="btn1">Τα Μαθήματα Μου</a></li>
<li><a href="#" class="btn2">Δήλωση Μαθημάτων</a></li>
<li><a href="#" class="btn3">Επικοινωνία</a></li>
<?php if(isset($_SESSION['logged_in']) && $_SESSION['logged_in'] == true){?>
<li><a href="#" class="log">Αποσύνδεση</a></li>
<?php } ?>
</ul>
</div>
</nav>

```

Εικ25(Dashboard.Php).

```

<ul class="sidenav" id="mobile-demo">
<li><a href="#" class="btn1"> Τα Μαθήματα Μου</a></li>
<li><a href="#" class="btn2">Δήλωση Μαθημάτων</a></li>
<li><a href="#" class="btn3">Επικοινωνία</a></li>
<li><a href="#" class="log">Αποσύνδεση</a></li>
</ul>
<div class="banner">

</div>
<div class="main">
<div class="row">
<div class="col s12">
<div class="card">
<div class="card-content">
</div>
</div>
</div>
</div>
<div class="temp">
<div class="contact">
<p>
Το Ελληνικό Μεσογειακό Πανεπιστήμιο είναι ένα
σημείο ακαδημαϊκό ίδρυμα της Ελλάδας, με έδρα το
Ηράκλειο Κρήτης και παραρτήματα στα Χανιά, το Ρέθυμνο,
τον Άγιο Νικόλαο, την Ιεράπετρα και τη Σητεία.<br>
Διεύθυνση: Gianni Kornarou, Estavromenos 1, Ηράκλειο 714 10<br><br>
Όρες:
Κλειστά - Ανοίγει στις 9:00 π.μ. Δευτέρα <br>
Τηλέφωνο: 281 037 9200<br>
Ίδρυση: 25 Απριλίου 2019<br>
Πρύτανης: Νικόλαος Καταράκης
</p>
</div>

```

Εικ26(Dashboard2.Php).

Μεσογειακο Πανεπιστημιο Κρητης
Κρανιωτακης Γιωργος

Οι 2 εικόνες παραπάνω μας δείχνουν τον κώδικα που χρησιμοποίησα εφόσον έκανε Login ο χρήστης μέσα στο σύστημα. Παρατηρούμε τις ονομασίες των κουμπιών καθώς και την μαύρη λωρίδα δίπλα τους. Ακόμα, όλα τα Links, Scripts που χρειάζεται το πρόγραμμα όπως materialize και εννοείται JQuery, όπου είναι βασισμένο όλο το πρόγραμμά μου. Στα btn1, btn2, btn3 είναι τα κουμπια του προγράμματος (Μαθήματα, Επιλογή μαθημάτων, Επικοινωνία) και στο επόμενο είναι το button για το logout απο το σύστημα μου. Στην εικόνα 26, δείχνω το πρόγραμμα μου σε στυλ κινητού και γενικά σε μικρότερη οθόνη απο Η/Υ. Το γνωστό navbar το οποίο δούλεψα και είναι ίδιο με τον Η/Υ το πρόγραμμα. Υπάρχουν οι ίδιες επιλογές ακριβώς και τα ίδια στοιχεία. Τέλος, φαίνεται το edit του κουμπιού επικοινωνίας όπου έγραψα ένα απλό text για όταν θέλει να επικοινωνήσει ο χρήστης με την γραμματεία του τμήματος.

3) Script.JS

Στο τελευταίο μας Script, το οποίο είναι και το πιο σημαντικό μιας και εκεί λειτουργούν τα πάντα ως javascript script, υπάρχει ο κώδικας απο το navbar μέχρι και τις function που καλούνται εδώ. Παρακάτω θα ανεβάσω εικόνα σαν παράδειγμα, απλα μικρότερη γιατί δεν χωράνε σε ένα printscreen όλα μαζί. Περαιτέρω, λειτουργώ το btn1, το οποίο είναι τα μαθήματα του χρήστη, μέσω της βάσης μου και μέσω ενός ajax, μου εμφανίζει τα μαθήματα που έχει διαλέξει ο κάθε χρήστης ξεχωριστά με την επιλογή της διαγραφής. Πριν γίνει αυτό καλεί τους user που έχω κάνει edit από το script admin.php και ελέγχει αν όντως υπάρχουν. Στο btn2 το οποίο είναι για την λίστα μαθημάτων που μπορεί να διαλέξει ο χρήστης, έχω βάλει ένα table ένα δηλαδή εμφανισιακό παραλληλόγραμμο για να μπορώ να ταξινομήσω όμορφα τα μαθήματα σε στήλες και να βάλω τις επιλογές για να μπορώ να διαλέξω. Επίσης κάθε μάθημα έχει δικιά του περίληψη που είναι ένα spot up παράθυρο κάθε φορά που κλικάρεις το κάθε μάθημα. Στο btn3 είναι η απλή περίληψη της επικοινωνίας και τέλος στο logout btn κάνεις όπως λέει, logout και παραμένουν τα μαθήματα αποθηκεύμενα σε κάθε χρήστη. Επίσης στο btn2, όπου είναι το checkbox, μετά την διαλογή των μαθημάτων, παραμένει γκρι το μάθημα και δεν μπορεί ο χρήστης να το ξαναδιαλέξει. Μπορεί όμως τα υπόλοιπα μαθήματα. Εφόσον πατήσει διαγραφή στα μαθήματα μου, τότε και μόνο ξανα ανοίγει η επιλογή στην επόμενη καρτέλα για τα μαθήματα.

```

$('.btn1').on('click', function (e) {
    ...
}

$('.btn2').on('click', function (e) {
    $('.contact').appendTo('.temp');
    $(".contact").hide();
    $.ajax({
        type: 'post',
        url: 'assets/php/requests.php',
        data: {
            csrf: $('.csrf').html(),
            action: "courses"
        },
        success: function (res) {
            try {
                res = JSON.parse(res);
                if (res.status == 'ok') {
                    $('.card-content').html('');
                    let table = `
                    <table>
                    <thead>
                    <tr>
                    <td>Όνομα Μαθηματος</td>
                    <td>Ώρες Μαθηματος</td>
                    <td>Επιλογη</td>
                    </tr>
                    </thead>
                    <tbody>
                    `;
                    res.message.forEach(function (course, key) {
                        let cb = '';
                        if(course.selected == true){cb = '<p><label><input type="checkbox" checked="checked"
                        else {cb = '<p><label><input type="checkbox" /><span></span></label></p>'}
                        table += `
                        <tr id="${course.id}">
                        <td><a href="#" class="show_info">${course.name}</a><span class="hidden" id=
                        <td>${course.hours}</td>
                        <td>${cb}</td>
                        </tr>
                        `;
                    });
                }
            }
        }
    });
}

```

Εικ27(Script.Js,Btn2 λειτουργία).

```

$('.btn1').on('click', function (e) {...
$('.btn2').on('click', function (e) {...
$('.btn3').on('click', function (e) {...

$('.log').on('click', function (e) {
  $.ajax({
    type: 'get',
    url: 'assets/php/requests.php',
    data: {
      csrf: $('#.csrf').html(),
      action: "logout"
    },
    success: function (res) {
      if (res == true) {
        window.location.replace("index.php");
      }
    }
  });
});

function deleteCoursesEventListener(button) {
  $.ajax({
    type: 'post',
    url: 'assets/php/requests.php',
    data: {
      csrf: $('#.csrf').html(),
      action: 'delete_course',
      id: button.attr('id')
    },
    success: function (res) {
      try {
        res = JSON.parse(res);
        if (res.status == 'ok') {
          button.closest('tr').remove();
        } else if (res.status == 'error') {
          alert('Failed to delete');
        }
      }
    }
  });
}

```

Εικ28(Script.Js λειτουργία για κουμπί διαγραφή).

Παρακάτω θα ανεβάσω εικόνες απο το πρόγραμμα ακριβώς όπως πρέπει να είναι.Θα συνδεθώ και θα διαλέξω ορισμένα μαθήματα καθώς και θα δείξω την επιλογή διαγραφής,logout και επικοινωνίας.Τελος θα δείξω και τις δυνατότητες της Navbar.

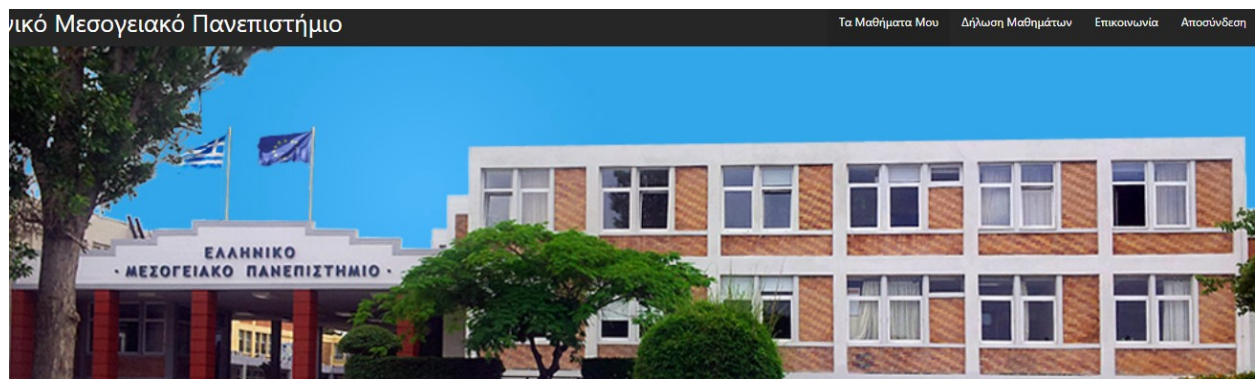


Username
test

Password
....|

LOGIN

Εικ29(Η εικόνα login όπου κάνω 'συνδεση με τον χρήστη test/test).



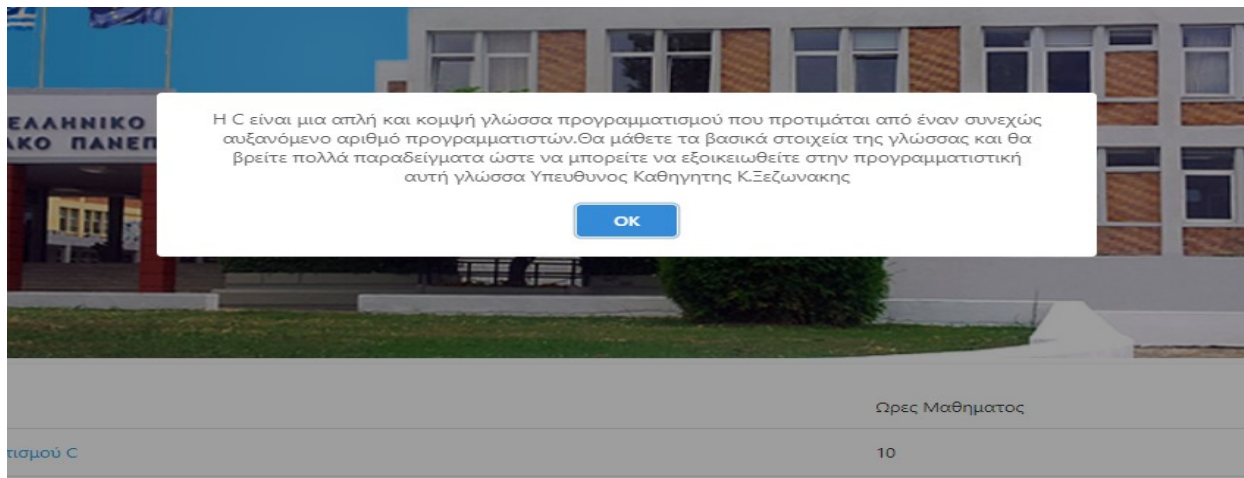
Εικ30(Η εικόνα στο dashboard,μετα το login μου).

Όνομα Μαθήματος	Περιγραφή Μαθήματος	Ωρες	
Προγραμματισμός γλώσσα προγραμματισμού C	Η C είναι μια απλή και κομψή γλώσσα προγραμματισμού που προτιμάται από έναν συνεχώς αυξανόμενο αριθμό προγραμματιστών.Θα μάθετε τα βασικά στοιχεία της γλώσσας και θα βρείτε πολλά παραδείγματα ώστε να μπορείτε να εξοικειωθείτε στην προγραμματιστική αυτή γλώσσα Υπευθυνος Καθηγητής Κ.Ξεζωνάκης	10	DELETE
Ανάλυση Σήματος	Αυτό το μάθημα απευθύνεται σε όσους παρακολουθούν μαθήματα σε επίπεδο τριτοβάθμιας εκπαίδευσης όπου χρειάζονται οι γνώσεις του μαθήματος Ψηφιακή Επεξεργασία Σήματος.Εξ ολοκλήρου χρειάζεται και καλό επίπεδο σε μαθήματα όπως Απειροστικός Λογισμός και Γραμμικής Αλγεβρας. Υπευθυνος Καθηγητής Κ.Παλλής	6	DELETE
Αρχές Τηλεπικοινωνιακών Συστημάτων	Το μάθημα ΑΤΣ περιλαμβάνει τα εισαγωγικά κεφάλαια όπου καλύπτουν τον μαθητή στο θέμα αναφέρεται η φασματική ανάλυση,σειρές Fourier και μετασχηματισμοί Fourier.Επίται στοχεύει στα συστήματα διαμόρφωσης θέρωρμα δειγματοληψίας,πολύπλεξη,ανάλυση των AM και FM και τέλος στην ανάλυση της απόδοσης των ψηφιακών συστημάτων επικοινωνίας.Μετά την ολοκλήρωση του μαθήματος ο φοιτητής θα έχει αποκτήσει τις απαραίτητες γνώσεις και δεξιότητες ώστε: • Να μπορεί να σχεδιάζει και να αναλύει απλά τηλεπικοινωνιακά συστήματα. • Να προσομοιώνει σε Η/Υ τηλεπικοινωνιακά κυκλώματα με την χρήση του Python και να ελέγχει την συμπεριφορά τους μεταβάλλοντας διάφορες παραμέτρους, ώστε να επιτύχει την καλύτερη δυνατή απόδοση. Υπευθυνος Καθηγητής Κ.Στρατακας	6	DELETE

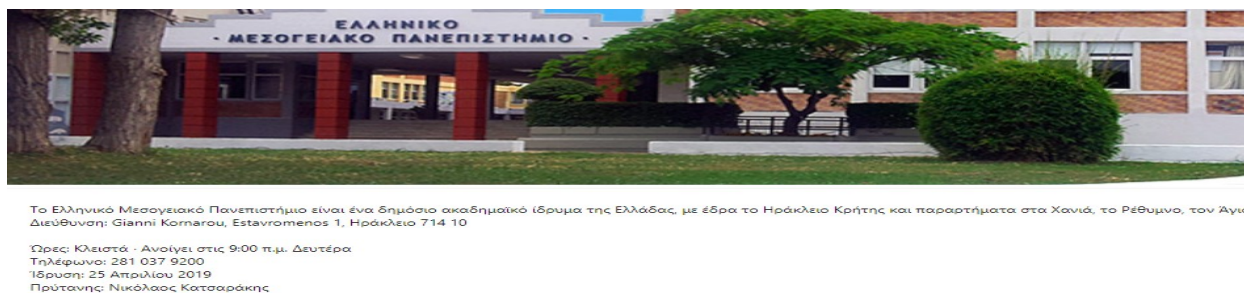
Εικ31(Τα μαθήματα που έχω διαλέξει σαν χρήστης test και η επιλογή διαγραφής τους).

νομα Μαθήματος	Ωρες Μαθηματος	Επιλογή
ρογραμματισμός γλώσσα προγραμματισμού C	10	✓
ασικές αρχές Τεχνολογίας Λογισμικού	8	<input type="checkbox"/>
νάλυση Σήματος	6	✓
ρχές Τηλεπικοινωνιακών Συστημάτων	6	✓
ικτνα Υπολογιστών	6	✓

Εικ32(Τα μαθήματα που έχω επιλέξει και αυτά που δεν έχω ανοιχτά ακόμα σαν επιλογή).



Εικ33(Το pop up window που πετάγεται σε κάθε μάθημα όταν το click(C σε αυτή την φωτο)



Εικ34(Το κουμπί επικοινωνία και τι μας δείχνει).

Μεσογειακο Πανεπιστημιο Κρητης
Κρανιωτακης Γιωργος

Αναφορές

- <https://el.wikipedia.org/wiki/PHP>
- <https://jquery.com/>
- <https://www.mysql.com/>
- <https://code.visualstudio.com/>
- <https://stackoverflow.com/questions/28417781/jquery-add-csrf-token-to-all-post-requests-data>
- <https://www.w3schools.com/>
- https://www.w3schools.com/jquery/jquery_ref_ajax.asp
- <https://api.jquery.com/jquery.ajax/>
- Basic Knowledge of course and my friend Vaggelis Kalosinakis helped a lot

