

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ**

**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**ΘΕΜΑ:**

**ΦΟΙΤΗΤΙΚΟ FORUM ΣΧΟΛΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**ΦΟΙΤΗΤΗΣ: ΔΗΜΗΤΡΙΟΣ ΚΑΡΑΝΤΟΥΣΗΣ**

**ΕΙΣΗΓΗΤΗΣ: ΑΝΤΩΝΙΟΣ ΔΑΔΑΛΙΑΡΗΣ**

**ΛΑΜΙΑ 2019**

## ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ

ΟΝΟΜΑ: ΔΗΜΗΤΡΙΟΣ ΚΑΡΑΝΤΟΥΣΗΣ

Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις (1), που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί χωρίς να τα περικλείω σε εισαγωγικά και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάστηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.

2. Δέχομαι ότι η αυτολεξεί παράθεση χωρίς εισαγωγικά, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.

3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια.

4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

ΥΠΟΓΡΑΦΗ: ΔΚ

## ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΕΧΟΜΕΝΑ.....	3
ΠΕΡΙΛΗΨΗ.....	4
1.ΕΙΣΑΓΩΓΗ.....	5
1.1 Αντικείμενο της εργασίας.....	5
1.2 Σκοπός της εργασίας.....	5
1.3 Διάρθρωση της εργασίας.....	6
1.4 Λέξεις - κλειδιά.....	7
2.ΛΕΙΤΟΥΡΓΙΕΣ.....	9
2.1 Αρχική σελίδα.....	9
2.2 Σύστημα σύνδεσης - εγγραφής.....	12
2.3 Ανακοινώσεις.....	16
2.4 Ομαδική συζήτηση.....	18
2.5 Συμπληρωματικές λειτουργίες.....	20
2.6 Προβολή αρχείων.....	21
2.7 Κατάθεση αποριών.....	23
2.8 Δυναμικό ημερολόγιο.....	29
2.9 Κώδικας επεξεργασίας εμφάνισης.....	32
3. ΠΑΡΑΡΤΗΜΑ.....	34
3.1 Εικόνες.....	34
4. ΣΥΜΠΕΡΑΣΜΑΤΑ.....	41
5. ΒΙΒΛΙΟΓΡΑΦΙΑ.....	42

## ΠΕΡΙΛΗΨΗ

Στην σύγχρονη εποχή η πλήρης ενημέρωση ενός σπουδαστή σχετικά με τα δρώμενα στην σχολή του ξεφεύγει από τα όρια της απλής παρουσίας στο χώρο σπουδών του, ενώ δυσχεραίνεται από τα μεταβαλλόμενα προγράμματα και τους υψηλούς ρυθμούς της καθημερινότητας. Η συγκεκριμένη εργασία επιχειρεί να δώσει λύση σε αυτό το πρόβλημα, με την δημιουργία μιας διαδικτυακής εφαρμογής που αποσκοπεί στην διευκόλυνση και υποστήριξη των φοιτητών, η οποία ονομάζεται Students Portal. Συγκεντρώνει χρήσιμες λειτουργίες οι οποίες εστιάζουν στην πολύπλευρη επαφή ενός σπουδαστή με το τμήμα του, χωρίς να απαιτείται η χρήση πολυάριθμων ιστοτόπων και υπηρεσιών. Αξίζει να αναφερθεί πως στην ιστοσελίδα μπορούν για αντίστοιχους λόγους να συμμετάσχουν φυσικά και οι καθηγητές του εκάστοτε εκπαιδευτικού ιδρύματος. Αφού δημιουργήσει τον λογαριασμό του, ο χρήστης αποκτά πρόσβαση σε έναν ικανό αριθμό από εργαλεία. Δύναται να προβάλλει ανακοινώσεις που περιλαμβάνουν οτιδήποτε σχετικό με την σχολή του, όπως μια πιθανή αλλαγή σε ώρα εξέτασης. Να συμμετάσχει σε ομαδικές συζητήσεις με τους υπόλοιπους χρήστες ή να μοιραστεί αρχεία. Επιπρόσθετα, να καταθέσει τις ερωτήσεις του αναφορικά με κάποιο μάθημα ή συμβάν, και αντίστοιχα να δώσει τις απαντήσεις του στις απορίες των υπολοίπων. Να σημειώσει events σε δυναμικό ημερολόγιο τα οποία είναι ορατά σε όλα τα μέλη της διαδικτυακής εφαρμογής, όπως μια συνάντηση του φοιτητικού συλλόγου, και να επικοινωνήσει με email με τον δημιουργό της εφαρμογής. Συμπερασματικά, πρόκειται για μια ενιαία πλατφόρμα η οποία με γνώμονα την αμεσότητα και την αποδοτικότητα υποβοηθά τόσο την ενημέρωση και επαφή ενός σπουδαστή με το τμήμα του, όσο και την αλληλοϋποστήριξη με τους συναδέλφους του.

# 1. ΕΙΣΑΓΩΓΗ

## 1.1 Αντικείμενο της εργασίας

Κεντρικό αντικείμενο της παρούσας εργασίας είναι η δημιουργία μιας διαδικτυακής εφαρμογής, αντίστοιχης με ένα forum, η οποία θα απευθύνεται τόσο σε φοιτητές όσο και σε καθηγητές σχολών Πληροφορικής και ονομάζεται Students Portal. Η εφαρμογή διαθέτει πολυάριθμες λειτουργίες και εργαλεία που μπορούν να αποδειχθούν χρήσιμα σε κάθε μέλος της.

Έπειτα από την δημιουργία του λογαριασμού του, ο χρήστης έχει την δυνατότητα να ενημερωθεί σχετικά με το τμήμα του, είτε μέσω της προβολής ανακοινώσεων, είτε μέσα από την περιήγηση στο δυναμικό ημερολόγιο, στο οποίο σημειώνονται σχετικά events.

Από την άλλη πλευρά, μπορεί να συναναστραφεί με τα υπόλοιπα μέλη της σχολής του, τόσο μέσω της προβολής αρχείων και της συμμετοχής του σε ομαδικές συζητήσεις, όσο και με την κατάθεση ερωτήσεων και επίλυση αντίστοιχων αποριών άλλων χρηστών.

## 1.2 Σκοπός της εργασίας

Κύριος σκοπός της εργασίας και κατά συνέπεια της διαδικτυακής εφαρμογής, είναι να υποβοηθήσει την ενημέρωση και την πλήρη επαφή των χρηστών με το εκπαιδευτικό ίδρυμα Πληροφορικής στο οποίο ανήκουν. Παράλληλα, να παρέχει μια πλατφόρμα δικτύωσης και αλληλοϋποστήριξης μεταξύ των μελών του, με σκοπό όχι μόνο την επικοινωνία και την ανταλλαγή απόψεων, αλλά και την επίλυση αποριών.

Η θεμέλια παράμετρος η οποία ενισχύει και τονίζει την χρησιμότητα της εφαρμογής είναι ότι οι λειτουργίες που εξυπηρετούν τους προαναφερθέντες σκοπούς, συγκεντρώνονται σε έναν μόνο ιστότοπο και παρουσιάζονται στοχευμένα. Σε διαφορετική περίπτωση οι χρήστες θα έπρεπε να ανατρέξουν σε πολλές διαφορετικές πηγές για να έχουν πρόσβαση σε ανάλογα εργαλεία μεμονωμένα, κάτι που αντιστοιχεί σε μια σημαντικά πιο χρονοβόρα διαδικασία.

Συμπερασματικά, η εφαρμογή αποσκοπεί στο να ικανοποιήσει τις ανάγκες ενημέρωσης και επαφής κάθε χρήστη με την σχολή Πληροφορικής που ανήκει και τα μέλη αυτής αντίστοιχα, με γνώμονα την εύκολη προσβασιμότητα και τον σεβασμό στον χρόνο του.

### 1.3 Διάρθρωση της εργασίας

Κατά πρώτον, υπάρχει το εισαγωγικό Κεφάλαιο 1, το οποίο σε ξεχωριστές ενότητες περιγράφει το αντικείμενο και τον σκοπό που εξυπηρετεί η εργασία, καθώς επίσης και τον συνοπτικό ορισμό στοιχειωδών όρων με τους οποίους ενδέχεται να μην είναι εξοικειωμένος ο αναγνώστης.

Το κύριο κομμάτι της εργασίας παρουσιάζεται στο Κεφάλαιο 2, το οποίο χωρίζεται σε οκτώ διαφορετικές ενότητες. Κάθε ενότητα αντιστοιχεί σε ένα θεμελιώδες τμήμα της διαδικτυακής εφαρμογής, με την πρώτη να αφορά την γενική δομή της αρχικής σελίδας. Η δεύτερη αναλύει το σημαντικότερο κομμάτι, το σύστημα εγγραφής και σύνδεσης των χρηστών στην ιστοσελίδα. Συνεχίζουμε με τις λειτουργίες ενημέρωσης, που αντιστοιχούν στις ανακοινώσεις και το δυναμικό ημερολόγιο. Έπειτα ακολουθεί μια ενότητα για τις δυο συμπληρωματικές λειτουργίες, την επικοινωνία και την προβολή προφίλ. Ακόμη, μια ενότητα για κάθε μια από τις τρεις λειτουργίες επαφής μεταξύ των χρηστών, που αποτελούνται από την κατάθεση αποριών, την ομαδική συζήτηση και την προβολή αρχείων. Ολοκληρώνουμε με μια ενότητα που αναλύει τις γενικότερες αλλαγές που πραγματοποιήθηκαν στην εμφάνιση της ιστοσελίδας.

Κάθε υποκεφάλαιο εξετάζει πολύπλευρα το κάθε τμήμα του forum, αναλύοντας τόσο την διεπαφή χρήστη και την εμφάνιση, όσο και την προγραμματιστική δομή και λειτουργικότητα που εξασφαλίζονται από την χρήση ανάλογων γλωσσών.

Το Κεφάλαιο 3 συνιστά ένα παράρτημα με εικόνες από την διαδικτυακή εφαρμογή, ώστε ο αναγνώστης του κειμένου να πάρει άμεσα μια ιδέα σχετικά με την ιστοσελίδα χωρίς να χρειαστεί να την επισκεφθεί.

Το Κεφάλαιο 4 συζητά τα συμπεράσματα που εξήχθησαν έπειτα από την υλοποίηση της εργασίας και πιθανές βελτιωτικές προτάσεις.

Τέλος, Το Κεφάλαιο 5 περιλαμβάνει την βιβλιογραφία, όπου αναφέρονται οι πηγές στις οποίες κατά τόπους στηρίχθηκε ο συγγραφέας κατά την εκπόνηση της εργασίας.

## 1.4 Λέξεις - Κλειδιά

-HTML: Γλώσσα προγραμματισμού που χρησιμοποιείται για τον προσδιορισμό της βασικής δομής μιας ιστοσελίδας.

-Tag: Θεμελιώδης όρος της γλώσσας HTML. Διαφορετικά είδη tags εξυπηρετούν συγκεκριμένους σκοπούς ανάλογα με τον τρόπο χρήσης τους.

-CSS: Γλώσσα προγραμματισμού που εστιάζει στην επεξεργασία και την βελτίωση της εμφάνισης μιας ιστοσελίδας.

-Class: Προστίθεται σε στοιχεία HTML και χρησιμοποιείται για την επεξεργασία της εμφάνισης τους.

-PHP: Γλώσσα προγραμματισμού που χρησιμοποιείται για την προσθήκη δυναμικού περιεχομένου και λειτουργικότητας σε ιστοσελίδες.

-Μεταβλητή: Χρησιμοποιείται σε προγράμματα για αποθήκευση πληροφορίας διαφόρων τύπων, όπως αριθμητικών τιμών ή κειμένου.

-Συνάρτηση: Εκτελέσιμο τμήμα κώδικα που εξυπηρετεί μια συγκεκριμένη λειτουργία.

-Javascript: Γλώσσα προγραμματισμού που χρησιμοποιείται για την προσθήκη διαδραστικών στοιχείων σε ιστοσελίδες.

-Βιβλιοθήκη: Έτοιμα πακέτα κώδικα, πληροφοριών και ρυθμίσεων που χρησιμοποιούνται σε προγράμματα.

-SQL: Γλώσσα υπολογιστή που εστιάζει στην ενημέρωση, ανάκτηση και διαχείριση δεδομένων σε βάσεις δεδομένων.

-Βάση δεδομένων: Οργανωμένη συλλογή δεδομένων διαφόρων τύπων, που αποθηκεύεται και είναι προσβάσιμη ηλεκτρονικά.

-Πίνακας: Στοιχείο αποθήκευσης πληροφορίας σε βάσεις δεδομένων, περιλαμβάνει γραμμές και στήλες που δημιουργούν κελιά, στα οποία αποθηκεύονται συγκεκριμένες τιμές.

-Ερώτημα SQL: Διαφορετικοί τύποι ερωτημάτων εξυπηρετούν ανάλογες λειτουργίες σε μια βάση δεδομένων, όπως το ερώτημα delete που διαγράφει εγγραφές από την βάση.



## 2. ΛΕΙΤΟΥΡΓΙΕΣ

### 2.1 Αρχική σελίδα

Το πρώτο πράγμα που συναντά ο χρήστης μόλις εισέλθει στο Students Portal είναι η αρχική σελίδα. Στην παρούσα ενότητα θα αναλύσουμε την γενική δομή και εμφάνιση της, οι οποίες αλλάζουν ανάλογα με το αν ο επισκέπτης χρησιμοποιεί mobile ή desktop συσκευή.

Στην κορυφή της σελίδας βλέπουμε πάνω αριστερά τον τίτλο του forum, με τέσσερις συνδέσμους να ακολουθούν. Συγκεκριμένα, ο σύνδεσμος Sign Up μεταφέρει τους επισκέπτες σε μια νέα σελίδα στην οποία δημιουργούν τους λογαριασμούς τους συμπληρώνοντας ορισμένα απαραίτητα στοιχεία που θα τους ζητηθούν.

Μόλις ολοκληρωθεί η διαδικασία, τα νέα μέλη του Students Portal χρησιμοποιούν το σύνδεσμο Log In προκειμένου να εισέλθουν με τους νέους τους λογαριασμούς στο forum, προϋπόθεση απαραίτητη έτσι ώστε να έχουν πλήρη πρόσβαση στο σύνολο των λειτουργιών του. Ο ίδιος σύνδεσμος χρησιμοποιείται και για την έξοδο των μελών από την ιστοσελίδα. Ακολουθούν οι σύνδεσμοι Contact και Profile. Στον πρώτο μπορούν να καταφύγουν οι χρήστες που θέλουν να στείλουν email στον δημιουργό του Students Portal, ενώ ο δεύτερος οδηγεί σε μια σελίδα που παρουσιάζει τα βασικά στοιχεία του λογαριασμού του κάθε μέλους. Συνεχίζουμε με το banner του forum κι ένα μήνυμα καλωσορίσματος σε αυτό, αλλά και το κεντρικό μενού που περιλαμβάνει 6 ακόμη συνδέσμους, έναν για καθεμία από τις σημαντικότερες και πιο συχνά χρησιμοποιούμενες λειτουργίες που παρέχονται.

Ο σύνδεσμος Announcements μεταφέρει τους χρήστες στην σελίδα προβολής ανακοινώσεων για διάφορα θέματα που αφορούν την σχολή τους, ενώ ο σύνδεσμος Messages τους επιτρέπει να συμμετάσχουν σε μια ομαδική συζήτηση με τα υπόλοιπα μέλη για οτιδήποτε τους απασχολεί.

Στη συνέχεια υπάρχουν οι σύνδεσμοι Calendar και Files. Ο πρώτος αντιστοιχεί σε ένα ημερολόγιο στο οποίο οι χρήστες μπορούν να σημειώσουν events όπως μια έκτακτη πρόοδο για κάποιο μάθημα, με τον δεύτερο να οδηγεί σε μια σελίδα προβολής και διαμοιρασμού αρχείων, όπως διαφάνειες διαλέξεων.

Προχωράμε με τους συνδέσμους Questions, που οδηγεί σε μια σελίδα όπου οι χρήστες καταθέτουν τις απορίες τους και απαντούν σε ενδεχόμενες ερωτήσεις άλλως μελών. Και τέλος Contact, ο οποίος όπως σημειώθηκε παραπάνω αφορά την αποστολή emails.

Καταλήγουμε στο κάτω μέρος του ιστοτόπου, το οποίο χωρίζεται σε δύο τμήματα.

Στο αριστερό τμήμα συναντάμε τον προαναφερθέντα σύνδεσμο Profile, τον σύνδεσμο About Me, ο οποίος συνοπτικά παρουσιάζει μερικές πληροφορίες για τον δημιουργό του forum σε απλό κείμενο, και το κουμπί Go to top, που οδηγεί πίσω στην κορυφή της αρχικής σελίδας.

Στο δεξί τμήμα υπάρχουν τρεις σύνδεσμοι με την μορφή εικονιδίων, τα οποία αντιστοιχούν σε λογαριασμούς μέσω κοινωνικής δικτύωσης, σε περίπτωση που οι χρήστες προτιμήσουν αυτόν τον τρόπο επικοινωνίας για να στείλουν προτάσεις ή να ζητήσουν διευκρινίσεις.

Υπογραμμίζεται πως η δομή και εμφάνιση που περιγράφηκε ως τώρα, συνιστά την desktop έκδοση του Students Portal. Εάν ο επισκέπτης χρησιμοποιεί mobile συσκευή το περιεχόμενο παραμένει ίδιο, ωστόσο η αρχική σελίδα μεταβάλλεται αισθητά ως προς την εμφάνιση της.

Συγκεκριμένα, αλλάζει η θέση του τίτλου και των συνδέσμων στην κορυφή, το φόντο, και κυρίως η διάταξη του κεντρικού μενού, με τις έξι λειτουργίες να ακολουθούν ένα κάθετο τρόπο παρουσίασης. Το κάτω μέρος της σελίδας παραμένει ως έχει και στις δυο εκδόσεις.

Από την άλλη πλευρά, θα πρέπει να κάνουμε μια αναφορά και στο προγραμματιστικό περίγραμμα που αφορά την αρχική σελίδα, μιλώντας με περισσότερο τεχνικούς όρους.

Για τον προσδιορισμό της δομής της χρησιμοποιήθηκε η γλώσσα HTML στο αρχείο index.php. Για αρχή έχουμε τα <html>, <head>, <title>, <link>, και <body> tags, τα οποία συναντώνται σε κάθε τυπικό HTML αρχείο κι επομένως δεν χρήζουν λεπτομερούς αναφοράς.

Κατά πρώτον, η κορυφή της αρχικής σελίδας περικλείεται σε ένα <header> tag. Εντός του έχουμε ένα <nav> tag που περιλαμβάνει το σύνολο των συνδέσμων που ακολουθούν, οι οποίοι δημιουργούνται με τα <a> tags. Οι σύνδεσμοι Log In, Sign Up και Contact, που βρίσκονται στα δεξιά του τίτλου, έχουν τοποθετηθεί σε μια λίστα με το <ul> tag. Οι τίτλοι των συνδέσμων δίνονται μέσω των <h3> tags.

Στη συνέχεια, το κεντρικό μέρος της αρχικής σελίδας περιλαμβάνεται σε ένα <main> tag και χωρίζεται σε δύο τμήματα. Κάθε ένα από τα δύο τμήματα περιγράφεται με ένα <section> tag. Το πρώτο περιέχει το φόντο και το μήνυμα καλωσορίσματος, ενώ το δεύτερο περικλείει το κύριο μενού, το οποίο αποτελείται από έξι <a> tags με τους αντίστοιχους συνδέσμούς τους.

Τέλος, το κάτω μέρος της σελίδας βρίσκεται εντός του <footer> tag. Οι τρεις σύνδεσμοι στο αριστερό τμήμα τοποθετούνται σε λίστα με το <ul> tag, ενώ μέσω του τυπικού <img> tag προστίθενται εικονίδια στους τρεις συνδέσμους που βρίσκονται στο δεξιό τμήμα.

Συνεχίζουμε με μια αναφορά στα βασικότερα στοιχεία της CSS που χρησιμοποιήθηκε, της γλώσσας που είναι υπεύθυνη για την επεξεργασία της γενικής εμφάνισης της αρχικής σελίδας, με το αρχείο style.css. Δεν κρίνεται απαραίτητο να υπάρξει ανάλυση για εντελώς θεμελιώδεις όρους, όπως αυτών που απευθύνονται στην μεταβολή των περιθωρίων ή της μορφοποίησης των γραμματοσειρών. Παραδείγματα τέτοιων κανόνων είναι μεταξύ άλλων τα margin, padding, font-family/size/weight, color, όροι υπερβολικά συγκεκριμένοι για να έχει νόημα εκτενής αναφορά σε αυτούς. Αντίθετα θα εστιάσουμε σε σημεία που επηρεάζουν την διάταξη ή την εμφάνιση πολλαπλών στοιχείων της αρχικής σελίδας ταυτόχρονα.

Οι πρώτες 50 γραμμές του αρχείου πραγματοποιούν styling reset, ακυρώνοντας ενδεχόμενη προεπιλεγμένη μορφοποίηση που προσδίδουν ορισμένοι περιηγητές σε ιστοσελίδες. Στο αρχείο index.php που περιλαμβάνει τον HTML κώδικα, αντιστοιχίζουμε classes στα αντικείμενα της αρχικής σελίδας, όπως τα sections και τις λίστες. Προσθέτουμε ακόμη <div> tags στους συνδέσμους, έτσι ώστε να τοποθετήσουμε και σε αυτούς classes.

Χρησιμοποιώντας αυτές τις κλάσεις η CSS καθιστά δυνατό τον ακριβή προσδιορισμό της επιθυμητής εμφάνισης των επιμέρους στοιχείων, όπως την επιλογή μιας συγκεκριμένης γραμματοσειράς και την μετέπειτα επεξεργασία της θέσης, του χρώματος και μεγέθους της.

Συγκεκριμένα στοιχεία του αρχείου style.css χρήζουν ωστόσο εκτενέστερης περιγραφής. Μέσω του head-logo class στο <header> tag σχηματίζεται το λογότυπο του forum, το οποίο βρίσκεται αριστερά στην κορυφή της αρχικής σελίδας. Ακόμη, ο κανόνας display:inline-block στοιχίζει την λίστα των τριών συνδέσμων κάτω από το λογότυπο, σε μια οριζόντια γραμμή. Έπειτα με τον κανόνα background-image στην κλάση index-banner, προστίθεται το φόντο στο κεντρικό τμήμα. Μέσω της επεξεργασίας των <div> tags εντός του index-links class, τροποποιείται το σύνολο των συνδέσμων του κεντρικού μενού, ώστε να παρουσιάζονται ως ορθογώνια κουμπιά. Τέλος, με την επεξεργασία του footer tag στοιχίζονται οι σύνδεσμοι στα αριστερά του κάτω μέρους της σελίδας, ενώ μέσω του social class τοποθετούνται τα εικονίδια των κοινωνικών δικτύων στα δεξιά. Σε περίπτωση που ο χρήστης περιηγείται στην ιστοσελίδα χρησιμοποιώντας οθόνη μικρότερη από 1000 pixels, βλέπει την mobile έκδοση.

Μόλις ξεπεραστεί το συγκεκριμένο όριο περνά στην desktop εκδοχή, με την CSS να εκτελεί αντίστοιχες αλλαγές στην εμφάνιση της αρχικής σελίδας, με την χρήση του κανόνα `@media only screen and (min-width: 1000px)`.

Κατά πρώτον, με τους κανόνες `float:left` που αφορούν το `<header>` tag αλλάζει η διάταξη του λογοτύπου και της λίστας των τριών συνδέσμων στην κορυφή, ώστε όλα πλέον να στοιχίζονται σε μια οριζόντια γραμμή. Ακόμη, με το `head-prof class` προστίθεται ένας ακόμη σύνδεσμος με όνομα `Profile`, ο οποίος έχει την μορφή ορθογώνιου κουμπιού και τοποθετείται στα δεξιά. Έπειτα επιλέγεται ένα διαφορετικό φόντο για το κεντρικό τμήμα της σελίδας, με τον κανόνα `background-image` της κλάσης `index-banner`.

Η σημαντικότερη όμως αλλαγή παρατηρείται στην εμφάνιση και διάταξη των έξι συνδέσμων του κύριου μενού. Μέσω του `linkRectangle class`, οι σύνδεσμοι `Announcements` και `Questions` παίρνουν την μορφή μεγάλων ορθογώνιων κουμπιών, ενώ με το `linkSquare class` οι σύνδεσμοι `Messages`, `Calendar`, `Files` και `Contact` παρουσιάζονται ως τετράγωνα κουμπιά. Τέλος, οι ίδιοι σύνδεσμοι πλέον δεν στοιχίζονται κάθετα, αλλά μοιράζονται σε δύο πανομοιότυπα οριζόντια τμήματα.

## 2.2 Σύστημα σύνδεσης-εγγραφής

Το σύστημα σύνδεσης-εγγραφής είναι ίσως το σημαντικότερο κομμάτι της ιστοσελίδας, καθώς διαχειρίζεται την δημιουργία των λογαριασμών των μελών καθώς κι ένα καίριο κομμάτι της ασφάλειας.

Η πρώτη κίνηση που χρειάζεται να κάνει ένας νέος επισκέπτης στο forum, είναι να επιλέξει τον σύνδεσμο `Sign Up` στην κορυφή της αρχικής σελίδας προκειμένου να δημιουργήσει τον λογαριασμό του, διαφορετικά δεν θα έχει πρόσβαση σε καμία από τις χρησιμότερες λειτουργίες.

Εκεί συναντά μια φόρμα στην οποία συμπληρώνει το ονοματεπώνυμο, το email, το όνομα χρήστη και τον κωδικό του, το τμήμα Πληροφορικής που ανήκει αλλά και αν είναι φοιτητής ή καθηγητής σε αυτό. Μόλις συμπληρώσει όλα τα στοιχεία ολοκληρώνει την διαδικασία με το κουμπί `Sign Up`, ενώ με το κουμπί `Reset` διαγράφει τα στοιχεία του και ξεκινά απ την αρχή.

Προτού προχωρήσουμε στα αρχεία που διαχειρίζονται όσα αναφέρθηκαν, χρειάζεται να

αναφερθούμε στην βάση δεδομένων που ήταν απαραίτητο να δημιουργηθεί. Ονομάζεται `students-portal` και προς το παρόν περιλαμβάνει δύο πίνακες. Ο πρώτος πίνακας λέγεται `unis` και περιέχει δύο στήλες, την `uni_id`, που είναι ο αριθμός του πανεπιστημίου που ανήκει ο χρήστης και πρωτεύον κλειδί, και την `uni_name`, που συνιστά το όνομα του πανεπιστημίου.

Ο δεύτερος πίνακας ονομάζεται `users` και αφορά τα μέλη του forum. Περιλαμβάνει 8 στήλες με ονόματα `user_id`, `user_first`, `user_last`, `user_email`, `user_uid`, `user_pwd`, `user_role`, και `uni_id`. Τα αντίστοιχα στοιχεία που αντιπροσωπεύουν είναι ο αριθμός, το όνομα και επώνυμο, το email, το όνομα χρήστη, ο κωδικός, ο ρόλος, και το id του πανεπιστημίου των μελών. Πρωτεύον κλειδί είναι το `user_id`.

Στη συνέχεια χρειάζεται να συνδεθεί η βάση δεδομένων με την ιστοσελίδα. Αυτό επιτυγχάνεται μέσω του αρχείου `dbh.php`. Σε αυτό, περνούν το όνομα του server, το username και ο κωδικός από την υπηρεσία `phpmyadmin`, και το όνομα της βάσης σε αντίστοιχες `php` μεταβλητές. Έπειτα εκτελείται η συνάρτηση `mysqli_connect` με τις μεταβλητές ως εισόδους, και περνά το αποτέλεσμα σε μια νέα μεταβλητή με όνομα `$conn`. Αν η σύνδεση αποτύχει τυπώνουμε ανάλογο μήνυμα.

Όσον αφορά τα αρχεία που είναι υπεύθυνα για την λειτουργία εγγραφής, αρχικά έχουμε το `supF.php`. Ξεκινά ένα session και ακολουθεί η φόρμα εγγραφής, η οποία περιλαμβάνει επτά πεδία εισόδου. Τα πρώτα πέντε αφορούν το όνομα, το επώνυμο, το email, το όνομα χρήστη και τον κωδικό. Τα τελευταία δύο πεδία σχετίζονται με το πανεπιστήμιο του νέου μέλους καθώς και με το αν είναι μαθητής ή καθηγητής σε αυτό, με τις διαθέσιμες επιλογές να εμφανίζονται σε ανάλογα drop-down μενού. Έαν το μέλος μετανιώσει για τα στοιχεία που πληκτρολόγησε μπορεί να τα επαναφέρει με το κουμπί `Reset`, διαφορετικά ολοκληρώνει την εγγραφή του με το κουμπί `Sign Up`.

Μόλις πατηθεί το κουμπί `Sign Up` εκτελείται το αρχείο `supB.php`. Ξεκινά ένα session, περιλαμβάνεται το αρχείο `dbh.php` για την επίτευξη σύνδεσης με την βάση δεδομένων, και με την συνάρτηση `isset` ελέγχεται εάν ο χρήστης πράγματι υπέβαλε την φόρμα εγγραφής.

Έπειτα οι τιμές των πεδίων εισόδου της φόρμας αντιστοιχίζονται σε `php` μεταβλητές αφού πρώτα περάσουν από την συνάρτηση `mysqli_real_escape_string`, εξασφαλίζοντας ότι τα SQL ερωτήματα που θα εκτελεστούν σχετικά με αυτές δεν θα περιέχουν ειδικούς χαρακτήρες και θα είναι αποδεκτά.

Στη συνέχεια πραγματοποιείται μια σειρά από ελέγχους με σκοπό να διασφαλιστεί η ομαλή και ασφαλής εγγραφή του νέου μέλους στο forum. Εάν οποιοσδήποτε από αυτούς αποτύχει, το μέλος παραμένει στην φόρμα εγγραφής και τυπώνεται ανάλογο μήνυμα σφάλματος.

Κατά πρώτον, με την php συνάρτηση `empty` ελέγχεται εάν κάποιο από τα πεδία εισόδου της φόρμας είναι κενό, ενώ με χρήση της συνάρτησης `preg_match` εάν πληκτρολογήθηκαν μη επιτρεπόμενοι χαρακτήρες στα πεδία ονόματος κι επωνύμου. Έπειτα με την συνάρτηση `filter_var` ελέγχεται εάν το email που εισήγαγε ο χρήστης είναι έγκυρο. Ακολουθούν έλεγχοι σχετικά με το αν το username και παραπλήσια το email που πληκτρολογήθηκαν είναι διαθέσιμα. Στις δύο αυτές περιπτώσεις χρειάζεται να εκτελεστούν ερωτήματα SQL, κι έτσι με δεδομένο πως η διαδικασία εγγραφής, όπως και αυτή της σύνδεσης, είναι κρίσιμης σημασίας, χρησιμοποιούμε prepared statements, τα οποία συνιστούν εναλλακτικούς τρόπους εκτέλεσης SQL ερωτημάτων. Αν και απαιτούν την εκτέλεση τεσσάρων συναρτήσεων για την λειτουργία τους, με ονόματα `mysqli_stmt_init / prepare/ bind_param` και `execute`, προσφέρουν δύο σημαντικά πλεονεκτήματα. Την προστασία από επιθέσεις τύπου SQL injection, οι οποίες είναι κυρίως γνωστές για την καταστροφή βάσεων δεδομένων, και την μείωση τόσο του φόρτου του περιηγητή όσο και του χρόνου εκτέλεσης των ερωτημάτων. Εάν παρατηρηθεί κάποιο σφάλμα κατά την διάρκεια χρήσης των prepared statements, τυπώνουμε αντίστοιχο μήνυμα. Σε περίπτωση που δεν υπήρξαν σφάλματα προχωρά η διαδικασία εγγραφής του χρήστη και κωδικοποιείται ο κωδικός του ώστε να είναι αδύνατο να φανεί μέσα στην βάση δεδομένων. Ακολουθεί ξανά η χρήση prepared statements για την εκτέλεση ενός SQL ερωτήματος insert, με αποτέλεσμα να εισαχθούν οι php μεταβλητές με τις τιμές των πεδίων εισόδου της φόρμας εγγραφής, στον πίνακα users της βάσης δεδομένων και να ολοκληρωθεί η εγγραφή.

Αφού έχει έτοιμο τον λογαριασμό του, το νέο μέλος επιλέγει το σύνδεσμο Log In στην κορυφή της αρχική σελίδας, και συμπληρώνοντας το όνομα χρήστη και τον κωδικό του εισέρχεται στο forum πατώντας το κουμπί Login, έχοντας πλέον πρόσβαση σε όλες τις λειτουργίες. Όπως και στην Sign Up φόρμα, υπάρχει ένα κουμπί επαναφοράς στοιχείων με όνομα Reset. Για να πραγματοποιήσει την έξοδο του από τον ιστότοπο, πρέπει να πατήσει το κουμπί Log Out που οδηγεί στην αρχική σελίδα.

Σχετικά με το προγραμματιστικό σκέλος της λειτουργίας σύνδεσης, έχουμε το αρχείο

linF.php. Αρχικά ξεκινά ένα session, κι έπειτα ελέγχεται εάν ο χρήστης είναι συνδεδεμένος, εξετάζοντας εάν υπάρχει το id του εντός του πίνακα \$\_SESSION.

Εάν είναι, προβάλλεται μήνυμα πως είναι ήδη συνδεδεμένος, καθώς και ένα κουμπί Log Out ώστε να μπορεί να πραγματοποιηθεί έξοδος από το forum. Διαφορετικά, προβάλλεται φόρμα Log In με πεδία εισόδου το όνομα χρήστη και τον κωδικό του. Και οι δύο επιλογές δημιουργούνται με το τυπικό <form> tag.

Σε περίπτωση που ο χρήστης πατήσει το κουμπί Login για να συνδεθεί, εκτελείται το αρχείο linB.php. Αρχικά ξεκινά ένα session, περιλαμβάνεται το αρχείο dbh.php για την σύνδεση με την βάση δεδομένων, και ελέγχεται εάν ο χρήστης πράγματι υπέβαλε την φόρμα σύνδεσης.

Έπειτα οι τιμές των πεδίων εισόδου της φόρμας αντιστοιχίζονται σε php μεταβλητές αφού πρώτα περάσουν από την συνάρτηση mysqli\_real\_escape\_string που αναφέρθηκε και προηγουμένως. Στη συνέχεια πραγματοποιείται μια αλληλουχία από ελέγχους ώστε να εξασφαλιστεί η έγκυρη και ασφαλής είσοδος του χρήστη. Εάν κάποιος από τους ελέγχους αποτύχει, ο χρήστης επιστρέφει στην αρχική σελίδα και τυπώνεται σχετικό μήνυμα σφάλματος στο url. Κατά πρώτον ελέγχεται εάν ο χρήστης δεν συμπλήρωσε ένα από τα δύο πεδία εισόδου. Έπειτα μέσω των προαναφερθέντων prepared statements ανατρέχουμε στην βάση δεδομένων για να επιλεγούν οι χρήστες με τους οποίους υπάρχει αντιστοιχία είτε σε όνομα χρήστη είτε σε email, και ελέγχεται μήπως δεν υπήρξε κάποιο αποτέλεσμα στην αναζήτηση. Εάν υπήρξε αποκωδικοποιείται ο κωδικός που εισήχθη στην βάση κατά την διαδικασία εγγραφής, και ελέγχεται εάν είναι έγκυρος. Εάν πράγματι είναι περνούν το id, το ονοματεπώνυμο, το email, το όνομα χρήστη και ο ρόλος, από τον πίνακα users της βάσης δεδομένων στον πίνακα \$\_SESSION. Επίσης με prepared statements, με βάση το id του αναζητείται το πανεπιστήμιο που έχει επιλέξει ο χρήστης, ώστε το όνομα του πανεπιστημίου να περάσει κι αυτό στον πίνακα \$\_SESSION. Εάν ολοκληρωθούν όλοι οι έλεγχοι χωρίς να παρουσιαστεί πρόβλημα, το μέλος πραγματοποιεί με επιτυχία την είσοδο του στο forum.

Τέλος, το αρχείο που πραγματοποιεί την έξοδο του χρήστη από την ιστοσελίδα ονομάζεται logout.php κι εκτελείται μόλις ο χρήστης πατήσει το κουμπί Log Out. Ελέγχεται εάν είναι συνδεδεμένος, και αν είναι, δημιουργείται και σε αυτό το αρχείο session, έπειτα καταστρέφεται και ο χρήστης αποσυνδέεται και ανακατευθύνεται στην αρχική σελίδα.

## 2.3 Ανακοινώσεις

Στην παρούσα ενότητα θα αναλυθεί η λειτουργία των ανακοινώσεων. Είναι διαθέσιμη μέσω του συνδέσμου Announcements στο κεντρικό μενού της αρχικής σελίδας. Με την είσοδο του στην σελίδα ο χρήστης βλέπει τον τίτλο της και την ροή ανακοινώσεων, οι οποίες χωρίζονται από μια οριζόντια γραμμή. Εάν η ανακοίνωση έχει υποβληθεί από τον ίδιο τον χρήστη, βλέπει ακόμη δύο κουμπιά που ονομάζονται Edit και Delete, που του επιτρέπουν να τροποποιήσει και να διαγράψει την ανακοίνωση του αντίστοιχα. Στο τέλος της σελίδας υπάρχει η επιλογή Home για επιστροφή στην αρχική, και ένας σύνδεσμος με όνομα Post Your Announcement, που επιτρέπει στον χρήστη να υποβάλλει την δική του ανακοίνωση.

Σε αυτή την σελίδα, στην κορυφή φαίνεται ένα μήνυμα καλωσορίσματος, ακολουθούμενο από τρία πεδία εισόδου με ονόματα title, subtitle, και content, που αφορούν τον τίτλο, τον υπότιτλο ή το θέμα, και το περιεχόμενο της ανακοίνωσης. Με το κουμπί Post πραγματοποιείται η υποβολή, και με το View Announcements η επιστροφή στην ροή ανακοινώσεων.

Για την διαχείριση των ανακοινώσεων προστίθεται ένας νέος πίνακας στην βάση δεδομένων students-portal με όνομα announcements, ο οποίος περιέχει πέντε στήλες. Την id, που είναι ο αριθμός της ανακοίνωσης και συνιστά το primary key του πίνακα, την uid, που είναι το όνομα του χρήστη που την υπέβαλε, και τις title, subtitle και content, που αποτελούν τον τίτλο, τον υπότιτλο και το περιεχόμενο των αναρτήσεων αντιστοίχως.

Σχετικά με το προγραμματιστικό σκέλος της λειτουργίας, έχουμε το αρχείο annsDisplay.php. Ξεκινά ένα session και με την συνάρτηση include περιλαμβάνονται το αρχείο dbh.php για να επιτευχθεί σύνδεση με την βάση δεδομένων, και το fnxs.php που περιέχει απαραίτητες συναρτήσεις που θα χρησιμοποιηθούν αργότερα. Ελέγχεται εάν ο χρήστης είναι συνδεδεμένος εξετάζοντας το id του εντός του πίνακα \$\_SESSION με την συνάρτηση isset. Εάν δεν είναι ανακατευθύνεται στην σελίδα Log In, διαφορετικά έχει πρόσβαση και βλέπει την ροή των ανακοινώσεων. Εκτελείται ένα SQL ερώτημα select, που επιλέγει τις ανακοινώσεις σε φθίνουσα σειρά με βάση το id τους, ώστε οι πιο πρόσφατες να φαίνονται πρώτα. Οι στήλες του πίνακα announcements της βάσης περνούν σε αντίστοιχες php μεταβλητές, και τυπώνονται με echo ώστε να εμφανιστεί η κάθε ανακοίνωση.



Στη συνέχεια εκτελείται η συνάρτηση `getAnns` από το αρχείο `fnxs.php` ώστε να εμφανιστούν τα κουμπιά `Edit` και `Delete` στα εξουσιοδοτημένα μέλη. Τέλος υπάρχουν δύο κουμπιά, το πρώτο οδηγεί στην αρχική σελίδα και το δεύτερο αφορά την υποβολή νέας ανακοίνωσης.

Το αρχείο `annsPost.php` αναλαμβάνει τις νέες αναρτήσεις. Ξεκινά ένα `session` και περιλαμβάνεται το αρχείο `dbh.php` για την σύνδεση με την βάση. Ελέγχεται εάν είναι συνδεδεμένος ο χρήστης και αν υποβλήθηκε από αυτόν η `html` φόρμα της νέας ανακοίνωσης. Τα δεδομένα περνούν από την φόρμα σε `php` μεταβλητές, εισέρχονται στις αντίστοιχες στήλες του πίνακα της βάσης με την εκτέλεση ενός `SQL` ερωτήματος `insert`, και τυπώνεται ειδοποίηση επιτυχούς ανάρτησης. Η φόρμα δημιουργείται με το τυπικό `<form>` tag και περιλαμβάνει τρία πεδία εισόδου, ένα για καθένα από τα τρία μέρη της ανακοίνωσης, τον τίτλο, το θέμα και το περιεχόμενο, ενώ υπάρχει κι ένα κουμπί που οδηγεί στην ροή ανακοινώσεων. Σαν επιπλέον δεδομένο στην φόρμα περνά το όνομα του χρήστη, ωστόσο μέσω της επιλογής `type=hidden` δεν είναι ορατό σε αυτόν.

Το αρχείο `annEdit.php` περιέχει την διεπαφή χρήστη για πιθανή επεξεργασία μιας ανακοίνωσης. Ως συνήθως ξεκινά ένα `session` και περιλαμβάνονται το αρχείο `dbh.php` για να επιτευχθεί σύνδεση με την βάση, και το αρχείο `fnxs.php` που περιέχει στοιχειώδεις `php` συναρτήσεις και θα αναλυθεί παρακάτω. Τα δεδομένα από την `html` φόρμα περνούν σε νέες `php` μεταβλητές. Ο χρήστης επεξεργάζεται το περιεχόμενο της ανακοίνωσης σε αντίστοιχο πεδίο που δημιουργείται με ένα `<textarea>` tag και περιέχεται σε μια νέα φόρμα. Ωστόσο αυτή τη φορά δεν μπορεί να δει και τα υπόλοιπα πεδία της, καθώς με την επιλογή `type=hidden` δεν είναι ορατά σε αυτόν. Μόλις πατήσει το κουμπί `Edit` εκτελείται η συνάρτηση `editAnn` του αρχείου `fnxs.php` και η ανακοίνωση μεταβάλλεται αναλόγως.

Στην συνέχεια εξηγείται το αρχείο `fnxs.php`, που όπως αναφέρθηκε ήδη περιλαμβάνει τρεις απαραίτητες για την λειτουργία των ανακοινώσεων συναρτήσεις, τις `getAnns`, `editAnn` και `deleteAnn`. Όλες δέχονται ως όρισμα την μεταβλητή `$conn`, στην οποία περνά το αποτέλεσμα της συνάρτησης `mysqli_connect` για την σύνδεση με την βάση δεδομένων, από το αρχείο `dbh.php`.

Η συνάρτηση `getAnns` δέχεται ως επιπλέον όρισμα την μεταβλητή `$row`, που περιέχει το αποτέλεσμα της αναζήτησης ανακοινώσεων από το ερώτημα `select` του αρχείου `annsDisplay.php`, και προβάλλει τα κουμπιά `Edit` και `Delete` στην σελίδα ροής. Όπως γίνεται

κατανοητό, τα κουμπιά αυτά δεν πρέπει να είναι ορατά σε όλα τα μέλη, αλλά μόνο στους χρήστες που έχουν αναρτήσει κάποια ανακοίνωση, και αποκλειστικά στις αναρτήσεις που έχουν κάνει οι ίδιοι. Τα ονόματα των χρηστών που τις υπέβαλαν περνούν από την μεταβλητή \$row σε μια νέα με όνομα \$id. Στη συνέχεια εκτελείται ένα SQL ερώτημα select κι αναζητούνται χρήστες από τον πίνακα users με κριτήριο την μεταβλητή \$id. Ελέγχεται εάν ο χρήστης είναι συνδεδεμένος με τον λογαριασμό του στο forum, κι έπειτα εάν το ID του ταιριάζει με αυτό των χρηστών που επιλέχθηκαν με το δεύτερο select. Εάν τα IDs είναι ίδια, σημαίνει πως ο συνδεδεμένος χρήστης έχει κάνει ο ίδιος την ανάρτηση, και άρα επιτρέπεται να έχει πρόσβαση στις επιλογές Edit και Delete.

Το κουμπί Edit, είναι ουσιαστικά μια φόρμα, κατά την υποβολή της οποίας εκτελείται το αρχείο annEdit.php που αναφέρθηκε παραπάνω. Στα πεδία της περιλαμβάνονται ως απαραίτητες πληροφορίες οι τιμές όλων των στηλών του πίνακα announcements της βάσης για την συγκεκριμένη ανακοίνωση, αλλά μέσω της επιλογής hidden κανένα από τα πεδία δεν είναι ορατά στον χρήστη. Το κουμπί Delete εκτελεί την συνάρτηση deleteAnn που περιγράφεται παρακάτω. Πρόκειται επίσης για μια φόρμα που περιέχει ως επιπλέον πληροφορία σε ένα κρυφό πεδίο το ID της ανακοίνωσης.

Η συνάρτηση editAnn πραγματοποιεί την επεξεργασία μιας ανάρτησης. Ελέγχεται εάν το μέλος πάτησε το κουμπί Edit της φόρμας επεξεργασίας, και οι πληροφορίες της φόρμας περνούν σε νέες php μεταβλητές. Εκτελείται ένα SQL ερώτημα update με κριτήριο το id της ανακοίνωσης ώστε μετά τις αλλαγές να ενημερωθεί η στήλη περιεχομένου του πίνακα της βάσης, κι ο χρήστης ανακατευθύνεται στην σελίδα ροής ανακοινώσεων.

Τέλος, η συνάρτηση deleteAnn αναλαμβάνει την διαγραφή μιας ανάρτησης. Ελέγχεται εάν ο χρήστης πάτησε το κουμπί Delete, και αν ναι εκτελείται ένα SQL ερώτημα delete με κριτήριο το id της ανακοίνωσης, το οποίο πάρθηκε από την προαναφερθείσα φόρμα μέσω της μεταβλητής \$\_POST. Διαγράφεται η ανάρτηση και γίνεται ανακατεύθυνση στην σελίδα ροής ανακοινώσεων.

## 2.4 Ομαδική συζήτηση

Στην ενότητα αυτή θα αναφερθούμε σε μια καίρια λειτουργία για την δικτύωση και την

επικοινωνία μεταξύ φοιτητών και καθηγητών, την ομαδική συζήτηση. Η λειτουργία είναι διαθέσιμη μέσω του συνδέσμου Messages του κεντρικού μενού της αρχικής σελίδας.

Στην κορυφή φαίνεται το όνομα χρήστη, ενώ ακολουθούν μια περιοχή στην οποία πληκτρολογείται το κάθε μήνυμα και το κουμπί Reply για την υποβολή του. Ακριβώς από κάτω βρίσκεται η ροή των μηνυμάτων σε φθίνουσα χρονολογική σειρά. Κάθε μήνυμα περιλαμβάνει το όνομα του χρήστη που το υπέβαλε, ακολουθούμενο από ένα αναγνωριστικό του ρόλου του, S εάν είναι φοιτητής (από την λέξη Student), και P εάν είναι καθηγητής (από την λέξη Professor). Στο τέλος υπάρχει επιλογή Back για επιστροφή στην αρχική σελίδα.

Για την διαχείριση των μηνυμάτων προστίθεται ένας ακόμη πίνακας στην βάση δεδομένων, με όνομα messages. Περιέχει τρεις στήλες, την id που αποτελεί τον αριθμό του μηνύματος και είναι το primary key του πίνακα, και τις username και msg, που συνιστούν το όνομα χρήστη και το περιεχόμενο του μηνύματος αντίστοιχα.

Σχετικά με τα αρχεία που είναι υπεύθυνα για την ομαδική συζήτηση, ξεκινάμε με το messages.php. Ελέγχεται εάν ο χρήστης είναι συνδεδεμένος εξετάζοντας εάν υπάρχει το id του εντός του πίνακα \$\_SESSION, και αν δεν είναι μεταφέρεται στην σελίδα Log In. Ακολουθεί η συνάρτηση υποβολής μηνύματος submitMsg. Αρχικά ελέγχεται εάν το μήνυμα είναι κενό, ώστε να προβληθεί ειδοποίηση ότι πρέπει να εισαχθεί κάτι. Περνά το περιεχόμενο του πεδίου εισόδου μηνύματος σε αντίστοιχη javascript μεταβλητή, η οποία θα χρειαστεί σε επόμενο αρχείο. Έπειτα ξεκινά ένα νέο http request, δημιουργείται μια συνάρτηση που προβάλλει το περιεχόμενο του αρχείου insertMsg.php, το οποίο αναλύεται αργότερα, και ανοίγει και στέλνεται το http request. Στη συνέχεια χρησιμοποιώντας μια ειδική javascript βιβλιοθήκη, εκτελούμε δύο συναρτήσεις. Η πρώτη απελευθερώνει την μνήμη cache του περιηγητή ώστε να μην φορτώνει παλιά δεδομένα. Η επόμενη ανανεώνει την ροή των μηνυμάτων κάθε 1 δευτερόλεπτο, ώστε να προβάλλονται συνεχώς τα νέα μηνύματα χωρίς να χρειάζεται να ανανεωθεί όλη η σελίδα, και με αυτή ολοκληρώνεται η συνάρτηση υποβολής μηνύματος.

Η διεπαφή χρήστη δημιουργείται με το τυπικό <form> tag, που περιλαμβάνει ένα <textarea> tag όπου πληκτρολογείται το μήνυμα. Από πάνω τυπώνεται το όνομα του χρήστη, ενώ από κάτω διακρίνεται το κουμπί υποβολής, με το πάτημα του οποίου εκτελείται η προαναφερθείσα συνάρτηση submitMsg. Ακολουθεί ένδειξη κατά την διάρκεια φόρτωσης

των μηνυμάτων, και τέλος υπάρχει επιλογή Back για επιστροφή στην αρχική σελίδα.

Το αρχείο insertMsg.php περνά το υποβληθέν μήνυμα στην βάση δεδομένων. Αρχικά ξεκινά ένα session και μέσω της συνάρτησης include περιλαμβάνεται το αρχείο dbh.php για να επιτευχθεί σύνδεση με την βάση. Μεταφέρεται το όνομα χρήστη και το περιεχόμενο του μηνύματος σε νέες php μεταβλητές, οι οποίες περνούν στην βάση με την εκτέλεση ενός SQL ερωτήματος insert.

Το αρχείο logs.php είναι υπεύθυνο για την ροή μηνυμάτων. Ως συνήθως ξεκινά ένα session και περιλαμβάνεται το αρχείο dbh.php για την επίτευξη σύνδεσης με την βάση. Εκτελείται ένα SQL ερώτημα select ώστε να επιλεγούν τα μηνύματα από την βάση δεδομένων σε φθίνουσα χρονολογική σειρά, και επίσης ελέγχεται ο ρόλος του χρήστη εξετάζοντας την τιμή u\_role εντός του πίνακα \$\_SESSION. Έπειτα τυπώνουμε το όνομα χρήστη στην ροή μηνυμάτων, ακολουθούμενο από το αναγνωριστικό του ρόλου του, S εάν είναι σπουδαστής ή P εάν είναι καθηγητής, και το περιεχόμενο του μηνύματος.

## 2.5 Συμπληρωματικές λειτουργίες

Οι λειτουργίες που θα χρησιμοποιούνται περισσότερο στην μεγάλη πλειοψηφία των επισκέψεων στο Students Portal είναι οι Announcements, Calendar, Files, Messages, και Questions. Ωστόσο υπάρχουν διαθέσιμες δύο ακόμη επιλογές που έχουν συμπληρωματικό ρόλο αλλά παρ'όλα αυτά την δική τους χρησιμότητα.

Κατά πρώτον, στο κάτω αριστερό και στο πάνω δεξί τμήμα της αρχικής σελίδας υπάρχει ο σύνδεσμος Profile. Σε αυτή τη σελίδα προβάλλονται τα στοιχεία που επέλεξαν τα μέλη κατά την δημιουργία του λογαριασμού τους στο forum. Συγκεκριμένα, βλέπουν το όνομα, το email, το username, τον ρόλο και το τμήμα τους. Εάν ένας χρήστης κρίνει πως τα στοιχεία που επέλεξε είτε είναι λάθος είτε δεν ισχύουν πλέον, είναι ελεύθερος να αλλάξει τον λογαριασμό του. Το αντίστοιχο αρχείο που εξυπηρετεί την προβολή προφίλ ονομάζεται prof.php. Αρχικά ξεκινά ένα session, περιλαμβάνεται το αρχείο dbh.php για την σύνδεση με την βάση δεδομένων, και ελέγχεται εάν έχει συνδεθεί ο χρήστης. Εάν δεν έχει συνδεθεί, μεταφέρεται στην σελίδα login, διαφορετικά συνεχίζει. Αφού εξεταστεί εάν είναι διαθέσιμο το id εντός του πίνακα \$\_SESSION, περνούν οι υπόλοιπες πληροφορίες από το session σε

αντίστοιχες php μεταβλητές. Τυπώνονται με echo το όνομα, το mail, το username, ο ρόλος και το τμήμα, και τέλος υπάρχει ένα κουμπί που οδηγεί στην αρχική σελίδα.

Συνεχίζουμε με την δεύτερη συμπληρωματική λειτουργία, μέσω της οποίας τα μέλη δύναται να στείλουν emails, και βρίσκεται στο σύνδεσμο Contact του κεντρικού μενού.

Στηρίζεται από το αρχείο contact.php, που περιέχει μια φόρμα στην οποία τα μέλη συμπληρώνουν το όνομα τους, καθώς και το θέμα και το περιεχόμενο του μηνύματος που επιθυμούν να στείλουν. Ωστόσο πρώτα ελέγχεται εάν ο χρήστης είναι συνδεδεμένος στην σελίδα, μέσω της εξέτασης του id του εντός του πίνακα \$\_SESSION με την συνάρτηση isset. Εάν δεν είναι μεταφέρεται στην σελίδα σύνδεσης, διαφορετικά έχει πρόσβαση. Όταν πατηθεί το κουμπί αποστολής της φόρμας, εκτελείται το αρχείο conForm.php. Αρχικά ελέγχεται εάν έχει υποβληθεί η φόρμα, και οι πληροφορίες της περνούν μέσω της μεταβλητής \$\_POST σε νέες php μεταβλητές, όπως και η διεύθυνση του χρήστη μέσω του πίνακα \$\_SESSION. Το mail αποστέλλεται με χρήση της php συνάρτησης mail, που δέχεται ως ορίσματα την διεύθυνση του αποστολέα και του παραλήπτη, το θέμα του μηνύματος, και μια ειδοποίηση στον παραλήπτη όταν δεχθεί μήνυμα.

## 2.6 Προβολή αρχείων

Στην ενότητα αυτή θα αναφερθούμε στην λειτουργία προβολής αρχείων, η οποία είναι προσβάσιμη μέσω του συνδέσμου Files στο κεντρικό μενού της αρχικής σελίδας. Μόλις μεταβεί στην συγκεκριμένη σελίδα, ο χρήστης βλέπει ένα μήνυμα καλωσορίσματος, κι ακριβώς από κάτω τα αρχεία που έχουν ήδη αναρτηθεί, καθένα με την δική του περιγραφή κι εντός ενός περιθωρίου. Ακολουθεί η επιλογή νέας ανάρτησης αρχείου από τον ίδιο τον χρήστη, μαζί με μια ανάλογη περιγραφή, ενώ στο τέλος αυτής της σελίδας υπάρχει η επιλογή Back η οποία οδηγεί πίσω στην αρχική.

Κατά πρώτον, προστέθηκε ένας πίνακας στην βάση δεδομένων για την διαχείριση των αρχείων με όνομα Files, ο οποίος περιλαμβάνει τέσσερις στήλες. Η πρώτη είναι η id, που είναι ο αριθμός του αρχείου και συνιστά το πρωτεύον κλειδί του πίνακα. Η δεύτερη είναι η uid, που αποτελεί το όνομα του χρήστη που ανέβασε το αρχείο. Η τρίτη είναι η file, που περιέχει τα αρχεία, και η τέταρτη η description, που αφορά την συνοδευτική περιγραφή.

Το αρχείο που είναι υπεύθυνο για την υλοποίηση της λειτουργίας ονομάζεται files.php. Ξεκινά ένα session και περιλαμβάνονται το αρχείο dbh.php ώστε να επιτευχθεί σύνδεση με την βάση δεδομένων, και το αρχείο del.php που θα αναλυθεί αργότερα. Αφού ελεγχθεί εάν έχει υποβληθεί η φόρμα υποβολής αρχείου που αναφέρθηκε προηγουμένως, περνούν σε ξεχωριστές php μεταβλητές το όνομα του χρήστη που υπέβαλε το αρχείο και οι τιμές που περιέχει η μεταβλητή \$\_FILES, όπως το όνομα, το είδος και το μέγεθος των αρχείων. Η μεταβλητή \$allowed περιέχει έναν πίνακα με όλους τους επιτρεπόμενους τύπους αρχείων, μεταξύ άλλων jpg, .png, pdf, ή txt.

Ακολουθούν έλεγχοι που εξασφαλίζουν την ομαλή υποβολή των αρχείων βάσει ορισμένων περιορισμών που τίθενται. Κατά πρώτον ελέγχεται εάν ο τύπος του αρχείου που επιθυμεί να υποβάλει ο χρήστης ανήκει στους επιτρεπόμενους. Έπειτα εάν υπήρξε κάποιο σφάλμα κατά την διάρκεια υποβολής, καθώς και εάν το αρχείο είναι μεγέθους έως 9 MB. Στη συνέχεια περνά η περιγραφή του αρχείου σε μια νέα php μεταβλητή, εισάγεται το όνομα χρήστη, το ίδιο το αρχείο και η περιγραφή του στην βάση δεδομένων, και ανεβαίνει το αρχείο στην σελίδα Files του forum.

Προκειμένου ο χρήστης να μπορεί να έχει πρόσβαση στην λειτουργία αυτή, θα πρέπει να είναι συνδεδεμένος με τον λογαριασμό του, γι'αυτό και γίνεται ο ανάλογος έλεγχος εξετάζοντας το id του εντός του πίνακα \$\_SESSION.

Έαν ο χρήστης δεν έχει συνδεθεί ανακατευθύνεται στην σελίδα Log In. Συνεχίζουμε εντός του αρχείου files.php με το μήνυμα καλωσορίσματος, και την επιλογή των αρχείων από την βάση δεδομένων. Αφού ελεγχθεί η κατάληξη του αρχείου, τοποθετείται στην ροή της σελίδας.

Εάν είναι εικόνα προβάλλεται απευθείας, ενώ σε διαφορετική περίπτωση, πχ αν είναι PDF, το αρχείο προβάλλεται με την μορφή συνδέσμου. Έπειτα εκτελείται η συνάρτηση getDel από το αρχείο del.php, η οποία θα αναλυθεί περισσότερο παρακάτω. Τέλος, η φόρμα υποβολής δημιουργείται μέσω αντίστοιχου <form> tag, και περιέχει επιλογή και περιγραφή αρχείου, καθώς και το όνομα του χρήστη ως επιπλέον πληροφορία, σε ένα πεδίο που είναι κρυφό με χρήση της επιλογής type=hidden. .

Το αρχείο del.php χρησιμοποιείται για την δημιουργία των κουμπιών διαγραφής αρχείων. Όπως γίνεται κατανοητό, τα κουμπιά αυτά δεν πρέπει να είναι ορατά σε όλα τα μέλη, αλλά

μόνο στους χρήστες που έχουν αναρτήσει κάποιο αρχείο, και αποκλειστικά σε αυτά που έχουν αναρτήσει οι ίδιοι. Για να συμβεί αυτό υπάρχει η συνάρτηση `getDel`, που εμφανίζει το κουμπί διαγραφής στην σελίδα. Δέχεται ως ορίσματα την μεταβλητή `$conn`, στην οποία περνά το αποτέλεσμα της συνάρτησης `mysqli_connect` για την σύνδεση με την βάση δεδομένων, από το αρχείο `dbh.php`, και την μεταβλητή `$row`, που περιέχει το αποτέλεσμα της αναζήτησης αρχείων από το ερώτημα `select` του αρχείου `files.php`. Τα ονόματα των χρηστών που υπέβαλαν τα αρχεία περνούν από την μεταβλητή `$row` σε μια νέα με όνομα `$id`. Εκτελείται ένα ακόμη SQL ερώτημα `select`, που επιλέγει `users` με κριτήριο το `username` τους να ταιριάζει με αυτό της μεταβλητής `$id`. Εάν ο χρήστης είναι συνδεδεμένος στο `forum` και το όνομα του ταιριάζει με αυτά που αναζητήθηκαν με το δεύτερο `select`, έχει πρόσβαση στο κουμπί `Delete`. Αυτό δημιουργείται με `<form>` tag που περιέχει σε κρυφό πεδίο ως επιπλέον πληροφορία τον αριθμό του αρχείου, και μόλις το κουμπί πατηθεί εκτελείται η συνάρτηση `deleteFile`.

Η συνάρτηση `deleteFile` πραγματοποιεί την διαγραφή του αρχείου και δέχεται επίσης ως όρισμα την μεταβλητή `$conn` από το αρχείο `dbh.php`. Ελέγχεται εάν πατήθηκε το κουμπί διαγραφής και ο αριθμός του αρχείου περνά μέσω της μεταβλητής `$_POST` σε `php` μεταβλητή με όνομα `$fid`. Με κριτήριο την νέα αυτή μεταβλητή εκτελείται SQL ερώτημα `delete` στον πίνακα `files`, το αρχείο διαγράφεται και ο χρήστης ανακατευθύνεται στην ροή αρχείων.

Όσον αφορά τον κώδικα CSS που χρησιμοποιήθηκε, βρίσκεται στο αρχείο `style5.css` και περιλαμβάνεται στην σελίδα μέσω του `<link>` tag στο αρχείο `files.php`. Μέσω της επεξεργασίας των `form` και `img` tags οριοθετούνται τα επιθυμητά μεγέθη των εικόνων και της φόρμας υποβολής. Στο αρχείο `files.php` έχουν τοποθετηθεί δύο `<div>` tags, ένα που αφορά ολόκληρη την ροή της σελίδας με όνομα `content`, κι ένα ακόμη που αφορά το κάθε υποβληθέν αρχείο ξεχωριστά. Με την επεξεργασία των `divs` τοποθετούνται τα αντίστοιχα περιθώρια πλάτους 1 pixel και καθορίζονται τα μεγέθη τους.

## 2.7 Κατάθεση αποριών

Στην παρούσα ενότητα θα αναλυθεί η λειτουργία της κατάθεσης αποριών. Είναι διαθέσιμη

μέσω του συνδέσμου Questions στο κεντρικό μενού της αρχικής σελίδας. Με την είσοδο του στην σελίδα ο χρήστης βλέπει τον τίτλο της και την ροή των ερωτήσεων, οι οποίες χωρίζονται από μια οριζόντια γραμμή. Σε κάθε ερώτηση υπάρχει ένα πεδίο εισόδου και δίπλα σε αυτό ένα κουμπί με όνομα Reply, με χρήση των οποίων κάθε χρήστης μπορεί να δώσει την δική του απάντηση. Ακολουθούν οι απαντήσεις που έχουν δοθεί σε κάθε ερώτηση εντός γκριζών πεδίων που χωρίζονται από ένα μικρό κενό. Στην κορυφή του πεδίου διακρίνεται το όνομα του μέλους που την υπέβαλε και δίπλα σε αυτό το αναγνωριστικό του ρόλου του, (S) εάν είναι σπουδαστής και (P) εάν είναι καθηγητής. Έπειτα φαίνονται η ημερομηνία και ώρα ανάρτησης, και από κάτω το περιεχόμενο της. Οι χρήστες που έχουν υποβάλλει κάποια απάντηση βλέπουν ακόμη δύο κουμπιά στο πάνω δεξί μέρος του ανάλογου πεδίου, με ονόματα Edit και Delete, μέσω των οποίων μπορούν να επεξεργαστούν ή να διαγράψουν τις απαντήσεις τους, αλλά όπως αναμένεται μόνο εάν ανήκουν στους ίδιους. Στο τέλος της σελίδας υπάρχει η επιλογή Home για επιστροφή στην αρχική, και ένας σύνδεσμος με όνομα Ask a Question, που επιτρέπει στον χρήστη να καταθέσει την δική του απορία.

Σε αυτή την σελίδα, στην κορυφή φαίνεται ένα μήνυμα καλωσορίσματος, ακολουθούμενο από τρία πεδία εισόδου με ονόματα subject, title, και content, που αφορούν την θεματολογία, τον τίτλο, και το περιεχόμενο της απορίας. Με το κουμπί Submit πραγματοποιείται η υποβολή, και με το View Questions η επιστροφή στην ροή ερωτήσεων.

Για την διαχείριση των δεδομένων της λειτουργίας προστίθενται δύο νέοι πίνακες στην βάση δεδομένων students-portal. Ο πρώτος λέγεται questions, αφορά τις ερωτήσεις και περιέχει πέντε στήλες. Την id, που είναι ο αριθμός της ερώτησης και συνιστά το primary key του πίνακα, την uid, που είναι το όνομα του χρήστη που την υπέβαλε, και τις subject, title και content, που αποτελούν την θεματολογία, τον τίτλο και το περιεχόμενο των αναρτήσεων αντιστοίχως. Ο δεύτερος πίνακας ονομάζεται replies και περιέχει πέντε στήλες. Την rid, που είναι ο αριθμός της απάντησης και το primary key του πίνακα, την uid, που είναι το όνομα του μέλους που την υπέβαλε, την qid, που είναι ο αριθμός της ερώτησης στην οποία υποβλήθηκε η απάντηση, την date, που είναι η ημερομηνία υποβολής, και τέλος την reply, που αφορά το περιεχόμενο της απάντησης.

Σχετικά με το προγραμματιστικό σκέλος της λειτουργίας, έχουμε το αρχείο qtsDisplay.php.

Ξεκινά ένα session και με την συνάρτηση include περιλαμβάνονται το αρχείο dbh.php για να



επιτευχθεί σύνδεση με την βάση δεδομένων, και τα replies.php και fnxs.php, που περιέχουν απαραίτητες συναρτήσεις που θα χρησιμοποιηθούν αργότερα. Με την συνάρτηση date\_default\_timezone\_set αλλάζει η προεπιλεγμένη ζώνη ώρας σε αυτή της Αθήνας. Έπειτα ελέγχεται εάν ο χρήστης είναι συνδεδεμένος εξετάζοντας το id του εντός του πίνακα \$\_SESSION με την συνάρτηση isset. Εάν δεν είναι ανακατευθύνεται στην σελίδα Log In, διαφορετικά έχει πρόσβαση και βλέπει την ροή των αποριών. Εκτελείται ένα SQL ερώτημα select, που επιλέγει τις απορίες σε φθίνουσα σειρά με βάση το id τους, ώστε οι πιο πρόσφατες να φαίνονται πρώτα. Οι στήλες του πίνακα questions της βάσης περνούν σε αντίστοιχες php μεταβλητές, και τυπώνονται με echo ώστε να εμφανιστεί η κάθε απορία. Ακολουθεί μια φόρμα με τέσσερα πεδία, με τρία από αυτά να είναι κρυφά. Το μόνο ορατό στον χρήστη είναι το πεδίο εισόδου απάντησης, που δημιουργείται από ένα <textarea> tag, και το κουμπί υποβολής της, με όνομα Reply. Τα τρία κρυφά πεδία είναι το όνομα του χρήστη που υπέβαλε την απάντηση, η ημερομηνία υποβολής, και ο αριθμός της ερώτησης. Δίνονται μέσω του πίνακα \$\_SESSION, της συνάρτησης date, και της μεταβλητής \$row που περιέχει το αποτέλεσμα αναζήτησης ερωτήσεων από το ερώτημα select, αντίστοιχα. Στη συνέχεια εκτελείται η συνάρτηση getReplies από το αρχείο replies.php η οποία θα αναλυθεί αργότερα, συνοπτικά ωστόσο εμφανίζει τις απαντήσεις που έχουν δοθεί σε κάθε απορία. Έπειτα ελέγχεται εάν δόθηκε κάποια απάντηση σε ερώτηση καθώς και αν δεν ήταν κενή. Αν ισχύουν και τα δύο εκτελείται η συνάρτηση setReplies, που επίσης θα περιγραφεί παρακάτω. Γίνεται ανακατεύθυνση στην σελίδα ροής αποριών και πραγματοποιείται έξοδος από τον κώδικα. Η πρακτική αυτή χρησιμοποιείται ώστε οι απαντήσεις να εμφανίζονται μόνο μια φορά και μόνο στην απορία που αντιστοιχούν. Ακολουθεί javascript έλεγχος ώστε να αποτρέπεται η επανυποβολή της φόρμας σε περίπτωση που ο χρήστης ανανεώσει την σελίδα. Τέλος υπάρχουν δύο κουμπιά, το Home που οδηγεί στην αρχική σελίδα και το Ask a Question που αφορά την κατάθεση νέας απορίας.

Το αρχείο qtsPost αναλαμβάνει τις καταθέσεις νέων ερωτήσεων. Ξεκινά ένα session και περιλαμβάνεται το αρχείο dbh.php για την σύνδεση με την βάση. Ελέγχεται εάν είναι συνδεδεμένος ο χρήστης και αν υποβλήθηκε από αυτόν η html φόρμα της νέας ερώτησης. Τα δεδομένα περνούν από την φόρμα σε php μεταβλητές με χρήση της μεταβλητής \$\_POST, εισέρχονται στις αντίστοιχες στήλες του πίνακα της βάσης με την εκτέλεση ενός SQL

ερωτήματος insert, και τυπώνεται ειδοποίηση επιτυχούς ανάρτησης. Η φόρμα δημιουργείται με το τυπικό <form> tag και περιλαμβάνει τρία πεδία εισόδου, ένα για καθένα από τα τρία μέρη της ερώτησης, την θεματολογία, τον τίτλο και το περιεχόμενο, ενώ υπάρχει κι ένα κουμπί που οδηγεί στην ροή αποριών.

Το αρχείο qtsEdit.php περιέχει την διεπαφή χρήστη για την επεξεργασία μιας ερώτησης. Ξεκινά ένα session και περιλαμβάνονται το αρχείο dbh.php για την επίτευξη σύνδεσης με την βάση, και το αρχείο fnxs.php που περιέχει απαραίτητες php συναρτήσεις και θα αναλυθεί παρακάτω. Τα δεδομένα από την html φόρμα περνούν σε νέες php μεταβλητές. Ο χρήστης επεξεργάζεται το περιεχόμενο της ερώτησης σε αντίστοιχο πεδίο που δημιουργείται με ένα <textarea> tag και περιέχεται σε μια νέα φόρμα. Ωστόσο αυτή τη φορά δεν μπορεί να δει και τα υπόλοιπα πεδία της, καθώς με την επιλογή type=hidden δεν είναι ορατά σε αυτόν. Μόλις πατήσει το κουμπί Edit εκτελείται η συνάρτηση editQts του αρχείου fnxs.php και η ερώτηση μεταβάλλεται αναλόγως.

Στην συνέχεια επεξηγείται το αρχείο fnxs.php, που όπως αναφέρθηκε ήδη περιλαμβάνει τρεις απαραίτητες για την λειτουργία των ερωτήσεων συναρτήσεις, τις getQts, editQts και deleteQts. Όλες δέχονται ως όρισμα την μεταβλητή \$conn, στην οποία περνά το αποτέλεσμα της συνάρτησης mysqli\_connect για την σύνδεση με την βάση δεδομένων, από το αρχείο dbh.php.

Η συνάρτηση getQts δέχεται ως επιπλέον όρισμα την μεταβλητή \$row, που περιέχει το αποτέλεσμα της αναζήτησης ερωτήσεων από το ερώτημα select του αρχείου annsDisplay.php, και προβάλλει τα κουμπιά Edit και Delete στην σελίδα ροής. Όπως γίνεται κατανοητό, τα κουμπιά αυτά δεν πρέπει να είναι ορατά σε όλα τα μέλη, αλλά μόνο στους χρήστες που έχουν αναρτήσει κάποια ερώτηση, και αποκλειστικά στις αναρτήσεις που έχουν κάνει οι ίδιοι. Τα ονόματα των χρηστών που τις υπέβαλαν περνούν από την μεταβλητή \$row σε μια νέα με όνομα \$id. Στη συνέχεια εκτελείται ένα SQL ερώτημα select κι αναζητούνται χρήστες από τον πίνακα users με κριτήριο την μεταβλητή \$id. Ελέγχεται εάν ο χρήστης είναι συνδεδεμένος με τον λογαριασμό του στο forum, κι έπειτα εάν το ID του ταιριάζει με αυτό των χρηστών που επιλέχθηκαν με το δεύτερο select. Εάν τα IDs είναι ίδια, σημαίνει πως ο συνδεδεμένος χρήστης έχει κάνει ο ίδιος την ανάρτηση, και άρα επιτρέπεται να έχει πρόσβαση στις επιλογές Edit και Delete.

Το κουμπί Edit, είναι ουσιαστικά μια φόρμα, κατά την υποβολή της οποίας εκτελείται το αρχείο qtsEdit.php που αναφέρθηκε παραπάνω. Στα πεδία της περιλαμβάνονται ως απαραίτητες πληροφορίες οι τιμές όλων των στηλών του πίνακα questions της βάσης για την συγκεκριμένη ανακοίνωση, αλλά μέσω της επιλογής hidden κανένα από τα πεδία δεν είναι ορατά στον χρήστη. Το κουμπί Delete εκτελεί την συνάρτηση deleteQts που περιγράφεται παρακάτω. Πρόκειται επίσης για μια φόρμα που περιέχει ως επιπλέον πληροφορία σε ένα κρυφό πεδίο το ID της ερώτησης.

Η συνάρτηση editQts πραγματοποιεί την επεξεργασία μιας ερώτησης. Ελέγχεται εάν το μέλος πάτησε το κουμπί Edit της φόρμας επεξεργασίας, και οι πληροφορίες της φόρμας περνούν σε νέες php μεταβλητές. Εκτελείται ένα SQL ερώτημα update με κριτήριο το id της ερώτησης ώστε μετά τις αλλαγές να ενημερωθεί η στήλη περιεχομένου του πίνακα της βάσης, κι ο χρήστης ανακατευθύνεται στην σελίδα ροής αποριών.

Η συνάρτηση deleteQts αναλαμβάνει την διαγραφή μιας ερώτησης. Ελέγχεται εάν ο χρήστης πάτησε το κουμπί Delete, και αν ναι εκτελείται ένα SQL ερώτημα delete με κριτήριο το id της ερώτησης, το οποίο πάρθηκε από την προαναφερθείσα φόρμα μέσω της μεταβλητής \$\_POST. Διαγράφεται η ανάρτηση και γίνεται ανακατεύθυνση στην σελίδα ροής αποριών.

Από την άλλη πλευρά, το αρχείο editReply.php περιέχει την διεπαφή χρήστη για πιθανή επεξεργασία μιας απάντησης. Ως συνήθως ξεκινά ένα session και περιλαμβάνονται το αρχείο dbh.php για να επιτευχθεί σύνδεση με την βάση, και το αρχείο replies.php που περιέχει στοιχειώδεις php συναρτήσεις και θα περιγραφεί παρακάτω. Τα δεδομένα από την html φόρμα περνούν σε νέες php μεταβλητές. Ο χρήστης επεξεργάζεται το περιεχόμενο της απάντησης σε αντίστοιχο πεδίο που δημιουργείται με ένα <textarea> tag και περιέχεται σε μια νέα φόρμα. Ωστόσο αυτή τη φορά δεν μπορεί να δει και τα υπόλοιπα πεδία της, καθώς με την επιλογή type=hidden δεν είναι ορατά σε αυτόν. Μόλις πατήσει το κουμπί Edit εκτελείται η συνάρτηση editReplies του αρχείου replies.php και η απάντηση μεταβάλλεται αναλόγως.

Στην συνέχεια εξηγείται το αρχείο replies.php, που όπως αναφέρθηκε ήδη περιλαμβάνει τέσσερις απαραίτητες για την λειτουργία των ερωτήσεων συναρτήσεις, τις setReplies, getReplies, editReplies και deleteReplies. Όλες σχετίζονται με την διαχείριση των απαντήσεων, και δέχονται ως όρισμα την μεταβλητή \$conn, στην οποία περνά το αποτέλεσμα της συνάρτησης mysqli\_connect για την σύνδεση με την βάση δεδομένων, από το αρχείο

dbh.php.

Η συνάρτηση `setReplies` χρησιμοποιείται για την εισαγωγή των απαντήσεων στην βάση δεδομένων. Μέσω της μεταβλητής `$_POST` το όνομα χρήστη, η ημερομηνία, ο αριθμός της ερώτησης και το περιεχόμενο της απάντησης περνούν από την φόρμα σε νέες `php` μεταβλητές. Έπειτα εκτελείται ένα SQL ερώτημα `insert` που εισάγει τις τιμές των μεταβλητών στις αντίστοιχες στήλες του πίνακα `replies` της βάσης.

Η συνάρτηση `getReplies` δέχεται ως επιπλέον όρισμα την μεταβλητή `$row`, που περιέχει το αποτέλεσμα της αναζήτησης ερωτήσεων από το ερώτημα `select` του αρχείου `qtsDisplay.php`, και προβάλλει τόσο τις απαντήσεις που δόθηκαν σε κάθε ερώτηση, όσο και τα κουμπιά `Edit` και `Delete`. Όπως γίνεται κατανοητό, τα κουμπιά αυτά δεν πρέπει να είναι ορατά σε όλα τα μέλη, αλλά μόνο στους χρήστες που έχουν υποβάλλει κάποια απάντηση, και αποκλειστικά στις απαντήσεις που έχουν δώσει οι ίδιοι. Για να συμβεί αυτό, τα ονόματα των χρηστών που τις υπέβαλαν περνούν σε μια νέα `php` μεταβλητή που λέγεται `$id`. Στη συνέχεια εκτελείται ένα ακόμα SQL ερώτημα `select` κι αναζητούνται χρήστες από τον πίνακα `users` με κριτήριο την μεταβλητή `$id`. Σε περίπτωση που ο χρήστης είναι φοιτητής, το αναγνωριστικό του ρόλου του παίρνει την τιμή (S), ενώ αν είναι καθηγητής παίρνει την τιμή (P). Σε κάθε απάντηση εμφανίζονται μέσω συναρτήσεων `echo` το όνομα του χρήστη και από δίπλα ο ρόλος του, κάτω από αυτά η ημερομηνία υποβολής της απάντησης, και τέλος ακολουθεί το κείμενο της απάντησης. Έπειτα ελέγχεται εάν ο χρήστης είναι συνδεδεμένος με τον λογαριασμό του στο `forum`, κι έπειτα εάν το ID του ταιριάζει με αυτό των χρηστών που επιλέχθηκαν με το δεύτερο `select`. Εάν τα IDs είναι ίδια, σημαίνει πως ο συνδεδεμένος χρήστης έχει υποβάλλει ο ίδιος την απάντηση, και άρα επιτρέπεται να έχει πρόσβαση στις επιλογές `Edit` και `Delete`.

Το κουμπί `Edit`, είναι ουσιαστικά μια φόρμα, κατά την υποβολή της οποίας εκτελείται το αρχείο `editReply.php` που αναφέρθηκε παραπάνω. Στα πεδία της περιλαμβάνονται ως απαραίτητες πληροφορίες οι τιμές όλων των στηλών του πίνακα `replies` της βάσης για την συγκεκριμένη απάντηση, αλλά μέσω της επιλογής `hidden` κανένα από τα πεδία δεν είναι ορατά στον χρήστη. Το κουμπί `Delete` εκτελεί την συνάρτηση `deleteReply` που περιγράφεται παρακάτω. Πρόκειται επίσης για μια φόρμα που περιέχει ως επιπλέον πληροφορία σε ένα κρυφό πεδίο το ID της απάντησης.

Η συνάρτηση `editReplies` πραγματοποιεί την επεξεργασία μιας απάντησης. Ελέγχεται εάν το

μέλος πάτησε το κουμπί Edit της φόρμας επεξεργασίας, και οι πληροφορίες της φόρμας περνούν σε νέες php μεταβλητές. Εκτελείται ένα SQL ερώτημα update με κριτήριο το id της απάντησης ώστε μετά τις αλλαγές να ενημερωθεί η στήλη περιεχομένου του πίνακα της βάσης, κι ο χρήστης ανακατευθύνεται στην σελίδα ροής ερωτήσεων.

Η συνάρτηση deleteReplies αναλαμβάνει την διαγραφή μιας απάντησης. Ελέγχεται εάν ο χρήστης πάτησε το κουμπί Delete, και αν ναι εκτελείται ένα SQL ερώτημα delete με κριτήριο το id της απάντησης, το οποίο πάρθηκε μέσω της μεταβλητής \$\_POST. Διαγράφεται η απάντηση και γίνεται ανακατεύθυνση στην προηγούμενη σελίδα.

## 2.8 Δυναμικό ημερολόγιο

Στην παρούσα ενότητα θα αναλύσουμε την λειτουργία του δυναμικού ημερολογίου. Πρόκειται για ένα ημερολόγιο στο οποίο ωστόσο οι χρήστες δύναται να προσθέσουν γεγονότα που πρόκειται να συμβούν, όπως μια έκτακτη πρόοδο σε κάποιο μάθημα. Είναι διαθέσιμο μέσω του συνδέσμου Calendar στο κεντρικό μενού της αρχικής σελίδας.

Στην κορυφή της σελίδας, στο κέντρο φαίνεται ο τρέχων μήνας. Στα αριστερά του υπάρχουν τρία κουμπιά. Τα δύο πρώτα με σχήμα βέλους αναλαμβάνουν την μετάβαση σε προηγούμενο και επόμενο μήνα, και το τελευταίο με όνομα today πάντα μεταφέρει τον χρήστη στην σημερινή ημέρα, η οποία τονίζεται με κίτρινο χρώμα.

Στα δεξιά υπάρχουν ακόμη τρία κουμπιά με το καθένα να αλλάζει την εμφάνιση και δομή του ημερολογίου. Το πρώτο ονομάζεται month και αφορά την προεπιλεγμένη μηνιαία δομή, αυτή που συναντά ο χρήστης κάθε φορά που ανοίγει τον σύνδεσμο Calendar. Κάθε γραμμή αποτελεί μια εβδομάδα και καθεμία από τις επτά στήλες περιέχει μια από τις επτά μέρες της εβδομάδας, ώστε τα τετραγωνίδια που σχηματίζονται να είναι όλες οι μέρες του μήνα.

Το κουμπί week αλλάζει την δομή του ημερολογίου σε εβδομαδιαία. Κάθε στήλη είναι πλέον μια από τις επτά μέρες της τρέχουσας εβδομάδας, και κάθε δύο γραμμές αφορούν μια ώρα της ημέρας. Πάνω από τις διαθέσιμες ώρες υπάρχει μια ενιαία γραμμή με τίτλο all-day στα αριστερά της, που αντιστοιχεί σε ολόκληρη την ημέρα. Τα δύο κουμπιά με σχήμα βέλους πάνω αριστερά μεταβαίνουν πλέον σε προηγούμενη και επόμενη εβδομάδα, όχι μήνα.

Τέλος, το κουμπί day μετατρέπει την δομή του ημερολογίου σε ημερήσια. Στο κέντρο της

κορυφής της σελίδας φαίνεται η τρέχουσα ημερομηνία. Υπάρχει μόνο μια στήλη, η ημέρα, ενώ οι γραμμές είναι πανομοιότυπες με αυτές της εβδομαδιαίας δομής. Αλλάζουν ωστόσο τα πρώτα δύο κουμπιά στο πάνω αριστερά μέρος της σελίδας, ώστε να μεταβαίνουν σε προηγούμενη κι επόμενη μέρα αντί για εβδομάδα ή μήνα.

Σε οποιαδήποτε από τις 3 διαθέσιμες δομές, όταν ο χρήστης πατήσει κλικ σε κάποιο από τα πεδία που σχηματίζονται από τις γραμμές και τις στήλες, ανοίγει ένα παράθυρο που προτρέπει τον χρήστη να προσθέσει ένα γεγονός. Αφού εισάγει τον τίτλο, το επιλεγμένο πεδίο χρωματίζεται πράσινο και προβάλλεται ο τίτλος και η ώρα του γεγονότος. Εάν ο χρήστης επιθυμεί να το διαγράψει, απλώς κάνει ακόμη μια φορά κλικ πάνω στο πεδίο και επιβεβαιώνει την διαγραφή, ενώ αν θέλει να αλλάξει την διάρκεια του μπορεί να σύρει το γεγονός ώστε να καλύπτει περισσότερα ή λιγότερα πεδία.

Για την διαχείριση των events προστίθεται ένας νέος πίνακας στην βάση δεδομένων students-portal, με όνομα events. Περιέχει 4 στήλες. Την id, που είναι ο αριθμός του γεγονότος και αποτελεί το primary key του πίνακα, και τις title, start\_event και end\_event, που συνιστούν τον τίτλο, την έναρξη και την λήξη των events αντίστοιχα.

Σχετικά με το προγραμματιστικό σκέλος της λειτουργίας, κατά πρώτον έχουμε το αρχείο index.php, στην αρχή του οποίου ξεκινά ένα session. Μέσω των δύο HTML <link> tags περιλαμβάνεται ο κώδικας CSS που αφορά την εμφάνιση και δομή του ημερολογίου, και με τα <script> tags περιλαμβάνονται ορισμένες στοιχειώδεις για την λειτουργία του βιβλιοθήκες. Με την συνάρτηση isset εξετάζεται αν υπάρχει το id του χρήστη εντός του πίνακα \$\_SESSION, ώστε να ελεγχθεί εάν είναι συνδεδεμένος. Εάν δεν είναι ανακατευθύνεται στον σύνδεσμο Log In, διαφορετικά έχει πρόσβαση στο ημερολόγιο. Ακολουθεί μια συνάρτηση που δίνει την δυνατότητα στον χρήστη να αλλάξει την αρχή και το τέλος των γεγονότων με την επιλογή editable, ενώ το header τοποθετεί τα τρία προαναφερθέντα κουμπιά στο αριστερό και δεξί μέρος της κορυφής της σελίδας, και ανάμεσα τους τον τίτλο του ημερολογίου ανάλογα με την επιλεγμένη δομή του. Στην συνέχεια ενεργοποιείται το αρχείο load.php που θα αναλυθεί παρακάτω. Οι επιλογές selectable και selectHelper επιτρέπουν στον χρήστη να επιλέξει πολλαπλά πεδία του ημερολογίου. Οι javascript μεταβλητές title, start, end και id που θα χρησιμεύσουν στις επόμενες συναρτήσεις, περιλαμβάνουν τον τίτλο, την αρχή, την λήξη και το id του event αντίστοιχα. Η μεταβλητή title εμφανίζει και μια ένδειξη στον χρήστη για

να εισάγει τον επιθυμητό τίτλο, και για την προσθήκη του event εκτελείται το αρχείο `insert.php`. Εάν ολοκληρωθεί επιτυχώς, εμφανίζεται ανάλογη ειδοποίηση και ανανεώνεται η σελίδα. Σε περίπτωση που ο χρήστης θελήσει να αλλάξει την διάρκεια ενός event ή να το μετακινήσει, ενεργοποιούνται οι αντίστοιχες συναρτήσεις `eventResize` και `eventDrop`. Εάν οι αλλαγές ολοκληρωθούν χωρίς πρόβλημα, εμφανίζεται σχετική ειδοποίηση και ανανεώνεται η σελίδα του ημερολογίου. Αξίζει να αναφερθεί πως σε όλες τις συναρτήσεις χρησιμοποιείται μέθοδος POST για την αποστολή των δεδομένων στον server. Τέλος, η συνάρτηση `eventClick` επιτρέπει στον χρήστη να διαγράψει ένα event εκτελώντας το αρχείο `delete.php`, και όπως αναμένεται, εμφανίζει παράθυρο με ανάλογη ειδοποίηση εάν η διαγραφή ολοκληρωθεί επιτυχώς, και ανανεώνει την σελίδα.

Στην συνέχεια θα αναλύσουμε τα αρχεία `load`, `insert`, `update` και `delete` που αναφέρθηκαν παραπάνω. Το αρχείο `load.php` είναι υπεύθυνο για να φορτώνει στο ημερολόγιο τα events που υπάρχουν στην βάση δεδομένων. Αρχικά πραγματοποιείται σύνδεση με την βάση, με χρήση της συνάρτησης PDO, η οποία δέχεται ως ορίσματα τα ονόματα του server και της βάσης, και το username και τον κωδικό από την υπηρεσία `phpmyadmin`, με το αποτέλεσμα να περνά στην `php` μεταβλητή `connect`. Τόσο σε αυτό όσο και στα επόμενα αρχεία που θα αναφερθούμε, τα αναγκαία SQL ερωτήματα περιλαμβάνουν χρήση `prepared statements`, τα οποία απαιτούν δύο συναρτήσεις για την λειτουργία τους, την `prepare` για την προετοιμασία τους και την `execute` για την εκτέλεση τους. Συνεχίζοντας, δημιουργείται ένας `php` πίνακας με όνομα `$data` και εκτελείται ένα SQL ερώτημα `select`. Επιλέγονται τα events σε αύξουσα σειρά με γνώμονα το `id` τους, περνούν στον πίνακα `$data` το `id`, ο τίτλος, η αρχή και το τέλος τους, και προβάλλονται με την συνάρτηση `echo`.

Το αρχείο `insert.php` εκτελεί την προσθήκη ενός event στην βάση δεδομένων. Αφού επιτευχθεί η σύνδεση με την βάση, εξετάζεται εάν έχει υποβληθεί ο τίτλος του event μέσω ελέγχου της μεταβλητής `$_POST`. Αν ναι, εκτελείται ένα SQL ερώτημα `insert` και οι τιμές από την μεταβλητή `$_POST` περνούν στις στήλες του πίνακα `events` ώστε να εισαχθεί το event στην βάση.

Σε περίπτωση που ο χρήστης αποφασίσει να αλλάξει την ώρα έναρξης και λήξης ενός event στο δυναμικό ημερολόγιο, το αρχείο `update.php` αναλαμβάνει να μεταβάλλει τις αντίστοιχες τιμές στην βάση δεδομένων. Αρχικά πραγματοποιείται σύνδεση με την βάση, και εξετάζεται

εάν υπάρχει το id του event εντός της μεταβλητής \$\_POST. Έπειτα εκτελείται ένα SQL ερώτημα update ώστε να ενημερωθούν οι στήλες start\_event και end\_event του πίνακα events.

Ολοκληρώνουμε με το αρχείο delete.php, που πραγματοποιεί την διαγραφή ενός event. Αφού υπάρξει σύνδεση με την βάση δεδομένων, εξετάζεται εάν υπάρχει το id του event μέσα στην μεταβλητή \$\_POST, και εκτελείται ένα SQL ερώτημα delete με βάση αυτό το id, ώστε να διαγραφεί το συγκεκριμένο event.

## 2.9 Κώδικας επεξεργασίας εμφάνισης

Στην παρούσα ενότητα περιγράφονται στοιχειώδη τμήματα του κώδικα CSS που προστέθηκε στην πλειοψηφία των λειτουργιών της εφαρμογής. Τα αρχεία CSS συμπεριλαμβάνονται στα υπόλοιπα με χρήση του HTML <link> tag. Για να γίνει δυνατή η επεξεργασία της εμφάνισης, προστίθενται classes στα ανάλογα HTML tags των σελίδων, όπως τα <div> και κυρίως μέσω του χειρισμού αυτών πραγματοποιούνται αλλαγές. Ως “κανόνα” εντός των CSS αρχείων, ορίζουμε αυτόν ακριβώς τον χειρισμό είτε των classes, είτε των ίδιων των tags απευθείας.

Στα υποκεφάλαια 2.1 και 2.6 έχει ήδη γίνει αναφορά στα αρχεία style.css και style5.css, τα οποία χρησιμοποιήθηκαν στην επεξεργασία της εμφάνισης της αρχικής σελίδας και της λειτουργίας προβολής αρχείων αντιστοίχως. Προς αποφυγή επανάληψης, στην συγκεκριμένη ενότητα αναλύεται ο κώδικας CSS που προστέθηκε σε παραπάνω από ένα αρχείο, ώστε να μην χρειαστεί να γίνει αναφορά σε αυτόν σε κάθε ξεχωριστή λειτουργία.

Κατά πρώτον, υπάρχει το αρχείο style2.css, το οποίο περιλαμβάνεται στα αρχεία linF.php και supF.php, που αφορούν την διεπαφή χρήστη για τις σελίδες σύνδεσης και εγγραφής αντίστοιχα. Οι πρώτες 50 γραμμές του αρχείου πραγματοποιούν styling reset, ακυρώνοντας ενδεχόμενη προεπιλεγμένη μορφοποίηση που προσδίδουν ορισμένοι περιηγητές σε ιστοσελίδες. Έπειτα, στις σελίδες σύνδεσης και εγγραφής, προστίθεται γκριζό χρώμα στο φόντο και λευκό στην κορυφή.

Στη συνέχεια, έχουμε το αρχείο style4.css. Περιλαμβάνεται στα αρχεία annsDisplay.php και editReply.php, που συνιστούν τις σελίδες ροής ανακοινώσεων και επεξεργασίας απάντησης. Ακόμη, στα php αρχεία qtsDisplay, qtsPost και qtsEdit, που αφορούν τις σελίδες ροής



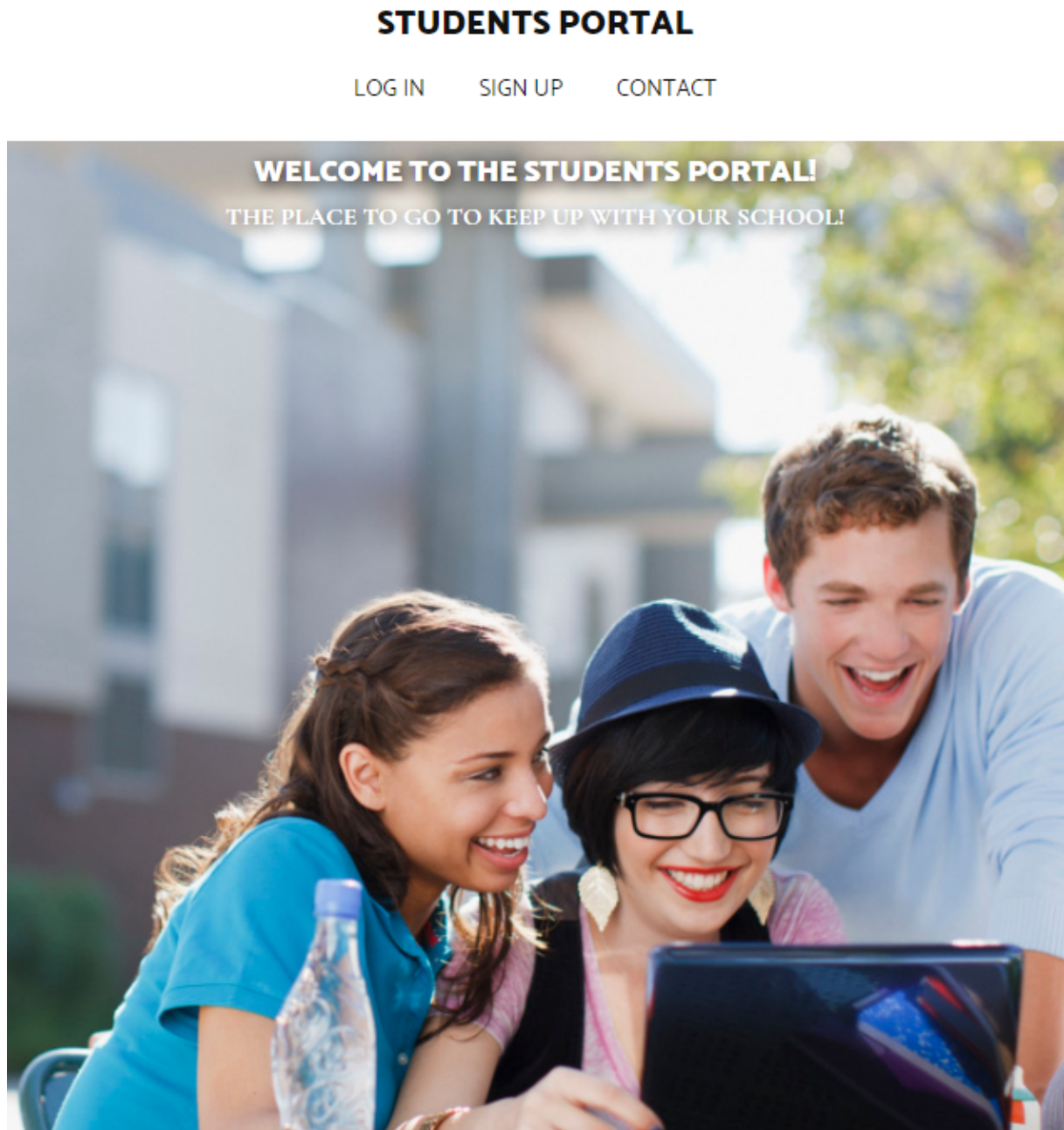
ερωτήσεων, ανάρτησης νέας ερώτησης και επεξεργασίας ερώτησης αντιστοίχως. Ο κανόνας `textarea` πραγματοποιεί αλλαγές στο πεδίο εισόδου κειμένου, ενώ οι κανόνες με όνομα `reply-box` βελτιώνουν την εμφάνιση των πεδίων απαντήσεων και του κειμένου που περιέχουν. Από την άλλη πλευρά, οι κανόνες με όνομα `edit-form` αφορούν την εμφάνιση της επεξεργασίας των αναρτήσεων, ενώ οι `delete-form` βελτιώνουν οπτικά τα κουμπιά διαγραφής, προσθέτοντας τους μεταξύ άλλων κόκκινο χρώμα.

Το τελευταίο αρχείο είναι το `main.css` και είναι αυτό που χρησιμοποιείται περισσότερο. Περιλαμβάνεται στα προαναφερθέντα `php` αρχεία `linF`, `supF`, `annsDisplay`, `qtsDisplay`, `qtsPost`, `qtsEdit`, και `editReply`. Επιπρόσθετα, στα `php` αρχεία `annsPost`, `annEdit`, `files`, `prof`, `contact`, και το `html` αρχείο `about`. Οι αντίστοιχες σελίδες είναι αυτές της ανάρτησης νέας ανακοίνωσης, επεξεργασίας ανακοίνωσης, ροής αρχείων, προβολής προφίλ, επικοινωνίας, και βασικών πληροφοριών για τον δημιουργό της εφαρμογής. Με τον κανόνα `body` τοποθετείται λευκό χρώμα στο φόντο των σελίδων, ενώ ο `form-container` χειρίζεται τις φόρμες ανάρτησης, όπως αυτή της υποβολής νέας ανακοίνωσης, προσθέτοντας μεταξύ άλλων σκιές και λευκό χρώμα φόντου. Οι επόμενοι κανόνες με αρχικό όνομα `.form` επεξεργάζονται την θέση, το κείμενο και τα πεδία εισόδου στις φόρμες. Για παράδειγμα, οι κανόνες `form-title` και `form-btn`, αλλάζουν οπτικά τον τίτλο στην κορυφή της φόρμας, και τοποθετούν τα κουμπιά σε οριζόντια αντί για κάθετη στοίχιση. Συνεχίζοντας, ο κανόνας `btn` αφορά τα κουμπιά υποβολής απάντησης, ανάρτησης αρχείου και αποστολής email, καθώς και τα κουμπιά σύνδεσης, αποσύνδεσης και εγγραφής, ενώ ο κανόνας `btn-danger` αναφέρεται στα κουμπιά διαγραφής. Ενδεικτικές τροποποιήσεις είναι η προσθήκη πράσινου χρώματος στα κουμπιά με `btn class`, και κόκκινου χρώματος σε αυτά με `btn-danger class`. Ο κανόνας `btn-secondary` βελτιώνει τα κουμπιά επεξεργασίας ανάρτησης ή απάντησης, και αυτά που οδηγούν στην προηγούμενη ή την αρχική σελίδα. Το class `card` τοποθετείται σε `<div>` tags που περικλείουν την κάθε ανάρτηση, και με τον χειρισμό του αλλάζει η δομή και εμφάνιση των αναρτήσεων, με την προσθήκη λόγου χάρη σκιών και λευκού χρώματος φόντου. Επιπλέον, οι κανόνες `file-container` επεξεργάζονται τις υποβολές αρχείων, όπως τον περιορισμό του μεγέθους των εικόνων, και τέλος ο κανόνας `input-file` βελτιώνει οπτικά το πεδίο επιλογής αρχείου προσθέτοντας μεταξύ άλλων γκριζό φόντο.

## 3. ΠΑΡΑΡΤΗΜΑ

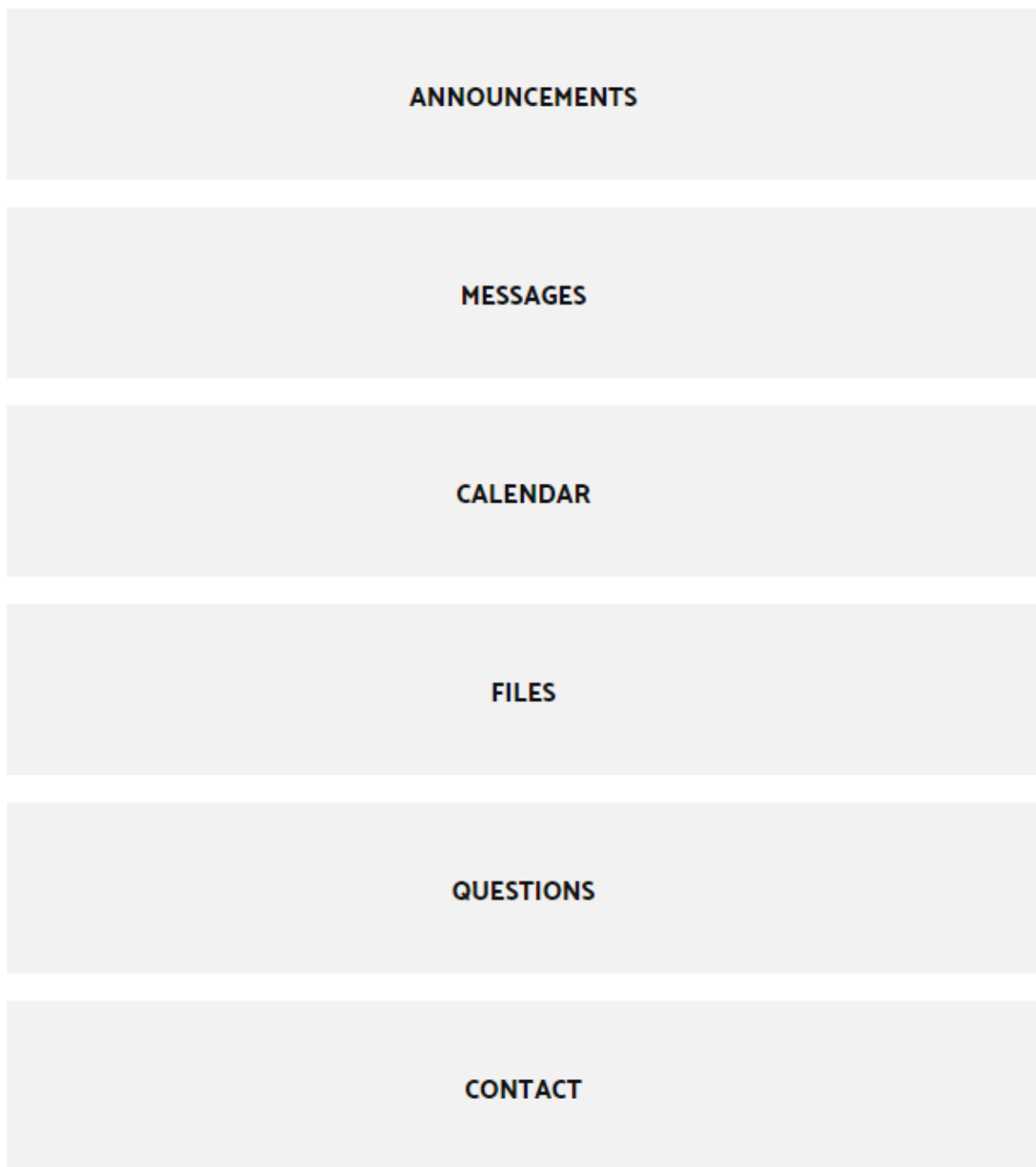
### 3.1 Εικόνες

Εικόνα 3.1: Η κορυφή της αρχικής σελίδας



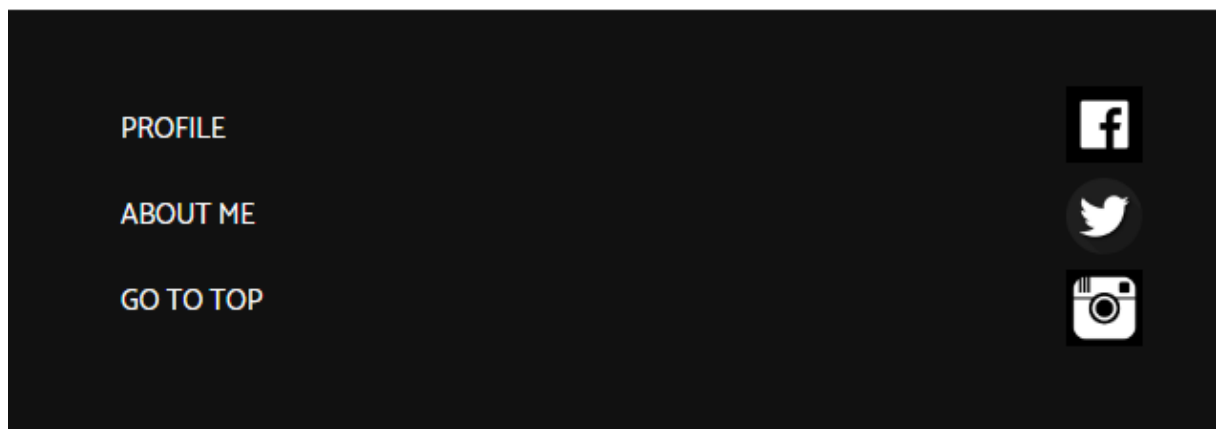
(3.1)

Εικόνα 3.2: Το κεντρικό τμήμα της αρχικής σελίδας



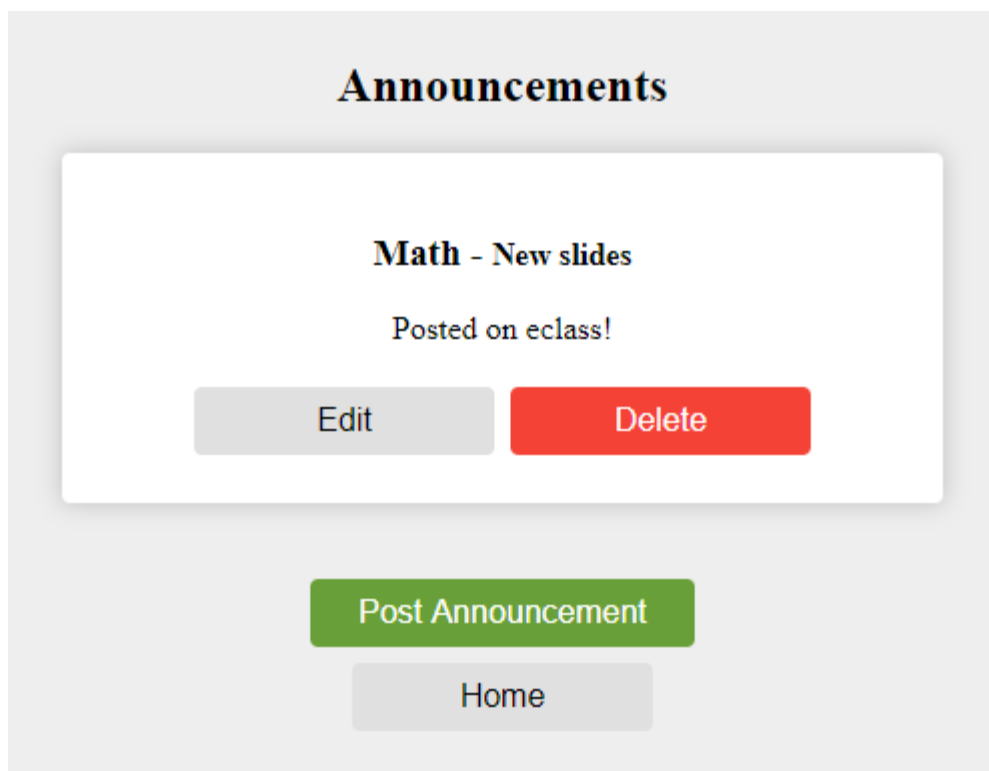
(3.2)

Εικόνα 3.3: Το κάτω τμήμα της αρχικής σελίδας



(3.3)

Εικόνα 3.4: Η σελίδα ανακοινώσεων



(3.4)

Εικόνα 3.5: Η σελίδα ομαδικής συζήτησης

**Messages**

**Name:** ianha

**Message:**

[Reply](#)

jimkar (S): Me too  
ianha (S): Fine mate  
jimkar (S): How are you doing  
ianha (S): Hello guys

[Home](#)

(3.5)

Εικόνα 3.6: Το ημερολόγιο, όπου οι χρήστες μπορούν να προσθέσουν events

## Calendar


< > today **July 2019** month week day

Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	1	2 12a Final Exam	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

(3.6)

Εικόνα 3.7: Η σελίδα προβολής αρχείων

## Hi ianha, upload your file!



shutterstock

10000 10 12250123  
www.shutterstock.com

Delete

[New Schedule.txt](#)

Delete

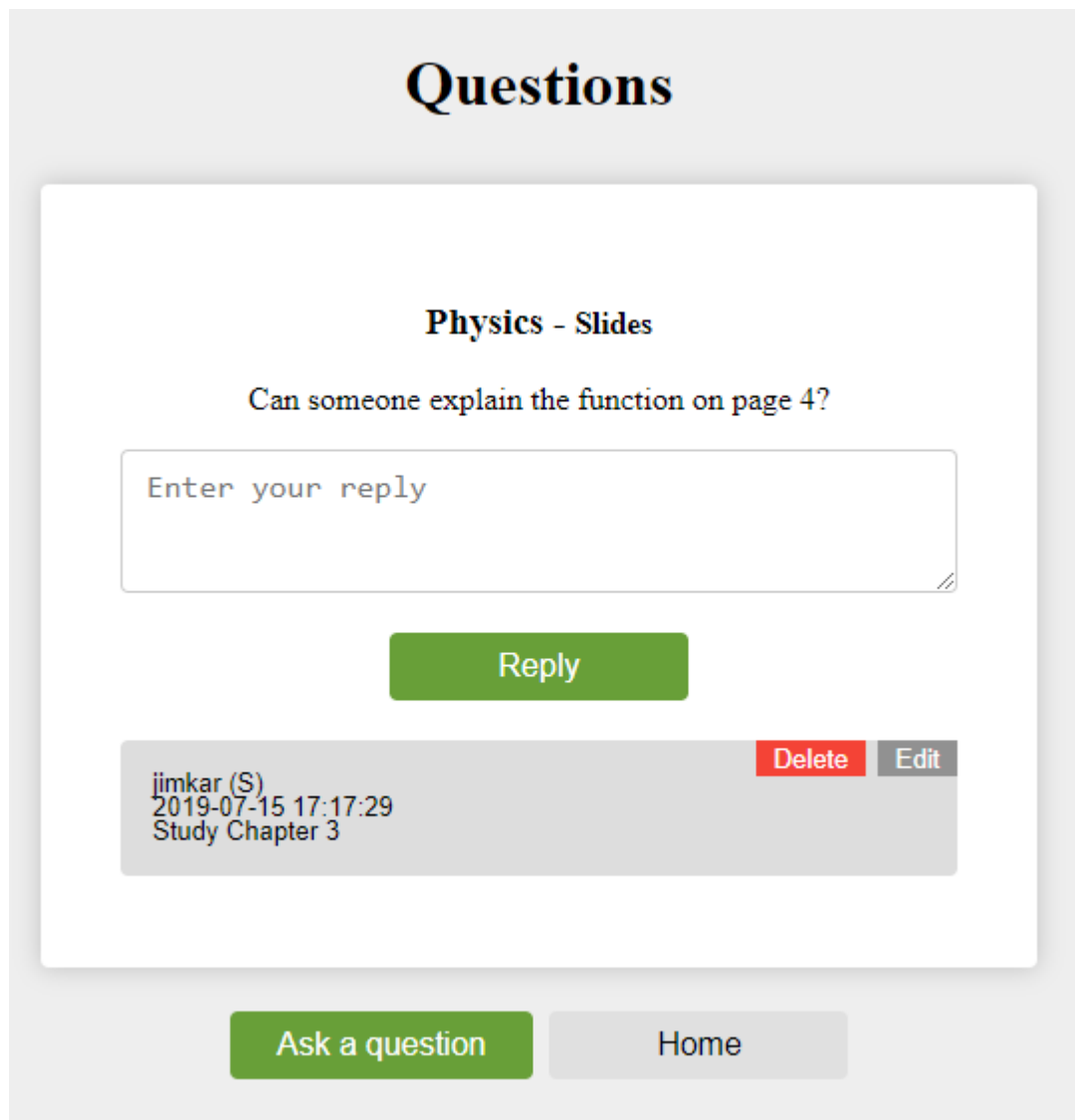
Choose File No file chosen

File description

Upload File

(3.7)

Εικόνα 3.8: Η σελίδα ερωτήσεων-απαντήσεων



(3.8)



## 4. ΣΥΜΠΕΡΑΣΜΑΤΑ

Το αντικείμενο της παρούσας εργασίας ήταν η δημιουργία μια διαδικτυακής εφαρμογής ονόματι Students Portal, η οποία θα απευθύνεται τόσο σε φοιτητές όσο και καθηγητές σχολών Πληροφορικής.

Στόχοι της ιστοσελίδας αυτής είναι τόσο η πλήρης και άμεση ενημέρωση των χρηστών σχετικά με το εκπαιδευτικό ίδρυμα στο οποίο ανήκουν, όσο και η υποβοήθηση της μεταξύ τους δικτύωσης και υποστήριξης.

Ο κύριος όγκος της εργασίας περιγράφηκε αρκούντως αναλυτικά στο Κεφάλαιο 2, όπου έγινε σφαιρική αναφορά για κάθε θεμέλιο τμήμα της εφαρμογής σε ξεχωριστές ενότητες. Πιο συγκεκριμένα, αναλύθηκαν η αρχική σελίδα, το σύστημα σύνδεσης κι εγγραφής, η προβολή προφίλ, η επικοινωνία, καθώς και οι βασικές λειτουργίες, που περιλαμβάνουν τις ανακοινώσεις, την κατάθεση αποριών, την ομαδική συζήτηση, την προβολή αρχείων και το δυναμικό ημερολόγιο. Σε κάθε ενότητα εξετάστηκε το αντίστοιχο τμήμα τόσο με βάση την δομή και την εμφάνιση του, όσο και από προγραμματιστικής πλευράς, με επεξήγηση των σημαντικότερων εντολών των γλωσσών προγραμματισμού που χρησιμοποιήθηκαν.

Εν κατακλείδι, κρίνεται πως η ιστοσελίδα Students Portal πετυχαίνει τον αρχικό σκοπό της.

Σε κάθε περίπτωση ωστόσο, όσο κι αν αυτή εξελιχθεί, πάντα θα υπάρχουν περιθώρια για περαιτέρω πρόοδο, επομένως αξίζει να διερευνηθεί η υλοποίηση τόσο πιθανών νέων λειτουργιών όσο και βελτιώσεων στην δομή κι εμφάνιση.

Ένα παράδειγμα θα συνιστούσε η προβολή περιεχομένου που αναλογεί μόνο στην σχολή στην οποία ανήκει ο χρήστης, έτσι ώστε εάν ένα μέλος επιλέξει για παράδειγμα το πανεπιστήμιο Πειραιώς, κάθε διαθέσιμη λειτουργία και κάθε εμφανιζόμενο μέλος να σχετίζονται αποκλειστικά με το συγκεκριμένο τμήμα.

## 5. ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Welling Luke, Thomson Laura. (2011). Ανάπτυξη Web Εφαρμογών με PHP και MySQL. Εκδόσεις Γκιούρδας Μ.

[2] Elmasri Ramez, Navathe B. Shamkant. (2016). Θεμελιώδεις αρχές συστημάτων βάσεων δεδομένων. Εκδόσεις Δίαυλος.

[3] Lemay Laura, Colburn Rafe. (2011). Πλήρες εγχειρίδιο της HTML και CSS. Εκδόσεις Γκιούρδας Μ.