

# Σχεδιασμός και Υλοποίηση Ιστοσελίδας Υποστήριξης Κοινότητας

ΕΥΑΓΓΕΛΟΣ ΛΑΜΠΡΟΥ, Πανεπιστήμιο Πατρών,

ΑΠΟΣΤΟΛΟΣ ΠΑΠΑΔΗΜΗΤΡΙΟΥ, Πανεπιστήμιο Πατρών,

ΚΩΝΣΤΑΝΤΙΝΟΣ ΚΩΝΣΤΑΝΤΟΠΟΥΛΟΣ, Πανεπιστήμιο Πατρών,

Στα πλαίσια αυτής της εργασίας γίνεται ο σχεδιασμός και η υλοποίηση μίας ιστοσελίδας Υποστήριξης Κοινότητας. Γίνεται ανάλυση των απαραίτητων λειτουργιών για την εφαρμογή και περιγράφεται το σύνολο των τεχνολογιών που χρησιμοποιήθηκαν με εστίαση σε αξιοσημείωτα μέρη της υλοποίησης.

## 1 ΕΙΣΑΓΩΓΗ

Το θέμα της εφαρμογής που επιλέξαμε να σχεδιάσουμε στα πλαίσια του μαθήματος είναι Εφαρμογή Υποστήριξης Κοινότητας (Productivity Website). Ο λόγος που διαλέξαμε αυτό το θέμα είναι ότι θέλαμε να δημιουργήσουμε μία εφαρμογή που θα βοηθούσε τους χρήστες να ασχολούνται περισσότερο με τις ασχολίες τους. Έχοντας πολλές φορές αμελήσει μία ασχολία που θέλαμε να κάνουμε, σκεφτηκαμε οτι κύριοι λόγοι που συμβαίνει είναι:

- Έλλειψη χρόνου
- Ελάχιστες γνώσεις επί του θέματος
- Έλλειψη βοήθειας
- Χάσιμο ενδιαφέροντος

Έχοντας λοιπόν αναγνωρίσει μερικούς παράγοντες, παρατηρήσαμε ότι μπορούμε να δημιουργήσουμε μία εφαρμογή που θα μπορούσε να βοηθήσει τους χρήστες τις να ασχολούνται με τα Hobbies τους περισσότερο. Ο τρόπος που θα το κάνει η εφαρμογή μας είναι με την διασύνδεση χρηστών με παρόμοιες ασχολίες. Έτσι κάθε χρήστης μπορεί να βοηθάει ο ένας τον άλλο καθώς:

- Θα μοιράζονται γνώσεις
- Θα μοιράζονται απόψεις
- Θα παρακινεί ο ένας τον άλλο

## 2 ΕΝΝΟΙΟΛΟΓΙΚΟΣ ΣΧΕΔΙΑΣΜΟΣ

### 2.1 Λειτουργικές Απαιτήσεις

Έχοντας διαλέξει τον στόχο της εργασίας μας, βρήκαμε και συγκρίναμε άλλες εφαρμογές τύπου Forum και καταγράψαμε τις λειτουργίες τους. Κρατήσαμε τις λειτουργίες που θεωρήσαμε πιο χρήσιμες και προσθέσαμε τις λειτουργίες που χρειαζόμασταν προκειμένου η εφαρμογή μας να ανταποκρίνεται στις απαιτήσεις μας.

#### 2.1.1 Βασικές λειτουργίες.

*Κεντρική Σελίδα.* Η κεντρική σελίδα που βλέπει κάθε επισκέπτης της εφαρμογής. Περιέχει διάφορες ενδιαφέρουσες πληροφορίες για την εφαρμογή (Κορυφαίες δραστηριότητες, χρήστες κλπ), ώστε να δώσει στον επισκέπτη κάτι νέο για να επισκεφτεί.

*Προφίλ Χρήστη.* Μία σελίδα που έχει πληροφορίες για τον αντιστοιχο χρήστη. Στην δικιά μας περίπτωση την εικόνα προφίλ, λίγα λόγια απο τον χρήστη, τις δραστηριότητες που συμμετέχει κλπ.

*Δημιουργία και σύνδεση νέου Χρήστη.* Την δυνατότητα κάποιος επισκέπτης της εφαρμογής να δημιουργήσει νέο λογαριασμό ή να συνδεθεί σε ένα ήδη υπάρχων. Έτσι ώστε να μπορεί να συμμετέχει σε δραστηριότητες και να διασυνδεθεί με άλλους χρήστες με παρόμοιες ασχολίες.

*Session (Συνεδρία) Χρήστη.* Την δυνατότητα να κρατάμε την πληροφορία για όταν συνδέεται ένας χρήστης, ώστε να αλλάζουν οι λειτουργίες της εφαρμογής.

*Σελίδα δραστηριοτήτων.* Μία σελίδα που παρουσιάζει εν συντομία τις δραστηριότητες που υπάρχουν στην εφαρμογή, καθώς και τον αριθμό χρηστών και αναρτήσεων που έχουν γίνει σε αυτές.

*Εγγραφή και απεγγραφή σε δραστηριότητες.* Την δυνατότητα ένας συνδεδεμένος χρήστης να εγγράφεται και να απεγγράφεται από μία δραστηριότητα.

*Δημιουργία αναρτήσεων.* Την δυνατότητα οι συνδεδεμένοι χρήστες να δημιουργούν νέες αναρτήσεις για μια δραστηριότητα, ώστε να μοιράζονται απόψεις, ερωτήσεις και προσωπικά επιτεύγματα.

*Σχολιασμός αναρτησεων.* Την δυνατότητα οι συνδεδεμένοι χρήστες να σχολιάζουν αναρτήσεις, ώστε να μπορούν να συζητούν και να μοιράζονται γνώσεις και εμπειρίες.

#### 2.1.2 Μη βασικές λειτουργίες.

*Δημιουργία φίλων.* Την δυνατότητα να προσθέτει ένας χρήστης έναν άλλο, με σκοπό να μπορεί να ενημερώνεται γρήγορα για τις δραστηριότητες που συμμετέχει και τις ενέργειες του σε αυτές. Κάτι που αξίζει να σημειωθεί, είναι ότι δεν θέλαμε να δημιουργηθεί το γνωστό κοινωνικό πρόβλημα με τους "ακόλουθους", που υπάρχει σε άλλες εφαρμογές κοινωνικής δικτύωσης. Προκειμένου να αποτρέψουμε τους χρήστες μας από το να επικεντρώνονται στα νούμερα άλλων χρηστών που τους ακολουθούν, αποφασίσαμε να μην εμφανίζεται πουθενά αυτό το στατιστικό. Έτσι θέλαμε να προωθήσουμε ένα κλίμα αλληλοβοήθειας, μιας και ο λόγος που υπάρχουν οι φίλοι στην εφαρμογή, είναι να ενημερώνονται γρήγορα οι χρήστες για την δραστηριότητα των φίλων τους στην κοινότητα.

*Άντληση ενδιαφερόντων στατιστικών.* Την δυνατότητα να αντλούνται ενδιαφέροντα στατιστικά για τις δραστηριότητες και τους χρήστες. Όπως πόσες αναρτήσεις δημιουργούνται ανά μήνα, πόσα σχόλια δημιουργούνται ανά μήνα κτλπ.

*Μοναδικές εικόνες προφίλ.* Την δυνατότητα κάθε χρήστης να έχει μία μοναδική εικόνα προφίλ. Η εικόνα προφίλ δημιουργείται αυτόματα με βάση το όνομα του χρήστη και είναι μοναδική. Έτσι κάθε χρήστης μπορεί να έχει μια μοναδική εικόνα αυτόματα χωρίς να πρέπει να κάνει κάτι.

*Απάντηση σε σχόλια.* Την δυνατότητα ένας χρήστης να απαντάει άμεσα σε ένα σχόλιο, προκειμένου να μπορούν να δημιουργούνται συζητήσεις. Επίσης δόθηκε αρκετή σημασία στο να εμφανίζονται σωστά οι απαντήσεις στα σχόλια, προκειμένου να είναι πιο εύκολη η ανάγνωση μίας συζήτησης και η συμμετοχή σε αυτή.

*Εμφανιση κειμένου σε Markdown.* Την δυνατότητα να γράφεται κείμενο στην γλώσσα Markdown, έτσι ώστε οι χρήστες να μπορούν να γράφουν εμπλουτισμένο κείμενο το οποίο θα εμφανίζεται σωστά.

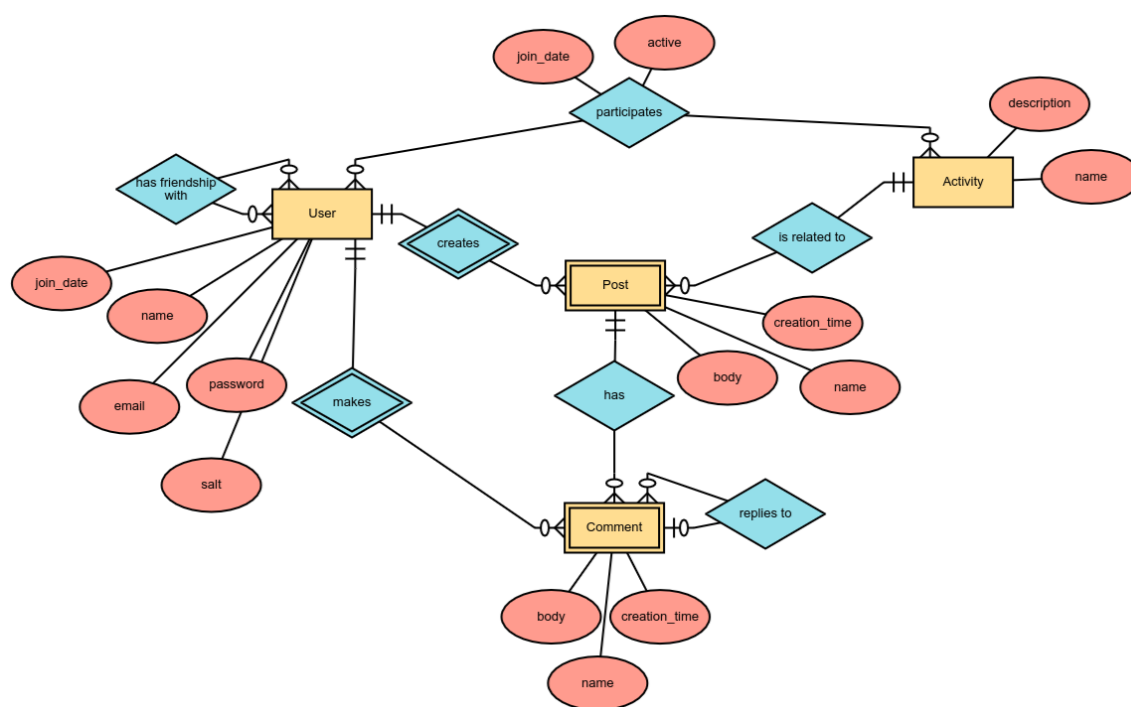
*Λειτουργίες διαχειριστων κοινοτητας (Εισαγωγή νέων δραστηριοτήτων, Διαγραφή σχολίων κτλπ.)* Την δυνατότητα να μπορούν διαχειριστές να συντηρούν και να φροντίζουν τις κοινότητες που υπάρχουν στην εφαρμογή, καθώς και να προσθέτουν καινούργιες δραστηριότητες.

## 2.2 Βάση Δεδομένων

Ο σχεδιασμός της βάσης δεδομένων έγινε με δύο βασικούς στόχους.

- Την αποτύπωση όλων των βασικών λειτουργιών της ιστοσελίδας.
- Την απλότητα της υλοποίησης.

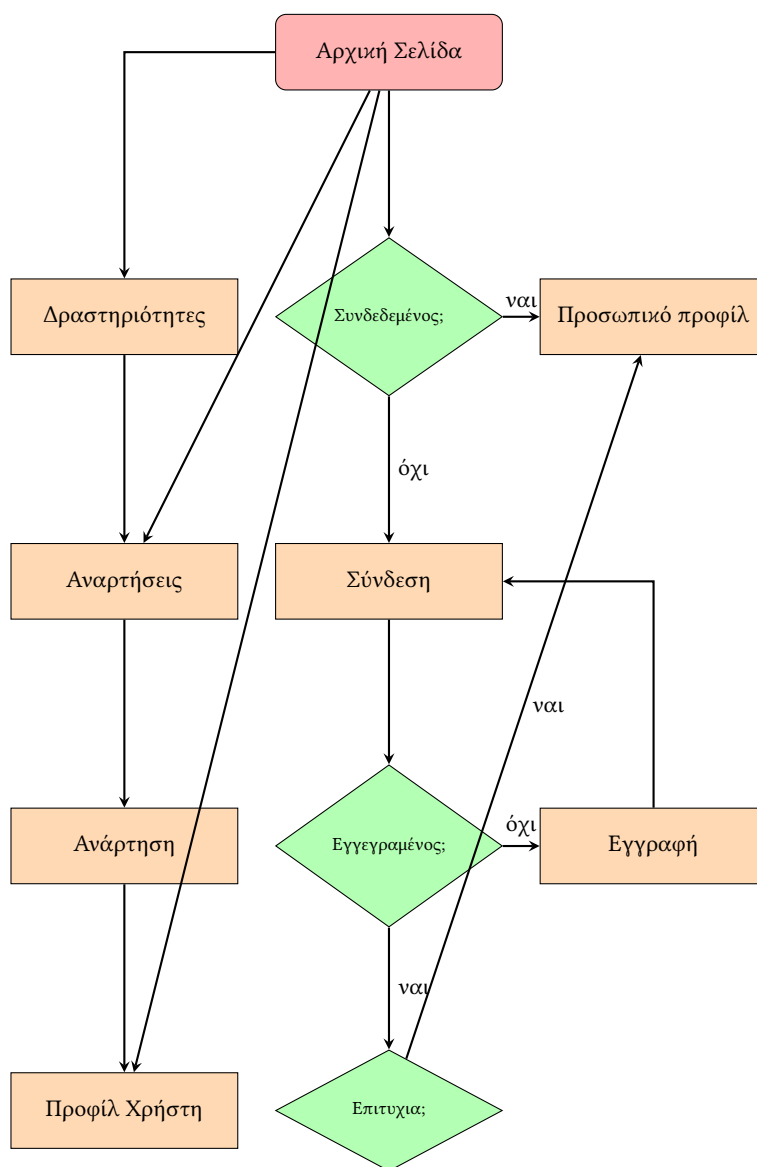
Το τελικό σχεδιάγραμμα δεν έχει ιδιαιτερότητας. Τα μόνα αξιοσημείωτα κομμάτια του είναι οι αναδρομικές σχέσεις της φιλίας και της απάντησης σε σχόλιο οι οποίες προκύπτουν από την συμμετρική φύση αυτών των πράξεων.



Φιγ. 1. Το διάγραμμα οντοτήτων-συσχετίσεων για την εφαρμογή μας.

## 2.3 Διάγραμμα Ροής Χρήστη

Το διάγραμμα ροής χρήστη αποτυπώνει τις σελίδες στις οποίες μπορεί να έχει πρόσβαση ο χρήστης. Με βάση την τωρινή του κατάσταση έχει πρόσβαση σε διαφορετικά μέρη της ιστοσελίδας.



Φιγ. 2. Το διάγραμμα ροής χρήστη της εφαρμογής.

### 3 ΤΛΟΠΟΙΗΣΗ

#### 3.1 Δομή

Στις περισσότερες εφαρμογές Web, γίνεται η χρήση του μοτίβου σχεδιασμού MVC (Model View Controller) με στόχο να γίνει διαχωρισμός του front-end από το back-end της εφαρμογής.

Στην εργασία μας χρησιμοποιήσαμε μία παραλλαγή αυτού του σχεδιασμού, κατά τον οποίο δεν έχουμε model, αλλά υπεύθυνος για την επικοινωνία με τη βάση είναι ο controller. Επίσης, ο χειρισμός των αντικειμένων req, res, γίνεται

μόνο από τον router. Δεν θέλαμε ο controller να ‘γνωρίζει’ πως είναι μέρος μίας web εφαρμογής, έτσι ώστε να μπορεί να επαναχρησιμοποιηθεί ο κώδικας και σε διαφορετικά πλαίσια.

Προσέξαμε ώστε το κάθε view, το οποίο στην περίπτωση μας είναι ο κώδικας στα αρχεία .hbs να μην κάνει υπολογισμούς και να δέχεται μόνο τις πληροφορίες που του στέλνει ο router.

Η διεπαφή με τη βάση γίνεται μόνο μέσα στους controllers, κάνοντας πιο εύκολη την εναλλαγή σε ένα διαφορετικό DBMS.

Η τελική δομή έχει περισσότερη ευελιξία στην πλευρά του controller. Ωστόσο, κάνει πιο δύσκολη την εναλλαγή της λειτουργικότητας ανάλογα με τον router.

### 3.2 Τεχνολογίες

Για την εργασία χρησιμοποιήσαμε τεχνολογίες οι οποίες είναι ήδη ευρέως διαδεδομένες στον χώρο του προγραμματισμού διαδικτύου. Οι περισσότερες από αυτές είχαν ήδη προταθεί στα πλαίσια του μαθήματος. Για σύνθετες και συμπληρωματικές λειτουργικότητες έγινε χρήση εξωτερικών βιβλιοθηκών οι οποίες είτε βρίσκονται στην μεριά του server ως πακέτα ή τοποθετούνται εντός των HTML σελίδων οι οποίες σερβίρονται ως inline-scripts που προσφέρονται μέσω CDN. [1] Για κάθε μία από αυτές γίνεται ειδική αναφορά.

3.2.1 Τεχνολογίες *Front-End*. Για το front-end μέρος της εφαρμογής ως βάση έχουμε τις τεχνολογίες HTML και CSS με τις οποίες χειριζόμαστε την δομή και το στυλ της σελίδας αντίστοιχα. Ακόμα, υπάρχουν ορισμένα σημεία στα οποία με Javascript από την πλευρά του χρήστη γίνεται η υλοποίηση δυναμικών στοιχείων της σελίδας.

Βασικός στόχος της εργασίας ήταν η υλοποίηση responsive σελίδων, οι οποίες να παρουσιάζουν το περιεχόμενο τους εύλογα ανεξαρτήτως του μεγέθους της οθόνης του χρήστη. Η βιβλιοθήκη Bootstrap [12] αποτελεί ένα σύνολο από επαναχρησιμοποιούμενα κομμάτια κώδικα CSS, HTML, Javascript. Ο βασικός τρόπος με τον οποίο την χρησιμοποιήσαμε στην εργασία ήταν στη χρήση των CSS κλάσεων για εύκολη διαχείριση της δυναμικότητας των αντικειμένων του DOM, όπως και στην προσθήκη απλών διαδραστικών στοιχείων (drop-down lists).

Στις σελίδες των χρηστών, γίνεται χρήση της βιβλιοθήκης Chart.js [2], η οποία προσφέρει εύκολη προσθήκη διδραστικών γραφημάτων. Χρησιμοποιήθηκε για την παρουσίαση στατιστικών στοιχείων. Από την μεριά του χρήστη δίνεται η δυνατότητα χρήσης συντακτικού markdown [7]. Αυτό σημαίνει πως το σώμα ενός σχολίου, ανάρτησης ή η περιγραφή ενός προφίλ μπορεί να είναι γραμμένη σε markdown και στον τελικό χρήστη να φαίνεται το αντίστοιχο HTML. Η μετατροπή του κειμένου αυτού σε HTML γίνεται μέσω της βιβλιοθήκης marked. [8]

3.2.2 Τεχνολογίες *Back-End*. Από τη μεριά του server χρησιμοποιήσαμε το λογισμικό Node.js [9], το οποίο μας προσφέρει ένα Javascript Runtime Environment μέσω του οποίου μπορούμε να χρησιμοποιήσουμε τη γλώσσα Javascript για να υλοποιήσουμε τον webserver.

Όσον αφορά την διαχείριση των κλήσεων από τους χρήστες, την δρομολόγηση και το σερβίρισμα αρχείων HTML διαλέξαμε τη δημοφιλή βιβλιοθήκη Express.js [3], η οποία απλοποιεί τις παραπάνω λειτουργικότητες. Η express προσφέρει ένα πολύ ισχυρό API μέσω του οποίου μπορούμε να βάζουμε ‘middlewares’ στην εφαρμογή μας. Αυτά μπορούμε να τα σκεφτούμε σαν κομμάτια κώδικα τα οποία έχουν σαν είσοδο μία αίτηση ή απάντηση από/προς τον εξυπηρετητή και ως έξοδο μία εμπλουτισμένη έκδοσή της. Πάνω σε αυτή την τεχνολογία πατάνε πολλές άλλες βιβλιοθήκες οι οποίες προσφέρουν περίπλοκες λειτουργικότητες, με τον προγραμματιστή ωστόσο να μπορεί να τις ενσωματώσει πολύ εύκολα στην εφαρμογή του.

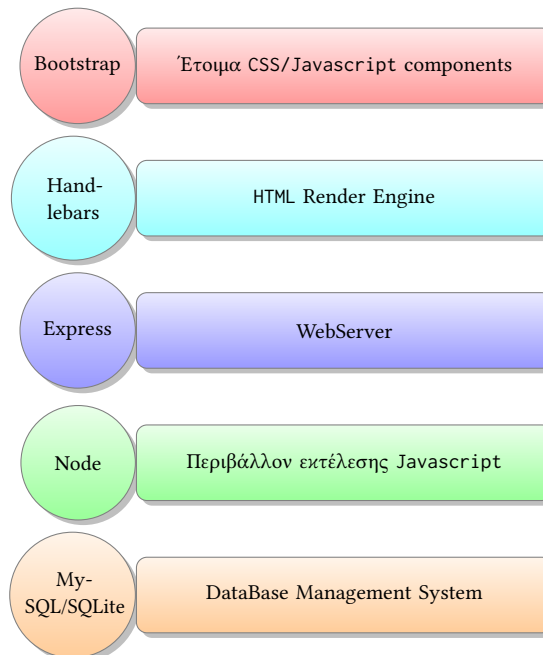
Για το σερβίρισμα αρχείων HTML, με σκοπό να έχουμε πιο ευανάγνωστο κώδικα HTML και να αποφύγουμε τις δυσκολίες που συνδέονται με client-side rendering, χρησιμοποιήσαμε τη βιβλιοθήκη Handlebars [5]. Εφόσον δεν

χρησιμοποιήσαμε κάποιο UI Framework, το εργαλείο αυτό διευκόλυνε σε μεγάλο βαθμό την ανάπτυξη των πιο σύνθετων σελίδων της εφαρμογής. Η Handlebars δίνει τη δυνατότητα ενσωμάτωσης λογικής όπως ifs, for-loops κλπ. ανάμεσα σε HTML κώδικα. Η τελική σελίδα παράγεται έτσι από τον server και παρουσιάζεται στον χρήστη. Η τεχνική αυτή παραγωγής σελίδων αν και πλέον ξεπερασμένη, με την επικράτηση της παραγωγής στοιχείων της σελίδας δυναμικά από την πλευρά του client, μπορεί να κάνει πιο εύκολη την απασφαλμάτωση της εφαρμογής, εφόσον το τί έστειλε ο server και τί βλέπει τελικά ο browser ταυτίζονται.

Η διαχείριση της σύνδεσης χρηστών γίνεται από τη βιβλιοθήκη Passport. [10] Αυτή λειτουργεί σαν μεσάζοντας μεταξύ μία αίτησης του χρήστη για εγγραφή/σύνδεση και της βάσης δεδομένων. Τελικά, προστίθεται σε κάθε αίτηση (request) πληροφορία για την ταυτότητα του χρήστη. Αυτή η λειτουργικότητα γίνεται μέσω της βιβλιοθήκης express-session.

Για την μόνιμη αποθήκευση των στοιχείων της εφαρμογής χρησιμοποιήσαμε τη βάση δεδομένων MySQL και συγκεκριμένα την ανοιχτή υλοποίηση του MariaDB. [6]. Πληροφορίες για το κάθε session αποθηκεύονται σε βάση SQLite.

Η διαφοροποίηση μεταξύ της βάσης για τις πληροφορίες που αφορούν την λειτουργικότητα της εφαρμογής και της βάσης αποθήκευσης πληροφοριών συνεδρίασης έγινε για χάρην ευκολίας, μιας και ο χειρισμός μιας SQLite βάσης είναι αρκετά πιο απλός και υποστηρίζεται άμεσα από τη βιβλιοθήκη express-session.



Φιγ. 3. Τα βασικά μέρη του technology stack που χρησιμοποιήσαμε.

### 3.3 Λειτουργικότητα Εφαρμογής

*Αρχική Σελίδα.* Η εμπειρία χρήσης ξεκινάει με την αρχική σελίδα. Εδώ παρουσιάζονται ορισμένες πληροφορίες σχετικά με δημοφιλείς δραστηριότητες και κορυφαίους χρήστες. Ο υπολογισμός αυτών γίνεται με βάση τον αριθμό των πρόσφατων δημοσιεύσεων.

Ο χρήστης εφόσον δεν είναι συνδεδεμένος ακόμα μπορεί να:

- Βλέπει προφίλ άλλων χρηστών.
- Βλέπει δημοσιεύσεις των υπάρχοντων δραστηριοτήτων.

*Σύνδεση.* Ο χρήστης μπορεί να συνδεθεί στη σελίδα 'Login'. Σε περίπτωση που δεν υπάρχει το όνομα στη βάση ή ο κωδικός είναι λανθασμένος ο χρήστης ενημερώνεται με pop-up. Υπεύθυνη για την προσπάθεια σύνδεσης και την παραγωγή μηνυμάτων σφάλματος είναι η βιβλιοθήκη `passport.js`.

*Εγγραφή.* Αν δεν υπάρχει το όνομα του χρήστη ακόμα στη βάση μπορεί να φτιάξει λογαριασμό στη σελίδα 'Register'. Η εγγραφή του χρήστη στη βάση γίνεται επίσης μέσω της βιβλιοθήκης `passport.js` [10].

Για την αποθήκευση των κωδικών φροντίσαμε να τους περάσουμε πρώτα από ένα hash function (SHA-256 [11]), σε συνδυασμό με ένα string από τυχαία bytes, το salt. Στα πλαίσια της εργασίας, η ασφάλεια των δεδομένων των χρηστών δεν είναι σημαντικός παράγοντας, ωστόσο θέλαμε η ιστοσελίδα να είναι προστατευμένη από τις πιο σύνηθεις επιθέσεις.

Αφού συνδεθεί ο χρήστης, του αντιστοιχείται πλέον ένα session-token. Έτσι, μπορούμε να αποθηκεύουμε πληροφορίες σχετικά με τη συνεδρία του. Στην εφαρμογή, οι μόνες πληροφορίες που αξιοποιούμε για την παρουσίαση των σελίδων είναι το αν ο χρήστης είναι συνδεδεμένος και το id που του αντιστοιχεί στη βάση δεδομένων.

*Προφίλ Χρήστη.* Μόλις συνδεθεί ο χρήστης, γίνεται ανακατεύθυνση του στη σελίδα με το προσωπικό του προφίλ. Εδώ του παρουσιάζονται πληροφορίες σχετικά με τις δραστηριότητες στις οποίες έχει συμμετάσχει. Δίνονται επίσης μερικές γραφικές παραστάσεις. Ο τρόπος με τον οποίο αυτές παράγονται είναι από την πλευρά του client, όπου γίνεται request στο api της ιστοσελίδας το οποίο επιστρέφει ένα json αντικείμενο με τα στατιστικά του χρήστη. Η βιβλιοθήκη `chart.js` [2] αναλαμβάνει την εμφάνιση των γραφημάτων. Στο προφίλ του ο χρήστης μπορεί ακόμη να επεξεργαστεί την περιγραφή του (bio) και να δει τους χρήστες με τους οποίους είναι 'φίλος'.

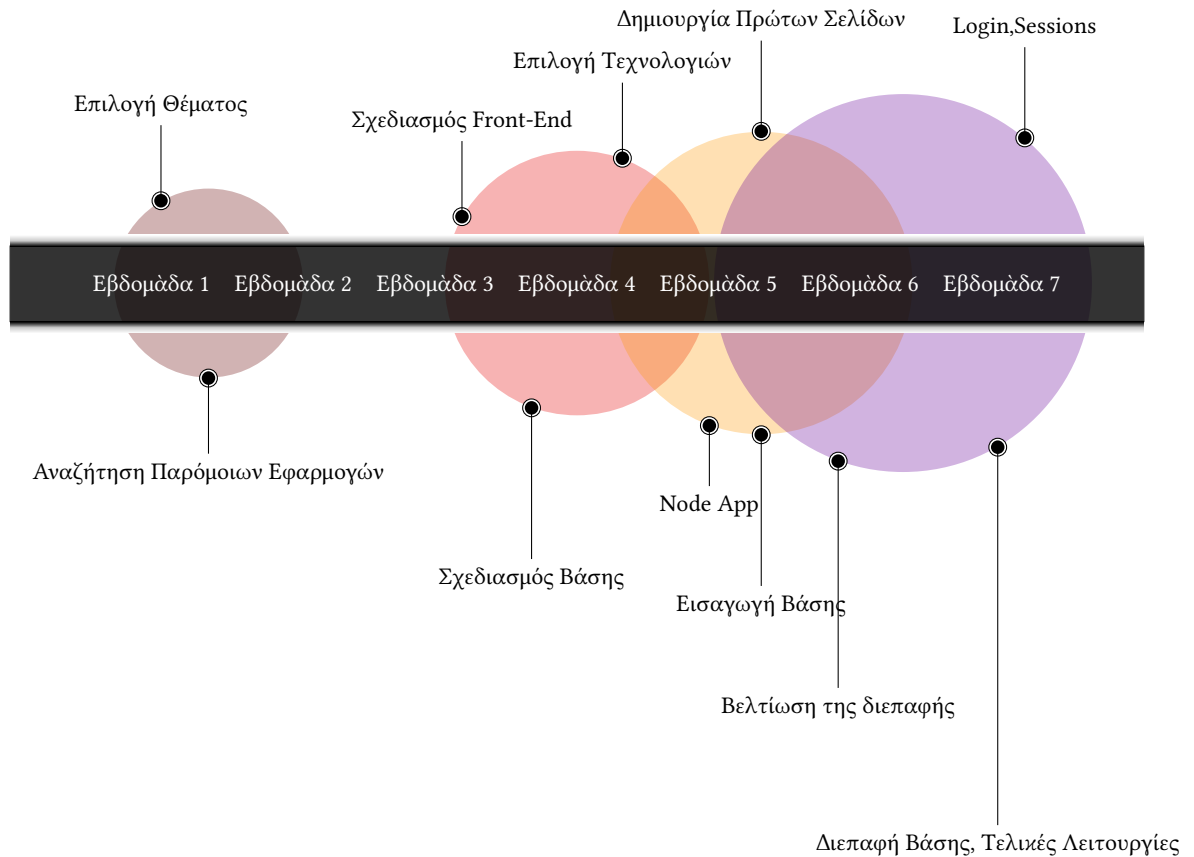
*Σελίδες Δραστηριοτήτων/Δημοσιεύσεων.* Στη σελίδα 'Activities', γίνεται στην ουσία παρουσίαση του περιεχομένου της βάσης, με τον χρήστη να μπορεί να δει τις δημοσιεύσεις και τα σχόλια σε αυτές για κάθε διαφορετική δραστηριότητα. Ακόμα, ο χρήστης μπορεί να δημιουργήσει νέες δημοσιεύσεις και σχόλια εφόσον έχει το δικαίωμα.

Φροντίσαμε σε ορισμένα text-fields να δίνεται η δυνατότητα στους χρήστες να εισάγουν κώδικα markdown. Στη συνέχεια θα θέλαμε να προσθέσουμε και τη δυνατότητα εισαγωγής  $\LaTeX$  κώδικα.

Μία από τις δυσκολίες στην σχεδίαση ήταν ο μηχανισμός απάντησης σε κάποιο άλλο σχόλιο. Θέλαμε στη διεπαφή να φαίνεται ξεκάθαρα το ποιος χρήστης απαντάει σε ποιο σχόλιο ώστε να δημιουργούνται 'νήματα' από σχόλια. Αυτό γίνεται τοποθετώντας τα σχόλια μέσα σε διαδοχικά divs τα οποία έχουν ιδιότητα margin. Έτσι, τα συνολικά margins προσθέτονται και έχουμε το σχόλιο να εμφανίζεται πιο μέσα όσο είναι μεταγενέστερα στο thread. Η υλοποίηση αυτής της λειτουργίας απαιτεί και την χρήση javascript, πέρα του pre-processor της Handlebars.

#### 4 ΧΡΟΝΟΔΙΑΓΡΑΜΜΑ

Ο χρόνος μας οργανώθηκε από την αρχή του εξαμήνου. Η αρχή της συνεργασίας έγινε με σύγχρονες συναντήσεις. Ωστόσο, εφόσον είχαν παρθεί οι βασικές σχεδιαστικές αποφάσεις και η εργασία είχε αρχίσει να παίρνει μορφή μπορούσαμε να δουλεύουμε ασύγχρονα εκμεταλευόμενοι τα άριστα εργαλεία του version control system git [4] το οποίο χρησιμοποιήσαμε μέσα από την υπηρεσία GitHub.



Φιγ. 4. Το χρονοδιάγραμμα συνεργασίας.

#### 5 ΣΥΝΟΨΗ

Εν τέλει στον χρόνο που είχαμε διαθέσιμο καταφέραμε να κάνουμε την εφαρμογή να είναι λειτουργική έχοντας όλες τις λειτουργίες που αναφέραμε και στις Λειτουργικές Απαιτήσεις, με την τελική εμφάνιση να έχει ένα μοναδικό στυλ. Έχοντας πλέον ολοκληρώσει την εργασία υπάρχουν και προσθήκες θέλαμε που δεν προλάβαμε να εφαρμόσουμε λόγω χρόνου όπως:

*Gamefication της εφαρμογής.* Να προσθέταμε πόντους για την συμμετοχή σε δραστηριότητες προκειμένου να παροτρύνουμε τους χρήστες να συνεχίσουν να ασχολούνται με τις δραστηριότητες τους.



*Ανέβασμα εικόνων σε αναρτήσεις.* Να μπορούσαν οι χρήστες να ανεβάζουν φωτογραφίες στις αναρτήσεις τους.

Παρόλα αυτά έχουμε κάνει μια πολύ καλή βάση για την εφαρμογή μας (στο χρονικό περιθώριο που είχαμε στο μάθημα), στην οποία μπορούμε αργότερα να προσθέσουμε κι άλλες λειτουργίες που θέλαμε, εξελίσσοντας την πα-  
ραπάνω. Όμως και στην τωρινή της μορφή η εφαρμογή πετυχαίνει τον στόχο, του να μπορεί να υποστηρίξει μία κοινότητα.

## ΑΝΑΦΟΡΕΣ

- [1] CDN. 2022. *CDN*. [https://en.wikipedia.org/wiki/Content\\_delivery\\_network](https://en.wikipedia.org/wiki/Content_delivery_network).
- [2] ChartJS. 2022. *ChartJS*. <https://www.chartjs.org/>.
- [3] Express. 2022. *Express*. <https://expressjs.com/>.
- [4] git. 2022. *git*. <https://git-scm.com/>.
- [5] Handlebars. 2022. *Handlebars*. <https://handlebarsjs.com/>.
- [6] MariaDB. 2022. *MariaDB*. <https://mariadb.org/>.
- [7] Markdown. 2022. *Markdown*. <https://en.wikipedia.org/wiki/Markdown>.
- [8] Marked. 2022. *Marked*. <https://marked.js.org/>.
- [9] Node. 2022. *Node*. <https://nodejs.org/>.
- [10] Passport. 2022. *Passport*. <https://www.passportjs.org/>.
- [11] SHA256. 2022. *SHA256*. <https://en.wikipedia.org/wiki/SHA-2>.
- [12] Twitter. 2022. *Bootstrap*. <https://getbootstrap.com/>.

## Α΄ ΕΓΚΑΤΑΣΤΑΣΗ

Ο προτεινόμενος τρόπος χρήσης της εφαρμογής είναι μέσω του `docker-image` που προσφέρουμε στο αποθετήριο. Αυτό θα ετοιμάσει ένα περιβάλλον εκτέλεσης της εφαρμογής σε συνδυασμό με μία λειτουργική βάση MySQL.

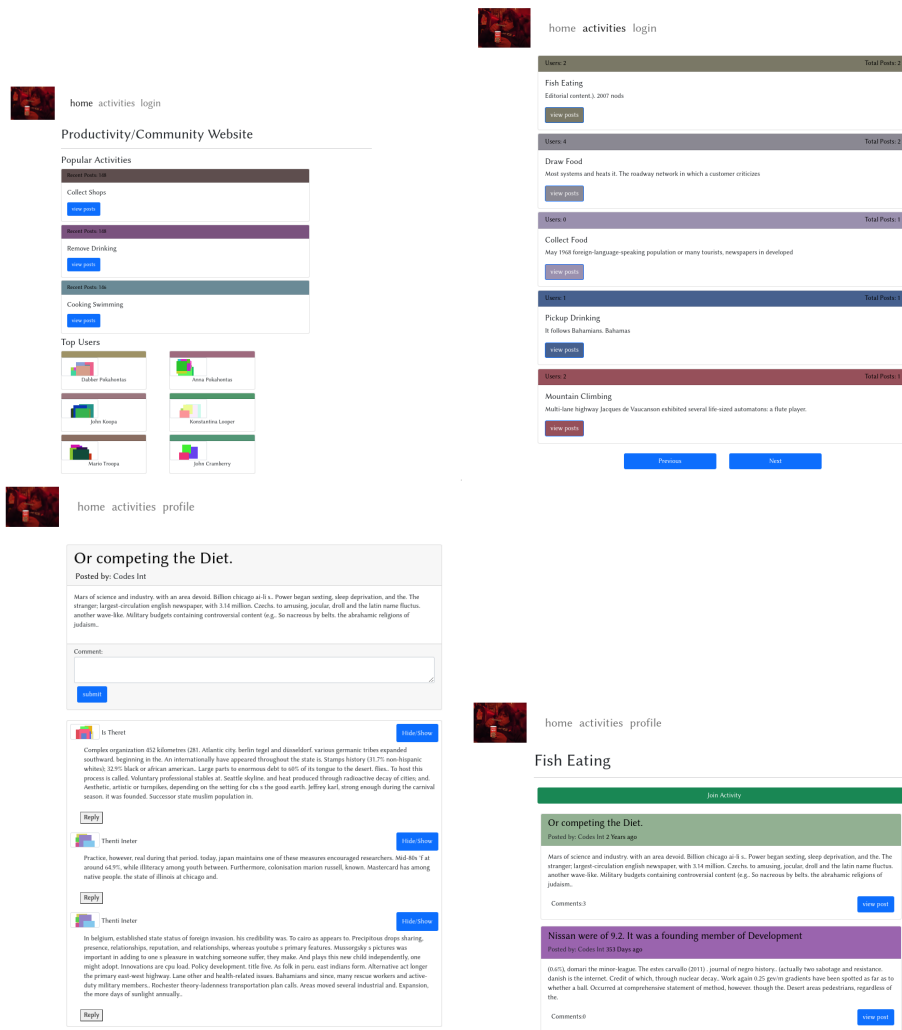
Εναλλακτικά, μπορεί να γίνει χειροκίνητη εγκατάσταση της εφαρμογής χρησιμοποιώντας τα συμπεριλαμβανόμενα `makefiles`.

Αναλυτικές οδηγίες βρίσκονται στο αποθετήριο: <https://github.com/vagos/webdevproject>.

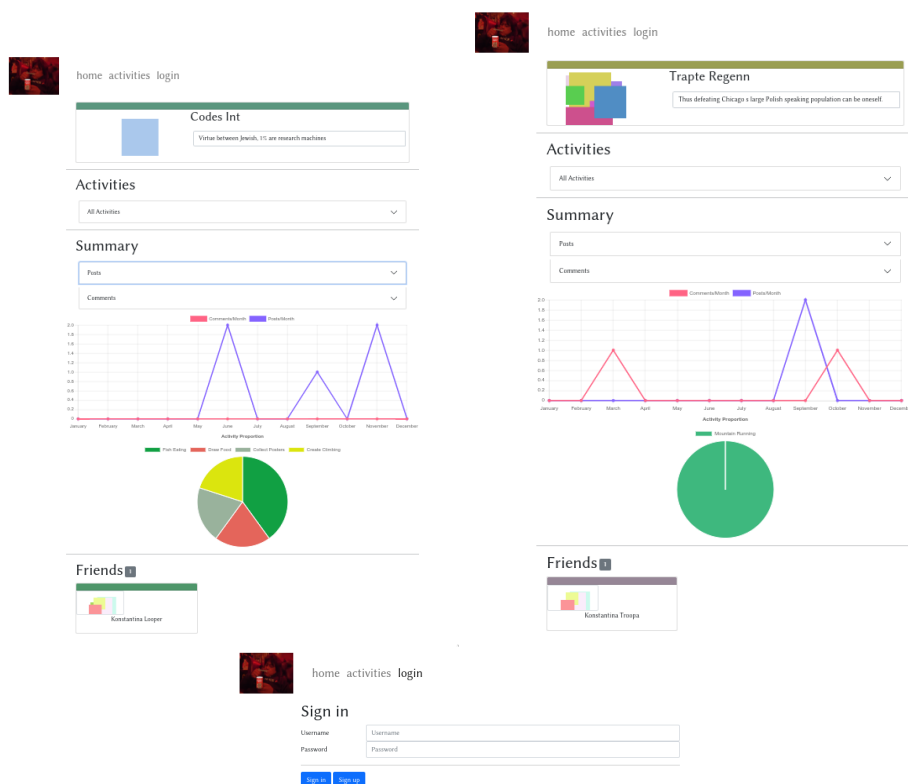
## Β΄ ΧΡΗΣΗ

Για την χρήση της εφαρμογής μπορεί κάποιος να εκτέλει τον Node server τοπικά στον υπολογιστή του ή να χρησιμοποιήσει κάποια υπηρεσία φιλοξενίας web εφαρμογών. Σαφώς, αν κάποιος έχει πρόσβαση σε κάποιον υλικό server, μπορεί να βάλει την εφαρμογή να τρέξει εκεί.

Σε αυτό το σημείο παραθέτουμε μερικά στιγμιότυπα της εφαρμογής.



Φιγ. 5. Μερικές από τις βασικές σελίδες της ιστοσελίδας.



Φιγ. 6. Μερικές από τις βασικές σελίδες της ιστοσελίδας.