

ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ ΠΑΓΚΟΣΜΙΟΥ ΙΣΤΟΥ

Β' ΜΕΡΟΣ ΕΡΓΑΣΙΑΣ

Βασίλειος Πιτσιάβας: 2859

Χριστίνα Μαναβή: 2767

Αθανάσιος Μπατσιούλας: 2776

Χαράλαμπος Παναγιωτόπουλος: 2784

Το θέμα της εργασίας είναι η προώθηση των ισχυρών θεσμικών οργάνων για την υπεράσπιση της Ειρήνης και της Δικαιοσύνης στον κόσμο, όπου αποτελεί τον δέκατο-έκτο στόχο των Ηνωμένων Εθνών. Μερικοί από τους πιο σημαντικούς επιμέρους στόχους του θέματος είναι η μείωση όλων των μορφών βίας και των σχετικών ποσοστών θανάτων που προέρχονται από αυτή. Επίσης ο τερματισμός της κακοποίησης, της εκμετάλλευσης, της εμπορίας και των βασανιστηρίων κατά των παιδιών. Τέλος η προώθηση του κράτους δικαίου σε εθνικό και διεθνές επίπεδο και διασφάλιση ίσης πρόσβασης στην δικαιοσύνη για όλους.

Γενικά για την επίτευξη του στόχου αυτού έχουμε υλοποιήσει ένα σάιτ τύπου Blog (Minimum Viable Product), με βασικά θέματα συζήτησης την υπεράσπιση της δικαιοσύνης, την παγκόσμια ειρήνη. Στο web app αυτό ο καθένας μπορεί να κάνει έναν λογαριασμό με τα προσωπικά του στοιχεία (name, surname, username, email, password) ή να συνδεθεί με τα στοιχεία του (username, password), εφόσον έχει ήδη λογαριασμό. Κατά την είσοδο του ο χρήστης εισέρχεται στο homepage, στο οποίο αρχικά διαβάζει συγκεκριμένα post-απόψεις άλλων χρηστών. Στην συνέχεια, του δίνεται η δυνατότητα να συντάξει το δικό του post-άποψη και να την κοινοποιήσει στο site. Τέλος εκτός από το homepage, ο κάθε χρήστης έχει και το δικό του profile, στο οποίο μπορεί να διαγράψει και να επεξεργαστεί τα προσωπικά του στοιχεία και τα post του. Επίσης αξίζει να αναφερθεί ότι υπάρχουν υποδομές για τον σχολιασμό και τα likes οποιουδήποτε άρθρου (έχει υλοποιηθεί το backend), καθώς και η επέκταση του blog σε κοινωνικό δίκτυο με την χρήση ηλεκτρονικών “φιλιών” (followers) μεταξύ των αρθρογράφων (έχει υλοποιηθεί στο backend). Επίσης, με την κατάλληλη διαμόρφωση το site μπορεί να υποστηρίξει indexing μεταξύ όλων των άρθρων αφού υπάρχει η δυνατότητα εύρεσης των most liked άρθρων.

Όσο για τους διαχειριστές (admin) της σελίδας υπάρχουν μερικές διαφορές. Αναλυτικά στο profile page του admin μπορεί να δει το συνολικό αριθμό των χρηστών και των post που έχουν δημοσιεύσει στη σελίδα, καθώς επίσης μπορεί να διαγράψει οποιονδήποτε χρήστη ή post επιθυμεί.

Η υλοποίηση μας απαρτίζεται από δύο κύρια components το frontend, όπου είναι γραμμένο σε HTML5, CSS3 και Javascript (jquery, es6) και το backend, όπου είναι γραμμένο σε node.js (με την χρήση του express server framework) και τα δεδομένα της βάσης αποθηκεύονται στην mysql. Αρχικά υλοποιήσαμε ένα RESTful API όπου ανάλογα με τα διάφορα endpoints URL (user, users, post, comment, session, friendship) και HTTP methods (Get, Post, Put, Delete) επικοινωνεί ο client με τον server αναλόγως. Η αρχιτεκτονική που έχει υλοποιηθεί το backend είναι η Routers-Controllers-Models. Οι express.Routers (userRouters, usersRouters, sessionRouters, postRouter, friendshipRouter, commentRouter). Ο Router που θα χρησιμοποιηθεί εξαρτάται από το endpoint του API και ανάλογα με το HTTP method θα καλέσει την αντίστοιχη συνάρτηση

από τον αντίστοιχο Controller. Επίσης υπάρχουν, όπως και με τους Routers, οι αντίστοιχοι Controllers για κάθε resource (usersController, userController, sessionController, postController, friendshipController, commentController).

https://app.swaggerhub.com/apis/mpatsiou7/pspi_project2020/1.0.0 Ο καθένας από αυτούς τους Controllers ελέγχουν αν τα δεδομένα από το body του API είναι σωστά ή όχι, αν είναι τότε καλούν από την κατάλληλη συνάρτηση από το αντίστοιχο model που χρειάζεται. Τέλος το τελευταίο επίπεδο της αρχιτεκτονικής μας είναι τα models (auth, comment, friendship, post, user, db) , το οποία κατάλληλες συναρτήσεις οι οποίες επικοινωνούν απευθείας με την βάση δεδομένων. Δηλαδή τα models εξυπηρετούν τα αντίστοιχα Controllers εκτός από το db.js το οποίο φτιάχνει ένα client ενός knex object , το knex είναι ένα εύκολος και ασφαλής query builder που βρήκαμε στο npm. Αναλυτικότερα τα models comment, friendship, post και user έχουν όλες τις συναρτήσεις για τα αντίστοιχα Controllers ανάλογα το HTTP method. Το auth.js έχει τις συναρτήσεις authenticate η οποία ελέγχει αν υπάρχει αυτός ο χρήστης με το username και password που δίνει το API, η createSession η οποία δημιουργεί ένα session id για τον κάθε χρήστη με τυχαιότητα, καθώς επίσης περιέχει και τις isLoggedIn και isLoggedInAsAdmin , οποίες χρησιμοποιούνται σαν middleware και ελέγχουν αν ο χρήστης είναι συνδεδεμένος , προτού ο κάθε Router δρομολογήσει τον αντίστοιχο Controller.

Τέλος το σχήμα της βάσης μας (schema.sql) αποτελείται από τα εξής tables (users, posts, comments, friendships, sessions). Το table users το primary key id και το email, name, role (normal, admin) , surname, username, password. Το table posts περιέχει το primary key id και τα χαρακτηριστικά user_id (foreign key), content, likes. Το table comments περιέχει το id (primary key), post_id (foreign key) , content. Το table friendships περιέχει το user_id και το post_id. Τέλος το table sessions περιέχει το user_id και το session_id.

Σχετικά με την μελλοντική επέκταση το app όπως αναφέραμε και παραπάνω θα υποστηρίξει “φιλίες” μεταξύ των χρηστών, καθώς επίσης θα μπορούν να αλληλεπιδρούν μεταξύ τους στο κάθε άρθρο ή να στέλνουν προσωπικά μηνύματα. Με την δημιουργία αυτού του κοινωνικού δικτύου στον κάθε χρήστη θα εμφανίζονται στο timeline τα άρθρα των χρηστών που αυτός επιθυμεί (με την χρήση φιλιών), καθώς θα του εμφανίζονται και διάφορες προτάσεις για διάφορα άρθρα σχετικά με τον στόχο του app από άλλα διάφορα δημοφιλή blogs, sites (π.χ. Medium, UN κλπ).

Για να το στήσουμε χρειάζεται να είναι προεγκατεστημένη η nodejs καθώς και κάποιος mysql-server. Ακολουθούμε τα εξής βήματα:

1. Δημιουργούμε μια βάση δεδομένων με οποιοδήποτε όνομα θέλουμε
2. Τρέχουμε το ./schema.sql στην καινούργια βάση (mysql -u my_user -p my_database < ./schema.sql)
3. Αλλάζουμε τα στοιχεία της βάσης (database, username, password) στο backend/models/db.js
4. Κάνουμε `npm install` τα libraries
5. Μέσα στο /backend τρέχουμε `npm start`