ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικόν και Καποδιστριακόν
Πανεπιστήμιον Αθηνών
ΙΔΡΥΘΕΝ ΤΟ 1837

Υποχρεωτική Εργασία - Τεχνολογίες Εφαρμογών Διαδικτύου

Εαρινό Εξάμηνο 2022

Φακορέλλης Ευάγγελος - 1115201900203
Φουκανέλης Χρήστος - Γεώργιος - 1115201900204
13 Σεπτεμβρίου 2022

Περιεχόμενα

Εισαγωγή

Η εργασία υλοποιήθηκε με βάση τις υποδείξεις της εκφώνησης αλλά και του καθηγητή. Στα παρακάτω κεφάλαια θα επεξηγηθούν οι σχεδιαστικές επιλογές και παραδοχές που έγιναν κατα την υλοποίηση της εργασίας. Το νωτιαίο άκρο της εφαρμογής έχει υλοποιηθεί με python fast api ενώ το μετωπειαίο άκρο της εφαρμογής έχει υλοποιηθεί με react.js. Η βάση δεδομένων χρησιμοποιεί ως dbms την PostgreSQL. Εν συντομία, η εργασία αφορά ένα ολοκληρωμένο σύστημα δημοπρασιών, όπου είναι διακριτοί 4 ρόλοι, του διαχειριστή συστήματος, του επισκέπτη, του προσφοροδότη και του πωλητή. Στα κεφάλαια που ακολουθούν θα αναλυθούν οι σχεδιαστικές επιλογές και αρχιτεκτονικές που έχουν υλοποιηθεί στα πλαίσια τόσο του front-end όσο και του backend, καθώς και το deployment της εφαρμογής χρησιμοποιώντας σύγχρονες μεθόδους.

Δομή-Αρχιτεκτονική συνολικός σχεδιασμός

Η εφαρμογή έχει σχεδιαστεί χρησιμοποιώντας headless λογική, όπως απαιτείται από την εκφώνηση. Το backend της εφαρμογής, το οποίο υλοποιήσαμε χρησιμοποιώντας το fast api framework της python είναι χωρισμένο σε κατάλληλες θεματικές ενότητες που απεικονίζουν τα αντικείμενα της εκφώνησης. Για την αυθεντικοποίηση των χρηστών χρησιμοποιούμε OAuth2 μηχανισμό και το backend παράγει ένα bearer token το οποίο αποθηκεύει με την σειρά του, το front end στο local storage, αφού κάνει πρώτα το κατάλληλο call στο σχετικό endpoint που αφορά την δημιουργία του token και την αυθεντικοποίηση. Από την άλλη μεριά το front-end στηρίζεται στο γνωστό React Template Argon Dashboard, το οποίο χρησιμοποιεί bootstrap. Αφού πραγματοποίησει τα κατάλληλα calls στα endpoints του api, φροντίζει να σερβίρει τα κατάλληλα routes αποκρύπτοντας ή εμφανίζοντας πληροφορίες ανάλογα με τα δικαιώματα του εκάστοτε χρήστη. Όπως αναφέρθηκε παραπάνω, η σχεσιακή βάση που χρησιμοποιούμε είναι η Postgres, όπου περισσότερες πληροφορίες θα αναλυθούν στην σχετική θεματική ενότητα παρακάτω. Τέλος, για το deployment της εφαρμογής, έχει χρησιμοποιηθεί docker, το οποίο επίσης θα αναλυθεί παρακάτω.

Database

Το DBMS το οποίο χρησιμοποιούμε στην εφαρμογή είναι η Postgres. Η επιλογή της Postgres, έγινε μετά από σχετική μελέτη, καθώς από ότι φάνηκε είναι πολύ σταθερή και διαθέτει κοινότητα που την υποστηρίζει πάνω από 20 χρόνια, είναι πολύ γρήγορη και αποδοτική. Επίσης, σύμφωνα με την πηγή https://www.enterprisedb.com/blog/4-reasons-postgresql-was-named-database-management-system-year-2020

«EDB is pleased to note that PostgreSQL has been named Database Management System of the Year 2020 by DB-Engines! PostgreSQL is the only database to have received this prestigious award three times based on PostgreSQL worldwide popularity.»

Back end

Το backend της εφαρμογής, το οποίο υλοποιήσαμε χρησιμοποιώντας το fast api framework της python, έχει δομημένα routes τα οποία μοντελοποιούν το business logic της εκφώνησης και το διαχωρίζουν σε κατάλληλα τμήματα-οντότητες. Κάθε router είναι υπεύθυνο να διαχειρίζεται τα Create Read Update Delete operations της εκάστοτε οντότητας. Τα routers που δημιουργήθηκαν είναι τα εξής: users, bids, auctions. Για την επικοινωνία με την βάση δεδομένων χρησιμοποιούμε το ORM της SqlAlchemy με το οποίο έχουν γίνει mapped τα tables της σχεσιακής βάσης δεδομένων με python classes. Στο πλαίσιο των crud operations και τα δεδομένα που ανταλλάσονται με το front end, χρησιμοποιούμε Pydantic Μοντέλα. Για την αυθεντικοποίηση χρήστη το backend διαθέτει κατάλληλο άκρο το οποίο παράγει jwt bearer token και το στέλνει πίσω στον χρήστη σε περίπτωση που έχει στείλει έγκυρα username και password

Front End

Το frontend της εφαρμογής, το οποίο υλοποιήσαμε χρησιμοποιώντας react.js. Σύμφωνα με την πηγή: https://www.freecodecamp.org/news/why-use-react-for-web-development/

React's popularity and usage are increasing day by day for good reason. As a developer, coding in React makes you better at JavaScript, a language that holds nearly 90% of the web development share today.

Η υλοποίηση είναι βασισμένη στο argon dashboard template, το οποίο έχει κάποιες έτοιμες υποδομές όπως sidebar, navbar και στηρίζεται στο bootstrap integration for react – Reactstrap το οποίο είναι πολύ αποτελεσματικό, καθώς πολλά κομμάτια css είναι ήδη υλοποιημένα. Επιπλέον, το template αυτό υποστηρίζει δυναμικά Routes, τα οποία βρίσκονται συγκεντρωμένα στο routes.js αρχείο. Έχει δημιουργηθεί ένα .env αρχείο με το link του api, ώστε να μην είναι «καρφωτό» μέσα στην εφαρμογή. Τέλος, η κεντρική ιδέα πίσω από την υλοποίηση των components είναι ότι πρώτα γίνονται τα api calls, όταν γίνεται resolve το promise αποθηκεύονται τα response data στο state του Component και τέλος γίνεται conditional rendering βάσει των δικαιωμάτων του συνδεδεμένου χρήστη.

Deployment

Το deployment της εργασίας γίνεται με docker. Υπάρχουν δύο κεντρικοί φάκελοι στην εργασία, το “**ergasia-api**” και το “**ergasia-front**”, όπου περιέχουν το backend και το frontend αντίστοιχα. Κάθε φάκελος από τους 2 περιέχει ένα dockerfile, το οποίο τελικά ενορχηστρώνει το κεντρικό docker-compose.yml που βρίσκεται εκτός των 2 φακέλων. Στο docker-compose συμπεριλαμβάνεται και postgres server. Η εντολή για να τρέξει το project είναι η εξής:

docker-compose up –d

και προυποθέτει την εγκατάσταση docker και docker compose στο μηχάνημα.

για να τρέξει η εφαρμογή επισκευθείτε το σύνδεσμο https://localhost

ο κωδικός του admin που παρέχεται by default απο την εγκατάσταση είναι 1234 (username: admin)

Σημειώσεις – Παρατηρήσεις - Επίλογος

Όλα τα ερωτήματα έχουν υλοποιηθεί πλήρως και η εφαρμογή υλοποιεί όλα τα ζητούμενα τους εκτός από το ερώτημα 10 το οποίο δεν υλοποιήθηκε αλλά και το ερώτημα 12 το οποίο έχει υλοποιηθεί αλλά δεν είναι σε stable μορφή ώστε να τρέξει ο κώδικάς του. Επιπλέον το bonus έχει υλοποιηθεί για mock δεδομένα τα οποία έχουν εξαχθεί με τυχαίο τρόπο από το dataset του eclass. Ο αλγόριθμος είναι πλήρως λειτουργικός και δίνει προβλέψεις. Δεν έχει ενταχθεί βέβαια μέσα στον κώδικα της εφαρμογής.

Γενικά ήταν μια απαιτητική εκτενής εργασία η οποία εξέταζε όλα τα πεδία του web development μέσω της οποίας αποκτήθηκε αρκετή τριβή με τις σχετικές έννοιες και frameworks. Μερικά ερωτήματα δεν υλοποιήθηκαν πλήρως λόγω έλλειψης χρόνου καθώς υπήρχαν υποχρεώσεις εργασιών / εξετάσεων και σε άλλα μαθήματα.