

Τεχνολογίες Εφαρμογών Διαδικτύου

Εργασία

Υλοποίηση ιστοχώρου

Μέλη Ομάδας:

Παπαθυμιόπουλος Θωμάς

1115201700119

Σιώτης Κωνσταντίνος

1115201700140



Homies

Περιεχόμενα

Περιεχόμενα	2
Εισαγωγή	3
Κεφάλαιο 1: Back end	4
Κεφάλαιο 2: Front end	7
Κεφάλαιο 3: Bonus	9
Επίλογος	11
Απαιτήσεις	11
Εκτέλεση	11
Δυσκολίες	11
Παραδοχές	12

Εισαγωγή

Στόχος της εργασίας είναι η ανάπτυξη ενός ιστοχώρου ενοικίασης κατοικιών και δωματίων, προσβάσιμη από φυλλομετρητές, καθώς και η ανάπτυξη της βάσης που θα υποστηρίξει τη λειτουργία του. Έχουν υλοποιηθεί όλα τα ερωτήματα της εκφώνησης της εργασίας του εαρινού εξαμήνου και το bonus ερώτημα.

Ακολουθούν τα κεφάλαια που αναλύουν τις σχεδιαστικές επιλογές για το frontend και το backend αντίστοιχα. Στο τρίτο κεφάλαιο αναφέρεται ο τρόπος υλοποίησης του ερωτήματος bonus. Στον επίλογο αναφέρονται κάποιες δυσκολίες που προέκυψαν και παραδοχές που επιλέχθηκαν.

Κεφάλαιο 1: **Back end**

Το back end κομμάτι της εργασίας βρίσκεται στο αρχείο **app.js** στον φάκελο src. Η υλοποίηση έχει γίνει σε Apache Server με MySQL βάση δεδομένων και node.js. Η διαχείριση των αιτημάτων γίνεται με την βιβλιοθήκη [express](#) και το [cors](#) middleware. Τα ερωτήματα προς την βάση SQL γίνονται μέσω της βιβλιοθήκης [mysql2](#).

Για το authentication του αποστολέα του αιτήματος έχει χρησιμοποιηθεί η βιβλιοθήκη [jsonwebtoken](#) με κρυφό κλειδί το "shhhhh". Για αυξημένη ασφάλεια του χρήστη, ο κωδικός πρόσβασής του φυλάσσεται κρυπτογραφημένος εντός της βάσης χρησιμοποιώντας την βιβλιοθήκη [bcryptjs](#).

Στην υλοποίηση του api, όπου χρειάζεται ανέβασμα εικόνας, έχει χρησιμοποιηθεί το middleware [multer](#) και η βιβλιοθήκη της node.js fs για την αποθήκευση και διαγραφή των εικόνων.

Τα endpoints του back end έχουν δομηθεί κατά την φιλοσοφία του REST API όπως ζητήθηκε από την εκφώνηση.

Σύνοψη endpoints:

- **GET /users:** γενική αναζήτηση στον πίνακα users της βάσης δεδομένων, επιστρέφει τις πληροφορίες όλων των χρηστών που πληρούν τα κριτήρια που θα δοθούν, εκτός του password. Τα parameters μπορεί να είναι username, email, role, id.
- **GET /users/:username:** αναζήτηση στον πίνακα users της βάσης δεδομένων, επιστρέφει τις πληροφορίες του χρήστη με το δοθέν username
- **POST /users:** προσθήκη χρήστη στον πίνακα users της βάσης, απαιτεί στο body τα εξής: username, email, name, surname, password, telephone, role και προαιρετικά μια εικόνα (χρησιμοποιεί [multer](#)).
- **POST /login:** σύνδεση υπάρχοντος χρήστη, ελέγχει αν τα username και password του body αντιστοιχούν σε χρήστη του πίνακα users.

- **PUT /users/:username:** επεξεργασία των στοιχείων του υπάρχοντος user με το username του url, θέτει τα στοιχεία του χρήστη βάσει του body που μπορεί να περιέχει username, email, name, surname, telephone, role και filename για την εικόνα (χρησιμοποιεί [multer](#)).
- **GET /messages:** επιστρέφει όλα τα μηνύματα από τον πίνακα messages της βάσης, απαιτεί να δοθεί το from ή το to που αναπαριστούν τα ids των χρηστών από τον πίνακα users, και ο χρήστης που έκανε το αίτημα να είναι ένας από τους δύο. Επιπλέον, μπορεί να δοθεί και page και perpage (και τα δύο) για την υλοποίηση του pagination.
- **POST /messages:** δημιουργεί ένα καινούργιο μήνυμα στον πίνακα messages με αποστολέα τον συνδεδεμένο χρήστη, απαιτείται στο body: to (το id του παραλήπτη), subject, message.
- **GET /:** επιστρέφει όλα τα δεδομένα της βάσης, αν ο συνδεδεμένος χρήστης είναι admin.
- **GET /accommodations:** επιστρέφει τα id των accommodations που πληρούν τα κριτήρια δοσμένα στο query. Το query πρέπει να περιέχει τα κριτήρια north, west, south, east, date1, date2, persons, wifi, freezer, heating, kitchen, tv, parking, lift, livingRoom όταν καλείται από τις φόρμες αναζήτησης. Το query πρέπει να περιέχει το κριτήριο host όταν καλείται από την φόρμα χώρων ενός οικοδεσπότη.
- **GET /accommodations/:id:** επιστρέφει όλες τις πληροφορίες της βάσης για το συγκεκριμένο accommodation id.
- **POST /accommodations/:id/reviews:** εισάγει στη βάση κριτική για το συγκεκριμένο accommodation id. Στο body περιέχει τις παραμέτρους UserId, Rating και Text.

- **POST /accommodations/:id/reservations:** εισάγει στη βάση κράτηση για το συγκεκριμένο accommodation id. Στο body περιέχει τις παραμέτρους UserId, Date1, Date2, Price.
- **POST /accommodations:** εισάγει στην βάση ένα νέο accommodation. Οι παράμετροι περνάνε σε μορφή string μέσω της παραμέτρου content του body (χρησιμοποιεί [multer](#)).
- **PUT /accommodations/:id:** ανανεώνει στην βάση τα πεδία για το accommodation με το συγκεκριμένο id. Οι παράμετροι περνάνε σε μορφή string μέσω της παραμέτρου content του body. (χρησιμοποιεί [multer](#))
- **POST users/:id/searches:** εισάγει στην βάση τις αναζητήσεις του χρήστη με το συγκεκριμένο id. Στο body περιέχει την παράμετρο accommodation.

Υπάρχουν συναρτήσεις στο app.js που ανήκουν στο backend που αναλύονται στο τρίτο κεφάλαιο επειδή χρησιμοποιούνται για την υλοποίηση του ερωτήματος bonus.

Κεφάλαιο 2: *Front end*

Το front end κομμάτι της εργασίας αναπτύχθηκε με τον package manager της Node.js [npm](#). Όλες οι σελίδες του ιστοχώρου(views) αναπτύχθηκαν στο framework της javascript **Vue.js** και έγινε εκτεταμένη χρήση του css framework της [Bootstrap](#) και της βιβλιοθήκης από components [Bootstrap-Vue](#).

Για την λειτουργικότητα των views έγινε χρήση της βιβλιοθήκης [axios](#) για την δημιουργία αιτημάτων στον εξυπηρέτη, της [jsonwebtoken](#) και της [vuex](#) για την αναγνώριση του συνδεδεμένου χρήστη, μέσω της αποκρυπτογράφησης του token και χρήσης reactive καθολικών μεταβλητών. Ακόμη, για την εξαγωγή των αρχείων σε μορφή xml χρησιμοποιήθηκε η βιβλιοθήκη [xml-js](#) στο **Admin.vue**.

Επιπλέον χρησιμοποιήθηκαν τα εξής:

- [iconify](#): χρησιμοποιήθηκε το σετ εικονιδίων του, ion-icons.
- [vue-hotel-datepicker](#): component που χρησιμοποιήθηκε στα **Home.vue** και **Results.vue** για την υλοποίηση του ημερολογίου της φόρμας αναζήτησης. Στο **Accommodation.vue** για την προβολή των μη διαθέσιμων ημερομηνιών και την επιλογή νέων ημερομηνιών σε περίπτωση κράτησης. Στο **AccommodationEdit.vue** για την προβολή και προσθήκη νέων μη διαθέσιμων ημερομηνιών από τον οικοδεσπότη στο χώρο.
- [vue-star-rating](#): component που χρησιμοποιείται στο **Accommodation.vue** για την βαθμολογία στην υποβολή κριτικών.
- [vue2-leaflet](#) & [leaflet](#): το vue2-leaflet εγκαθιστά την leaflet. Χρησιμοποιήθηκαν τα Lmap, LTileLayer, LMarker και Icon για την προβολή των χαρτών OpenStreetMap και των marker πάνω σε αυτούς στα **Accommodation.vue** και **AccommodationEdit.vue**.

- [leaflet-geosearch](#): plugin του leaflet. Το OpenStreetMapProvider της leaflet-geosearch χρησιμοποιήθηκε στο **geoInput.vue** για την προβολή των τοποθεσιών με βάση την αναζήτηση του χρήστη και την μετάφραση του σε bounds συντεταγμένων.

Κεφάλαιο 3: **Bonus**

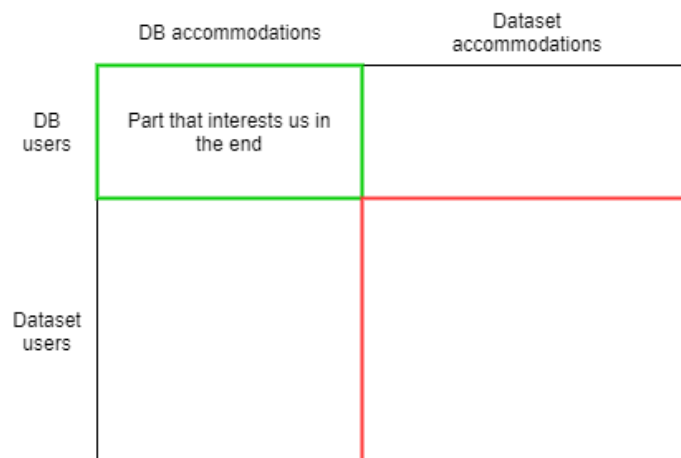
Υλοποίηση του bonus έγινε σε [Python3](#) με την χρήση των βιβλιοθηκών [re](#), [pandas](#), [NumPy](#), [Natural Language Toolkit](#), [json](#), [shutil](#) και [TextBlob](#).

Στον φάκελο bonus dataset βρίσκονται τα csv αρχεία του recommended dataset. Στον φάκελο bonus υπάρχουν τα απαιτούμενα tables της βάσης δεδομένων exported σε αρχεία json καθώς και το python script **bonus.py** που περιέχει την υλοποίηση του αλγορίθμου.

Κατά την εκτέλεση του **bonus.py** γίνεται ανάγνωση του bonus dataset το οποίο υπόκειται σε κάποιες απαραίτητες τροποποιήσεις (αφαίρεση μη ASCII χαρακτήρων, αφαίρεση σημείων στίξης, μετατροπή σε lowercase, αφαίρεση κενών entries και αφαίρεση stopwords) για να γίνει εφικτή η εφαρμογή της συνάρτησης sentiment της [TextBlob](#). Σε κάθε entry ανατέθεται ένα score με βάση την τιμή polarity που έχει προκύψει από το text processing.

Ακολουθεί η ανάγνωση των exported δεδομένων της βάσης και η κατανομή των score. Αν κάποιος χρήστης έχει αφήσει κριτική ως score κρατάμε το rating της κριτικής του στον χώρο αυτό. Αν ένας χρήστης έχει μείνει στον χώρο χωρίς να αφήσει κριτική τότε το score παίρνει την τιμή 3. Τέλος αν ένας χρήστης δεν έχει μείνει σε κάποιον χώρο αλλά έχει δει την πλήρη σελίδα προβολής τότε το score λαμβάνει την τιμή 2. Στην περίπτωση που ο χρήστης δεν έχει αλληλεπιδράσει καθόλου με έναν χώρο τότε το score είναι 0.

Έπειτα φτιάχνεται ένας πίνακας users x accommodations με τις τιμές των score που έχουν προκύψει.



Ο πίνακας αυτός λειτουργεί σαν είσοδος για τον αλγόριθμο του matrix factorization με gradient descent. Επιλέχθηκε ως παράμετρος K να μπει το 3 διότι διαφορετικά αυξάνεται υπερβολικά πολύ ο χρόνος εκτέλεσης, για μεγαλύτερη ακρίβεια μπορεί να αυξηθεί για μεγαλύτερη ακρίβεια. Ο αλγόριθμος δημιουργεί 2 πίνακες $users \times K$ και $K \times accommodations$, αρχικά τα γεμίζει με τυχαίες τιμές, δημιουργεί έναν νέο πίνακα από τα εσωτερικά γινόμενα κάθε user με κάθε accomodation, και εκτελεί τον αλγόριθμο του gradient descent με στόχο τις μη μηδενικές τιμές του αρχικού πίνακα. Προσαρμόζει τις τυχαίες τιμές με τέτοιο τρόπο έτσι ώστε να έχουν καλύτερα αποτελέσματα προς το τετράγωνο του αθροίσματος της διαφοράς κάθε γνωστής τιμής. Η διαδικασία αυτή επαναλαμβάνεται μέχρι αυτό να γίνει κάτω του 1 (μπορεί να μειωθεί για καλύτερα αποτελέσματα εις βάρος του χρόνου εκτέλεσης). Ο τελικός πίνακας γίνεται export σε μορφή json.

Η εκτέλεση του **bonus.py** και το exportation της βάσης δεδομένων είναι διαδικασίες που διαχειρίζεται το **app.js**. Η ρουτίνα ανανέωσης των recommendations εκτελείται κατά την εκκίνηση του Apache Server καθώς και περιοδικά την πρώτη μέρα της εβδομάδας ([cron](#)).

Επίλογος

Απαιτήσεις

Για την λειτουργία του ιστότοπου:

- [XAMPP](#) για την εκτέλεση του Apache Server και MySQL.
- `npm install` για την εγκατάσταση όλων των dependencies της node.js.
- `pip3 install pandas numpy nltk textblob json` για την εγκατάσταση όλων των dependencies της python.

Εκτέλεση

Ανοίξτε δύο terminal στον φάκελο της εργασίας και εκτελέστε στον καθένα τις ακόλουθες εντολές:

```
npm run serve
```

```
nodemon src\app.js
```

(το src\app.js είναι το relative path του app.js αν αλλάξει η θέση του app.js πρέπει να αλλάξει και η εντολή).

Δυσκολίες

- Το [vue-hotel-datepicker](#) έχει ένα bug με τα disabled dates όπου δεν τις προβάλλει στο ημερολόγιο αν δεν επιλέξει ο χρήστης πρώτα κάποιες ημερομηνίες.
- Το [vue-hotel-datepicker](#) έκανε πρόσφατα αναβάθμιση που έβαλε δυνατότητα προβολής κρατήσεων στο ημερολόγιο ξεχωριστά από τις disabled ημερομηνίες, αλλά είχε bug κατά το οποίο το ημερολόγιο δεν πρόβαλε καθόλου τα disabled dates.
- Οι τιμές των numpy arrays μπορούν να πάθουν overflow, με αποτέλεσμα ο τελικός πίνακας να μην δίνει έγκυρα αποτελέσματα. Για να αποφευχθεί αυτό, χρησιμοποιήσαμε χαμηλότερο learning rate για να κοινωνικοποιηθούν περισσότερο οι τιμές και να μην ξεφύγουν από το overflow limit.

Παραδοχές

- Δεν υπάρχει η δυνατότητα διαγραφής χώρου και εικόνας.
- Στον χρήστη μπορούν να προταθούν χώροι που έχει ήδη επισκεφθεί.
- Έχει γίνει διαφορετική υλοποίηση σε ότι αφορά τα *past searches* και τις προβολές χώρων ως προς την επιρροή τους στα *recommendations* (αναλυτικότερα στο κεφάλαιο 3).
- Στο πεδίο τοποθεσίας ο χρήστης πρέπει να επιλέξει μια περιοχή από αυτές που του προτείνει το *geoSearch*, διαφορετικά δεν θα λάβουν τις σωστές τιμές οι συντεταγμένες.