



Εργασία στο Μάθημα «Τεχνολογίες Εφαρμογών Διαδικτύου»

Τμήμα Πληροφορικής και Τηλεπικοινωνιών - Πανεπιστήμιο Αθηνών

Εργασία

Ακαδ. Έτος 2021-2022

ΟΝΟΜΑΤΕΠΩΝΥΜΟ	A.M.	EMAIL
ΑΡΙΣΤΟΤΕΛΗΣ ΒΕΡΓΙΝΗΣ	1115201700014	sdi1700014@di.uoa.gr
ΜΑΝΤΖΟΥΡΑΝΗΣ ΓΕΩΡΓΙΟΣ	1115201700076	sdi1700076@di.uoa.gr

Περιεχόμενα

Περιβάλλον Υλοποίησης	1
Λεπτομέρειες Υλοποίησης	2
Πλοήγηση	3
Οδηγίες Εγκατάστασης	4

Περιβάλλον Υλοποίησης

Windows 10

Λεπτομέρειες Υλοποίησης

Στόχος της εργασίας είναι η ανάπτυξη ενός Web API δημοπρασιών, τύπου ebay. Η εφαρμογή αυτή δίνει τη δυνατότητα στους χρήστες, δημιουργώντας λογαριασμό να μπορούν να πουλήσουν, να αγοράσουν προϊόντα αλλά και να λάβουν μέρος σε δημοπρασίες. Οι επισκέπτες μπορούν να δουν τα προϊόντα, χωρίς να λαμβάνουν μέρος στις διαδικασίες αγοραπωλησίας. Δίνεται επίσης η δυνατότητα στους χρήστες με τη χρήση OpenStreetMap, να βλέπουν την τοποθεσία των αντικειμένων. Επίσης, υπάρχει η δυνατότητα ανταλλαγής μηνυμάτων μεταξύ των χρηστών. Τέλος, υπάρχει και ένας διαχειριστής, ο οποίος είναι σε θέση να δεχτεί/απορρίψει αιτήσεις χρηστών, να κατεβάσει όλα τα αντικείμενα σε JSON και XML μορφή κ.α.

Τεχνολογίες Ανάπτυξης

- Back end

Χρησιμοποιήσαμε node.js και express.js. Η εφαρμογή ξεκινάει από το αρχείο `backend/server.js` και ακούει στην θύρα **5000**. Το base url είναι <https://localhost/5000>. Γίνεται επίσης χρήση ssl certificate.

- Front end

Το Μετωπιαίο Άκρο υλοποιήθηκε με ReactJS. Η επικοινωνία των δύο άκρων βασίζεται σε κλήσεις https, καθώς και στο front end γίνεται χρήση self-signed certificate, με κλειδιά τα οποία έχουμε δημιουργήσει και μοιράζουμε τόσο στο back όσο και στο front.

- Database

Χρησιμοποιούμε MySql με Serialize. Οι εγγραφές φαίνονται με τη χρήση του phpmyadmin.

Πλοήγηση

☐ **SignIn/SignUp:**

Εισαγωγή και έλεγχος των username, email, password, afm, phoneNumber, firstname, lastname κ.α. Με την εγγραφή οι χρήστες περιμένουν την έγκριση της αίτησής τους από τον **admin**. Αν γίνει δεκτή προχωρούν στην εφαρμογή και στο **SignIn**, αλλιώς μπορούν να πλοηγηθούν στην αναζήτηση προϊόντων.

- ☐ PATCH **api/v1/admin/users/:id/accept**
- ☐ PATCH **api/v1/admin/users/:id/decline**
- ☐ GET **api/v1/admin/users/download/json**
- ☐ GET **api/v1/admin/users/download/xml**
- ☐ GET **api/v1/users/**
- ☐ DELETE **api/v1/id/:id**

Όλα τα παραπάνω είναι routes, που ο admin μπορεί να εγκρίνει/απορρίψει τους χρήστες, να διαγράψει με βάση το **id** και να κατεβάσει τα auction σε **json/xml** μορφή. Όλα τα παραπάνω requests, ανάλογα με το αν πέτυχαν ή όχι επιστρέφουν κάποιο **status**, ανάλογα με το αν πέτυχαν ή όχι.

- ☐ POST **api/v1/items**
- ☐ GET **api/v1/items**
- ☐ GET **api/v1/items/:userId**
- ☐ POST **api/v1/items/:itemId/bid**
- ☐ POST **api/v1/items/:itemId/buy**
- ☐ DELETE **api/v1/items/:id**

Όλα τα παραπάνω routes, αφορούν τις δημοπρασίες. Για την δημιουργία νέας δημοπρασίας, έχουμε μια φόρμα συμπλήρωσης πεδίων όπως : **location**, **first_bid**, **buy_price**, **country**, **end_date** τα οποία και στέλνουμε με ένα **post request** στο **api/v1/items**. Κατά τη συμπλήρωση της φόρμας, δίνεται η δυνατότητα στον χρήστη να συμπληρώσει την ακριβή του τοποθεσία με τη χρήση ενός χάρτη, ρυθμίζοντας τα πεδία **latitude**, **longitude** αυτόματα. Επίσης, ένα **post request**, στο **api/v1/items/:itemId/bid**, μπορεί ο χρήστης να δημιουργήσει ένα bid για το αντικείμενο **itemId**, αν το **amount** που τοποθέτησε είναι μεγαλύτερο του **current_bid**. Μπορεί ακόμα να αγοράσει

ένα αντικείμενο με παρόμοια λογικά με το **bid**, αρκεί να υπάρχει το **buy_price**, αλλιώς επιστρέφεται **error_bad_request 400**.

☐ GET **api/v1/search?**

Ο χρήστης μπορεί να αναζητήσει στα auctions με βάση **category**, **location**, **price**, **text**, **ένα κάθε φορά**. Χρησιμοποιούμε επίσης, pagination, για να εμφανίζονται τα αντικείμενα ομαδοποιημένα και να μπορούν οι χρήστες να προχωρούν στις σελίδες.

☐ GET **api/v1/upload**

Με ένα **GET** στο route αυτό, αρχικοποιούμε με fake data τη βάση μας, διαβάζοντας τις τιμές από τα αρχεία που μας δώσατε.

☐ POST **api/v1/messages/:username/send**

☐ GET **api/v1/messages/:recieverId/recieved**

☐ GET **api/v1/messages/:senderId/sent**

☐ DELETE **api/v1/messages/:userId/delete**

Οι χρήστες με post, get, delete requests, μπορούν να στείλουν, να λάβουν και να διαγράψουν μηνύματα.

Οδηγίες Εγκατάστασης

1. Εγκατάσταση βάσης δεδομένων
 1. Εγκατάσταση mysql και δημιουργία βάσης **auction-website-clone** , διαφορετικά χρήση **phpmyadmin** μαζί με **xampp**. Ξεκινώντας τον apache και τον mysql η βάση βρίσκεται στο <http://localhost/phpmyadmin>.
 2. Το αρχείο με τους κωδικούς και το όνομα της βάσης είναι στο **config/db.config.js**.
2. Προετοιμασία και εκτέλεση server: **https://localhost:5000**
 1. Εγκατάσταση nodejs **sudo apt install nodejs**
 2. **sudo apt install npm**
 3. **npm i** μέσα στο **back end**

4. **npm start** για την εκτέλεση

3. Προετοιμασία και εκτέλεση client (React) <https://localhost:3000>

1. **npm install react-typed --legacy-peer-deps**
 2. **npm i** μέσα στο front-end φάκελο
 3. **npm start**
- Υπάρχει περίπτωση κατά την είσοδο στο <https://localhost:3000> να υπάρχει πρόβλημα με το self-signed certificate.

Πίνακες βάσης

users

items

bids

messages