

**ΕΡΓΑΣΙΑ ΣΤΑ ΠΛΑΙΣΙΑ ΤΟΥ ΜΑΘΗΜΑΤΟΣ  
ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΚΑΙ ΣΥΣΤΗΜΑΤΑ ΣΤΟΝ  
ΠΑΓΚΟΣΜΙΟ ΙΣΤΟ**

ΖΕΡΒΟΥ ΠΟΥΛΧΕΡΙΑ-1067511-5 ΕΤΟΣ

ΠΑΝΔΗΣ ΑΝΑΣΤΑΣΙΟΣ-1056287-6 ΕΤΟΣ

ΠΟΡΤΟΚΑΛΟΓΛΟΥ ΑΝΔΡΟΝΙΚΗ-1067539-5 ΕΤΟΣ

## ΣΧΕΔΙΑΣΜΟΣ ΣΧΕΣΙΑΚΗΣ ΒΑΣΗΣ

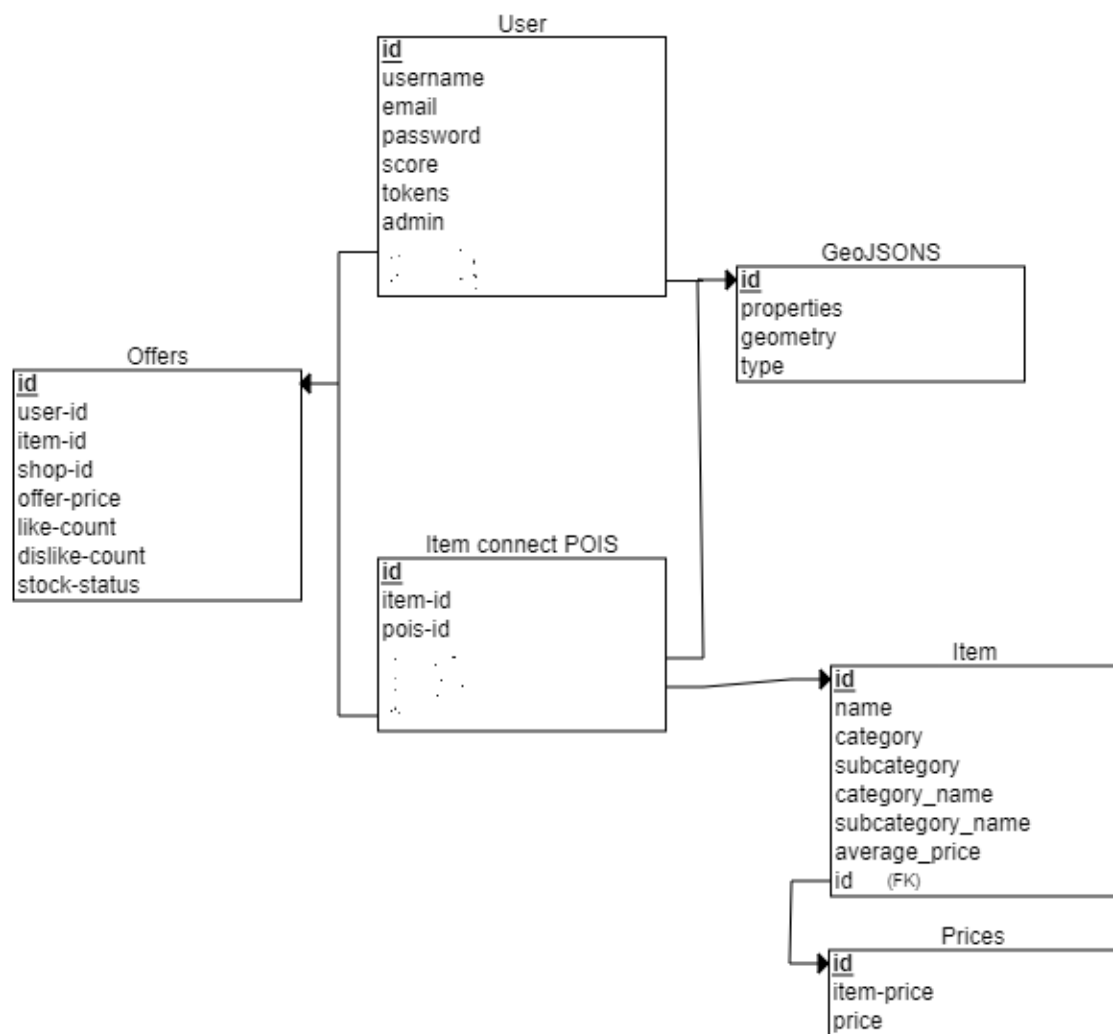
### ΑΝΑΛΥΣΗ ΚΩΔΙΚΑ

#### **App.js**

Ο κώδικας δημιουργεί έναν web server χρησιμοποιώντας τη βιβλιοθήκη Express στο Node.js. Ο server ακούει σε μια καθορισμένη θύρα (4242). Τα modules που χρησιμοποιούνται είναι express, express-handlebars, jsonwebtoken, body-parser, cors, node-schedule και Sequelize. Χρησιμοποιούνται τα αρχεία JSON: items.json και prices.json περιέχουν τα προϊόντα των μαγαζιών και τις τιμές των προϊόντων αντίστοιχα και ένα αρχείο geojson "new\_data.geojson" το οποίο περιέχει τα μαγαζιά/POIS.

Ο κώδικας δημιουργεί πίνακες σε μια βάση δεδομένων SQLite: Items, Prices, Users, Offers, GeoJSON, ItemConnectPOIS. Ο πίνακας Items ορίζεται και οι εγγραφές του δημιουργούνται από τα δεδομένα στο αρχείο items.json. Ο πίνακας Prices ορίζεται και οι εγγραφές του δημιουργούνται από τα δεδομένα στο αρχείο prices.json. Ο πίνακας GeoJSON ορίζεται και οι εγγραφές του δημιουργούνται από τα δεδομένα στο αρχείο new\_data.geojson. Οι πίνακες Users, Offers, ItemConnectPOIS ορίζονται και παίρνουν τιμές τις οποίες δημιουργούμε εμείς.

Ο κώδικας δημιουργεί επίσης ένα Express app, ρυθμίζει middlewares, όπως cors και body-parser, και χρησιμοποιεί τη βιβλιοθήκη Sequelize για να αλληλεπιδράσει με τη βάση δεδομένων SQLite. ο κώδικας χρησιμοποιεί το Sequelize ORM για να συνδεθεί σε μια βάση δεδομένων. Εάν η σύνδεση είναι επιτυχής, καταγράφει ένα μήνυμα και εκτελεί τρεις λειτουργίες: "calculateAveragePrice", "calculateScoreLikeDislike" και "calculateScoreOffer". Εάν η σύνδεση αποτύχει, καταγράφει ένα μήνυμα σφάλματος.



## Auth.js

Εξάγουμε μια συνάρτηση με όνομα "verifyJWT" που λειτουργεί ως middleware για την επαλήθευση ενός JSON Web Token (JWT) που περνάει στο request header με κλειδί "x-access-token". Το JWT αποκωδικοποιείται χρησιμοποιώντας τη βιβλιοθήκη "jsonwebtoken" και το μυστικό κλειδί "veryverysecret". Εάν το JWT είναι άκυρο, η συνάρτηση επιστρέφει ένα error code 401 με το error message "Invalid JWT". Εάν το JWT είναι έγκυρο, το request object παίρνει την τιμή του αποκωδικοποιημένου id του χρήστη και καλείται το επόμενο middleware.

## Sign-up

### Html signup

Έχει τρία fields για να εισάγει ο χρήστης το όνομά του, το email του και τον κωδικό πρόσβασης, καθώς και ένα κουμπί υποβολής με την ένδειξη "Sign up".

Η φόρμα έχει id "Sing-up" και τα πεδία εισόδου έχουν unique id's "name", "email" και "password". Το κουμπί υποβολής έχει identifier "log-button".

Στο τέλος του κώδικα, υπάρχει αναφορά σε ένα εξωτερικό αρχείο δέσμης ενεργειών με όνομα "script\_SingUp.js".

### Script Sign Up

Ξεκινάμε με τη μέθοδο document.getElementById. Έπειτα προστίθεται ένας event listener να ακούει για το event submit. Στη συνέχεια, ο κώδικας χρησιμοποιεί την document.getElementById για να ανακτήσει τις τιμές που εισήγαγε ο χρήστης στα πεδία εισόδου της φόρμας. Οι τιμές των πεδίων "name", "email" και "password" αποθηκεύονται στις μεταβλητές username, email και password αντίστοιχα.

Πραγματοποιείται μια απλή επικύρωση φόρμας για να ελεγχθεί αν ο χρήστης έχει συμπληρώσει όλα τα απαιτούμενα πεδία. Εάν κάποιο από τα πεδία είναι κενό, εμφανίζεται ένα μήνυμα στο χρήστη και η συνάρτηση ολοκληρώνει χωρίς να προχωρήσει περαιτέρω.

Στη συνέχεια, τα δεδομένα της φόρμας αποστέλλονται στον server με χρήση της fetch. Η διεύθυνση URL του server που καθορίζεται είναι "http://127.0.0.1:4242/signup", η οποία είναι μια διεύθυνση URL localhost στη θύρα 4242. Η fetch πραγματοποιεί POST request στον διακομιστή με τα δεδομένα της φόρμας σε μορφή JSON. Εάν το property data.success είναι true, εμφανίζεται ένα

μήνυμα ειδοποίησης στο χρήστη που υποδεικνύει την επιτυχή εγγραφή. Εάν το `data.success` είναι `false`, εμφανίζεται ένα μήνυμα σφάλματος στο χρήστη, συμπεριλαμβανομένου του μηνύματος `data.error` από το διακομιστή.

### Signup endpoint

Δημιουργήσαμε ένα end point για τη φόρμα "Sign Up". Το end point ορίζεται χρησιμοποιώντας τη μέθοδο `app.post`, η διεύθυνση URL του endpoint είναι `"/signup"`.

Όταν ένας χρήστης υποβάλλει τη φόρμα "Sign Up", τα δεδομένα της φόρμας αποστέλλονται σε αυτό το endpoint μέσω ενός POST request από το script από την πλευρά του client. Το εισερχόμενο αίτημα αντιμετωπίζεται από τη συνάρτηση (`req, res`), όπου `req` είναι το εισερχόμενο αίτημα και `res` είναι η απάντηση που θα σταλεί πίσω στον client.

Καταγράφεται το `body` του εισερχόμενου αιτήματος στην κονσόλα χρησιμοποιώντας την εντολή `console.log(req.body)`. Το `body` του request αποδομείται για να εξαχθούν τα `properties` όνομα, email και κωδικός πρόσβασης, οι οποίες αποθηκεύονται σε μεταβλητές με το ίδιο όνομα.

Στη συνέχεια καλείται η μέθοδος `Users.create` για τη δημιουργία ενός νέου χρήστη στη βάση δεδομένων με τις παραπάνω ιδιότητες. Εάν η δημιουργία είναι επιτυχής, εκτελείται η μέθοδος `.then` και αποστέλλει μια απάντηση JSON στον πελάτη με το `property success` να έχει την τιμή `true`.

Εάν υπάρχει σφάλμα στη δημιουργία, τρέχει η μέθοδος `.catch` και καταγράφει το σφάλμα στην κονσόλα. Η μέθοδος `.catch` στέλνει ένα JSON στον client με το `property success` να είναι `false` και ένα `property error` που περιέχει το `error message`.

Το endpoint είναι υπεύθυνο για τη δημιουργία νέων χρηστών στη βάση δεδομένων και την αποστολή μιας απάντησης στον client που υποδεικνύει την επιτυχία ή την αποτυχία απόπειρας δημιουργίας νέου χρήστη.

### Create account

Sign up

## Log Out

Η συνάρτηση `logout()` διαφράζει το JSON Web Token (JWT) από το local storage του client και ανακατευθύνει το χρήστη στη σελίδα του log in.

## Log-In

### Html Login

Δημιουργούμε μια βασική δομή ιστοσελίδας με head και body. Στην ενότητα head, ο τίτλος της καρτέλας έχει οριστεί σε "Login" και γίνεται σύνδεση με ένα stylesheet. Στο τμήμα body, η φόρμα σύνδεσης βρίσκεται μέσα σε ένα div για ευκολότερο χειρισμό. Η φόρμα έχει τρία πεδία εισαγωγής όνομα, email, κωδικό πρόσβασης και ένα κουμπί υποβολής με το κείμενο "Login". Μια κλάση div έχει επίσης ρυθμιστεί να εμφανίζει ένα μήνυμα που ζητά από τους χρήστες να δημιουργήσουν έναν λογαριασμό εάν δεν έχουν ήδη, με ένα link προς τη σελίδα SignUp. Ένα αρχείο JavaScript συνδέεται στο τέλος της ενότητας του σώματος για περαιτέρω λειτουργικότητα.

### Javascript Login

Ο κώδικας ξεκινά λαμβάνοντας το form element με αναγνωριστικό "login" και επισυνάπτοντας ένα event listener στο event submit. Το event listener είναι μια συνάρτηση που καταγράφει τα δεδομένα του γεγονότος κατά την υποβολή της φόρμας.

Στη συνέχεια, η συνάρτηση ανακτά τις τιμές που έχουν εισαχθεί στα forms για το όνομα χρήστη, το email και τον κωδικό πρόσβασης και τις αποθηκεύει στο αντικείμενο "data". Έπειτα, η συνάρτηση ελέγχει εάν κάποιο από τα απαιτούμενα πεδία είναι κενό. Εάν κάποιο από τα πεδία είναι κενό, η συνάρτηση θα εμφανίσει την αντίστοιχη ειδοποίηση.

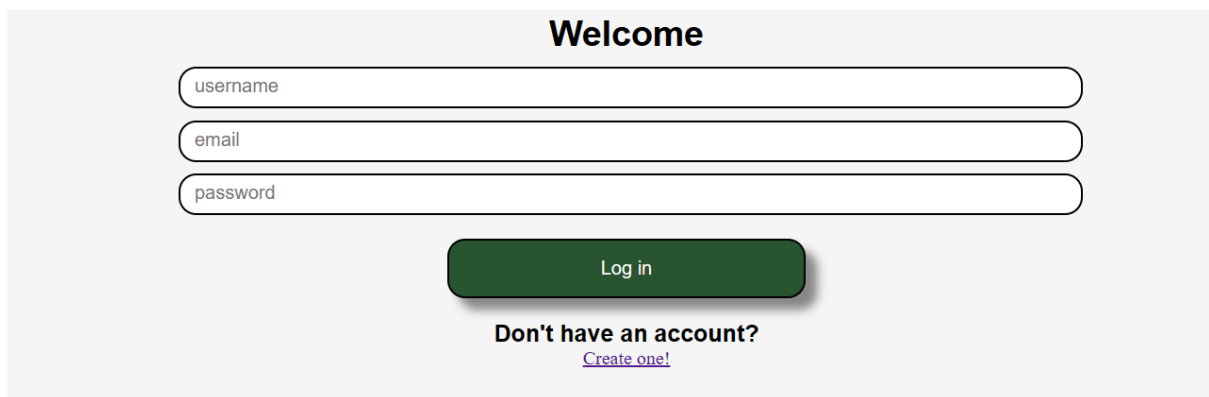
Εάν όλα τα πεδία είναι συμπληρωμένα, η συνάρτηση POST request στη διεύθυνση "http://127.0.0.1:4242/login" με το αντικείμενο "data" ως body. Το αίτημα περιλαμβάνει μια κεφαλίδα που υποδεικνύει τον τύπο περιεχομένου ως "application/json".

Η απάντηση από τον server έχει τη μορφή JSON, το οποίο ο κώδικας κάνει parse σε ένα JavaScript object. Εάν το αποτέλεσμα περιέχει πεδίο "succes" με τιμή "true", ο κώδικας αποθηκεύει την τιμή του πεδίου "token" τοπικά και ανακατευθύνει τον χρήστη στη διεύθυνση URL "http://127.0.0.1:5500/Map.html".

Εάν το πεδίο "succes" είναι ψευδές, θα εμφανιστεί ένα μήνυμα που υποδεικνύει λανθασμένο το όνομα χρήστη ή τον κωδικό πρόσβασης. Σε περίπτωση σφάλματος καταγράφεται το σφάλμα στην κονσόλα.

### Login End-Point

Δημιουργούμε ένα End-Point για τη σύνδεση των χρηστών. Ένα αίτημα HTTP POST λαμβάνει ένα body που περιέχει όνομα χρήστη, email και τον κωδικό πρόσβασης. Στη συνέχεια, ρωτά τη βάση δεδομένων "Users" για αντιστοίχιση με το παρεχόμενο όνομα χρήστη, email και κωδικό πρόσβασης. Εάν βρεθεί μια αντιστοίχιση, end-point στέλνει απάντηση όπου το property: succes είναι true και ένα JSON Web Token (JWT) που περιέχει το αναγνωριστικό του χρήστη, υπογεγραμμένο με μια μυστική συμβολοσειρά "veryverysecret". Εάν δεν βρεθεί αντιστοίχιση, το end-point στέλνει μια απάντηση όπου το property: succes είναι false και ένα error 401.



**Welcome**

username

email

password

Log in

Don't have an account?  
[Create one!](#)

## Map

### Html Map

Δηλώνουμε τίτλο καρτέλας "Home". Κάνουμε import το stylesheet.

Το body περιέχει το navigation bar, το οποίο περιλαμβάνει συνδέσμους προς άλλες σελίδες ("Αρχική σελίδα", "Είδη", "Προσφορές", "Πίνακας κατάταξης", "Ο λογαριασμός μου"), ένα field, ένα κουμπί υποβολής και μια φόρμα φίλτρου με ένα drop-down μενού για την επιλογή μιας κατηγορίας και ένα κουμπί για την εφαρμογή του φίλτρου.

Προσθέτουμε έναν χάρτη από τη βιβλιοθήκη Leaflet. Το script.js περιλαμβάνεται στο τέλος.

### Javascript Map

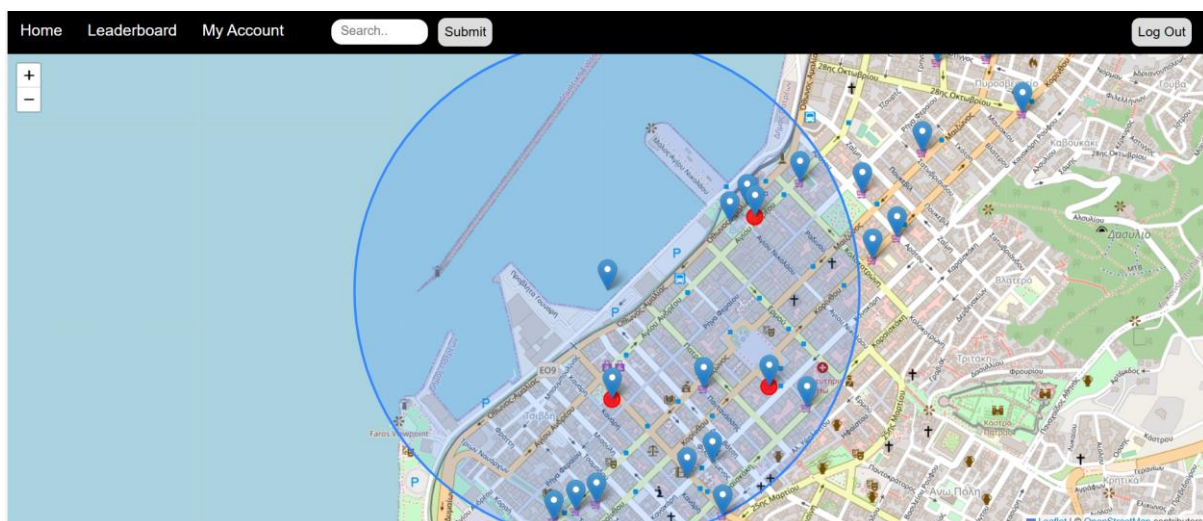
Χρησιμοποιούμε τη βιβλιοθήκη Leaflet για να δημιουργήσουμε έναν χάρτη και να εμφανίσουμε διάφορες τοποθεσίες σε αυτόν. Ο κώδικας ορίζει την ακτίνα ενός κύκλου σε 600 μέτρα που θα είναι κεντραρισμένος γύρω από την τρέχουσα θέση του χρήστη. Ορίζεται μια ασύγχρονη συνάρτηση "getGeolocation" για να πάρουμε την τρέχουσα τοποθεσία του χρήστη χρησιμοποιώντας το Geolocation API. Ο χάρτης

αρχικοποιείται και προστίθεται σε αυτόν ένα tile layer από το OpenStreetMap. Στη συνέχεια, ο κώδικας χρησιμοποιεί το fetch API για να κάνει ένα GET request στο "http://127.0.0.1:4242/Map" για να ανακτήσει τα geojson για όλες τις τοποθεσίες που θα εμφανιστούν στο χάρτη. Οι τοποθεσίες εμφανίζονται στο χάρτη αναδυόμενα παράθυρα που εμφανίζουν το όνομα της τοποθεσίας, τις ώρες λειτουργίας και τις πληροφορίες του καταστήματος, με κουμπιά για "Δημιουργία προσφορών" και "Εμφάνιση προσφορών" αν βρίσκονται εντός του κύκλου αλλιώς εμφανίζεται μόνο το κουμπί εμφάνισης προσφορών. Επίσης υλοποιήθηκε search function.

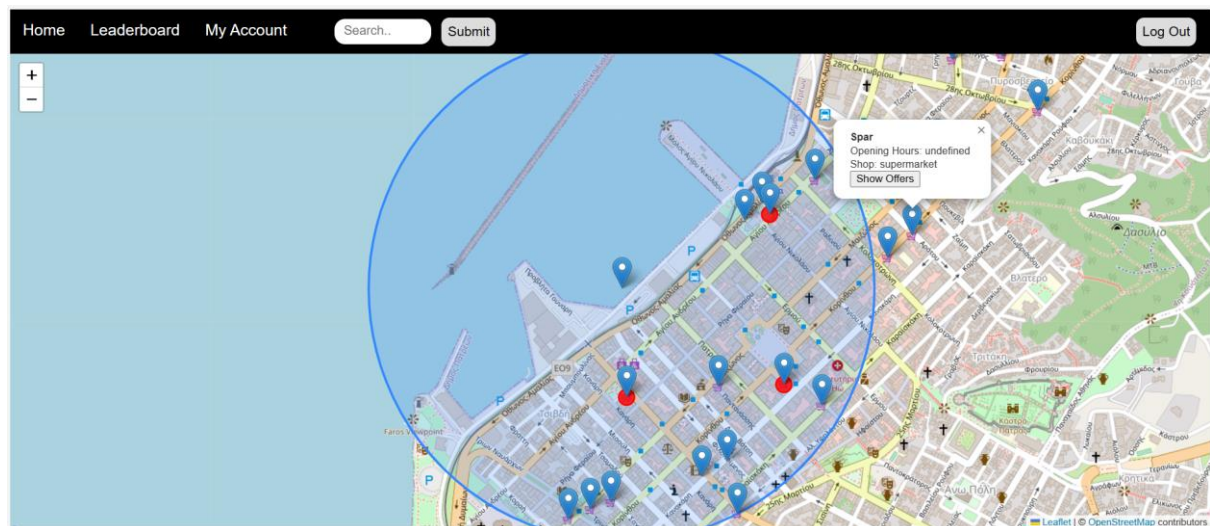
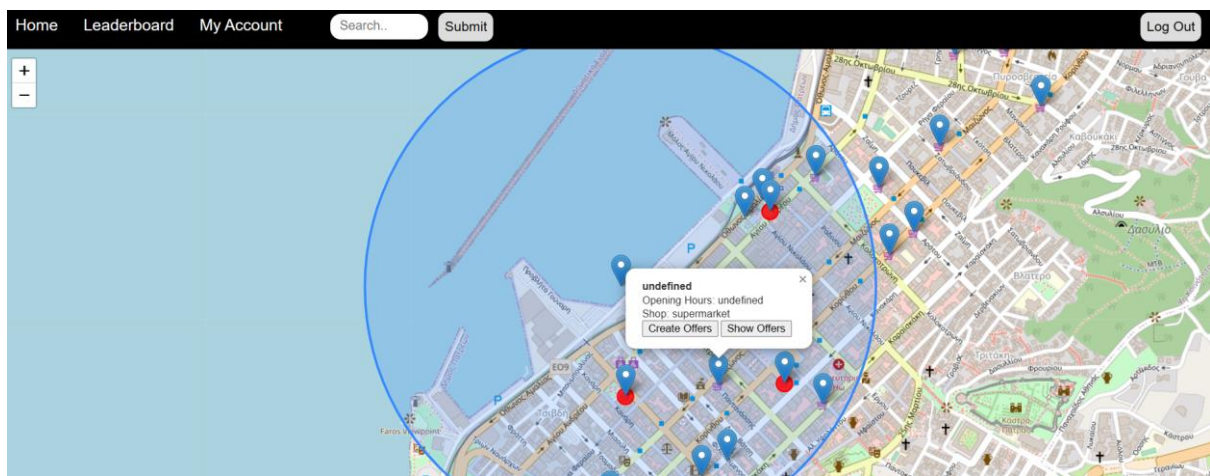
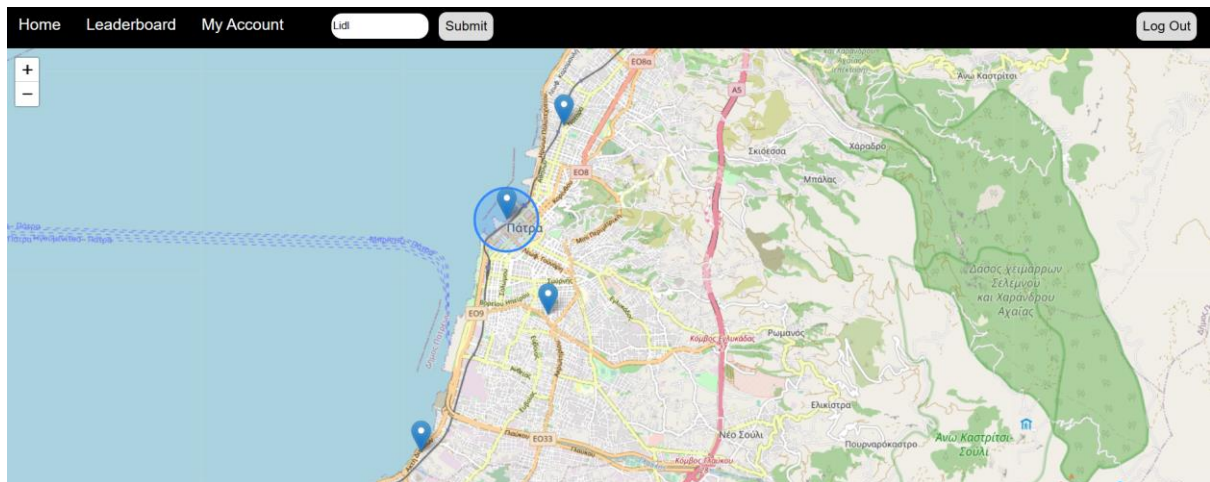
### Map End-Point

Το GET request, "/Map", ανακτά όλα τα entries από το μοντέλο "GeoJSON" και επιστρέφει τα δεδομένα σε μορφή JSON. Η μέθοδος findAll() χρησιμοποιείται για την ανάκτηση όλων των εγγραφών από τον πίνακα "GeoJSON". Στη συνέχεια, τα αποτελέσματα μεταφέρονται στο res ως JSON string χρησιμοποιώντας τη μέθοδο res.json(). Εάν προκύψει σφάλμα κατά τη διάρκεια της διαδικασίας, θα καταγραφεί στην κονσόλα το σφάλμα και θα σταλεί ως απάντηση στον client Error 500.

Το GET request, "/Map/:id", χρησιμοποιείται για τη λήψη όλων των καταστημάτων με προσφορές. Το request έχει μια παράμετρο id η οποία αντιπροσωπεύει το id του καταστήματος. Η εντολή "SELECT" χρησιμοποιείται για την άντληση της στήλης shop\_id από τον πίνακα "offers" όπου το shop\_id δεν είναι null. Στη συνέχεια, τα αποτελέσματα επιστρέφονται ως JSON string χρησιμοποιώντας τη μέθοδο res.json(). Εάν προκύψει σφάλμα κατά τη διάρκεια της διαδικασίας, θα καταγραφεί στην κονσόλα το σφάλμα και θα σταλεί ως απάντηση στον client Error 500.







## Items

Html Itemspage

Το αρχείο αυτό αποτελείται από τον HTML κώδικα που χρειάζεται για την υλοποίηση της λειτουργίας του της υποβολής προσφοράς. Τα βασικά του στοιχεία είναι το navigation bar, dropdown λίστα για κατηγοριοποίηση βάση κατηγορία και υποκατηγορία και search bar για αναζήτησης βάση ονόματος προϊόντος.

### Javascript Itemspage

Επιτρέπει τη λογική λειτουργία της αναζήτησης βάση κατηγορίας και υποκατηγορίας όπως και τη λειτουργία αναζήτηση προϊόντος με βάση το όνομα του. Η συνάρτηση updateUI παίρνει σαν παράμετρο ένα πίνακα που αποτελείτε από τα προϊόντα του συγκεκριμένου μαγαζιού που επιλέχθηκε από το pop-up του χάρτη. Όταν το κουμπί Create an Offer πατηθεί εμφανίζεται ένα νέο πεδίο εισαγωγής στο οποίο ο χρήστης μπορεί να προσθέσει τη νέα τιμή της προσφορά του προϊόντος. Όταν πατηθεί το κουμπί Save Offer αποθηκεύει τη νέα προσφορά στέλνοντας ένα POST request στο url 'http://127.0.0.1:4242/newoffer' με το id του προϊόντος , τη νέα τιμή και το id του μαγαζιού ως το request body.

Η συνάρτηση searchProduct ανακτά το όνομα του προϊόντος που εισήγαγε ο χρήστης και το χρησιμοποιεί στο GET request στο url 'http://127.0.0.1:4242/search ' με το όνομα και το id του μαγαζιού ως παραμέτρους. Έπειτα καλεί τη συνάρτηση updateUI με την απάντηση από τον server.

Η συνάρτηση searchCategory ανακτά την επιλεγμένη από τον χρήστη κατηγορία και υποκατηγορία και τις χρησιμοποιεί στο GET request στο url 'http://127.0.0.1:4242/itemspage' με τη κατηγορία , υποκατηγορία και το id του μαγαζιού ως παραμέτρους. . Έπειτα καλεί τη συνάρτηση updateUI με την απάντηση από τον server.

### Category/ Sub-category endpoint

Διαχειρίζεται τη κατηγοριοποίηση των προϊόντων βάση κατηγορίας και υποκατηγορίας. Πρώτα επιλέγονται τα προϊόντα που ανήκουν στο συγκεκριμένο κατάσταση που έχει επιλεγθεί και έπειτα τα προϊόντα κατηγοριοποιούνται με βάση την κατηγορία ή/και την υποκατηγορία που έχει επιλέξει ο χρήστης. Τα κατηγοριοποιημένα προϊόντα επιστρέφονται σαν απάντηση σε JSON μορφή.

### Search by name endpoint

Διαχειρίζεται την αναζήτηση των προϊόντων με βάση το όνομά τους. Πρώτα επιλέγονται τα προϊόντα που ανήκουν στο συγκεκριμένο κατάσταση που έχει επιλεγθεί και έπειτα τα προϊόντα κατηγοριοποιούνται με βάση τη τιμή του ονόματος

που εισήγαγε ο χρήστης. Τα κατηγοριοποιημένα προϊόντα επιστρέφονται σαν απάντηση σε JSON μορφή.

[Home](#) [Leaderboard](#) [My Account](#)

Filter

Category: Όλα

Subcategory: Όλα

Search...

Search

Items

Όμο Σκόνη 425γρ	Καθαριότητα	Απορρυπαντικά	Create Offer
Γιώτης Κρέμα Παιδική Φαρίν Λακτέ 300γρ	Βρεφικά Είδη	Βρεφικές τροφές	Create Offer
Μύλοι Αγίου Γεωργίου Καλαμποκάλευρο 500γρ	Τρόφιμα	Αλεύρι - Σιμιγδάλι	Create Offer
Heineken Μπίρα Premium Lager 500ml	Ποτά - Αναψυκτικά	Μπίρες	Create Offer

Filter

Category: Βρεφικά Είδη

Subcategory: Βρεφικές τροφές

Search...

Search

Items

Γιώτης Κρέμα Παιδική Φαρίν Λακτέ 300γρ	Βρεφικά Είδη	Βρεφικές τροφές	Create Offer
--	--------------	-----------------	--------------

[Home](#) [Leaderboard](#) [My Account](#)

Filter

Category: Όλα

Subcategory: Όλα

Μπίρα

Search

Items

Heineken Μπίρα Premium Lager 500ml	Ποτά - Αναψυκτικά	Μπίρες	Create Offer
------------------------------------	-------------------	--------	--------------

## Offers

### Html offers

Το αρχείο αυτό αποτελείται από τον HTML κώδικα που χρειάζεται για την υλοποίηση της λειτουργίας του της προβολής των προσφορών. Τα βασικά του στοιχεία είναι το navigation bar και ένα div στοιχείο στο οποίο προβάλλονται οι προσφορές.

### Javascript offers

Ανακτά τις προσφορές ενός μαγαζιού από τον server και τις προβάλλει στην ιστοσελίδα. Αποτελείται από δύο βασικές συναρτήσεις τις `displayOffers(offers)` και την `getOffers()`.

`displayOffers(offers)`: παίρνει ως παράμετρο ένα πίνακα προσφορών και δημιουργεί HTML elements για να απεικονίσει τα στοιχεία κάθε προσφοράς. Τα στοιχεία συμπεραλαμβάνουν το όνομα του χρήστη που έκανε τη προσφορά, το όνομα του μαγαζιού, τη τιμή της προσφοράς και τον αριθμό των likes ή dislikes για μία προσφορά.

getOffers(): Αυτή η συνάρτηση ανακτά τα δεδομένα των προσφορών από τον server χρησιμοποιώντας μέθοδο fetch. Στέλνει ένα GET request στο 'http://127.0.0.1:4242/offers/\${shopID}' με το id του μαγαζιου (shopID) περασμένο σαν παραμετρος του query.

Ο header x-access-token χρησιμοποιείται για την αποστολή του διακριτικού στον διακομιστή με το API request, το οποίο μπορεί να χρησιμοποιήσει ο διακομιστής για να επαληθεύσει την αυθεντικότητα του πελάτη και να καθορίσει εάν είναι εξουσιοδοτημένος να έχει πρόσβαση στον πόρο που ζητήθηκε.

## Offers END-POINT

### Shop offers

Ανακτά όλες τις προσφορές ενός συγκεκριμένου μαγαζιού. Επιστρέφει το όνομα του μαγαζιού , το όνομα του προϊόντος , το όνομα του χρήστη που έκανε την προσφορά , τον αριθμό των likes και των dislikes , την τιμή προσφοράς του προϊόντος και το id της προφοράς.

### Offer likes/ offer dislike

Επιτρέπει στο χρήστη να κάνει like ή dislike σε μία προφορά αυξάνοντας ή μειώνοντας τον αριθμό τους. Επιστρέφει την ανανεωμένη τιμή των like/dislike.

### Offer out of stock

Επιτρέπει στον χρήστη να θέτει out of stock ένα προϊόν μίας προσφοράς ένα δεν το βρει στο μαγαζί.

[Home](#) [Leaderboard](#) [My Account](#)

## Offers

Username: Eric

Item Name: Fix Hellas Μπύρα 6X330ml

Offer Price: 3

Offers ID:1

Like (0) Dislike (1) Out of stock

Username: Bob

Item Name: Babyline Πάνες Μωρού Sensitive 3-6κιλ No 2 26τεμ

Offer Price: 5

Offers ID:4

Like (3) Dislike (0) In stock

Username: Bob

Item Name: Sanitas Σακ Απορ Ultra 52X75cm 10τεμ

Offer Price: 3

Offers ID:5

Like (2) Dislike (1) In stock

## My account

### Html myaccount

Η σελίδα αυτή διατηρεί το style που έχει το αρχείο style.css .Η σελίδα έχει στο πάνω μέρος το μενού το οποίο επιτρέπει στο χρήστη να μετακινηθεί προς οποιαδήποτε άλλη σελίδα επιθυμεί δυο λειτουργίες που υλοποιεί αυτή η σελίδα είναι να δίνει την δυνατότητα στο χρήστη να τροποποιήσει το κωδικό του και το email.Ακόμη προβάλλει τα δεδομένα του ώστε να μπορεί να τα βλέπει. Αυτές οι λειτουργίες πραγματοποιούνται μέσω του script\_myaccount.js το οποίο και γίνεται σε link μέσω του <script></script> στην html της σελίδας. Το script αυτό σε συνεργασία με τον app.js που είναι η πλευρά του server εμπεριέχουν δυο fetch και δυο endpoints αντίστοιχα .

### Myaccount END-POINT

Αρχικά, γίνεται ένα get request έτσι ώστε να πάρουμε τα δεδομένα του χρήστη μας και έπειτα χρησιμοποιώντας το .innerHTML στη javascript τα εμφανίζουμε στο επιθυμητό σημείο. Ακόμη γίνεται ένα post request όπου ανάλογα με τα δεδομένα που δίνει ο χρήστης γίνονται update τα δεδομένα του και αυτό χρησιμοποιώντας έναν eventlistener στο κουμπί edit που υπάρχει κάτω από την φόρμα αλλαγής.

Username:

Bob

Email:

newBob@email.com

Password:

\*\*\*\*\*

Edit account

Username: Bob

Email: Bob@example.com

Password : password

Score : 18

Username:

username

Email:

email

Password:

password

Edit account

Username: Bob

Email: newBob@email.com

Password : Iambob

Score : 18

## Leaderboard

### Html leaderboard

Το αρχείο αυτό αποτελείται από τον HTML κώδικα που χρειάζεται για την υλοποίηση της λειτουργίας του Leaderboard. Τα βασικά του στοιχεία είναι το navigation bar , το table το οποίο γεμίζει με τους χρήστες και τα κουμπιά Previous και Next που χρησιμοποιούνται για τη λειτουργία της σελιδοποίησης.

### Javascript leaderboard

Ο κώδικας αυτού του αρχείου ανακτά τις κατάλληλες πληροφορίες από τον server χρησιμοποιώντας fetch api. Ορίζεται η αρχική σελίδα ως 1 και δημιουργούνται δύο κουμπιά τα Previous και Next που χρησιμοποιούνται για την σελιδοποίηση. Όταν πατηθεί ένα από τα δύο κουμπιά ο αριθμός της σελίδας μειώνεται ή αυξάνεται αντίστοιχα και η συνάρτηση fetchUsers(page) καλείται έτσι ώστε να ανανεώσει τον πίνακα.

Η συνάρτηση fetchUsers(page) στέλνει ένα αίτημα GET στον server που περιλαμβάνει τον αριθμό της σελίδας και το limit 10. Ανανεώνει τον πίνακα με βάσει των επιστρεφόμενων JSON data αφαιρώντας τα προηγούμενα δεδομένα και προσθέτοντας νέες σειρές για κάθε χρήστη με στήλες για το όνομα του χρήστη και τη βαθμολογία.

## Leaderboard END-POINT

Δημιουργούμε ένα endpoint για τη διασύνδεση του Leaderboard. Το endpoint διαχειρίζεται το αίτημα GET για την διεύθυνση url `"/leaderboard"`. Η συνάρτηση ανακτά τον αριθμό της σελίδας και τον αριθμό των στοιχείων ανά σελίδα, εάν δεν παρέχεται ο αριθμός της σελίδας ορίζεται ως 1 και ορίζεται και ο αριθμός των χρηστών ανά σελίδα ως 10. Έπειτα υπολογίζεται ο συνολικός αριθμός σελίδων ως `#χρηστών/#χρηστών ανά σελίδα`. Η συνάρτηση επιστρέφει τους κορυφαίους χρήστες, ταξινομημένους με φθίνουσα σειρά. Τέλος επιστρέφει ως απάντηση στον client τους κορυφαίους χρήστες σε μορφή JSON.

Leaderboard	
Eric	60
Nikos	20
kwstas	19
Bob	18
Steven	18
John	17
Tassos	15
Niki	11
Poulcheria	10
spiros	7
<div>PreviousNext</div>	

Leaderboard	
Giorgos	7
Elenh	5
<div>PreviousNext</div>	

## Admin items

### Html Admin items

Η σελίδα αυτή δίνει την δυνατότητα στο διαχειριστή να δημιουργήσει ,να ενημερώσει ,να δει αλλά και να διαγράψει. Διατηρείται και πάλι το στυλ όλου του project δηλαδή υπάρχει το μενού στο πάνω μέρος .Έπειτα ακολουθούν δυο form όπου ο χρήστης μπορεί να δημιουργήσει και να ενημερώσει ένα αντικείμενο. Ακριβώς από κάτω βρίσκεται η λίστα με όλα τα αντικείμενα που υπάρχουν στη βάση αλλά και ένα κουμπί δίπλα στο κάθε αντικείμενο έτσι ώστε ο διαχειριστής να τα διαγράψει όχι μόνο από την σελίδα αλλά και από την ίδια την βάση.

Αυτές οι λειτουργίες πραγματοποιούνται μέσω του script\_adminitems.js το οποίο και γίνεται σε link μέσω του <script></script> στην html της σελίδας. Στο server γίνονται ένα get request έτσι ώστε να ζητήσουμε από την βάση τα δεδομένα ώστε να τα προβάλλουμε στη σελίδα. Στο script γίνεται το fetch και προβάλλονται σε πίνακα με καθένα από αυτά να έχει και delete button.

### Javascript and Admin items END-POINT

Στο server υπάρχει το delete όπου και γίνεται διαγραφή οποιό αντικείμενο επιλέξει ο διαχειριστής. Ακόμα γίνεται ένα post και ένα put request τα οποία είναι υπεύθυνα για τη δημιουργία αντικειμένου και την ενημέρωσή αντίστοιχα. Αυτά λειτουργούν με event listener όπου κάθε φορά που ο χρήστης συμπληρώσει την φόρμα και πατήσει το κουμπί create ή update εκτελείται η αντίστοιχη πράξη στη βάση.

Items

POIS

Log Out

### Create Items

Id:

Name:

Category:

Subcategory:

Category id:

Subcategory id:

Create Item

### Update Items

Id:

Name:

Category:

Subcategory:

Category id:

Subcategory id:

Update Item



Items						
Id	Name	Category	Subcategory	Category id	Subcategory id	
7	Pampers Πάνες Premium Care No 4 8-14 κιλ 52τεμ	Βρεφικά Είδη	Πάνες	8016e637b54241f8ad242ed1699bf2da	e0efaa1776714351a4c17a3a9d412602	Delete
13	Melissa Σιμιγδάλι Ψιλό 500γρ	Τρόφιμα	Αλεύρι - Σιμιγδάλι	ee0022e7b1b34eb2b834ea334cda52e7	c761cd8b18a246eb81fb21858ac10093	Delete
19	Κρασί Της Παρέας Λευκό 1λιτ	Ποτά - Αναψυκτικά	Κρασιά	a8ac6be68b53443bbd93b229e2f9cd34	3d01f4ce48ad422b90b50c62b1f8e7f2	Delete
26	Heineken Μπύρα 6X330ml	Ποτά - Αναψυκτικά	Μπύρες	a8ac6be68b53443bbd93b229e2f9cd34	329bdd842f9f41688a0aa017b74ffde4	Delete
27	Ava Υγρό Πιάτων Action Λευκό Ξυδι Αντλία 650ml	Καθαριότητα	Απορρυπαντικά	d41744460283406a86f8e4bd5010a66d	e60aca31a37a40db8a83ccf93bd116b1	Delete
28	Μύλοι Αγίου Γεωργίου Αλεύρι Για Πίτσα 1κιλ	Τρόφιμα	Αλεύρι - Σιμιγδάλι	ee0022e7b1b34eb2b834ea334cda52e7	c761cd8b18a246eb81fb21858ac10093	Delete
29	Ω3 Αυγά Αμερικ Γεωργ Σχολής 6τ Large 63-73γρ	Τρόφιμα	Αυγά	ee0022e7b1b34eb2b834ea334cda52e7	6d2babbc7355444ca0d27633207e4743	Delete
39	Nutricia Biskotti Ζωάκια 180γρ	Βρεφικά Είδη	Βρεφικές τροφές	8016e637b54241f8ad242ed1699bf2da	7e86994327f64e3ca967c09b5803966a	Delete
40	Γιώτης Φρουτόκρεμα 5 Φρούτα 300γρ	Βρεφικά Είδη	Βρεφικές τροφές	8016e637b54241f8ad242ed1699bf2da	7e86994327f64e3ca967c09b5803966a	Delete
61	Οπο Σκόνη 425γρ	Καθαριότητα	Απορρυπαντικά	d41744460283406a86f8e4bd5010a66d	e60aca31a37a40db8a83ccf93bd116b1	Delete
71	Γιώτης Κρέμα Παιδική Φαρίν Λακτέ 300γρ	Βρεφικά Είδη	Βρεφικές τροφές	8016e637b54241f8ad242ed1699bf2da	7e86994327f64e3ca967c09b5803966a	Delete
79	Babyino Sensitive No3 Econ 4-9κιλ 56τεμ	Βρεφικά Είδη	Πάνες	8016e637b54241f8ad242ed1699bf2da	e0efaa1776714351a4c17a3a9d412602	Delete
95	Moscato D'Asti Casarito Κρασί 750ml	Ποτά - Αναψυκτικά	Κρασιά	a8ac6be68b53443bbd93b229e2f9cd34	3d01f4ce48ad422b90b50c62b1f8e7f2	Delete

## Admin pois

Η σελίδα αυτή είναι ίδια ακριβώς με την σελίδα Admin items απλά αυτή την φορά για τα pois. Διαθέτει ιδίου τύπου requests και fetch αλλά και ίδια δομή στην εμφάνιση.

Create Pois

Id:

Location:

Name:

Type:

Create Pois

Update Pois

Id:

Shop name:

Type:

Location:

Update Pois

## Pois

Id	Shop name	Type	Coordinates	
node_354449389	Lidl	supermarket	21.712654,38.2080319	Delete
node_360217468	The Mart	supermarket	21.7806567,38.28931	Delete
node_360226900	Lidl	supermarket	21.7434265,38.2633511	Delete
node_364381224	Σουπερμάρκετ Ανδρικόπουλος	supermarket	21.7908028,38.2952086	Delete
node_364463568	Σκλαβενίτης	supermarket	21.7642075,38.2104365	Delete
node_598279836	Papakos	convenience	21.7622778,38.23553	Delete
node_633369803	undefined	convenience	21.7540236,38.2612908	Delete
node_980515550	Lidl	supermarket	21.740082,38.2312926	Delete
node_1643373636	Κρόνος	supermarket	21.7814957,38.3013087	Delete
node_1643373639	undefined	supermarket	21.790383,38.2949372	Delete
node_1643713403	undefined	convenience	21.7666723,38.2852364	Delete
node_1643713405	undefined	convenience	21.7714546,38.2911121	Delete
node_1643713406	undefined	convenience	21.7666079,38.2913332	Delete
node_1643818244	undefined	convenience	21.7625472,38.2779126	Delete

## Tokens

Η συνάρτηση `distributeTokens()` μετράει τους χρήστες και έπειτα υπολογίζει τα συνολικά tokens και τα τοποθετεί στο token-balance του κάθε χρήστη. Ακόμη δημιουργείται ένα rule το οποίο κάθε μήνα θα ενημερώνει τα tokens όπως ζητείται από την εκφώνηση.

```

356 // Create a function for distributing tokens to users
357 function distributeTokens() {
358   // Get the total number of registered users
359   Users.count()
360     .then((totalUsers) => {
361     // Calculate the total number of tokens to be created
362     let totalTokens = totalUsers * 100
363     // Reserve 20% of the tokens for future use
364     let reservedTokens = totalTokens * 0.2
365     // Distribute the remaining 80% of the tokens to users
366     let distributedTokens = totalTokens - reservedTokens
367
368     // Get the evaluation score for each user
369     Users.findAll()
370       .then((users) => {
371       // Loop through each user
372       users.forEach((user) => {
373       // Calculate the number of tokens to be distributed to the user
374       let userTokens = Math.round(distributedTokens * (user.evaluation_score / 100))
375       // Update the user's token balance in the database
376       Users.update({ token_balance: user.token_balance + userTokens }, { where: { id: user.id } })
377       })
378     })
379     .catch((err) => {
380     console.error(err)
381     })
382   })
383   .catch((err) => {
384   console.error(err)
385   })
386 }
387
388 // Set up a schedule to run the distributeTokens function at the end of each month
389 const rule = new schedule.RecurrenceRule();
390 rule.month = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11];
391 rule.date = 31;
392 rule.hour = 23;
393 rule.minute = 59;
394
395 const j = schedule.scheduleJob(rule, distributeTokens);
396

```

## Score

Αρχικά ,φτιάξαμε την συνάρτηση calculateScoreLikeDisline() οπού για κάθε offer που έχει κάνει ένας χρήστης και ανάλογα με τα like και dislike που συγκεντρώσει το κάθε offer αυτό διαμορφώνεται ανάλογα και το evaluation\_score.

```

400 // Score
401 function calculateScoreLikeDislike(){
402 // Find all the users
403
404 console.log('i am niki ');
405 Users.findAll()
406 .then(users => {
407     users.forEach(user => {
408         // Find all the offers made by the user
409         Offers.findAll({ where: { user_id: user.id } })
410         .then(offers => {
411             let likeCount = 0;
412             let dislikeCount = 0;
413             offers.forEach(offer => {
414                 likeCount += offer.like_count;
415                 dislikeCount += offer.dislike_count;
416             });
417             // Calculate the user's score
418             user.evaluation_score += likeCount * 5;
419             user.evaluation_score -= dislikeCount * 1;
420             // Save the updated user to the database
421             user.save()
422             .then(() => {
423                 console.log(`Score for user ${user.username} updated successfully`);
424             })
425             .catch(err => {
426                 console.error(err);
427             })
428         })
429     })
430 })
431
432
433 }
434
435

```

Η συνάρτηση calculateAveragePrice() αποτελείται από ένα query έτσι ώστε να συμπληρώνεται η μέση τιμή κάθε αντικειμένου σύμφωνα με τις προσφορές.

```

438
439
440 function calculateAveragePrice() {
441     sequelize.query('UPDATE items SET average_price = (SELECT AVG(offer_price) FROM offers WHERE item_id = items.id)')
442     .then(() => {
443         console.log("Average prices on items filled successfully");
444     })
445     .catch(err => {
446         console.error(err);
447     })
448 }
449
450
451

```

Αυτή η εξίσωση κάθε φορά που ο χρήστης κάνει μια προσφορά ανάλογα με την τιμή που έχει βρει παίρνει και αντίστοιχους πόντους.

```

453
454 function calculateScoreOffer(){
455
456   console.log('i am poul ');
457   Users.findAll()
458     .then(users => {
459     users.forEach(user => {
460       // Find all the offers made by the user
461       Offers.findAll({ where: { user_id: user.id } })
462       .then(offers => {
463
464
465         offers.forEach(offer => {
466           Items.findAll({ where: { id: offer.item_id } })
467           .then(item => {
468             let AveragePrice=item.average_price;
469             if (offer.offer_price < AveragePrice * 0.8){
470               user.evaluation_score += 50;
471
472             else if (offer.offer_price < AveragePrice * 0.2){
473               user.evaluation_score += 20; }
474           });
475
476           user.save()
477             .then(() => {
478               console.log('Score for user ${user.username} updated successfully');
479             })
480             .catch(err => {
481               console.error(err);
482             })
483           });
484         });
485       });
486     });
487   });
488 }
489
490

```

## Items python script

Το συγκεκριμένο script βασίζεται σε αυτό που μας δόθηκε στη εκφώνηση και αφού κατεβάσουμε τοπικά όλα τα αντικείμενα χρησιμοποιείται η python για να επιλεγούν τα 40 από αυτά. Συγκεκριμένα, χρησιμοποιούνται τα ids των subcategories και των categories και αφού τα μετατρέψουμε σε dataframes. Ακόμα, μέσω merge διαμορφώνεται το dataframe το οποίο περιέχει τα ονόματα των κατηγοριών και των υποκατηγοριών μαζί με το όνομα του αντικειμένου και τα αντίστοιχα ids. Τέλος , το dataframe αυτό μετατρέπεται σε dictionary τύπου records και έπειτα σε json .

## Prices Python script

Το συγκεκριμένο script βασίζεται σε αυτό που μας δόθηκε στη εκφώνηση για τις τιμές των προϊόντων προσαρμοσμένο στα 40 προϊόντα που προέκυψαν από το items python script.