



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

Προγραμματισμός και Συστήματα στον Παγκόσμιο Ιστό

Εργασία 2021-2022

Βουλδής Άγγελος, ΑΜ: 1059624

Γεωργία – Μαρία Φωτοπούλου, ΑΜ: 1059597

Περιεχόμενα

Εισαγωγή	2
Τεχνολογίες που χρησιμοποιήθηκαν στην παρούσα εργασία	2
Ο Σχεδιασμός Βάσης Δεδομένων (ER)	3
Αρχείο server.js	4
Screenshots του Συστήματος	5
Home page	5
About page	6
Register page	6
Login page	7
User Profile Page	9
Map & Visit Registration.....	9
Πιθανή επαφή με κρούσμα.....	11
Δήλωση κρούσματος.....	11
Επεξεργασία προφίλ	12
Admin Profile Page	14
Upload, Update, Delete Files.....	15
Statistics.....	16
Μορφή σελίδων	18

Εισαγωγή

Στην παρούσα εργασία αναπτύξαμε ένα σύστημα πληθοποριστικών δεδομένων, που αφορούν στην επισκεψιμότητα σημείων ενδιαφέροντος και τη διαχείριση πιθανής επαφής με κρούσμα COVID-19.

Στο site μας, μπορούν να συνδεθούν χρήστες και διαχειριστές οι οποίοι εκτελούν ορισμένες ενέργειες σύμφωνα με τον ρόλο τους.

Ο χρήστης μπορεί να εγγραφεί ή/και να συνδεθεί στο σύστημα και έχει τις εξής δυνατότητες:

- Μπαίνοντας στο προφίλ του μπορεί να δει την τοποθεσία του τη στιγμή της σύνδεσης μέσω ενός χάρτη
- Μπορεί να αναζητήσει σημεία ενδιαφέροντος (P.O.I.s) στον χάρτη
- Έχει τη δυνατότητα να καταχωρήσει επίσκεψή του σε συγκεκριμένο P.O.I
- Μπορεί να δηλώσει τον εαυτό του ως ενεργό κρούσμα
- Μπορεί να δει εάν έχει έρθει σε επαφή με κρούσμα
- Μπορεί να αλλάξει τα στοιχεία σύνδεσης του στο σύστημα (username/password). Εκτός από την αλλαγή στα στοιχεία του λογαριασμού μπορεί να δει και το αναλυτικό ιστορικό των επισκέψεών του.

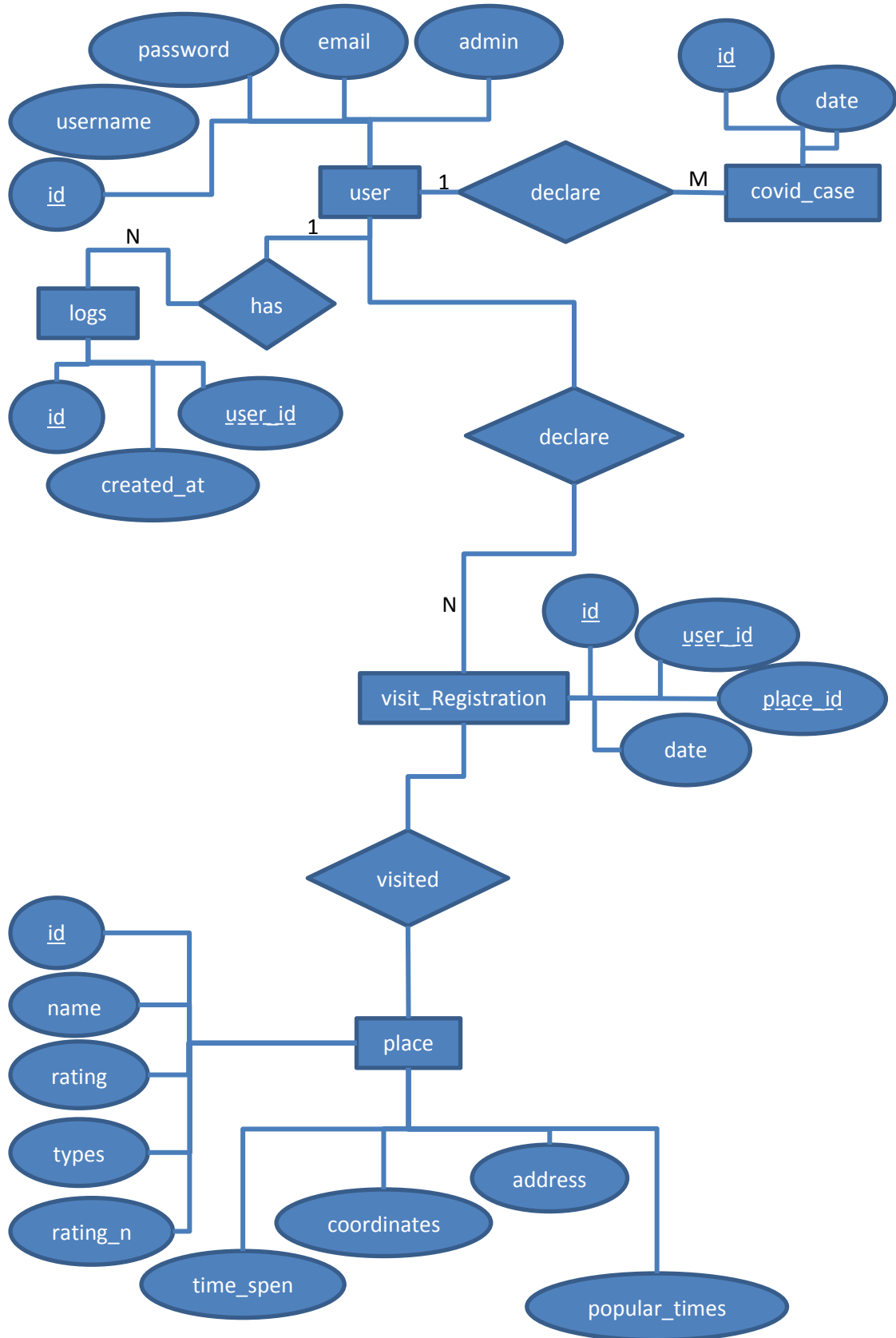
Ο διαχειριστής (τον οποίο έχουμε ορίσει εμείς στη βάση δεδομένων) συνδέεται στο σύστημα και έχει τις εξής δυνατότητες:

- Μπορεί να κάνει Upload, Update και Delete δεδομένα από JSON αρχεία
- Μπορεί να δει τα στατιστικά που αφορούν τις επισκέψεις, τις δηλώσεις κρουσμάτων και τις κατατάξεις των P.O.I.s σύμφωνα με τις ενέργειες των χρηστών.

Τεχνολογίες που χρησιμοποιήθηκαν στην παρούσα εργασία

- **HTML, CSS**
- **Javascript**
- **NodeJS, Express**
- **MongoDB**, για την δημιουργία της βάσης δεδομένων
- **Ajax server**, για την σύνδεση server με client

Ο Σχεδιασμός Βάσης Δεδομένων (ER)



Αρχείο server.js

```
if (process.env.NODE_ENV !== 'production'){
  require('dotenv').config()
}

const express = require('express')
const app = express()
const expressLayouts = require('express-ejs-layouts')
const bodyParser = require('body-parser')

const indexRouter = require('./routes/index')
const aboutRouter = require('./routes/about')
const loginRouter = require('./routes/login')
const registerRouter = require('./routes/register')

app.set('view engine', 'ejs')
app.set('views', __dirname + '/views')
app.set('layout', 'layouts/layout')
app.use(expressLayouts)
app.use(express.json())
//set static folders
app.use(express.static(__dirname + '/public'))
app.use(express.static(__dirname + '/utils'))
app.use(bodyParser.urlencoded({ limit: '10mb', extended: false }))

const mongoose = require('mongoose')
const req = require('express/lib/request')
mongoose.connect(process.env.DATABASE_URL, { useNewUrlParser: true })

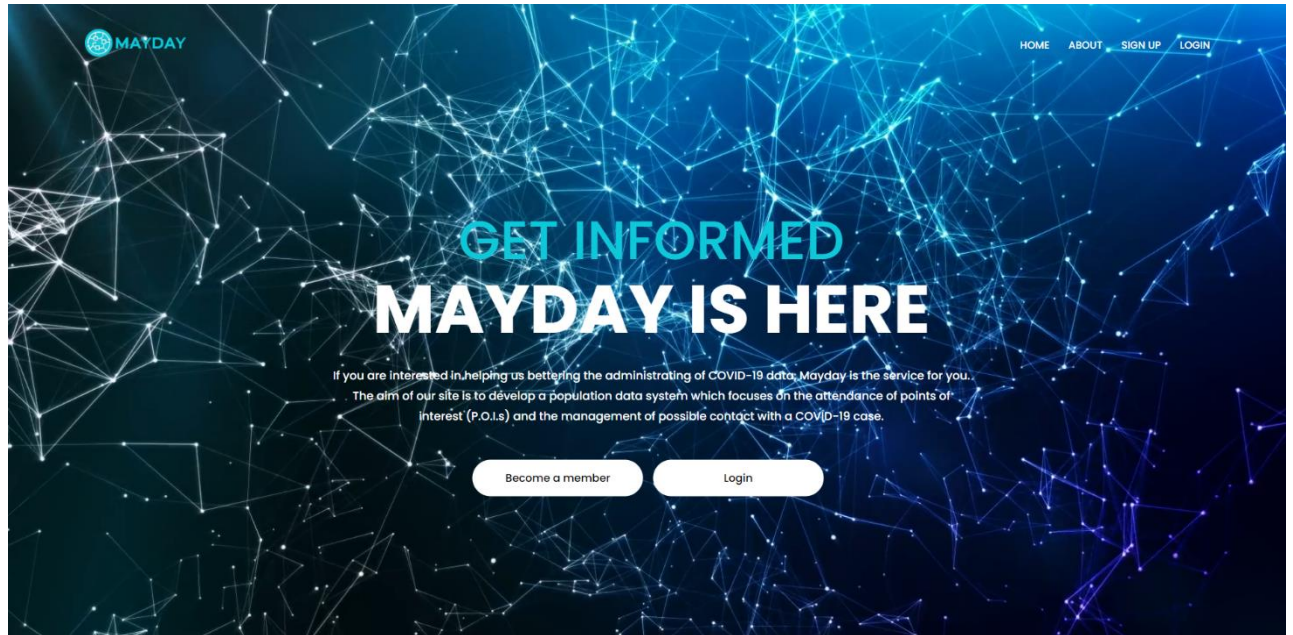
//check connection
const db = mongoose.connection
db.on('error', error => console.error(error))
db.once('open', () => console.log('Connected to Mongoose'))

app.use('/', indexRouter)
app.use('/about', aboutRouter)
app.use('/login', loginRouter)
app.use('/register', registerRouter)

app.listen(process.env.PORT || 3000)
```

Screenshots του Συστήματος

Home page



Επιτυγχάνοντας την σύνδεσή μας στο σύστημα και με τη βάση δεδομένων στο MongoDB, γράφουμε τη διεύθυνση localhost:3000 στον browser και εισερχόμαστε στην αρχική σελίδα του site. Δημιουργήθηκε ο φάκελος **routes** για την προώθηση των HTTP requests, όπου για το home page ισχύει το εξής (index.js):

```
const express = require('express')
const router = express.Router()

router.get('/', (req, res) => {
  res.render('index', {
    'page': 'index'
  })
})

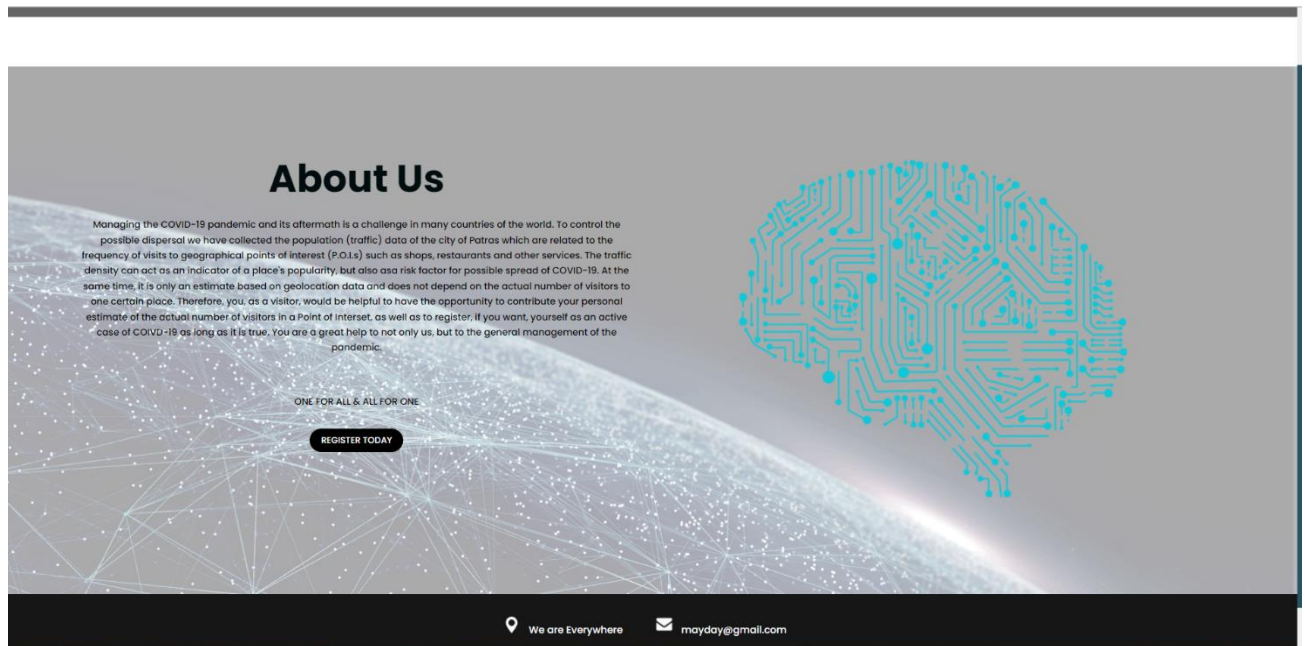
module.exports = router
```

Έπειτα, στον φάκελο **views** υπάρχει το αντίστοιχο αρχείο index.ejs με τον κώδικα της HTML για τη μορφή της σελίδας.

Η ίδια λογική ισχύει και για τη δημιουργία των υπόλοιπων pages.

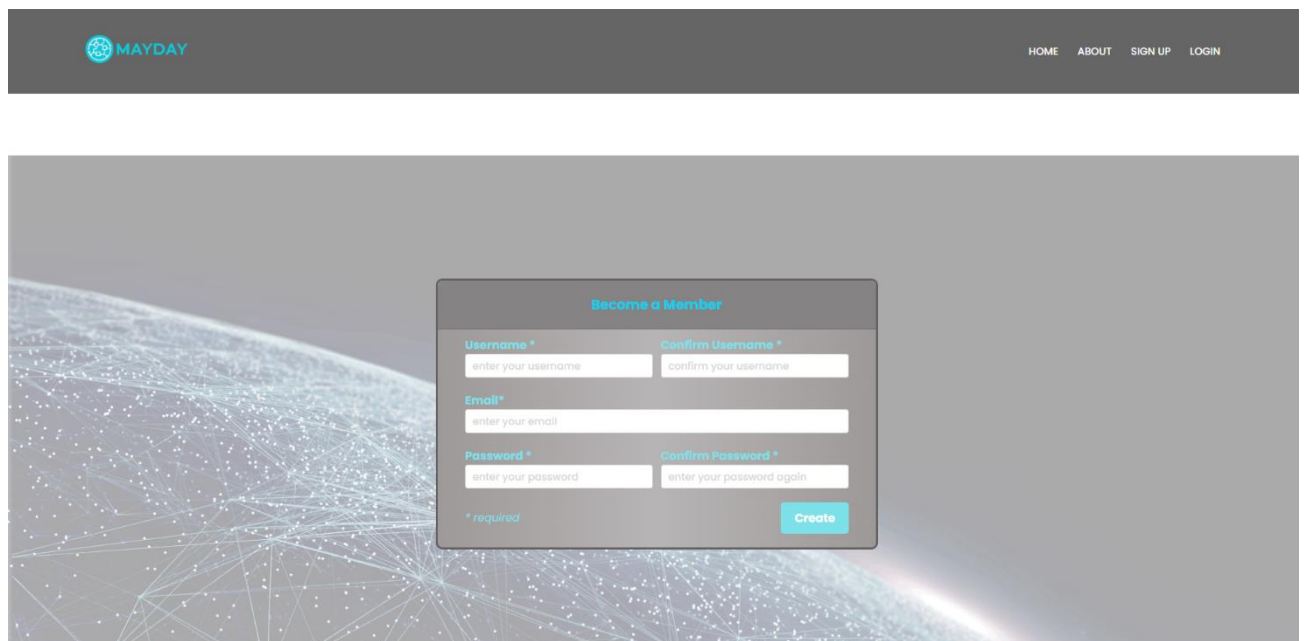
About page

Το About page με τις πληροφορίες του site πριν την σύνδεση.



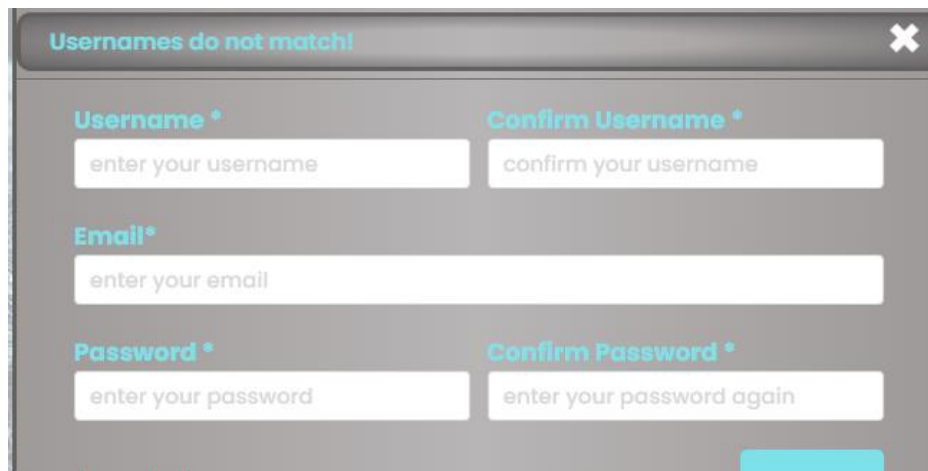
Register page

Στη συνέχεια παρατίθεται το στιγμιότυπο οθόνης από την σελίδα εγγραφής χρήστη.



Όπως φαίνεται ο εν δυνάμει χρήστης υποχρεούται να επιλέξει username, password να τα κάνει confirm και καταχωρήσει το email του.

Αν για παράδειγμα τα πεδία για το username δεν είναι ίδια θα εμφανιστεί κατάλληλο μήνυμα:



The image shows a registration form with a dark grey background. At the top, a red error message bar reads "Usernames do not match!". Below this, the form has four input fields: "Username *" (placeholder: "enter your username"), "Confirm Username *" (placeholder: "confirm your username"), "Email*" (placeholder: "enter your email"), and "Password *" (placeholder: "enter your password"). To the right of the "Password *" field is a "Confirm Password *" field (placeholder: "enter your password again"). A red "Login" button is visible at the bottom right.

Καθώς και για όλες τις υπόλοιπες περιπτώσεις:

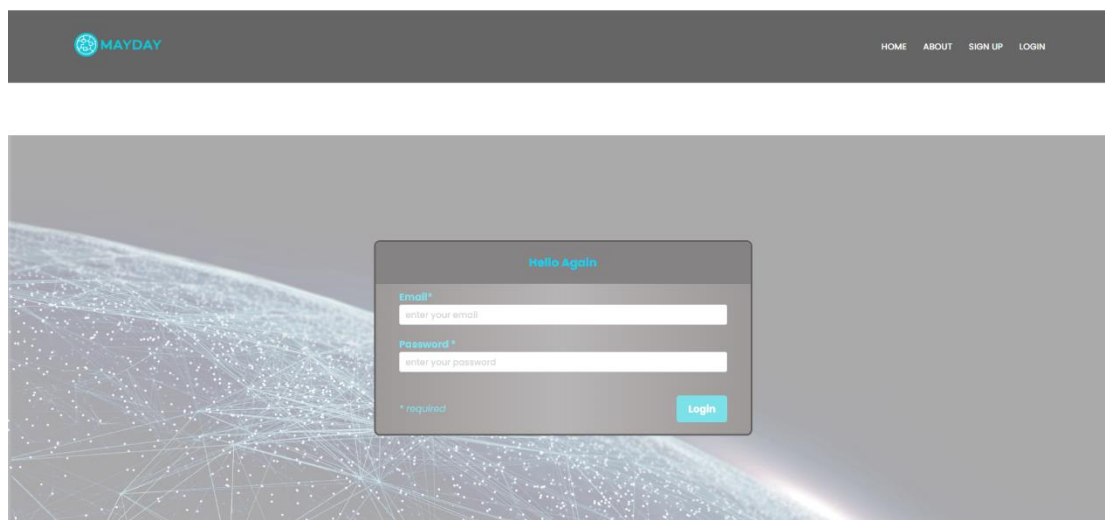


The image shows three separate red error message bars. The first bar reads "Password must have at least 1 capital letter, 1 number and 1 symbol". The second bar reads "Passwords do not match!". The third bar reads "Email is not valid". Each bar has a white 'X' icon on the right side.

Ο κώδικας στον οποίο οφείλονται τα παραπάνω είναι στο αρχείο register.js στον φάκελο routes.

Login page

Ταυτόχρονα η σελίδα της σύνδεσης χρήστη/διαχειρηστή εμφανίζεται ως:



The image shows the login page. At the top is a dark grey header with the "MAYDAY" logo on the left and navigation links "HOME", "ABOUT", "SIGN UP", and "LOGIN" on the right. Below the header is a large background image of a globe with a network of lines. In the center, there is a white login form with a red title "Hello Again". The form has two input fields: "Email*" (placeholder: "enter your email") and "Password*" (placeholder: "enter your password"). Below the "Password*" field is a small red text "* required". A red "Login" button is at the bottom right of the form.

Ο κώδικας στον οποίο οφείλεται η σελίδα του login βρίσκεται στο αρχείο login.js στον φάκελο routes. Ο διαχωρισμός του admin με τον user δηλώνεται από:

```
router.post('/', async (req, res) => {

  User.findOne({ email : req.body.email }).then( async (user) => {
    if(user){
      if (user.password === req.body.password) {
        if (user.admin){
          const loginLog = new log({
            user_id : user._id
          });

          //log the admin login
          loginLog.save((newlog) =>{
            //login admin
            res.redirect('login/admin/' + user._id)
          })

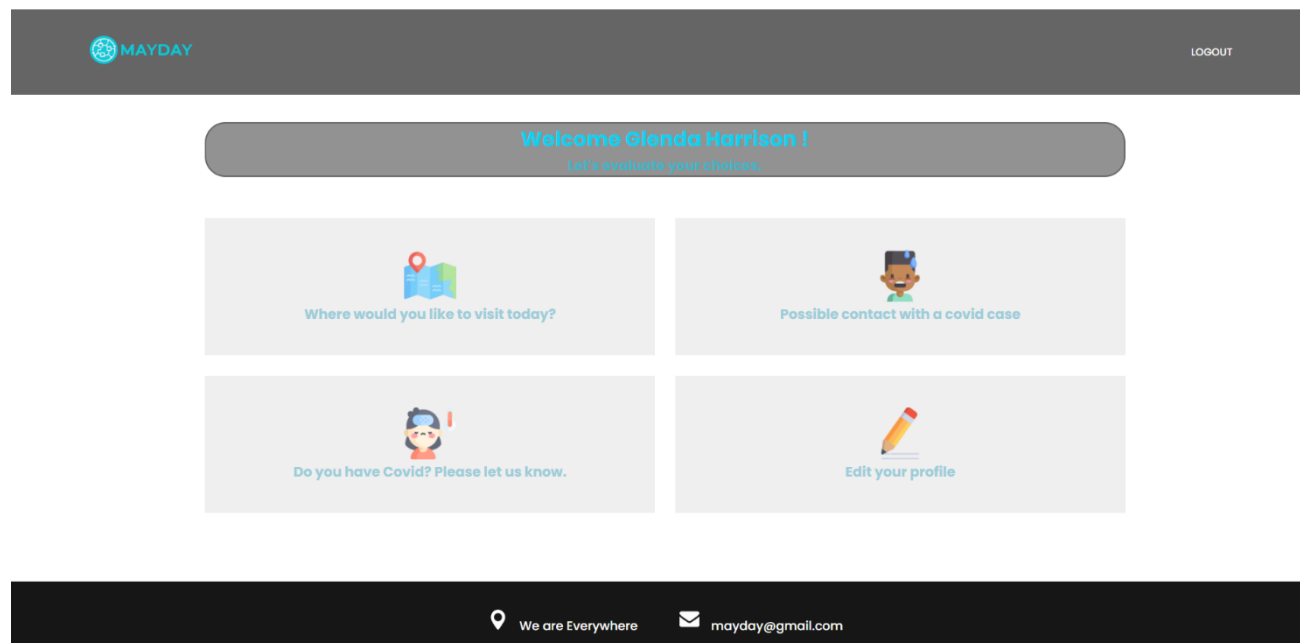
        }else{
          const loginLog = new log({
            user_id : user._id
          });

          //log the user login
          loginLog.save((newlog) =>{
            //login admin
            res.redirect('login/user/' + user._id)
          })
        }
      }
    }else{ [.....]
```

User Profile Page

Μετά την σύνδεση του user βλέπουμε το προφίλ του.

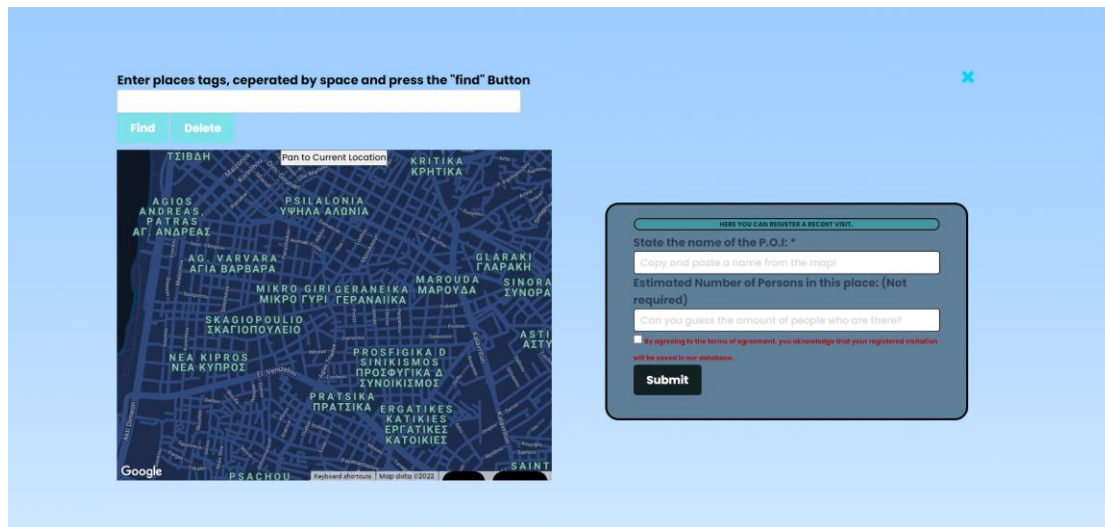
Για παράδειγμα, έχοντας δημιουργήσει έναν χρήστη στη βάση δεδομένων με username 'Glenda Harrison', κάνοντας login:



Στην κεφαλίδα της σελίδας εμφανίζεται κατάλληλο μήνυμα “Welcome <%= user.username %>” εμφανίζοντας το username του χρήστη. Στον header έχουν αφαιρεθεί τα Home & About pages (layout.ejs) και έχει προστεθεί η επιλογή Logout. Ας δούμε παρακάτω τις επιλογές του χρήστη.

Map & Visit Registration

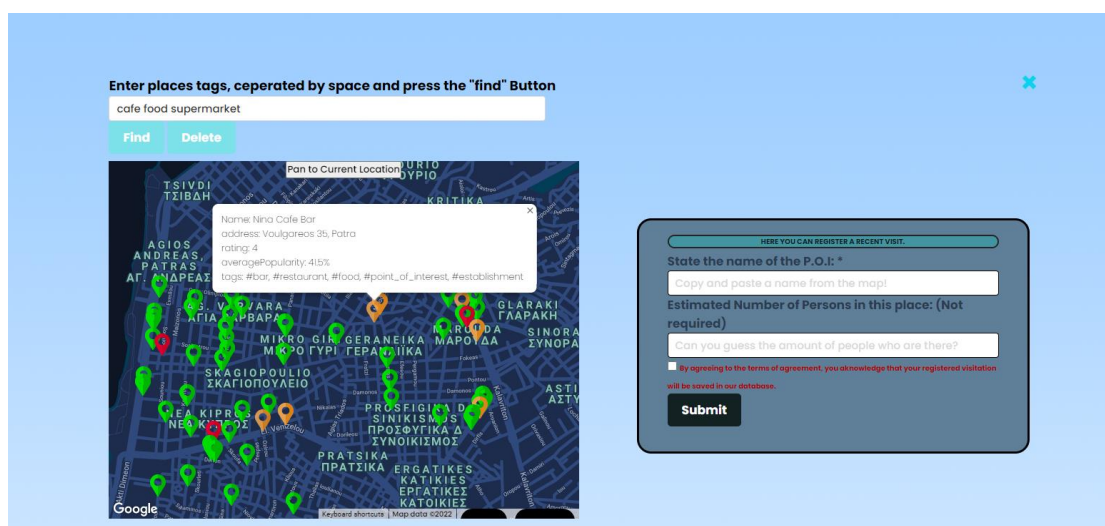
Πατώντας το πρώτο “box” με τίτλο “Where would you like to visit today?” έχουμε ορατότητα του χάρτη και πατώντας το κουμπί “pan to current location” εμφανίζεται η τοποθεσία του χρήστη, την επιλογή να κάνει εγγραφή της επίσκεψής του σε συγκεκριμένο Ρ.Ο.Ι. καθώς και, αν εκείνος επιθυμεί, να υποβάλλει στο περίπου τον αριθμό που πιστεύει ότι έχουν επισκεφθεί το μέρος που δήλωσε. Επίσης μπορεί να αναζητήσει τα σημεία ενδιαφέροντος στην περιοχή που βρίσκεται.



Ο χρήστης μπορεί να ψάξει με τα διάφορα tag από τα POIs που θέλει να επισκεφθεί και να εμφανιστούν στον χάρτη κατά αυτό τον τρόπο (έχει και την δυνατότητα να ψάξει πολλαπλά tags, αφήνοντας απλά κενό μεταξύ των tags)



Πατώντας πάνω σε κάθε marker εμφανίζεται ένα ξεχωριστό menu για κάθε POI



Αντιγράφοντας απλά το Όνομα του POI στο οποίο θέλει να κάνει visit register μπορεί με το κουμπί submit να δηλώσει ότι βρίσκεται σε αυτό το POI και ενημερώνεται με τα εξής μηνύματα:

Your visit has been registered



We could not find the place you are naming



Πιθανή επαφή με κρούσμα (Δεν λειτουργεί)

Στη συνέχεια, δίνεται η επιλογή να δει ο χρήστης εάν έχει βρεθεί σε επαφή με ενεργό κρούσμα τις τελευταίες 7 ημέρες στο box με τίτλο “Possible contact with a covid case”. Από κάτω εμφανίζεται το μέρος και η ώρα επίσκεψης.



Where would you like to visit today?



Possible contact with a covid case



Do you have Covid? Please let us know.



Edit your profile

Click Me!

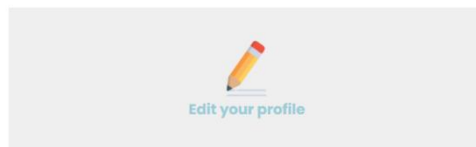
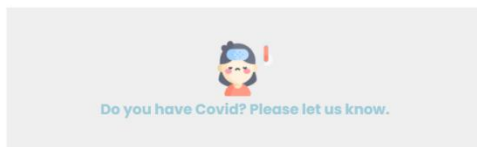
In the last 7 days you have visited these places:

Place	Time
-------	------



Δήλωση κρούσματος

Κάνοντας click στο box με τίτλο “Do you have Covid? Please let us know.” ο χρήστης μπορεί να δηλώσει εάν είναι ενεργό κρούσμα καταχωρώντας την ημερομηνία διάγνωσής του.



Επεξεργασία προφίλ

Τέλος, δίνεται η δυνατότητα επεξεργασίας του προφίλ όπου ο χρήστης μπορεί να αλλάξει username/password καθώς και να δει το αναλυτικό ιστορικό συνδέσεων και καταχώρησης κρούσματος του.

Για παράδειγμα, πατώντας το “View your login History” εμφανίζεται:

Ενώ για το ιστορικό του Covid:

**View Your Login
History**


**View Your Covid
History**


You have had the covid virus a total of 2 times

Admin Profile Page


Δημιουργήθηκαν δύο διαχειριστές στη βάση δεδομένων με την τιμή του “admin:true”.


Welcome Dear Admin Aggelos !
Let's evaluate your choices.


Upload, Update, Delete Files



Statistics


Welcome Dear Admin Georgia !
Let's evaluate your choices.


Upload, Update, Delete Files


Statistics

Ο admin έχει δύο επιλογές: να ανεβάσει, να επεξεργαστεί και να διαγράψει δεδομένα καθώς και να δει τα στατιστικά. Η συνέχεια της σελίδας:


Upload, Update, Delete Files


Statistics

**You are an Admin,
do whatever**

Click on a box and look below.



Upload, Update, Delete Files

Πατώντας το συγκεκριμένο box, εμφανίζονται τα παρακάτω.



You have the ability to load P.O.I data: ✕

Choose file

Upload Places data in Database

Choose file

Update Places data in Database

Delete Every Place data from Database

☐ Forreal, this is dangerous. Are you sure you want to delete everything?

Τα buttons Upload/Update ενεργοποιούνται μόνο στην περίπτωση που επιλεγεί ένα αρχείο για το ανέβασμα.

You have the ability to load P.O.I data: ✕

Choose file

```
{
  "_id": "CHU8yJ6neICFhL5o2KqEv0keZU",
  "name": "Moxi Patra Marina",
  "address": "Iroon Politechniou, Patra",
  "types": ["hotel", "marina"],
  "coordinates": [{"lat": 38.260716, "lng": 21.738991}],
  "rating": 3.8,
  "rating_n": 298,
  "current_popularity": 10,
}
```

Upload Places data in Database

Υπάρχουν 2 περιπτώσεις μετά το upload ενός file:

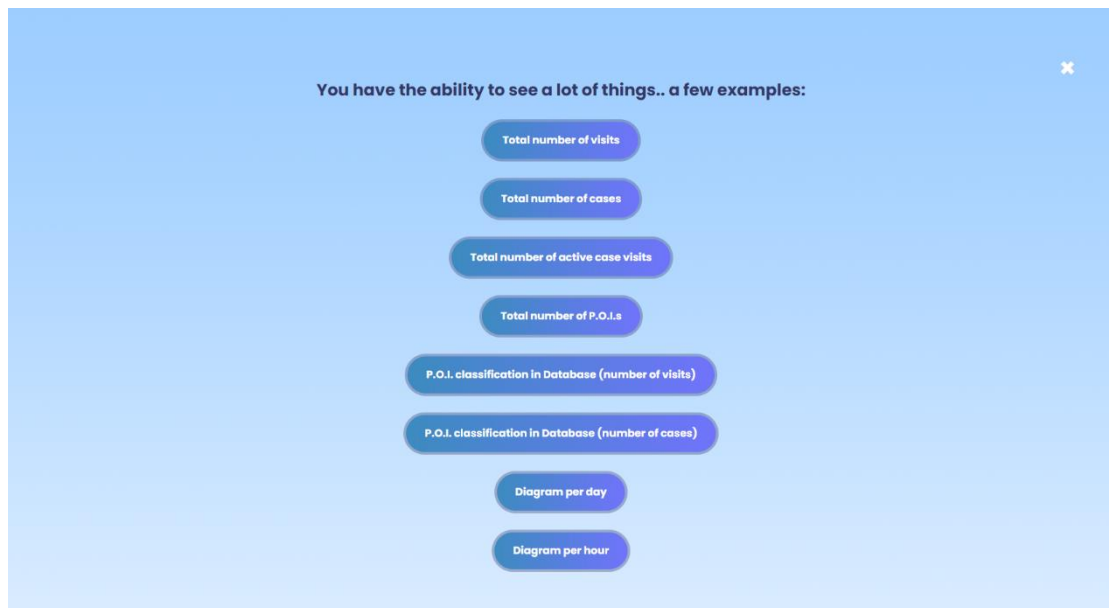
Data inserted ✕

The file you are trying to upload is not a JSON file ✕

Στην επιλογή delete, ο διαχειριστής οφείλει να είναι απολύτως βέβαιος πριν διαγράψει όλα τα δεδομένα από τη βάση!

Statistics (Δεν λειτουργούν όλα)

Εδώ εμφανίζονται τα στατιστικά.



Κάποια παραδείγματα στατιστικών:

- Συνολικός αριθμός επισκέψεων:



- Συνολικός αριθμός κρουσμάτων:



- Συνολικός αριθμός P.O.I.s:

Total number of P.O.I.s

Number of places: 161

- Μέσω array η εμφάνιση ανά κατηγορία του τύπου ενός P.O.I καθώς και ο αριθμός επισκέψεων σε αυτό:

P.O.I. classification in Database (number of visits)

P.O.I. type: bakery,food,point_of_interest,store,establishment P.O.I. Total visits: 5

Μορφή σελίδων

Τα αρχεία με την javascript για τον user και τον admin (user.js, admin.js) βρίσκονται στο path [..\public\js].

Τα “boxes” που αποτελούν τις επιλογές του κάθε χρήστη έχουν γίνει με την τεχνική dropdown. Για παράδειγμα, από το αρχείο user.js ο κώδικας για το συγκεκριμένο είναι ο εξής:

```
function openDropdown(from) {  
  const menu = document.getElementById('menu')  
  const map = document.getElementById('item-map')  
  const nervous = document.getElementById('item-nervous')  
  const patient = document.getElementById('item-patient')  
  const edit = document.getElementById('item-edit')  
  
  closeEverything(menu, map, nervous, patient, edit)  
  menu.style.display = 'block'  
  
  switch(from){  
    case 'map':  
      map.style.display = 'block'  
      break;  
    case 'nervous':  
      nervous.style.display = 'block'  
      break;  
    case 'patient':  
      patient.style.display = 'block'  
      break;  
    case 'edit':  
      edit.style.display = 'block'  
      break;  
  }  
  delay(100).then(() => menu.scrollIntoView({ behavior: 'smooth', block: 'center' }));  
}  
  
function closeEverything(menu, map, nervous, patient, edit){  
  
  menu.style.display = 'none'  
  map.style.display = 'none'  
  nervous.style.display = 'none'  
  patient.style.display = 'none'  
  edit.style.display = 'none'  
}
```

Αντίστοιχες συναρτήσεις δημιουργήθηκαν και για τον admin. Τα ονόματα των μεταβλητών χρησιμοποιήθηκαν έπειτα στο user.ejs, π.χ. για το button του χάρτη:

```
<button onclick="openDropdown('map')" class="drop_btn" method = "post"><h1>Where would you like to visit  
today?</h1></button>
```

Της κλάσης dropdown. Το animation οφείλεται στον κώδικα της css:

```
@keyframes dropdown {  
  0%{  
    transform : translateY(-500%)  
  }  
  100%{  
    transform : translateY(0%)  
  }  
}
```