



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΠΑΤΡΩΝ  
UNIVERSITY OF PATRAS

---

**Πληθοποριστικά δεδομένα**  
**επισκεψιμότητας για έλεγχο διασποράς**  
**COVID-19**

*Project - Αναφορά*

***Συγγραφείς:***

Αθανασόπουλος Παναγιώτης Κωνσταντίνος - 1058112 - 6ο Έτος

Μέντης Φίλιππος-Γεώργιος-1067466 - 4ο έτος

Μέντζι Εμανουέλο - 1071103 - 4ο έτος

***Η εργασία κατατέθηκε για το μάθημα:***

Προγραμματισμός και Συστήματα στον Παγκόσμιο Ιστό 2021-2022

# ***Περιεχόμενα***

<i>1) Εισαγωγή</i>	3
<i>2) Σχεδιασμός της βάσης</i>	4
<i>3) Υλοποίηση server</i>	7
<i>4) Υλοποίηση κώδικα για επισκέψεις χρηστών</i>	11
4.1) Αρχική Σελίδα - Authentication	11
4.2) Εγγραφή	15
4.3) Σύνδεση στην εφαρμογή	20
4.3.1) Χρήστης - Επιλογές	20
Χάρτης	22
Edit profile	25
Επαφή με κρούσμα	28
Δήλωση Κρούσματος	30
Προβολή ιστορικού επίσκεψης και νόσησης	33
4.3.2) Διαχειριστής - Επιλογές	36
<i>5) Σχεδιαστικές Επιλογές</i>	50
<i>6) Παραδοχές-Επεξηγήσεις</i>	51

## ***1) Εισαγωγή***

Το παρόν κείμενο αποτελεί την αναφορά μας για το πρότζεκτ του μαθήματος Προγραμματισμός και Συστήματα στον Παγκόσμιο Ιστό, στο οποίο κληθήκαμε να αναπτύξουμε ένα σύστημα συλλογής πληθοποριστικών δεδομένων που αφορούν την επισκεψιμότητα σημείων ενδιαφέροντος, με σκοπό τη διαχείριση πιθανής επαφής με κρούσμα Covid-19. Η υλοποίηση αυτού του έργου έγινε με την χρήση των εξής εργαλείων:

Αρχικά, χρησιμοποιήσαμε το Xampp ώστε να φιλοξενήσουμε την εφαρμογή μας στον apache server και να επιτύχουμε την επικοινωνία της εφαρμογής με την βάση.

Για την ανάπτυξη του κώδικα χρησιμοποιήθηκαν κατά κύριο λόγο τα εργαλεία webstorm, datagrid, vscode και phpmysqladmin.

Ακόμα, χρησιμοποιήθηκε η HTML για τον σχεδιασμό κάθε σελίδας, η CSS για το στυλ κάθε σελίδας, η PHP για το backend μέρος του project και η JScript για τον σχεδιασμό του χάρτη (με leaflet.js) και των γραφημάτων(με chart.js).

## ***2) Σχεδιασμός της βάσης***

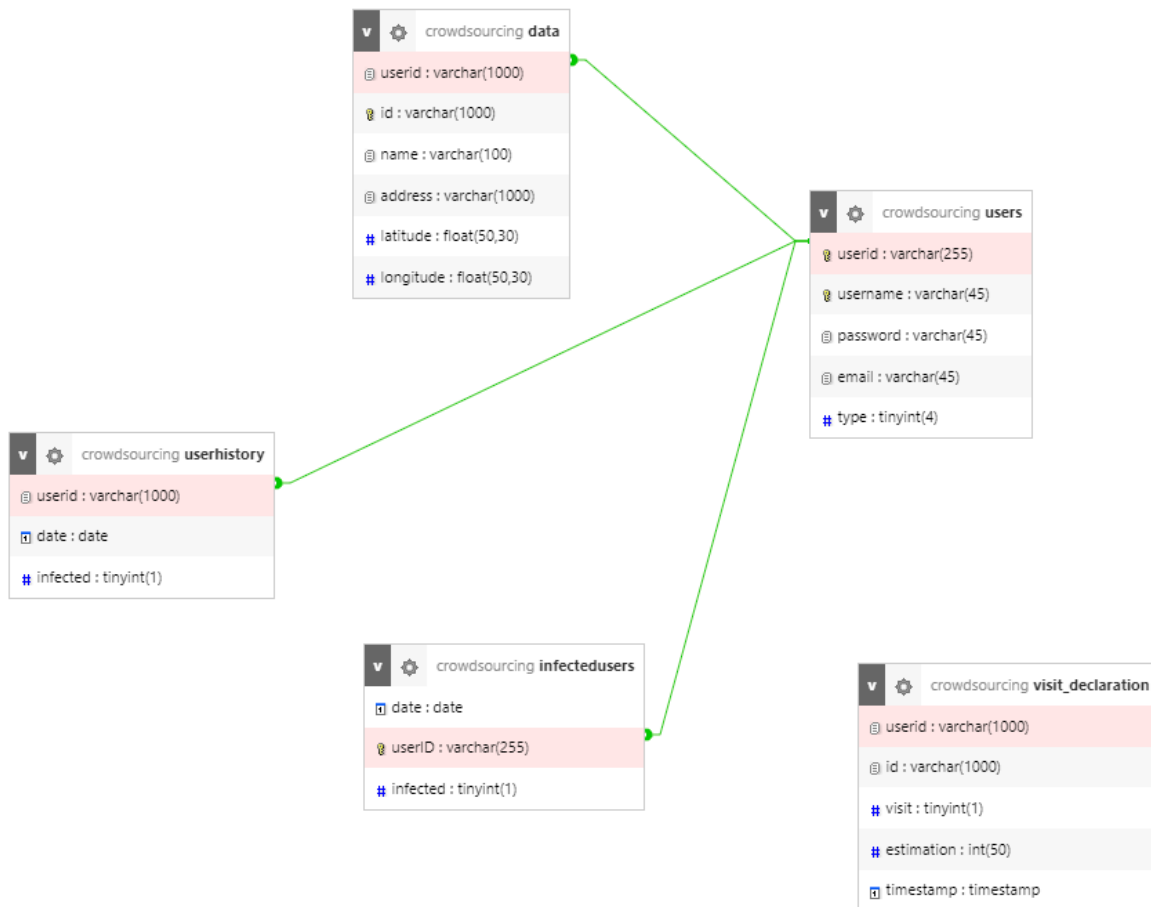


Figure 1 - DataBase ER

Για τις ανάγκες του πρότζεκτ δημιουργήσαμε μία βάση δεδομένων όπως η παραπάνω ώστε να μπορούμε να αποθηκεύουμε τις απαραίτητες πληροφορίες για τη χρηστικότητα της σελίδας.

Αρχικά για την εγγραφή ενός χρήστη στο σύστημά μας δημιουργήσαμε τον πίνακα `users` ο οποίος με κάθε εγγραφή ενός χρήστη αποθηκεύει το ονομά που επιθυμεί να έχει (`username`), τον κωδικό του (`password`), καθώς επίσης και το email του (`email`), ενώ παράλληλα δημιουργούμε και ένα μοναδικό αναγνωριστικό `id`, που χρησιμεύει ως βασικό κλειδί στον πίνακα μας. Στη συνέχεια, ήταν αναγκαία η δημιουργία ενός πίνακα για τη δήλωση κρουσμάτων του κάθε χρήστη της σελίδας. Ο

πίνακας αυτός είναι ο infectedUsers, ο οποίος αποθηκεύει το id του εκάστοτε χρήστη που έχει συνδεθεί (userID) , την ημερομηνία που αυτός ο χρήστης δήλωσε ότι διαγνώστηκε θετικός στον ιό (date), καθώς και μία στήλη που δημιουργείται για να σηματοδοτήσει αν ο χρήστης είναι ακόμα θετικός ή όχι (infected). Αυτός ο πίνακας κρατάει μόνο τους χρήστες που είναι θετικοί αυτή τη στιγμή και μετά το πέρας της νόσησης τους (max of 14 days) διαγράφονται μέσω ενός SQL Event. Συνάμα με κάθε δήλωση του χρήστη για τη νόσηση έχουμε και έναν δεύτερο πανομοιότυπο πίνακα που έχει ως στόχο να αποθηκεύει μόνιμα τα δεδομένα νόσησης όλων των χρηστών με στόχο ο κάθε χρήστης να μπορεί να βλέπει κάθε φορά όλο το ιστορικό νοσήσεων του. Αυτός ο πίνακας είναι ο userhistory. Ένας άλλος πίνακας που δημιουργήσαμε είναι ο πίνακας data που χρησιμοποιείται για την αποθήκευση των σημείων ενδιαφέροντος (POIS), ο οποίος αποθηκεύει τη διεύθυνση του μέρους ενδιαφέροντος (address), καθώς και το γεωγραφικό μήκος (longitude) και πλάτος (latitude). Επιπροσθέτως, σε αυτό τον πίνακα συγκρατούμε και πληροφορίες όπως το όνομα του POI αλλά και 2 μοναδικά id, το πρωτεύον είναι το id το οποίο υπάρχει για να διασφαλίσει τη μοναδικότητα του κάθε σημείου ενώ το δεύτερο (userID) υπάρχει καθώς όντας εξωτερικό κλειδί μας συνδέει το κάθε σημείο ενδιαφέροντος με τον χρήστη που το δημιούργησε. Τέλος έχουμε τη δημιουργία του πίνακα Visit declaration. Ο συγκεκριμένος πίνακας μας χρησιμεύει στην καταχώρηση επίσκεψης σε κάποιο σημείο ενδιαφέροντος από έναν χρήστη ο οποίος βρίσκεται σε απόσταση εντός 20 μέτρων από το σημείο αυτό. Ο πίνακας μας συγκρατεί δεδομένα για το id του χρήστη (userid), για το id της τοποθεσίας (id) και για το χρόνο που έγινε η καταχώρηση της εκάστοτε επίσκεψης (timestamp), όπως επίσης και μία εκτίμηση του κάθε χρήστη, που επιθυμεί, για το πόσα άτομα βρίσκονται στο μέρος αυτό τη δεδομένη χρονική στιγμή (estimation).

### 3) Υλοποίηση server

Στην υλοποίηση του server μας ασχοληθήκαμε κυρίως με το caching διότι παίζει έναν μεγάλο ρόλο στην αύξηση της ταχύτητας της σελίδας μας. Ορίσαμε τον χρόνο διατήρησης των δεδομένων στη μνήμη cache (ico|flv|jpg|jpeg|png|gif|css|swf και javascript) ίσο με έως 1 μήνα, τον χρόνο διατήρησης των html αρχείων σε 2 ώρες και τέλος τον χρόνο των αρχείων pdf σε μία ημέρα. Ακόμη, ορίσαμε την max-age τιμή για τα παραπάνω αρχεία στην επικεφαλίδα του πρωτοκόλλου HTTP. Η max-age εντολή δηλώνει το μέγιστο χρονικό διάστημα σε δευτερόλεπτα που επιτρέπεται να χρησιμοποιηθούν ξανά οι ληφθείσες απαντήσεις (από τη στιγμή που υποβάλλεται ένα αίτημα).

Η must-revalidate οδηγία χρησιμοποιείται για να πει σε μια κρυφή μνήμη ότι πρέπει πρώτα να επανεπικυρώσει ένα στοιχείο με την προέλευση αφού καταστεί άκυρο. Το στοιχείο δεν πρέπει να παραδοθεί στον πελάτη χωρίς να γίνει επανεπικύρωση από άκρο σε άκρο. Εν ολίγοις, τα άκυρα στοιχεία πρέπει πρώτα να επαληθεύονται και τα ληγμένα περιουσιακά στοιχεία δεν πρέπει να χρησιμοποιούνται.

Παρακάτω παραθέτουμε τον κώδικα υλοποίησης και κάποιες μετρήσεις που ολοκληρώσαμε:

```

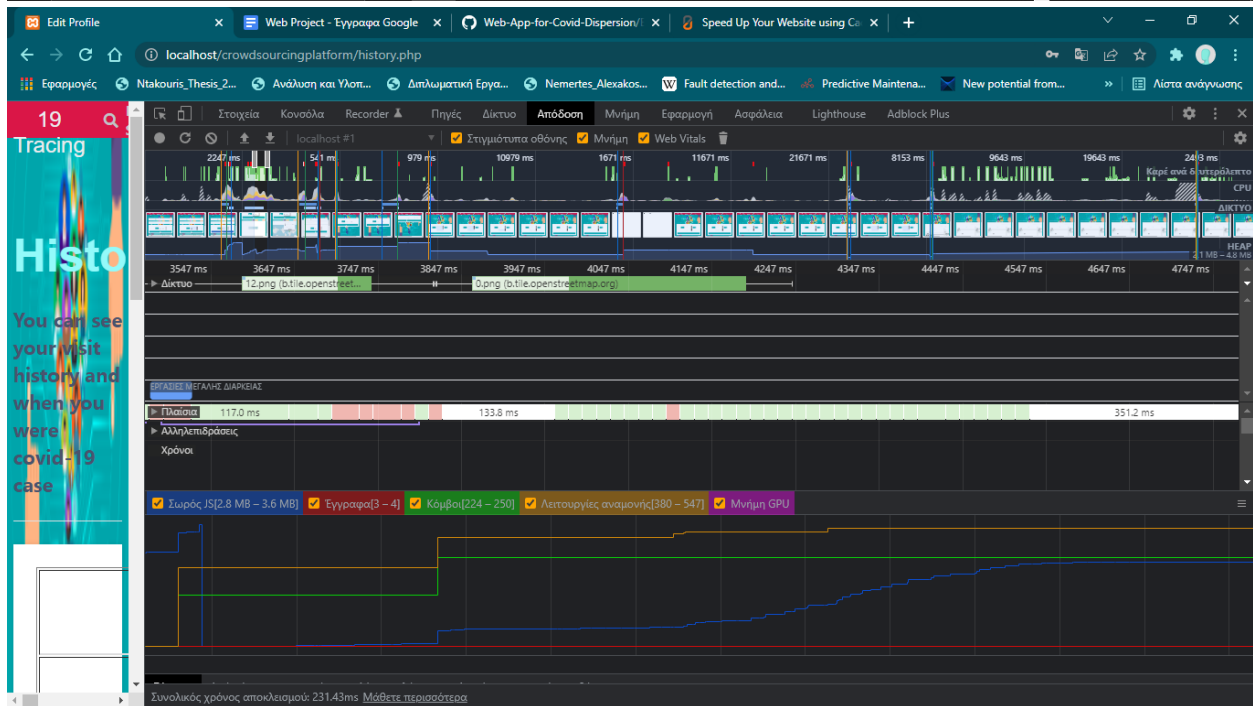
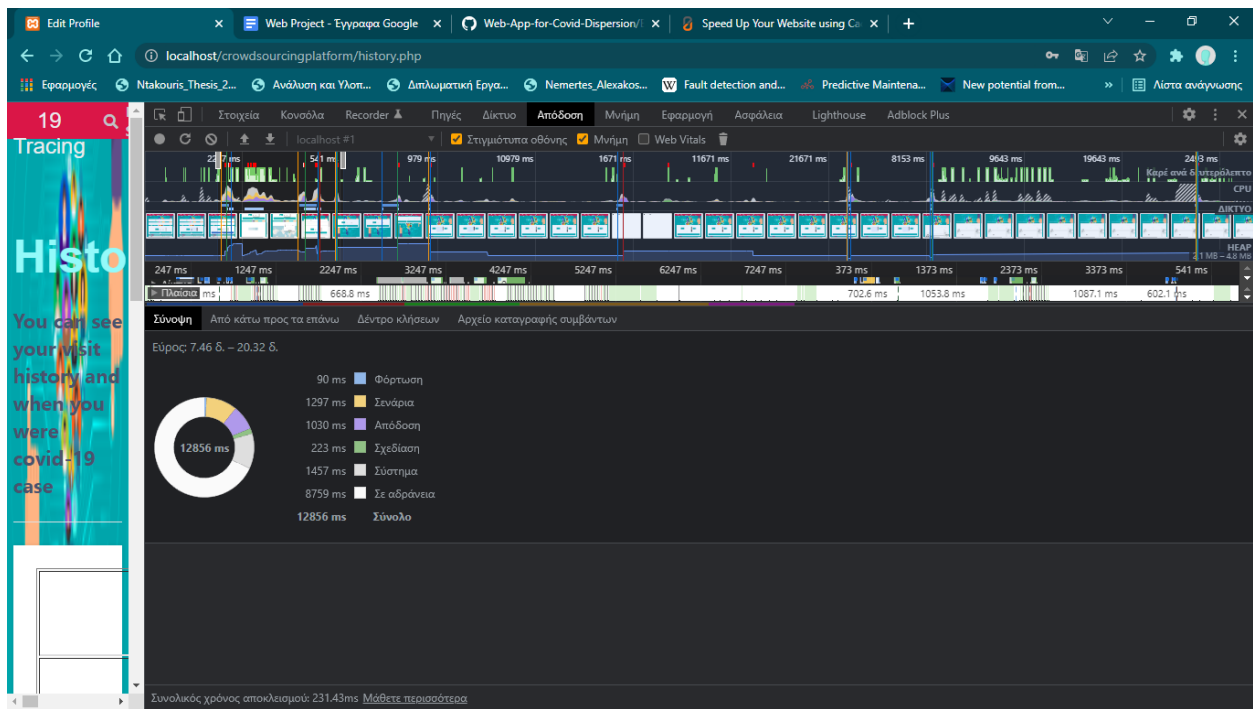
1  # BEGIN Expire headers
2  <IfModule mod_expires.c>
3      # Turn on the module.
4      ExpiresActive on
5      # Set the default expiry times.
6      ExpiresDefault "access plus 2 days"
7      ExpiresByType image/jpg "access plus 1 month"
8      ExpiresByType image/svg+xml "access 1 month"
9      ExpiresByType image/gif "access plus 1 month"
10     ExpiresByType image/jpeg "access plus 1 month"
11     ExpiresByType image/png "access plus 1 month"
12     ExpiresByType text/css "access plus 1 month"
13     ExpiresByType text/javascript "access plus 1 month"
14     ExpiresByType application/javascript "access plus 1 month"
15     ExpiresByType application/x-shockwave-flash "access plus 1 month"
16     ExpiresByType image/ico "access plus 1 month"
17     ExpiresByType image/x-icon "access plus 1 month"
18     ExpiresByType text/html "access plus 600 seconds"
19
20     <IfModule mod_headers.c>
21         Header append Cache-Control "public"
22         <FilesMatch "\.(ico|flv|jpg|jpeg|png|gif|css|swf)$">
23             Header set Cache-Control "max-age=2678400, public"
24         </FilesMatch>
25         <FilesMatch "\.(html|htm)$">
26             Header set Cache-Control "max-age=7200, private, must-revalidate"
27         </FilesMatch>
28         <FilesMatch "\.(pdf)$">
29             Header set Cache-Control "max-age=86400, public"
30         </FilesMatch>
31         <FilesMatch "\.(js)$">
32             Header set Cache-Control "max-age=2678400, private"
33         </FilesMatch>
34     </IfModule>
35
36 </IfModule>
37 # END Expire headers

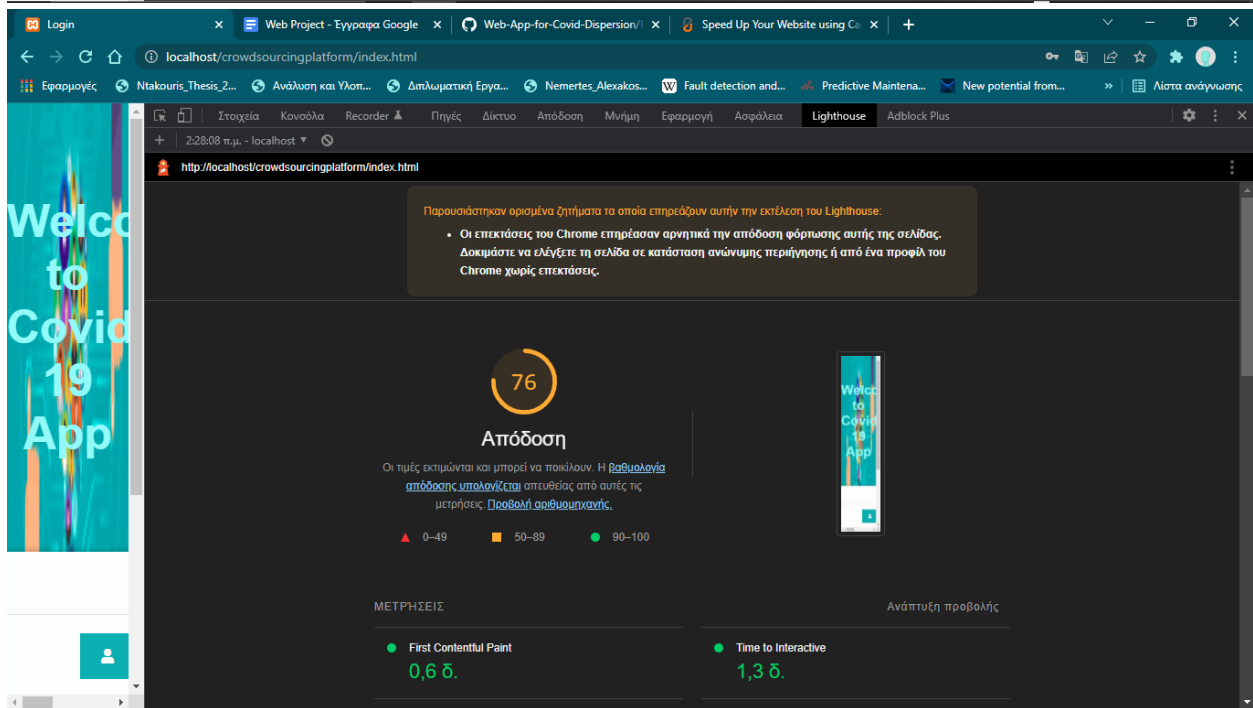
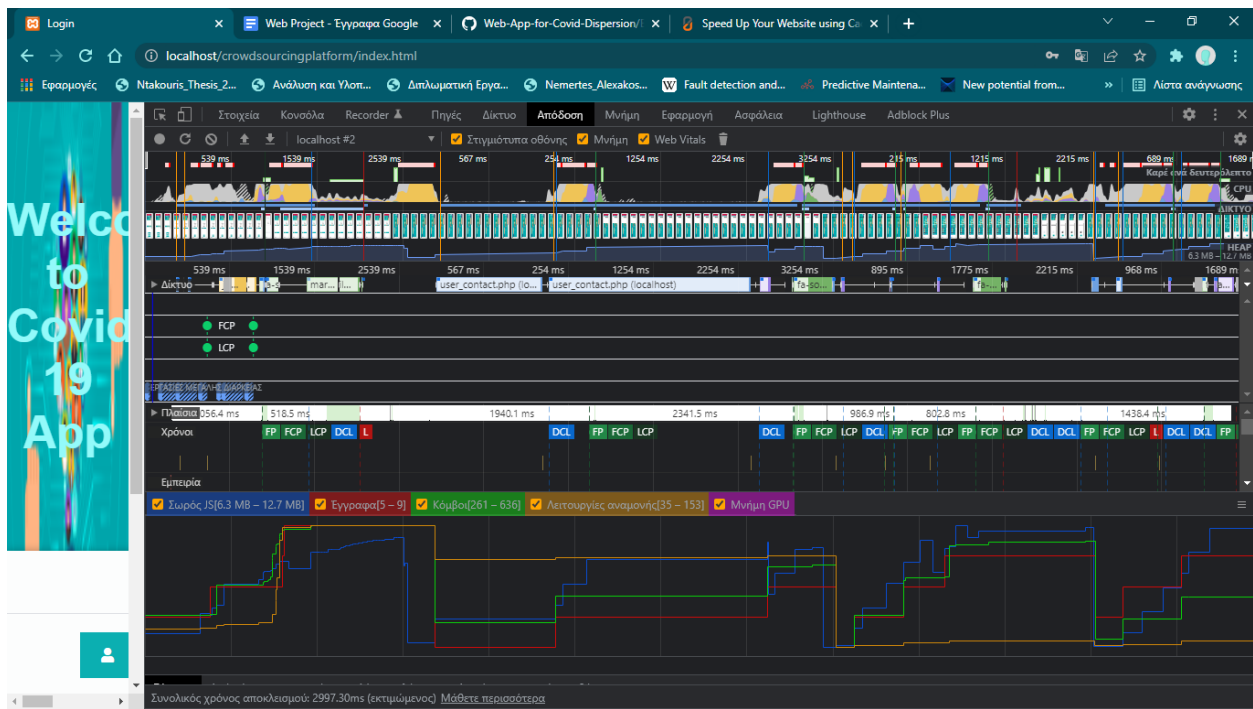
```

Figure 2.1 - .htaccess

Μετρήσεις επίδοσης στο dev tools:







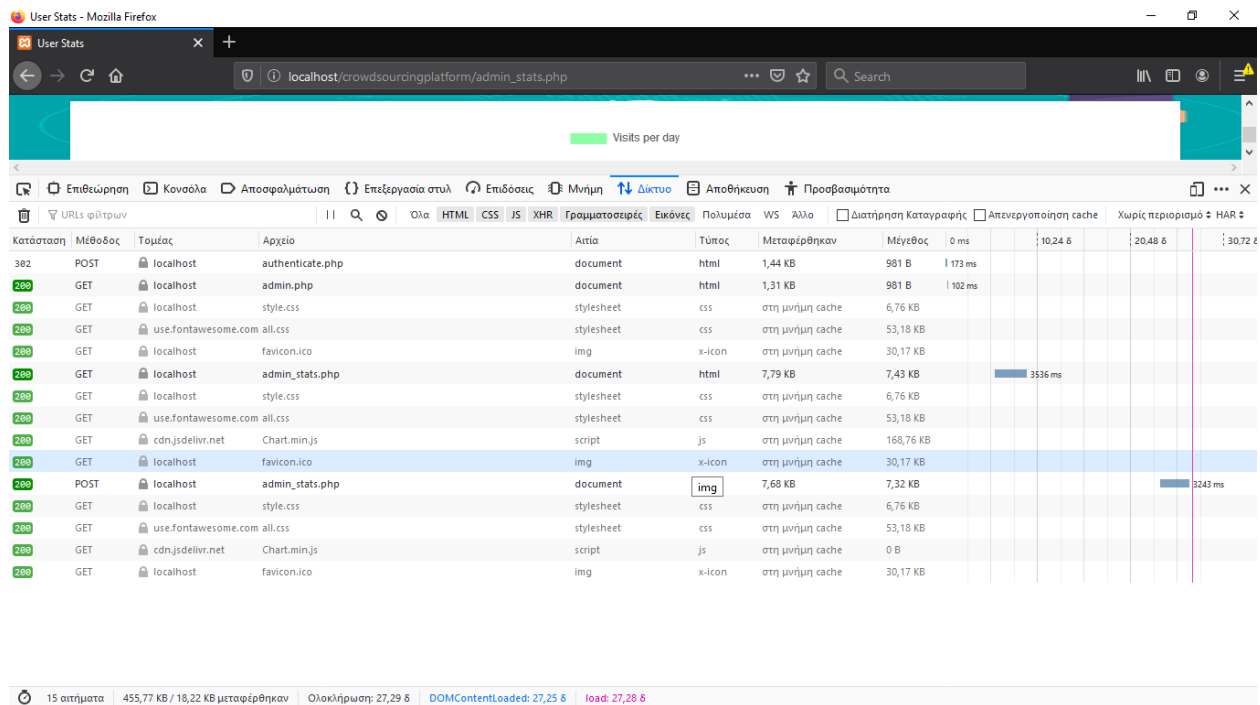


Figure 2.2 - Μετρήσεις με Dev Tools

## 4) Υλοποίηση κώδικα για επισκέψεις χρηστών

### 4.1) Αρχική Σελίδα - Authentication

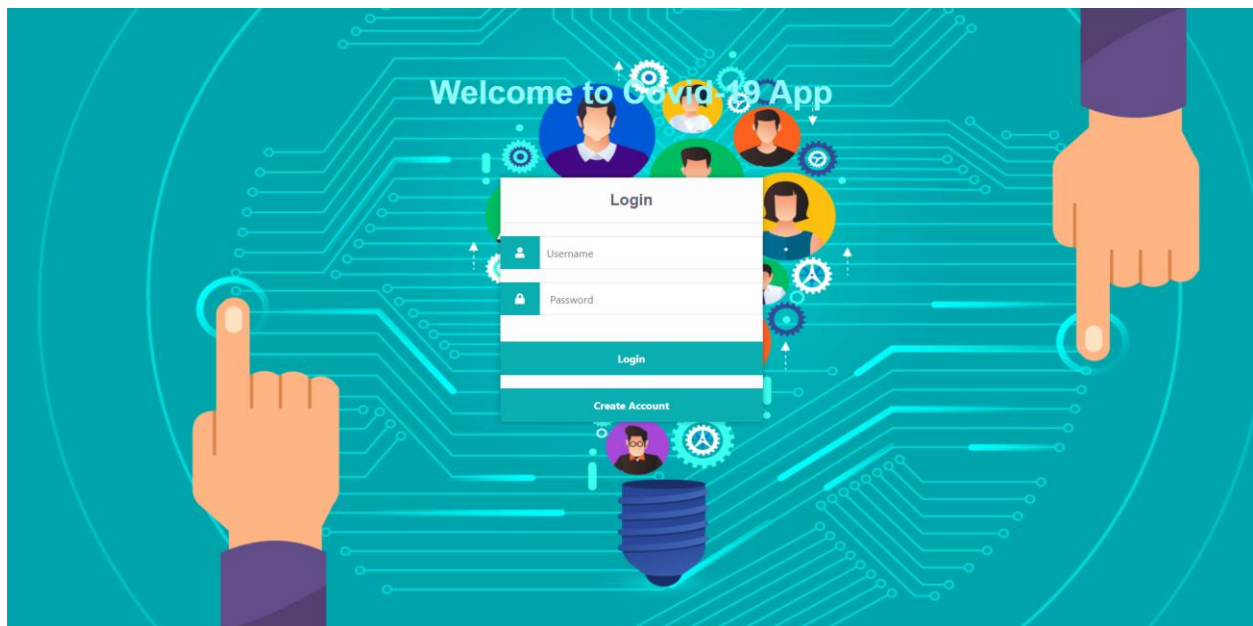
Το παρόν πρότζεκτ ξεκινάει από την αρχική σελίδα του ιστότοπου που έχουμε υλοποιήσει στο αρχείο index.html, όπου ο χρήστης μπορεί να επιλέξει να συνδεθεί αν έχει ήδη λογαριασμό ή να πραγματοποιήσει μία εγγραφή στο σύστημα. Ο κώδικας index.html παρατίθεται παρακάτω:

```

<> index.html > ...
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8">
5      <title>Login</title>
6      <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.1/css/all.css">
7      <link href="style.css" rel="stylesheet" type="text/css">
8  </head>
9  <body>
10 <h1>Welcome to Crowd-Sourcing Platform</h1>
11 <div class="login">
12     <h1>Login</h1>
13     <form action="authenticate.php" method="post">
14         <label for="username">
15             <i class="fas fa-user"></i>
16         </label>
17         <input type="text" name="username" placeholder="Username" id="username" required>
18         <label for="password">
19             <i class="fas fa-lock"></i>
20         </label>
21         <input type="password" name="password" placeholder="Password" id="password" required>
22         <input type="submit" name="login_button" value="Login">
23         <input type="button" onclick="location.href = 'register.html';" value="Create Account">
24     </form>
25 </div>
26 </body>
27 </html>
28
29

```

Και η εμφάνιση του:



Όπως παρατηρούμε τον κώδικα, βλέπουμε πως έχουμε συνδέσει τον κώδικα με τον `authenticate.php` προκειμένου να μπορέσουμε να ελέγξουμε την αυθεντικότητα των στοιχείων που εισάγει ο χρήστης. Στη συνέχεια δημιουργούμε δύο πεδία συμπλήρωσης για το ονομα και το συνθηματικό του κάθε χρήστη όπως και 2 κουμπιά, το ένα για σύνδεση με τα προαναφερθέντα στοιχεία και το δεύτερο για να μας μεταφέρει στη σελίδα εγγραφής νέου χρήστη.

Για την ομαλή λειτουργία της σελίδας και τη μοναδικότητα του κάθε χρήστη υλοποιήθηκε ένας κώδικας (`authenticate.php`), ο οποίος φαίνεται παρακάτω:

```

authenticate.php > ...
1  <?php
2
3  // Include the database
4  require('config.php');
5
6  // Start the session
7  session_start();
8
9  // Check if the data from the login form was submitted, isset() will check if the data exists
10 if (!isset($_POST['username'], $_POST['password'])) { die ('Please fill the username and password fields!'); }
11
12 // Prepare the SQL statement
13 if ($stmt = $con->prepare('SELECT userid, password, type FROM users WHERE username = ?')) {
14
15     // Bind the parameters and execute the query
16     $stmt->bind_param('s', $_POST['username']);
17     $stmt->execute();
18     $stmt->store_result();
19
20     // Check if the username exists in the database
21     if ($stmt->num_rows > 0) {
22
23         // Load the 'userid', 'password' and 'type' of the account
24         $stmt->bind_result($id, $password, $type);
25         $stmt->fetch();
26         // Terminate the SQL statement
27         $stmt->close();
28
29         // Verify the password
30         if ($_POST['password'] === $password) {
31
32             // Remember the login information for this session
33             session_regenerate_id();
34             $_SESSION['loggedin'] = TRUE;
35             $_SESSION['name'] = $_POST['username'];
36             $_SESSION['id'] = $id;
37
38             // $con->close();
39
40             // Go to the appropriate page based on the type of the account
41             if ($type == 0) {
42                 // Go to the admin dashboard
43                 header('Location: admin.php');
44             } else {
45                 // Go to the user dashboard
46                 header('Location: user.php');
47             }
48
49         } else {
50             die('Incorrect password!');
51         }
52     } else {
53         die('Incorrect username!');
54     }
55 }
56
57
58 ?>

```

Στον συγκεκριμένο κώδικα ξεκινάμε με σύνδεση στη βάση δεδομένων που το πετυχαίνουμε με την εντολή `require('config.php');` του οποίου ο κώδικας φαίνεται εδώ:

```

config.php > ...
1  <?php
2
3  // Database elements
4  $DATABASE_HOST = 'localhost';
5  $DATABASE_USER = 'root';
6  $DATABASE_PASS = '';
7  $DATABASE_NAME = 'crowdsourcing';
8
9  // connection
10 $con = mysqli_connect($DATABASE_HOST, $DATABASE_USER, $DATABASE_PASS, $DATABASE_NAME);
11 mysqli_set_charset($con,"utf8");
12 // If there is an error with the connection, stop the script and display the error.
13 if (mysqli_connect_errno()) { die ('Failed to connect to MySQL: ' . mysqli_connect_error()); }
14

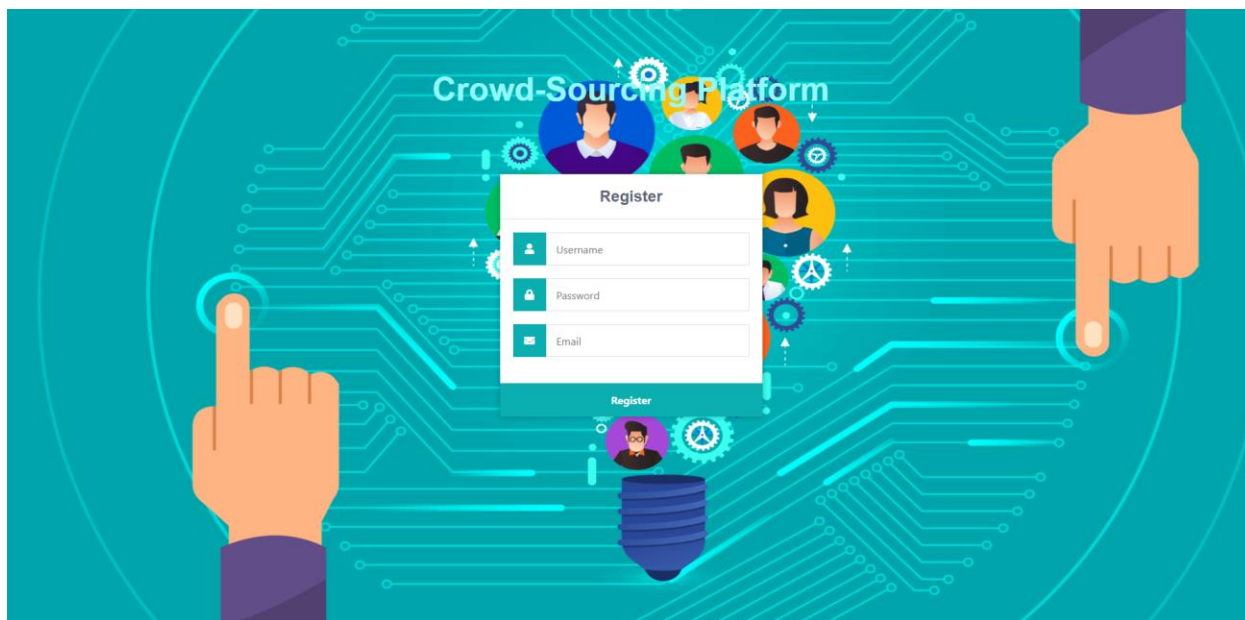
```

[Λόγος υλοποίησης της είναι να μην επαναλαμβάνουμε έναν τετριμμένο κώδικα για τη σύνδεση στη βάση δεδομένων μας.]

Στη συνέχεια ελέγχουμε αν τα δεδομένα που έχει εισάγει ο χρήστης στα πλαίσια ονόματος χρήστη και συνθηματικό έχουν υποβληθεί στο σύστημα μας (δηλαδή αν ο χρήστης έχει συμπληρώσει και τα 2 κενά) και έπειτα ελέγχουμε αν υπάρχουν στη βάση δεδομένων μας . Μετά κάνουμε τη συσχέτιση των εισαχθέντων στοιχείων με αυτά που υπάρχουν στη βάση μας και ανάλογα τον τύπο των δεδομένων έχουμε προγραμματίσει τις κατάλληλες ενέργειες μέσω εμφωλευμένων if statements. Αν έχουμε εισάγει σωστά στοιχεία και είμαστε χρήστης της ιστοσελίδας τότε το σύστημα μας παραπέμπει στην ανάλογη σελίδα ενώ αν είμαστε ο admin μεταφερόμαστε σε μία διαφορετική. Σε περίπτωση που δεν έχουμε βάλει σωστό όνομα χρήστη ή κωδικό θα μας εμφανιστεί το ανάλογο μήνυμα από το σύστημα .

## 4.2) Εγγραφή

Η εγγραφή του χρήστη στο σύστημα γίνεται μέσω των αρχείων register.php και της register.html. Παρακάτω υπάρχει αναλυτική περιγραφή για τα δύο αρχεία κώδικα.



Αρχικά έχουμε την register.html, η οποία αποτελεί το περιβάλλον χρήστη για την εγγραφή το στο σύστημα της οποίας ο κώδικας βρίσκεται παρακάτω:



```

<> register.html > ...
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8">
5      <title>Register</title>
6      <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.1/css/all.css">
7      <link href="style.css" rel="stylesheet" type="text/css">
8  </head>
9  <body>
10     <h1>Crowd-Sourcing Platform</h1>
11     <div class="register">
12         <h1>Register</h1>
13         <form action="register.php" method="post" autocomplete="off">
14             <label for="username">
15                 <i class="fas fa-user"></i>
16             </label>
17             <input type="text" name="username" placeholder="Username" id="username" required>
18             <label for="password">
19                 <i class="fas fa-lock"></i>
20             </label>
21             <input type="password" name="password" placeholder="Password" id="password" required>
22             <label for="email">
23                 <i class="fas fa-envelope"></i>
24             </label>
25             <input type="email" name="email" placeholder="Email" id="email" required>
26             <input type="submit" value="Register">
27         </form>
28     </div>
29 </body>
30 </html>

```

Η register.html αποτελείται από τρία πεδία για την προσθήκη του ονόματος χρήστη τον κωδικό που επιθυμεί να έχει στο σύστημα μας και το email του χρήστη, όπως επίσης και ένα κουμπί για την υποβολή των δεδομένων στο σύστημα. Επίσης συνδέεται με τον κώδικα register.php για τον έλεγχο των δεδομένων που εισάγονται και την αποθήκευσή τους στη βάση. Ο κώδικάς της παρατίθεται εδώ:

register.php > ...

```
1  <?php
2
3  ini_set('display_errors', 1);
4  ini_set('display_startup_errors', 1);
5  error_reporting(E_ALL);
6
7  // connection info.
8  $DATABASE_HOST = 'localhost';
9  $DATABASE_USER = 'root';
10 $DATABASE_PASS = '';
11 $DATABASE_NAME = 'crowdsourcing';
12
13 // Try and connect using the info above.
14 $con = mysqli_connect($DATABASE_HOST, $DATABASE_USER, $DATABASE_PASS, $DATABASE_NAME);
15 mysqli_set_charset($con,"utf8");
16 if (mysqli_connect_errno()) {
17     // If there is an error with the connection, stop the script and display the error.
18     die ('Failed to connect to MySQL: ' . mysqli_connect_error());
19 }
20
21 // check if the data was submitted, isset() function will check if the data exists.
22 if (!isset($_POST['username'], $_POST['password'], $_POST['email'])) {
23     // Could not get the data that should have been sent.
24     die ('Please complete the registration form!');
25 }
26 // Make sure the submitted registration values are not empty and valid.
27 if (empty($_POST['username']) || empty($_POST['password']) || empty($_POST['email'])) {
28     // One or more values are empty.
29     die ('Please complete the registration form!');
30 }
31 if (!filter_var($_POST['email'], FILTER_VALIDATE_EMAIL)) {
32     die ('Email is not valid!');
33 }
34 if (preg_match('/[A-Za-z0-9]+/', $_POST['username']) == 0) {
35     die ('Username is not valid!');
36 }
37 $uppercase = preg_match('@[A-Z]@', $_POST['password']);
38 $number = preg_match('@[0-9]@', $_POST['password']);
39 $symbols = "!@#%^&*()_+~{}[]|\?><.,/:'";
40 $symbol = strpbrk($_POST['password'], $symbols);
41 if (!$uppercase) { die ('Password must contain at least one uppercase letter!'); }
42 if (!$number) { die ('Password must contain at least one number!'); }
43 if (!$symbol) { die ('Password must contain at least one symbol!'); }
44 if (strlen($_POST['password']) < 8) { die ('Password must be at least 8 characters long!'); }
45
46 // check if the account with that username exists.
47 if ($stmt = $con->prepare('SELECT userid, password FROM users WHERE username = ? OR email = ?')) {
48     // Bind parameters (s = string, i = int, b = blob, etc), hash the password using the PHP password_hash function.
49     $stmt->bind_param('ss', $_POST['username'], $_POST['email']);
50     $stmt->execute();
51     $stmt->store_result();
52     $exists = $stmt->num_rows > 0;
53     $stmt->close();
54     // Store the result so we can check if the account exists in the database.
55     if ($exists) {
56         // Username already exists
57         echo 'Username or email already exists, please choose another!';
58     }
59     else {
```

```

59     else {
60         // Insert new account
61         // Username doesnt exists, insert new account
62         if ($stmt = $con->prepare('INSERT INTO users ( userid, username, password, email) VALUES (?, ?, ?, ?)')) {
63             // We do not want to expose passwords in our database, so hash the password and use password_verify when a user logs in.
64             $password = md5($_POST['password']);
65             $cipher = "aes-128-gcm";
66             $iv = "random_bytes";
67             $userid = openssl_encrypt($_POST['email'], $cipher, $_POST['password'], $options=0, $iv, $tag);
68             $stmt->bind_param('ssss', $userid, $_POST['username'], $password, $_POST['email']);
69             $stmt->execute();
70             $stmt->close();
71             echo 'You have successfully registered, you can now login!';
72             header('Location: index.html');
73             die();
74         } else {
75             // Something is wrong with the sql statement, check to make sure accounts table exists with all 3 fields.
76             echo 'Could not prepare statement!';
77         }
78     }
79 }
80 else {
81     // Something is wrong with the sql statement, check to make sure accounts table exists with all 3 fields.
82     echo 'Could not prepare statement!';
83 }
84 $con->close();
85 >>

```

Με τη σειρά της λοιπόν η register.php συνδέεται στη βάση δεδομένων μας και έπειτα ελέγχει αν έχει συμπληρωθεί η φόρμα εγγραφής από τον χρήστη της σελίδας και έχει υποβληθεί ορθά. Στη συνέχεια εξετάζει αν το email είναι ορθά γραμμένο, αν το username του χρήστη έχει χαρακτήρες εκτός των επιτρεπόμενων και αν ο κωδικός σύνδεσης που ορίζει ο χρήστης εμπεριέχει τις προϋποθέσεις που του ορίζουμε (τουλάχιστον έναν κεφαλαίο χαρακτήρα, τουλάχιστον έναν αριθμό, τουλάχιστον ένα σύμβολο και να έχει μέγεθος άνω των 8 χαρακτήρων). Αφού γίνει αυτός ο έλεγχος έχουμε έναν δεύτερο έλεγχο στη περίπτωση που έχουμε πανομοιότυπα email ή ονόματα και στη περίπτωση που υπάρχουν στη βάση τότε αναφέρουμε στο χρήστη ότι ο λογαριασμός με αυτά τα στοιχεία υπάρχει ήδη.

Τέλος εφόσον ο χρήστης έχει βάλει σωστά τα στοιχεία του και δεν έχει ήδη λογαριασμό δημιουργούμε τον καινούριο λογαριασμό εισάγοντας στο table users.sql. Αφού συμβεί αυτό το σύστημα μας αναφέρει ότι κάναμε εγγραφή με επιτυχία και μας μεταφέρει στη σελίδα σύνδεσης (index.html) προκειμένου να συνδεθούμε κανονικά σε οποιαδήποτε

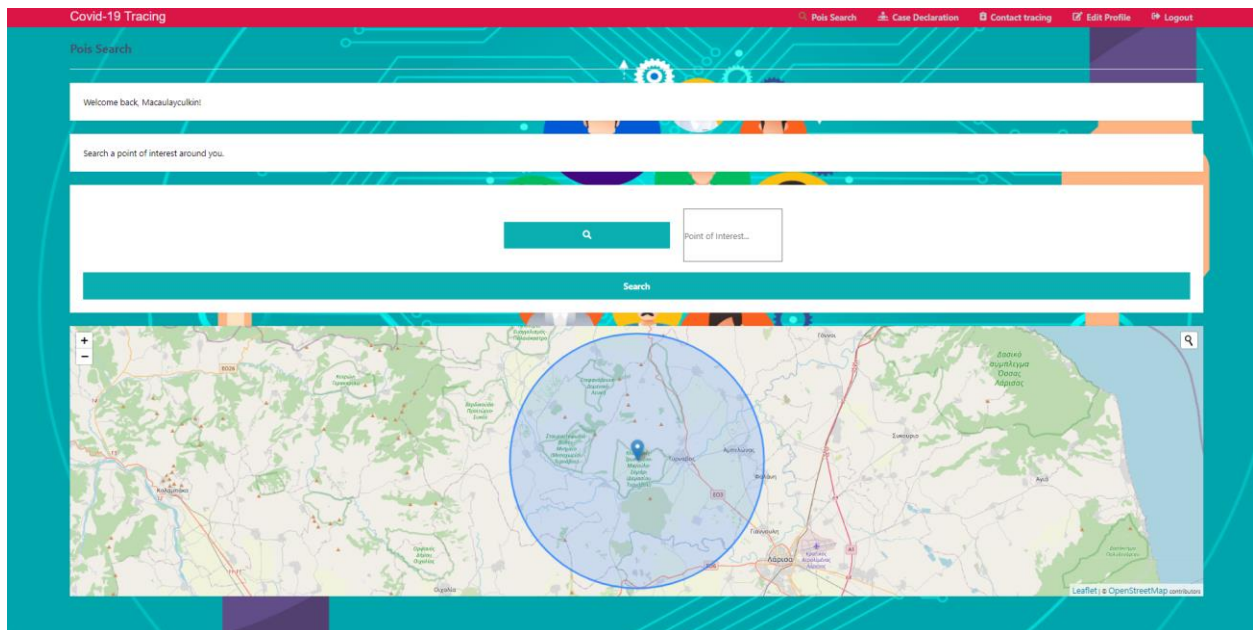
άλλη περίπτωση που το σύστημά μας δε λειτουργήσει έχουμε το μήνυμα ότι δεν μπόρεσε να εκτελεστεί σωστά η δήλωση του χρήστη.

### 4.3) Σύνδεση στην εφαρμογή

Πλέον έχουμε συνδεθεί στο σύστημα και μεταφερόμαστε στην σελίδα user.php ή admin.php ανάλογα το άτομο που συνδέεται στο σύστημα και την εκτέλεση του αρχείου authenticate.php.

#### 4.3.1) Χρήστης - Επιλογές

Αρχικά υποθέτουμε πως ο χρήστης μας **δεν είναι ο διαχειριστής** του συστήματός μας και άρα τον μεταφέρουμε στο αρχείο user.php του οποίου ο κώδικας βρίσκεται εδώ :



```

21 <!DOCTYPE html>
22 <html>
23 <head>
24 <meta charset="utf-8">
25 <title>User Dashboard</title>
26 <link href="style.css" rel="stylesheet" type="text/css">
27 <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.1/css/all.css">
28
29 <link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css" integrity="sha512-xod2BNT5n17Tt2at1PuE1HxjVMSvLW9ocqJLsCCSCXdbqCmb1Ash0MAS6/keqq/sM2M219scR4PsZChSR7A=="
30 crossorigin="" />
31 <link rel="stylesheet" href="https://unpkg.com/leaflet-control-geocoder/dist/Control.Geocoder.css" />
32
33 </head>
34 <body class="loggedin">
35 <nav class="navtop">
36 <div>
37 <h1>Covid-19 Tracing</h1>
38
39 <!-- Navigation Bar -->
40 <a href="user.php"><i class="selected fas fa-search"></i>Pois Search</a>
41
42 <a href="instance_declaration.html"><i class="fas fa-diagnoses"></i>Case Declaration</a>
43 <a href="user_contact.php"><i class="fas fa-notes-medical"></i>Contact tracing</a>
44 <a href="editprofile.html"><i class="fas fa-edit"></i>Edit Profile</a>
45 <a href="logout.php"><i class="fas fa-sign-out-alt"></i>Logout</a>
46 </div>
47 </nav>
48 <div class="content">
49 <h2>Pois Search</h2>
50 <p>Welcome back, <?=$_SESSION['name']>!</p>
51
52 <p>Search a point of interest around you.</p>
53
54 <div>
55 <form action="search.php" method="post" enctype="multipart/form-data">
56 <label for="username"><i class="fas fa-search"></i></label>
57 <input type="text" name="name" placeholder="Point of Interest..." id="name" required>
58 <input type="submit" name="submit" value="Search">
59 </form>
60 </div>

```

Στόχος του user.php είναι να αποτελεί το βασικό περιβάλλον του χρήστη μας στο οποίο, ύστερα από την αποδοχή ενός cookie επίτρεψης τοποθεσίας, θα εμφανίζεται η θέση του στον χάρτη της σελίδας, όπου επίσης υπάρχουν και τα γειτονικά σημεία ενδιαφέροντος (POIS). Μπορεί επίσης να αναζητήσει ένα σημείο ενδιαφέροντος στον χάρτη (μέσω του search.php) ή να πατήσει κάποια από τις άλλες επιλογές που διαθέτει η μπάρα πλοήγησης που βρίσκεται στο πάνω μέρος της σελίδας μας. Αρχικά συνδεόμαστε στη βάση δεδομένων με την προσθήκη της config.php και στη συνέχεια κάνουμε έλεγχο αν ο χρήστης έχει συνδεθεί αλλιώς τον ανακατευθύνουμε στην αρχική σελίδα (index.html). Στο πάνω μέρος της σελίδας υπάρχει το navigation bar στο οποίο προσθέσαμε τα κουμπιά για τις διεργασίες που μπορεί να κάνει ο χρήστης. Παρακάτω συμπεριλαμβάνουμε το σύστημα γεωγραφικής απεικόνισης. Αφού λοιπόν καλωσορίσουμε τον χρήστη, του εμφανίζεται μία μπάρα αναζήτησης για τα σημεία ενδιαφέροντος καθώς και ο χάρτης με έναν marker να υποδεικνύει την ακριβή τοποθεσία του. Υλοποιούμε

και μια συνάρτηση για το cookie που μας επιτρέπει να λάβουμε την τοποθεσία του χρήστη και να εστιάσουμε σε αυτή στο χάρτη .

Χάρτης

Στη συνέχεια έχουμε την ανάλυση του κώδικα για την λειτουργία του χάρτη μας του οποίου ο κώδικας είναι ο παρακάτω:

JS map.js > ...

```
1  var map_init = L.map('map', {
2    |   center: [9.0820, 8.6753],
3    |   zoom: 8
4  });
5  var osm = L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
6    |   attribution: '&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors'
7  }).addTo(map_init);
8  L.Control.geocoder().addTo(map_init);
9  if (!navigator.geolocation) {
10   |   console.log("Your browser doesn't support geolocation feature!")
11 } else {
12   |   setInterval(() => {
13     |     navigator.geolocation.getCurrentPosition(getPosition)
14     |   }, 5000);
15 };
16 var marker, circle, lat, long, accuracy;
17
18 function getPosition(position) {
19   |   console.log(position)
20   |   lat = position.coords.latitude
21   |   long = position.coords.longitude
22   |   accuracy = position.coords.accuracy
23
24   |   if (marker) {
25     |     map_init.removeLayer(marker)
26   |   }
27
28   |   if (circle) {
29     |     map_init.removeLayer(circle)
30   |   }
31
32   |   marker = L.marker([lat, long])
33   |   circle = L.circle([lat, long], { radius: accuracy })
34
35   |   var featureGroup = L.featureGroup([marker, circle]).addTo(map_init)
36
37   |   map_init.fitBounds(featureGroup.getBounds())
38
39   |   console.log("Your coordinate is: Lat: " + lat + " Long: " + long + " Accuracy: " + accuracy)
40 }
41
42
43 var x = document.cookie;
44 var numbers = [];
45 x.replace(/[-+]?[0-9]*\.\?[0-9]+/g, function( x ) { var n = Number(x); if (x == n) { numbers.push(x); } })
46 var average = numbers[0];
47 visit_estimation = numbers[1];
48 latitude = numbers[2];
49 longitude = numbers[3];
50 console.log(latitude);
51 console.log(longitude);
52 console.log(average);
53 console.log(numbers);
54 console.log(visit_estimation);
55 console.log(x);
56
57 var greenIcon = new L.Icon({
58   |   iconUrl: 'https://raw.githubusercontent.com/pointhi/leaflet-color-markers/master/img/marker-icon-2x-green.png',
59   |   shadowUrl: 'https://cdnjs.cloudflare.com/ajax/libs/leaflet/0.7.7/images/marker-shadow.png',
60   |   iconSize: [25, 41],
61   |   iconAnchor: [12, 41],
62   |   popupAnchor: [1, -34],
63   |   shadowSize: [41, 41]
64 });
65
```

```

65
66 var orangeIcon = new L.Icon({
67   iconUrl: 'https://raw.githubusercontent.com/pointhi/leaflet-color-markers/master/img/marker-icon-2x-orange.png',
68   shadowUrl: 'https://cdnjs.cloudflare.com/ajax/libs/leaflet/0.7.7/images/marker-shadow.png',
69   iconSize: [25, 41],
70   iconAnchor: [12, 41],
71   popupAnchor: [1, -34],
72   shadowSize: [41, 41]
73 });
74
75 var redIcon = new L.Icon({
76   iconUrl: 'https://raw.githubusercontent.com/pointhi/leaflet-color-markers/master/img/marker-icon-2x-red.png',
77   shadowUrl: 'https://cdnjs.cloudflare.com/ajax/libs/leaflet/0.7.7/images/marker-shadow.png',
78   iconSize: [25, 41],
79   iconAnchor: [12, 41],
80   popupAnchor: [1, -34],
81   shadowSize: [41, 41]
82 });
83
84 if (visit_estimation <= 32) {
85   var marker2 = L.marker([latitude, longitude], {icon: greenIcon}).addTo(map_init);
86 } else if (33 <= visit_estimation && visit_estimation <= 65) {
87   var marker2 = L.marker([latitude, longitude], {icon: orangeIcon}).addTo(map_init);
88 } else {
89   var marker2 = L.marker([latitude, longitude], {icon: redIcon}).addTo(map_init);
90 }
91
92 //var marker2 = L.marker([latitude, longitude], {icon: redIcon}).addTo(map_init);
93 var popup = marker2.bindPopup('<h3>Visit declaration:</h3><br>The visit estimation of the pol in % is:</br>' + visit_estimation + '<br>The average number of the visitors is:</br>' + average + '\
94   <div class="declaration" id="popup">\
95     <form action="popup.php" method="post" enctype="multipart/form-data">\
96       <label for="input-speed">Estimation of the crowd:</label>\
97       <input id="estimation" class="estimation" type="number" name="estimation"/>\
98       <button id="button-submit" type="submit">Declare Visit</button>\
99     </form>\
100   </div>');

```

Ο κώδικας για τον χάρτη μας, ο οποίος περιέχεται στο αρχείο user.php, αφού οριστεί το σημείο της ιστοσελίδας που θα φιλοξενηθεί και το μέγεθος του, ξεκινά με την ενσωμάτωση των απαραίτητων scripts και JS βιβλιοθηκών για την φόρτωση του. Στη συνέχεια τοποθετούμε τα Tiles και υλοποιούμε τη συνάρτηση getPosition μέσω της οποίας βρίσκουμε τη θέση του χρήστη μας στο χάρτη. Τέλος, έχοντας ήδη έναν marker που εμφανίζει την τρέχουσα τοποθεσία του χρήστη, δημιουργούμε markers για τα σημεία ενδιαφέροντος και ανάλογα με την επισκεψιμότητα που παρουσιάζει το κάθε ένα θα έχει και το κατάλληλο χρώμα (πράσινο:χαμηλή, πορτοκαλί: μέτρια, κόκκινο: υψηλή).

Ακολούθως, στο navigation bar έχουμε τοποθετήσει επιλογές τις οποίες θα αναλύσουμε εδώ.



## Edit profile

Αρχικά ο χρήστης μας έχει τη δυνατότητα να αλλάξει τα στοιχεία του προφίλ του κάτι που πετυχαίνουμε μέσω της `editprofile.html/php`. Ο πρώτος εκ των δύο κώδικας μας, μας δημιουργεί για άλλη μία φορά την μπάρα πλοήγησης και εν συνεχεία δημιουργούμε δύο πλαίσια για την τοποθέτηση του νέου ονόματος και του νέου συνθηματικού για το προφίλ τη χρήστη. Παρακάτω ο κώδικας:



```

editprofile.html > html > head
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8">
5      <title>Edit Profile</title>
6      <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.1/css/all.css">
7      <link href="style.css" rel="stylesheet" type="text/css">
8
9
10
11 <body class="loggedin">
12 <nav class="navtop">
13 <div>
14     <h1>CrowdScope</h1>
15     <!-- Navigation Bar -->
16     <a href="user.php"><i class="selected fas fa-user"></i>Dashboard</a>
17     <a href="instance_declaration.php"><i class="fas fa-chart-line"></i>Instance Declaration</a>
18     <a href="user_contact.php"><i class="fas fa-phone"></i>Contact User </a>
19     <a href="infectedUsers.html"><i class="fas fa-bell"></i>Upload day of infection </a>
20     <a href="editprofile.html"><i class="fas fa-wrench"></i>Edit Profile</a>
21     <a href="logout.php"><i class="fas fa-sign-out-alt"></i>Logout</a>
22
23 </div>
24 </nav>
25
26 <div class="content">
27     <h1>Edit Profile</h1>
28     <h2>You can change your username and password</h2>
29 <div>
30 <form action="editprofile.php" method="post">
31 <label for="username">
32     <i class="fas fa-user"></i>
33 </label>
34 <input type="text" name="username" placeholder=" New Username" id="username" required>
35 <label for="password">
36     <i class="fas fa-lock"></i>
37 </label>
38 <input type="password" name="password" placeholder="New Password" id="password" required>
39 <input type="submit" name="update_button" value="Update">
40 </form>
41 </div>
42 </div>
43 <table>
44
45 </table>
46 </body>
47
48
49 </html>

```

Επίσης ο παραπάνω κώδικας συνδέεται με τον editprofile.php του οποίου ο κώδικας είναι αυτός:

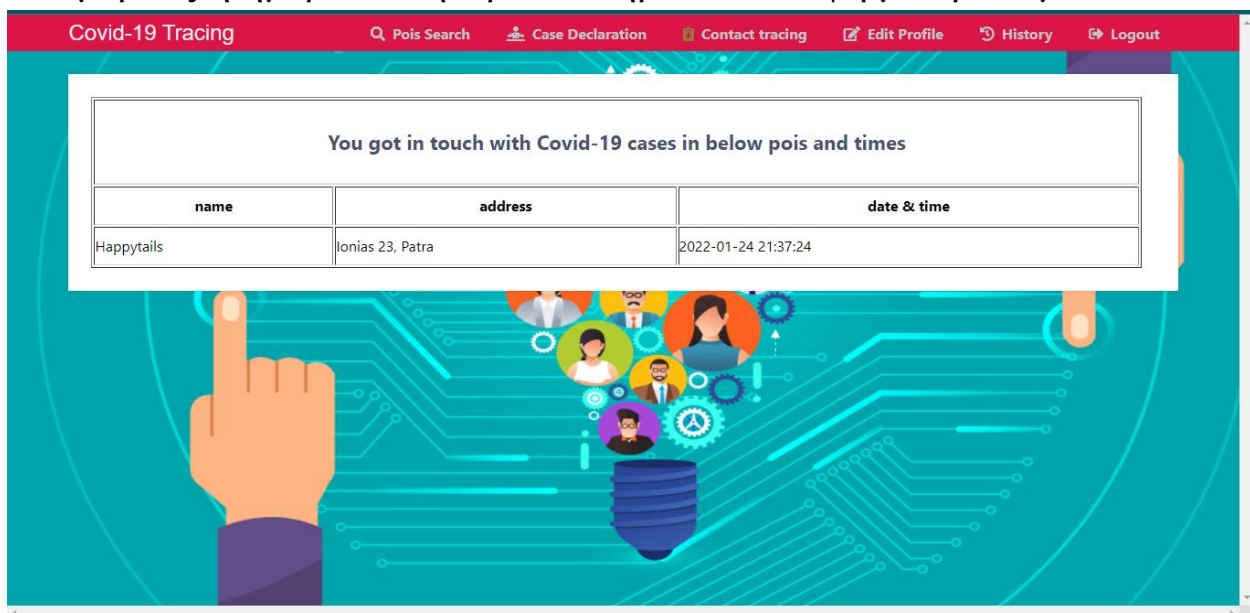
editprofile.php > ...

```
1  <?php
2  //Include the database
3  require('config.php');
4
5  // Start the session
6  session_start();
7
8  // If the user is not logged in redirect to the login page
9  if (!isset($_SESSION['loggedin'])) {
10     header('Location: index.html');
11     exit();
12 }
13
14 $userid = $_SESSION["id"];/* userid of the user */
15
16
17
18 // check if the data was submitted, isset() function will check if the data exists.
19 if (!isset($_POST['username'], $_POST['password'])) {
20     // Could not get the data that should have been sent.
21     die ('Please complete the form!');
22 }
23 // Make sure the submitted registration values are not empty and valid.
24 if (empty($_POST['username']) || empty($_POST['password'])) {
25     // One or more values are empty.
26     die ('Please complete the registration form');
27 }
28 if (preg_match('/[A-Za-z0-9]+/', $_POST['username']) == 0) {
29     die ('Username is not valid!');
30 }
31
32 $uppercase = preg_match('@[A-Z]@', $_POST['password']);
33 $number = preg_match('@[0-9]@', $_POST['password']);
34 $symbols = "!@#%&*()_+~{}[]|\?><.,/:\"";
35 $symbol = strpbrk($_POST['password'], $symbols);
36 if (!$uppercase) { die ('Password must contain at least one uppercase letter!'); }
37 if (!$number) { die ('Password must contain at least one number!'); }
38 if (!$symbol) { die ('Password must contain at least one symbol!'); }
39 if (strlen($_POST['password']) < 8) { die ('Password must be at least 8 characters long!'); }
40
41 if ($stmt = $con->prepare('UPDATE users SET username = ? , password = ? WHERE userid = ?')) {
42     $stmt->bind_param('sss', $_POST['username'], $_POST['password'], $userid);
43     $stmt->execute();
44     $stmt->close();
45     echo 'You have successfully update your data!';
46     // Redirect to the Dashboard page
47     header('Location: user.php');
48     header('Location: editprofile.html');
49     die();
50 }
51 else {
52     // Something is wrong with the sql statement, check to make sure .
53     echo 'Could not prepare statement!';
54 }
55
56 // Redirect to the Dashboard page
57 header('Location: user.php');
58 ?>
```

Εδώ παρατηρούμε πως ο κώδικάς μας ξεκινά από τη σύνδεση στη βάση δεδομένων με τη βοήθεια της εντολής `require('config.php')`; και του κώδικα `config.php` και στη συνέχεια αποθηκεύουμε σε μία μεταβλητή το `id` του χρήστη που βρίσκεται αυτή τη στιγμή στο τρέχον `session`. Έπειτα ο κώδικας ελέγχει αν έχει γίνει συμπλήρωση των δεδομένων για το νέο όνομα και το νέο κωδικό και στην περίπτωση που δεν έχει γίνει μας προβάλλει το ανάλογο μήνυμα. Αφού έχει υποβάλλει ο χρήστης τα δεδομένα το σύστημα ελέγχει όπως και στην `register.php` αν το νέο όνομα και ο καινούργιος κωδικός πληρούν τις ιδιότητες του συστήματός μας και εφόσον τις πληρούν το σύστημά μας ενημερώνει τη βάση δεδομένων `users` και επιστρέφει ότι η ενημέρωση έγινε με επιτυχία. Σε αντίθετη περίπτωση επιστρέφει το ανάλογο μήνυμα.

### Επαφή με κρούσμα

Στο αρχείο `contact_user.php` ο χρήστης έχει την δυνατότητα να δει σε ποιά μέρη και αν έχει έρθει σε επαφή με κρούσμα καθώς και την ημέρα και την ώρα που ήρθε σε επαφή. Πιο συγκεκριμένα, εμφανίζεται ένας πίνακας στον χρήστη που εμφανίζεται το όνομα του μέρους ή διεύθυνση του μέρους η ημέρα και η ώρα που ήρθε σε επαφή με κρούσμα.



You got in touch with Covid-19 cases in below pois and times		
name	address	date & time
Happytails	Ionias 23, Patra	2022-01-24 21:37:24

Στο κομμάτι του κώδικα:

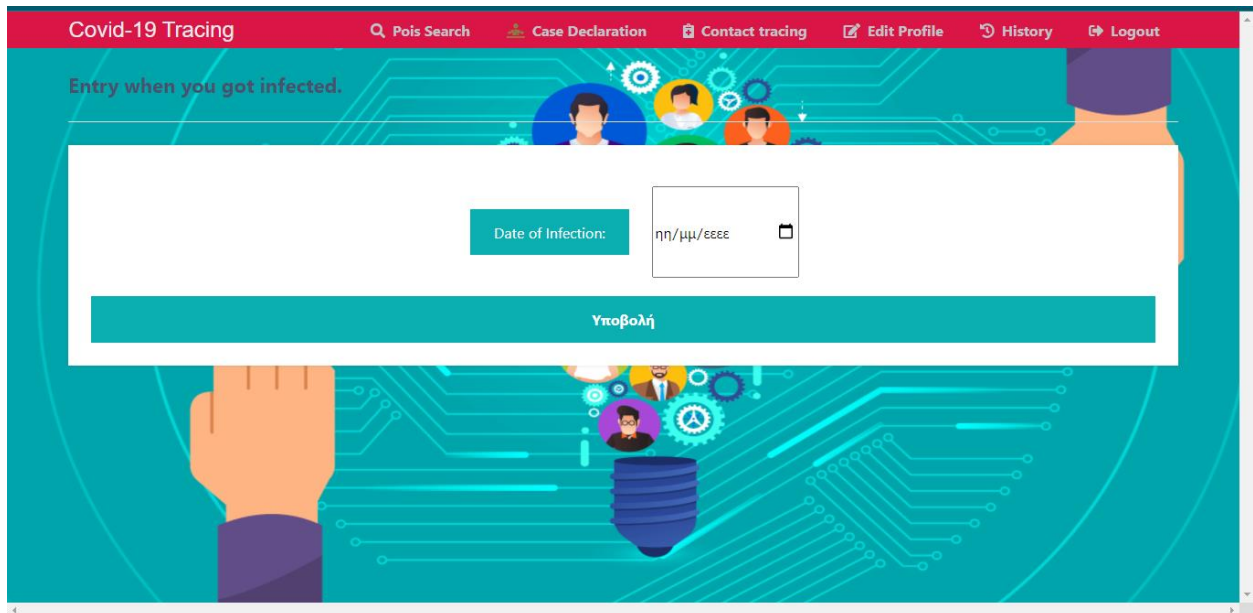
```
1 <php>  
2  
3 // Resume the previous session  
4 session_start();  
5  
6 // Include the database  
7 require('config.php');  
8  
9 //If the user is not logged in redirect to the login page  
10 if (!isset($_SESSION['loggedin'])) {  
11     header("location: index.html");  
12     exit();  
13 }  
14  
15  
16  
17  
18  
19 ?>  
20</DOCTYPE html>;  
21<html>  
22<head>  
23    <meta charset='utf-8'>  
24    <title>Contact Tracing</title>  
25    <link href= "style.css" rel= "stylesheet" type= "text/css">. ;  
26    <link rel= "stylesheet" href= "https://use.fontawesome.com/releases/v5.7.1/css/all.css">  
27</head>  
28<body class= "logged-in">  
29    <nav class= "navbar">  
30        <div>  
31            <a href= "#!><i>Covid-19 Tracing</i></a>  
32  
33            <div> Navigation Bar :</div>  
34            <a href= "/?fa=search"><i>Pois Search</i></a>  
35            <a href= "/instance_declaration.html"><i>Case Declaration</i></a>  
36            <a href= "/user_contact.php"><i>Contact tracing</i></a>  
37            <a href= "/editprofile.html"><i>Edit Profile</i></a>  
38            <a href= "/logout.php"><i>Logout</i></a>  
39        </div>  
40    </nav>  
41    <div class= "content">  
42        <div>  
43            <table align= "center" border= "1px solid powderblue" padding= "30px" style= "width:1150px;line-height:40px;">  
44                <tr>  
45                    <th colspan= "3">d2>You got in touch with Covid-19 cases in below pois and times/</th>  
46                </tr>  
47                <tr>  
48                    <th>name</th>  
49                    <th>address</th>  
50                    <th>date & time</th>  
51                </tr>  
52                <tr>  
53                    <td colspan= "3"><!-- php -->  
54  
55 $userid = $_SESSION["id"];  
56  
57 $query = $con->prepare("CREATE TABLE instances_visits AS select id,timestamp from visit_declaration as t1 inner join infectedusers t2 on t1.userid = t2.userid where t2.date BETWEEN DATE(t1.timestamp) AND ADDDATE(DATE(t1.timestamp), INTERVAL 7 DAY)");  
58 $query->execute();  
59 $query->close();  
60  
61 $query2 = $con->prepare("CREATE TABLE user_visits AS select id,timestamp from visit_declaration where userid = ?",  
62 $query2->bind_param('s', $userid);  
63 $query2->execute();  
64  
65  
66  
67 $query3 = "SELECT user.visits.id,user.visits.timestamp FROM user_visits INNER JOIN instances_visits ON user.visits.id = instances_visits.id WHERE DATE(user.visits.timestamp) = DATE(instances_visits.timestamp) AND user.visits.timestamp  
68 BETWEEN DATE_SUB(instances_visits.timestamp, INTERVAL 2 HOUR ) AND ADDDATE(instances_visits.timestamp, INTERVAL 2 HOURS)";  
69 $result = $con->query($query3);  
70 if ($result->num_rows > 0) {  
71     // Output data of each row  
72     while($row = $result->fetch_assoc()) {  
73         // Create select query to get ids of pois from data table  
74         $query4 = $con->prepare("SELECT name,address FROM data WHERE id = ?");  
75         // Bind the parameters and execute the query  
76         $query4->bind_param('s', $row["id"]);  
77         $query4->execute();  
78         $query4->store_result();  
79         $query4->bind_result($name,$address);  
80         $query4->fetch();  
81         $query4->close();  
82  
83         <br>  
84         <tr>  
85             <td><!-- php echo $name--></td>  
86             <td><!-- php echo $address--></td>  
87             <td><!-- php echo $row[timestamp]--></td>  
88         </tr>  
89     }  
90 }  
91  
92 $query5 = $con->prepare("DROP TABLE instances_visits");  
93 $query5->execute();  
94 $query5->close();  
95  
96 $query6 = $con->prepare("DROP TABLE user_visits");  
97 $query6->execute();  
98 $query6->close();  
99  
100 ?>  
101</table>  
102</div>  
103</div>  
104</body>  
105</html>
```

Δημιουργούμε έναν πίνακα ο οποίος παίρνει δεδομένα από τα query που υλοποιούμε. Το πρώτο query δημιουργεί έναν πίνακα με τα κρούσματα που έχουν επισκεφθεί κάποιο σημείο αν το έχουν διαγνωστεί θετικοί με covid-19 έως και 7 μέρες μετά την επίσκεψη. Το δεύτερο query δημιουργεί έναν πίνακα με τα σημεία που έχει επισκεφτεί ο χρήστης. Το τρίτο query κάνει match τα κοινά σημεία που έχει επισκεφτεί ο χρήστης και κάποιο κρούσμα με τις ώρες επίσκεψης να απέχουν έως και  $\pm 2$

ώρες. Τέλος το τέταρτο query εξάγει τα ονόματα των κοινών σημείων στον πίνακα.

### Δήλωση Κρούσματος

Με το αρχείο `instance_declaration.html` δημιουργούμε μια οθόνη στην οποία ο χρήστης μπορεί να δήλωσει αν είναι κρούσμα καθώς και την ημερομηνία που βρέθηκε θετικός στον Covid-19. Πιο συγκεκριμένα δημιουργούμε μια φόρμα στην οποία ο χρήστης έχει επιλογή να επιλέξει ποια μέρα βρέθηκε θετικός και να πατήσει ένα κουμπί ώστε να ολοκληρώσει την δήλωση.



The screenshot displays the 'Covid-19 Tracing' web application. The top navigation bar is red and contains links for 'Pois Search', 'Case Declaration', 'Contact tracing', 'Edit Profile', 'History', and 'Logout'. Below the navigation bar, the main content area has a teal background with a circuit-like pattern and icons of people. A white form box is centered on the screen. Inside the form, there is a label 'Date of Infection:' next to a date input field with a calendar icon. Below the input field is a teal button with the text 'Υποβολή' (Submit).

Στο κομμάτι του κώδικα:

```

<> instance_declaration.html > ...
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8">
5      <title>Instance Declaration</title>
6      <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.1/css/all.css">
7      <link href="style.css" rel="stylesheet" type="text/css">
8
9  </head>
10 <body class="loggedin">
11 <nav class="navtop">
12     <div>
13         <h1>Covid-19 Tracing</h1>
14
15         <!-- Navigation Bar -->
16         <a href="user.php"><i class="fas fa-search"></i>Pois Search</a>
17         <a href="instance_declaration.html"><i class="selected fas fa-diagnoses"></i>Case Declaration</a>
18         <a href="user_contact.php"><i class="fas fa-notes-medical"></i>Contact tracing</a>
19         <a href="editprofile.html"><i class="fas fa-edit"></i>Edit Profile</a>
20         <a href="logout.php"><i class="fas fa-sign-out-alt"></i>Logout</a>
21     </div>
22 </nav>
23 <div class="content">
24     <h2>Enter date if Infection:.</h2>
25     <div>
26         <form action="instance_declaration.php" method="post">
27             <label>Date of Infection:</label>
28             <br/><br/>
29             <input type="date" name="date">
30             <input type="submit" name="submit">
31         </form>
32     </div>
33 </div>
34 </body>
35 </html>
36

```

Στις γραμμές 3-9 του αρχείου instance\_declaration.html καθορίζουμε την κεφαλίδα του κώδικα html στον οποίο ορίζουμε το όνομα της σελίδας στον browser και επισυνάπτουμε αρχεία που καθορίζουν το στυλ της σελίδας.

Στις γραμμές 12-21 καθορίζουμε το όνομα της εφαρμογής καθώς και τις επιλογές του navigation bar. Τέλος, στις γραμμές 23-32 δημιουργούμε μια φόρμα με μια είσοδο για ημερομηνία και ένα κουμπί καταχώρισης.

```

instance_declaration.php > ...
1  <?php
2  // Include the database
3  require_once('config.php');
4
5  // Resume the previous session
6  session_start();
7
8  // If the user is not logged in redirect to the Login page
9  if (!isset($_SESSION['loggedin'])) {
10     header('Location: index.html');
11     exit();
12 }
13 if (isset($_POST["date"])){
14
15     $date = $_POST["date"];
16     $userid = $_SESSION["id"];
17     $infected = 1;
18
19     $sql = "insert into infectedusers(date, userID, infected) values ('$date', '$userid', '$infected')";
20     $sql1 = "insert into userhistory(userid, date, infected) values ('$userid', '$date', '$infected')";
21
22     $run = mysqli_query($con,$sql);
23     $run = mysqli_query($con,$sql1);
24     if($run){
25
26
27     echo "<h2>New record created successfully!</h2>";
28     } else {
29     echo "Error: " . $sql . "<br>" . mysqli_error($con);
30     }
31
32     header('Location: instance_declaration.html');
33 }
34 else { die ('Please complete the date!'); }
35 }>
36

```

Στο αρχείο instance\_declaration.php δημιουργούμε το backend κομμάτι του αρχείου instance\_declaration.html. Αρχικά συμπεριλαμβάνουμε την βάση στην γραμμή 3. Στις γραμμές 13-30 ελέγχουμε αν έχει συμπληρωθεί η ημερομηνία στην φόρμα και εάν ναι τότε τρέχουν δύο query που μεταφέρουν τα δεδομένα σε δύο πίνακες. Δημιουργήσαμε επίσης ένα SQL Event το οποίο διαγράφει από τον πίνακα infectedUsers τα entries που είναι μεγαλύτερα από 14 μέρες, εφόσον τότε ο χρήστης θα έχει θεωρητικά ξεπεράσει τον ιό.



```
CREATE EVENT IF NOT EXISTS `Clean_Older_Than_14_Days`
ON SCHEDULE
    EVERY 1 DAY_HOUR
DO
DELETE FROM infectedusers
    WHERE date < DATE_SUB(NOW(), INTERVAL 14 DAY)
```

Προβολή ιστορικού επίσκεψης και νόσησης

Με το αρχείο history.php δημιουργούμε την παρακάτω οθόνη. Αρχικά δημιουργούμε δύο πίνακες. Στον ένα πίνακα εμφανίζονται οι τοποθεσίες που έχει επισκεφθεί ο χρήστης. Στον άλλο πίνακα εμφανίζονται οι ημερομηνίες που δήλωσε ότι είναι κρούσμα ο χρήστης.



You visited below pois at below times		
name	address	date & time
Happytails	Ionias 23, Patra	2022-02-09 16:22:15
DERMATOLOGOS GIANNAKOPOULOS Patras KOSTAS	Voulgareos 27, Patra	2022-02-10 16:17:56
Pet Beauty G.S.	Βορείου Ηπείρου 170 και, Maizonos, Patra	2022-02-10 16:18:15
SUPERMARKET "3A ARAPIS"	Maizonos 22-24, Patra	2022-02-10 16:18:34
Μυλωνάς Ζαχαροπλαστείο	Leoforos Georgiou Papandreou 44, Patra	2022-02-10 16:19:05
www.abbyspaws.gr	Themistokleous 56, Patra	2022-02-10 16:19:29

Ελληνικά Ταχυδρομεία (ΕΛΤΑ)	Maizonos 32, Patra	2022-02-10 16:25:38
Σαν όνειρο	Maizonos 29, Patra	2022-02-10 16:26:09
Grill house πατρα	Voriou Ipirou 89, Patra	2022-02-10 16:26:32
Abige Hair&Style	Kapodistriou 28, Patra	2022-02-10 16:26:58
PHARMACY Damvounelis G. Kon / nos	Leoforos Georgiou Papandreou 85, Patra	2022-02-10 16:27:18
Παραδοσιακό Οπωροπαντοπωλείο	6 συντάγματος 2, Πάτρα	2022-02-10 16:28:15
Zolotas S.A.	Riga Fereou 32, Patra	2022-02-10 16:29:18
Σαπουναρία Πάτρας (S.P)	Πόντου κ, Efesou 47, Patra	2022-02-10 16:29:38

You were covid-19 case at below date
date
2022-02-09

Παρακάτω ακολουθεί ο κώδικας του αρχείου:

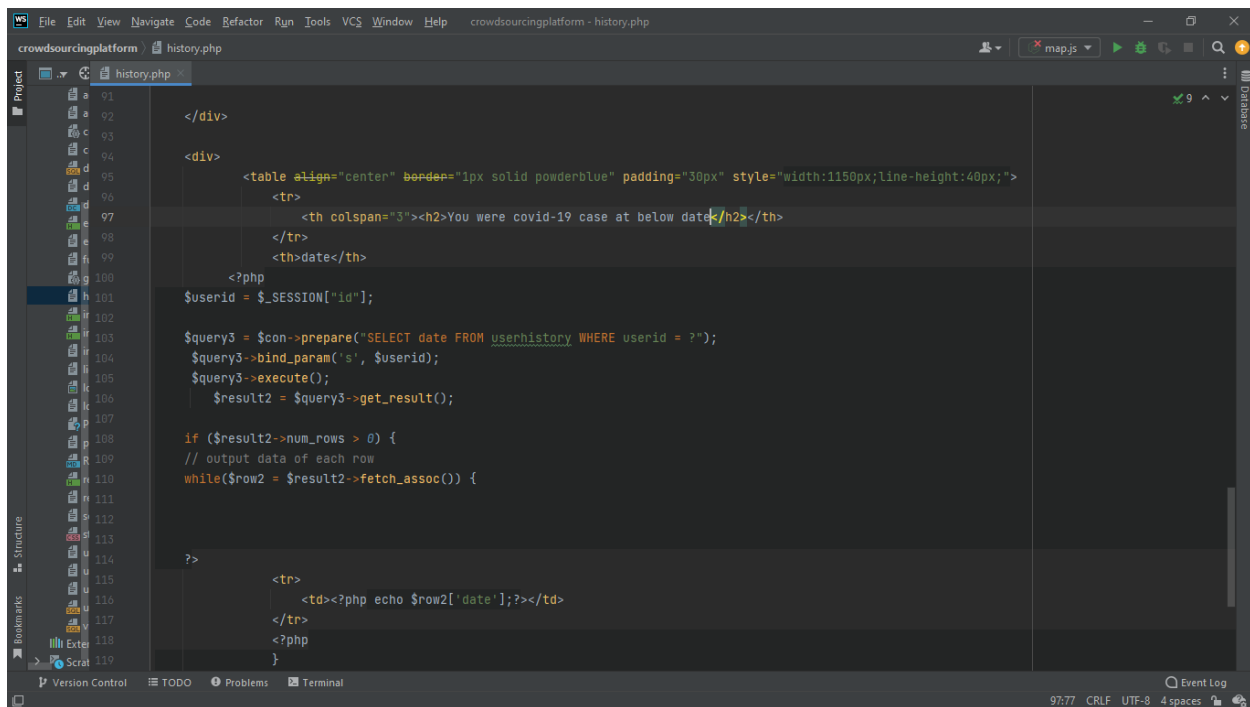
```

1 <?php
2
3 // Resume the previous session
4 session_start();
5
6 // Include the database
7 require('config.php');
8
9 //If the user is not logged in redirect to the login page
10 if (!isset($_SESSION['loggedin'])) {
11     header('Location: index.html');
12     exit();
13 }
14
15
16
17 ?>
18
19 <!DOCTYPE html>
20 <html>
21 <head>
22     <meta charset="utf-8">
23     <title>Edit Profile</title>
24     <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.1/css/all.css">
25     <link href="style.css" rel="stylesheet" type="text/css">
26
27
28 <body class="loggedin">
29

```

```
crowdsourcingplatform - history.php
crowdsourcingplatform - history.php
25 <link href="style.css" rel="stylesheet" type="text/css">
26
27
28
29 <body class="loggedin">
30 <nav class="navtop">
31 <div>
32 <h1>Covid-19 Tracing</h1>
33 <!-- Navigation Bar -->
34 <a href="user.php"><i class="fas fa-search"></i>Pojs Search</a>
35 <a href="instance_declaration.html"><i class="fas fa-diagnoses"></i>Case Declaration</a>
36 <a href="user_contact.php"><i class="fas fa-notes-medical"></i>Contact tracing</a>
37 <a href="editprofile.html"><i class="fas fa-edit"></i>Edit Profile</a>
38 <a href="history.php"><i class="selected fas fa-history"></i>History</a>
39 <a href="logout.php"><i class="fas fa-sign-out-alt"></i>Logout</a>
40
41 </div>
42 </nav>
43
44 <div class="content">
45 <h1>History</h1>
46 <h2>You can see your visit history and when you were covid-19 case</h2>
47
48 <div>
49 <table align="center" border="1px solid powderblue" padding="30px" style="width:1150px;line-height:40px;">
50 <tr>
51 <th colspan="3"><h2>You visited below pojs at below times</h2></th>
52 </tr>
53 <th>name</th>
```

```
crowdsourcingplatform - history.php
crowdsourcingplatform - history.php
54 <th>date & time</th>
55
56 <?php
57 $userid = $_SESSION["id"];
58
59 $query = $con->prepare("SELECT id,timestamp FROM visit_declaration WHERE userid = ?");
60 $query->bind_param('s', $userid);
61 $query->execute();
62 $result = $query->get_result();
63
64 if ($result->num_rows > 0) {
65 // output data of each row
66 while($row = $result->fetch_assoc()) {
67
68 // Create select query to get name of pojs from data table
69 $query2 = $con->prepare("SELECT name,address FROM data WHERE id = ?");
70 // Bind the parameters and execute the query
71 $query2->bind_param('s', $row["id"]);
72 $query2->execute();
73 $query2->store_result();
74 $query2->bind_result($name, $address);
75 $query2->fetch();
76 $query2->close();
77
78 ?>
79
80 <tr>
81 <td><?php echo $name;?></td>
82 <td><?php echo $address;?></td>
83 <td><?php echo $row['timestamp'];?></td>
```



```
91
92
93
94
95 <table align="center" border="1px solid powderblue" padding="30px" style="width:1150px;line-height:40px;">
96
97 <tr>
98 <th colspan="3"><h2>You were covid-19 case at below date</h2></th>
99 </tr>
100 <th>date</th>
101
102 <?php
103 $userid = $_SESSION["id"];
104
105 $query3 = $con->prepare("SELECT date FROM userhistory WHERE userid = ?");
106 $query3->bind_param('s', $userid);
107 $query3->execute();
108 $result2 = $query3->get_result();
109
110 if ($result2->num_rows > 0) {
111 // output data of each row
112 while($row2 = $result2->fetch_assoc()) {
113
114
115 ?>
116
117 <tr>
118 <td><?php echo $row2['date'];?></td>
119 </tr>
120 <?php
121 }
```

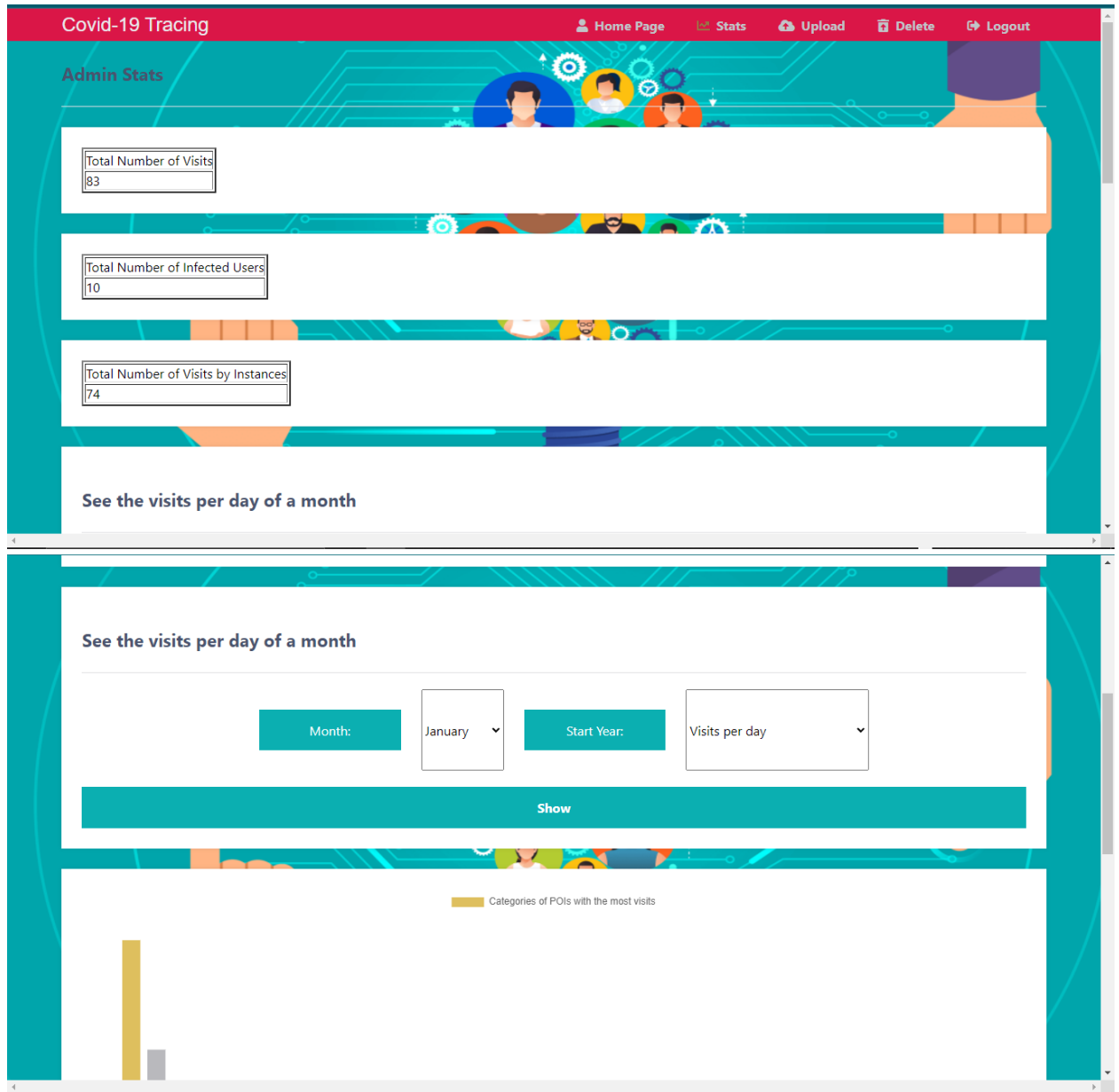
### 4.3.2) Διαχειριστής - Επιλογές

Εφόσον ο χρήστης **είναι διαχειριστής** του συστήματός μας, ύστερα από την εισαγωγή των στοιχείων του τον μεταφέρουμε στο αρχείο admin.php του οποίας ο κώδικας βρίσκεται εδώ :

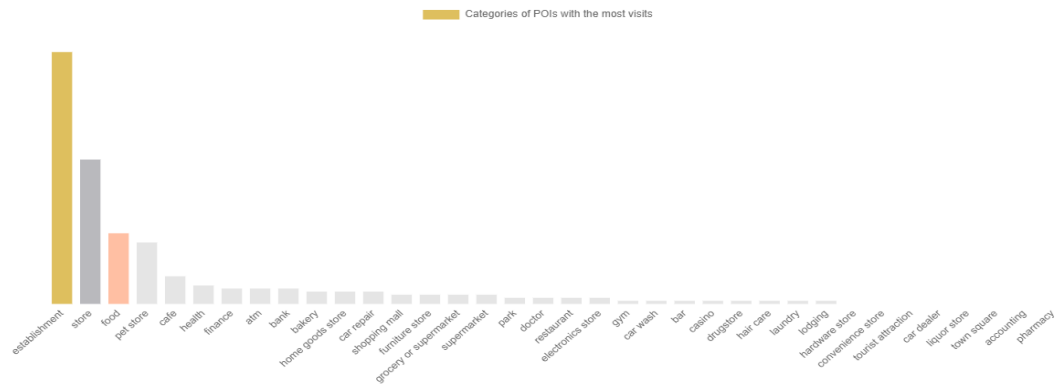
Ο διαχειριστής μέσω του navigation bar του έχει την πρόσβαση στις εξής δυνατότητες:

1. Προβολή στατιστικών της υπηρεσίας μέσω του πλήκτρου “Stats” και εκτέλεσης του αρχείου admin\_stats.php. Σε αυτό το αρχείο δημιουργούμε αρχικά των html κώδικα της οθόνης ο οποίος περιέχει τους πίνακες που εμφανίζονται τα δεδομένα. Στη συνέχεια, δημιουργούμε τα queries για να επικοινωνεί το frontend μέρος με τη βάση και να εξάγονται τα δεδομένα στους πίνακες και στα javascript γραφήματα. Τέλος, αρχείο περιέχει javascript

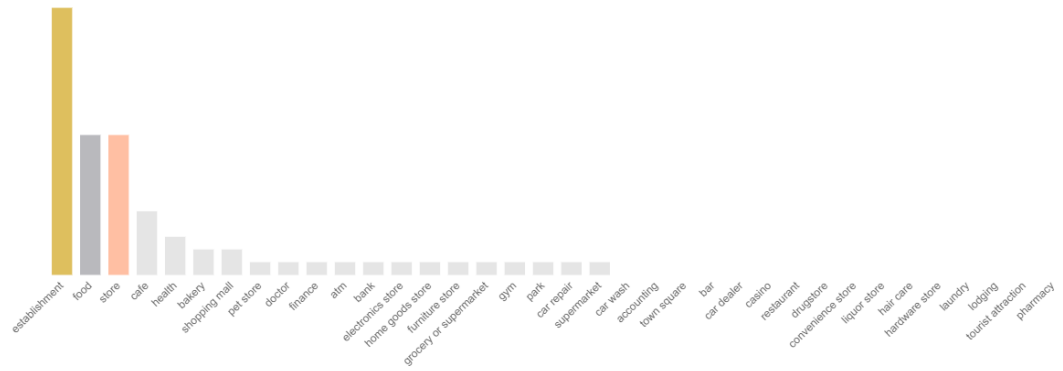
κώδικα με τον οποίο δημιουργούμε γραφήματα για να αναπαρασταθούν τα δεδομένα. Το αποτέλεσμα της οθόνης είναι ως εξής:

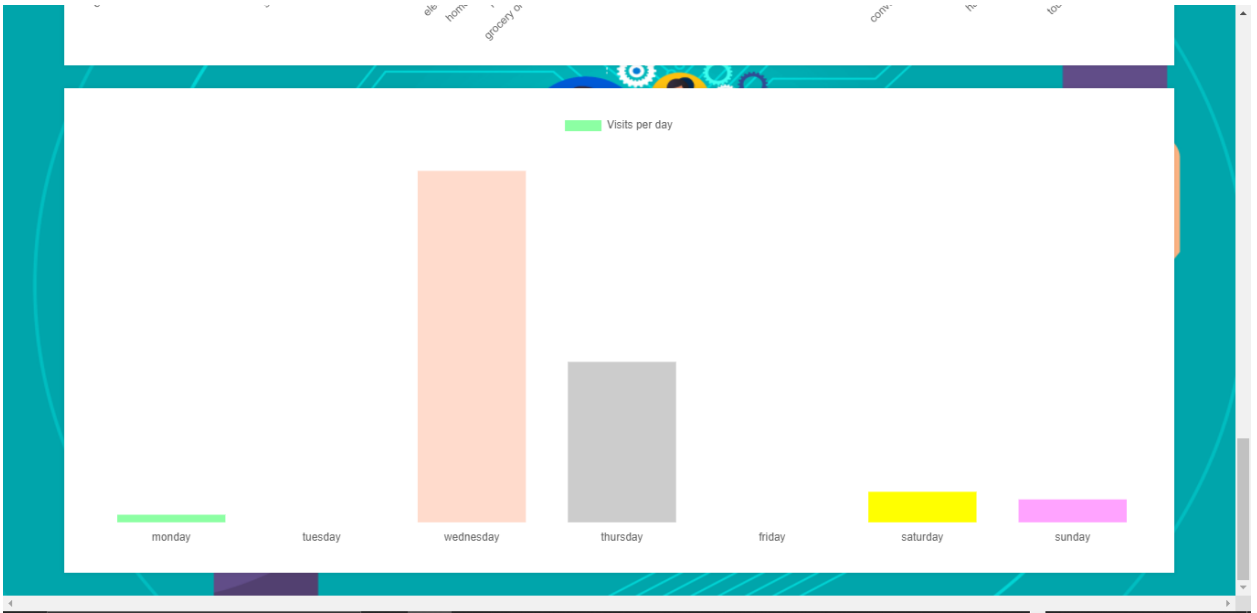


Show



Categories of POIs with the most visits by instances





Παρακάτω αποσπάσματα από τον κώδικα του admin\_stats:

```
148     <?php
149     //categories' leaderboard by visits
150     // Create select query to get POIs ids from visit_declaration table
151     $query2 = "SELECT id FROM visit_declaration";
152     $result2 = $con->query($query2);
153     $pet_store = NULL;
154     $cafe = NULL;
155     $restaurant = NULL;
156     $food = NULL;
157     $establishment = NULL;
158     $convenience_store = NULL;
159     $grocery_or_supermarket = NULL;
160     $store = NULL;
161     $bakery = NULL;
162     $supermarket = NULL;
163     $car_repair = NULL;
164     $park = NULL;
165     $tourist_attraction = NULL;
166     $gym = NULL;
167     $health = NULL;
168     $car_wash = NULL;
169     $liquor_store = NULL;
170     $shopping_mall = NULL;
171     $furniture_store = NULL;
172     $home_goods_store = NULL;
173     $lodging = NULL;
174     $laundry = NULL;
175     $hardware_store = NULL;
176     $electronics_store = NULL;
177     $hair_care = NULL;
178     $drugstore = NULL;
179     $bank = NULL;
180     $atm = NULL;
181     $finance = NULL;
182     $doctor = NULL;
183     $casino = NULL;
184     $car_dealer = NULL;
185     $bar = NULL;
186     $town_square = NULL;
187     $accounting = NULL;
188     $pharmacy = NULL;
189
```



```

//count pet_store number
$stmt4 = $con->prepare("SELECT COUNT(*) FROM data WHERE categories LIKE 'pet_store%' AND id = ?");
// Bind the parameters and execute the query
$stmt4->bind_param('s', $row2["id"]);
$stmt4->execute();
$stmt4->store_result();
$stmt4->bind_result($pet_store_result);
$stmt4->fetch();
$stmt4->close();
$pet_store = $pet_store + $pet_store_result;

//count cafe number
$stmt5 = $con->prepare("SELECT COUNT(*) FROM data WHERE categories LIKE 'cafe%' AND id = ?");
// Bind the parameters and execute the query
$stmt5->bind_param('s', $row2["id"]);
$stmt5->execute();
$stmt5->store_result();
$stmt5->bind_result($cafe_result);
$stmt5->fetch();
$stmt5->close();
$cafe = $cafe + $cafe_result;

//count restaurant number
$stmt6 = $con->prepare("SELECT COUNT(*) FROM data WHERE categories LIKE 'restaurant%' AND id = ?");
// Bind the parameters and execute the query
$stmt6->bind_param('s', $row2["id"]);
$stmt6->execute();
$stmt6->store_result();
$stmt6->bind_result($restaurant_result);
$stmt6->fetch();
$stmt6->close();
$restaurant = $restaurant + $restaurant_result;

//count food number
$stmt7 = $con->prepare("SELECT COUNT(*) FROM data WHERE categories LIKE 'food%' AND id = ?");
// Bind the parameters and execute the query
$stmt7->bind_param('s', $row2["id"]);
$stmt7->execute();
$stmt7->store_result();
$stmt7->bind_result($food_result);
$stmt7->fetch();

```

```

<script>

var ctx = document.getElementById('visitsperdayofmonthChart').getContext('2d');
var myChart = new Chart(ctx, {
  type: 'bar',
  data: {
    labels: <?php echo $arrayLabel2; ?>,
    datasets: [{
      label: 'Visits per day' ,
      data: <?php echo $arrayValue2; ?>,
      backgroundColor: ['rgba(25, 255, 71, 0.5)', 'rgba(52, 75, 239, 0.5)', 'rgba(255, 77, 0, 0.2)',
        'rgba(0, 0, 0, 0.2)', 'rgba(0, 255, 185, 1)', 'rgba(255, 255, 0, 1)', 'rgba(255, 163, 255, 1)'],
    }]
  },
  options: {
    maintainAspectRatio: false,
    scales: {
      yAxes: [{
        gridLines: { display: false },
        ticks: { display: false, beginAtZero: true }
      }],
      xAxes: [{ gridLines: { display: false } }]
    }
  }
});
</script>
<script>

var ctx = document.getElementById('visitsperdayofmonthinstancesChart').getContext('2d');
var myChart = new Chart(ctx, {
  type: 'bar',
  data: {
    labels: <?php echo $arrayLabel3; ?>,
    datasets: [{
      label: 'Visits per day' ,
      data: <?php echo $arrayValue3; ?>,
      backgroundColor: ['rgba(25, 255, 71, 0.5)', 'rgba(52, 75, 239, 0.5)', 'rgba(255, 77, 0, 0.2)',
        'rgba(0, 0, 0, 0.2)', 'rgba(0, 255, 185, 1)', 'rgba(255, 255, 0, 1)', 'rgba(255, 163, 255, 1)'],
    }]
  },
  options: {
    maintainAspectRatio: false,

```

2. Προσθήκη σημείων ενδιαφέροντος μέσω του πλήκτρου “Upload” και εκτέλεσης του αρχείου admin\_upload.php, στο οποίο έχουμε ενσωματώσει το upload.php, εκεί όπου, αφού συνδεθούμε στην βάση, γράφουμε τις απαραίτητες εντολές για να αναλυθούν οι πληροφορίες από τα JSON αρχεία και να μπορέσουν να γίνουν entries στον πίνακα data.sql. Τεχνικά δημιουργούμε μια φόρμα με html όπου ο χρήστης πατάει το κουμπί “επιλογή αρχείου” και του εμφανίζεται ένα popup παράθυρο που μπορεί να επιλέξει ένα

αρχείο .json από τον υπολογιστή του. Στη συνέχεια δημιουργούμε το κουμπί “submit” ώστε όταν το πατήσει ο χρήστης να γίνει parsing το αρχείο που ανεβάζει και μέσω ενός query να σταλούν τα δεδομένα στον πίνακα data. Η οθόνη που προκύπτει είναι η εξής:

Ο κώδικας του αρχείου admin\_upload.php είναι ο παρακάτω:

```
14 <!DOCTYPE html>
15 <html>
16   <head>
17     <meta charset="utf-8">
18     <title>Upload Data</title>
19     <link href="style.css" rel="stylesheet" type="text/css">
20     <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.1/css/all.css">
21
22   </head>
23   <body class="loggedin">
24     <nav class="navtop">
25       <div>
26         <h1>Covid-19 Tracing</h1>
27
28         <!-- Navigation Bar -->
29         <a href="admin.php"><i class="fas fa-user"></i>Home Page</a>
30         <a href="admin_stats.php"><i class="fas fa-chart-line"></i>Stats</a>
31         <a href="admin_upload.php"><i class="selected fas fa-cloud-upload-alt"></i>Upload</a>
32         <a href="admin_delete.php"><i class="fas fa-trash-restore-alt"></i>Delete</a>
33         <a href="logout.php"><i class="fas fa-sign-out-alt"></i>Logout</a>
34
35       </div>
36     </nav>
37
38     <div class="content">
39       <h2>Upload Data</h2>
40       <p>Select the location of file in JSON format.</p>
41       <div>
42         <form action="upload.php" method="post" enctype="multipart/form-data">
43           <label for="month">Filename:</label>
44           <input type="file" name="file" id="file">
45           <input type="hidden" name="areas" id="areas" value="[]">
46           <input type="submit" name="submit" value="Submit">
47         </form>
48       </div>
49       <p>When uploading big JSON files it will take a while to process all the locations. Please be patient.</p>
50     </div>
51
52   </body>
53 </html>
```

```

1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>Upload Data</title>
6      <link href="style_.css" rel="stylesheet" type="text/css">
7      <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.1/css/all.css">
8    </head>
9    <body class="loggedin">
10     <nav class="navtop">
11       <div>
12         <h1>CrowdScope</h1>
13
14         <!-- Navigation Bar -->
15         <a href="admin.php"><i class="fas fa-user"></i>Dashboard</a>
16         <a href="admin_stats.php"><i class="fas fa-chart-line"></i>Stats</a>
17         <a href="admin_upload.php"><i class="selected fas fa-cloud-upload-alt"></i>Upload</a>
18         <a href="admin_delete.php"><i class="fas fa-trash-restore-alt"></i>Delete</a>
19         <a href="logout.php"><i class="fas fa-sign-out-alt"></i>Logout</a>
20
21       </div>
22     </nav>
23     <div class="content">
24       <h2>Upload Data</h2>
25
26       <?php
27         // Include the database
28         require('config.php');
29         // Resume the previous session
30         session_start();
31         // If the user is not logged in redirect to the login page
32         if (!isset($_SESSION['loggedin'])) {
33           header('Location: index.html');
34           exit();
35         }
36         ini_set('display_errors', 1);
37         ini_set('display_startup_errors', 1);
38         error_reporting(E_ALL);
39         require_once('vendor/autoload.php');
40         //require_once('functions.php');
41
42         // Load the censored areas
43         $rect = json_decode($_POST["areas"], true);

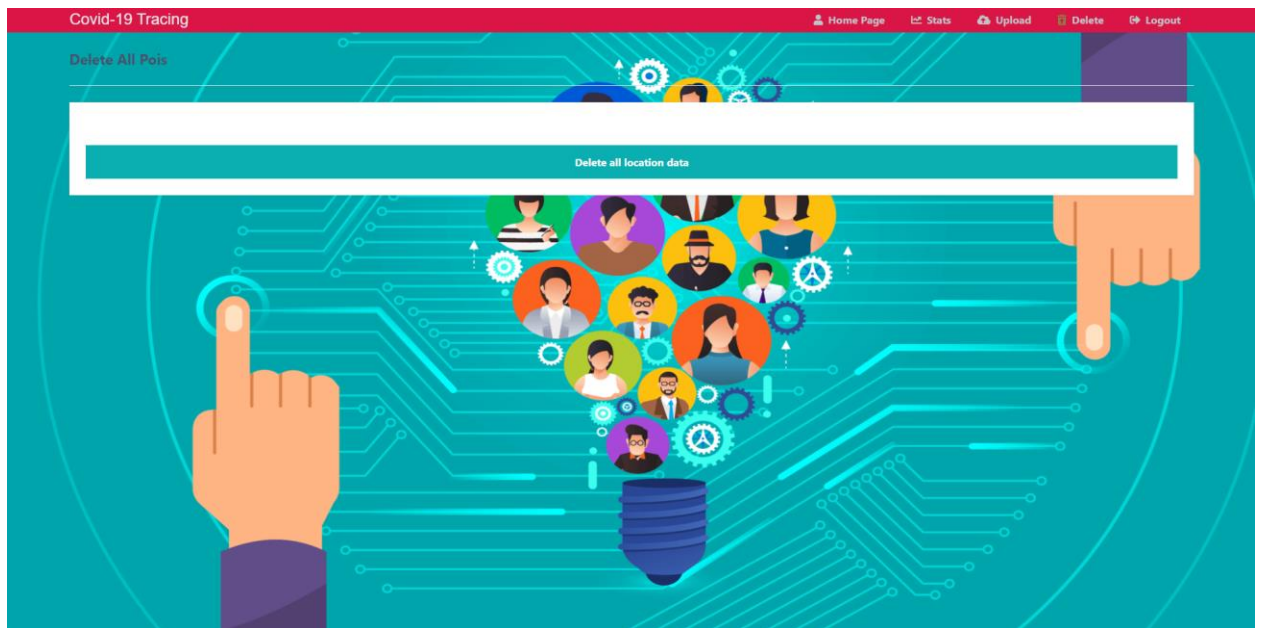
```

```

45     if(isset($_POST["submit"]) && isset($_POST["areas"])) {
46
47         $target_path = $_SERVER["DOCUMENT_ROOT"]."uploads";
48
49         if (file_exists($target_path.$_FILES["file"]["name"])) {
50             // echo "<p>An error occured while uploading your data. Please try again.</p>";
51             unlink($target_path.$_FILES["file"]["name"]);
52         }
53         move_uploaded_file($_FILES["file"]["tmp_name"], $target_path.$_FILES["file"]["name"]);
54         // Load file process
55         try {
56             // Create the pointer for the JSON parser
57             $jsonStream = \JsonMachine\JsonMachine::fromFile($target_path.$_FILES["file"]["name"]);
58             // Counters to count the number of Locations that were processed
59             $cnt = 0;
60             $userid = $_SESSION["id"];
61             // For each entry in the JSON
62             foreach ($jsonStream as $name => $data) {
63                 // Load information
64                 $id = $data["id"];
65                 $name = $data["name"];
66                 $address = $data["address"];
67                 $latitude = $data["coordinates"]["lat"];
68                 $longitude = $data["coordinates"]["lng"];
69                 $array = $data["types"];
70                 $categories = implode(',', $array);
71                 // Create insert query
72                 $query = "INSERT INTO data (userid, id, name, address, latitude, longitude, categories) VALUES ('$userid', '$id', '$name', '$address', '$latitude', '$longitude', '$categories')";
73                 $con->query($query);
74                 // Increment the counter
75                 $cnt = $cnt + 1;
76                 //echo "<h2>" . $query . "</h2>";
77             }
78         } catch (Exception $e) {
79             echo "<p>Something went wrong. Please try again.</p>";
80         }
81         //unlink($target_path.$_FILES["file"]["name"]);
82         echo "<p>Uploaded $cnt locations successfully!</p>";
83     } else {
84         echo "<p>Something went wrong. Please try again.</p>";
85     }
86 }
87 $con->close();

```

3. Διαγραφή σημείων ενδιαφέροντος μέσω του πλήκτρου “Delete” και εκτέλεσης του αρχείου admin\_delete.php το οποίο εκτελεί το απλό query του αρχείου delete.php για την διαγραφή των των πληροφοριών του πίνακα data. Τεχνικά δημιουργούμε ένα κουμπί που όταν το πατά ο χρήστης πυροδοτεί ένα query μέσω της php που διαγράφει όλα τα δεδομένα από τον πίνακα data. Η οθόνη που προκύπτει είναι η εξής:



Ο κώδικας του αρχείου `admin_delete` ακολουθεί παρακάτω:

```
<head>
  <meta charset="utf-8">
  <title>Admin Dashboard</title>
  <link href="style.css" rel="stylesheet" type="text/css">
  <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.1/css/all.css">
</head>
<body class="loggedin">
<nav class="navtop">
  <div>
    <h1>Covid-19 Tracing</h1>

    <!-- Navigation Bar -->
    <a href="admin.php"><i class="fas fa-user"></i>Home Page</a>
    <a href="admin_stats.php"><i class="fas fa-chart-line"></i>Stats</a>
    <a href="admin_upload.php"><i class="fas fa-cloud-upload-alt"></i>Upload</a>
    <a href="admin_delete.php"><i class="selected fas fa-trash-restore-alt"></i>Delete</a>
    <a href="logout.php"><i class="fas fa-sign-out-alt"></i>Logout</a>
  </div>
</nav>
<div class="content">
<h2>Delete All Pois</h2>
  <div>
    <form action="delete.php">
      <input type="submit" value="Delete all location data"/>
    </form>
  </div>

</div>

</body>
</html>
```

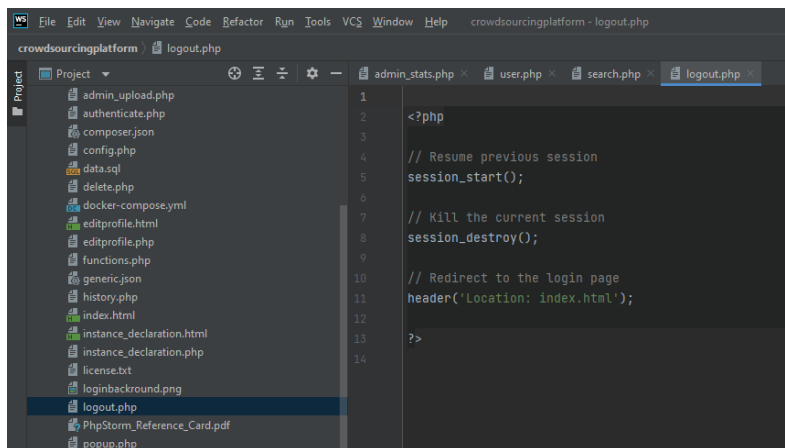


```

3 <head>
4 <meta charset="utf-8">
5 <title>Delete Data</title>
6 <link href="style.css" rel="stylesheet" type="text/css">
7 <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.1/css/all.css">
8 </head>
9 <body class="loggedin">
10 <nav class="navtop">
11 <div>
12 <h1>CrowdScope</h1>
13
14 <!-- Navigation Bar -->
15 <a href="admin.php"><i class="fas fa-user"></i>Dashboard</a>
16 <a href="user_stats.php"><i class="fas fa-chart-line"></i>Stats</a>
17 <a href="admin_upload.php"><i class="fas fa-cloud-upload-alt"></i>Upload</a>
18 <a href="delete.php"><i class="fas fa-trash-restore-alt"></i>Delete</a>
19 <a href="logout.php"><i class="fas fa-sign-out-alt"></i>Logout</a>
20
21 </div>
22 </nav>
23 <div class="content">
24
25 <?php
26
27 // Include the database
28 require('config.php');
29 // Resume the previous session
30 session_start();
31 // If the user is not logged in redirect to the login page
32 if (!isset($_SESSION['loggedin'])) {
33     header('Location: index.html');
34     exit();
35 }
36
37 // Delete data from all the users
38 $con->query("DELETE FROM data");
39 $con->close();
40
41 echo "<p>Data deleted successfully!</p>";
42 ?>
43
44 </div>
45 </body>

```

4. Αποσύνδεση μέσω του πλήκτρου “Logout” και εκτέλεση του αρχείου logout.php. Τεχνικά δημιουργούμε το κουμπί “Logout” στο navigation bar που όταν το πατήσει ο χρήστης τον αποσυνδέει από την εφαρμογή. Αυτό εκτελείται από τον παρακάτω κώδικα:



Ανα πάσα στιγμή ο διαχειριστής μπορεί να επιστρέψει στην αρχική του σελίδα με το πάτημα της επιλογής “Home Page”, το οποίο ανακαλεί το αρχείο admin.php.

## 5) Σχεδιαστικές Επιλογές

Αρχικά, επιλέξαμε μια χρωματική παλέτα που βασίζεται στο μπλε καθώς σύμφωνα με την χρωματική θεωρία αυτό εμπνέει αισθήματα εμπιστοσύνης, ευστάθειας και παγιότητας, κάτι που χρειαζόμαστε καθώς απαιτούμε από τον χρήστη να καταθέσει προσωπικές πληροφορίες. Παράλληλα, οι γειτονικοί μπλέ τόνοι είναι άνετοι στο μάτι και δεν δυσκολεύουν τον χρήστη (όπως θα έκαναν έντονα χρώματα). Για να μην είναι μονότονος ο σχεδιασμός, προστέθηκε ένα βαθύ κόκκινο στο πάνω μέρος της σελίδας. Παρά που το συμπληρωματικό του μπλε είναι το πορτοκαλί, θεωρήσαμε πως το γειτονικό του

τελευταίου κόκκινο θα επέφερε μια ιδιαίτερη, μα συνάμα διακριτική, όψη. Έστερα, για να εγγυηθούμε την μέγιστη ευχρηστία της εφαρμογής μας, τοποθετήσαμε τις διαθέσιμες επιλογές στο πάνω μέρος της σελίδας και σηματοδοτήσει την κάθε μία με ένα σύμβολο ώστε να είναι ευκολότερη η πρόσβαση από τον χρήστη.

## **6) Παραδοχές-Επεξηγήσεις**

Παρακάτω οι παραδοχές που λάβαμε για την υλοποίηση του άνωθεν κώδικα:

1η Παραδοχή: Ο μέσος όρος των επισκεπτών ενός σημείου υπολογίζεται από τον μέσο όρο των εκτιμήσεων που καταχωρούν οι επισκέπτες για το σημείο τις δύο τελευταίες ώρες πάνω κάτω.

2η Παραδοχή: Η εκτίμηση της επισκεψιμότητας για ένα σημείο επί εκατό τοις εκατό υπολογίζεται από τον τύπο: (επισκέπτες ενός σημείου/συνολικοί χρήστες)\*100.

3η Παραδοχή: Στον διαχειριστή στο ερώτημα 2b υπολογίζουμε τον συνολικό αριθμό των κρουσμάτων(ενεργών και μη).

4η Παραδοχή: Στον διαχειριστή στο ερώτημα 2e κάνουμε κατάταξη των κατηγοριών που έχουν επισκεφτεί τα συνολικά κρούσματα(ενεργά και μη).

5η Παραδοχή: Στον διαχειριστή στο ερώτημα 2.f.ii εξάγουμε το πλήθος των επισκέψεων των συνολικών κρουσμάτων(ενεργών και μη) για κάθε μέρα.

6η Παραδοχή: Ο χρήστης πρέπει να γράψει ολόκληρο το όνομα του ροι στο search bar αλλιώς αν γράψει μη ολοκληρωμένο ή λάθος ή ροι που δεν υπάρχει εκτυπώνεται κατάλληλο μήνυμα.

**Τέλος**