

1059633

Νίκηφόρος-Γεώργιος Παπαγεωργίου  
st1059633@ceid.upatras.gr

Αλέξανδρος Ξιάρχος  
st1059619@ceid.upatras.gr

1059619

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ · ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ & ΣΥΣΤΗΜΑΤΑ ΣΤΟ ΠΑΓΚΟΣΜΙΟ ΙΣΤΟ

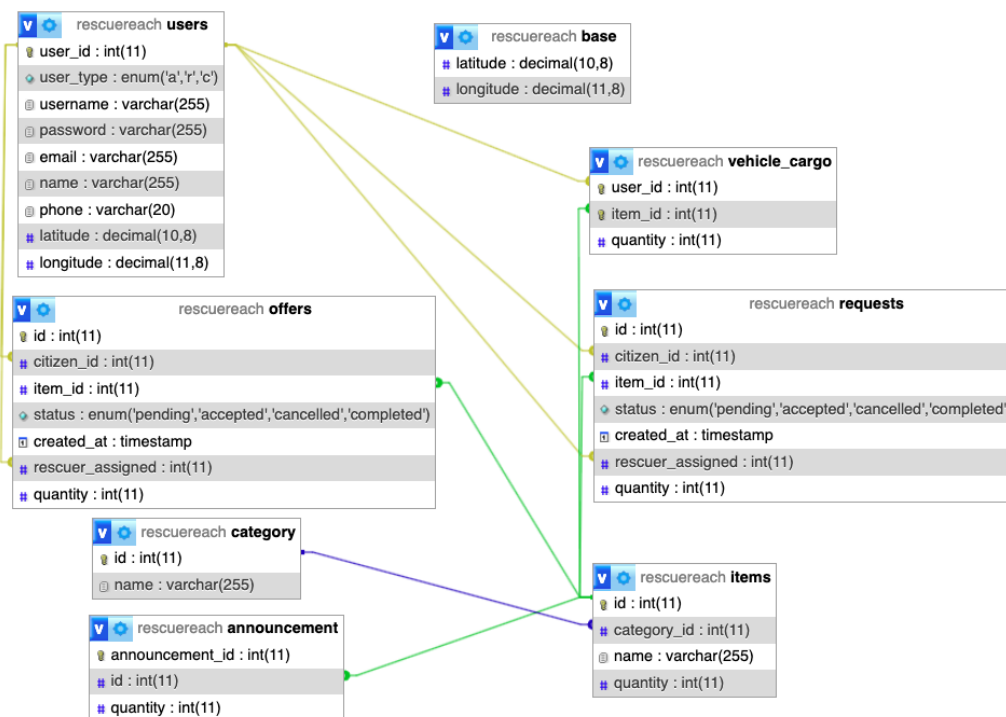
ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ · 2023 – 2024

**ΠΕΡΙΕΧΟΜΕΝΑ**

<b>1</b>	<b>ΣΧΕΔΙΑΣΜΟΣ ΒΑΣΗΣ</b>	<b>2</b>
<b>2</b>	<b>ΥΛΟΠΟΙΗΣΗ</b>	<b>2</b>
2.1	LOGIN . . . . .	2
2.1.1	login.php . . . . .	3
2.2	REGISTER . . . . .	3
2.2.1	signup.js . . . . .	4
2.2.2	register_user.php . . . . .	4
2.3	ADMIN . . . . .	5
2.3.1	Dashboard . . . . .	5
2.3.1.1	Quick Summary . . . . .	5
2.3.1.2	Create Announcement . . . . .	5
2.3.1.3	Create Rescuer Account . . . . .	6
2.3.1.4	Announcements Summary . . . . .	6
2.3.1.5	Graphs . . . . .	6
2.3.2	Base . . . . .	7

## 1 ΣΧΕΔΙΑΣΜΟΣ ΒΑΣΗΣ

Η σχεσιακή βάση μας είναι υλοποιημένη σε περιβάλλον ΧΑΜΡΡ χρησιμοποιώντας MariaDB. Παρατίθεται το στιγμιότυπο που εξάχθηκε από το γραφικό περιβάλλον του PhpMyAdmin:



Σχήμα 1.1: Entity-Relationship Diagram

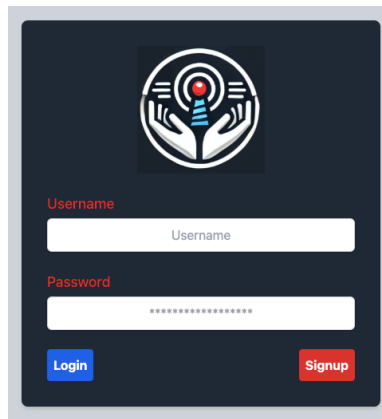
## 2 ΥΛΟΠΟΙΗΣΗ

Η εφαρμογή μας σχεδιάστηκε και υλοποιήθηκε κυρίως με χρήση vanilla τεχνικών, όπως PHP και JavaScript. Για τον χειρισμό των αιτημάτων προς τον διακομιστή, χρησιμοποιήθηκε η βιβλιοθήκη jQuery, η οποία διευκολύνει την αποστολή αιτημάτων στην PHP μέσω της μεθόδου AJAX.

Επιπλέον, για την κατασκευή των χαρτών στο frontend, αξιοποιήθηκε η βιβλιοθήκη Leaflet, που παρέχει διαδραστικούς και ευέλικτους χάρτες. Τέλος, για το responsive design της εφαρμογής χρησιμοποιήσαμε Tailwind CSS με Flexbox ώστε να έχουμε σωστή στοιχειοθέτηση σε πηληθώρα αναλύσεων και συσκευών.

### 2.1 LOGIN

Ως index.html ορίζεται η σελίδα Login, στην οποία γίνεται η σύνδεση ή η εγγραφή των χρηστών-πολιτών.



Σχήμα 2.1: Φόρμα Login

Στη βάση δεδομένων έχουν ήδη δημιουργηθεί κάποιοι λογαριασμοί πολιτών:

				user_id	user_type	username	password	email	name	phone
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	1	a	admin	admin	NULL	NULL	NULL
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	2	r	rescuer	rescuer	rescuer@gmail.com	rescuer	6912345567
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	3	c	nikf	nikhforos1!	nik@gmail.com	nikhforos	6945123321
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	4	r	rescuer2	rescuer!@#\$\$%	rescuer2@gmail.com	rescuer2	69123455678
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	7	r	rescuer3	rescuer3!	rescuer3@gmail.com	rescuer3	6912345566
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	8	c	alex	alex!@	alex@hotmail.com	Alexandros	6912345565
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	9	r	rescuer4	rescuer4!	rescuer4@gmail.com	rescuer4	6234221312
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	10	c	Aerakis	Aerakis	aerakis@gmail.com	Giannis Aerakis	6969696969
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	11	c	skasselakis	kassssss	skasselakis@gov.com	Stefanos Kasselakis	6911111111
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	12	c	Blondie420	TonYpethino	karen000@gmail.com	Karen	6987654300

Σχήμα 2.2: πίνακας users

Η συνάρτηση `login` δέχεται τα ορίσματα των `input fields` με `id = username` και `id = password`, ελέγχεται αν έχουν πληροφορία (αν δεν έχουν, στέλνεται `alert` μέσω της `SweetAlert`), και αν έχουν, στέλνονται με ένα `AJAX request` στο `login.php`.

Σε περίπτωση επιτυχίας καλείται η `handleSuccess()`. Αυτή, στο αντικείμενο που επιστρέφεται <sup>1</sup>, ελέγχει το `user_type` και κάνει το αντίστοιχο `redirect` (με ένα `delay 2 sec`) στην επόμενη σελίδα.

### 2.1.1 login.php

Αφού ανακτήσει το `username` και `password` από το `AJAX request`, στέλνει `query` στη βάση δεδομένων βάσει του `username`. Αν βρει, ελέγχει εξίσου και το `password` και στη συνέχεια δημιουργεί ένα `array` με τις πληροφορίες (`username`, `user_id` και `user_type`) του χρήστη, και ένα `cookie` με διάρκεια 30 ημερών.

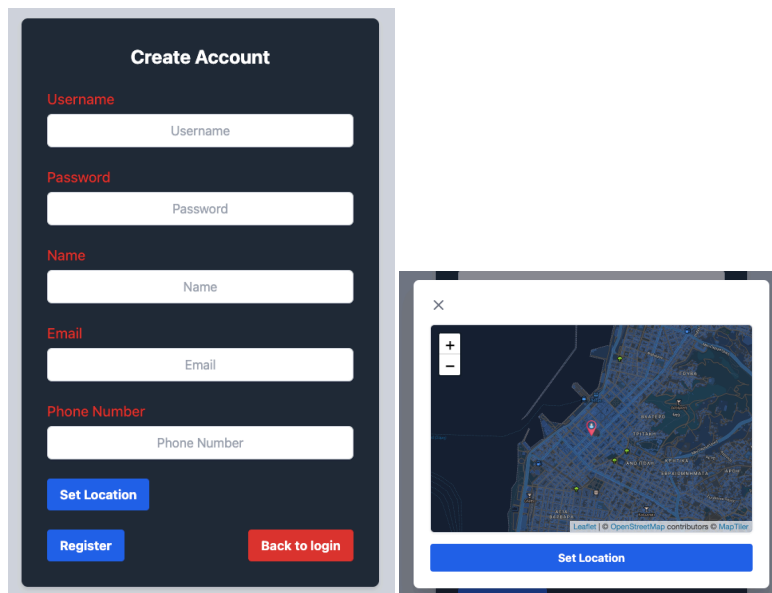
Εν τέλει, στέλνεται με `echo` ένα `JSON` αρχείο της μορφής

```
[{"username": <...>, "user_id": <...>, "user_type": <"a" ή "r" ή "c">}]
```

## 2.2 REGISTER

Στη σελίδα `signup.html` ορίζεται η φόρμα εγγραφής των χρηστών-πολιτών:

<sup>1</sup>redundant, αλλιώς ελέγχεται αν όντως επιστρέφεται αντικείμενο· αν όχι εμφανίζεται `error`



Σχήμα 2.3: Σελίδα Signup και modal προσδιορισμού τοποθεσίας

### 2.2.1 `signup.js`

Η συνάρτηση `signup()` δέχεται τα ορίσματα των `input` fields, ελέγχει αν έχουν πληροφορία (αν δεν έχουν, στέλνεται `alert`), και αν έχουν, στέλνει ένα AJAX request στο `register_user.php`.

Οι συναρτήσεις `openModal()` και `closeModal()`, εμφανίζουν και αποκρύπτουν το modal προσδιορισμού τοποθεσίας, το οποίο εμφανίζεται πατώντας στο κουμπί **Set Location**. Αυτό επιτυγχάνεται με την κατάλληλη αφαίρεση και προσθήκη κλάσεων του Tailwind CSS.

Στην εμφάνιση του **Set Location**, αρχικοποιείται ο χάρτης μέσω της `initializeMap()`. Ορίζεται η σταθερά `map` μέσω της Leaflet, με κεντραρισμένο view στο κέντρο της Πάτρας. Ορίζεται ο χάρτης που θα χρησιμοποιήσουμε ως συγκεκριμένο `tileLayer` (ο οποίος χάρτης εξάγεται μέσω API από το `maptiler.com`), και προστίθεται ένα `marker` (`userIcon` που αντιστοιχεί στην εικόνα `user.png`), το οποίο αρχικοποιείται με αρχική θέση στο κέντρο της Πάτρας (ίδιο `location` με την αρχικοποίηση του view). Ελέγχεται το event του `click`, και σε εκείνη την περίπτωση αποθηκεύονται οι αλληλαγμένες συντεταγμένες ως `value` στο `input` με `id = location`.

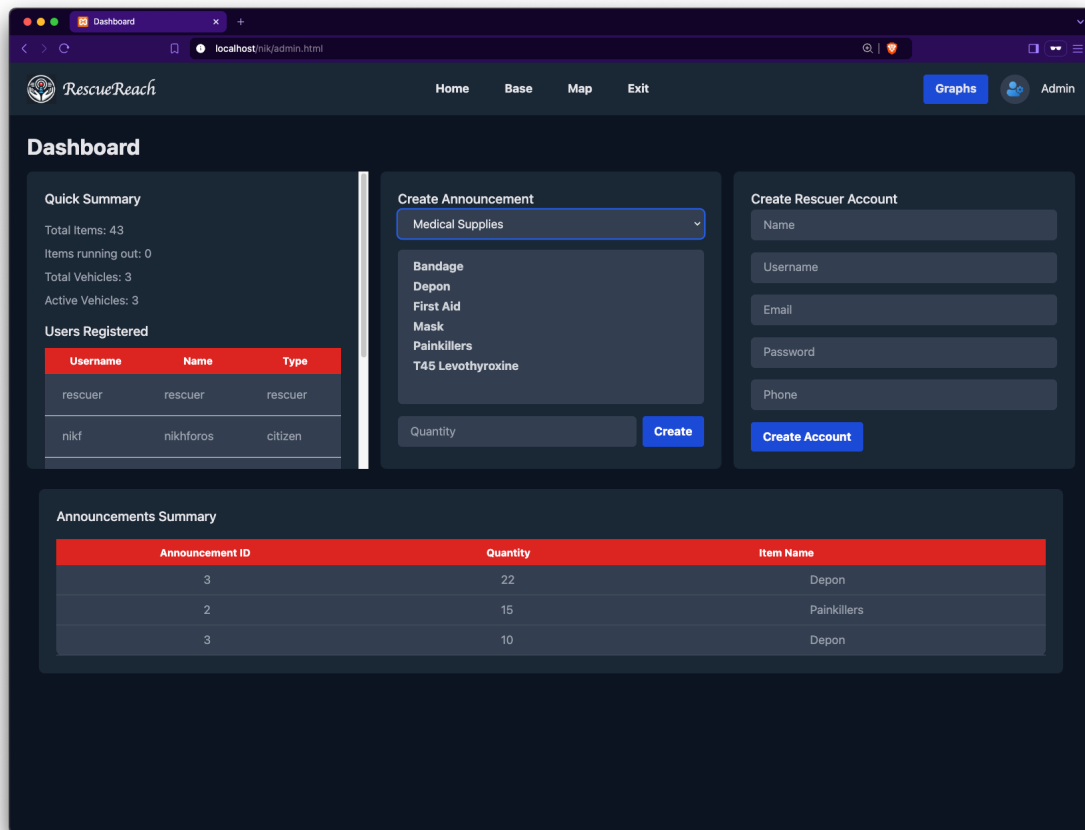
### 2.2.2 `register_user.php`

Αφού ανακτηθούν οι πληροφορίες, γίνονται `insert` στη βάση δεδομένων και γίνεται `echo` μήνυμα επιτυχίας. Στην περίπτωση της μεταβλητής της τοποθεσίας, είναι αναγκαία η διάσπασή της σε ξεχωριστές μεταβλητές για το `latitude` και `longitude`.

## 2.3 ADMIN

### 2.3.1 Dashboard

Η κεντρική σελίδα του διαχειριστή περιλαμβάνει τέσσερα διαφορετικά modules: ένα με μια καταγραφή των items, των vehicles και των χρηστών, ένα όπου μπορούμε να δημιουργήσουμε ανακοινώσεις, ένα που μπορούμε να δημιουργήσουμε ένα λογαριασμό διασώστη, και ένα με την καταγραφή όλων των ανακοινώσεων.



Σχήμα 2.4: Dashboard admin

Το αρχείο `dashboard.js` χειρίζεται τη συγκεκριμένη σελίδα. Στην αρχή καθλούνται όλες οι απαραίτητες συναρτήσεις για την εμφάνιση των διαφόρων modules.

#### 2.3.1.1 Quick Summary

Η συνάρτηση `loadDashboardSummary()` στέλνει ένα AJAX request στο `summary.php`, ζητώντας τον συνολικό αριθμό των items (`totalItems`), και τον αριθμό των items που τελειώνουν (`lowStockItems`), και επιστρέφονται σε JSON μορφή. Αντίστοιχα η `fetchActiveVehicles()` στέλνει ένα AJAX request στο `active_vehicles.php`, ζητώντας τον αριθμό των vehicles που είναι ενεργά (`activeVehicles`). Η `registeredUsers()` δέχεται δεδομένα από το `registered_users.php`, και μέσω της `populateTable()` εμφανίζει έναν πίνακα με τους εγγεγραμμένους χρήστες.

#### 2.3.1.2 Create Announcement

Για τη φόρτωση των διαφορετικών κατηγοριών των items για τη δημιουργία της ανακοίνωσης, καλείται αρχικά η `loadCategoriesAndItems()` η οποία στέλνει ένα AJAX request στο `announcement.php`. Η PHP επιστρέφει ένα JSON αρχείο της μορφής:

```
{"categories": [id, name], "items": [id, name, category_id]}
```

Τα δεδομένα που επιστρέφονται χρησιμοποιούνται από τη `loadDashboardSummary()` για να γίνουν populate οι κατηγορίες της φόρμας. Αν ο χρήστης επιλέξει μια κατηγορία, η `filterItemsByCategory()` κάνει populate τη λίστα με τα αντίστοιχα items μέσω του JSON αρχείου.

Όταν πατηθεί το κουμπί **Create**, αποθηκεύονται σε μεταβλητές τα επιλεγμένα items, όπως επίσης και το πλήθος τους και στέλνονται στη `insert_announcement.php`, η οποία τα κάνει insert στη βάση. Σε περίπτωση επιτυχίας στέλνεται το αντίστοιχο μήνυμα.

### 2.3.1.3 Create Rescuer Account

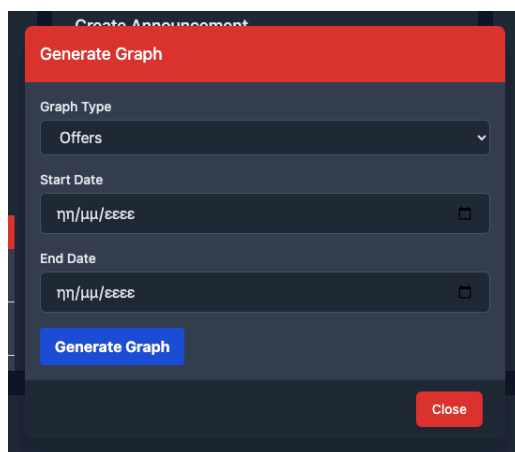
Όταν πατηθεί το κουμπί **Create Account**, καλείται η `createRescuerAccount()` η οποία μέσω Promise δημιουργεί μεταβλητές για κάθε value της φόρμας, τις οποίες στέλνει μέσω AJAX request στο `create_rescuer.php`. Η PHP ελέγχει αν υπάρχει ήδη τέτοιο username ή email, στέλνει κατάλληλα echos αν υπάρχουν, και αν δεν υπάρχουν κάνει insert τον νέο χρήστη στη βάση δεδομένων.

### 2.3.1.4 Announcements Summary

Ακολουθείται παρόμοια διαδικασία με τη δημιουργία του πίνακα των εγγεγραμμένων χρηστών. Χρησιμοποιείται η `loadAnnouncementSummary()` σε συνδυασμό με τη `get_ann.php`.

### 2.3.1.5 Graphs

Το κουμπί **Graphs** ενεργοποιεί ένα modal<sup>2</sup> με στατιστικά γραφήματα για τα requests και τα offers.



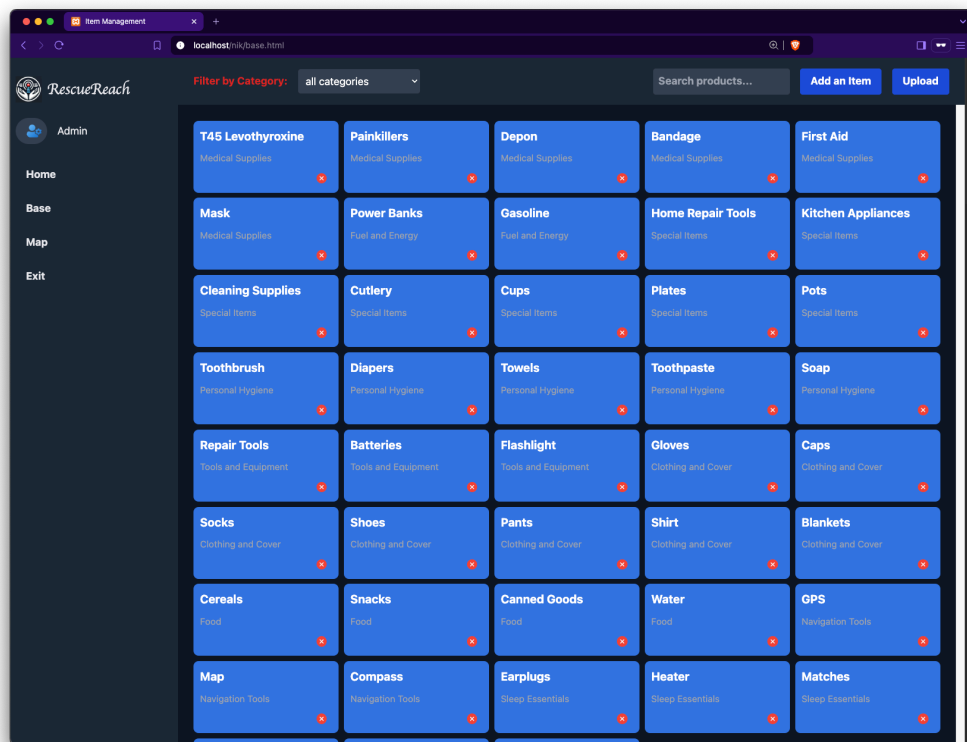
Σχήμα 2.5: Generate Graph model

Δημιουργούνται οι σταθερές `startDate`, `endDate`, και `graphType`. Μέσω της `fetchGraphData()` δημιουργείται ένα Promise ανάλογα με το αν το `graphType` είναι `offer` ή `request` προς τα `graph_<offers ή requests>.php?startDate=${startDate}&endDate=${endDate}`.

Με το JSON αρχείο που επιστρέφεται σε κάθε περίπτωση, η `fetchGraphData()` μέσω της ChartJS δημιουργεί το αντίστοιχο γράφημα.

<sup>2</sup>με κατάλληλη προσπάφαιρηση κλάσεων του Tailwind

## 2.3.2 Base



Σχήμα 2.6: Σελίδα με τα αντικείμενα της Βάσης

Το αρχείο `base.js` χειρίζεται τη σελίδα `base.html`. Κατά την εμφάνιση της σελίδας δημιουργείται ένα AJAX request στο `item_categories.php`, το οποίο επιστρέφει όλες τις κατηγορίες και τα προϊόντα της βάσης. Συγκεκριμένα επιστρέφεται ένα JSON αρχείο της μορφής:

```
{"categories": [...], "products": [id, name, category_id, category]}
```