Programming and Systems on the Web

Laboratory Exercise Academic Year 2021-2022

Subject : Development of a system of abundant data concerning the visitation of points of interest and the management of possible contact with a COVID-19 case.

DIDO MAPS

The Logo of our team.

Team Members

SPYRO SULI

MT : 1070263

FEZZO METSHI

AM : 1070727

How we worked and what we created as a team.

Welcome Page

Our team's goal was and is to complete as many questions as possible in the most efficient way possible.

Initially we thought it would be good to create an html page to become our homepage where we would present our project and our team. This page is the file :

frontend1.html

Inside this page the user can see an introduction to our project, the members of our team and their interests in relation to computer science, he can help in our project and signup using the appropriate button where we provide him with an appropriate message.Finally we have put some elements for easy communication with our team and we have created a footer.The logic of the design of the page in the aesthetic part follows minimalism and this is because all modern websites follow the same for the design of the page. The homepage works mainly as a one-page website but includes redirection buttons such as the Log In button that redirects us to the log in form , the Sign Up button that redirects us directly to the Register form and finally the favicon which is our logo that redirects us to the homepage again. The rest of the buttons redirect us to different parts of the home page. Our palette is specific and it also follows a minimalist aesthetic the main colors are dark and light blue , white and black.

Less is more.

Resposive website

One of the main requests is to make our website responsive for mobile, tablet and large screens. We implemented this part in 2 ways , the first one is shown with this code:

```
<!-- make navivgation buttons responsive -->

            <button class="navbar-toggler" type="button" data-toggle="collapse"
            data-target="#navbarResponsive">
                <span class="navbar-toggler-icon"></span>
            </button>

            <!--vale ta li koumpia -->

            <div class="collapse navbar-collapse" id="navbarResponsive">
                <ul class="navbar-nav ml-car">
                    <li class="nav-item"><a href="frontend1.html" class="nav-link
active">Home</a></li>
                    <li class="nav-item"><a id="sec-team" href="#team"
class="nav-link">Our Team</a></li>.
                    <li class="nav-item"><a href="login.php" class="nav-link">Log
In</a></li>
                </ul>

        </div>
```

We created a navigation bar toggler for the redirection buttons for mobile and tablet users and used a combination of CSS techniques such as :

```
@media(max-width: 767.98px){
    #mapid { height: 500px; width :80%; margin-left:auto; margin-
right:auto;transition: 1s;z-index: 0;}
}
```

With similar codes like the one above we were able to give commands through CSS so that when the website is used by small devices like mobile devices it will format the page accordingly to make it functional and beautiful. Note that when

the width is greater than or equal to 768 then we are talking about tablet devices or computer monitors etc. All the pages and forms we implemented are responsive except for the admin page where it was not requested since the login is done via PC only.

Login/Register Form

For the login and register forms we used php to connect to the database , mysql to create and edit the database , html , css , and javascript libraries. The forms also follow a minimalist approach and the same palette. We also added an mp4 file as background because we thought it is aesthetically beautiful.More specifically in the login form we can put the username and password to log in as a user or as an administrator and we provide the possibility to redirect as to the register form with an appropriate message button. In the register form the user registers and gains access to the system by choosing a username & password of his choice, and providing his email, The password is checked for being at least 8 characters long and containing at least one capital letter, a number and a symbol (e.g. #$*&@) and if it does not meet the requirements, the registration is not possible and an appropriate message is displayed.There is also a button to redirect to the login form.

user page

On the user page,the user can easily logout through the appropriate button and redirect through the navigation buttons that are in the header on the right side.For each function there is a redirect button.In the mobile version the redirect buttons are not present but the user will find all the functions by scrolling

down.For the user page we used php to connect to the database , mysql to create and edit the database , html , css , and javascript. Data loading from the DB was done using only AJAX techniques. Leaflet , chart js etc libraries were also used.

```
<!--javscript locations-->
<script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js"></script>
<script src="src/leaflet-search.js"></script>.
<script
src="//cdnjs.cloudflare.com/ajax/libs/fuse.js/1.2.2/fuse.min.js"></script>
<script src="src/leaflet-gps.js"></script>
<script
  src="https://code.jquery.com/jquery-3.6.0.min.js"
  integrity="sha256-/xUj+3OJU5yExlq6GSYGSHk7tPXikynS7ogEvDej/m4="
  crossorigin="anonymous"></script>
<script>
```

As requested in the queries, the user has the option to initially display the map, search for points of interest, register a visit, declare an outbreak, possible contact with an outbreak and edit a profile. The design of the page was done in a minimalist way and with the same palette.

admin page

On the administrator page, the administrator can easily logout via the appropriate button and be redirected via the navigation buttons in the header on the right side. For each function there is a redirection button.For the user page we used php to connect to the database , mysql to create and edit the database , html , css , and javascript. Data loading from the DB was done using only AJAX techniques. A

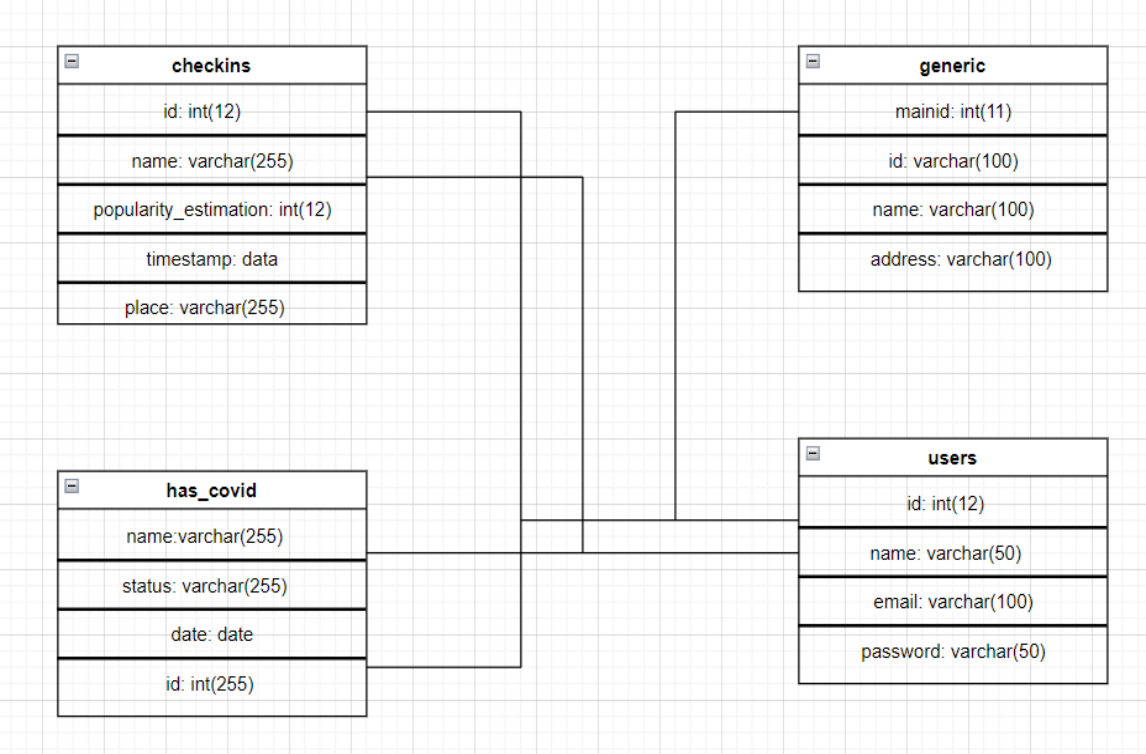
custom responsive table was also made with the database for query upload, update and delete data and table visualization was done with the help of chart js.

```
div class="table-responsive">
                    <div align="right">
                         <button type="button" name="add" id="add" data-
toggle="modal" data-target="#add_data_Modal" class="btn btn-warning">Add
Data</button>
              <button type="button" name="delete" id="delete" class="btn btn-
warning">Delete All Data</button>
                    </div>
```

.....

```
 <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"><
/script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></scrip
t>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.6.0/Chart.min.js"></script
>
```

Entity-Relationship Diagram



Database

Tables

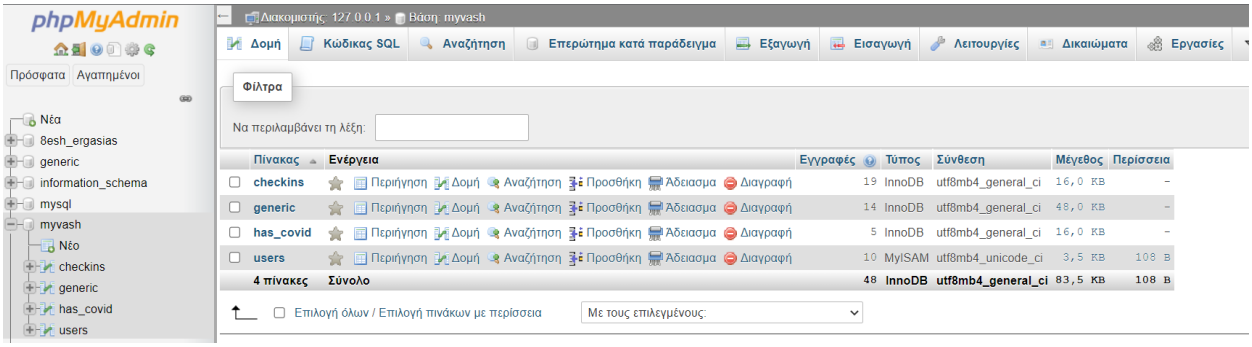


Table checkins

	#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Σχόλια	Πρόσθετα	Ενέργεια
☐	1	id	int(12)			Όχι	*Καμία*		AUTO_INCREMENT	Αλλαγή Διαγραφή Περισσότερα
☐	2	name	varchar(255)	utf8mb4_general_ci		Όχι	*Καμία*			Αλλαγή Διαγραφή Περισσότερα
☐	3	popularity_estimation	int(12)			Όχι	*Καμία*			Αλλαγή Διαγραφή Περισσότερα
☐	4	timestamp	date			Όχι	current_timestamp()			Αλλαγή Διαγραφή Περισσότερα
☐	5	place	varchar(255)	utf8mb4_general_ci		Όχι	*Καμία*			Αλλαγή Διαγραφή Περισσότερα

This table contains the columns that help us to implement user queries such as visit registration and possible contact with an outbreak.

id: is the number-identity of each user where it is unique whether he is a simple user or an administrator, updated when a new user enters the table.

name: is the alphanumeric name of each user, whether the user is a regular user or an administrator who has made a visit.

popularity_estimation: is a number that indicates how many people the user thinks are currently at a particular location.

timestamp: is a date indicating when the check in was done by the specific user

place: is an alphanumeric indicating the location of the user.

Table has_covid

	#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Σχόλια	Πρόσθετα	Ενέργεια
☐	1	name	varchar(255)	utf8mb4_general_ci		Όχι	*Καμία*			Αλλαγή Διαγραφή Περισσότερα
☐	2	status	varchar(255)	utf8mb4_general_ci		Όχι	*Καμία*			Αλλαγή Διαγραφή Περισσότερα
☐	3	date	date			Όχι	*Καμία*			Αλλαγή Διαγραφή Περισσότερα
☐	4	id	int(255)			Όχι	*Καμία*		AUTO_INCREMENT	Αλλαγή Διαγραφή Περισσότερα

This table contains the columns that help us to implement user queries such as outbreak declaration, possible contact with an outbreak.

name: is the alphanumeric name of each user, whether the user is a regular user or an administrator who checked in to the incident report.

date: is the date when the user checked in to the outbreak report.

id: is the identity number of the user who checked in to the outbreak statement.

Table generic

	#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Σχόλια	Πρόσθετα	Ενέργεια
☐	1	mainid	int(11)			Όχι	*Καμία*		AUTO_INCREMENT	Αλλαγή Διαγραφή Περισσότερα
☐	2	id	varchar(100)	utf8mb4_general_ci		Ναι	*NULL*			Αλλαγή Διαγραφή Περισσότερα
☐	3	name	varchar(100)	utf8mb4_general_ci		Ναι	*NULL*			Αλλαγή Διαγραφή Περισσότερα
☐	4	address	varchar(100)	utf8mb4_general_ci		Ναι	*NULL*			Αλλαγή Διαγραφή Περισσότερα
☐	5	types0	varchar(100)	utf8mb4_general_ci		Ναι	*NULL*			Αλλαγή Διαγραφή Περισσότερα
☐	6	types1	varchar(100)	utf8mb4_general_ci		Ναι	*NULL*			Αλλαγή Διαγραφή Περισσότερα
☐	7	types2	varchar(100)	utf8mb4_general_ci		Ναι	*NULL*			Αλλαγή Διαγραφή Περισσότερα

It was given to us by you we just put an extra column to have mainid by number and to be able to do mappings and easier finding, it is updated when entering new POI in the table.

Table users

	#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Σχόλια	Πρόσθετα	Ενέργεια
☐	1	**id**	int(12)			Όχι	*Καμία*		AUTO_INCREMENT	Αλλαγή Διαγραφή Περισσότερα
☐	2	**name**	varchar(50)	utf8_general_ci		Όχι	*Καμία*			Αλλαγή Διαγραφή Περισσότερα
☐	3	**email**	varchar(100)	utf8_general_ci		Όχι	*Καμία*			Αλλαγή Διαγραφή Περισσότερα
☐	4	**password**	varchar(50)	utf8_general_ci		Όχι	*Καμία*			Αλλαγή Διαγραφή Περισσότερα

This table contains the columns that help us implement queries such as registering and logging in to the system.

id: number-identity of the user and/or administrator who has registered in the system, updated when a new user registers in the system.

name: is the alphanumeric name of each user, whether the user is a regular user or an administrator registered in the system.

email: is the alphanumeric which holds the emails of each user, whether they are a regular user or an administrator registered in the system.

password: is the alphanumeric which holds the password of each user, whether the user is an ordinary user or an administrator registered in the system.

THANK YOU VERY MUCH !