Candidate number: 089789

Date: 07/04/2021

**Web Development coursework**

Application URL:<http://ml-lab-4d78f073-aa49-4f0e-bce2-31e5254052c7.ukwest.cloudapp.azure.com:62164/index.php>

**HTML pages**

HTML forms were used in the Login, Registration**,** Add Visit, Report and Settings Page to redirect the user to the appropriate pages (e.g. if the users press the “Login” button in the Login Page or the “Register” button in the Registration Page, they are redirected to the Home Page).

Validation was performed using alert boxes which display error messages on the top of the screen (e.g. if invalid credentials were typed in the Login Page, an alert was showing up at the top of the screen). Also, the “required” attribute of the “input” tag was used, so that the program asks the user to fill in the required text fields, in case they had remained empty.

**CSS for Page Styling**

Page Styling was performed using css files as well as the “style” attribute of the “div” tag. For text fonts, the font-size and font-family features were used to select the size and the style of a text.

The “header” class was retrieved from all css files used for that application and was used to style the “COVID-19 Contact Tracing” header using the display, justify-content, align-items, height, background-color features.

For the Home, Overview, Add Visit, Report and Settings pages, there were two main classes apart from the “header” one, the “main-screen” and the “sidebar” classes. The “sidebar” class was used to style the sidebar of each of these pages whereas the “main-screen” class styled the main content of each page.

For colors, two features of the style attribute were used, the background-color and the border-color.

For the watermark, there were a number of features used:

* **background-image:** used to load the watermark.png file
* **background-position:** used toplace it in the correct position on the screen
* **background-repeat**(value=no-repeat): used to display it only once.
* **Background-blend-mode**(value=lighten): used in combination with the background-color feature to reduce its opacity.

**PHP for server-side scripting**

PHP was used to:

* Connect with the database using the mysqli() function
* Check whether the user ID cookie was empty and if so redirect the user to index.php.
* Check the validity of the Login credentials by retrieving username and password input and comparing them with the data that already exists in the database. Data retrieval was performed with the use of $\_POST variables and comparison with the database was achieved with an SQL query.
* Register users to the application by posting the information retrieved from the textfields (Name, Surname (optional), username, password) to the database. This was achieved with the use of an SQL query and $\_POST variables.
* Check whether a username already exists in the Registration Page following the same prodecure as in the above bullet point.
* Add visits, report infections and change settings using $\_POST variables and SQL queries
* Create an overview of the visits at the Overview Page by looping through every visit in the database

**Secure sessions and cookies**

The PHP setcookie() function was used in the index.php page to create cookies. The cookies were then accessed by $\_COOKIE variables.

The logout.php file was used to remove cookies and redirect the user to the index.php page, every time they logged out from the app.

**JavaScript and Ajax**

A function called deleterowFunction() which received an id as an argument, was used at the Overview Page to delete a visit from the Overview table.

A file called deleteVisit.php, included a script to find the id of a row and delete it from the table. An AJAX call was made at that file from the HomePage.php requesting the id of the row to be deleted and then removing it from the Overview table.

Moreover, in order to select a location from the map in the Home Page, a function called triggerInfo() was used. Its arguments date, time and duration were the information displayed in the form of an alert box on the user’s screen, every time a user clicked on a marker.

**Security is implemented**

The password\_hash() and password\_verify() PHP functions were used to encrypt and decipher users’ password. For cross site scripting a PHP script [[[1]](#footnote-1)] stored in a PHP file called xss.php was used.

For SQL injections the real\_escape\_string() function was used.

Both functions were implemented to secure every text-field of the application, however, the real\_escape\_string() function was also implemented to secure $\_COOKIE variables.

1. <https://portswigger.net/web-security/cross-site-scripting/preventing> [↑](#footnote-ref-1)