Contact Tracing App Cover Sheet- 700048312

Link to website:

http://ml-lab-4d78f073-aa49-4f0e-bce2-31e5254052c7.ukwest.cloudapp.azure.com:61958/index.php

The index.php redirects the user to the login.html page.

To access the website you must switch on its associated virtual machine.

How I have accomplished what the specification is asking for:

All of the HTML pages and static content are present and validated, in some cases using JavaScript and/or PHP when necessary.

The CSS styling conforms completely to the style guide given in section B of the specification.

The PHP is fully functioning for all of the requirements of the specification.

A session is created after login and destroyed after logout, and cookies are used to store all settings plus the username of the user that is logged in.

The JavaScript is correct for removing a location from the overview and for selecting a location from the map. There is also much more JavaScript in use, such as to set the maximum input date as today's date to avoid users inputting future dates for infections and visits. Another example is using JavaScript to enable a checkbox to toggle visibility of password input. JavaScript is also heavily used for form validation, alongside PHP.

Ajax is used to remove visits from the overview table and the database. The asynchronous ajax call works, as does the corresponding PHP code.

The web service is used both to report and to check for infections. Reported infections are also stored in the database in a table called 'infections'. The other tables in the database are named 'users' and 'visits'.

Security measures are fully implemented. Any user-input variables being used in an SQL statement are sanitised with 'mysqli_real_escape_string()' before use, and prepared statements are used for INSERTs. These measures prevent both cross-site scripting and SQL injection attacks. Further measures against cross-site scripting include json encoding xmlhttp responses and avoiding the use of HTTP GET requests apart from getting infection data from the external web service. Passwords are protected via the use of a hashing algorithm and adding salt. Options is also used to increase cost to 10 to slow down password verification, protecting against brute force attacks. There are also a multitude of checks to prevent weak passwords from being used, including checking the password against a list of common passwords, helping to avoid dictionary attacks and lookup tables.