Link:
http://ml-lab-4d78f073-aa49-4f0e-bce2-31e5254052c7.ukwest.cloudapp.azure.com:53959/

Points Covered:
HTML:
- All static pages were rendered with various ways of ensuring the correct proportions, before later on, I decided to shift to the display: grid style, and adjusting my HTML around that. I ensured that the types entered were of a certain format by using the type tag in the inputs

CSS:
- As I mentioned in the HTML section, I began with a much more static approach, before shifting over to reflexive styling. I was given the fonts, the colours, and the watermark to work with, and it was all a matter of estimating where the boundaries lied on the given images

PHP:
- By utilising HTML forms, I was able to build outwards to construct the register and login pages, before moving into SQL databases to store the data from the report pages. For the settings, I used cookies, and overall I stored the username and id in session variables.

Secure sessions and cookies:
- After constructing sessions in a user's login, I would store the username and id there until they logged out, and when the next user logged in, the previous session was destroyed
- I used cookies to store the window and distance settings for a week, and in the case that a user's preference hadn't been set, the default values were chosen.

JavaScript:
- By calling functions using the onclick property in HTML, I was able to gather the positions of that click and set a marker to that spot.
- Also, by using AJAX intertwined with the JavaScript, I could delete an element by making an XMLHTTPRequest from a JavaScript function.

AJAX:
- AJAX was used to properly send out the XMLHTTPRequest, and PHP was used to handle it on the other side

Web Service:
- Both databases were used to store/get infections for the database of users. The infections database was used on the homepage to show infections on the page.

Security:
- By using password_hash, combined with the BCRYPT encoding, I salted the password before preventing SQL injections by using htmlspecialchars to escape suspicious characters.
- I prevented off-site scripting by using stmt_prepare and circumventing the possibility of someone manually changing variables.