Software Implementation and Testing Document

For

Group <20>

Version 1.0

Authors:
Chelsea Mensah
Brady Henderson
Sebastian Metellus
Ryan Fontaine
Nevin Ramjitsingh

1. Programming Languages (5 points)

HTML, CSS, JavaScript, MySQL.

2. Platforms, APIs, Databases, and other technologies used (5 points)

FullCalendar API (Sebastian Metellus)

- A JavaScript library that provides a customizable calendar with various renders allowing for user interaction and event planning, such as specifying various calendar renders. This was used in the Dashboard component with the 'Dashboard.js' file being used to import the scripts and rendering from the scripts in the 'index.html'. The 'Dashboard.js' also entails 'FullCalendar' and 'dayGrid' plugins to load in the calendar into the dashboard.

Figma

- Figma is a collaborative web application for interface design. We have used it in our project to design the landing page, nav bar, and login page. We rendered the designs in HTML and CSS. We put the HTML code in React components and the CSS code in either the index.css file or separate CSS files specified for the specific component.

React

- React is a free and open-source front-end JavaScript library for building user interfaces based on components. React is how we have the components for our webpage and it is also how we have been able to set up the routing of the web app.

Node.js

- Node.js is a cross-platform, open-source server environment. Node.js is a back-end JavaScript runtime environment, that runs on the V8 JavaScript engine and executes JavaScript code outside a web browser. We are using it to test and make sure our website is building up to our standards.

3. Execution-based Functional Testing (10 points)

1. The buttons on the Nav Bar must route the user to the corresponding page.
    a. We have tested this by using the navbar and going from the home page to the About page and to the login page by clicking on their respective buttons on the navbar.
2. The user must be able to use the login prompt to authenticate themselves on the Login page.
    a. A user is able to input a username and password in the login page, we will work on the actual authentication when we link the database to the login page.

The rest of the functional requirements will be tested as we fully implement the functional requirements into our project

4. Execution-based Non-Functional Testing (10 points)

*Describe how/if you performed non-functional testing for your project (i.e., tested for the **non-functional requirements** listed in your RD).*

1. Pages should render consistently across different screen sizes.\
    a. We have been working on testing the webpage on different screen sizes and tweaking the designs for what we need.
2. Users should not be able to view other users' information.
    a. Each user can only view the application from their perspective and can't see other users as there are no user interactions on the website.

5. Non-Execution-based Testing (10 points)

Our code was peer-reviewed as it was developed, not much testing outside of that yet.