Software Implementation and Testing Document

For

Group <1>

Version 2.0

Authors:
Michael Clark
Nathan Waring
Christian Riley
Phillip Sheng
Jose Ibarra

1. Programming Languages (5 points)
List the programming languages use in your project, where you use them (what components of your project) and your reason for choosing them (whatever that may be).

Python, HTML, CSS, JavaScript

These languages were chosen for their compatibility and ease of use for development. Most team members in the group are also familiar with the languages listed making it ideal for us to use them in our project.

2. Platforms, APIs, Databases, and other technologies used (5 points)
List all the platforms, APIs, Databases, and any other technologies you use in your project and where you use them (in what components of your project).

SQLite database, Flask, TheOdds API

We created a web application using Flask and store users and other relevant data using a SQLite database. For pulling sports matches and betting odds, we have chosen to use TheOdds API as they have a free option available for use that gives us the data that we need.

3. Execution-based Functional Testing (10 points)
Running the Flask-based application locally to then manually test out the website. Making sure that if you press a button or a hyperlink you are then sent to the appropriate render of an HTML or outcome populate.

Creating test programs for making API calls using the programs from the TheOdds API website as basis. Confirmed that we can access an API key for free and pull odds using Python.

4. Execution-based Non-Functional Testing (10 points)
Running the Flask-based application locally to then manually test out the website, even though some components such as username and email duplication check, encryption password, ability to reset password, etc. Has not been functional components these features can be checked on both the front end and back end. For example, the back end can be used to check if the passwords were properly saved and retrieved in an encrypted format. For the duplication check, if the emails or the usernames are duplicated then they can get feedback specifying that the username or email has been taken.

5. Non-Execution-based Testing (10 points)
By reading through the codebase, we were able to check for errors and walk through to see how we can make it simpler for all of our team's understanding. For example, in the beginning we had implemented a MySQL database, but after initial non-execution-based testing (aka the team was trying to install the MySQL server on their local devices) was deemed too complex and unnecessary and therefore we decided to move our database to SQLite.