Testing plan

1. Introduction
    a. Objective: By implementing a testing plan for our project we will ensure a consistency throughout the development process, while also promoting test-driven development practices.
    b. Scope: The scope for the testing is mainly focus on all of the user stories we are implementing. Before release we want to ensure that all our features are thoroughly tested, with a high level of coverage and test automation to ensure that the app remains healthy long into the future.
2. Testing Strategy
    a. Testing Levels
        i. Unit testing: After implementing any new code unit, we must develop a test file that tests the code to ensure it is working as intended.
        ii. Integration testing: Once units of code are merged in our code base, we will conduct both black and white box testing to ensure that the code works after integrating.
        iii. System testing: Going a step further from integration testing, our system testing process will involve writing test cases involving multiple units interacting with eachother, ensuring working code. The amount of tests written we be based on any common user interaction that involves communication between two or more units of code.
    b. Risks: While different levels and types of testing each have their own risks, we will be taking an approach that involves multiple testing strategies that will allow for us to overcome the downsides of some testing methods.
3. Testing Environment
    a. Software Requirements: Most of our testing will be done in VS Code, Python, SQL Lite and through the live code environment. We will be able to write test cases in Python, and can do QA work in the live environment to find and fix bugs throughout the development process.

    b. Managing Test Data: Throughout the development process we will be creating test data in the backend. For SQL Lite we will create test login data to test the login and other features with mock data. For Python testing we will use assert statements and create mock user interactions that would allow for testing to be most effective.
4. Objectives
    a. Testing Coverage: For any Python code we will want at minimum 90% code coverage from our corresponding test files. For database testing we want to ensure that each column or data has been tested individually. Lastly for front end testing we will implement User / QA testing where the team will test the front-end functionality after every code integration. The requirements for this will be specific based on the code being merged.
    b. Goals: Through conducting specific testing in this manner our code should remain healthy throughout the development process.
5. Prioritization and Management
    a. Tracking: We will track testing on the Kanban board and within the specific issues on the board. Each programming task will have a test file requirement to ensure that the code has been tested properly, the requirements will differ depending on the type of programming task that was undertaken. Also upon a pull request the team will perform user testing to ensure that all requirements have been met and test the code on their local machine to ensure everything is functioning as it should.
    b. Prioritization: Testing and pull requests will be FIFO, meaning that as they come in we will test the teams code and ensure that there is no backlog. As code is being written we will follow test-driven development so before any major programming we should have the proper tests in place to ensure that the code written will meet the task requirements.
6. Test Automation
    a. Tools: Using Github actions we will implement test automation.

    b. Scope: The scope for our test automation will mainly cover the tests in our test files and some integration testing to ensure that each time new code is added, that the system as a whole still works properly.
7. Approvals and Reporting
    a. Approval Process: When creating a PR we will ensure that at least 2 team members approve of the code, with at least 1 performing QA tasks to test the code.
    b. Test reporting: We will communicate testing process on the Kanban board and on PR's with comments to keep the team updated throughout the testing process.