Test Plan

Overall Test Plan

Our testing approach for the web application is designed to ensure the reliability, functionality, and performance of the system. The testing will be conducted in two phases: individual component testing and integrated system testing. First, as we develop the application, we will ensure each component of the entire solution performs as expected using small unit tests for each piece of functionality. Closer to completion, we will be testing our system and its integrations, utilizing UI automation tests on the TaskHero website itself. These tests will use the front-end to manipulate data in the back end, where we can guarantee that changes were properly propagated throughout our application.

Test Case Descriptions

1. Web Functionality Test 1
    a. This test ensures that the home web page is working correctly and that all the functionality is working as it should.
    b. Runs a series of tests to ensure that each element of user interface is working correctly, such as drop-down lists, scroll bars, page dimensions, etc. These tests will be done by simply testing their functionality doing test runs for the pages desired use.
    c. Inputs: Any of the desired inputs for the said elements, so on the home page most of the inputs would be non-applicable as it would more likely be cursor implemented re-directs to another page.
    d. Outputs: That the elements would redirect the user to the desired webpage, and that changes on other webpages would be represented on the home screen (if desired).
    e. Normal
    f. Whitebox
    g. Functional
    h. Unit
    i. Result: each element displayed the correct redirect for the user and/or the elements displayed the desired change on the home page.
2. Web Functionality Test 2
    a. This test ensures that the login page is working correctly, and all functionality is working as it should.
    b. Runs a series of tests to verify that the login page accepts valid credentials, rejects invalid ones, and securely handles user authentication. This includes testing for password encryption, account lockout mechanisms, and any multi-factor authentication processes.

    c. Inputs: Valid and invalid username/password combinations, as well as variations in input format to assess the system's robustness.
    d. Outputs: Successful login redirects the user to the designated landing page, while failed attempts display appropriate error messages. Additionally, check for the proper handling of account lockouts and the functionality of any multi-factor authentication processes.
    e. Normal
    f. Whitebox
    g. Functional
    h. Unit
    i. Result: Successful login attempts lead to the correct landing page, failed attempts show accurate error messages, and the system handles security features appropriately.
3. Web Functionality Test 3
    a. This test ensures the settings page is working correctly, and all functionality is working as it should.
    b. Runs tests to ensure that users can change settings, save preferences, and that these changes are accurately reflected throughout the system. This involves checking dropdown menus, radio buttons, and any other interactive elements.
    c. Inputs: Various settings changes, such as theme adjustments, notification preferences, or account-related configurations.
    d. Outputs: Settings changes should be saved and immediately visible, impacting the user experience accordingly.
    e. Normal
    f. Whitebox
    g. Functional
    h. Unit
    i. Result: Settings modifications are successfully saved, and the system responds appropriately to changes, ensuring a seamless user experience.
4. Web Functionality Test 4
    a. This test ensures the leaderboard page is working correctly, and all functionality is working as it should.
    b. Runs tests to verify that the leaderboard updates in real-time, accurately reflecting user performance. This includes checking for sorting mechanisms and the visibility of user profiles.
    c. Inputs: None applicable, as this test primarily focuses on observing the dynamic nature of the leaderboard.
    d. Outputs: The leaderboard should display current and accurate rankings based on user activities.
    e. Normal
    f. Whitebox
    g. Functional
    h. Unit

    i. Result: The leaderboard updates in real-time, providing accurate rankings and maintaining a smooth and responsive interface.
5. Web Functionality Test 5
    a. This test ensures the friend search page is working correctly, and all functionality is working as it should.
    b. Runs tests to ensure that the search functionality accurately retrieves user profiles, and the friend request mechanism works as intended.
    c. Inputs: Various search queries to test the system's ability to locate friends based on different parameters.
    d. Outputs: Search results should display relevant user profiles, and friend requests should be sent and received successfully.
    e. Normal
    f. Whitebox
    g. Functional
    h. Unit
    i. Result: The friend search page effectively retrieves user profiles, and the friend request system functions smoothly, facilitating easy connections between users.
6. Web Functionality Test 6
    a. This test ensures the created account page is working correctly, and all functionality is working as it should.
    b. Runs tests to confirm that the account creation process is secure, user-friendly, and that account details are stored appropriately.
    c. Inputs: User registration details, including username, password, email, etc.
    d. Outputs: Successful account creation leads to user login credentials, and the system should accurately store and manage newly created accounts.
    e. Normal
    f. Whitebox
    g. Functional
    h. Unit
    i. Result: The account creation process is secure, efficient, and new accounts are successfully integrated into the system for user access.
7. Database Query Test
    a. This test ensures that the data is being pulled from the database correctly.
    b. Using different stored procedures, we will test to see if the correct data is being pulled from the database for the web application.
    c. Inputs: using the keywords and table names from the database from the stored procedures.
    d. Outputs: the data that has been pulled and manipulated from the database.
    e. Normal
    f. Whitebox
    g. Functionality
    h. Unit
    i. Results: the obtained results match the expected ones.

8. Azure Database Connection Test
    a. This test will make sure that the Azure database is configured properly with the Angular application.
    b. With this test, we ensure that two separate components of our application can communicate properly and that we have a safe and secure method of propagating data throughout our systems.
    c. Inputs: None; This simply checks our configuration of two systems and their ability to interact.
    d. Outputs: We expect to see a successful connection between the Angular application and Azure database.
    e. Normal
    f. Whitebox
    g. Functional
    h. Integration
    i. Results: the results obtained match the expected output.
9. User Personalization Test
    a. Ensuring that the user can personalize their account, hence providing feedback on the front and back-end functionality.
    b. Runs a series of tests to ensure that users can customize their profiles, preferences, and settings. This includes testing the functionality of features such as theme selection, notification preferences, and any other personalization options available.
    c. Inputs: Various personalization choices, such as theme changes, notification preferences, and any other customizable settings.
    d. Outputs: Changes should be reflected immediately, impacting on the user interface and experience accordingly. Additionally, backend data storage and retrieval should accurately manage personalized user data.
    e. Normal
    f. Blackbox
    g. Functional
    h. Integration
    i. Result: Users can successfully personalize their accounts, with changes immediately visible on the front-end, and the system accurately stores and retrieves personalized data on the back end.
10. User Login Test
    a. This test ensures a user can log into their account and view saved data.
    b. Runs tests to verify that the login process works correctly, including the acceptance of valid credentials, handling of invalid ones, and secure authentication. Additionally, it assesses the system's ability to retrieve and display saved user data upon login.
    c. Inputs: Valid and invalid username/password combinations for login testing.
    d. Outputs: Successful login redirects the user to the appropriate landing page, and saved data associated with the user account should be displayed.

e. Normal
f. Whitebox
g. Functional
h. Unit
i. Result: Users can log in securely, with successful attempts leading to the designated landing page and the accurate display of saved user data. The system handles authentication and data retrieval effectively.

11. Back-end functionality test/unit tests
    a. These tests will ensure that each of the algorithms used to retrieve and manipulate the webpages data are working correctly.
    b. Running a series of unit tests on each of the pages backend functionality to test if the data is being manipulated and delt with correctly as it is intended to be.
    c. Inputs: the data from the database that is being delt with in these algorithms. These will be fixed for testing purposes.
    d. Outputs: the desired fixed result via the algorithm so that we can compare to what the expected value would be.
    e. Normal
    f. Blackbox
    g. Functional
    h. Unit Test
    i. Results: the obtained outcomes match the expected outcomes.
12. Web Application Speed Test
    a. Tests the speed of the webpage redirects and data retrieval.
    b. Using the application “dotcom-monitor” which allows us to test the speed from different areas around the globe. We will use this to see how long it takes for the webpage to redirect to different pages, load user data, and retrieve data from the database.
    c. Inputs: there are not specified inputs other than running the redirects and data retrieval.
    d. Output: the speed of this functionality.
    e. Abnormal
    f. Performance Test
    g. Blackbox
    h. Results: the overall performance of the website.

Test Case Matrix

Test Case ID	Normal/Ab.	Black/Whitebox	Funct/Perform	Unit/Integr.
1	Normal	Whitebox	Functional	Unit
2	Normal	Whitebox	Functional	Unit
3	Normal	Whitebox	Functional	Unit
4	Normal	Whitebox	Functional	Unit

5	Normal	Whitebox	Functional	Unit
6	Normal	Whitebox	Functional	Unit
7	Normal	Whitebox	Functional	Unit
8	Normal	Whitebox	Functional	Integration
9	Normal	Blackbox	Functional	Integration
10	Normal	Whitebox	Functional	Unit
11	Normal	Blackbox	Functional	Unit
12	Abnormal	Blackbox	Performance	Unit