

Testing the dictionary feature:

A user should receive information concerning a word (i.e. word definition) when inputting and submitting a valid word

| Test No. | Test Case    | Steps   | Test data:   | Expected Behavior  | Testing Environment   |
|----------|--------------|---|--|--|---|
| WORD 001 | Valid Word   | <ol style="list-style-type: none"><li>1. Login</li><li>2. Click on the dictionary tab</li><li>3. Enter a word into the search bar</li><li>4. Click Search</li></ol> | User: Test<br>Password: 1234<br>Input Word: Potato         | User should be able to see the definition for the word potato        | Localhost, host locally on a computer and run the tests using Chai / JavaScript tests |
| WORD 002 | Invalid Word | <ol style="list-style-type: none"><li>1. Login</li><li>2. Click on the dictionary tab</li><li>3. Enter a word into the search bar</li><li>4. Click Search</li></ol> | User: Test<br>Password: 1234<br>Input Word: aeiofaiojc ads | Should receive a blank field or a message that says No Results Found | Localhost, host locally on a computer and run the tests using Chai / JavaScript tests |

Testing the Hangman Game Feature:

The game should function as a typical game of hangman. If the user guesses a wrong letter / word, a part of the stickman hang appears and when all parts of the stickman appear, the game ends for the user and the user loses. If the user guesses the right word

/ letters, parts of the word will appear and if the user manages to fill in the entire word, the user will win and user statistics will be updated accordingly.

| Test No. | Test Case | Steps  | Test Data:  | Expected Behavior  | Testing environment   |
|----------|-----------|--|---|--|---|
| STICK001 | Win Case  | <ol style="list-style-type: none"> <li>1. Login</li> <li>2. Click On Play Hangman</li> <li>3. Input Ward</li> <li>4. Input Wurd</li> <li>5. Input Word</li> </ol>                        | <p>Create a sample hangman game where the word is "word" and the computer guesses word in 3 tries</p> <p>Ward<br/>Wurd<br/>Word</p> | User wins the game, i.e. a victory screen and user stats are updated accordingly to how many guesses it took | Localhost, host locally on a computer and run the tests using Chai / JavaScript tests |
| STICK002 | Lose Case | <ol style="list-style-type: none"> <li>1. Login</li> <li>2. Click on Play Hangman</li> <li>3. Input Wurl</li> <li>4. Input Wool</li> <li>5. Input Wirl</li> <li>6. Input Woof</li> </ol> | Using the same sample hangman game as before where the sample word is "word", the computer incorrectly inputs 6                     | User loses the game i.e. a game over screen  | Localhost, host locally on a computer and run the tests using Chai / JavaScript tests |

|  |  |                                |   |  |  |
|--|--|--------------------------------|---|--|--|
|  |  | 7. Input Weel<br>8. Input Wall | incorrect inputs<br>thereby losing the game |  |  |
|--|--|--------------------------------|---|--|--|

Testing accept friend request function:

When friend request pop up appears, if the user clicks accept friend request, the pop up goes away and the new friend is added to their friends list.

| Test No. | Test Case             | Steps   | Test Data   | Expected Behavior   | Testing Environment   |
|----------|-----------------------|---|---|---|---|
| FREND001 | Accept Friend Request | <ol style="list-style-type: none"> <li>1. Log in as Testuser1</li> <li>2. Click on User Search</li> <li>3. Enter Testuser2</li> <li>4. Click on Send Friend Request</li> <li>5. Log out</li> <li>6. Login as Testuser2</li> <li>7. Click on Friend Invites</li> <li>8. Click Accept Friend</li> </ol> | Two Users Testuser1 and Testuser2, Testuser2 receives a friend request from Testuser1 | After clicking on Accept Friend Request, Testuser1 and Testuser2 should be in both of their respective friend lists | Cloud host with two computers, user1 sends a friend request to user2 and user2 should receive a friend request in their friend request menu |

|          |                       | Request   |   |   |   |
|----------|-----------------------|---|---|---|---|
| FREND002 | Reject Friend Request | <ol style="list-style-type: none"> <li>1. Log in as Testuser1</li> <li>2. Click on User Search</li> <li>3. Enter Testuser2</li> <li>4. Click on Send Friend Request</li> <li>5. Log out</li> <li>6. Login as Testuser2</li> <li>7. Click on Friend Invites</li> <li>8. Click Reject Friend Request</li> </ol> | Two Users Testuser1 and Testuser2, Testuser2 receives a friend request from Testuser1 | After clicking on Decline Friend Request, The friend request is rejected / discarded and the two users should not be in each others friend list | Cloud host with two computers, user1 sends a friend request to user2 and user2 should receive a friend request in their friend request menu |

Login tests: As a user, upon entering valid login credentials, user should be logged in and have the homepage rendered for them.

| Test No. | Test Case        | Steps  | Test Data                            | Expected Behavior   | Testing Environment   |
|----------|------------------|--|--------------------------------------|---|---|
| LOG001   | Successful Login | <ol style="list-style-type: none"> <li>1. Input Logintest for the username field</li> <li>2. Input pass123 for the password field</li> </ol> | User: Logintest<br>Password: pass123 | The User is successfully logged in and the home page is rendered for them | Localhost, host locally on a computer and run the tests using Chai / JavaScript tests |

|        |              |  |  |  |   |
|--------|--------------|--|--|--|---|
|        |              | 3. Click Login   |  |  |   |
| LOG002 | Failed Login | 1. Input Logintest for the username field<br>2. Input banana1 for the password field | User: Logintest<br>Wrong Password: banana1 | An error message pops up saying wrong username or password | Localhost, host locally on a computer and run the tests using Chai / JavaScript tests |

User Acceptance Testers: User Acceptance Testers can be friends of the developers or other members of the class. Some of these tests are automated however it is still good to have actual people experiment / input various tests to make sure the program is up and running properly.