# Testing

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Number | Test Type | Test Name | Test Purpose | Test Data | Expected Outcome | Actual Outcome | Pass/  Fail |
| 1 | Unit Test | Login | To see if a user can log in | Correct Email and Password entered | Should log in successfully and redirect to the dashboard | User is logged in and redirected to a welcome page with a button to continue to the dashboard. | Pass |
| 2 | Unit Test | Registration | To see if a student or teacher can create an account | A valid email, password, and full name | User should be created successfully and redirect to the login screen | User is created and redirected to a slightly different welcome screen | Pass |
| 3 | Unit Test | Modals | To see if modals can be created and displayed using jQuery AJAX, PHP and the BladeOne template engine | Click on a task entry to open the modal | Modal should be created with all correct data and be sent to the client using jQuery’s $.ajax() method | Task modal content is generated dynamically using an AJAX post system I have written. | Pass |
| 4 | Interface Test | Navigation Bar | To see if a user can navigate between pages smoothly | Click on a link to a different page | Current page should fade out and new page should fade in | Page will fade out, then, the new page will load and fade back in. | Pass |
| 5 | Integration Test | Login System (Login, Register, Logout) | To see if a user can log in, log out and register with the service | Create an account and log in, then, log out | User should be able to create, log in and log out of their account without error | User can create an account, log in and log out without error. | Pass |
| 6 | Unit Test | Task Setting | To see if a teacher can assign a task to a teaching group | Open the Task page as a teacher and assign a task to a teaching group | All students in the teaching group should be able to see the set task along with details about it | Teacher can view and assign tasks to a teaching group or student. | Pass |
| 7 | Unit Test | Login | To see if the incorrect password system works properly | Enter an invalid password 4 times | User should be unable to try again for 5 minutes. | Password attempts will not increment, meaning passwords could be brute forced, This is a security issue. | Fail |
| 8 | Unit Test | Login | To see if a user can register with an invalid password (too short or too long) | Submit with an empty password or with one that’s too long | Password icon should turn orange and text should read: “Password entered is invalid” | Icon changes colour which shows that the password is being verified but text does not change from default text. | Fail |
| 9 | Integration Test | Tasks | To see if a student can view, fill out and complete a task. | Submit task with one correct answer and the rest random. | Page should reload and task should now be in the feedback panel with the text “Awaiting Feedback” next to it. | Page fades on reload, task is moved to the feedback section and points are added in the leader board accordingly | Pass |
| 10 | Unit Test | Tasks | To see if an appropriate error is displayed when a task is run without a corresponding entry file | Attempt to start task without an entry on the server | Page should display a user-friendly error message “This task could not be started, please try again later” preferably, the task would be greyed out after this. | Verbose PHP error is displayed. | Fail |

# Write-up

In the design stages and when thinking on how to implement a successful prototype, I decided that the user interface must be simple and easy to understand, with as little clutter as possible. To do this I spent time looking at different background styles and fonts before I decided on the final look. I made sure that any third-party assets where either open source or had a license that would allow me to use it freely. Not following a license could bring along legal implications or may just be viewed as unethical (not giving credit to the original creator.)

Graphical user interface, website

Description automatically generated

For the home page I used free, easy to understand icons that are relative to the text. These are from bootstrap. I tried to keep the text concise and to the point so that visitors wouldn’t lose interest in the content meaning they would be more likely to sign up.

Graphical user interface, application

Description automatically generated

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa