# Testing

## Introduction

The libraries and plugins used for testing are:

* Mocha, as the testing framework
* Chai, the assertion library used in conjunction with Mocha
* Chai-http, an additional plugin for Chai

In summary, a total of 9 modules are tested, with each corresponding to a backend route. A total of 46 tests have been ran for these modules, and all 46 passed in testing. Each route was tested for how it should respond after a specific action was successful, and whether it handles illegal or invalid actions correctly.

## Results

|  |  |  |  |
| --- | --- | --- | --- |
| **Administrator Functionalities** | | | |
| **Module** | **Test/ Expected results** | **Method** | **Actual results** |
| Admin login route | On success:  Returns HTML code 200 | Assert that the response contains 200 as its status code | Matches expected output |
| On success:  Creates a session key (authorization key) | Assert that the response contains a token in its body | Matches expected output |
| On success:  Redirects user to the admin home page after login | Assert that the response contains a redirect to the admin home page | Matches expected output |
| For an incorrect password:  Returns HTML code 200 | Assert that the response contains 200 as its status code | Matches expected output |
| For an incorrect password:  Rejects the login attempt by redirecting back to the admin login page | Assert that the response contains a redirect to the admin login page | Matches expected output |
| For an invalid email:  Returns HTML code 200 | Assert that the response contains 200 as its status code | Matches expected output |
| For an invalid email:  Rejects the login attempt by redirecting back to the admin login page | Assert that the response contains a redirect to the admin login page | Matches expected output |
| Edit user route | On success:  Returns HTML code 200 on | Assert that the response contains 200 as its status code | Matches expected output |
| On success:  Changes the user’s information | Assert that the mock user created for this test has its information changed, via a database look-up | Matches expected output |
| Delete route | On success:  Returns HTML code 200 | Assert that the response contains 200 as its status code | Matches expected output |
| On success:  Deletes user from database | Assert that the mock user created for this test no longer exists in the database | Matches expected output |
| On success:  Redirects user back to admin home page after deletion | Assert that the response contains a redirect to the admin login page | Matches expected output |
| User sign-up route | On success:  Returns HTML code 200 | Assert that the response contains 200 as its status code | Matches expected output |
| On success:  Creates one user account in the database | Assert that only one user is created in the database, with matching credentials, via a database look-up | Matches expected output |
| On success:  Password is hashed in the database | Assert that the raw-input password does not match the password stored in the database | Matches expected output |
| On unauthorised request:  Returns HTML code 401 | Assert that the response contains 401 as its status code | Matches expected output |
| On unauthorised request:  Rejects the registration attempt, not creating an account | Assert that no user with the provided credentials exists via a database look-up | Matches expected output |
| For registering with an existing email:  Returns HTML code 200 | Assert that the response contains 200 as its status code | Matches expected output |
| For registering with an existing email:  Rejects the registration attempt, not creating an account | Assert that no user with the provided credentials exists via a database look-up | Matches expected output |
| For an invalid email format:  Returns HTML code 200 | Assert that the response contains 200 as its status code | Matches expected output |
| For an invalid email format:  Rejects the registration attempt, not creating an account | Assert that no user with the provided credentials exists via a database look-up | Matches expected output |
| For a password less than 6 characters in length:  Returns HTML code 200 | Assert that the response contains 200 as its status code | Matches expected output |
| For a password less than 6 characters in length:  Rejects the registration attempt, not creating an account | Assert that no user with the provided credentials exists via a database look-up | Matches expected output |

|  |  |  |  |
| --- | --- | --- | --- |
| **User Functionalities** | | | |
| **Module** | **Test/ Expected results** | **Method** | **Actual results** |
| User login route | On success:  Returns HTML code 200 | Assert that the response contains 200 as its status code | Matches expected output |
| On success:  Creates a session key (authorization key) | Assert that the response contains a token in its body | Matches expected output |
| On success:  Redirects user to the search page after login | Assert that the response contains a redirect to the user home page | Matches expected output |
| For an incorrect password:  Returns HTML code 200 | Assert that the response contains 200 as its status code | Matches expected output |
| For an incorrect password:  Rejects the login attempt by redirecting back to the user login page | Assert that the response contains a redirect to the user login page | Matches expected output |
| For an invalid email:  Returns HTML code 200 | Assert that the response contains 200 as its status code | Matches expected output |
| For an invalid email:  Rejects the login attempt by redirecting back to the user login page | Assert that the response contains a redirect to the user login page | Matches expected output |
| Update info route | On success:  Returns HTML code 200 | Assert that the response contains 200 as its status code | Matches expected output |
| On success:  Updates/changes the user’s information | Assert that the test user’s information has changed via a database look-up | Matches expected output |
| On unauthorised request:  Returns HTML code 401 | Assert that the response contains 401 as its status code | Matches expected output |
| On unauthorised request:  Does not update the user’s information | Assert that the test user’s information remains unchanged via a database look-up | Matches expected output |
| Reset password route | On success:  Returns HTML code 200 | Assert that the response contains 200 as its status code | Matches expected output |
| On success:  Reset’s the user’s password | Asserts that the user can no longer login using the old password | Matches expected output |
| On unauthorised request:  Returns HTML code 401 | Assert that the response contains 401 as its status code | Matches expected output |
| On unauthorised request:  Does not reset the user’s password | Asserts that the user can still login using the old password | Matches expected output |
| Search route | On success:  Returns HTML code 200 | Assert that the response contains 200 as its status code | Matches expected output |
| On success:  For a query with expected results, returns results. | Assert that the query inputted matches both the response body’s result and the database’s result. | Matches expected output |
| On success:  For a query expected to return no results, returns empty results. | Assert that the response contains and empty body | Matches expected output |
| Profile route | On success:  Returns HTML code 200 | Assert that the response contains 200 as its status code | Matches expected output |
| On success:  Returns correct user information | Asserts that the response contains the user’s information in its body, and that it matches the user’s information stored in the database | Matches expected output |