# Testing

* 1. Introduction

My web application will undergo functional testing. I will be testing various parts of the website such as Crud functionality, user login and registration, and navigation. These are all functional requirements for my website. It is important to test that they all work as expected so my web app can function correctly. For this chapter I am doing functional testing and user testing.

* 1. Functional Testing

I will be testing the login, registration, crud, and navigation functionality. I will be recording my test results using a Black Box. I will log the test number, input, expected output and actual output. It is important to check that the output is what you expected. If its not you must record it and debug the problem.

* + 1. Login/Registration

To test Login functionality, I first checked to see if a user can register an account. This will require an email and password which will be stored in the database. The register page must be loaded and then the data entered. Once submitted I check the database to see if that user was added to the records.

Then to test login functionality, I use the same credentials and login. The data for logging in and registering is passed through the Login Controller and Register Controller.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No | Description of test case | Input | Expected Output | Actual Output | Comment |
| 1 | Testing if a user can register. | User enters an email and password, submits the data. | An account is created in the database | An account is created in the database |  |
| 2 | Testing if a user can login. | A user logs in with the registered email and password | The user can login | The user can login |  |
|  |  |  |  |  |  |

* + 1. Navigation

Navigation is important to check that the flow of the web app is as intended. For my WebApp I want to test can a user get from the home page to creating a post and view it. This will require routes for the user and the correct pages to be loaded.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No | Description of test case | Input | Expected Output | Actual Output | Comment |
| 1 | The user can navigate from the home page to the create page | A user will click on the create button and the page is loaded | The user can navigate to the correct page | The user can navigate to the correct page |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

* + 1. CRUD

For the Crud functionality testing I tested my create function first. I went to the web app’s create page and added data to the database through a forum. I then submitted the data and checked the database to see if the correct data had been added and a post was created. When the data is being submitted it is checked in the store function of my controlled. It uses a validate function to make sure the data that the user is submitting is valid.  
Text

Description automatically generated

For testing the read functionality, I was able to load my webapp and go to the post I submitted to the webapp and find its Id. I was then able to load that data by id to a web page.   
To check if update functionality was working, I went to the update page and edited the data with a forum and resubmitted it to the database. It then also uses a require function to check if the data is valid. I then check in the database if the data has changed correctly and then check in the webapp if it is displayed correctly.  
I can then check to see if delete functionality is working by deleting the data, I added earlier by using the web app to call the controllers destroy function and passing it the correct id. I can then check in the webapp and database to see if the data has been deleted.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No | Description of test case | Input | Expected Output | Actual Output | Comment |
| 1 | Create function works | The user can create a post | A post is created | A post is created |  |
| 2 | The user can see the post they created | The user views the post by Id | The user can view the post by Id | The user can view the post by Id |  |
| 3 | The user can update the post in a forum | Data is entered into the forum | The data in the database is updated correctly | The data in the database is updated correctly |  |
| 4 | The user can delete the post from the database through the webapp | The user calls the destroy function | The post is deleted from the database | The post is deleted from the database |  |

### Discussion of Functional Testing Results

While testing the functionality of my web app I found everything to work as expected. All data that I created or updated was valid. When reading and deleing it everything also worked as intended. All navigation functionality worked as intended. All pages loaded and routed correctly. Login and registration also had no problems when carrying out testing.

* 1. User Testing

When doing the user testing, I made sure the front end of the application was working correctly. I checked the pages loaded correctly. I made sure all buttons for navigation worked correctly. I made sure that all forums worked as expected and a user could interact with them.

* 1. Conclusion

I had no issues when testing and everything worked as expected. I found it useful to record the steps taken, the input, expected output and actual output. Recording these makes it much easier to review and make changes where necessary. Logging each step is also useful incase you must go back and fix other problems or debug.