# Testing

* 1. Introduction

This chapter describes the testing that has been undertaken for the application. This chapter is presented in two sections:

1. Functional Testing
2. User Testing

Functional testing is a type of software testing whereby the system is tested against the functional requirements. The app is tested by looking to see if the actual output for a given input corresponds with the expected output. The tests should be based on the requirements for the app. The results of functional testing can indicate if a piece of software is functional and working, but not if the software is easy to use.

User testing looks to see if a piece of software is easy and intuitive for the user.

* 1. Functional Testing

This section describes the functional tests which were carried out on the app. These functional tests can be categorised as: (whatever is relevant to your app)

Login/Registration

Navigation

Calculation

CRUD

Functional testing generally uses a Black Box Testing technique which means that the internal logic of the system being tested is not of interest to the tester. The tester is only interested in whether the actual output agrees with the expected output.

* + 1. Login/Registration

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No | Description of test case | Input | Expected Output | Actual Output | Comment |
| 1 | User logs in with a valid email and password | [example@blazar.ie](mailto:example@blazar.ie)  \*\*\*\*\*\*\*\*\*\* | You are now logged in | You are now logged in | Pass |
| 2 | User enters an unknown email and/or password | [unknown@blazar.ie](mailto:unknown@blazar.ie)  \*\*\*\*\*\*\*\*\*\* | These credentials do not match our records. | These credentials do not match our records. | Pass |
| 3 | User enters an email without the @ sign with a valid password | Example.ie  \*\*\*\*\*\*\*\*\*\* | Please include a @ the email address example.ie is missing a @ | Please include a @ the email address example.ie is missing a @ | Pass |

* + 1. Navigation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No | Description of test case | Input | Expected Output | Actual Output | Comment |
| 1 | Click home button | http://127.0.0.1:8000/ | Blazar homepage | Blazar  homepage | Pass |
| 2 | Click browse button | http://127.0.0.1:8000/user/shoes | Blazar products page | Blazar products page | Pass |
| 3 | Click view product | <http://127.0.0.1:8000/user/shoes/> {id} | Blazar product view page | Blazar product view page | Pass |

* + 1. Calculation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No | Description of test case | Input | Expected Output | Actual Output | Comment |
| 1 | User buying product, shipping calculated |  | Shipping added to product price | Shipping added to product price |  |
| 2 | User selling product, selling fees calculated |  | Fees subtracted from product selling | Fees subtracted from product selling |  |
|  |  |  |  |  |  |

* + 1. CRUD

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No | Description of test case | Input | Expected Output | Actual Output | Comment |
| 1 | Admin edits product in the database | http://127.0.0.1:8000/admin/shoes/1/edit | Redirected to http://127.0.0.1:8000/admin/shoes | Redirected to http://127.0.0.1:8000/admin/shoes | Pass |
| 2 | Admin deleted product from the database | http://127.0.0.1:8000/admin/shoes | Product is deleted | Product is deleted | Pass |
| 3 | Admin adds product to the database | http://127.0.0.1:8000/admin/shoes/create | Product is added to database | Product is added to the database | Pass |

### Discussion of Functional Testing Results

The results were positive anything that failed was a quick fix because they were minor syntax errors.

* 1. User Testing
  2. Conclusion

The results were good the admin has crud functionality over the application as intended. The next thing to do is to give the user create functionality so the can create a product to be added to the database.