# Chapter 5

## 5.1 Introduction to Testing

Software testing is an integral part of software development which ensures that the system is free of any kind of defects. During testing, it is checked whether the actual results match the expected results. It might involve the execution of the whole system or a small component for evaluation.

Testing helps in identifying errors or missing requirements in contrast to the provided requirements. Testing can be performed both manually or with the help of automated tools.

There are a number of techniques to test a system. They can be listed as:

1. Integration Testing
2. Interface Testing
3. Unit Testing
4. System Testing
5. Regression Testing
6. Black Box Testing
7. Usability Testing
8. Vulnerability Testing
9. Acceptance Testing and many more.

Among these testing methods, I have chosen **Unit Testing** and **Black Box Testing** for my system.

## 5.2 Unit Testing

Unit testing can be defined as a type of testing that tests the system’s components. Since this type of testing requires adequate understanding of the system’s architecture, design and code, it is usually done by the programmer and not the testers. It often requires to develop test driver modules.

I have chosen unit testing because of the following reasons:

1. **Agile Process:** It makes the developing process more agile since it tests individual components which allows refactoring in a very simple and safe way.
2. **Quality of Code:** Unit testing helps us identify potential defects in the code.
3. **Identify Issues Early:** Since the testing involves the test of individual codes, it is more probable that the issues are found early.
4. **Documentation:** With the help of documentation developed during the testing, other developers can easily understand the functions of individual units.

### 5.2.1 Adding a pet

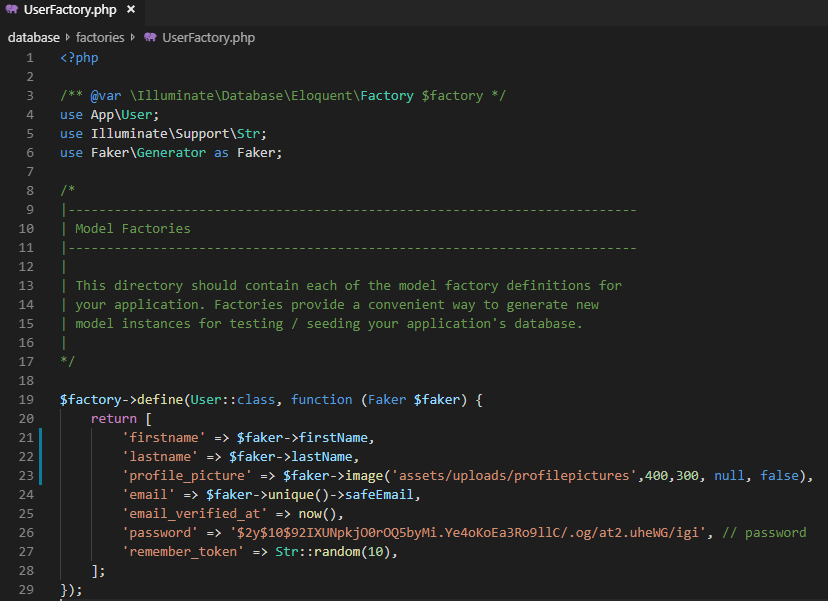


Figure 1: Database Factory to create Users

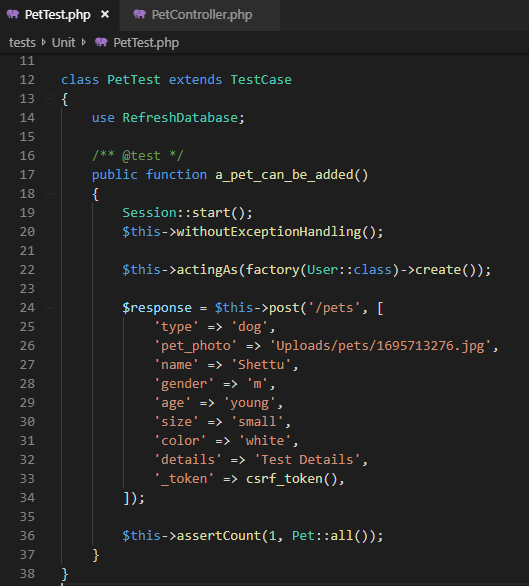


Figure 2: Test to store pet details

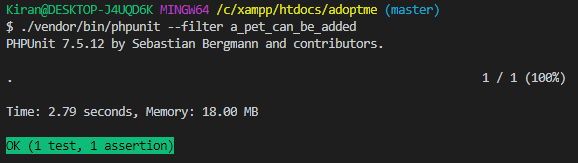


Figure 3: Test Result Success

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test ID** | **Test Class** | **Expected Output** | **Actual Output** | **Result** |
| UT 1 | Pet | Successful Data Entry | Successful Data Entry | Pass |

### 5.2.2 Unauthorized access to User Dashboard

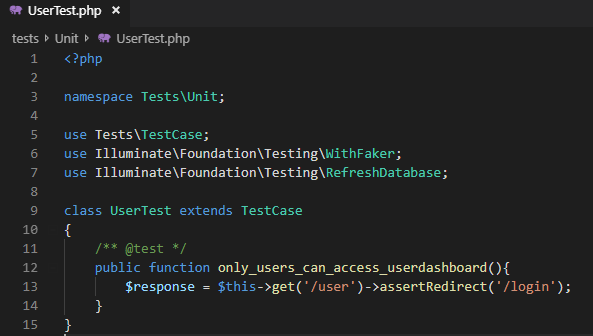


Figure 4: Test to check unauthorized access to user dashboard

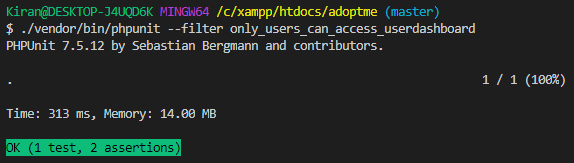


Figure 5: Test Result Success

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test ID** | **Test Class** | **Expected Output** | **Actual Output** | **Result** |
| UT 2 | Route | Redirection to Login Page | Redirected to Login Page | Pass |

### 5.2.3 Creating a purchase order by a registered user

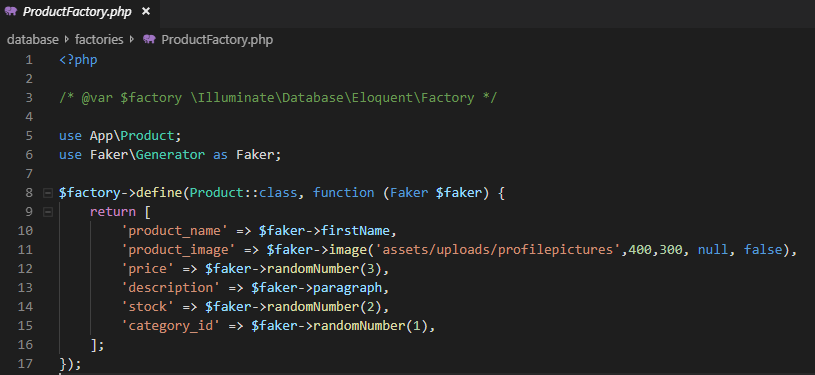


Figure 6: Database Factory to create Products

### 

Figure 7: Test to create a purchase order by a registered user

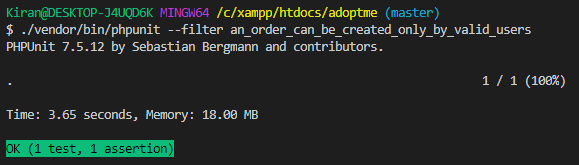


Figure 8:Test Result Success

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test ID** | **Test Class** | **Expected Output** | **Actual Output** | **Result** |
| UT 3 | Order | Successful Order | Successful Order | Pass |

### 5.2.4 Updating a Blog

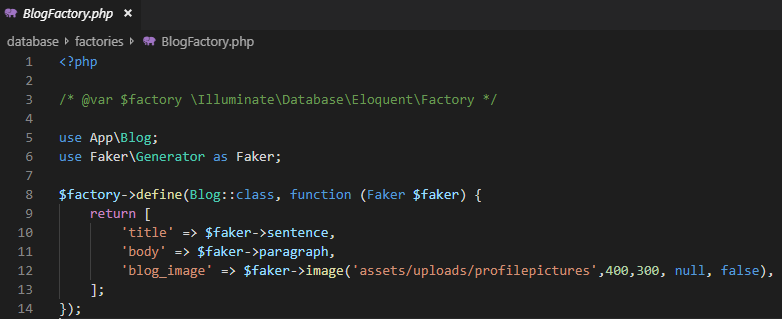


Figure 9: Factory to create Blogs

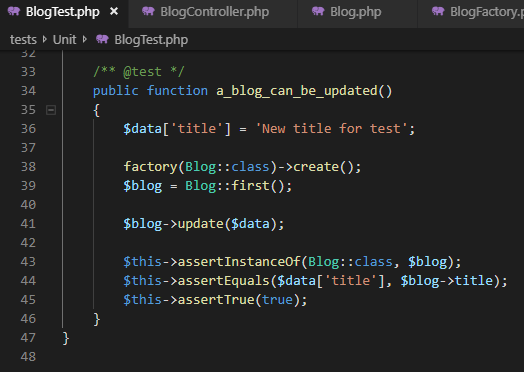


Figure 10: Test to update blog

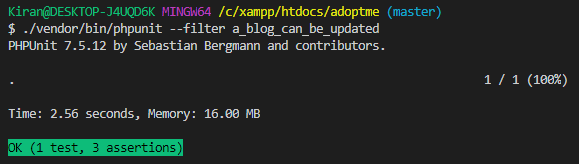


Figure 11: Test Result Successful

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test ID** | **Test Class** | **Expected Output** | **Actual Output** | **Result** |
| UT 4 | Blog | Successful Blog Update | Successful Blog Update | Pass |

### 5.2.5 Counting the number of Pets

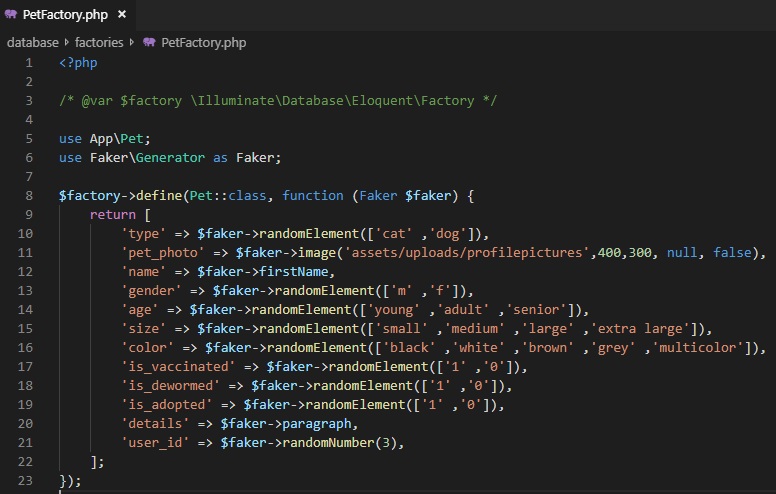


Figure 12: Factory to create Pets

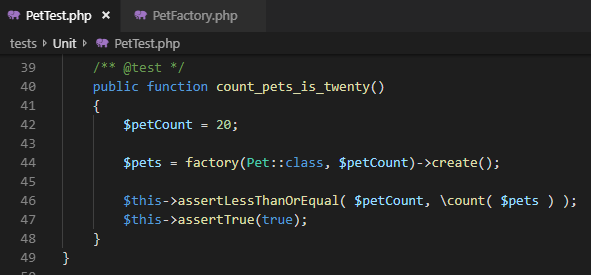


Figure 13: Test to count the number of pets

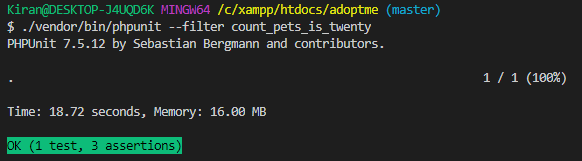


Figure 14: Test Result Successful

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test ID** | **Test Class** | **Expected Output** | **Actual Output** | **Result** |
| UT 5 | Pet | Successful pet count | Successful pet count | Pass |

### 5.2.6 Testing the relation between users and pets

### 

Figure 15: Testing the relation between users and pets

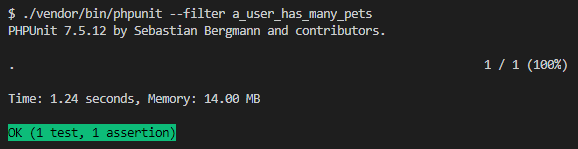


Figure 16: Test Result Successful

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test ID** | **Test Class** | **Expected Output** | **Actual Output** | **Result** |
| UT 6 | User and Pet | Relation: User has many pets | Relation: User has many pets | Pass |

### 5.2.7 Testing the total price of an order

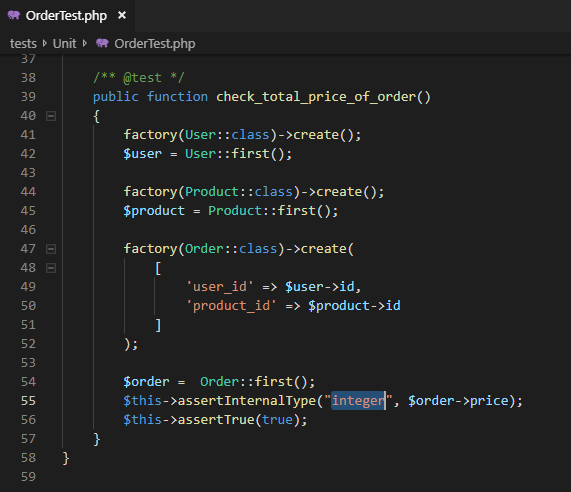


Figure 17: Test to check total price of an order

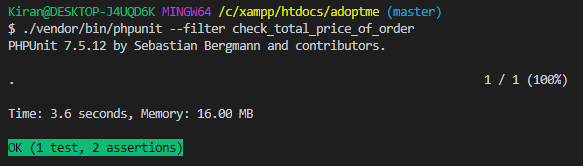


Figure 18: Test Result Successful

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test ID** | **Test Class** | **Expected Output** | **Actual Output** | **Result** |
| UT 7 | User, Product and Order | Total price check | Total price checked | Pass |

### 5.2.8 Testing the route to display help page

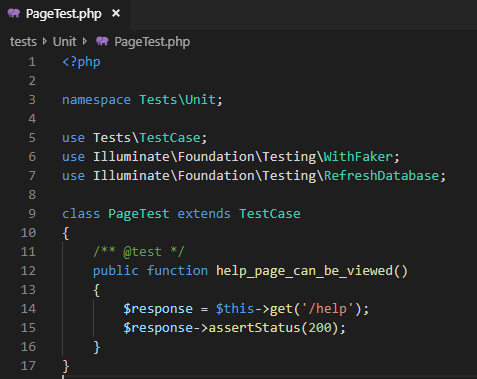


Figure 19: Test to display help page

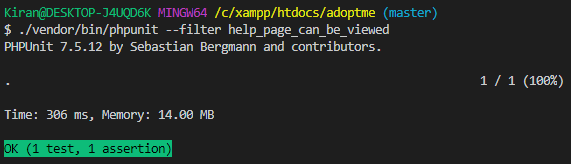


Figure 20: Test Result Successful

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test ID** | **Test Class** | **Expected Output** | **Actual Output** | **Result** |
| UT 8 | Route | Successful redirection to Help Page | Successful redirection to Help Page | Pass |

# 5.3 Black Box Testing

Black box testing is a technique of testing a system in which the tester does not know the internal structure and implementation of the item being tested. Generally, the functionality of the application is tested based on the software requirements and specifications.

I have chosen Black box testing in order to test my system because of the following reasons:

1. It is efficient to use for both small and large systems.
2. The testing is balanced and unbiased since the tester and developer work independently.
3. It can be performed by a non-technical person and he/she doesn’t need to have adequate knowledge of the system.
4. Test is performed from a user’s point-of-view.

### 5.3.1 Load Webpage

### C:\Users\Kiran\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\5D23233E.tmp

Figure 21: Home Page

### 5.3.2 Register User

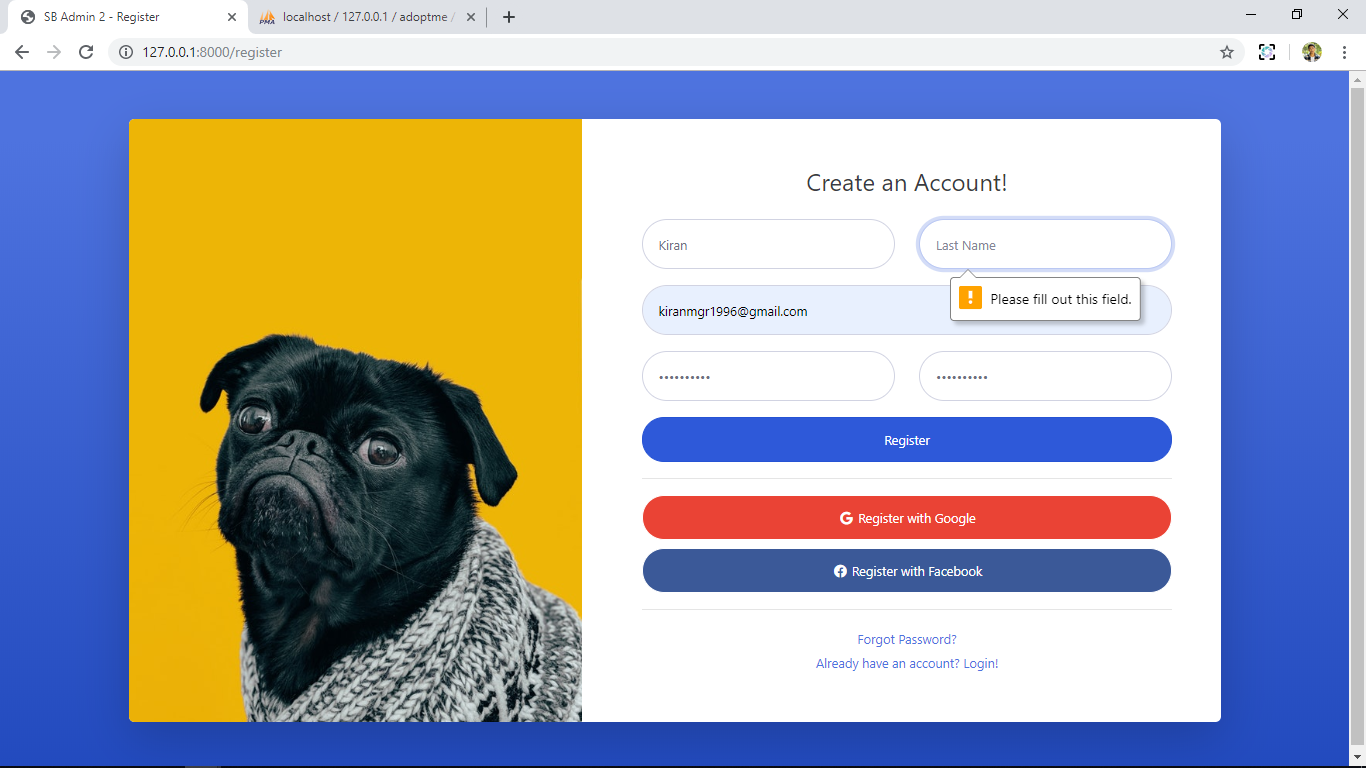


Figure 22: Registering User without inserting Last Name

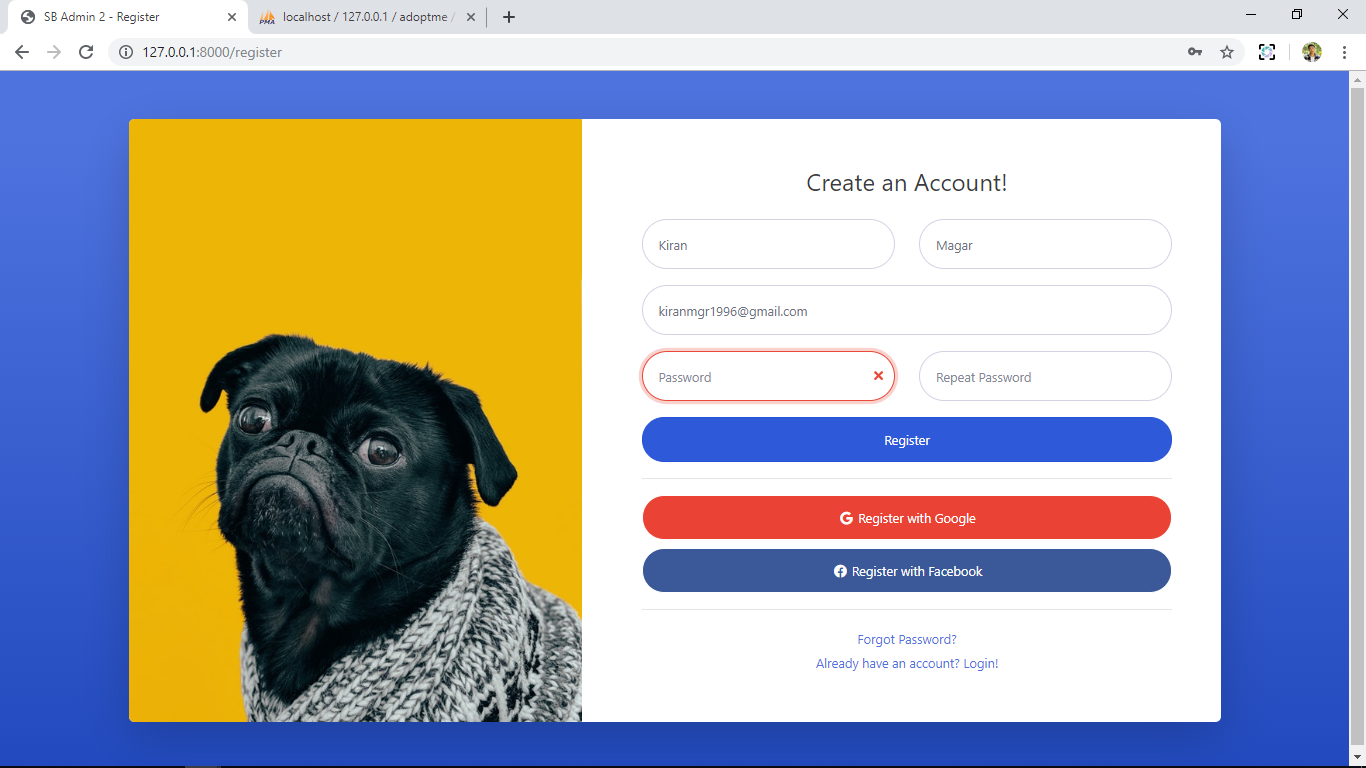


Figure 23: Registering user with different values in 'Password' and 'Repeat Password' fields

### 5.3.3 Validate Login

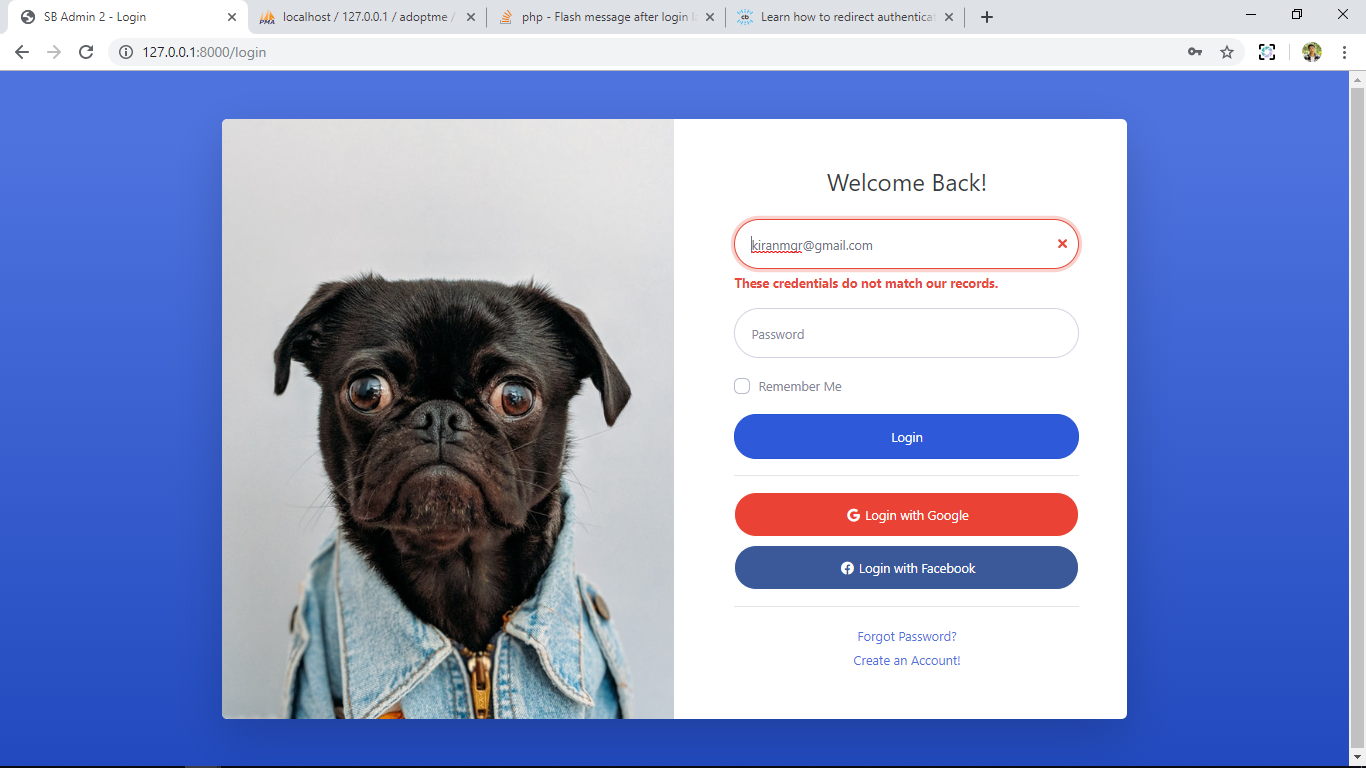


Figure 24: Login attempt with invalid details

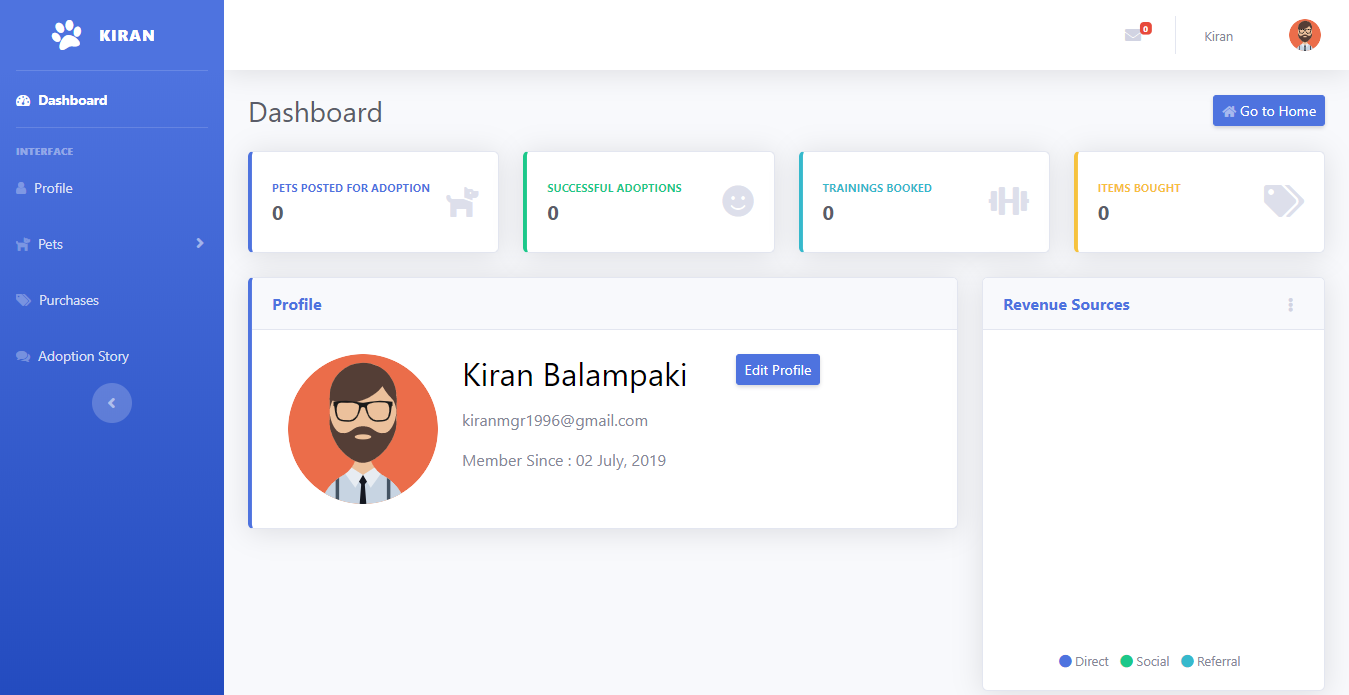


Figure 25: Redirected to User Dashboard

### 5.3.4 Add a Pet

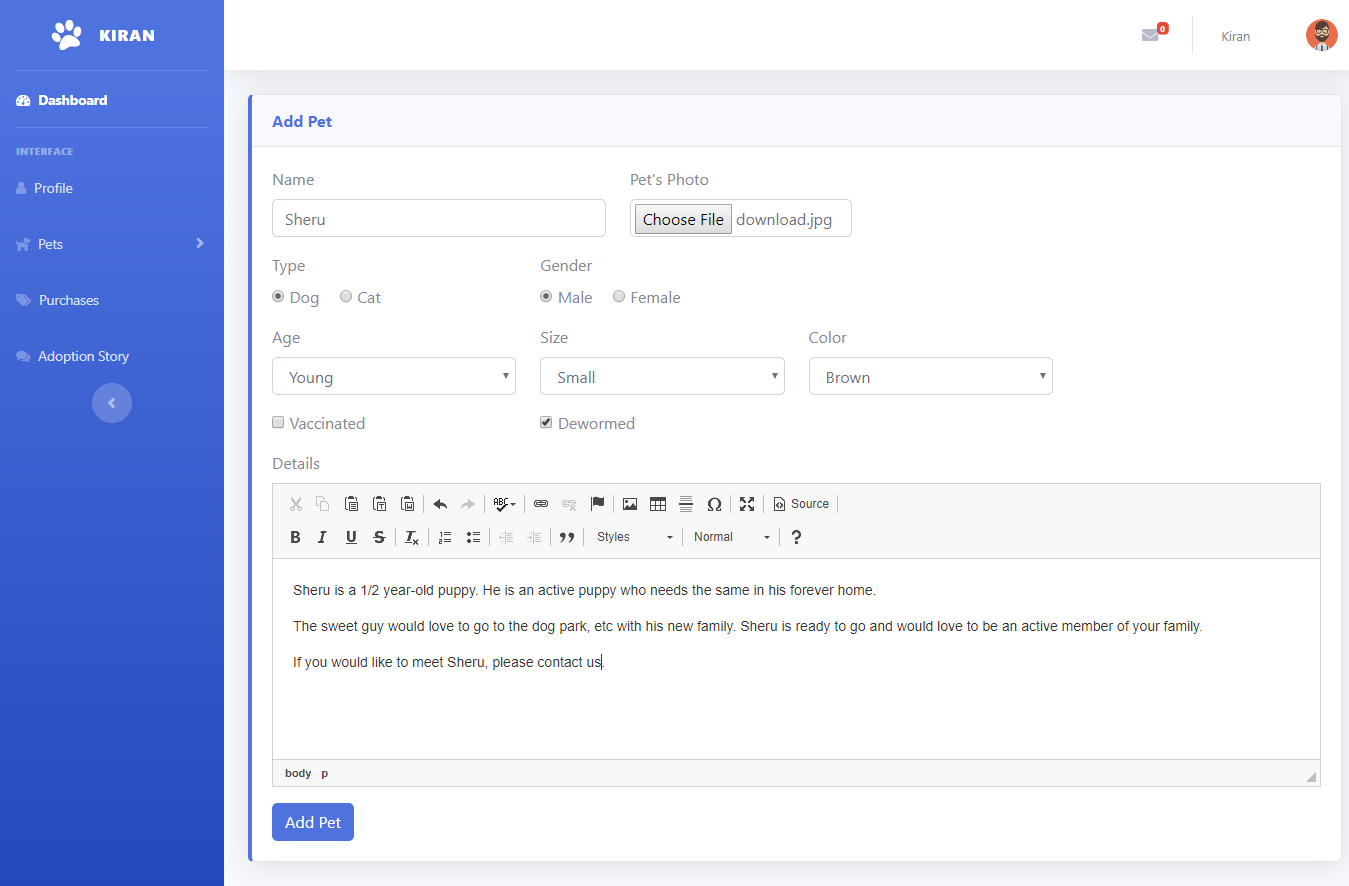


Figure 26: Pet add for with valid details.

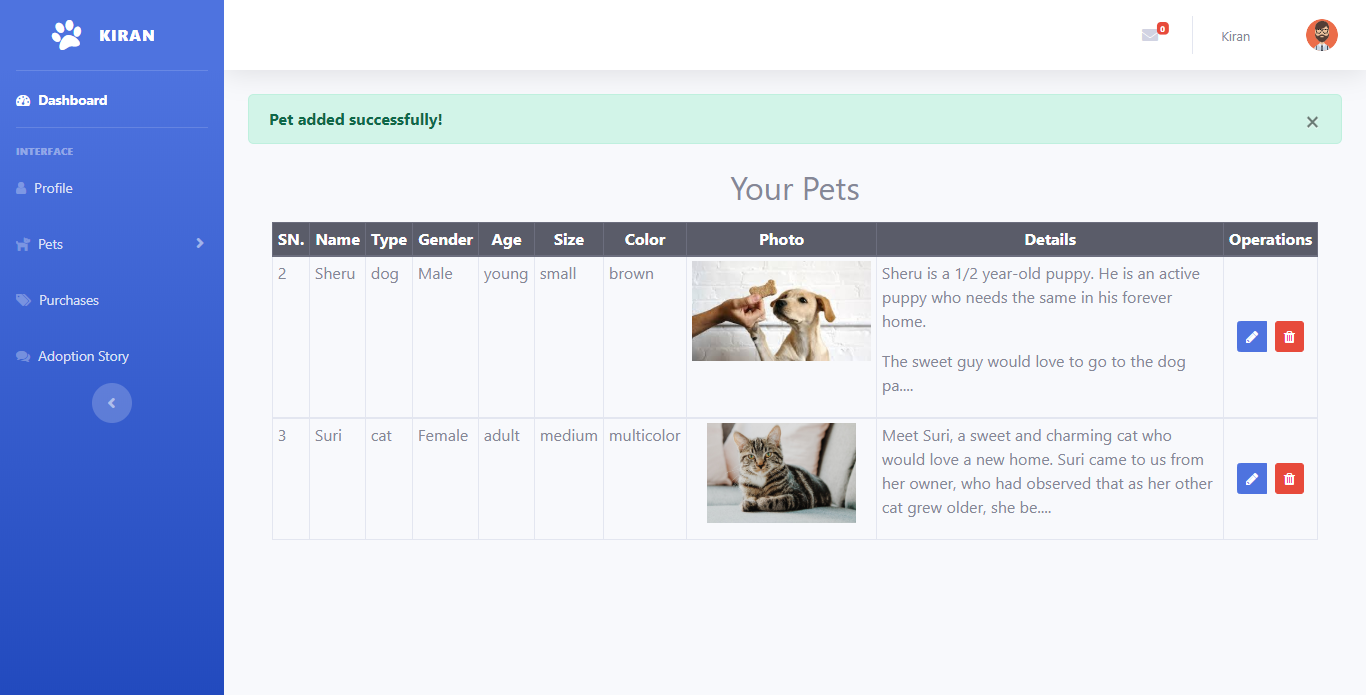


Figure 27: Redirected to pet list

### 5.3.5 Add a Pet

**Test Planned By**: Kiran Balampaki

**Test Plan Date**: 30th June, 2019

**Test Done By**: Kiran Balampaki

**Test Executed Date**: 1st July, 2019

**Test For**: Adoptme Website

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case No: | Test Case | Input | Expected Result | Actual Result | Remarks |
| 5.3.1 | Loading Webpage | URL of the website was entered | The website should load with the home page. | The website loaded with the home page. | PASS |
| 5.3.2 | Register user in the website | Register form was filled leaving the ‘Last Name’ field empty and ‘Register’ button was clicked | The website should prompt to enter ‘Last Name’. | The website displayed ‘Please fill out this field’ in the ‘Last Name’ field. | PASS |
| Register form was filled with different values in ‘Password’ and ‘Repeat Password’ field and ‘Register’ button was clicked | The website should prompt to enter the passwords fields again. | The website displayed error in password field making user to enter it again. | PASS |
| Register form was correctly filled and ‘Register’ button was clicked | Success message should be displayed and redirect to Login page. | “Successful Registration” message was displayed and redirected to Login page. | PASS |
| 5.3.3 | Validate Login | Invalid data was filled in the login form and ‘Login’ button was clicked | Login should not be validated. | “These credentials do not match our records” message was displayed | PASS |
| Valid data was filled in the login form and ‘Login’ button was clicked | Login should be validated and redirected to User Dashboard | Login was validated and redirected to User Dashboard | PASS |
| 5.3.4 | Add a pet | Add pet form was filled with valid data and ‘Add Pet’ button was clicked | Pet should be added and the user should be redirected to his/her pet list page. | Pet was added and the user was redirected to his/her pet list page. | PASS |
|  |  |  |  |  |  |