**Chapter 5: Testing**

**Introduction**

Testing is an action to analyze the functionality of a software application in order to find out the specified requirements that the developed software has met or not and in order to produce the quality software, weakness is identify to assure that the product is defect free.

There are different types of testing but I have used unit testing and black box testing which have their own features and goals for testing.

Test case: It is a document of test data, precondition, expected results, actual results and outcome as expected.

**Advantage of test case:**

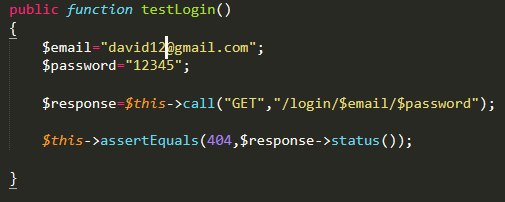
* **Time-saving.**
* **Provide good quality software.**
* **Reduce weakness.**

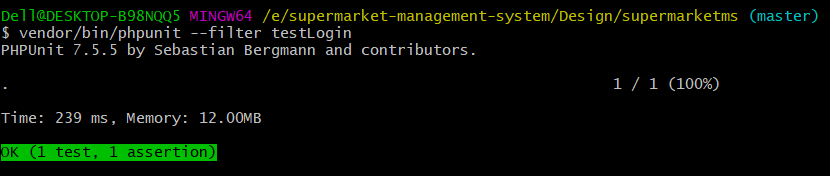
**White box testing**

White box is known as code-based testing in which the internal structure of the several programming and coding is tested. This testing is also known as unit testing. The white box testing for my supermarket is given below:

**1. Login**

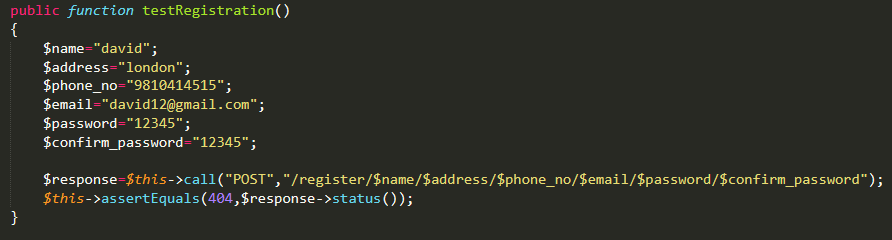
|  |  |
| --- | --- |
| **Test case** | **WB1** |
| **Purpose of test case** | Email and password can be retrieved or not to login the system. |
| **Test data** | Email= david12@gmail.com  Password= 12345 |
| **Class name** | supermarketTest |
| **Function name** | testLogin |
| **Expected result** | To login into the system. |
| **Actual result** | Successfully login. |
| **Outcome as expected** | Yes |

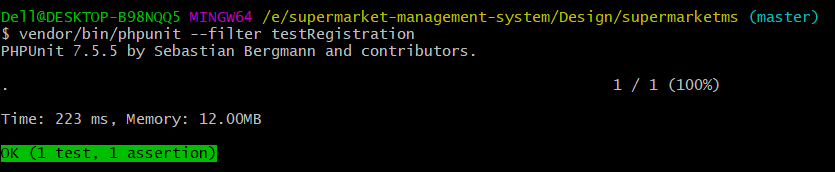
****

****

2. Registration

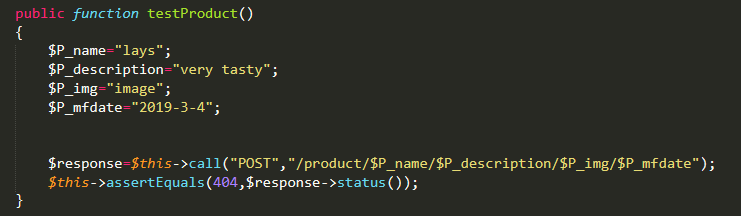
|  |  |
| --- | --- |
| **Test case** | **WB2** |
| **Purpose of test case** | User can successfully register or not. |
| **Test data** | Name= david  Address= London  Phone\_no= 9810414515  Email= [david12@gmail.com](mailto:david12@gmail.com)  Password= 12345  Confirm\_password= 12345 |
| **Class name** | supermarketTest |
| **Function name** | testRegistration |
| **Expected result** | To successfully register user. |
| **Actual result** | User registered successfully. |
| **Outcome as expected** | Yes |

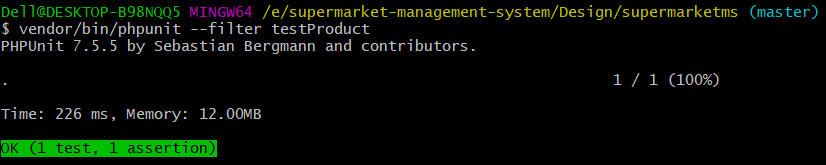




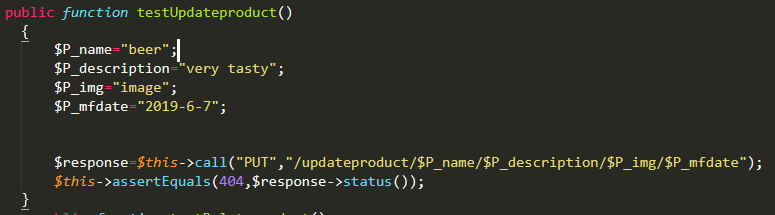
3. Insert Product

|  |  |
| --- | --- |
| **Test case** | **WB3** |
| **Purpose of test case** | Product can insert successfully or not. |
| **Test data** | P\_name= lays  P\_description= very tasty  P\_img= image  P\_mfdate= 2019-03-04 |
| **Class name** | supermarketTest |
| **Function name** | testRegistration |
| **Expected result** | To insert product successfully. |
| **Actual result** | Successfully inserted in the database. |
| **Outcome as expected** | Yes |

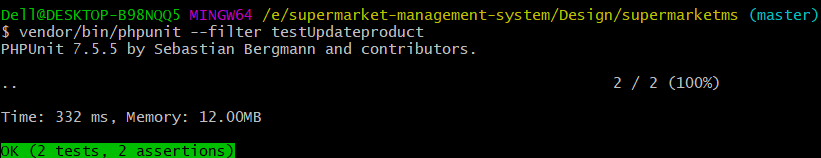




4. Update Product

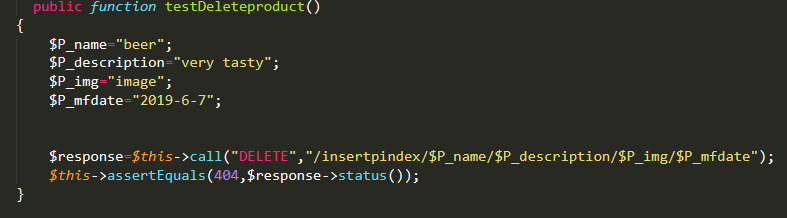


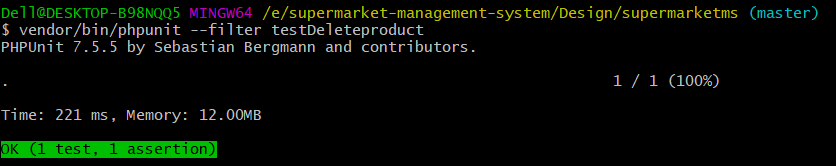
|  |  |
| --- | --- |
| **Test case** | **WB4** |
| **Purpose of test case** | Product can update successfully or not. |
| **Test data** | P\_name= beer  P\_description= very tasty  P\_img= image  P\_mfdate= 2019-06-07 |
| **Class name** | supermarketTest |
| **Function name** | testRegistration |
| **Expected result** | To update product successfully. |
| **Actual result** | Successfully updated in the database. |
| **Outcome as expected** | Yes |



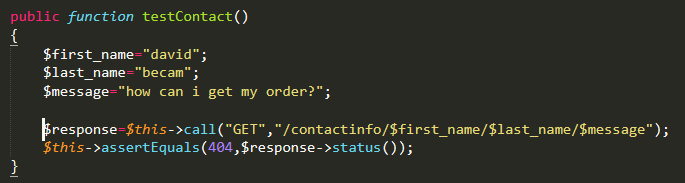
5. Delete Product

|  |  |
| --- | --- |
| **Test case** | **WB5** |
| **Purpose of test case** | To check product can delete successfully or not from the database. |
| **Test data** | P\_name= beer  P\_description= very tasty  P\_img= image  P\_mfdate= 2019-06-07 |
| **Class name** | supermarketTest |
| **Function name** | testRegistration |
| **Expected result** | To delete product successfully. |
| **Actual result** | Successfully deleted from the database. |
| **Outcome as expected** | Yes |

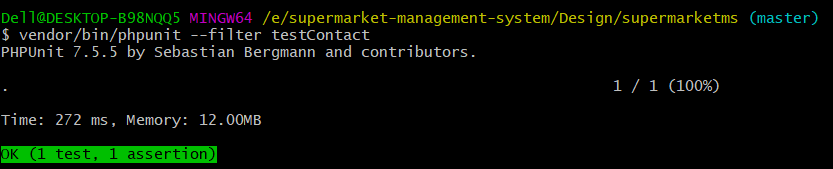




6. Contact

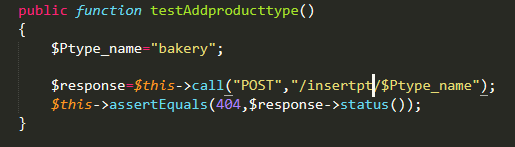


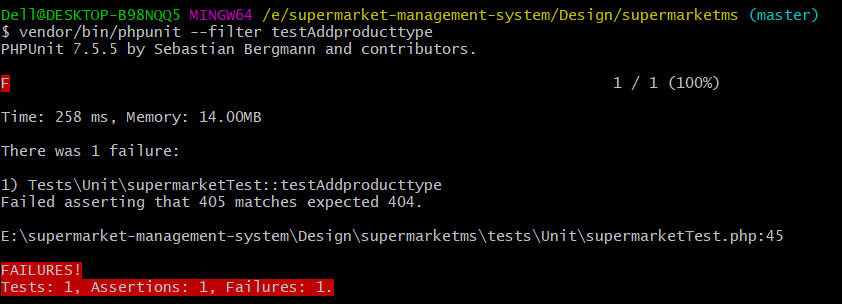
|  |  |
| --- | --- |
| **Test case** | **WB6** |
| **Purpose of test case** | To check user can directly contact to the seller or not. |
| **Test data** | First\_name=david  Last\_name=becam  Message= how can I get my order? |
| **Class name** | supermarketTest |
| **Function name** | testRegistration |
| **Expected result** | To insert information in the database. |
| **Actual result** | Successfully inserted in the database. |
| **Outcome as expected** | Yes |



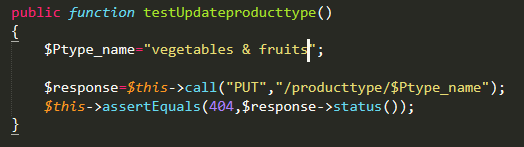
7. Insert Product Type

|  |  |
| --- | --- |
| **Test case** | **WB7** |
| **Purpose of test case** | Product type can insert successfully or not. |
| **Test data** | Ptype\_name=bakery |
| **Class name** | supermarketTest |
| **Function name** | testAddproducttype |
| **Expected result** | To insert product type successfully. |
| **Actual result** | Successfully inserted in the database. |
| **Outcome as expected** | No |

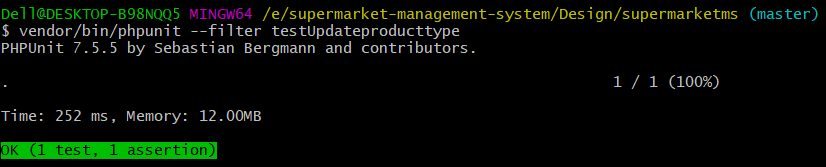




8. Update Product Type

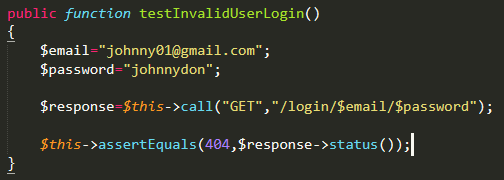


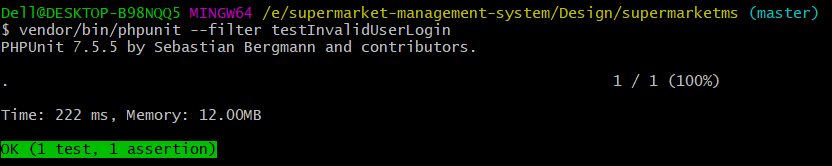
|  |  |
| --- | --- |
| **Test case** | **WB8** |
| **Purpose of test case** | Product type can update successfully or not. |
| **Test data** | Ptype\_name=bakery |
| **Class name** | supermarketTest |
| **Function name** | testAddproducttype |
| **Expected result** | To update product type successfully. |
| **Actual result** | Successfully updated in the database. |
| **Outcome as expected** | Yes |



9. Invalid User Login

|  |  |
| --- | --- |
| **Test case** | **WB8** |
| **Purpose of test case** | To check invalid user can login into the system or not. |
| **Test data** | [Email= johnny01@gmail.com](mailto:Email=%20johnny01@gmail.com)  Password=johnnydon |
| **Class name** | supermarketTest |
| **Function name** | testInvalidUserLogin |
| **Expected result** | Show error message and return back to the login page. |
| **Actual result** | Invalid user return back to the login page with error message. |
| **Outcome as expected** | Yes |





10. Open login page

|  |  |
| --- | --- |
| **Test case** | **WB10** |
| **Purpose of test case** | To check invalid user can login into the system or not. |
| **Test data** | [Email= johnny01@gmail.com](mailto:Email=%20johnny01@gmail.com)  Password=johnnydon |
| **Class name** | supermarketTest |
| **Function name** | testInvalidUserLogin |
| **Expected result** | Show error message and return back to the login page. |
| **Actual result** | Invalid user return back to the login page with error message. |
| **Outcome as expected** | Yes |

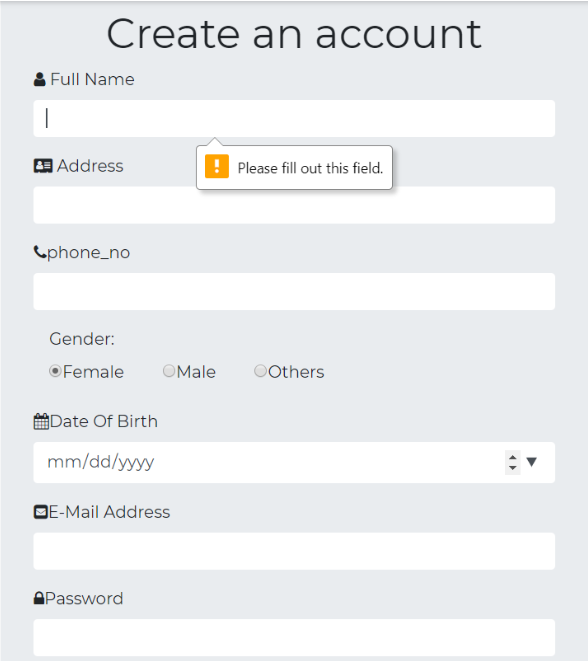
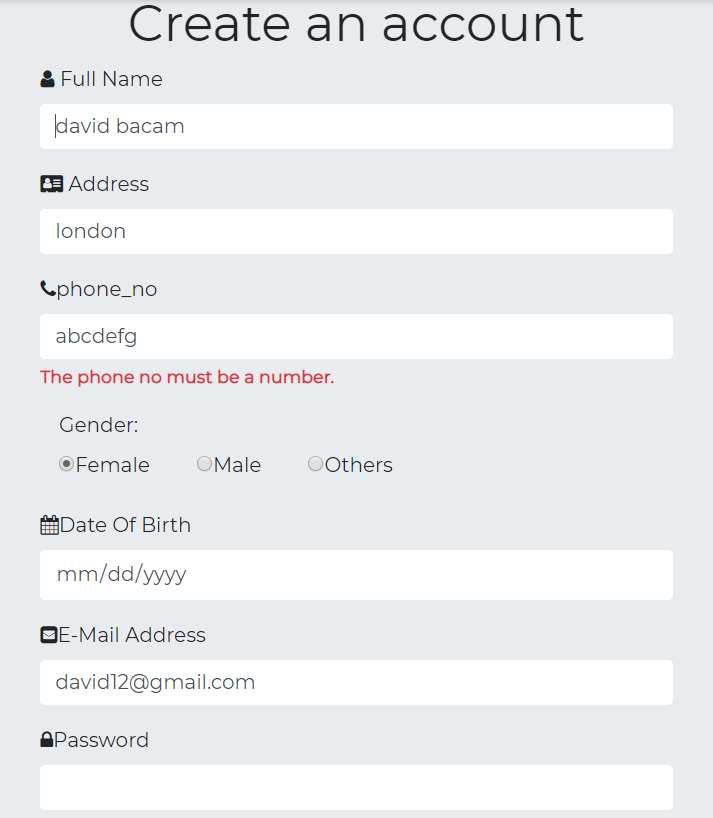
**Black box testing**

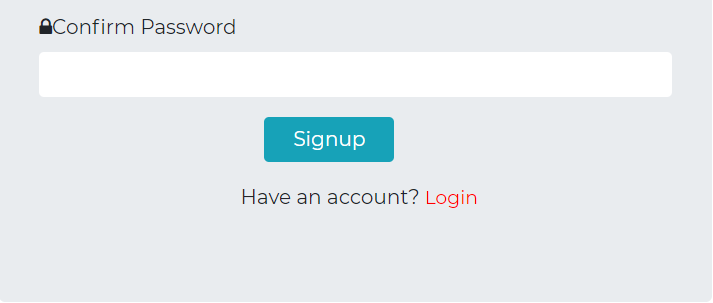
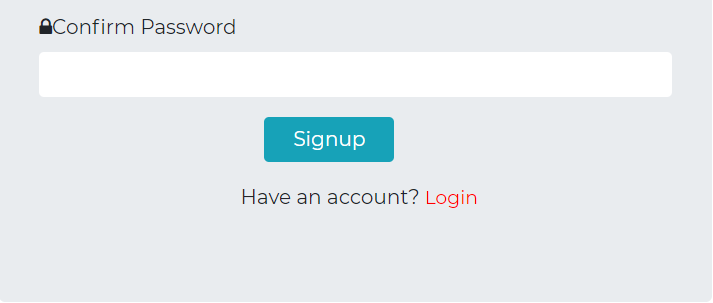
**1. Registration**

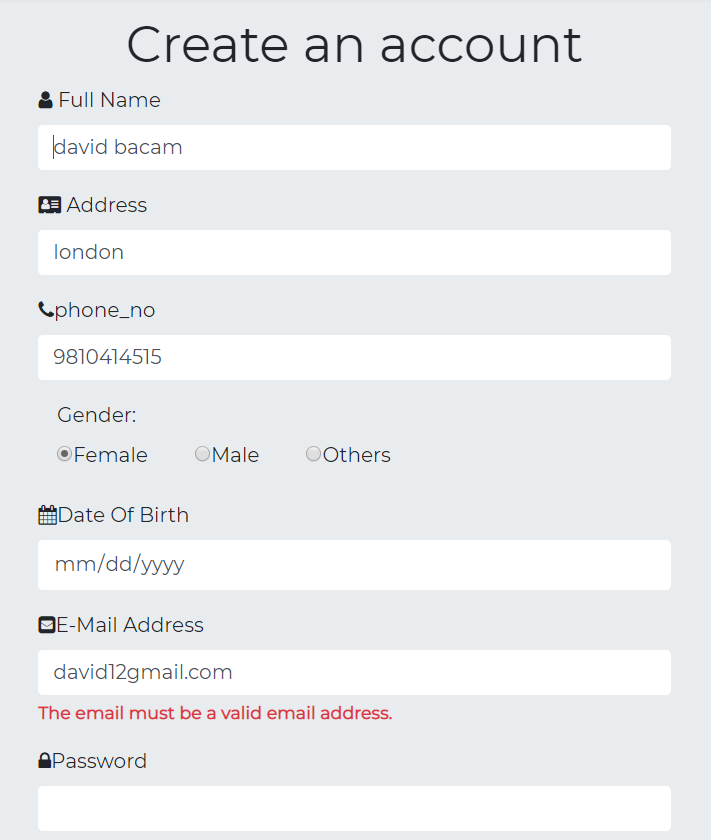
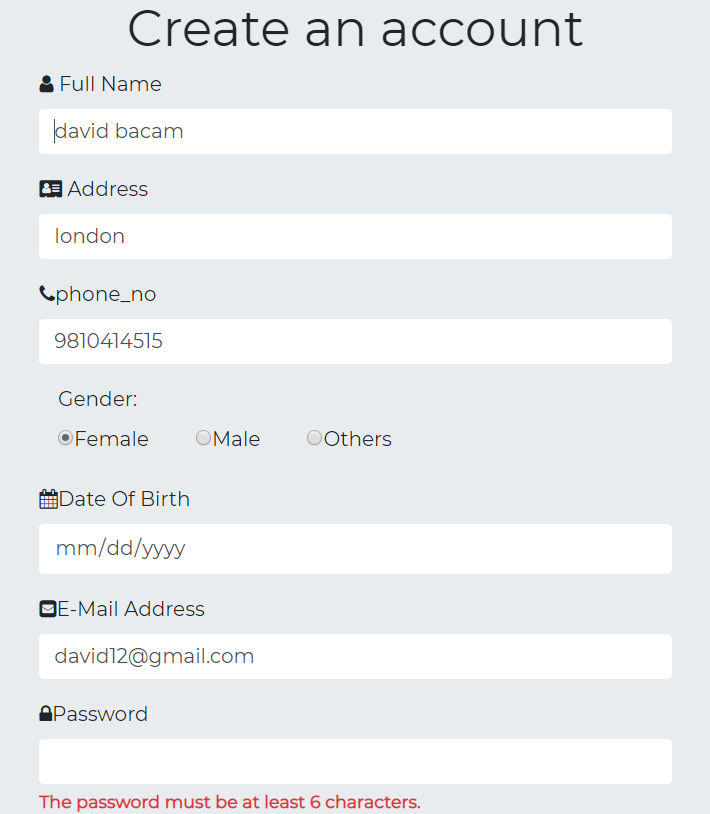
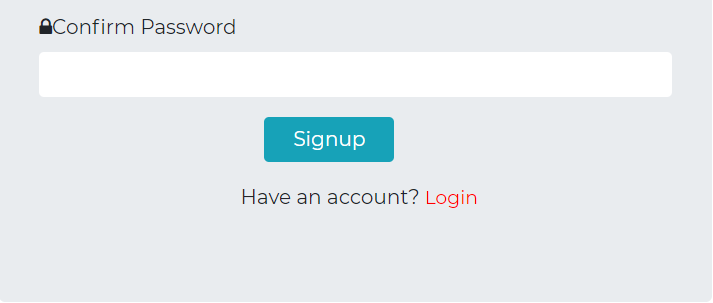
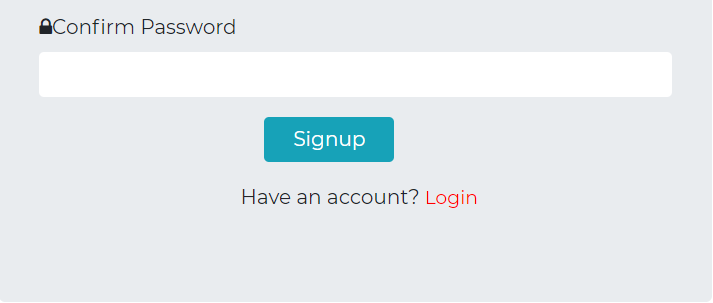
**Unit Test: 1 Test Class: Validation of registration**

**Data Source: User Objectives: To check the validation level of registration.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case** | **Description** | **Tasks** | **Expected Result** | **Actual Result** | **Conclusion** |
| 1.1 | Providing the blank value for registration. | User doesn’t input the value for the registration | Throw error message with “Please fill out this field”. | Successfully shown error message i.e. “please fill out this field”. | Pass |
| 1.2 | The phone number must be a number. | User input the alphabetic letter in a number input. | Throw error message with “the phone no must be a number”. | Successfully shown error message i.e. “the phone no must be a number”. | Pass |
| 1.3 | Must provide email in an email format. | User doesn’t provide email in an email format. | Throw error message with “The email must be a valid email address”. | Successfully shown error message i.e. “The email must be a valid email address”. | Pass |
| 1.4 | Password must be at least six characters. | User input only 5 characters. | Throw error message with “The password must be at least 6 characters”. | Successfully shown error message i.e. “The password must be at least 6 characters”. | Pass |

Output:





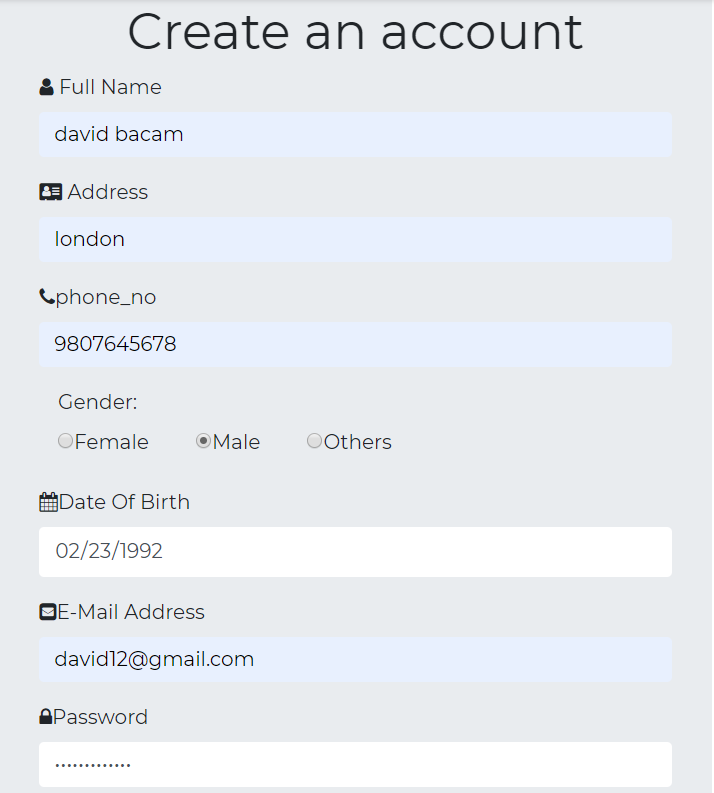
**2.** **Valid Registration**

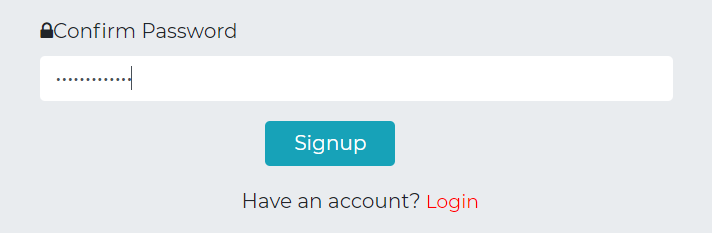
**Unit Test: 2 Test Class: Valid registration**

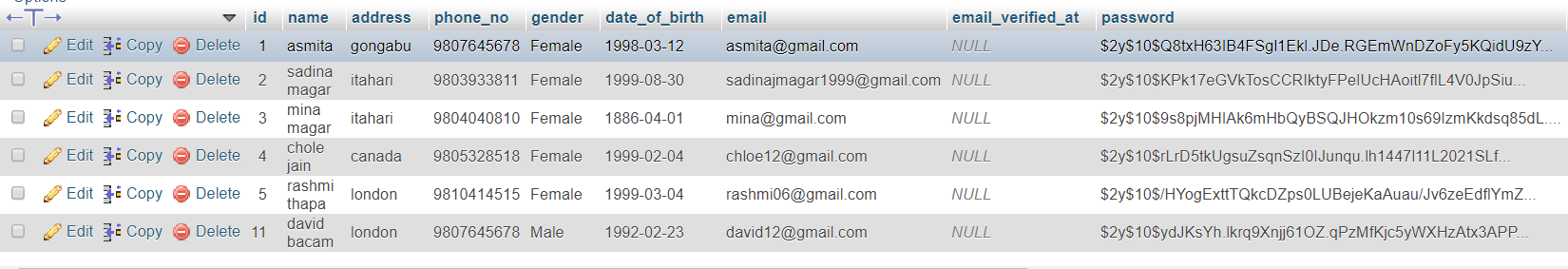
**Data Source: User Objectives: To check the user valid registration.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case** | **Description** | **Tasks** | **Expected Result** | **Actual Result** | **Conclusion** |
| 2.1 | Valid user  Attempt the registration. | User input all the valid registration details. | User registered successfully | Successfully user is registered | Pass |

Output:







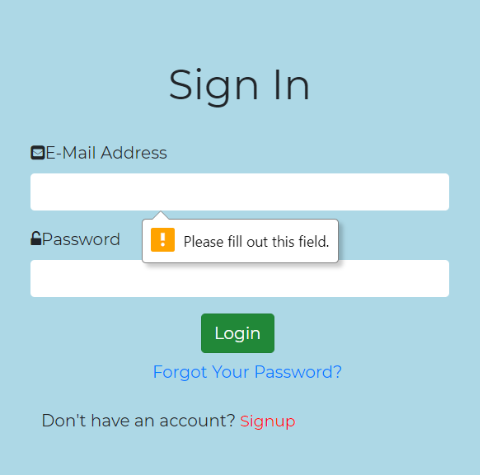
**3**. **Login**

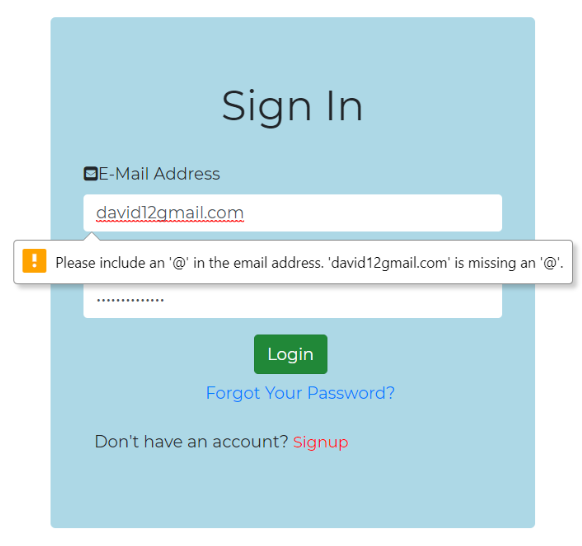
**Unit Test: 3 Test Class: Validation of login**

**Data Source: User Objectives: To check the validation level of login.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case** | **Description** | **Tasks** | **Expected Result** | **Actual Result** | **Conclusion** |
| 3.1 | Providing the blank value for login. | User doesn’t input the value for the login. | Throw error message with “Please fill out this field”. | Successfully shown error message i.e. “please fill out this field”. | Pass |
| 3.2 | Must provide email in an email format. | User doesn’t provide email in an email format. | Throw error message with include @ in email address validation. | Successfully shown error message. | Pass |

Output:





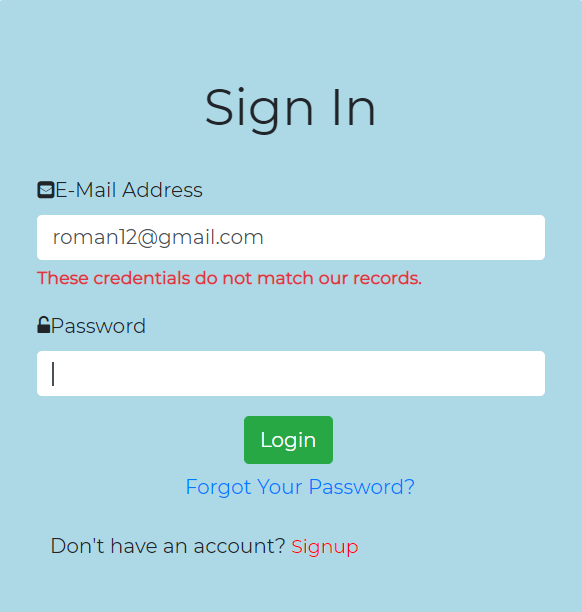
**4.** **Invalid user** **login**

**Unit Test: 4 Test Class: Invalid user**

**Data Source: User Objectives: To check the invalid user login.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case** | **Description** | **Tasks** | **Expected Result** | **Actual Result** | **Conclusion** |
| 4.1 | Invalid user attempt the login | User input wrong email and password. | Throw error message with “These credentials do not match our records”. | Successfully shown error message. | Pass |

**Output:**

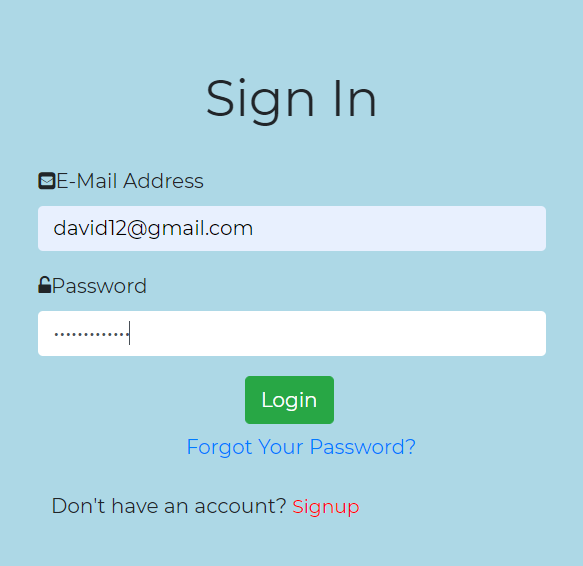


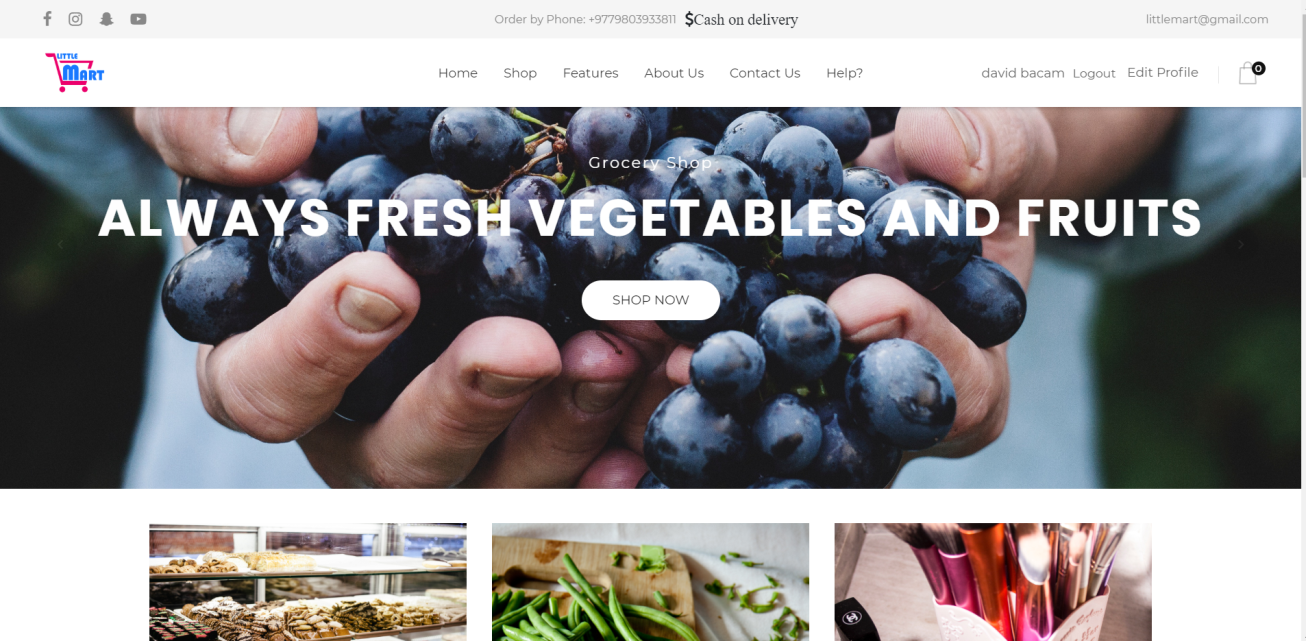
**5. Valid login**

**Unit Test: 5 Test Class: Test valid user**

**Data Source: User Objectives: To check the valid user login.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case** | **Description** | **Tasks** | **Expected Result** | **Actual Result** | **Conclusion** |
| 5.1 | Valid user attempt the login | User input both valid email and password. | Open home page | Successfully open home page | Pass |

Output:

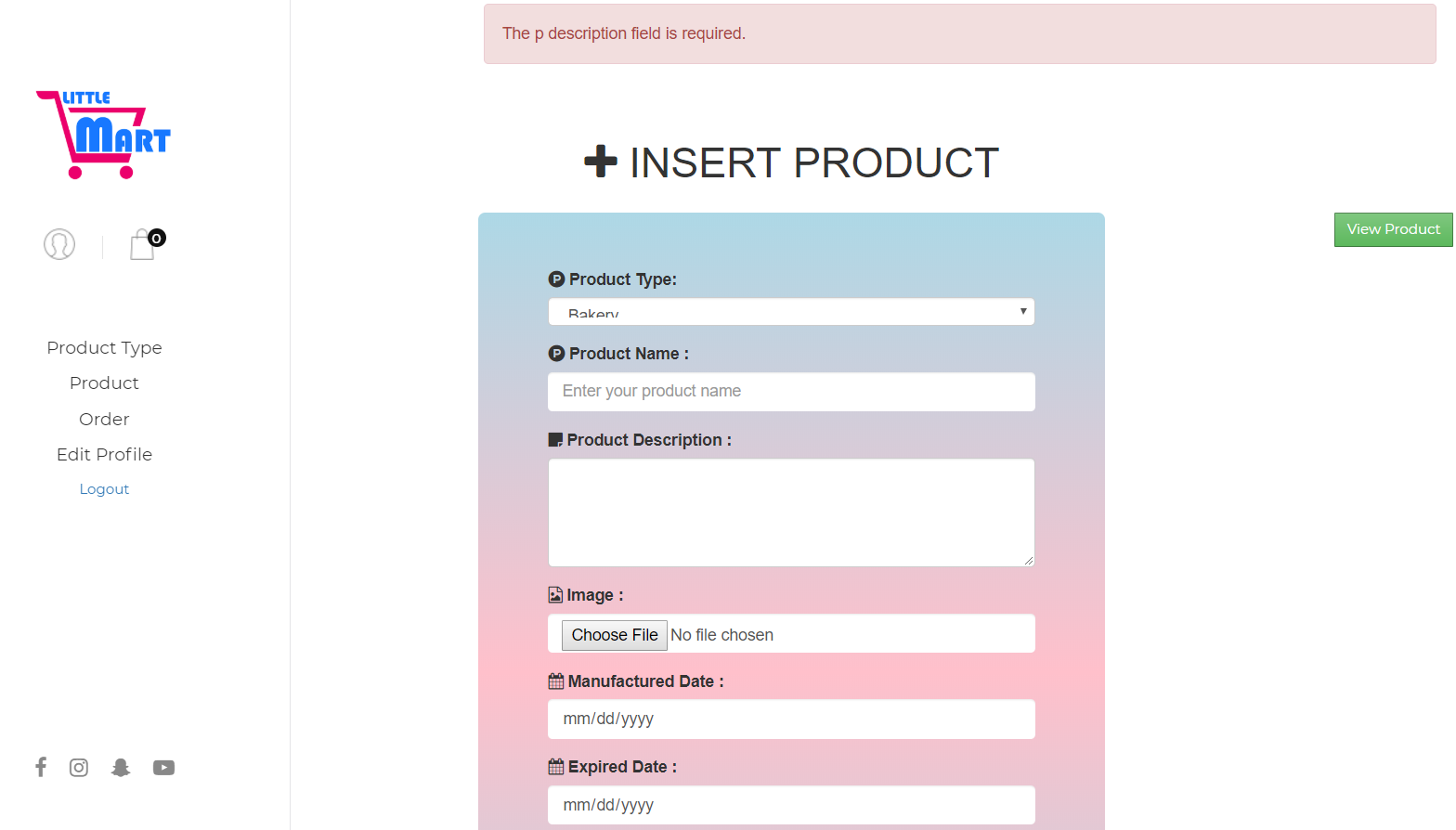


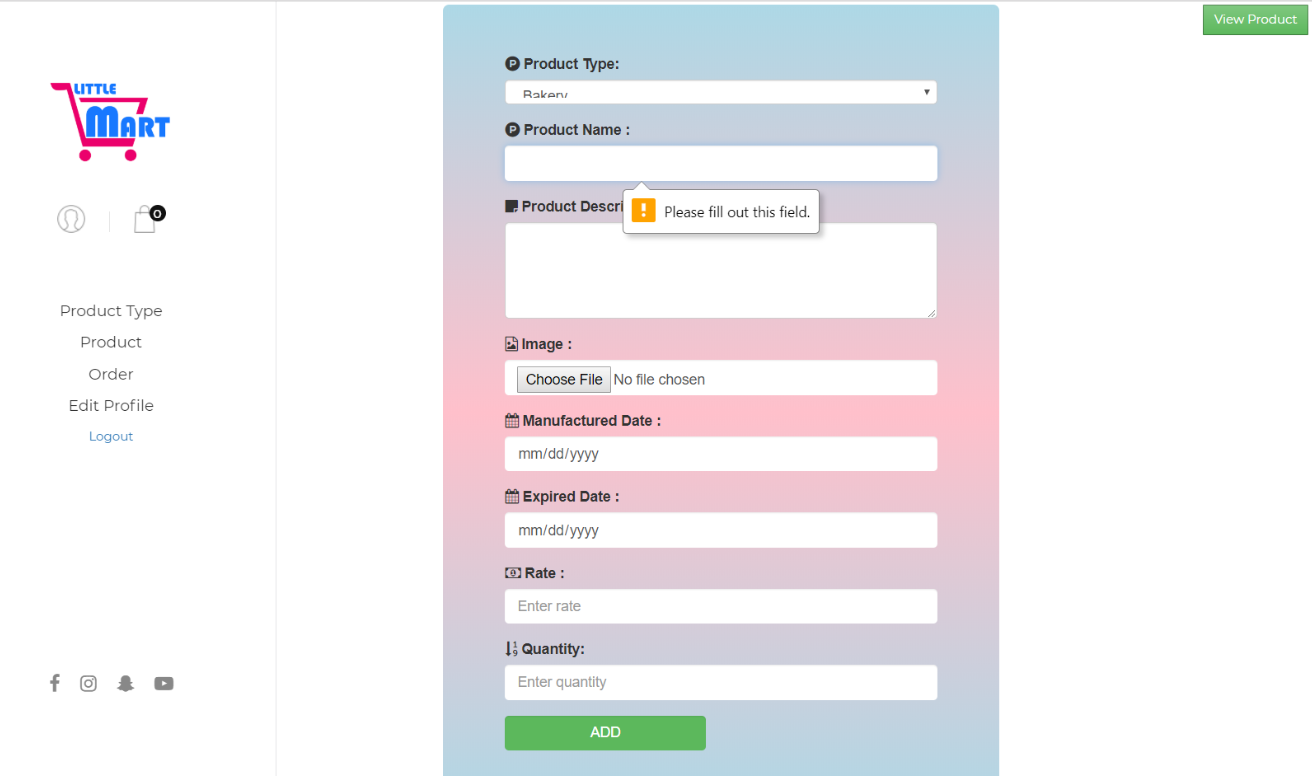
**6. Product**

**Unit Test: 6 Test Class: Validation of product form**

**Data Source: Admin Objectives: To check the validation level of product form.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case** | **Description** | **Tasks** | **Expected Result** | **Actual Result** | **Conclusion** |
| 6.1 | Description must be provided | Admin doesn’t provide description to the product | Throw error message | Successfully shown error message | Pass |
| 6.2 | Providing the blank details in product form. | Admin doesn’t input any details. | Throw error message with “Please fill out this field”. | Successfully shown error message i.e. “please fill out this field”. | Pass |

Output:



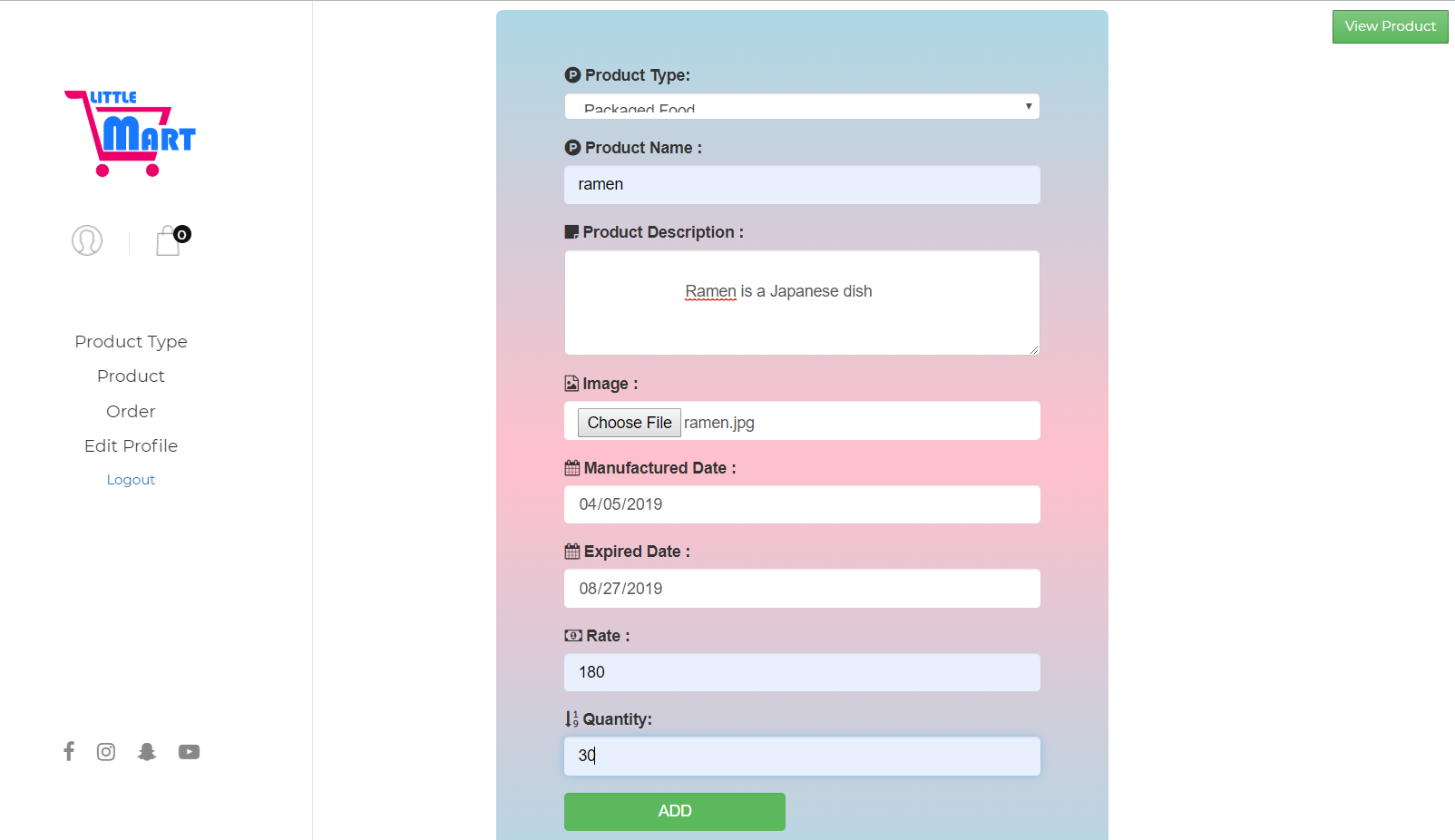
**7. Add Product**

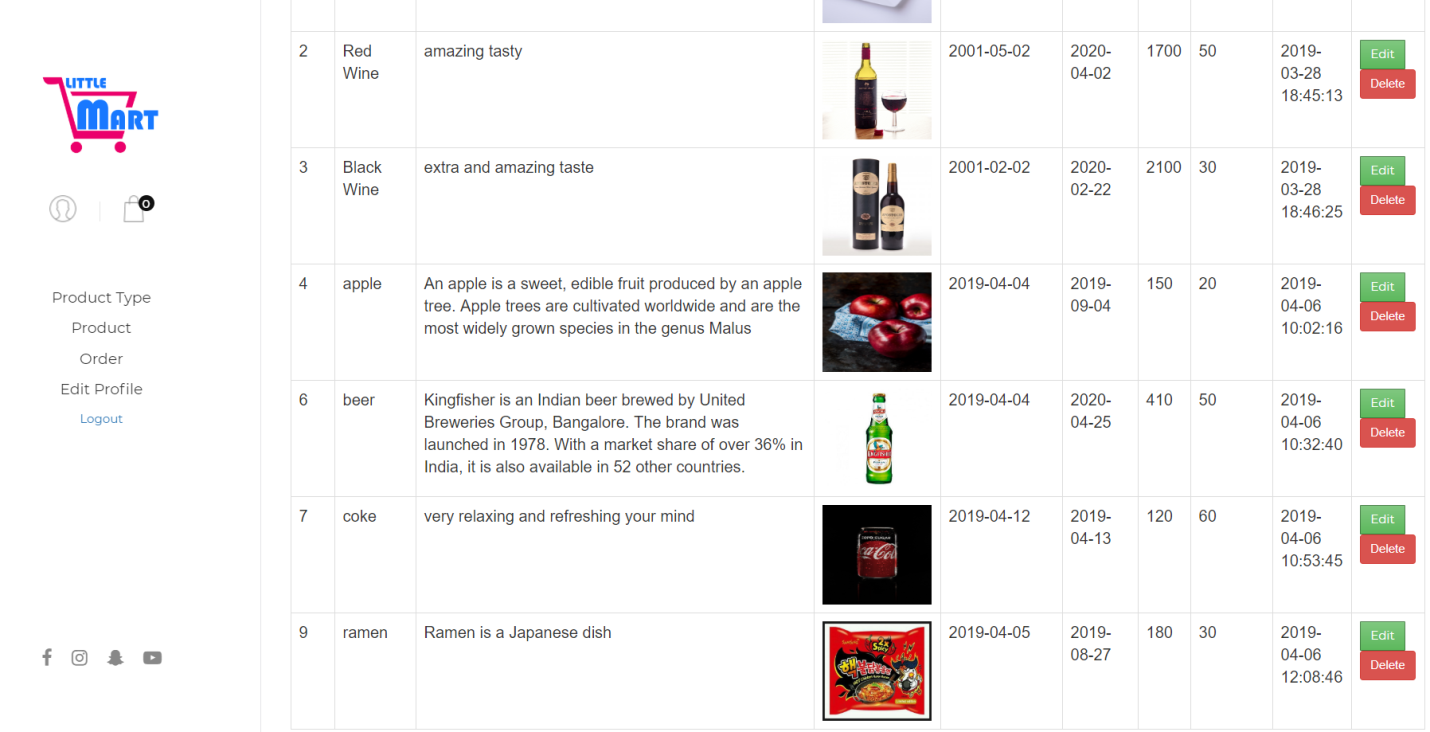
**Unit Test: 7 Test Class: Add product by admin**

**Data Source: Admin Objectives: To add product by admin.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case** | **Description** | **Tasks** | **Expected Result** | **Actual Result** | **Conclusion** |
| 7.1 | Valid product detail added | Admin gives the valid details and image of the product | Product is added. | Successfully added in the database. | Pass |

Output:



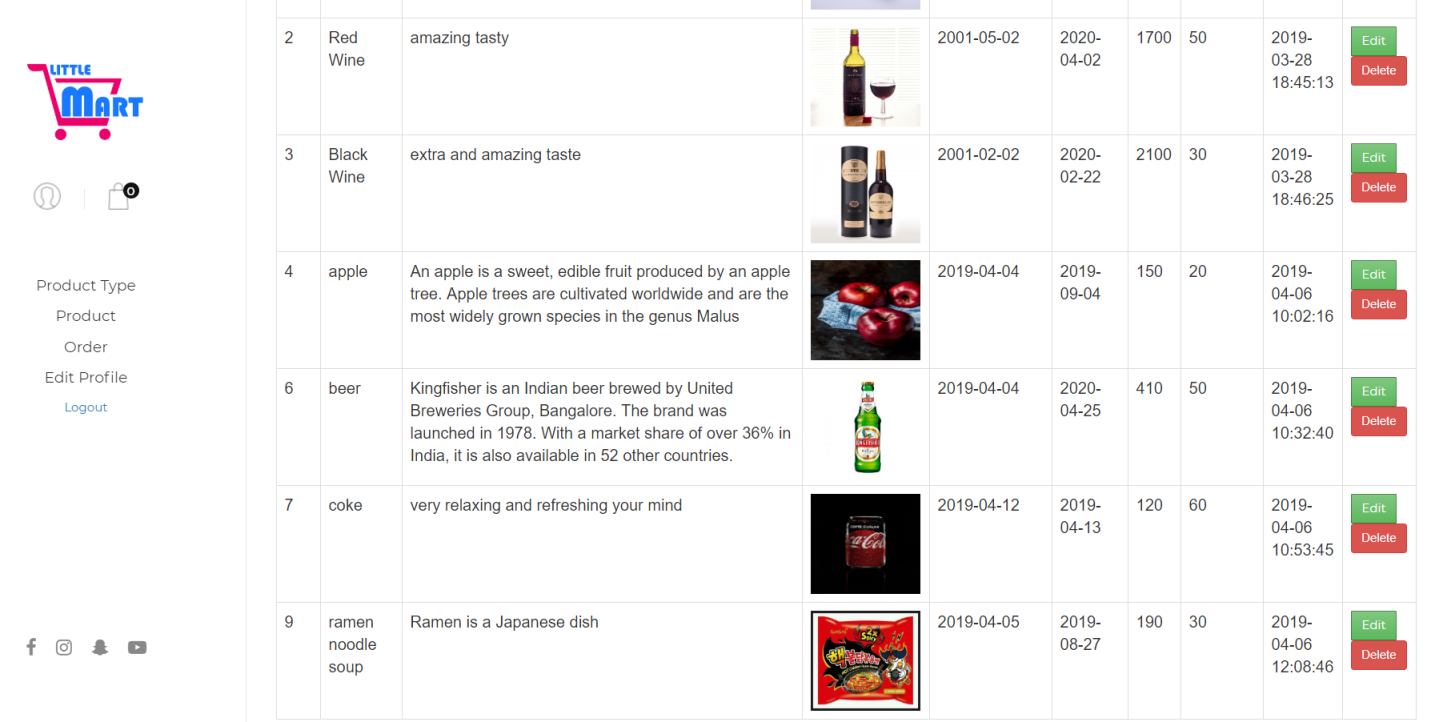


**8.**  **Update Product**

**Unit Test: 8 Test Class: Update product by admin**

**Data Source: Admin Objectives: To update product by admin.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case** | **Description** | **Tasks** | **Expected Result** | **Actual Result** | **Conclusion** |
| 8.1 | Update product detail | Admin update the details of the product | Successfully update product and show in a table | Successfully product is updated and shown in a table | Pass |



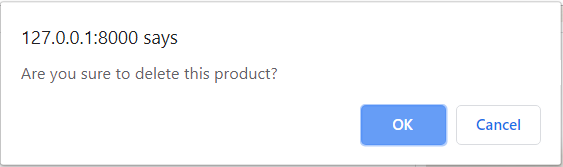
**9.**  **Delete Product**

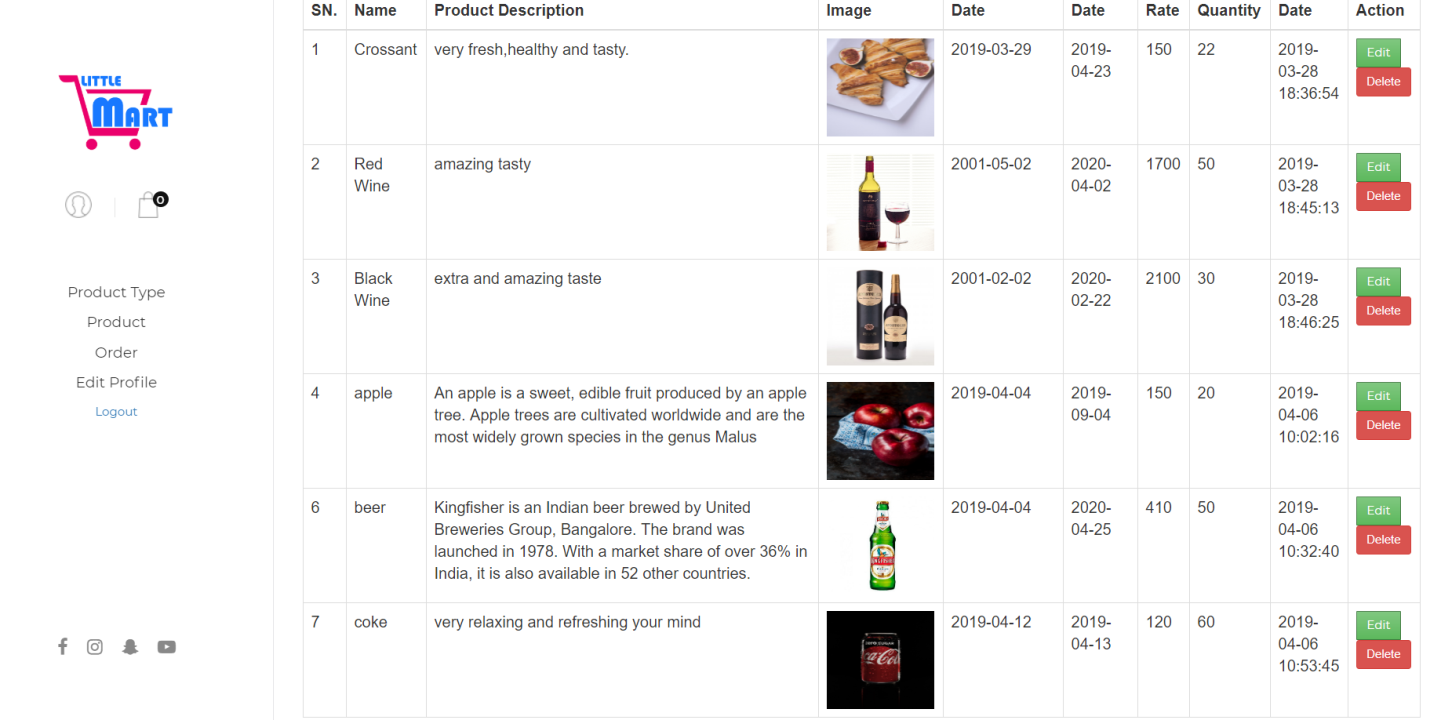
**Unit Test: 9 Test Class: delete product by admin**

**Data Source: Admin Objectives: To delete product by admin.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case** | **Description** | **Tasks** | **Expected Result** | **Actual Result** | **Conclusion** |
| 9.1 | Delete product detail | Admin delete the details of the product | Delete product successfully | Successfully product is deleted | Pass |

Output:





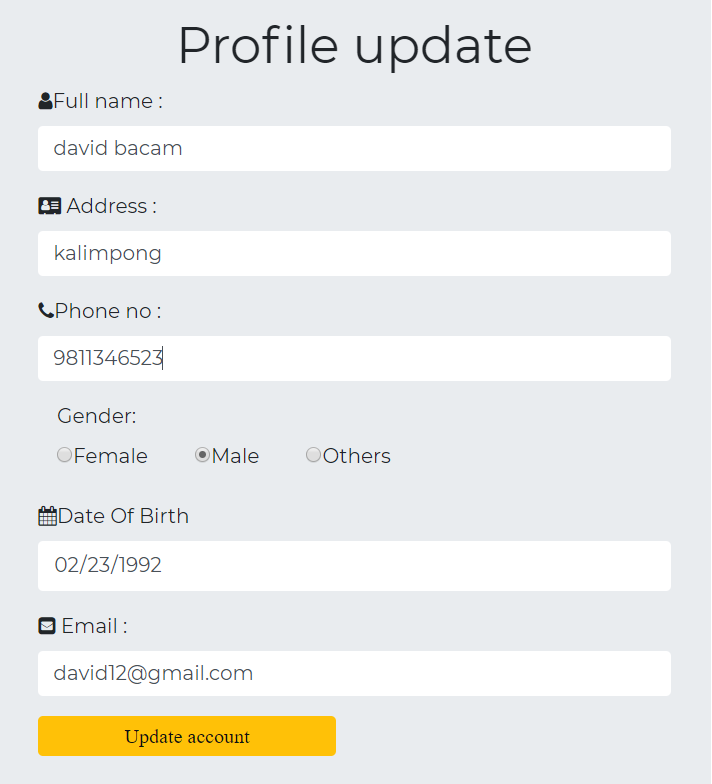
**10.**  **Edit profile**

**Unit Test: 10 Test Class: User edits profile**

**Data Source: User Objectives: To check the user edits profile.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case** | **Description** | **Tasks** | **Expected Result** | **Actual Result** | **Conclusion** |
| 10.1 | Valid user updates their profile. | User provide the updating profile details | Successfully update profile | Successfully profile is updated | Pass |

Output:



Phone no is updated

Address is updated

