## **End to End testing**

### Test Approach:

* We can use a TDD Framework with TestNG and Selenium Java for E2E testing.
* Test Data can be stored in Excel. Each row representing one End to End Testing.
* In TestNG Classes, test methods could test small unit tests sequentially depending on the results of previous test methods completing an end to end scenario.

### Test Plans:

Scenario 1: Filter Products- In Product Page, products should be filtered based on Brand and Type. Product names and prices of filtered products should be validated against expected value.

Scenario 2: Add to Cart- In Product Page, desired products should be added to basket. In Basket product names and prices of selected products should be validated. Desired Quantities of each product should be selected and updated total price of each product should be validated against calculated value. Total Cart Amount should be validated against calculated value. Total Quantity in basket should be validated.

Total Product Price = Price of Product \* Quantity of Product

Total Cart Amount = SUM (Total product Prices)

Total Basket Quantity = SUM (Quantity of Products)

Scenario 3: Checkout and Login - In Basket Page, after selection of products, should click Checkout button to be redirected to login page. After provide valid credentials and logging in, User should be navigated to Review Page. Product details and basket count should be validated against expected value. Should click Pay Now button to complete checkout. Validate completion message.

Scenario 4: Order Details Validations - In My Orders page of user, validate that completed order details is present. Compare the order details with the expected value.

Test Data: Sample values to consider during testing:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Products** | **Product Price** | **Required Quantities** | **Total Product Price** | **Total Quantity** | **Total** |
| 1  2 | Cup<T> White Mug Roslyn Red Sheet | $ 12.00 $ 8.50 | 2 3 | $ 24.00 $ 25.50 | 5 | 49.5 |

## **Unit Testing**

### Test Approach:

We can use a BDD Framework with Cucumber, Junit and Selenium Java for unit testing. Scenario Outline can be used for data driven testing where same test has to be done with multiple sets of data. Test Data can be stored in Excel. Each row representing one Unit Testing.

### Test Plan:

#### Filter Validation Feature:

Scenario Outline : Filter Validation - Multiple combinations and Brand and Type should be used to filter products in In Product Page. Product names and prices of filtered products in each combination should be validated against expected values.

Test Data: Sample values to consider during testing

|  |  |  |  |
| --- | --- | --- | --- |
| Brand Option | Type Option | Products List | Product Prices |
| .NET | Mug | .NET Black & White Mug | $ 8.50 |
| Other | Mug | Cup<T> White Mug | $ 12.00 |
| All | Sheet | Roslyn Red Sheet, .NET Foundation Sheet, Cup<T> Sheet | $ 8.50, $ 12.00, $ 8.50 |

#### Login Feature:

Scenario 1: User try to login without username and password. Validate if user gets email not present error message and password not present error message.

Scenario 2: User try to login without username. Validate if user gets user should get email not present error messages

Scenario 3: User try to login without password. Validate if user gets password not present error message.

Scenario 4: User try to login with invalid email ID. Validate if user gets invalid login attempt error messages.

Scenario 5: User try to login with invalid password. Validate if user gets invalid login attempt error messages

Scenario 6: User try to login with correct username and password. Validate User is logged in

|  |  |
| --- | --- |
| **loginEmail** | **loginPassword** |
| [demouser@microsoft.com](mailto:demouser@microsoft.com) | [Pass@word1](mailto:Pass@word1) |