Limehome Assessment Solution

1) Test cases of automation practice

After analysis the requirements and approach, below are the test case format

Test Case ID	TC_01	**Test Case Description**	Verify that adding products to cart functionality are working fine		
Created By	Raza	**Reviewed By**		**Browser Version**	125

Test

QA Tester's Log

Tester Name	Raza	Date Tested	09-07-2024	Test Cases (Pass/Fail/ Not Executed)	

Sr #	Prerequisites:
1	Browser is open
2	
3	
4	

Test Environment:

Operating System: MacOS, Linux, Windows
Browser: Google Chrome, Firefox, Safari
Test Framework: Playwright with Page object model (Design Framework)
Automation Test: High Priority

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / No executed / Suspended
1	Navigate to Automation website	User should redirect to website	User is redirecting to website successfully	Pass
2	Search for specific Product (T-shirts) by entering relevant information in search bar	User should able to search product	User is able to search product in search are successfully	Pass
3	Click on the search icon or press Enter to initiate the search	User should able to click on search button	Yes user is able to click on search button	Pass
4	Click on the desired product to access its details page	User should able to view detail page of product	Yes user is able to redirect to detail page of the product and check details	Pass

5	Verify that all the necessary information about the product is displayed correctly.	User should able to see details of product like Name, pic etc	Users are able to get detail of product like price, name, description etc	Pass
6	Select the desired quantity for the product	User is able to select product quantity from dropdown	Yes user are able to select quantity of product like 5 etc	Pass
7	Click on the "Add to Cart/Shopping cart" button	User should able to redirect to shopping cart Page	Users are able to clickand redirect to shopping cart page	Pass
8	Verify that the cart page is displayed with the added product	User should able to view the product in cart page	Users are able to seethe product in cart page	Pass
9	Verify that the product name, price, and quantity are correctly displayed in the cart.	User should able to view the details of product like name, price etc	User are able to see the product details and validate it	Pass
10	Verify that the total price in the cart is updated accurately based on the added product	User should see updated price value based on the quantity of product	User are able to see the updated price value based on the quantity selections of product	Pass

Test Case ID	SA_02	Test Case Description	Verify the checkout notification	
Created By	Raza	Reviewed By		Version

Test Scenario	Verify that user get notification of purchase

Manual Test

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Notexecuted / Suspended
1	Navigate to webstore Website	User should redirect to website	User is redirectingto website successfully	Pass
2	Verify the product details in add to cart and click on purchase	User should able tosee details of purchase product	User is able to view the details of product purchase	Pass
3	User should get notification of purchase product	User should able to get confirmation of product purchase	User should able to get confirmation of product purchase	Pass

Test Cases automated:

- Adding Product
- Viewing Cart
- Checkout Confirmation & Notification
- Payment Processing
- Handling unexpected challenges
- Filtered products

Testing Approach:

While working in that project, I have considered these below points for making the effective product and quality.

a. Scope and overview
b. Test Approach
c. Test Environment
d. Testing tools
e. Review and approvals
f. Conclusion

1) Scope and Overview:

The scope of this test approach is to validate the "Search and Shopping product" functionality of the automation Practice. This includes testing the ability to add products to the cart, display relevant product information, and select the quantity of the product. The testing will cover both positive and negative scenarios to ensure the functionality meets the requirements and delivers a seamless userexperience.

2) Test Automation Approach & Plan:

In automation plan below are the major important points to cover

- **Automation testing:**

  While automating this task, I have covered the features like Search item, listItem, add to Cart, Quantity of product, checkout Product but need to cover performance testing and security testing for more scalability and reliability of the system

- **Tool and Framework to be used:**

  For this task, I have used Playwright tool with with POM (Page Object model framework) using Allure report. Based on the acceptance criteria, have ensured validations and assertions are mapping accordingly. **Below are the automation tools and framework list is used.**

  - Playwright
  - Maven POM
  - POM (Page Object model)
  - Yarn
  - Azure Kubernetes (CI/CD pipeline is set up in Azure DevOps, automated tests are configured & executed).—Future Plan

The benefits of using the Page object model framework is

- It makes ease in maintaining the code (flow in the UI is separated from verification)
- Makes code readable (Methods get more realistic names)
- Makes the code reusable (object repository is independent of test cases)
- The Code becomes less and optimized.

For CI/CD Pipeline with Microsoft Azure , the following steps will be followed:

- Create a Pipeline:
- Set up a pipeline in Azure DevOps.
- Define build steps to compile the code and run automated tests.
- Deploy to Azure.
- Azure Monitor for logging and monitoring

Configure Pipeline YAML:

```
trigger:
- main

pool:
  vmImage: 'ubuntu-latest'

steps:
- task: UseNode@2
  inputs:
    version: '14.x'

- script: |
    npm install
    npm run build
displayName: 'Build Angular
Frontend'

- task: Maven@3
  inputs:
    mavenPomFile: 'pom.xml'
    goals: 'package'
  displayName: 'Build Java Backend'

- task: AzureWebApp@1
  inputs:
    azureSubscription: '< Azure
Subscription>'
    appType: 'webApp'
    appName: '<Our Web App
Name>'
    package:
'$(System.DefaultWorkingDirectory)
/**/*.jar'
  displayName: 'Deploy Backend to
Azure'

- task: AzureRmWebAppDeployment@4
inputs:
azureSubscription: '<Azure Subscription>'
WebAppName: '<Web App Name>'
Package:
'$(System.DefaultWorkingDirectory)/dist'
displayName: 'Deploy Frontend to Azure'

- task: Playwright@1
 inputs:
 gridUrl: 'http://<Your Selenium Grid URL>'
displayName: 'Run Playwright Tests'

- task: PostmanCollectionRunner@0
inputs:
collectionFileSource: 'postman_collections/*.json'
 displayName: 'Run Postman Tests'
```

3) Test Environment:

The testing will be performed on the following test environment:

Env: QA, Stagging, UAT & Production

Browsers: Chrome, Firefox, Safari

Operating Systems: Windows, macOS, Linux

website: http://www.automationpractice.pl/index.php

4) Review and approvals:

The test cases and test artifacts will undergo review and approval by the following stakeholders:

Test Lead: To ensure the test approach aligns with project objectives and quality standards.

Product Owner: To validate the test coverage and ensure it meets the requirements.

Development Team: To provide insights and feedback on the technical feasibility and implementation details.

Conclusion:

In conclusion, this test plan is designed to ensure the functionality, performance, and usability of the webstore (automation practice) application where users can browse, view details, and purchase forklifts. The combination of automated tests ensures thorough coverage of both frontend and backend functionalities.

Thanks !