Readme

Task and deliverables.

Test Strategy Plan

1. Objective:

The main objective of this test strategy plan is to define the scope, approach, and methodology for comprehensive testing of e-commerce, ensuring its functionality, performance, security, and usability, and ensuring a satisfactory shopping experience for users.

2. Scope:

Testing will cover all aspects of the e-commerce, including:

Functionalities: User registration and login, category and product navigation, product search, search filters, product details display, adding to cart, checkout process, payments, account creation and deletion, address management, etc.

Performance: Page loading speed, response times, optimization for mobile devices, ability to handle multiple simultaneous users.

Security: Protection of user data, input validation, and encryption of sensitive data.

Usability: Intuitive design, easy navigation, smooth user experience, accessibility for people with disabilities.

3. Approach:

An exploratory testing approach will be used, prioritizing testing of critical functionalities and those most likely to exhibit bugs. Various testing techniques will be employed, including:

Functional testing: Verify that e-commerce functionalities behave as expected.

Performance testing: Evaluate e-commerce performance under load and different conditions.

Security testing: Identify and correct security vulnerabilities in the e-commerce.

Usability testing: Evaluate the usability and user experience of the e-commerce.

Compatibility testing: Verify that the e-commerce works correctly in different browsers and devices.

4. Methodology:

Testing will be carried out following a defined process, which includes:

Planning: Definition of scope, objectives, test cases, and resources needed for testing.

Test case design: Creation of detailed test cases covering all e-commerce functionalities.

Test execution: Execution of test cases using automation tools and manuals.

Bug logging: Documentation of bugs found during testing.

Failure analysis: Failure analysis to determine root cause and priority.

Re-testing: Re-execution of tests to verify that bugs have been corrected.

Some Test Cases defined

TC 01- Validate User Registration.

**Steps:**

-Navigate to the home page.

-Click on the 'Signup / Login' button.

- Enter name and email address

-Press Signup

- Complete all Account Information

- Complete all Address Information

- Pres Create account button

**Expected Result:**

A confirmation message is displayed indicating the account created.

The user is logged in to their new account.

--------------------------------------------------

TC-02: Verify that the product search function works correctly.

**Steps:**

Navigate to the product search page.

Enter a valid product search term (e.g., "Blue Top").

Click the "Search" button.

**Expected Result:**

A list of relevant products matching the search term is displayed.

Each product listing includes product name, image, price, and other relevant information.

----------------------------------

TC-03: Verify that users can successfully add products to their shopping cart.

**Prerequisites:**

User is logged in to their account.

A product is selected and viewed.

**Steps:**

- Select the desired quantity of the product.

- Click the "Add to Cart" button.

**Expected Result:**

A confirmation message is displayed indicating the product has been added to the cart.

The cart icon reflects the updated number of items in the cart.

-----------------------------------------------------

TC-04: Verify that the checkout process is seamless and allows users to complete their purchases successfully.

**Prerequisites:**

User is logged in to their account.

Items are added to the shopping cart.

**Steps:**

- Navigate to the checkout page.

- Review order details

- Select a payment method.

- Confirm the purchase.

**Expected Result:**

An order confirmation page is displayed.

--------------------------------------------

TC-05: Verify that the login process fails for invalid credentials.

**Steps:**

Navigate to the login page.

Enter an invalid or existing email address.

Enter an incorrect password.

Click the "Login" button.

**Expected Result:**

An error message is displayed indicating invalid login credentials.

------------------------------------

**TC-06: Verify that users can successfully delete their accounts.**

**Prerequisites:**

User is logged in to their account.

**Steps:**

Press Delete Account Link

**Expected Result:**

A confirmation message is displayed indicating successful account deletion.

The user is automatically logged out of the website.

Attempting to access the account with the deleted credentials results in an error message or login failure.

----------------------------------------------

TC-07 Verify that users can successfully remove unwanted items from their shopping cart.

**Prerequisites:**

User is logged in to their account

At least one product has been added to the shopping cart.

**Steps:**

Navigate to the shopping cart page.

Locate the product to be removed.

Press the X button to remove the item

**Expected Result:**

The product is no longer listed in the shopping cart.

-----------------------------------------------------

TC-08: Verify that users can submit the contact us form.

**Steps:**

Access the contact us page.

Fill out the form with valid information

Press Submit

Press Ok in the Popup.

**Expected Result:**

A confirmation message is displayed on the page after successful submission.

---------------------------------------------------------------------

Framework

1- Used cypress framework and IntelliJ like IDE.

2-Framework Structure

Folders

e2e- contains the file with the test

fixture- contains the data that will be used in the test

reports- includes the report of the final execution of the tests (see report in /mochareports/report.html)

support- contains the folder of all page objects created and common commands to be reusable. It also has an environment file where you can define the URL that you can visit.

videos- contains the video of each test executed

package.json- contains all dependencies used in the project and a script of execution of tests.

**To run the project:**

1- Unzip the project

2- Open a console in the root of the project

3- run npm install to install cypress and the dependencies of the package.json

4- run npm test to launch the test

5- When the execution is finished you can see the report in /mochareports/report.html and a video in /videos