TEST PLAN

- SWAG LABS-

Valpovo, May 2024.

Table of Contents

[1. Introduction 3](#_Toc167530602)

[2. Scope of Testing 3](#_Toc167530603)

[3. Testing Environment Setup 3](#_Toc167530604)

[3.1. Tools and Frameworks 3](#_Toc167530605)

[3.2. Browser Configuration 4](#_Toc167530606)

[3.3 Installation and Setup 4](#_Toc167530607)

[4. Dependencies 4](#_Toc167530608)

[5. Test Execution 4](#_Toc167530609)

[6. Test Plan Outline 5](#_Toc167530610)

[7. Additional Considerations 8](#_Toc167530611)

[8. Conclusion 8](#_Toc167530612)

# 1. Introduction

This document outlines a test plan for automated testing of the Swag Labs website (<https://www.saucedemo.com/>) using the Selenium testing framework with C#. The test plan includes the scope, testing methodologies, environment setup, different dependencies, and responsibilities. The objective is to ensure the website’s functionality, performance, and security through systematic testing.

# 2. Scope of Testing

|  |  |
| --- | --- |
| Test type | Description |
| Functionality Testing | Verify all user interactions such as login, adding/removing items from the shopping cart, and checkout process. Example: Ensure users can successfully log in with valid credentials and perform transactions. |
| Compatibility Testing | Ensure the website functions correctly across different browsers and devices. Example: Test the site on Google Chrome, Mozilla Firefox, and Microsoft Edge to ensure consistent behavior. |
| Performance Testing | Check the website’s performance under various conditions, such as load testing to see how the site performs with multiple users simultaneously. |
| Security Testing | Validate security measures such as user authentication and data protection. Example: Ensure secure login processes and protect user data from unauthorized access. |

# 3. Testing Environment Setup

## 3.1. Tools and Frameworks

* Visual Studio 2022
  + IDE for writing and managing test scripts
* Selenium WebDriver
  + Tool for automating web application testing
* Browser drivers
  + Required for running tests on different browsers
* NUnit
  + Testing framework for writing and running tests in C#

## 3.2. Browser Configuration

* Google Chrome (latest version)
* Mozilla Firefox (latest version)
* Microsoft Edge (latest version)

## 3.3 Installation and Setup

* Install Visual Studio 2022.
* Add Selenium WebDriver and browser drivers through NuGet Package Manager
* Set up NUnit by installing the NUnit and NUnit3TestAdapter packages via NuGet

# 4. Dependencies

* NuGet Packages
  + Selenium.WebDriver
    - For controlling browser operations
  + Selenium.Support
    - For additional Selenium functionalities
  + NUnit
    - For writing and managing test cases
* External Libraries
  + Any additional libraries that might be required for reading test data (e.g., Excel)

# 5. Test Execution

* Tester: Branimir Križanec
* Responsibilities:
  + Write and execute functional, compatibility, and performance tests
  + Document and track test cases and results using Trello and Excel
  + Follow a structured schedule to ensure all scenarios are covered
  + Analyze and report test results, addressing issues promptly to ensure application quality
* Execution Schedule:
  + Tests will be run after each major build
  + Daily smoke tests to check basic functionality
  + Weekly regression tests to ensure new changes don’t affect existing functionalities

# 6. Test Plan Outline

The following tables serve to roughly outline the Test Scenarios for the web page.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Login Page Test Scenario | Steps to Execute | Expected Result |
| 1. | Verify Login Form – Enabled Elements | 1. Navigate to the login page  2. Verify if both username and password fields, and login button are displayed and enabled. | All elements are displayed and enabled. |
| 2. | Verify Login Form - Placeholders | 1. Navigate to the login page  2. Verify placeholder text for username and password fields | Placeholder text for username and password is correct. |
| 3. | Verify Login – Test Valid Credentials | 1. Navigate to the login page  2. Enter valid username and password  3. Click on login button  4. Verify successful login | User is logged in successfully and navigated to the home page. |
| 4. | Verify Login – Test Invalid Credentials | 1. Navigate to the login page  2. Enter invalid username and/or password  3. Click on login button  4. Verify error message is displayed | Error message is displayed indicating invalid credentials. |
| 5. | Verify Login – Account lockout | 1. Navigate to the login page  2. Enter valid username and password of a user that was locked out  3. Click on login button | Error message is displayed indicating the locked-out status. |

|  |  |  |  |
| --- | --- | --- | --- |
| # | Home Page Test Scenario | Steps to Execute | Expected Result |
| 1. | Verify Inventory Items | 1. Navigate to the home page  2. Verify all inventory items are displayed with descriptions, images, titles, and prices  3. Click on each item to verify navigation to item details | Inventory items are displayed correctly and are clickable to navigate to details page. |
| 2. | Verify Cart Functionality | 1. Navigate to the home page  2. Add items to cart  3. Verify items are added to cart  4. Verify that the cart is properly displaying the added items  5. Remove items from the cart  6. Verify items are removed from the cart  7. Verify that the cart is properly displaying the removed items  8. Click the cart button link to verify that the link takes us to the cart page | Items can be added and removed from the cart, with the cart icon adjusting as the user manipulates the items. The cart button is clickable and takes us to the cart page. |
| 3. | Verify Main Menu Navigation | 1. Navigate to the home page  2. Separately click on the elements to verify their functionality | The All Items tab should return the user to the home page.  The About tab should navigate the user to a different webpage.  The Logout tab should return the user to the login page and log them out of the website.  The Reset App State should reset the app to its default values and empty the Cart. |
| 4. | Verify Sorting Dropdown Menu | 1. Navigate to the home page  2. Iterate through the different sorting options  3. Verify that the items on the home page are sorted according to the option selected | Items are sorted correctly based on the selected option. |
| 5. | Verify Footer Social Network Links | 1. Navigate to the home page  2. Verify that the social network links are clickable  3. Click each item to verify navigation | Footer links are clickable and navigate to the correct pages. |

|  |  |  |  |
| --- | --- | --- | --- |
| # | Cart Page Test Scenario | Steps to Execute | Expected Result |
| 1. | Verify Removing Items from Cart | 1. Navigate to the Cart page  2. Remove an item from the cart  3. Verify that the item is removed, and the cart icon updates accordingly | Item is removed from the cart and the cart icon updates. |
| 2. | Verify Cart Item Details | 1. Navigate to the Cart page  2. Verify each item in the cart has correct details (name, quantity, price)  3. Verify the total price updates accordingly | Cart items display correct details (name, quantity, price) |
| 3. | Verify Cart Quantity Update | 1. Navigate to the Cart page  2. Change the quantity of an item | Total price updates according to the new quantity of the item. |
| 4. | Verify Cart Persistence | 1. Add items to the cart  2. Log out and log back in  3. Navigate to the cart page  4. Verify items are still in the cart | Items remain in the cart after logging out and logging back in. |
| 5. | Verify Continue Shopping Button | 1. Navigate to the Cart page  2. Click on “Continue shopping” button  3. Verify navigation back to the Home page | User is navigated back to the Home page. |
| 6. | Verify Checkout Button | 1. Navigate to the Cart Page  2. Click on “Checkout” button  3. Verify navigation to the Checkout page | User is navigated to the Checkout page |

|  |  |  |  |
| --- | --- | --- | --- |
| # | Checkout Page Test Scenario | Steps to Execute | Expected Result |
| 1. | Verify Checkout Form – Enabled Elements | 1. Navigate to the Checkout page  2. Verify that all input fields (first name, last name, postal code) are displayed and enabled | All input fields are displayed and enabled. |
| 2. | Verify Checkout Form - Placeholders | 1. Navigate to the Checkout page  2. Verify placeholder text for all input fields (first name, last name, postal code) | Placeholder texts for input fields are correct. |
| 3. | Verify Checkout with Valid Information | 1. Navigate to the Checkout page  2. Enter valid information in the input fields  3. Click on “Continue” button  4. Verify navigation to the overview page | User is navigated to the overview page. |
| 4. | Verify Checkout with Invalid Information | 1. Navigate to the Checkout page  2. Enter invalid or incomplete information in the input fields  3. Click on “Continue” button  4. Verify error message is displayed | Error message is displayed indicating invalid or incomplete information. |
| 5. | Verify Cancel Button | 1. Navigate to the Checkout Page  2. Click on the “Cancel” button  3. Verify navigation back to the Cart page | User is navigated back to the Cart page. |
| 6. | Verify Checkout Overview Page | 1. Navigate to the Checkout Overview page  2. Verify the details of the order (items, quantities, prices, total price) are correct | Order details on the overview page are correct. |
| 7. | Verify Finish Button | 1. Navigate to the Checkout Overview Page  2. Click on “Finish” button  3. Verify order completion and navigation to the order confirmation page | Order is completed and user is navigated to the order confirmation page. |
| 8. | Verify Order Confirmation Page | 1. Complete an order  2. Verify that the order confirmation page displays a thank you message and order summary | Order confirmation page displays a thank you message and order summary. |
| 9. | Verify Back to Products Button | 1. Complete an order  2. On the order confirmation page, click on “Back to Products” button  3. Verify navigation back to the Home page | User is navigated back to the Home page. |

# 7. Additional Considerations

* Data-Driven Testing: Use data driven approaches to test with multiple sets of input data
  + Example: Trying to login with different usernames to test various user scenarios
  + Implement data-driven tests using NUnit’s [TestCase] or [TestCaseSource] attributes to handle multiple sets of input data efficiently

# 8. Conclusion

This test plan serves as a foundational guide for automated testing of the Swag Labs website. By following the outlined steps and considering the additional factors, we aim to ensure the website’s reliability, user-friendliness, and security. Regular reviews and updates to the test plan will help maintain its effectiveness and relevance as the project evolves. The Design Requirements and detailed Test Scenarios are supplemented in separate Excel documents.