TEST PLAN

- SWAG LABS-

Valpovo, May 2024.

Table of Contents

[1. Introduction 3](#_Toc167166137)

[2. Scope of Testing 3](#_Toc167166138)

[3. Testing Environment Setup 3](#_Toc167166139)

[3.1. Tools and Frameworks 3](#_Toc167166140)

[3.2. Browser Configuration 4](#_Toc167166141)

[3.3 Installation and Setup 4](#_Toc167166142)

[4. Dependencies 4](#_Toc167166143)

[5. Test Execution 4](#_Toc167166144)

[6. Test Plan Outline 5](#_Toc167166145)

[7. Additional Considerations 5](#_Toc167166146)

[8. Conclusion 5](#_Toc167166147)

# 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
  + 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

|  |  |  |
| --- | --- | --- |
| # | Test | Description |
| 1. | Login | Validate user login functionality with valid credentials |
| 2. | Add Item to Cart | Verify adding items to the shopping cart |
| 3. | Remove Item from Cart | Ensure items can be removed from the cart |
| 4. | Checkout Process | Test the complete checkout process |
| 5. | Logout | Verify user logout functionality |

# 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 are supplemented in a separate Excel document.