

# Automation Test Plan for Swag Lab

## Introduction:

The Automation Test Plan for Swag Labs defines the approach, tools, scope, and responsibilities for automating the test cases of core functionality to improve efficiency and effectiveness of the testing process.

## Objective

- Automate high-priority and repetitive test cases for faster regression cycles.
- Increase test coverage to ensure the application's core functionalities work as expected after updates.
- Establish a framework for reusable, maintainable, and scalable test scripts.

## Scope of Automation:

### In Scope of Automation:

- Functional Regression: Login, product catalogue, cart, checkout, and order history.
- Smoke Testing: Basic workflow validation to ensure critical paths are functional.
- Data-Driven Testing: Verify application behaviour with various data inputs.
- Cross-Browser Testing: Test on key browsers like chrome, Firefox, edge etc to ensure compatibility.

### Out of Scope:

- Exploratory testing.
- Non-functional aspects like performance and load testing.

## Test Automation Strategy

### Approach:

Automation scripts will focus on high-priority workflows and repetitive tasks that can benefit from automation for regression and validation across multiple environments.

- **Framework Selection:** Using a BDD framework approach for flexibility and ease of maintenance and business team can execute the test cases in plain BDD language.

Selenium WebDriver for cross-browser compatibility and BDD approach as business team can write their test cases by calling the re-usable functions and execute the test scripts.

- **Automation Tool and Language:** Selenium WebDriver with Java and cucumber
- **Script Development:**

# Automation Test Plan for Swag Lab

- Reusable Functions: Functions to handle common actions like login, product selection, and checkout.
- Test Data Management: Managing the data using feature files in BDD format.
- **Reporting:**

Test Extent Reports plugin used for Test reporting as it is more user friendly to check the overall test status.

## Test Environment:

### Software Requirements:

- Browsers: Chrome, Firefox, and Edge.
- Operating Systems: Windows, macOS, and Linux.
- Test Automation Tools: Java, Selenium, Junit, Maven dependencies, BDD (Cucumber), IntelliJ IDE

### Hardware Requirements:

- Desktop/Laptop Machines: Sufficient RAM, CPU, and storage for running browsers and testing tools.

## Test Cases for Test automation:

### Search Functionality:

- Verify that search results are accurate and relevant.
- Test various search queries (product names, categories, keywords).
- Validate filters and sorting options.

### Product Browsing:

- Ensure product pages display correct information (price, description, images).
- Test product recommendations and related items.
- Verify availability and stock status.

### Checkout Process:

- Add products to the cart.
- Test different payment methods.
- Validate order confirmation and email notifications.

### Account Management:

- Test user registration and login.
- Verify account settings (address, payment methods).
- Test order history and returns.

### Risks and mitigation:

# Automation Test Plan for Swag Lab

- Usage of real test data may lead to exposing of personal data so always recommended to use synthetic test data for test automation.
- Changes in the web elements can lead to element not found errors. To mitigate this risk the web locators will be organized in a different file in a dynamic way, and we may need some maintenance of script.

## **Deliverables:**

### **Test Summary Report:**

- Summarizes testing activities, results, and metrics.

### **Test Closure Report:**

- Created at the end of testing.
- Includes lessons learned, unresolved issues, and recommendations.

### **Incident/defect Report:**

- Documents defects or incidents found during testing, and it includes details like severity, priority, and steps to reproduce.

## **Conclusion:**

The automation test plan outlines the strategy for automation testing of the Swag Labs application. This plan will ensure that the application is stable and reliable. The plan will also reduce the overall cost of testing and speed up the testing process.