

Test Plan – SauceDemo Web Application

1. Introduction

This test plan explains how manual testing and UI automation will be performed on the SauceDemo website. The goal is to make sure the main features work correctly and that users can complete their tasks without issues.

2. System Under Test (SUT)

The system under test is the SauceDemo web application. It is an e-commerce demo website that allows users to log in, view products, add items to the cart, and complete the checkout process.

URL: <https://www.saucedemo.com>

3. Test Scenarios

3.1 Scenario

You are the QA team for a demo e-commerce website. The application includes user login, product listing, cart management, and a basic checkout flow.

3.2 Key Requirements

- Users must log in before accessing products
- Users can add and remove products from the cart
- Checkout requires entering basic user information
- The system must display confirmation after checkout

4. Test Scope:

4.1 In Scope:

The following features are included in testing:

- User login (valid and invalid credentials)
- Product listing and product details
- Add and remove products from the cart
- Checkout process
- Manual testing and UI automation of critical user flows

4.2 Out of Scope:

The following features are not included in testing:

- API testing
- Performance testing
- Security testing
- Mobile and cross-browser testing
- Database Testing

5. Assumptions

The following assumptions are made for this test plan:

- The SauceDemo website will be accessible during the testing period
- Valid and invalid test user credentials are available
- Product data on the website is preconfigured and stable
- Internet connection will be available during test execution
- Testing will be performed on a demo/test environment, not production

6. Test Objectives

The objectives of testing are:

- To ensure that core functionalities of the website operate correctly
- To validate that users can complete critical actions without errors
- To detect functional and UI issues at early stages
- To reduce the risk of defects reaching end users
- To verify the stability of the application through manual and automated testing

7. Test Types

The following test types will be performed:

- Manual Functional Testing
- UI Automation Testing
- Smoke Testing
- Regression Testing
- Black Box Testing

8. Test Environment

8.1 Manual Testing Environment

Manual testing **will be performed** on the SauceDemo website using the following setup:

- **Application URL:** <https://www.saucedemo.com>
- **Environment:** Demo/Test Environment
- **Browser:** Google Chrome
- **Operating System:** Windows
- **Test Case Management:** Excel Sheets
- **Testing Type:** Manual Web Testing

8.2 UI Automation Testing Environment

UI automation testing **will be performed** using the following tools and environment:

- **Application URL:** <https://www.saucedemo.com>
- **Environment:** Demo/Test Environment
- **Browser:** Google Chrome
- **Operating System:** Windows
- **IDE:** Eclipse
- **Automation Tool:** Selenium WebDriver
- **Test Framework:** TestNG
- **Programming Language:** Java

9. Test Data

The following test data will be used during testing:

- Valid and invalid user credentials
- Sample products available on the website
- Test checkout information (first name, last name, postal code)
- No real personal or payment data will be used

10. Entry Criteria

Testing will start when the following conditions are met:

- The application is accessible and stable
- Test cases are prepared and reviewed
- Test environment is ready
- Required tools are installed and configured

11. Exit Criteria

Testing will be considered complete when:

- All planned manual test cases are executed
- Automated UI tests are executed successfully
- Major defects are identified and documented
- Test results and evidence are collected

12. Risks and Mitigation

<i>Risk</i>	<i>Mitigation</i>
Demo website is unavailable or down	Pause testing, document the issue, and continue testing once the website is available
Internet connection issues	Retry testing when the connection is stable and document delays
Testing machine failure (device crash or shutdown)	Use backups and version control (GitHub) to avoid data loss
Limited testing time	Prioritize critical user flows such as login, cart, and checkout
Automation test instability	Use reliable locators and proper wait strategies

13. Deliverables

The following deliverables will be provided:

- More than 20 manual test cases with execution evidence (Excel sheets)
- More than 8 defect reports with clear steps and screenshots
- UI automation test scripts (Selenium WebDriver + TestNG)
- More than 8 automated UI test cases with evidence and screenshots

14. Test Scenarios (High-Level)

The following high-level test scenarios will be executed to verify the main functionalities of the SauceDemo web application. These scenarios describe what will be tested without detailing execution steps.

1. Verify that a user will be able to log in using valid credentials.
2. Verify that an appropriate error message will be displayed when a user logs in with invalid credentials.
3. Verify system behavior when the username field is left empty during login.
4. Verify system behavior when the password field is left empty during login.
5. Verify that products will be displayed correctly after a successful login.
6. Verify that each product image will match the corresponding product information.
7. Verify that a user will be able to add a product to the shopping cart.
8. Verify that a user will be able to remove a product from the shopping cart.
9. Verify that selected products will be displayed correctly in the shopping cart.
10. Verify that a user will be able to proceed to checkout with products in the cart.
11. Verify that validation messages will be displayed when required checkout information is missing.
12. Verify that an order confirmation page will be displayed after a successful checkout.

15. Signoff

This test plan has been reviewed and approved. Testing activities will be conducted according to this plan.

Asmaa Khasawneh QA Engineer *Asmaa*