Test Plan – SauceDemo (Web)

Prepared by: Mai Taha

Date: 6 January 2026

Scenario

The system under test is a **web-based retail application** that allows users to browse products, manage a shopping cart, and complete the checkout process.

The current testing scope focuses on the following web features:

- User login and authentication
- Product browsing and product details pages
- Shopping cart functionality:
  - Add items to cart
  - Remove items from the cart.
  - Update product quantities
- Checkout flow:
  - Cart review
  - Checkout process
  - Order confirmation display

Key requirement

- Only logged-in users can proceed to checkout
- Users must be able to add, update, and remove products from the cart
- Cart totals must update correctly when quantities change
- The checkout flow must complete successfully without errors
- The confirmation page must display a valid order confirmation message

1) Scope & Objectives

scope

This test plan focuses on testing the **SauceDemo web application** to validate core e-commerce functionality, including user login, product browsing, shopping cart operations, and the checkout flow up to order confirmation. Testing is limited to the web user interface only.

Test Objectives

The objectives of this test plan are to:

Validate core web-based e-commerce functionality
Ensure correct behavior for authenticated and unauthenticated users
Identify functional and usability defects
Verify checkout completion and confirmation display
Ensure overall stability of the web application

2) In-scope / Out-of-scope

in Scope

- Web UI functional testing

- Manual testing of:
    Login
    Product listing
    Cart operations
    Checkout flow
- Basic UI automation for critical user flows
- Defect reporting and documentation

Out of Scope

- API testing
- Backend integration testing
- Performance and load testing at API level
- Security and penetration testing
- Real payment processing
- Mobile application testing

3) Assumptions & Dependencies

Assumptions

- The SauceDemo web application is accessible during the testing period.
- Test user accounts and demo data are available and can be used safely.
- Testing is performed in a demo environment using sample products and data.
- The application behavior remains stable during the test execution period.
- No real customer or payment data is used during testing.

Dependencies

- Successful login depends on the availability of authentication services.
- Checkout completion depends on integrated backend services provided by the demo application.
- The checkout flow relies on a third-party payment gateway operating in sandbox mode.
- Stable internet connectivity is required to execute all web-based test scenarios.

Hint: Document dependencies (APIs, third parties) and assumptions (data, access).

4) Test Approach (levels + types)

Test Levels

System Testing:
End-to-end testing of the web application to validate complete user flows from login to checkout confirmation.

Test Types

Functional Testing: To verify core e-commerce features.

Manual Testing: for detailed validation of user interactions and UI behavior.
Automated UI Testing: For critical user journeys such as login, add to cart, and checkout.
Regression Testing: To ensure existing functionality is not impacted by new changes.
Exploratory Testing: To identify unexpected behavior and usability issues.

5) Test Scenarios Overview

Happy Path Scenarios

The following scenarios represent the expected system behavior when valid data and actions are provided:

1. The user logs in successfully using valid credentials.
2. The user views the product listing page without errors.
3. The user opens a product details page.
4. The user adds a product to the shopping cart.
5. The user updates the quantity of a product in the cart, and the total is updated correctly.
6. The user removes a product from the cart.
7. The logged-in user proceeds to checkout with items in the cart.
8. The user
9. completes the checkout process successfully.
9. The order confirmation page is displayed after checkout completion.

Negative Scenarios

The following scenarios validate system behavior when invalid actions or conditions occur:

1. The user attempts to access the checkout page without logging in.
2. The user enters invalid login credentials.
3. The user attempts to check out with an empty cart.
4. The user enters an invalid product quantity (e.g., zero or negative).
5. The user refreshes or navigates away during the checkout process.
6. The user removes all items from the cart before checkout.
7. The system displays an appropriate message when a required checkout step is missing.
8. The user attempts to proceed with checkout while the application is unavailable or unresponsive.

6) Test Design (data)

Test Design Techniques

The following test design techniques will be applied to ensure sufficient coverage and effective defect detection:

- **Equivalence Partitioning:** Used to divide input data into valid and invalid partitions, especially for login credentials and product quantity fields.

- **Boundary Value Analysis: Applied** to numeric fields such as product quantity (e.g., minimum, maximum, zero, and negative values).
- **Positive and Negative Testing: To** validate correct system behavior with valid inputs and ensure proper handling of invalid or unexpected inputs.
- **End-to-End Flow Design: Used** for designing complete user journeys from login through checkout and order confirmation.
- **Exploratory Testing: Performed** alongside structured test cases to identify usability issues and unexpected system behavior.

Test Case Coverage

Test cases will be designed to cover:

- All in-scope features are defined in the test plan.
- Happy path and negative scenarios.
- Critical business rules such as authentication requirements and cart total accuracy.

All test cases will include:

- Test Number
- Test Case Description
- Test Data
- Expected Results
- Actual Results
- Pass/Failed

7) Environment & Tools

Web Application: SauceDemo (https://www.saucedemo.com)
Browser: FireFox
Operating System: Linux
Test data: Demo users and sample products provided by the application

8) Defect Management Process

Defect Severity

Critical: Blocks core functionality such as login or checkout completion.
High: Major functionality works incorrectly with no workaround.
Medium: Functionality works with limitations or incorrect behavior.
Low: Minor UI or cosmetic issues with no functional impact.

Defect Priority

High: Must be fixed before release.
Medium: Fix recommended before release if time allows.
Low: The fix can be deferred to a later release.

Defect Workflow

1. Defects identified during test execution.
2. Defect logged with complete details and evidence.

3. Defect reviewed and prioritized.
4. Fix applied by the development team.
5. Defects retested by QA.
6. Defects are closed or reopened if not resolved.

Required Defect Fields

- Defect ID
- Summary
- Description
- Steps to Reproduce
- Expected Result
- Actual Result
- Severity
- Priority
- Status
- Attachments (screenshots)

9) Metrics & Reporting

Metrics

Test case pass/fail rate
Test coverage by feature
Number of defects by severity
Defect trends during execution
Retest and closure rate

Reporting

Daily test execution status updates
Defect status updates during testing
Final test summary report at the end of the test cycle

10) Schedule & Resources

Schedule

Test planning and preparation: 2 days
Manual test case design: 2 days
Test execution and defect reporting: 2 days
Retesting and closure: 1 day

Resources

QA Engineer: Responsible for test planning, execution, and reporting.
Tools: Web browser, test management documents, defect tracking sheets.

Effort & Buffer

Total estimated effort: 7 working days
Buffer: 1–2 days for rework and unexpected issues

11) Risks & Mitigation

Risk	Mitigation	Contingency
Test environment unavailable	Schedule testing during off-peak hours	Resume testing when system is available
Limited testing time	Focus on critical user flows	Defer low-priority tests
Unstable application behavior	Document issues clearly	Retest after stabilization

12) Entry/Exit Criteria & Sign-off

Entry Criteria

The application is deployed and accessible.
Test cases are reviewed and approved.
The test environment is ready.
Test data is available.

Exit Criteria

All planned test cases are executed.
Critical and high-severity defects are documented.
Test results and evidence are completed.
The test summary report is finalized.

Sign-off

QA approval confirms that testing activities are complete, and results are documented.
The release decision is based on test results and defect status.