

Test Plan Document

| **Version** | **Date** | **Author** | **Description** |
| --- | --- | --- | --- |
| 1.0 | 2025-05-08 | Rupa Mandal | Initial draft created |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1 Introduction 3](#_Toc197637827)

[1.1 Purpose 3](#_Toc197637828)

[1.2 Project Overview 3](#_Toc197637829)

[2 Scope 3](#_Toc197637830)

[2.1 In-Scope 3](#_Toc197637831)

[2.2 Out-of-Scope 4](#_Toc197637832)

[3 Testing Strategy 4](#_Toc197637833)

[3.1 Test Objectives 4](#_Toc197637834)

[3.2 Test Assumptions 5](#_Toc197637835)

[3.3 Testing Approach 5](#_Toc197637836)

[3.4 Test Deliverables 6](#_Toc197637837)

[3.5 Tools and Framework 6](#_Toc197637838)

[3.6 Test Schedule 8](#_Toc197637839)

[4 Execution Strategy 8](#_Toc197637840)

[4.1 Entry Criteria 8](#_Toc197637841)

[4.2 Exit criteria 9](#_Toc197637842)

[4.3 Defect Management 9](#_Toc197637843)

[4.3.1 Defect Tracking Tool 9](#_Toc197637844)

[4.3.2 Defect Lifecycle 9](#_Toc197637845)

[4.3.3 Severity Levels 10](#_Toc197637846)

[4.3.4 Reporting & Metrics 10](#_Toc197637847)

[5 Environmental Requirements 11](#_Toc197637848)

[5.1 Test Environments 11](#_Toc197637849)

[6 Risks and Mitigations 11](#_Toc197637850)

[7 Sign-off and Approval 12](#_Toc197637851)

# Introduction

## Purpose

The purpose of this test plan is to outline the strategy and approach for functional testing of the demo website (<https://www.saucedemo.com/v1/> ). The primary objective is to ensure that all core features of the website operate as intended and deliver seamless user experience. This includes verifying that key functionalities such as navigation, form submissions, user interactions, and content rendering behave consistently across supported browsers and devices.

By leveraging SauceLabs for automated and cross-browser testing, this plan aims to identify any functional defects early in the development lifecycle, ensuring a stable and reliable product for end users. The test results will serve as a benchmark for assessing the website's readiness for broader deployment or demonstration purposes.

## Project Overview

The project under test is a demo online shopping website designed to simulate a typical e-commerce experience. The website enables users to browse a catalogue of products, view product details, add selected items to a shopping cart, and proceed through a checkout process to complete a purchase.

This demo site serves as a reference implementation for showcasing key functionalities of an online store, including responsive UI behaviour, real-time cart updates, and a streamlined checkout flow. It is intended for demonstration, testing, and training purposes, rather than for real-world transactions.

The goal of this project is to ensure that all core shopping functionalities are fully operational, intuitive, and meet expected usability and reliability standards across various browsers and platforms.

# Scope

## In-Scope

The scope of this testing effort is limited to validating the **functional aspects** of the demo online shopping website. The focus will be on ensuring that all key features behave as expected and support a positive user experience. Specifically, the following areas are in scope:

* ***User Login Functionality****: User able to login successfully with valid credentials.*
* ***Homepage functionality:*** *Loading of product listings, banners, and basic navigation.*
* ***Product browsing:*** *Ability to view product and access individual product detail pages.*
* ***Shopping cart functionality:*** *Adding, removing, and updating product quantities in the shopping cart.*
* ***Checkout process:*** *Functionality covering entering user details, proceeding through the checkout steps, and confirming orders.*
* ***Navigation and links:*** *Ensuring that menus, buttons, and internal links direct users correctly.*
* ***Input validations:*** *Basic input validation for required fields during checkout.*
* ***Cross-browser compatibility testing:*** *Functional testing on major browsers (e.g., Chrome, Firefox, Safari, Edge) using SauceLabs.*
* ***Basic error handling:*** *Display of appropriate messages for invalid actions or missing inputs.*
* ***performance testing:*** *Performance testing of backend services.*

*Specify interfaces, process, workflow, vendor integration.*

## Out-of-Scope

The following areas are considered out of scope for this testing effort and will not be covered as part of this test plan:

* **Payment gateway integration:** No actual payment processing or third-party payment validation will be tested, as this is a demo site.
* **Localization and internationalization:** Testing of multilingual support, currency conversions, or region-specific formats is not included.
* **User account management:** Registration and account-specific features are excluded if not part of the demo flow
* **Third-party integrations:** Any APIs or services integrated purely for demonstration and not functionally supported will be excluded from validation.
* **Security testing:** Vulnerability scanning, penetration testing, and other security assessments are outside the scope.

# Testing Strategy

## Test Objectives

The main objective of this testing effort is to verify that the core functionalities of the demo online shopping website work as intended and support a smooth and intuitive user experience. The specific test objectives are:

* To validate the end-to-end functionality of the product browsing and shopping flow.
* To ensure users can successfully add products to the shopping cart and proceed through the checkout process.
* To confirm that the website responds correctly to user actions and provides appropriate feedback (e.g., validation messages, confirmations).
* To verify that navigation elements, buttons, and links work consistently across different pages and browsers.
* To identify and report any functional defects that may impact the usability or reliability of the website.
* To test the website’s behaviour across multiple browser and platform combinations using SauceLabs.
* To ensure that the basic form validations and error messages function as expected during checkout.

These objectives will help determine whether the website is functionally complete and ready for demonstration or further development.

## Test Assumptions

The following assumptions have been made during the planning of this testing effort:

* The demo website is stable and accessible throughout the test cycle.
* Core functionalities such as product browsing, cart management, and checkout are fully implemented and ready for testing.
* No real payment transactions will be processed; the checkout process will simulate order placement without external payment gateway integration.
* Required test data (e.g., product listings, user inputs) is either preloaded or can be created during testing.
* The test environment (including SauceLabs setup and browser configurations) will be available and properly configured before test execution begins.
* Functional specifications or requirements for the demo site are available or the behaviour is based on standard e-commerce flows.
* There will be minimal changes to the application during the test cycle; if changes occur, they will be communicated in advance.
* Defects found will be logged and tracked but may not all be fixed due to the demo nature of the application.

These assumptions help clarify the testing context and reduce ambiguity around expectations and deliverables.

## Testing Approach

The testing approach defines how functional testing will be conducted to meet the objectives of this project. The approach includes:

* **Functional Testing** will be performed for all core functionalities such as browsing products, adding items to the cart, and checking out. Test cases will be executed step-by-step, verifying expected outcomes and UI behavior.
* **Error Handling Testing** will be conducted to ensure appropriate messages are displayed for invalid inputs or failed actions.
* **UI Testing** needs to be performed to checks that the user interface elements are aligned, responsive, and interact as intended.
* **Boundary Testing** need to be performed to ensure values are not going beyond boundaries.
* **Cross-Browser Testing** will be conducted using **SauceLabs**, which allows automated/manual tests across multiple browser and OS combinations to ensure consistent functionality.
* **Exploratory Testing** will be used in parallel to structured testing to uncover unexpected defects by simulating real user behaviour.
* **Regression Testing** will be executed after any bug fixes or minor updates to ensure existing functionalities remain unaffected.
* **Test Case Design** will be based on user flows and standard e-commerce behaviours, even in the absence of detailed functional specifications.
* **Defect Reporting and Tracking** will be managed using a defect tracking tool (e.g., JIRA), with issues logged, prioritized, and verified upon fix.

The approach aims to balance thorough functional coverage with flexibility for changes or demo-specific constraints.

## Test Deliverables

The following artifacts will be produced and shared as part of the testing process:

* **Test Plan Document** – Outlines the scope, objectives, strategy, and schedule for testing.
* **Test Cases/Test Scripts** – A documented set of test cases covering core functional areas.
* **Test Data Sets** – Input data required to execute test cases (Example login details).
* **Test Execution Report** – Daily or cycle-based execution logs showing pass/fail status.
* **Defect Reports** – Logged issues with detailed descriptions, steps to reproduce, and severity.
* **Test Summary Report** – Final report summarizing testing outcomes, coverage, defects, and recommendations.
* **Sign-off Document** – Approval from stakeholders confirming successful test completion.

## Tools and Framework

The automation framework for this project will be built using Python and integrated with SauceLabs for cross-browser testing. The goal is to enable efficient, scalable, and maintainable test automation for core website functionality.

|  |  |  |
| --- | --- | --- |
| Component | Tools/Technology | Purpose |
| **Programming Language** | Python | Main language for scripting test automation. |
| **Automation Framework** | Selenium WebDriver (Python) | Automates user interactions with the browser for functional testing. |
| **Test Runner** | pytest | Framework for structuring, executing, and grouping test cases. |
| **Design Pattern** | Page Object Model (POM) | Organizes page interactions into reusable Python classes/modules. |
| **Cross-Browser Testing** | SauceLabs | Executes tests across various browsers/OS configurations in the cloud. |
| **Reporting** | pytest-html, Allure | Generates detailed test reports, including logs and screenshots. |
| **Source Code Management** | Git + GitHub | Version control for test code and CI workflows; pull request integration. |
| **CI/CD Integration** | GitHub Actions | Automates test runs on commits/pull requests; integrates with SauceLabs. |
| **Defect Tracking** | Jira | Logs, tracks, and manages bugs discovered during test execution. |
| **Logging** | Python logging module | Captures test run logs for debugging and audit purposes. |
| **Test Data Management** | JSON/CSV files | External data-driven input for tests such as product selection or forms. |

**Key Features of the Framework:**

* **Parallel Testing:** Enabled using pytest-xdist for faster execution.
* **Cross-Browser Configuration:** Defined in external config files (e.g., .env, config.yaml) to dynamically switch browsers/platforms via SauceLabs.
* **Defect Lifecycle:** Failed test results are linked to Jira issues where applicable, allowing traceability from test to defect.

## Test Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| **Activity** | **Start Date** | **End Date** | **Responsible** |
| Test Planning & Review | 2025-05-08 | 2025-05-09 | Test Manager |
| Test Case Design & Preparation | 2025-05-10 | 2025-05-15 | Sr. Test Engineer |
| Environment Setup (SauceLabs etc.) | 2025-05-10 | 2025-05-22 | Sr. Test Engineer / Admin |
| Test Execution | 2025-05-13 | 2025-05-29 | Test Engineer |
| Defect Reporting & Re-testing | 2025-05-13 | 2025-06-05 | Test Engineer / Dev Team |
| Test Closure & Summary Report | 2025-05-18 | 2025-06-12 | Sr. Test Engineer/ Test Manager |

# Execution Strategy

## Entry Criteria

* *The entry criteria refer to the desirable conditions in order to start test execution*
* *Entry criteria are flexible benchmarks. If they are not met, the test team will assess the risk, identify mitigation actions and provide a recommendation.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Entry Criteria** | **Test Team** | **Technical Team** | **Notes** |
| *The demo website is stable and accessible environment* | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png |  |  |
| *Core functionality (browsing, cart, and checkout) is implemented and ready for testing* |  |  |  |
| *Required test data (product listings, user inputs) is available.* |  |  |  |
| *Access to SauceLabs and any test management (Visual Studio with Selenium) and defect tracking tool (Jira) is granted.* |  |  |  |
| *Test cases are documented and reviewed.* |  |  |  |
| *No critical or blocking defects are outstanding from the development team.* |  |  |  |

## Exit criteria

* *The exit criteria are the desirable conditions that need to be met in order proceed with the implementation.*
* *Exit criteria are flexible benchmarks. If they are not met, the test team will assess the risk, identify mitigation actions and provide a recommendation.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Exit Criteria** | **Test Team** | **Technical Team** | **Notes** |
| *100% Test Scripts executed* | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png |  |  |
| *90% pass rate of Test Scripts* |  |  |  |
| *No open Critical and High severity defects* |  |  |  |
| *All remaining defects are either cancelled or documented as Change Requests for a future release* |  |  |  |
| *All expected and actual results are captured and documented with the test script* |  |  |  |
| *Test summary report is created and reviewed* |  |  |  |
| *All defects logged in Defect Tracker/Spreadsheet* |  |  |  |
| *Test summary report is created and reviewed* |  |  |  |

## Defect Management

Defect management outlines the process of identifying, logging, tracking, and resolving bugs found during testing. The goal is to ensure that all issues are properly addressed and resolved based on their severity and impact.

### Defect Tracking Tool

* **Tool Used:** Jira
* **Integration:** Optionally integrated with GitHub and test reporting tools for traceability between defects, test cases, and automation failures.

### Defect Lifecycle

1. **Defect Identification:** Testers identify and reproduce the defect.
2. **Defect Logging:** Defect is logged in Jira with the following details:
   * Title/summary
   * Environment details
   * Steps to reproduce
   * Expected vs. actual results
   * Screenshots/logs (if applicable)
   * Severity and priority
   * Associated test case ID (if applicable)
3. **Triage & Assignment:** QA lead or product owner reviews the defect, assigns priority, and routes it to the responsible developer.
4. **Fixing & Unit Testing:** Developer resolves the defect and performs internal validation.
5. **Retesting:** Tester verifies the fix in the target environment.
6. **Closure:** If retesting passes, the defect is closed; otherwise, it is reopened with comments.

### Severity Levels

|  |  |
| --- | --- |
| **Severity** | **Impact** |
| *1 (Critical)* | * *Functionality is blocked and no testing can proceed* * *Application/program/feature is unusable in the current state* |
| *2 (High)* | * *Functionality is not usable and there is no workaround, but testing can proceed* |
| *3 (Medium)* | * *Functionality issues but there is workaround for achieving the desired functionality* |
| *4 (Low)* | * *Unclear error message or cosmetic error which has minimum impact on product use.* |

### Reporting & Metrics

* Weekly defect reports including:
  + Open/closed/reopened bugs
  + Defects by severity
  + Defect resolution time
* Defect trends will be analysed to assess product quality and readiness.

# Environmental Requirements

## Test Environments

The test environment refers to the setup of hardware, software, tools, and network configurations required to execute the test cases for the demo online shopping website.

**7.1 Environment Components**

* **Application Under Test (AUT):** Demo online shopping website hosted on a publicly accessible web server.
* **Test Tools:**
  + **SauceLabs:** For cross-browser and cross-platform functional testing.
  + **Test Management Tool:** (e.g., TestRail, Zephyr – if applicable) for organizing and tracking test cases and results.
  + **Bug Tracking Tool:** (e.g., JIRA) for logging and managing defects.

**7.2 Supported Browsers & Platforms (via SauceLabs)**

* Google Chrome (latest stable version)
* Mozilla Firefox (latest stable version)
* Microsoft Edge (latest stable version)
* Safari (latest stable version on macOS)
* Optional: Mobile browsers (iOS Safari, Android Chrome) if responsive testing is desired

**7.3 Test Data**

* Sample product catalog available in the test database.
* Test user details for checkout flow (no real user accounts required).
* Dummy input data for form validation testing.

**7.4 Network & Access**

* Stable internet connection to access the demo site and SauceLabs platform.
* Proper credentials and permissions to access SauceLabs and any test management tools.

# Risks and Mitigations

|  |  |  |
| --- | --- | --- |
| Potential Risk | Impact | Mitigation Strategy |
| **Demo site is not stable or accessible** | Testing delays: test results may be invalid | Confirm environment readiness before test execution; escalate downtime promptly. |
| **Incomplete or unclear functional requirements** | Test coverage gaps; misaligned expectations | Use standard e-commerce behaviour as reference; clarify assumptions with stakeholders. |
| **Changes in application during testing phase** | Rework on test cases and re-testing required | Freeze code during test cycle or establish change control procedures. |
| **Defects in core functionality block further testing** | Testing halted or delayed | Prioritize defect triaging and focus on testing unaffected components in parallel. |
| **Delays in environment or tool setup (e.g., SauceLabs access)** | Loss of testing time | Begin environment setup early; have a backup local test configuration if possible. |
| **Cross-browser inconsistencies not detected early** | Incomplete functional coverage | Schedule cross-browser testing early and repeat after key bug fixes. |
| **Unavailability of key personnel (e.g., Test Engineer)** | Missed deadlines or incomplete testing | Ensure backup testers or cross-trained team members are identified. |

# Sign-off and Approval

This section confirms that all stakeholders agree with the contents of the test plan and authorize the execution of the testing activities as outlined.

**Sign-off Criteria**

* All sections of the test plan are reviewed and agreed upon by relevant stakeholders.
* Testing scope, objectives, timelines, and responsibilities are clearly understood.
* Required resources, environments, and tools are confirmed to be available.
* Risks and mitigations have been acknowledged.

**Approval Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Role** | **Approval Status** | **Date** | **Signature** |
| Rupa Mandal | Senior Test Engineer | ✅ Approved | 10/05/2025 | Rupa Mandal |
| [Test Manager Name] | QA/Test Manager |  |  |  |
| [Developer Lead Name] | Development Lead |  |  |  |
| [Project Owner Name] | Product Owner/Stakeholder |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |