**Test Plan for Swag Labs Website**

**Created by:** Ankit Kumar
**Version:** 1.0
**Date:** 08-Oct-2025

---

**1. Objective**

This Test Plan outlines the comprehensive strategy for testing the **Swag Labs** web application — an e-commerce simulation platform designed to evaluate web automation skills, user interface functionality, and overall user experience.

The primary objective is to validate that all functionalities, from user authentication to checkout flow, perform reliably and consistently across supported environments. This plan ensures the system meets both functional and non-functional requirements for the target audience — **QA engineers, automation testers, and general users simulating online shopping scenarios.**

---

**2. Scope**

**In-Scope**

The following areas of the Swag Labs application are included in testing:

- **Authentication:** Validation of login with multiple user types (standard, locked_out, problem, glitch, etc.)
- **Product Inventory:** Ensuring product list, sorting options, and images load correctly
- **Cart Management:** Adding, removing, and updating items in the cart
- **Checkout Workflow:** Capturing user details, reviewing orders, and confirming purchases
- **Logout & Navigation:** Testing proper session termination and navigational consistency
- **Error Handling:** Ensuring meaningful and user-friendly error messages

**Testing Types**

- **Manual Testing** – Exploratory, functional, and UI validation
- **Automated Testing** – Regression and smoke tests using Selenium + Python
- **Performance Testing** – Load time and responsiveness
- **Accessibility Testing** – WCAG compliance checks for usability

**Evaluation Criteria**

- Functional coverage completion ≥ 95%
- Critical defect leakage ≤ 2%
- Zero blockers in production release
- Average defect resolution time ≤ 24 hours

**Team Roles & Responsibilities**

| Role | Responsibility |
|---|---|
| **Test Lead** | Define strategy, manage QA team, track progress |
| **QA Testers** | Execute tests, document results, report defects |
| **Developers** | Analyze and fix reported issues |
| **Project Manager** | Oversee schedule, ensure timely deliverables |
| **Client/Stakeholder** | Review and approve QA documentation |

---

**3. Inclusions**

**Introduction**

The Swag Labs Test Plan defines the purpose, objectives, and strategy to ensure the product meets all acceptance criteria. Testing will focus on functionality, usability, and performance consistency.

**Test Objectives**

- Detect and report functional defects early in the SDLC.
- Validate system stability across browsers and devices.
- Enhance user experience by identifying and fixing UI/UX flaws.
- Establish a robust regression suite for future releases.

---

**4. Exclusions**

The following areas are outside the current testing scope:

- API and backend database testing
- Payment gateway integration (mocked in demo environment)
- Email/SMS notification flows
- Localization or multi-language testing
- Mobile app (Android/iOS) testing

---

**5. Test Environments**

| Category | Details |
|---|---|
| **Operating Systems** | Windows 10, macOS Monterey, Ubuntu 22.04 |
| **Browsers** | Chrome (latest), Firefox, Edge |

| Category | Details |
|---|---|
| **Devices** | Desktop, Laptop, Tablet |
| **Network Connectivity** | Wi-Fi (50 Mbps), Wired (LAN) |
| **Hardware Requirements** | 8 GB RAM, 2 GHz dual-core CPU |
| **Software Requirements** | Python 3.10+, Selenium WebDriver, PyTest, Browser Drivers |
| **Security Protocols** | Secure HTTPS connection, password encryption |
| **Access Permissions** | QA Team with Swag Labs login credentials |

---

**6. Defect Reporting Procedure**

**Defect Identification Criteria**

A defect will be logged for any:

- Deviation from expected functional behavior
- User experience or performance issue
- Broken element, missing validation, or visual inconsistency

**Defect Reporting Steps**

1. Reproduce the issue consistently.
2. Document with detailed steps, actual vs. expected results.
3. Attach screenshots, logs, or screen recordings.
4. Log the defect in **JIRA** under the correct module.
5. Assign severity and priority levels (Critical, Major, Minor).

**Triage and Tracking**

- **Critical:** Login failure, data corruption, crash
- **Major:** Checkout or cart malfunction
- **Minor:** UI misalignment, cosmetic issues

**Metrics to Track:**

- Total defects logged per build
- Average time to fix
- Reopen rate (%)
- Severity distribution graph

---

**7. Test Strategy**

**Step 1: Test Design**

**Test Design Techniques:**

- Equivalence Partitioning (valid/invalid credentials)
- Boundary Value Analysis (input limits)
- Use Case & Decision Table Testing (checkout flow)
- Exploratory and Error Guessing (unhandled scenarios)

**Step 2: Execution Approach**

1. **Smoke Testing** – Validate key workflows: login, cart, checkout.
2. **Functional Testing** – Execute all planned test cases.
3. **Regression Testing** – Re-run after fixes to ensure stability.
4. **Usability Testing** – Validate UI layout, navigation flow, and readability.
5. **Cross-Browser Testing** – Confirm consistency across browsers.

**Step 3: Testing Best Practices**

- **Shift Left Testing:** Begin validation early in development.
- **Context-Driven Testing:** Focus on real user workflows.
- **End-to-End Testing:** Simulate complete purchase flow.
- **Pair Testing:** Collaboration between tester and developer for rapid feedback.

---

**8. Test Schedule**

| Phase | Task | Start Date | End Date | Owner |
|---|---|---|---|---|
| Planning | Test Plan preparation | 08-Oct-2025 | 09-Oct-2025 | Test Lead |
| Design | Test case design & review | 09-Oct-2025 | 11-Oct-2025 | QA Team |
| Execution | Test execution & defect logging | 12-Oct-2025 | 16-Oct-2025 | QA Team |
| Retesting | Defect revalidation | 17-Oct-2025 | 18-Oct-2025 | QA Team |
| Closure | Final reports & summary | 19-Oct-2025 | 19-Oct-2025 | Test Lead |

---

**9. Test Deliverables**

- Approved **Test Plan** document
- **Test Scenarios** and **Test Cases** (Excel/Sheet format)

- **Defect Reports** (JIRA exports)
- **Test Summary Report** post execution
- **Automation Scripts** (for regression)

---

**10. Entry and Exit Criteria**

| Phase | Entry Criteria | Exit Criteria |
|---|---|---|
| **Requirement Analysis** | Finalized requirement document | All doubts clarified and approved |
| **Test Execution** | Approved test cases, stable build | All planned tests executed, defects logged |
| **Test Closure** | All defects closed or deferred | Final test summary approved and signed off |

---

**11. Tools**

- **JIRA:** Defect tracking and workflow management
- **Selenium + Python:** Test automation
- **PyTest:** Automation execution and reporting
- **Excel / Google Sheets:** Manual test case tracking
- **Snipping Tool / OBS:** Capturing evidence
- **MindMeister / Miro:** Test scenario visualization

---

**12. Risks and Mitigation Strategies**

| Potential Risk | Impact | Mitigation |
|---|---|---|
| Test environment unavailability | High | Maintain backup staging environment |
| Unstable build | High | Early build validation, developer sync-up |
| Limited testing time | Medium | Prioritize high-risk and business-critical scenarios |
| Team member absence | Medium | Assign backup resource and document handover clearly |
| Browser compatibility issues | Low | Perform cross-browser checks in early cycles |

---

**13. Approvals**

Documents to be approved before execution:

- Test Plan
- Test Scenarios and Test Cases

- Test Summary Report

**Approval Signatures:**

| Role | Name | Date | Signature |
|---|---|---|---|
| Test Lead | John | 08-Oct-2025 | _____ |
| QA Manager | — | — | _____ |
| Client Representative | — | — | _____ |