Lean STP – Swag Labs (Sauce Demo)

1. Document Information
Document Name: Lean STP- Swag Labs
Version: 1.0
Author: Shani Peretz
Date: 01.02.26

2. Introduction
This document describes the test approach for the Swag Labs web application.
The goal of this project is to demonstrate manual and automated testing skills, including the use of AI as a supporting tool in the testing process.
The testing focus is on critical business flows and overall user experience.

3. Project Overview
Swag Labs is a demo e-commerce web application.
The system allows users to:

- Log in
- View products
- Add products to the cart
- Complete a checkout process

The application is used for QA practice and learning purposes.
Target users are end users performing basic e-commerce actions.

4. Test Objectives
The main objectives of this testing project are:
- Verify that the main user flows work as expected
- Identify functional issues that may affect user experience
- Ensure stability of core features after changes (regression)
- Demonstrate professional QA documentation and processes
- Combine manual and automated testing for critical flows

5. Test Scope

In Scope:

-Login functionality
-Product list page
-Add to cart
-Cart page
-Checkout process
-Logout functionality
-Basic Security testing

Out of Scope:

-Performance testing
-Mobile testing
-Backend and API testing
Out of scope areas were excluded due to the project focus on functional UI testing of core user flows in a demo environment.

6. Test Types

The following test types will be performed:
Manual functional testing - core and exploratory user flows
Regression testing - verification of main features after changes
Automated UI testing - critical end-to-end user paths

7. Test Environment

Application: Swag Labs (Sauce Demo)
Environment: Web
Browsers: Chrome (main), Firefox (optional)
Operating System: Windows
Test data: predefined demo users provided by the application.

8. Test Tools

Test Management: Google Sheets
Automation Tool: Playwright 1.48.0
Programming Language: Python 3.8.0
POM (Page Object Model) implemented using Playwright and Python
AI Tools: Used for test idea generation, test case refinement, and documentation support, AI tools were occasionally used to suggest additional edge cases for checkout-related scenarios.

9. Entry and Exit Criteria

Entry Criteria:
Application is accessible
Test environment is stable
Test cases are prepared
Exit Criteria:
All planned test cases are executed
All identified critical defects are documented and reported.
Automated tests for main flows are executed successfully

10. Risks and Mitigation

Risk	Probability	Severity	Mitigation
Limited application complexity	M	L	Focus on test quality and clear documentation
Automation instability	L	M	Automate only stable and critical user flows
Limited test data	L	L	Include negative and boundary test cases
Environment dependency	M	M	Plan test execution based on environment availability
Incomplete coverage of edge cases	M	M	Use exploratory testing and AI-assisted test idea generation

11. Deliverables

Lean STP document
STD (Test Cases)
Bug Reports
Automation test scripts
STR (Test Summary Report)

12. AI Usage Declaration

AI tools were used as a supporting assistant for:
Test idea generation
Test case improvement
Documentation language refinement
All testing decisions and validations were performed manually.