

# Test Plan Document

**Project:** Swag Labs Automation Testing

**Author:** Purva Chopdekar

**Framework:** Cypress (Mocha + Chai)

**Reporting Tool:** Mochawesome

**Data Source:** JSON

## 1. Introduction

This document outlines the test strategy and scenario documentation for end-to-end automation of the Swag Labs (<https://www.saucedemo.com>) web application. The objective is to validate core functionalities such as login, product filtering, cart operations, and the checkout process.

The tests are automated using Cypress with the Mocha + Chai framework, utilizing JSON files for data-driven testing and generating test reports via Mochawesome.

Note on URL Used: The assignment originally referenced <https://www.saucedemo.com/v1/>, however, this version appears to be deprecated or inaccessible at the time of testing.

As a result, I have used the current live and functional version of the application:

<https://www.saucedemo.com>

This ensures stable test execution, valid UI elements, and a complete user journey for automation.

## 2. Environment Setup Instructions

### 2.1 Prerequisites

- Operating System: Windows/macOS/Linux
- Node.js
- npm
- Git

### 2.2 Setup Steps

# 1. Clone this repository

```
git clone https://github.com/purvaac/CypressTests.git
```

```
cd CypressTests
```

# 2. Install all dependencies

npm install

# 3. Open Cypress in interactive mode

npx cypress open

# 4. To run a specific test file, e.g., login test file

npx cypress run --spec "cypress/e2e/login.cy.js"

# 4. Run tests in CLI

npx cypress run

# 5. Run tests with Dashboard reporting

npx cypress run --record --key 4ddae88b-42f0-406f-8e45-53305813a2fe

## File Structure:

- **/cypress/e2e/**  
Contains all test scripts: **login.cy.js**: Login test cases  
**cartoperation.cy.js**: Add/remove cart items  
**checkout.cy.js**: Checkout flow  
**productfilter.cy.js**: Product sorting/filter tests
- **/cypress/fixtures/**  
**loginData.json**: Stores test data
- **/cypress/downloads/**  
Stores downloaded test files like filtered product CSVs
- **/cypress/json-reports/, /screenshots/, /videos/**  
Save screenshots, videos, and test execution reports for debugging
- **cypress/screenshots/**  
Stores automatic screenshots taken by Cypress.
- **cypress/videos/**  
Stores video recordings of each test run.

### 3. Test Scenarios

This section describes each test scenario along with the related positive and negative test cases.

#### 3.1 Scenario: Login Functionality

**Objective:** Verify login with valid and invalid credentials.

<u>Test Case ID</u>	<u>Description</u>	<u>Type</u>	<u>Input</u>	<u>Expected Outcome</u>
TC001	Login with valid credentials	Positive	standard_user / secret_sauce	Redirects to inventory page
TC002	Login with locked out user	Negative	locked_out_user / secret_sauce	Shows "locked out" error message
TC003	Login with problem user	Positive	problem_user / secret_sauce	Login successful, but UI issues may appear
TC004	Login with performance glitch user	Positive	performance_glitch_user / secret_sauce	Login successful with delayed load
TC005	Login with error user	Positive	error_user / secret_sauce	Login successful, but product images may not load

TC006	Login with visual user	Positive	visual_user / secret_sauce	Login successful, used for visual UI testing
-------	------------------------	----------	----------------------------	--

### 3.2 Scenario: Product Filtering

**Objective:** Verify product filtering from low to high price.

<u>Test Case ID</u>	<u>Description</u>	<u>Type</u>	<u>Input</u>	<u>Expected Outcome</u>
TC007	Filter by Price (Low to High)	Positive	Select "Low to High"	Products are reordered based on ascending price

### 3.3 Scenario: Cart Operations

**Objective:** Validate add to cart and cart content verification.

<b>Test Case ID</b>	<b>Description</b>	<b>Type</b>	<b>Input</b>	<b>Expected Outcome</b>
TC008	Add a product to cart	Positive	Click "Add to cart" on a product	Cart icon updates; product is added to cart

### 3.4 Scenario: Checkout Process

**Objective:** Verify that user can complete checkout flow successfully

Test Case ID	Description	Type	Input	Expected Outcome
TC009	Complete checkout with valid data	Positive	Valid user info (name, postal code, etc.)	"Thank you for your order!" confirmation shown

## 4. Test Data

The following data is used for executing the test cases:

### 4.1 Login Credentials (stored in `loginData.json`)

```
[
  { "username": "standard_user", "password": "secret_sauce" },
  { "username": "locked_out_user", "password": "secret_sauce" },
  { "username": "problem_user", "password": "secret_sauce" },
  { "username": "performance_glitch_user", "password": "secret_sauce" },
  { "username": "error_user", "password": "secret_sauce" },
  { "username": "visual_user", "password": "secret_sauce" }
]
```

### 4.2 Checkout Form

Field	Value
First Name	John
Last Name	Doe
Postal Code	12345

## 5. Automation & Reporting Stack

Component	Tool
Automation Tool	Cypress
Testing Framework	Mocha + Chai
Data Format	JSON (/fixtures)
Reporting	Mochawesome + Cypress Dashboard
Screenshots	Auto-captured on failure
Videos	Recorded per test spec

## 6. Test Execution Results

- All test cases are executed using Cypress
- Recorded run (with videos, screenshots, and results):  
[https://cloud.cypress.io/projects/1s26wv/runs/1/test-results?actions=%5B%5D&browsers=%5B%5D&groups=%5B%5D&isFlaky=%5B%5D&modificationDateRange=%7B%22startDate%22%3A%221970-01-01%22%2C%22endDate%22%3A%222038-01-19%22%7D&orderBy=EXECUTION\\_ORDER&oses=%5B%5D&specs=%5B%5D&statuses=%5B%5D&testingTypesEnum=%5B%5D](https://cloud.cypress.io/projects/1s26wv/runs/1/test-results?actions=%5B%5D&browsers=%5B%5D&groups=%5B%5D&isFlaky=%5B%5D&modificationDateRange=%7B%22startDate%22%3A%221970-01-01%22%2C%22endDate%22%3A%222038-01-19%22%7D&orderBy=EXECUTION_ORDER&oses=%5B%5D&specs=%5B%5D&statuses=%5B%5D&testingTypesEnum=%5B%5D)