[Test Plan] Saucedemo.com Website

|  |  |
| --- | --- |
| Document Status | [DRAFT] |
| Test Plan Creator | Anak Agung Sagung Naya Aiswarya Putri Arnaya |
| Test Plan Reviewer | * Reviewer\_1 * Reviewer\_2 * Reviewer\_3 |

# Introduction

## Overview Feature to be tested

The Test Plan has been created to communicate the test approach to team members. It includes the objectives, scope, schedule, risks and approach. This document will clearly identify what the test deliverables will be and what is deemed in and out of scope.

This Test Plan is made to ensure 3 features on the saucedemo.com website such as Login, Sorting, and Logout function properly and meet the criteria according to user needs.

## Goals and Objective

Test Case Tamer is a web-based Test Management tool used to create and store tests as well as the results of running those tests. This tool is a new product written with Ruby on Rails using a ‘mysql’ database. The test team is responsible for testing the product and ensuring it meets their needs. The test team is both the customer and the tester in this project.

Phase 1 of the project will deliver TCT (Test Case Tamer) with functionality to create and store manual tests. This will allow the test team to start transferring tests over to the new system. Must have functionality is considered more important than the delivery date in this project.

The saucedemo.com website is a website that simulates a simple online store (e-commerce) that has features such as a login page, product list dashboard, shopping cart, and checkout. The stages of testing the features carried out by manual testing (Manual Test) are as follows:

1. Testing the Login page features aims to ensure users do not experience problems when performing the login with the right credentials, as well as receiving appropriate error messages with the incorrect ones, under various conditions. The test scenario is to use a valid username, invalid, stable connection and unstable connection.
2. Testing the Sort features aims to make sure that users do not experience problems when performing the sort features and meets user expectations. The test scenario is to use Sort price low to High, Sort price high to low, Sort product name A to Z, Sort product name Z to A
3. Testing the Logout features aims to ensure that the functionality works as expected, allowing users to log out successfully, and preventing invalid access after logout.

# Test Item

List of test items (feature / product)

|  |  |  |
| --- | --- | --- |
| No | Feature | Test Item |
| #1 | Login | 1. Login with valid credentials 2. Login with empty username 3. Login with empty password 4. Login with user doesn’t exist 5. Login with locked\_out\_user 6. Login with perfomance\_glitch\_user 7. Login without filling any form 8. Login with problem\_user 9. Login with error\_user 10. Login with visual\_user |
| #2 | Sort | 1. Sort price High to Low 2. Sort price Low to High 3. Sort product name A to Z 4. Sort product name Z to A |
| #3 | Logout | 1. Function logout works properly |

# In Scope Testing

List of in feature to be tested

|  |  |  |
| --- | --- | --- |
| No | Feature | Requirement / Design specifications |
| #1 | Login |  |
| #2 | Sort |  |
| #3 | Logout |  |

# Out of Scope Testing

List the features of the software / product which will not be tested.

|  |  |  |
| --- | --- | --- |
| No | Feature | Reasons |
| #1 | Register | This testing phase will be focus to test login feature. |
| #2 | Checkout | This testing phase will be focus to test sort features. |
| #3 | Cart | This testing phase will be focus to test sort features. |
| #4 | Add To Cart | This testing phase will be focus to test sort features. |
| #5 | Remove | This testing phase will be focus to test sort features. |
| #6 | Continue Shopping | This testing phase will be focus to test sort features. |
| #7 | Checkout Info | This testing phase will be focus to test sort features. |
| #8 | Finish Order | This testing phase will be focus to test logout feature. |

# Testing Approach

Summary of testing approach that will be implemented during testing phase

The project is using an agile approach, with weekly iterations. At the end of each week the requirements identified for that iteration will be delivered to the team and will be tested.

Exploratory testing will play a large part of the testing as the team has never used this type of tool and will be learning as they go. Tests for planned functionality will be created and added to TCT as we get iterations of the product.

The testing process uses Exploratory testing type because the tester does not know from the beginning the process of making the saucedemo.com website. So that the tester explores the website to test whether the website is suitable and can be used by users without any problems.

# Test Environment

Information about testing environment that will be used during testing phase e.g URL, and

* Website: <https://www.saucedemo.com/>

# Test Deliverables

1. Test Plan (this document itself): [link]
2. Test Case: [link]
3. Test Script (if any): [link]
4. Test data (if any): [link]
5. Defect Report: [link]
6. Test Report: [link]

# Testing Timeline

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Task | Estimation Time | | Actual Time | |
| Start Date | Finish Date | Start Date | Finish Date |
| #1 | Develop Test Planning | 3 Jun 2025 | 10 Jun 2025 | 6 Jun 2025 | 8 Jun 2025 |
| #2 | Develop Test Case | 3 Jun 2025 | 10 Jun 2025 | 6 Jun 2025 | 8 Jun 2025 |
| #3 | Test Case Review | 3 Jun 2025 | 10 Jun 2025 | 6 Jun 2025 | 8 Jun 2025 |

# References

List of related documents e.g PRD, Figma design, API Contract, Copy document etc