

# TEST PLAN FOR

## <<DEMOSTORE.SUPERSQA.COM>>

### *ChangeLog*

Version	Change Date	By	Description
version number	Date of Change	Name of person who made changes	Description of the changes made

<b>1</b>	<b>INTRODUCTION .....</b>	<b>2</b>
1.1	SCOPE .....	2
1.1.1	<i>In Scope</i> .....	2
1.1.2	<i>Out of Scope</i> .....	2
1.2	QUALITY OBJECTIVE .....	2
1.3	ROLES AND RESPONSIBILITIES .....	2
<b>2</b>	<b>TEST METHODOLOGY .....</b>	<b>3</b>
2.1	OVERVIEW .....	3
2.2	TEST LEVELS.....	3
2.3	BUG TRIAGE .....	3
2.4	SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS .....	3
2.5	TEST COMPLETENESS.....	3
<b>3</b>	<b>TEST DELIVERABLES.....</b>	<b>4</b>
<b>4</b>	<b>RESOURCE &amp; ENVIRONMENT NEEDS.....</b>	<b>4</b>
4.1	TESTING TOOLS .....	4
4.2	TEST ENVIRONMENT.....	4
<b>5</b>	<b>TERMS/ACRONYMS .....</b>	<b>5</b>

# 1 Introduction

This test plan outlines the strategies, process, workflow and methodologies for the QA automation project using Python Selenium. The objective is to ensure the e-commerce website (demostore.supersqa.com) meets the specified functional and non-functional requirements and delivers a high-quality user experience.

## 1.1 Scope

---

### 1.1.1 In Scope

- Registration and login functionality
- Product browsing and details
- Shopping cart and checkout process
- Search functionality
- User account management
- Quantity manipulation on the product page
- Lost password functionality

### 1.1.2 Out of Scope

- Payment gateway integration (considered a dummy process in the demo site)
- Advanced security testing (e.g., penetration testing) beyond basic vulnerability checks

## 1.2 Quality Objective

---

Ensure the automation suite achieves at least 95% test coverage and identifies critical defects in the specified functionalities before deployment.

## 1.3 Roles and Responsibilities

---

- QA Analyst: Develop test scenarios, automate test cases, execute tests and report defects
- Test Manager: Oversee the testing process, track progress and ensure adherence to the test plan
- Configuration Manager: Manage test environment and version control
- Developers: Address defects and issues identified during testing
- Installation Team: Facilitate deployment and ensure compatibility with the test environment

## 2 Test Methodology

### 2.1 Overview

---

Agile methodology will be adopted for this project, allowing for iterative testing, quick feedback and continuous improvement.

### 2.2 Test Levels

---

- Unit Testing: Unit tests for code components
- End-to-end Testing: Automation testing covering the entire application flow.

### 2.3 Bug Triage

---

- Triage will categorize defects based on severity (e.g., critical, major, minor)
- Defect resolution and priority will be determined by the development team in collaboration with QA.
- Defect reports will be tracked us Jira software

### 2.4 Suspension Criteria and Resumption Requirements

---

- Testing may be suspended if critical issues hinder the testing process.
- Resumption will occur after critical defects are resolved, and the test environment is stable.

### 2.5 Test Completeness

---

Testing will be considered complete when:

- All critical functionalities are thoroughly tested.
- Automation achieves 95% test coverage.
- All open defects are addressed or marked for future release

## 3 Test Deliverables

- Test scenarios and test cases documents
- Automated test scripts
- Test execution reports
- Defect reports with severity and priority

## 4 Resource & Environment Needs

### 4.1 Testing Tools

---

- Python: The test automation scripts will be implemented using Python as the primary programming language. Python offers simplicity, readability, and a rich set of libraries suitable for web automation tasks.
- Selenium WebDriver: Selenium WebDriver will be used for automating interactions with the web browsers. It provides a powerful API for controlling browsers and simulating user actions.
- PyCharm IDE: PyCharm IDE will serve as the integrated development environment for writing, managing, and debugging the automation code. Its features enhance productivity and code maintainability.
- Git: Git will be used as the version control system to manage the test code. It allows team members to collaborate efficiently and track changes across the project.
- Jira (or similar test management tool): Jira will be used for defect tracking and test case management. It helps in organizing test cases, assigning tasks, and monitoring testing progress.

### 4.2 Test Environment

---

- Operating System: macOS Ventura will be used as the primary operating system for test execution. The choice of macOS ensures compatibility with the target platform and enables us to leverage macOS-specific features.
- Browsers: The tests will be executed on the latest versions of Chrome, Safari and Firefox, both optimized for macOS. Cross-browser testing will be performed to ensure consistent behavior across different browsers.
- Internet Connectivity: A stable internet connection is required to access the demostore.supersqa.com website, which is the application under test.

- **Dependency Management:** Python's virtual environment will be used to isolate the project's dependencies and manage the necessary Python packages. This approach ensures a clean and consistent environment for running the test scripts.
- **Test Data:** Test data required for test scenarios will be generated or obtained as needed. It will be handled securely and professionally, ensuring data privacy and compliance.
- **Other Software:** Additional tools and libraries essential for the project will be carefully selected to ensure compatibility and optimal performance on macOS.

By configuring the test environment with the specified tools and on macOS, we aim to achieve a seamless and efficient QA automation process for the demo e-commerce website, resulting in reliable and high-quality test results.

## 5 Terms/Acronyms

Make a mention of any terms or acronyms used in the project

TERM/ACRONYM	DEFINITION
AUT	Application Under Test
IDE	Integrated Development Environment
QA	Quality Assurance