Swag Labs Test Plan (e-commerce)

**Index**

1. [**Introduction**](#introduction)
   * 1.1 Purpose
   * 1.2 Scope
   * 1.3 Objectives
   * 1.4 Glossary
2. [**Test Objectives**](#test-objectives)
   * 2.1 Functional Testing
   * 2.2 Performance Testing
   * 2.3 Security Testing
   * 2.4 Regression Testing
3. [**Test Approach**](#test-approach)
   * 3.1 Testing Levels
   * 3.2 Testing Types
   * 3.3 Testing Environment
   * 3.4 Test Tools
4. [**Test Deliverables**](#test-deliverables)
   * 4.1 Documentation
   * 4.2 Test Artifacts
   * 4.3 Reports
5. [**Testing Resources**](#testing-resources)
   * 5.1 Personnel
   * 5.2 Training
   * 5.3 Equipment
6. [**Test Schedule**](#test-schedule)
   * 6.1 Milestones
   * 6.2 Dependencies
7. [**Risks and Mitigation**](#risks-and-mitigation)
   * 7.1 Risks
   * 7.2 Mitigation Strategies
8. [**Dependencies**](#dependencies)
   * 8.1 Development Dependencies
   * 8.2 External Dependencies
9. [**Approval**](#approval)
   * 9.1 Sign-off

**1. Introduction**

**1.1 Purpose**

This QA Test Plan outlines the strategy, scope, objectives, and processes for testing the Swag Labs e-commerce website application. The testing process aims to ensure that the website functions correctly, provides a seamless shopping experience, and meets business and technical requirements. This plan focuses on verifying the website's functionality, performance, security, and usability to deliver a reliable and high-quality product to end-users.

**The primary objectives are:**

* Ensure Functional Integrity: Validate that all website features, such as user registration, product browsing, cart management, checkout process, and order tracking, operate according to specified requirements.
* Guarantee Performance: Assess the website's performance under various conditions, including normal and peak loads, to ensure fast and reliable user interactions.
* Verify Security: Identify and mitigate security vulnerabilities to protect user data, transaction integrity, and prevent unauthorized access.
* Enhance Usability: Ensure that the website is user-friendly, accessible, and provides a positive experience for users across different devices and browsers.
* Maintain Compatibility: Validate compatibility across different browsers, devices, and operating systems to ensure a consistent user experience.
* Support Regression Testing: Ensure that new updates or changes do not introduce defects into existing functionalities through comprehensive regression testing.

**1.2 Scope**

The scope of this QA Test Plan encompasses the comprehensive testing of the e-commerce website to ensure it meets all functional, performance, security, and usability requirements. The plan outlines the areas to be tested, the extent of testing activities, and any limitations or exclusions.

**In-Scope:**

1. Functional Testing:

* User Registration and Authentication: Verify functionalities related to user sign-up, login, password recovery, and account management.
* Product Browsing and Search: Test product listings, filters, search functionalities, and product detail views.
* Shopping Cart Management: Validate adding, updating, and removing items from the cart, as well as applying discounts and coupons.
* Checkout Process: Ensure the correct flow of the checkout process, including address entry, payment options, and order confirmation.
* Order Management: Test functionalities related to viewing order history, tracking shipments, and managing returns and cancellations.

Content Management: Verify the management and display of dynamic content like banners, promotions, and static pages.

2. Performance Testing:

* Load Testing: Assess the website's performance under expected user load to ensure it can handle the predicted traffic.
* Stress Testing: Determine the website's breaking point by testing it beyond its expected capacity.
* Scalability Testing: Evaluate the website’s ability to scale and handle increased load efficiently.

3. Security Testing:

* Vulnerability Assessment: Identify security vulnerabilities, including SQL injection, XSS, CSRF, and others.
* Data Protection: Ensure encryption and protection of sensitive user data, including personal and payment information.
* Authentication and Authorization: Validate that only authorized users can access restricted areas and functionalities.

4. Usability Testing:

* User Experience (UX): Assess the ease of navigation, overall user interface, and responsiveness across different devices.
* Accessibility: Verify compliance with accessibility standards (e.g., WCAG) to ensure the website is usable by people with disabilities.

5. Compatibility Testing:

* Cross-Browser: Test the website's functionality and appearance on different browsers (e.g., Chrome, Firefox, Safari, Edge).
* Cross-Device: Validate the website's compatibility across various devices (e.g., desktops, tablets, smartphones).
* Cross-Platform: Ensure the website performs consistently on different operating systems (e.g., Windows, macOS, iOS, Android).

6. Regression Testing:

* Automated Regression Suite: Execute automated tests to verify that new changes do not negatively impact existing functionalities.
* Manual Regression: Perform manual testing on areas not covered by automated tests to identify any unintended issues.

7. Integration Testing:

* Payment Gateways: Verify integrations with payment processors for accurate transaction processing.
* Third-Party Services: Test integrations with third-party services like shipping providers, email services, and social media plugins.
* APIs: Validate the correct functioning of internal and external APIs used by the website.

**Out-of-Scope:**

1. Non-Web Platforms: Excludes testing on native mobile applications or other platforms not directly related to the web interface.

2. External Systems Integration: Testing of systems external to the website, such as supplier systems or internal enterprise systems not interfacing directly through the e-commerce site.

3. Non-Functional Business Processes: Testing of business processes not directly related to website functionalities, such as internal inventory management beyond the API level.

4. Beta Features: Testing of features not yet released to production or still under development beyond basic verification.

This scope ensures a thorough and structured approach to testing, focusing on key areas critical to delivering a robust and user-friendly e-commerce website.

**1.3 Objectives**

* Verify the application’s functionalities against the specified requirements.
* Ensure the application performs well under expected and peak loads.
* Identify and mitigate security vulnerabilities.
* Validate that changes and updates do not introduce new defects.

**1.4 Glossary**

* UAT: User Acceptance Testing
* RTM: Requirements Traceability Matrix
* CI/CD: Continuous Integration/Continuous Deployment
* QA: Quality Assurance

**2. Test Objectives**

**2.1 Functional Testing**

* Objective: Ensure each feature functions according to the requirements.
* Key Areas:

User Authentication

Product Catalog

Shopping Cart

Checkout Process

Order Management

**2.2 Performance Testing**

* Objective: Assess application performance under various conditions.
* Key Areas:

Load Testing

Stress Testing

**2.3 Security Testing**

* Objective: Identify and address security vulnerabilities.
* Key Areas:

Authentication

Data Protection

Vulnerability Scanning

**2.4 Regression Testing**

* Objective: Ensure new updates do not negatively impact existing functionality.
* Key Areas:
* Retesting previously tested features
* Automated regression test suite execution

**3. Test Approach**

**3.1 Testing Levels**

**3.1.1 Unit Testing**

* Responsibility: Developers / Tester
* Objective: Validate individual components.

**3.1.2 Integration Testing**

* Responsibility: QA Team
* Objective: Ensure combined components work together.

**3.1.3 System Testing**

* Responsibility: QA Team
* Objective: Validate the complete and integrated system.

**3.1.4 User Acceptance Testing (UAT)**

* Responsibility: Business Users/Clients
* Objective: Verify the system meets user requirements.

**3.2 Testing Types**

**3.2.1 Manual Testing**

* Objective: Execute test cases manually to identify defects.
* Tools: TestRail for managing and executing test cases.

**3.2.2 Automated Testing**

* Objective: Automate repetitive and regression test cases to improve efficiency.
* Tools: Selenium WebDriver for UI testing, JUnit/TestNG for backend components.

**3.3 Testing Environment**

* Development Environment: Initial testing by developers.
* Staging Environment: QA testing and UAT.
* Production-like Environment: Final validation before release.

**3.4 Test Tools**

* Bug Tracking: JIRA for logging and tracking defects.
* Test Case Management: TestRail for managing test cases.
* Automated Testing: Selenium WebDriver for UI automation.
* Performance Testing: Apache JMeter for load and stress testing.
* Security Testing: OWASP ZAP for security assessments.

**4. Test Deliverables**

**4.1 Documentation**

* Test Plan: This document
* Test Cases: Detailed scenarios for testing
* Bug Reports: Documentation of identified defects
* Test Summary Reports: Summaries of test results

**4.2 Test Artifacts**

* Test Scripts: Automated test scripts
* Test Data: Data used for test execution

**4.3 Reports**

* Test Execution Reports: Daily/weekly summaries of testing progress
* Defect Reports: Detailed analysis of defects
* Performance Test Reports: Results of performance tests

**5. Testing Resources**

**5.1 Personnel**

* QA Engineer and Automation Engineer: Sanjana Varma

**5.2 Training**

* Training on test automation, performance testing tools, and security testing methodologies.

**5.3 Equipment**

* Hardware: Laptops/desktops with necessary specifications.
* Software: Access to test environments, automation tools, and bug tracking systems.

**6. Test Schedule**

**6.1 Milestones**

#### 6.1 Milestones

|  |  |
| --- | --- |
| Milestone | Date |
| Test Plan |  |
| Test Case Development |  |
| Test Environment Setup |  |
| Test Execution Start |  |
| Test Execution Complete |  |
| UAT Start |  |
| UAT Complete |  |
| Final Test Report |  |

**6.2 Dependencies**

* Development Completion: All features must be implemented and unit tested.
* Environment Availability: Staging environment should be ready before testing begins.
* Build Stability: Regular updates and stable builds from the development team.

**7. Risks and Mitigation**

**7.1 Risks**

* Incomplete Requirements: Could lead to missing test cases or scenarios.
* Environment Unavailability: Delays in environment setup can affect testing schedules.
* Resource Constraints: Limited personnel may hinder the testing process.

**7.2 Mitigation Strategies**

* Requirement Review: Regular updates and reviews of requirements.
* Environment Planning: Early setup and validation of test environments.

**8. Dependencies**

**8.1 Development Dependencies**

* Stable Builds: Regular and stable builds provided by the development team.
* Feature Documentation: Detailed documentation of implemented features.

**8.2 External Dependencies**

* Third-Party Services: Availability and reliability of third-party APIs or services.

**9. Approval**

**9.1 Sign-off**

* QA Tester: Chandrashekar kahate (Chandrashekharkahate7@gmail.com)
* Developers Reviewer………………………………….

This Test Plan provides a structured approach to testing the Swag Labs application, covering all aspects required to ensure a quality product.