Test strategy

Swag Labs is a demo e-commerce website designed for testing web applications. It is mainly used for training software testers. The site includes a login system with different user roles, a shopping cart, and a checkout process. It is intentionally buggy to help testers identify issues. This makes it a useful platform for practicing manual and automated testing.

**Test Types**

To ensure comprehensive coverage and high product quality, the following test types will be executed as part of the overall testing strategy:

**Manual Testing**

Manual testing involves executing test cases manually without using automation tools. It allows testers to mimic end-user behavior and identify usability and UI issues, logical errors, and unexpected behavior that may not be caught by automated scripts. It is particularly effective during early stages, complex scenarios, or when the application is still changing rapidly.

Manual testing in this project will include the following specific types:

1. **Component Testing**

To verify the functionality and behavior of individual components or modules in isolation.

**Scope:**

* Testing each module independently to ensure it meets its design and requirement specifications.
* Checking for functional correctness, error handling, and input validation within a module.

1. **Integration Testing**

To test data flow and interaction between two or more components or modules after integration.

**Scope:**

* Ensure integrated components interact correctly and data is passed accurately between them.
* Detect interface issues, mismatches in data formats, and integration logic flaws.

1. **System Testing**

To validate the complete and fully integrated system against the specified requirements.

**Scope:**

* Covers both functional and non-functional requirements (e.g., UI/UX, error messages, field validation).

1. **Component Integration Testing**

To test the interaction between two or more closely related components.

**Scope:**

* Validate that components integrated as part of a single functional module are working together as expected.
* Detect issues that arise due to integration within a component group.

1. **System Integration Testing**

To verify the interaction between independent systems or subsystems within the overall architecture.

**Scope:**

* Test interfaces and communication between multiple systems, such as front-end, backend, databases, or third-party services.
* Identify issues related to protocols, data formats, request/response handling.

**Performance testing**

Using Jira

**Security testing**

Using JMeter

**Automation testing**

Using Postman and Selenium

**Mobile App testing**

**Test Activity for Swag Labs**

**1. Test Suite**

A test suite for Swag Labs includes various categories of tests to ensure the proper functionality of the application. The test suite consists of:

Login Tests

Product Page Tests

Cart Functionality Tests

Checkout Process Tests

Performance Tests

Security Tests

Filter & Sorting Tests

User Session Management Tests

Error Handling Tests

Mobile Responsiveness Tests

1. **Test Cases in an excel sheet**

**3. Test Plan**

**Objectives:**

Validate the login functionality, product listing, cart, and checkout process.

Ensure UI and performance stability.

Identify security vulnerabilities.

Test filtering, sorting, and session management.

Verify error handling and mobile responsiveness.

**Scope:**

Web and mobile versions of Swag Labs.

Functional, security, usability, and performance testing.

**Schedule:**

**4. Test Summary**

Total Test Cases Executed: 80

Passed: 72

Failed: 8

Defects Identified: 10

Status: In Progress

**Conclusion:**

The Swag Labs application performs well in most functional areas, but some issues require attention, particularly in checkout validation, session handling, and mobile responsiveness. Further testing and bug fixes are recommended before production deployment