Test Plan for E-Commerce Website

1. Introduction

The purpose of this document is to outline the test plan and strategy for the [E-commerce Website]. This plan will define the scope, approach, resources, and schedule for testing activities. The main objective is to ensure the website functions as expected across different browsers, devices, and user scenarios, and provides a seamless shopping experience.

2. Test Objectives

- Validate the functionality of the website (e.g., user registration, product search, cart management, payment gateway integration).

- Ensure compatibility across different browsers (Chrome, Firefox, Edge, Safari) and devices (desktop, mobile, tablet).

- Verify performance under load (handling high user traffic during events like sales).

- Ensure data integrity in the database (e.g., customer details, orders, inventory).

- Validate the security of transactions, user data, and payment information.

3. Scope of Testing

- In Scope:

  - Functional Testing

  - Integration Testing

  - Performance and Load Testing

  - Security Testing

  - Usability Testing

  - Compatibility Testing (Browsers and Devices)

  - Regression Testing

  - Payment Gateway Testing

  - Database Testing

- SEO Testing

- Out of Scope:

  - Third-party vendor systems that are not directly related to the website.

  - Development of APIs, unless directly related to website functionality.

4. Testing Approach and Strategy

4.1 Functional Testing

Functional testing will cover the core features:

  - User Registration and Login: Validate that users can register, login, and recover passwords.

  - Product Search and Filtering: Test that products can be searched, sorted, and filtered correctly.

  - Product Details: Ensure that product descriptions, images, prices, and stock availability are correctly displayed.

  - Cart Management: Verify that users can add products to the cart, update quantities, and remove items.

  - Checkout Process: Ensure smooth navigation through the checkout, including shipping and billing information.

  - Payment Gateway Integration: Validate payments are processed correctly through various methods (credit card, PayPal, etc.).

  - Order Confirmation and Tracking: Check that users receive confirmation and can track their orders.

4.2 Integration Testing

Integration testing will ensure that modules like product listing, inventory management, payment gateway, and shipping services work together without issues.

4.3 Performance and Load Testing

Performance tests will simulate peak loads (e.g., flash sales or Black Friday events). The goal is to ensure that the website remains responsive under load, and the database can handle a large

number of simultaneous transactions.

4.4 Security Testing

Security testing will validate:

  - SSL certificate installation for encrypted data transmission.

  - Secure login and password storage mechanisms.

  - Protection against SQL injection, cross-site scripting (XSS), and other vulnerabilities.

  - Payment security, including PCI-DSS compliance.

4.5 Usability Testing

Usability testing will focus on:

  - Navigation: Is it easy to navigate the site, find products, and complete transactions?

  - Design consistency: Are buttons, fonts, and images clear and professional?

  - Mobile responsiveness: Does the site work well on mobile devices, and does it support gestures like swiping?

4.6 Compatibility Testing

The website will be tested across: Browsers (Chrome, Firefox, Safari, Edge) and Devices (Desktop, Tablets, Smartphones).

4.7 Regression Testing

Whenever new features are introduced or bugs are fixed, regression testing will ensure that existing functionalities remain intact.

4.8 Database Testing

Database testing will ensure data integrity and accuracy, focusing on user details, product inventory, and order information.

4.9 SEO Testing

SEO testing will validate proper use of meta tags, alt tags, optimized page load times, and SEO-friendly URLs.

5. Test Environment

Testing Environment:

  - Web Server: Apache

  - Application Server: Node.js

  - Database: MySQL

  - Browsers: Chrome, Firefox, Safari, Edge

  - Devices: iPhone, Android phones, Tablets, Desktop

  - Tools: Selenium, JMeter, Burp Suite, BrowserStack, Jenkins

6. Roles and Responsibilities

  - Test Manager: Oversee the entire testing process, manage test execution.

  - Test Analysts: Write test cases, execute functional, integration, and regression tests.

  - Performance Test Engineers: Execute load and performance tests.

  - Security Experts: Conduct security assessments.

  - Automation Engineers: Implement and maintain automated tests using Selenium.

  - Developers: Resolve bugs, assist in performance tuning.

7. Entry and Exit Criteria

  - Entry Criteria:

    - Development is complete for the modules under test.

    - Test environment is ready and stable.

    - Test data has been prepared.

    - All test cases have been written and reviewed.

  - Exit Criteria:

    - All critical and high-severity defects have been fixed.

- 95% test case execution is complete with pass results.

- Performance benchmarks have been met.

- Security vulnerabilities have been addressed.

- Test coverage has reached an acceptable level.

8. Risks and Mitigation

- Risk: Unstable test environment could delay testing.

  - Mitigation: Collaborate with the DevOps team to ensure environment stability before testing begins.

- Risk: Tight deadlines may lead to incomplete testing.

  - Mitigation: Prioritize critical test cases and features for the first round of testing.

9. Test Schedule

- Test planning: Week 1

- Test case creation: Week 2

- Functional testing: Week 3-4

- Performance and load testing: Week 5

- Security testing: Week 6

- Regression testing: Week 7

- Final review and reporting: Week 8

10. Deliverables

- Test Plan Document

- Test Cases

- Test Scripts (Automation)

- Test Execution Reports

- Defect Reports

- Performance and Security Test Reports