Test plan

New e-commerce web site

Submitted by:

Vinuri Peiris

Contents

1. Introduction

This test plan outlines the strategy and approach for testing an e-commerce web application. The application includes features like multiple payment methods, shipping options, address management, order summary, review, order confirmation, and receipt generation. The primary objective is to ensure the seamless operation, reliability, and security of these features, ultimately delivering a positive user experience.

2. Scope

This test plan covers functional, integration, and security testing of the e-commerce site. It will focus on the following areas:

- **Payment Methods**: Credit cards, PayPal, bank transfers, etc.
- **Shipping Options**: Multiple carriers, real-time shipping cost calculations, address management.
- **Order Summary and Review**: Display of products, pricing, taxes, shipping charges.
- **Order Confirmation and Receipt Generation**: Correct display of final order details and receipt generation.

3. Quality Objective

The goal is to ensure the e-commerce platform delivers a seamless shopping experience by verifying,

- **Functionality:** Ensure that all website features work as expected and meet user requirements.
- **Performance:** The website should load quickly and respond efficiently to user interactions.
- **Usability:** The website should be easy to navigate and use, with a clear and intuitive interface.
- **Security:** The website should be protected against security vulnerabilities, such as data breaches and unauthorized access.
- **Compatibility:** The website should be compatible with different browsers, devices, and operating systems.
- **Integrity:** Data integrity between modules (payment, shipping, and orders).

4. Methodology

4.1 Overview

The test strategy involves both manual and automated testing to ensure end-to-end functionality and coverage of critical features. Test cases will be derived from user stories, functional requirements, and business rules.

4.2 Risk Assessment Methodology

The likelihood and impact of potential risks are assessed based on functionality and user interactions.

4.2.1 Likelihood

- High likelihood of issues in payment processing and order generation due to complex integrations with third-party APIs.
- Medium likelihood in shipping calculations and address management due to varying user inputs.

4.2.2 Impact

- High impact on business if payment or checkout flow fails, resulting in lost revenue and customer trust.
- Medium impact on shipping and address management issues, as they affect the user's ability to complete an order.

Risk	Likelihood	Impact
Payment processing errors	High	High
Data breaches	High	Critical
Poor performance	Medium	High
Usability issues	Medium	Medium
Compatibility problems	Low	Medium

4.3 High-Level Test Methodology

4.3.1 Dependencies

Test cases depend on stable third-party integrations (e.g., payment gateways, shipping providers). Delays in response from third parties may hinder test execution. The testing effort will depend on the availability of the website and its underlying systems.

4.3.2 Client-Side Testing

Testing the website's user interfaces focusing on validating user inputs for payment details, shipping address formats, and the proper display of order information across multiple browsers and devices.

4.3.3 Exposed Design Vulnerabilities (Design Testing)

Design testing will identify weaknesses in the UI/UX that may confuse users during payment or shipping selection. In here the website's design and architecture for potential security vulnerabilities are reviewed.

Example: unclear error messages or poor button placements could affect the user experience.

4.3.4 Exposed Implementation Vulnerabilities (Implementation Testing)

Tests will be conducted to find security flaws in payment gateways, address storage, and receipt generation. This includes testing against SQL injection, XSS, and ensuring proper encryption during transactions.

4.4 Suspension Criteria & Resumption Requirement

Testing will be suspended if (Suspension Criteria)

- Critical blocking issues such as inability to complete payment or order receipt generation are detected.
- Third-party API services are down.

Testing will resume once (Resumption Requirement)

- Blocking issues are resolved, critical defects have been addressed., and third-party services are restored.

4.5 Test Completeness

Testing will be considered complete when:

- All major functional and security test cases have been executed.
- All critical and high-severity defects have been fixed and verified.
- Performance testing meets the required benchmarks for handling user load and transaction times.

5. Test Deliverables

- **Test Plan:** A comprehensive document outlining the testing strategy, objectives, scope, and methodology.
- **Test Cases:** Detailed scenarios and steps to be followed during testing to verify specific functionalities.
- **Test Scripts:** Automated scripts (if applicable) that execute test cases repeatedly and efficiently.
- **Test Data:** Sample data used to simulate real-world scenarios and test the application's behavior.
- **Test Reports:** Documents summarizing the progress, results, and outcomes of the testing process.
- **Defect Tracking Information:** Detailed records of identified defects, including their description, severity, and resolution status.
- **Bug Reports:** Detailed descriptions of defects encountered during testing, including steps to reproduce, expected vs. actual results, and severity.
- **Automation Scripts:** Scripts used to automate repetitive test cases, improving efficiency and reducing human error.
- **Performance Testing Results**: Data collected from load and stress tests to assess the website's performance under various conditions

6. Security Checklist with Tools

PCI-DSS Compliance

- **Encryption of Payment Data:** Ensure all payment card data is encrypted in transit and at rest using strong encryption algorithms
- **Access Controls:** Restrict access to cardholder data to authorized personnel only.

Tools for Security Testing

- **OWASP ZAP:** A popular open-source web application security scanner that can identify vulnerabilities like XSS, SQL injection, and CSRF.

Data Encryption

- **Encryption in Transit:** Use HTTPS to encrypt data transmitted between the website and the user's browser.
- **Key Management:** Implement secure key management practices to protect encryption keys.

Session Management

- **Session Timeouts:** Set appropriate session timeouts to automatically log out inactive users.

-END-