**Assignment Document: Software Testing Specifications**

**Project Title**: E-commerce **Alicart**

**Objective:**

The objective of this assignment is to develop a comprehensive software testing plan for E – Commerce. This document will outline the scope, objectives, testing types, environment, test cases, execution plan, defect tracking(if any), reporting, and documentation requirements for the software testing process.

1. **Scope:**

The scope of testing for E-commerce encompasses a wide array of features, functionalities, and modules integral to the software’s operations and user interactions.

User Interface: Testing the E-commerce platform's user interface in all respects, including responsiveness to various browsers and devices, consistency in layout, and ease of navigation.

Product Catalog: Guaranteeing accurate functionality of product listings, descriptions, photos, pricing, and categorization.

Verification of the shopping cart's features, such as the ability to add or remove products, change quantities, calculate prices, and complete the checkout process, which includes order confirmation and payment transactions, is necessary.

Account Management: Testing user account-related features such as order history, registration, login, password recovery, and profile editing.

Payment Gateway Integration: Using integrated payment gateways to handle payments in a safe and accurate manner.

Order Management System: Testing order processing, inventory management, shipping integration, and status updates for admin users.

Search Functionality: Checking the precision, filters, and sorting choices of the search function.

Security features include data protection, particularly for user and transaction data, and vulnerability testing.

1. **Objectives:**

To ensure the software's reliability, functionality, performance, and security, we are employing the following techniques in our testing procedures:

Reliability: Make sure there are no unanticipated breakdowns and that the e-commerce software functions properly in a variety of settings.

Functionality: Check that every feature performs as intended and meets user expectations.

Performance: Verify that the program can quickly respond to transactions and manage the anticipated load.

Security: Verify that user data is sufficiently safeguarded and that the application is secure against common online vulnerabilities (such as SQL injection and XSS).

User Satisfaction: To improve the entire user experience, make sure the program is accessible, easy to use, and intuitive.

1. **Testing Types:**

***Unit Testing:*** This involves testing individual components or units of the software in isolation.

User Interface Components: Such as buttons, forms, and input fields.

Backend Services: Including APIs for product management, order processing, and user authentication.

Database Operations: Testing CRUD (Create, Read, Update, Delete) operations for accuracy and efficiency.

***Integration Testing:***

Focuses on testing the interaction between integrated components or systems.

Payment Gateway Integration: Ensuring seamless data flow and functionality between the E-commerce platform and payment processing services.

Order Management System: Testing the integration of inventory management, shipping services, and status updates.

Account Management: Verifying the interaction between user profile management and the database for registration, login, and password recovery.

***System Testing:***

Tests the system as a whole, ensuring it meets the specified requirements.

Entire E-commerce Platform: Including the user interface, product catalog, shopping cart, and checkout process to ensure overall functionality and performance.

***Acceptance Testing:***

Ensures the software meets user needs and requirements.

User Satisfaction: Testing the software with real users to ensure it is intuitive, easy to use, and meets their expectations.

Security Features: User-focused testing to ensure data protection measures are understood and trusted.

***Performance Testing:***

Evaluates the software’s performance under various conditions.

Load Management: Testing the application's ability to handle a significant number of users and transactions simultaneously.

Response Times: Ensuring the application responds quickly to user inputs and requests, especially during peak usage.

***Security Testing:***

Identifies vulnerabilities and ensures the protection of user and transaction data.

Vulnerability Testing: Checking for common web vulnerabilities such as SQL injection and XSS.

Data Protection: Ensuring encryption and secure storage of user data.

There are 3 ways developers use to prevent SQL Injection attacks.

Prepared Statements (with Parameterized Queries), Input Validation, and Stored Procedures: These ways are considered to protest this website from SQL Injection and XSS

To stop Cross-Site Scripting (XSS) attacks one needs to :

***Encode Output:*** To guarantee that special characters are handled as data rather than code when they are shown on web pages, remove them from user inputs.

***Employ Content Security Policy (CSP):*** CSP headers are used to restrict the resources that a browser is permitted to load, hence lowering the possibility that harmful content may be executed.

***Validate and Sanitize Input:*** Make sure that only intended and secure data is processed by carefully reviewing and cleaning user input.

By taking these precautions, dangerous scripts injected by adversaries are prevented from running in users' browsers.

***Usability Testing:***

Focuses on the user's interaction with the software.

User Interface and Navigation: Testing the ease of use, accessibility, and intuitiveness of the platform's interface.

Search Functionality: Ensuring the search feature is user-friendly and delivers accurate results.

Additionally, A/B testing is a potent technique for contrasting two iterations of a webpage or app to see which one works better and should be used on the website. To test version "A" against version "B," one essentially divides the audience and measures their interaction to determine which version is more successful in reaching a predetermined goal, such as more engagement or increased conversions. Using user preferences and behavior as a basis for decision-making, this data-driven method assists.

***Regression Testing:***

Conducted after changes to the software to ensure that new code does not adversely affect existing functionality.

After Updates/Changes: Especially in areas that have been recently modified or are critical to the application’s operation.

1. **Testing Environment:**

In setting up the testing environment for an e-commerce platform, a range of dependencies crucial for mirroring production settings was considered. Here is the approach:

***Hardware:*** A diverse selection of devices, including desktops, tablets, and smartphones, has been gathered to ensure the website delivers a seamless and responsive experience across all user endpoints.

***Software:*** Django version 4.2.2 is to be installed on the test server. This consistency between the test and production environments aims to eliminate any surprises upon going live. The web server configuration utilizes software like Gunicorn (version 21.2.0), with Nginx serving as a reverse proxy to efficiently manage client requests. PostgreSQL has been chosen for its robustness and compatibility with Django's ORM.

***Network Configurations:*** A network setup has been established to simulate various user experiences through both Wi-Fi and different SIM data, to understand platform behavior under varying conditions. This is crucial for preempting any user experience issues arising from network variability.

***Test Data and Tools:*** Automated testing frameworks such as Selenium are being used to interact with the site as a real user would. Tools like Apache JMeter are employed to stress-test the platform, assessing its resilience under heavy traffic.

***Security Testing:*** Security is prioritized, with tools like OWASP ZAP implemented to scan for vulnerabilities, ensuring the security of user data.

***Other Dependencies:*** The platform's comprehensive nature necessitates support for functions ranging from content management with django-ckeditor to payment processing with Stripe and PayPal-Payouts-SDK. Each package is tested for compatibility and performance.

1. **Test Cases:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario ID** | | UserRegistration-1 | | | **Test Case ID** | | | UserRegistration-1 | | |
| **Test Case Description** | | Successful user registration | | | **Test Priority** | | | High | | |
| **Pre-Requisite** | | No pre-existing account with the test email | | | **Post-Requisite** | | | Confirmation email sent to the new user | | |
| Test Execution Steps: | | | | | | | | | | | |
| **S.No** | **Action** | | **Inputs** | **Expected Output** | | **Actual Output** | **Test Browser** | | **Test Result** | **Test Comments** | |
| 1 | Navigate to sign-up page | | Alicart sign-up URL(haven’t yet purchased the domain) | Sign-up page | | Sign-up page | Firefox Latest | | Pass | [Mabo 03/09/2024 11:44 AM]: The operation is successful | |
| 2 | Fill in registration form | | Valid personal details | Registration success message | | Registration success message | Firefox Latest | | pass | [Mabo 03/09/2024 11:44 AM]: The operation is successful | |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario ID** | | Login-1 | | | **Test Case ID** | | | Login-1B | | |
| **Test Case Description** | | Login – Negative test case | | | **Test Priority** | | | High | | |
| **Pre-Requisite** | | NA | | | **Post-Requisite** | | | NA | | |
| Test Execution Steps: | | | | | | | | | | | |
| **S.No** | **Action** | | **Inputs** | **Expected Output** | | **Actual Output** | **Test Browser** | | **Test Result** | **Test Comments** | |
| 1 | Enter invalid Email & any Password and hit login button | | Email id : max@max.com  Password: \*\*\*\*\*\* | The email address or phone number that you've entered doesn't match any account. [Sign up for an account.](https://www.facebook.com/r.php) | | The email address or phone number that you've entered doesn't match any account. [Sign up for an account.](https://www.facebook.com/r.php) | IE -11 | | Pass | [Mabo 03/09/2024 12:23 AM]: Invalid login attempt stopped | |
| 2 | Enter valid Email & incorrect Password and hit login button | | Email id : max@gmail.com  Password: \*\*\*\*\*\* | The password that you've entered is incorrect. [Forgotten password?](https://www.facebook.com/recover/initiate?lwv=120&lwc=1348092) | | The password that you've entered is incorrect. [Forgotten password?](https://www.facebook.com/recover/initiate?lwv=120&lwc=1348092) | IE -11 | | Pass | [Mabo 03/09/2024 09:05 AM]: Invalid login attempt stopped | |

­­

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario ID** | | CheckoutProcess-1 | | | **Test Case ID** | | | CheckoutProcess-1A | | |
| **Test Case Description** | | Checkout process with valid payment details | | | **Test Priority** | | | High | | |
| **Pre-Requisite** | | User is logged in, items in cart, payment method added | | | **Post-Requisite** | | | Order confirmation sent to the user | | |
| Test Execution Steps: | | | | | | | | | | | |
| **S.No** | **Action** | | **Inputs** | **Expected Output** | | **Actual Output** | **Test Browser** | | **Test Result** | **Test Comments** | |
| 1 | Navigate to cart and proceed to checkout | | NA | Checkout page | | Checkout page | Brave Latest | | Pass | [Mabo 03/08/2024 10:44 AM]: The operation is successful | |
| 2 | Enter shipping and payment details | | Valid address and payment info | Order placed with confirmation number | | Order placed with confirmation number | Brave Latest | | pass | [Mabo 03/08/2024 10:44 AM]: The operation is successful | |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario ID** | | CheckoutProcessCartFunctionality-1 | | | **Test Case ID** | | | CartFunctionality-1 | | |
| **Test Case Description** | | Adding a product to the shopping cart | | | **Test Priority** | | | Medium | | |
| **Pre-Requisite** | | User is logged in and product is in stock | | | **Post-Requisite** | | | NA | | |
| Test Execution Steps: | | | | | | | | | | | |
| **S.No** | **Action** | | **Inputs** | **Expected Output** | | **Actual Output** | **Test Browser** | | **Test Result** | **Test Comments/execution** | |
| 1 | Log into admin account | | admin credentials | Admin dashboard | | Admin dashboard | Brave Latest | | Pass | [Mabo 03/08/2024 10:44 AM]: The operation is successful | |
| 2 | Search and add product | | “Laptop” | Product added to cart | | Product added to cart | Brave Latest | | pass | [Mabo 03/08/2024 10:44 AM]: The operation is successful | |

**Comprehensive Testing Report Summary**

This report outlines the results and insights from the recent testing cycle conducted on key functionalities of our web application. The testing focused on user registration, login processes, the checkout process, and cart functionality, ensuring that these critical components perform as expected across different web browsers.

Test Coverage

***Features Tested:***

User Registration: Validation of the user sign-up process and subsequent confirmation.

Login Functionality: Testing of both positive and negative scenarios for user login.

Checkout Process: Verification of the checkout flow, including payment processing.

Cart Functionality: Ensuring products can be successfully added to the shopping cart.

***Browsers Tested:***

Firefox (Latest version)

Brave (Latest version)

***Test Results:***

All tests were completed successfully, with the functionalities behaving as expected across all browsers. This includes:

Successful User Registration: Users can register without issues, and a confirmation email is sent automatically.

Robust Login Validation: The application correctly handles invalid login attempts, providing appropriate error messages.

Smooth Checkout Process: Users can proceed through the checkout process, entering payment details without any problems, and receiving order confirmations.

Functional Cart: Products can be added to the cart, indicating proper inventory management and user session handling.

***Suggestions for Development:***

Extend Test Coverage: To better guarantee compatibility and user pleasure, future test cycles should incorporate additional browsers and mobile devices.

Describe Automated Testing: Repetitive test cases can be automated to save time and money and enable greater coverage.

Testing for security and performance: To preserve and improve user trust, performance optimization and security evaluations must be the key priorities of continuing development activities.

In summary the testing cycle proved to be extremely effective, as all of the intended capabilities operated as planned in various online settings. The suggestions made here will direct future efforts to improve the caliber of the application as well as the testing procedure.

**Submission Requirements:**

* Submit the completed assignment document by the due date.
* Ensure the document is well-organized, clear, and concise.
* Include any additional materials or resources used in the testing process.

**Grading Criteria:**

The assignment will be graded based on the following criteria: (during a presentation)

* Completeness and accuracy of the testing specifications
* Clarity and organization of the document
* Adherence to submission requirements and deadlines