### **Manual Testing Scope for Luma eCommerce Platform**

#### **Project Overview**

Luma is an eCommerce platform developed using Magento. It allows users to browse products, add items to their cart, checkout, and manage their orders. The primary goal of manual testing for Luma is to ensure the application functions correctly, providing a seamless user experience. The testing will cover various aspects, including user interface, functionality, performance, and security.

#### **Testing Objectives**

1. **Verify core functionalities of the Luma eCommerce platform:**
   * User registration and login.
   * Browsing and searching for products.
   * Managing the shopping cart and wishlist.
   * Completing the checkout process.
   * Order history and tracking.
   * Customer account management.
2. **Ensure the application is user-friendly and responsive across different devices and browsers.**
3. **Identify and document any bugs, defects, or inconsistencies.**
4. **Validate the security measures in place to protect user data and transactions.**

#### **Scope of Testing**

1. **Functional Testing:**
   * **User Registration and Login:**
     + Verify user registration with valid and invalid data.
     + Test login functionality with valid and invalid credentials.
     + Check password recovery process.
   * **Product Browsing and Searching:**
     + Verify product listing and categorization.
     + Test search functionality with various keywords and filters.
   * **Shopping Cart and Wishlist:**
     + Add and remove items from the shopping cart.
     + Update item quantities and validate totals.
     + Add and remove items from the wishlist.
   * **Checkout Process:**
     + Test checkout with different payment methods (credit card, PayPal, etc.).
     + Verify shipping options and costs.
     + Validate order confirmation and email notifications.
   * **Order Management:**
     + Verify viewing and tracking order history.
     + Test order cancellation and return processes.
   * **Account Management:**
     + Validate updating user profile information.
     + Test address book management.
     + Verify viewing and managing saved payment methods.
2. **Usability Testing:**
   * Assess the user interface for ease of use.
   * Ensure the layout is consistent across different pages.
   * Verify navigation flows are logical and intuitive.
   * Test application responsiveness on various devices (mobile, tablet, desktop).
3. **Compatibility Testing:**
   * Test the application on different browsers (Chrome, Firefox, Safari, Edge).
   * Validate functionality on different operating systems (Windows, macOS, iOS, Android).
4. **Performance Testing:**
   * Assess page load times for various sections of the application.
   * Verify the application's performance under different network conditions.
5. **Security Testing:**
   * Validate user authentication and authorization processes.
   * Check for vulnerabilities such as SQL injection, XSS, and CSRF.
   * Ensure sensitive data is encrypted and protected.
6. **Regression Testing:**
   * Perform regression testing to ensure new changes do not affect existing functionalities.
   * Re-test critical paths and high-impact areas after bug fixes and updates.

#### **Out of Scope**

* Automated testing (as this scope focuses on manual testing).
* Load testing and stress testing (requires specific tools and setups).
* Backend and database testing beyond user interface interactions.

#### **Testing Deliverables**

1. **Test Plan:**
   * Document outlining the testing strategy, objectives, scope, resources, schedule, and deliverables.
2. **Test Cases:**
   * Detailed test cases covering all functional, usability, compatibility, performance, and security aspects.
3. **Defect Reports:**
   * Log of all identified defects with descriptions, steps to reproduce, severity, screenshots, and status.
4. **Test Summary Report:**
   * Summary of testing activities, test results, defect metrics, and overall assessment of the application quality.

#### **Resources Required**

* Testers with experience in manual testing.
* Test environment with various devices and browsers.
* Access to test data and user accounts.

#### **Timeline**

* **Planning:** 1 week
* **Test Case Design:** 2 weeks
* **Test Execution:** 3 weeks
* **Defect Reporting and Retesting:** 2 weeks
* **Test Summary Report:** 1 week

### **Steps to Complete the Project and Add to GitHub**

1. **Set Up the Project Repository:**
   * Create a GitHub repository named Luma-Manual-Testing.
   * Initialise the repository with a README file.
2. **Plan and Document the Test Plan:**
   * Draft a comprehensive test plan outlining the testing strategy, scope, objectives, resources, and schedule.
   * Upload the test plan document to the GitHub repository.
3. **Design Test Cases:**
   * Create detailed test cases covering all aspects of the application.
   * Use a tool like Excel, Google Sheets, or a test management tool.
   * Save the test cases in a structured format and upload them to the repository.
4. **Execute Test Cases:**
   * Manually execute the test cases.
   * Log defects in a defect tracking tool (e.g., JIRA) or a spreadsheet.
5. **Report and Document Defects:**
   * Create a defect report detailing all identified defects with descriptions, steps to reproduce, and screenshots.
   * Upload the defect report to the repository.
6. **Perform Regression Testing:**
   * Re-test the application after bug fixes.
   * Ensure that new changes do not introduce new defects.
7. **Compile a Test Summary Report:**
   * Summarise the testing activities, test results, defect metrics, and overall application quality.
   * Upload the test summary report to the repository.
8. **Organise and Update the GitHub Repository:**
   * Ensure all documents are well-organised and named appropriately.
   * Update the README file with an overview of the project, testing scope, and a summary of findings.

By following these steps, you will have a comprehensive manual testing project for the Luma eCommerce platform that you can showcase on your GitHub resume. This project will demonstrate your ability to plan, execute, and report on manual testing activities for a real-world application.