Capstone Project: Automated Testing of OpenCart Demo Website

Project URL: https://demo.opencart.com/

Project Overview:

This project focuses on testing the demo version of the OpenCart website (https://demo.opencart.com/), an open-source e-commerce platform. The testing will involve UI automation with Selenium, API testing, manual testing, and behavior-driven development (BDD) using Cucumber.

Assignment Tasks:

1. Automated UI Testing with Selenium

• Task: Develop an automated test suite to verify the UI functionalities of the OpenCart demo website using Selenium.

o Test Environment Setup:

 Set up Selenium WebDriver with a test framework like TestNG or JUnit, and integrate with Maven or Gradle.

 Configure browser drivers for cross-browser testing.

 Use a CI/CD tool like Jenkins to automate test execution.

o Automated Test Cases:

 Product Search and Filter:

 Automate the search functionality, ensuring that search results are accurate.

 Verify filtering options (e.g., price range, categories) and sorting functionalities.

 Product Details and Add to Cart:

 Automate the process of viewing product details, adding products to the cart, and verifying that the cart updates correctly.

 Checkout Process:

 Automate the checkout process, including entering shipping details, selecting payment methods, and confirming orders.

 User Account Management:

 Automate registration, login, and password recovery processes.

 Verify account features such as order history and profile updates.

 Cross-Browser Testing: Ensure tests are executed across multiple browsers (Chrome, Firefox, Safari).

o Reporting: Generate detailed test execution reports using tools like ExtentReports.

2. Manual Testing

• Task: Perform manual testing to validate key functionalities and explore edge cases.

o Test Plan and Test Cases:

 Product Search and Filter: Manually test the product search, filtering, and sorting functionalities.

 Product Details and Cart Management: Verify product detail pages, cart updates, and checkout processes.

 User Account Management: Test account registration, login, and profile management.

 Document test cases with steps, expected results, and actual outcomes.

 Provide a summary of the manual testing process, including identified issues and test coverage.

 Log defects and track their resolution.

3. Behavior-Driven Development (BDD) with Cucumber

• Task: Implement BDD using Cucumber to define and automate test scenarios for the OpenCart demo website.

o Gherkin Syntax: Write feature files in Gherkin syntax to describe the behavior of the OpenCart platform.

o Test Scenarios:

 Product Search and Filter: Define scenarios for searching and filtering products.

 Product Details and Cart Management: Describe scenarios for viewing product details, adding items to the cart, and managing the cart.

 Checkout Process: Write scenarios for completing the checkout process, including payment and shipping.

 User Account Management: Define scenarios for user registration, login, and profile management.

o Execution and Reporting: Run Cucumber tests and generate reports showing the execution of each feature, including pass/fail status and detailed logs.

Tools and Technologies:

• UI Testing: Selenium WebDriver, TestNG/JUnit, Maven, Jenkins, ExtentReports.

• Manual Testing: Test management tools like Jira or TestRail for documentation and bug tracking.

• BDD: Cucumber, Gherkin syntax, Jenkins for integration.

• CI/CD: Jenkins.

• Expected Deliverables:

• Automated Test Scripts: Selenium test scripts covering key functionalities.

• Manual Test Cases: Documented test cases, test plans, and bug reports.

• Test Reports: Detailed reports from both automated and manual testing, summarizing results and findings.

• CI/CD Pipeline: A fully integrated CI/CD pipeline with automated tests.

• BDD Feature Files: Gherkin feature files outlining system behavior.

• Project Presentation: A presentation covering the testing strategy, tools used, challenges faced, and overall results.

consider suggestions

Here API Testing is mentioned only in project overview and more details can be added for the same including the validations , assertions and Error reporting mechanisms