Test Plan (nopCommerce)

Objective

In this document of the Test Plan for the nopCommerce web application, we are going to test the functional aspects of https://demo.nopcommerce.com/, nopcommerce leads in many countries as a key Ecommerce website, and we will be testing nopcommerce

Nopcommerce Tech stacks:

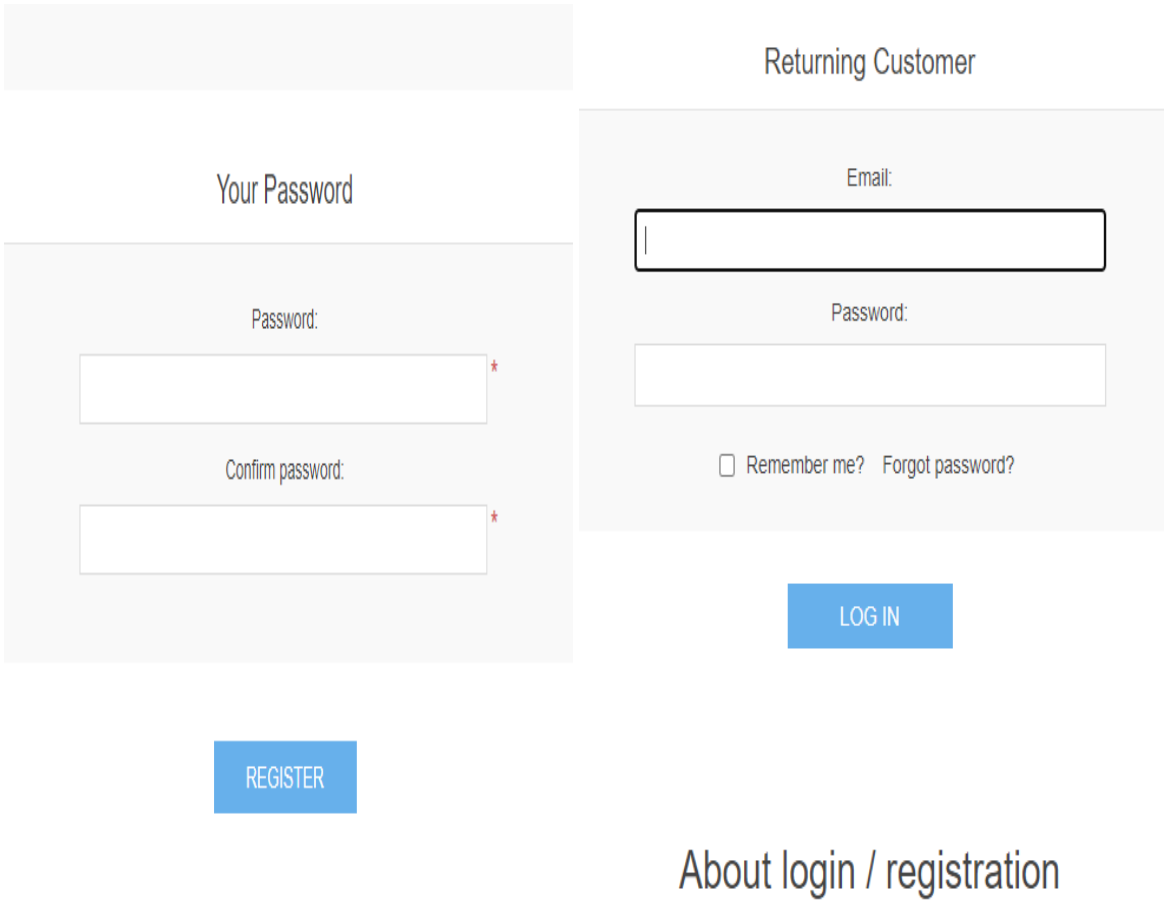
- Frontend: HTML, CSS, JavaScript, React, Redux, and TypeScript.
- Backend: Java, C++, Ruby on Rails, Node.js, and AWS Lambda.
- Database: Redis is used to store old data as an in-memory cache dataset.

Scope

The testing scope for demo.nopcommerce.com includes evaluating various features and functionalities, such as the user interface, the checkout process, search functionality, and mobile compatibility. The testing will involve functional testing to verify all features, smoke testing to ensure critical functionalities work correctly, and sanity testing for quick checks after bug fixes and updates. Testing will be conducted across different browsers (Chrome, Firefox, Safari, Edge, and Opera), operating systems (Windows, macOS, Linux, Android, and iOS), and various device types (desktops, laptops, tablets, and smartphones).

The success of the testing will be evaluated based on the number of defects found, categorized by severity, the time taken to complete the testing, and user satisfaction ratings gathered through feedback and surveys. The team members involved in the testing include the test lead, who will plan and coordinate activities and manage resources and schedules; testers, who will execute test cases, report and document defects, and perform regression testing; and developers, who will fix identified defects and ensure code quality and stability.

The tools and equipment used for testing will include test management tools like Jira and TestRail, automation tools such as Selenium and Appium, and bug tracking systems. Testing devices will comprise PCs, laptops, tablets, and smartphones, with network configurations to simulate different user environments. Documentation templates will be used for test plans and cases, defect reports, and test summary reports. This comprehensive scope ensures thorough testing of the nopcommerce demo site to provide a high-quality user experience.

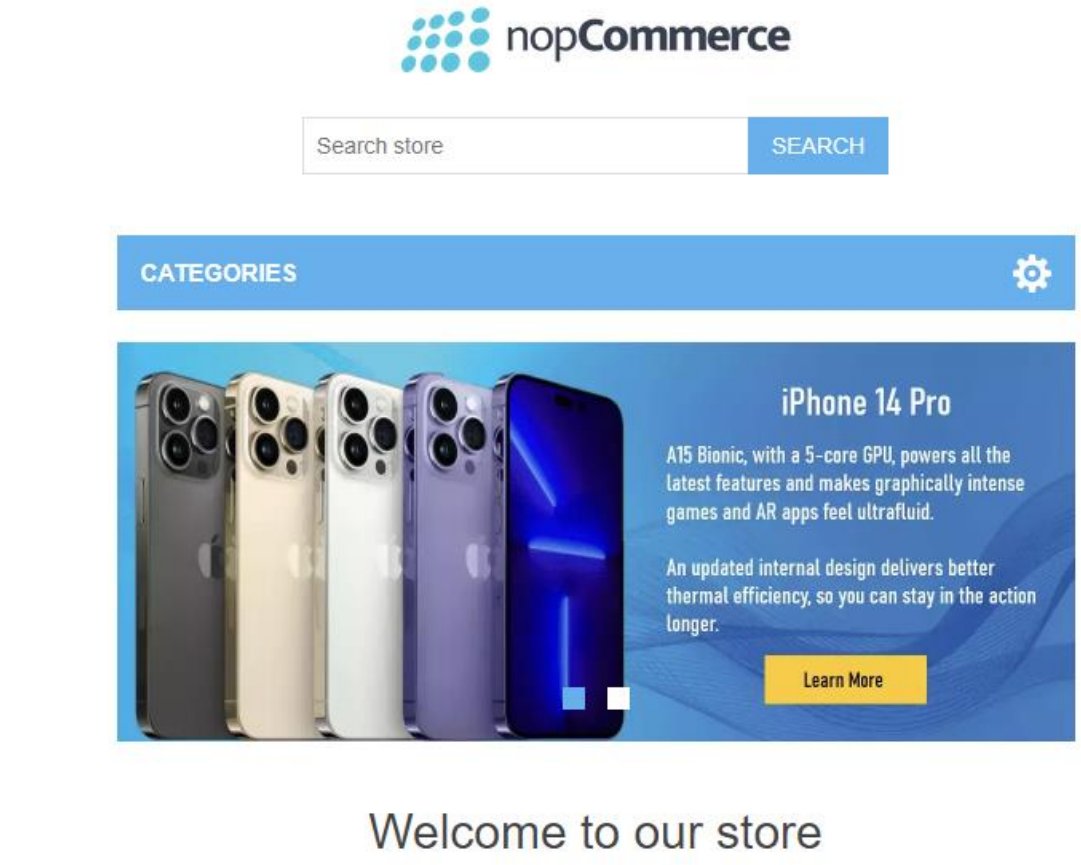


Inclusions

Introduction: This section would provide an overview of the test plan, including its purpose, scope, and goals.

Test Objectives: This section would outline the specific objectives of the testing, such as identifying and fixing defects, improving the user experience, or achieving a certain level of performance.

- Create account
- Sign In
- Sign out
- Search
- Category
- Cart
- Product page



Test Environments

The operating systems and versions that will be used for testing, such as Windows 10, macOS, and various mobile operating systems like Android and iOS. The browsers and versions that will be tested include Google Chrome, Firefox, Edge, and Safari. The device types and screen sizes that will be used for testing include desktop computers, laptops, tablets, and smartphones. The network connectivity and bandwidth that will be available for testing will include Wi-Fi, cellular, and wired connections. The hardware and software requirements for running the test cases will specify certain processor speeds, memory capacities, and storage capacities.

- Windows 10: Chrome, Firefox, and Edge
- Mac OS: Safari
- Android Mobile OS: Chrome
- iPhone Mobile OS: Safari

Defect Reporting Procedure

The criteria for identifying a defect include any deviation from the requirements, issues that negatively impact user experience, or technical errors. Defects are recognized when the application does not perform as expected, when user interactions are hindered, or when any system malfunctions occur.

The steps for reporting a defect involve using a designated template, providing detailed reproduction steps, and attaching relevant screenshots or logs. Testers are required to document the issue comprehensively to facilitate understanding and resolution by the development team.

The process for triaging and prioritizing defects involves assigning severity and priority levels to each defect. Severity indicates the impact of the defect on the system, while priority determines the urgency for fixing it. Defects are then assigned to the appropriate team members for investigation and resolution based on these levels.

Tools:

- Excel
- MS Word
- Google Sheets
- Google Docs

Test Strategy

Step 1: Test Scenario and Test Case Creation:

The first step involves creating test scenarios and test cases for the various features in scope. During this process, we will utilize a number of test design techniques, including Equivalence Class Partition, Boundary Value Analysis, Decision Table Testing, State Transition Testing, and Use Case Testing. Additionally, we will leverage our expertise by applying Error Guessing and Exploratory Testing to create comprehensive test cases. These test cases will be prioritized based on their importance and impact.

Step 2: Testing Procedure

Upon receiving a request for testing, our procedure is as follows:

- **Smoke Testing:** We will first conduct smoke testing to verify that the critical functionalities of the application are working. If the smoke testing fails, we will reject the build and wait for a stable build before performing in-depth testing of the application functionalities.
- **Stable Build Testing:** Once we receive a stable build that passes smoke testing, we will perform in-depth testing using the pre-created test cases.

- **Bug Reporting:** Bugs will be reported in the bug tracking tool. We will also send a status email to development management at the end of each day, detailing the defects found.

As part of the testing process, we will perform various types of testing, including Smoke Testing, Sanity Testing, Regression Testing, Retesting, and Functionality & UI Testing.

Step 3: Best Practices

To enhance our testing process, we will follow these best practices:

- **Context-Driven Testing:** We will perform testing as per the context of the given application, ensuring relevance and effectiveness.
- **Exploratory Testing:** In addition to executing the test cases, we will conduct exploratory testing based on our expertise to uncover issues that might not be covered by structured test cases.
- **End-to-End Flow Testing:** We will test end-to-end scenarios that involve multiple functionalities to simulate real user flows and ensure comprehensive coverage of the application.

Test Schedule

Following is the test schedule planned for the project –Task Time Duration

Task	**Dates**
▪ Creating Test Plan	June 10 - June 14, 2024
▪ Test Case Creation	June 15 - June 21, 2024
▪ Test Case Execution	June 22 - July 5, 2024
▪ Summary Reports Submission Date	July 6, 2024

Test Deliverables

- Test Plan
- Test Cases
- Test Execution Reports
- Defect Reports
- Summary Reports

Entry and Exit Criteria

The below are the entry and exit criteria for every phase of Software Testing Life Cycle:
Requirement Analysis

Entry Criteria:

• Once the testing team receives the Requirements Documents or details about the Project

Exit Criteria:

• List of Requirements are explored and understood by the Testing team
• Doubts are cleared

Test Execution

Entry Criteria:

• Test Scenarios and Test Cases Documents are signed-off by the Client
• Application is ready for Testing

Exit Criteria:

• Test Case Reports, Defect Reports are ready

Test Closure

Entry Criteria:

• Test Case Reports, Defect Reports are ready

Exit Criteria:

• Test Summary Reports

Tools

In this project, we will utilize the following tools:

- Word and Excel documents for documentation and reporting purposes.

Risks and Mitigations;

Risks:

1. Non-Availability of a Resource:
   - Mitigation: Implement backup resource planning to ensure continuity of work in case of resource unavailability.

2. Build URL is not working:
   - Mitigation: Allocate resources to work on other tasks while the issue with the build URL is being resolved.

3. Less time for Testing:
   - Mitigation: Ramp up resources dynamically based on the client's needs to ensure adequate testing within the given timeframe.

Approvals;

Various documents will be sent for client approval, including:

1. Test Plan
2. Test Scenarios
3. Test Cases
4. Reports

Testing activities will proceed to the next steps only after receiving the necessary approvals from the client.