|  |
| --- |
| TEST PLAN |
| Product Name: Selenium CI/CD Test Suite for nopCommerce |
|  |

|  |
| --- |
| Prepared by: Inbar .k.  Date: 2024 |

Overview 2

Scope 2

Inclusions 2

Test Environments 2

Exclusions 2

Test Strategy 3

Defect Reporting Procedure 3

Roles/Responsibilities 3

Test Schedule 3

Test Deliverables 4

Pricing 4

Entry and Exit Criteria 4

Suspension and Resumption Criteria 4

Tools 4

Risks and Mitigations 5

Approvals 5

# Overview

The purpose of this test plan is to outline the strategy and process for validating the functionality and performance of the Selenium CI/CD Test Suite for nopCommerce. The test suite will automate the testing of key features of the nopCommerce platform, integrating Selenium with CI/CD pipelines to ensure reliable and consistent test execution.

# Scope

The scope of the project includes testing the following features of ‘https://demo.nopcommerce.com/’ web application.

## Inclusions

* Automated functional testing of nopCommerce features using Selenium WebDriver.
* Integration of tests into a CI/CD pipeline (GitLab CI).
* Execution of tests on multiple browsers and operating systems (cross-browser testing).
* Reporting of test results, including logs, screenshots, and video recordings of failed tests.
* Testing of the following functionalities in nopCommerce:
  + User registration and login.
  + Product search and checkout process.
  + Admin panel functionality (product management, orders, customers).
  + Payment gateway integrations.
  + Responsiveness of the website.

## Test Environments

* Browsers: Chrome, Firefox, Edge.
* Operating Systems: Windows
* CI/CD Tools: GitLab CI.
* Test Automation Tools: Selenium WebDriver.
* Version Control System: Git.

## Exclusions

* Non-functional testing (performance, security, load testing) unless specified otherwise.
* Any features of nopCommerce outside the functional scope mentioned above.

# Test Strategy

The strategy will include the following approaches:

* Test Automation: Selenium WebDriver will be used for automating functional tests across various browsers and platforms. Tests will be written in Java, using TestNG as the testing framework and Maven for dependency management.
* Integration with CI/CD Pipeline: The test suite will be integrated with a CI/CD tool (GitLab CI), enabling automatic execution of tests on code commits or scheduled builds.
* Cross-Browser Testing: Tests will be executed on multiple browsers (Chrome, Firefox, and Edge) to ensure compatibility.
* Parallel Test Execution: Using Selenium Grid, tests will be executed in parallel to reduce execution time.

# Defect Reporting Procedure

* Bug Identification: During test execution, any failed test case will generate a detailed report with logs, screenshots, and video captures.
* Bug Logging: Defects will be logged in a defect tracking tool (browserstack), including information such as the test case, environment, steps to reproduce, and severity.
* Defect Resolution: Development teams will address defects, and re-testing will be done after bug fixes.
* Defect Closure: After validation of the defect fix, the defect will be closed in the defect tracking tool.

# Project Management Approach (Agile Principles)

The approach will be managed using Agile principles, ensuring flexibility and iterative progress. Key features include:

* Iterative Development: Work will be broken into smaller tasks and completed in short iterations (sprints).
* Prioritization: Tasks are prioritized based on their importance and dependencies, with regular review and adjustment of priorities.

# Roles/Responsibilities

NA

# Test Schedule

* Test Planning: Initial planning will include defining the scope, objectives, and test cases for the demo website, ensuring alignment with project goals.
* Test Case Development: Development of automated test scripts for key functionalities, ensuring coverage of all critical features (e.g., user registration, login, checkout).
* Manual Test Case Registration: All nopCommerce test cases have been manually registered on browserstack.com for better test case management and tracking.
* Test Environment Setup: Configure the test environment, including setting up browsers, drivers, and integrating with the CI/CD pipeline for seamless test execution.
* Test Execution: Tests will be executed continuously with each code commit through the CI/CD pipeline, ensuring tests run automatically on every update.
* Defect Resolution and Retesting: Defects will be identified during test execution, logged for resolution, and retested once fixes are applied.

# Test Deliverables

* Test Plan: Document outlining the testing approach, scope, and strategy.
* Test Scripts: Selenium WebDriver scripts for automating key functionalities (e.g., registration, login, checkout).
* Test Allure Reports: Detailed test reports with logs, and screenshots.
* Defect Logs: List of reported defects with their resolution status.
* CI/CD Integration: Fully functional CI/CD pipeline integration with test execution.

# Pricing

NA

# Entry and Exit Criteria

**Entry Criteria:**

* Test environment (browsers, drivers, CI/CD pipeline) is set up and configured.
* CI/CD pipeline is integrated with the test suite for automated execution.
* Test cases for critical functionalities are developed and ready for execution.

**Exit Criteria:**

* All planned test cases have been executed through the CI/CD pipeline.
* All critical defects have been resolved or deferred.
* Test documentation (including test plan, Test Scenarios, scripts, reports) is complete and finalized.

# Suspension and Resumption Criteria

* Suspension: Testing may be suspended if critical defects are found that block further testing or if the test environment is not available.
* Resumption: Testing will resume once the issues are fixed and the environment is restored.

# Tools

* Selenium WebDriver: For automating browser interactions.
* Python (Pytest): Testing framework for structuring and executing test cases and generating reports.
* GitLab CI: For continuous integration and automation of test executions.
* Allure: For generating detailed test reports.

# Risks and Mitigations

* Risk: Inconsistent test results due to browser or OS differences.
  + Mitigation: Use Selenium Grid for consistent test execution environments.
* Risk: Environmental issues, such as unavailable servers or CI/CD failures.
  + Mitigation: Set up backup environments and ensure proper monitoring of CI/CD tools.
* Risk: Lack of sufficient test coverage for certain features.
  + Mitigation: Prioritize critical features and regularly review test coverage.

# Approvals

NA