

Test Plan (demo.opencart.com)

| | |
|----------------------------|---|
| Test Plan | 1 |
| Objective | 2 |
| Scope | 2 |
| Inclusions | 2 |
| Test Environments | 2 |
| Defect Reporting Procedure | 3 |
| Test Strategy | 4 |
| Test Schedule | 6 |
| Test Deliverables. | 7 |
| Entry and Exit Criteria | 7 |
| Entry Criteria: | 7 |
| Exit Criteria: | 7 |
| Test Execution | 8 |
| Entry Criteria: | 8 |
| Exit Criteria: | 8 |
| Test Closure | 8 |
| Entry Criteria: | 8 |
| Exit Criteria: | 8 |
| Tools | 8 |
| Risks and Mitigations | 8 |
| Approvals | 8 |

Objective

The objective of this test plan is to ensure that the e-commerce platform meets all the functional requirements, provides a user-friendly experience, is secure, and performs well under various load conditions.

- React 18.2.0
- jQuery 2.1.1
- JavaScript
- Database Postgres SQL
- Web Server (Apache suggested)
- Nginx

Scope

The scope of this test plan includes the following areas:

Login and registration functionality
Add to cart and checkout functionality
Payment gateway functionality
Order management and fulfillment functionality
Performance testing of the platform

The criteria that will be used to evaluate the success of the testing, such as the number of defects found, the time taken to complete the testing, and user satisfaction ratings.

The roles and responsibilities of the team members involved in the testing, such as the test lead, testers, and developers.

The schedule and milestones for the testing, including the start and end dates, and the planned testing activities.

The tools and equipment that will be used for testing, such as testing software, hardware, and documentation templates.

Inclusions

The following items are included in this test plan:

Test strategy document
Test cases document
Test execution report
Defect report
Performance test report

Test Environments

The following test environments will be used:

- Development environment
- Test environment
- Production environment

The **operating systems** and versions that will be used for testing, such as Windows 10, macOS, or Linux.

The **browsers and versions** that will be tested, such as Google Chrome, Mozilla Firefox, or Microsoft Edge.

The **device types and screen sizes** that will be used for testing, such as desktop computers, laptops, tablets, and smartphones.

The **network connectivity and bandwidth** that will be available for testing, such as Wi-Fi, cellular, or wired connections.

The hardware and software requirements for running the test cases, such as a specific processor, memory, or storage capacity.

The **security protocols and authentication methods** that will be used to access the test environment, such as passwords, tokens, or certificates.

The access permissions and roles of the team members who will be using the test environment, such as testers, developers, or stakeholders.

| Name | Env url |
|----------|----------------------|
| QA | demo.opencart.com |
| Pre Prod | preprod.opencart.com |
| UAT | uat.opencart.com |
| Prod | app.opencart.com |

Windows 10 – Chrome, Firefox and Edge

- Mac OS – Safari Browser
- Android Mobile OS – Chrome
- iPhone Mobile OS - Safari

Defect Reporting Procedure

The criteria for identifying a defect, such as deviation from the requirements, user experience issues, or technical errors.

The **steps for reporting a defect**, such as using a designated template, providing detailed reproduction steps, and attaching screenshots or logs.

The **process for triaging and prioritizing defects**, such as assigning severity and priority levels, and assigning them to the appropriate team members for investigation and resolution.

The **tools and systems** that will be used for tracking and managing defects, such as a defect tracking software or a project management tool.

The **roles and responsibilities of the team members** involved in the defect reporting process, such as testers, developers, and the test lead.

The **communication channels** and frequencies for updating stakeholders on the progress and status of defects.

The metrics and metrics that will be used to measure the effectiveness of the defect reporting process, such as the number of defects found, the time taken to resolve them, and the percentage of defects that were successfully fixed.

| Defect Process | POC |
|----------------|--------|
| New Frontend | Wajid |
| Backend | Parvej |
| Dev Ops | Aakash |

Tools - JIRA

Test Strategy

| Component | Description |
|-------------|---|
| Objectives | List the overall goals and objectives of the testing process. |
| Test Levels | Specify the testing levels (unit, integration, system, acceptance) and their respective purposes, scopes, and objectives. |
| Test Types | List the types of testing to be conducted (functional, non-functional, regression) and their purposes and scopes. |

| | |
|---------------------|--|
| Test Techniques | Detail the testing techniques to be used for each test type (black-box, white-box, grey-box) and whether manual, automated, or a combination of both approaches will be employed. |
| Test Deliverables | List the test artifacts to be produced during the testing process (test plans, test cases, test scripts, test reports). |
| Test Environment | Describe the hardware, software, and network configurations required for testing, including target browsers, devices, and operating systems, as well as any tools or frameworks to be used. |
| Test Schedule | Provide an estimate of the time needed for each testing phase, taking into account resource availability, dependencies, and project deadlines. |
| Resource Allocation | Identify team members responsible for different testing tasks and outline their roles and responsibilities. |
| Risk Management | List potential risks and challenges that may arise during the testing process, along with contingency plans to address them. |
| Test Exit Criteria | Define the criteria that must be met before testing can be considered complete, such as a specific percentage of test cases executed, a certain level of test coverage, or a maximum number of unresolved defects. |

The first step is to create test scenarios and test cases for the various features in Scope.

While developing test cases, we'll use a number of test design techniques.

- o Equivalence Class Partition
- o **Boundary Value Analysis**
- o Decision Table Testing
- o State Transition Testing
- o Use Case Testing

We also use our expertise in creating Test Cases by applying the below:

- o Error Guessing

- o Exploratory Testing
- We prioritize the Test Cases

Step 2: Our testing procedure when we receive a request for testing:

- First, we'll conduct smoke testing to see if the various and important functionalities of the application are working.
- We reject the build, if the Smoke Testing fails and will wait for the stable build before performing in depth testing of the application functionalities.
- Once we receive a stable build, which passes Smoke Testing, we perform in depth testing using the Test Cases created.
- Multiple Test Resources will be testing the same Application on Multiple Supported Environments simultaneously.

We then report the bugs in bug tracking tool and send dev. management the defect found on that day in a status end of the day email.

As part of the Testing, we will perform the below types of Testing:

- o Smoke Testing and Sanity Testing
- o Regression Testing and Retesting
- o Usability Testing, Functionality & UI Testing
- We repeat Test Cycles until we get the quality product.

Step3 – We will follow the below best practices to make our Testing better:

- **Context Driven Testing** – We will be performing Testing as per the context of the given application.
- **Shift Left Testing** – We will start testing from the beginning stages of the development itself, instead of waiting for the stable build.
- **Exploratory Testing** – Using our expertise we will perform Exploratory Testing, apart from the normal execution of the Test cases.
- **End to End Flow Testing** – We will test the end-to-end scenario which involve multiple functionalities to simulate the end user flows.

Test Schedule

Following is the test schedule planned for the project –

Task Time Duration

| Task | Dates |
|-----------------------------------|----------|
| ▪ Creating Test Plan | 27/11/25 |
| ▪ Test Case Creation | 28/11/25 |
| ▪ Test Case Execution | 29/11/25 |
| ▪ Summary Reports Submission Date | 30/11/25 |

2. Sprints to Test the Application

Test Deliverables

The following are to be delivered to the client:

| Deliverables | Description | Target Completion Date |
|-----------------------|--|------------------------|
| Test Plan | Details on the scope of the Project, test strategy, test schedule, resource requirements, test deliverables and schedule | Date |
| Functional Test Cases | Test Cases created for the scope defined | Date |
| Defect Reports | Detailed description of the defects identified along with screenshots and steps to reproduce on a daily basis. | NA |
| Summary Reports | Summary Reports – Bugs by Bug#, Bugs by Functional Area and Bugs by Priority | Date |

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

The following are the list of Tools we will be using in this Project:

- JIRA Bug Tracking Tool
- Mind map Tool
- Snipping Screenshot Tool
- Word and Excel documents

Risks and Mitigations

The following are the list of risks possible and the ways to mitigate them:

Risk: Non-Availability of a Resource

Mitigation: Backup Resource Planning

Risk: Build URL is not working

Mitigation: Resources will work on other tasks

Risk: Less time for Testing

Mitigation: Ramp up the resources based on the Client needs dynamically

Approvals

Team will send different types of documents for Client Approval like below:

- Test Plan
- Test Scenarios
- Test Cases
- Reports

Testing will only continue to the next steps once these approvals are done