****

****

**TEST PLAN**

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **TOPICS** | **PG NO** |
| Objective |  |
| Purpose of Test Plan |  |
| Scope |  |
| QA Resources Allocated |  |
| QA Backup Resources |  |
| Out of Scope |  |
| Test Environments |  |
| Test Strategy |  |
| Test Schedule |  |
| Testing |  |
| Defect Reporting Procedure |  |
| Fixing |  |
| Verification |  |
| Not Fixed |  |
| Test Deliverables |  |
| Entry and Exit Criteria |  |
| Entry Criteria: |  |
| Exit Criteria: |  |
| Test Execution |  |
| Entry Criteria |  |
| Exit Criteria |  |
| Test Closure |  |
| Entry Criteria |  |
| Exit Criteria |  |
| Tools |  |
| Risks and Mitigations |  |
| Approvals |  |

****

**TEAM MEMBERS**

|  |  |
| --- | --- |
| **ROLE** | **NAME** |
| Project Manager | Rajesh Varma |
| Development Team | Rahul  Mohit  Shreyash |
| QA Team | Shefali Saroj  Amit Sharma  Varun Patel |

**DOCUMENT LOG**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Name** | **Designation** | **Version No** | **Date** |
| **Created By** | Shefali Saroj | QA | Version 1.0 | 04-Jan-2024 |
| **Reviewed By** | Amit Sharma | QA Lead | Version 1.0 | 06-Jan-2024 |

**OBJECTIVE**

OpenCart, a free and open-source e-commerce platform, serves as a foundational tool for individuals ranging from web developers to shop owners aiming to establish their online presence. Focused on the frontend in this test plan, the following essential features will be assessed.

Conduct comprehensive testing of OpenCart, ensuring foundational website development support, successful user registration, validation of homepage structure, effective product display, seamless user authentication, functional shopping cart operations, smooth checkout processes, interactive product reviews, and validated gift voucher functionalities.

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



**PURPOSE OF TEST PLAN**

The OpenCart test plan ensures a methodical approach to validate the e-commerce platform's functionality, performance, and security. It outlines testing goals, scope, and methods to identify and address issues, ensuring the delivery of a reliable and high-quality system that meets user expectations.

**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 fulfilment 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.

**QA Resources Allocated**

* Shefali Saroj
* Amit Sharma



**QA Backup Resources**

* Varun Patel

**Out of Scope**

* + Test automation.
  + All the features that doesn’t mentioned in scope.
  + Any third-party features.

**TEST ENVIRONMENTS**

The following test environments will be used:

1. Operating System: Window 11.
2. Browser: Google Chrome.
3. Device: Laptop or Computer.
4. Network connectivity: Wi-fi cellular.

**TEST STRATEGY**

**Step 1**: 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.

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

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

* Error Guessing
* 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:

* Smoke Testing and Sanity Testing
* Regression Testing and Retesting
* 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** | **DATE** |
| Creating Test Plan |  |
| Test Case Creation |  |
| Test Case Execution |  |
| Summary Reports Submission  Date |  |

**TESTING**

* QA will develop test scenarios.
* QA will develop test cases based on the test scenarios.
* QA will execute test cases.

**DEFECT REPORTING PROCEDURE**

* QA will make bug reports using Microsoft Excel and Jira.
* QA will make RTM (Requirement Traceability Matrix) based on the test case execution and bug reports.
* QA will assign bugs to Project Manager.

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, s**uch 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 a**nd frequencies for updating stakeholders on the progress and status of defects.

The 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 | Devesh |
| Backend | Sonal |
| Dev Ops | Prajeeth |

**Fixing**

Developer will fix the assigned bug and assign it to QA.

**Verification**

QA will verify the fix on assigned bugs

**Not Fixed**

If bug is not fixed QA will re-assign the bug to the developer.

****

**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 requirement, test deliverables and schedule. |  |
| Functional Test Cases | Test Cases created for the scope defined. |  |
| Defect Reports | Detailed description of the defects identified along with screenshots and steps to reproduce on the daily basis. |  |
| Summary Reports | Bugs by Bugs,  Bugs by Functional Area,  Bugs by Priority. |  |

**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