Open Cart

Functional Requirement Specification (FRS)

Functional Requirements Specification (FRS)

For Manual Testing of OpenCart Demo Website

Table of Contents

1. Overview
2. Scope
3. Functional Requirements
    a. 3.1 Frontend Requirements
    b. 3.2 Backend (Admin Panel) Requirements
    c. 3.3 Database Requirements
4. Non-Functional Requirements
5. Test Environment
6. Test Deliverables
7. Roles and Responsibilities
8. Tools Required
9. Risk & Mitigation Strategies
10. Exit Criteria
11. Approvals

1. Overview

OpenCart is an open-source e-commerce platform designed for creating, managing, and scaling online stores. It provides a complete storefront (frontend) and a powerful admin panel (backend) that allows businesses to manage products, categories, customers, orders, payments, shipping, extensions, and theme customization. The platform is modular, user-friendly, and widely used for commercial online shops.

This document captures the functional requirements for manually testing all major features of the OpenCart demo website. The entire analysis, planning, and testing activities for this project are performed solely by the author.

Author: Kedar Gaikwad (Independent Project Execution)

2. Scope

- Includes testing of all modules available in the OpenCart demo website frontend and backend.
- Covers functionality, UI behavior, input validations, navigation, and data consistency.
- Excludes performance testing, load testing, and deep database security audits.

3. Inclusion

- Register
- Login &Logout
- Forgot Password
- Search
- Product Compare
- Product Display page
- Add to Cart
- Wish List
- Shopping Cart
- Currencies
- Home Page
- Checkout Page
- My Account Page
- Order History Page
- Download Page
- Contact Us Page
- Menu Options
- Footer Options
- Category Pages

4. Functional Requirements

3.1 Frontend Requirements

Home Page

- Must display main banner, featured products, navigation menu, footer sections.
- All hyperlinks must be clickable and direct to correct pages.

User Registration & Login

- New users must be able to register with mandatory fields.
- Login must validate credentials correctly.
- Invalid credentials must trigger appropriate error messages.

Product Catalogue

- Categories and subcategories must display relevant products.
- Sorting and filtering options must function correctly.
- Product details page must show images, pricing, specs, reviews.

Search Functionality

- Search bar must return relevant results based on keywords.
- No-result scenario must show proper message.

Cart & Checkout

- Users must be able to add products to cart.
- Quantity update and remove item must work.
- Checkout must support guest checkout and registered user checkout.
- Payment and shipping methods must display correctly.

Wishlist & Comparison

- Logged-in users must be able to add items to wishlist.
- Product comparison must display feature differences.

Contact Us Page

- Form submission must validate required fields.
- Confirmation message must appear.

Account Section

- User account dashboard must allow viewing and editing profile details.
- Order history must load correctly.

3.2 Backend (Admin Panel) Requirements

Admin Login

- Only valid admin credentials should allow access.
- Password reset workflow must function.

Dashboard

- Must display analytics widgets like sales, orders, customers.

Catalog Management

- Admin should be able to add, edit, delete products.
- Categories must support CRUD operations.
- Product image upload must work.

Customer Management

- Admin must see list of customers with filters.
- CRUD operations must work.

Order Management

- Admin must view order list with status filters.
- Order status updates must reflect properly.

Extensions & Modules

- Enable/disable modules should work.
- Payment/shipping method configuration must save settings.

Reports Section

- Sales report must load with correct filters.
- Product views/purchases reports must work.

3.3 Database Requirements

- All CRUD operations must reflect accurately in DB tables.
- User data must be stored in correct tables.
- Orders must create correct entries in order-related tables.
- Product updates must reflect immediately.

4. Non-Functional Requirements

- Website must render properly on major browsers.
- UI must maintain consistency across pages.
- All pages must load within acceptable response time.

5. Test Environment

Hardware

- Windows/Linux/Mac system
- Minimum 4 GB RAM
- Stable internet connection

Software

- Browser: Chrome, Firefox, or Edge
- OpenCart demo URL
- (Optional) Local XAMPP environment for deeper testing

6. Test Deliverables

- Test Plan
- Test Scenarios
- Test Cases
- Test Execution Report
- Defect Report
- Traceability Matrix

7. Roles and Responsibilities

Test Engineer

- Prepare test cases
- Execute manual tests
- Log defects
- Retest fixes

Test Lead

- Approve test plan & strategy
- Review test cases and execution
- Ensure timely delivery

Developer

- Fix reported issues
- Communicate resolutions

Project Manager

- Oversee entire testing process
- Provide resources and approvals

Testing Project

1. Purpose

Define the overall approach, testing scope, techniques, responsibilities, environment, and risk handling for manual testing of the OpenCart demo website.

All testing activities are executed independently by **Kedar Gaikwad**.

2. Testing Objectives

- Verify that all OpenCart frontend & backend features behave as expected.

- Ensure data consistency between UI actions and database entries.
- Validate user flows like registration, login, cart management, checkout, orders, admin operations, and catalog management.
- Identify defects early, document them clearly, and ensure retesting until closure.

3. Scope of Testing

3.1 In Scope

- **Functional Testing**
  - UI navigation
  - Product catalogue
  - Search
  - Cart & checkout
  - Payment/shipping configuration
  - Order management
  - Profile & Account
  - Admin CRUD operations
- **Validation & Error Handling**
  - Mandatory field validation
  - Incorrect input handling
- **Database Verification**

    - CRUD consistency
    - Data mapping across tables
- **Compatibility Testing**
    - Chrome, Firefox, Edge (latest versions)
- **Regression Testing**
    - After major fixes
- **Smoke Testing**
    - On each testing cycle start

3.2 Out of Scope

- Load testing, stress testing
- Backend API testing
- Security penetration testing
- Automation testing

4. Testing Approach

4.1 Functional Testing Approach

- Use **scenario-based** and **end-to-end user flow** testing.
- Derive test cases from:
    - Business flows
    - UI behavior
    - OpenCart standard functionalities
- Validate:

- Inputs
- Navigation
- Output correctness
- Data consistency

4.2 Test Design Approach

- High-level scenarios → detailed test cases
- Every test case includes:
  - Pre-conditions
  - Steps
  - Expected results
  - Actual results
  - Status
  - Severity/Priority

4.3 Defect Management

- Every defect logged with:
  - Steps to reproduce
  - Expected vs actual result
  - Screenshot
  - Severity & priority
- Defects tracked until closure.

5. Test Levels

5.1 Smoke Testing

Ensures critical features work:

- Homepage load
- Product details
- Add to cart
- Admin login

5.2 Functional Testing

Deep validation of each module:

- User module
- Product module
- Orders
- Checkout
- Payment
- Admin CRUD

5.3 Regression Testing

Executed after major defect fixes to ensure:

- No existing workflow breaks
- Core checkout & cart flows are stable

6. Test Environment

Environment Setup

- Device: Windows 10+
- Browsers: Chrome, Firefox, Edge
- Network: Stable internet (demo site is cloud-hosted)
- Tools:
    - Test cases: Excel/Google Sheets
    - Defects: Excel/Jira (if used)
    - Browser Developer Tools (for validation)

No server installation required unless local environment is set up via XAMPP.

7. Test Data Strategy

- Create fresh test data for:
    - Users
    - Orders
    - Products (if admin access is available)
- Clean up test data where possible
- Avoid using shared demo data as truth source

8. Entry & Exit Criteria

Entry Criteria

- Requirements understood
- Test environment ready
- Test cases prepared
- Access to frontend & admin areas available

Exit Criteria

- All test cases executed
- No high or medium severity defects open
- Minimum pass rate: **≥95%**
- Test summary report prepared

9. Risks & Mitigation

Risk	Impact	Mitigation
Demo site downtime	High	Keep secondary local installation
Unstable demo data	Medium	Create new data for each test
Browser inconsistencies	Medium	Cross-browser validation
Delayed fixes	Medium	Prioritize critical paths
No control on backend config	Low	Limit tests to user-level features

10. Roles & Responsibilities

Since this is an independent project:

- Requirement analysis
- Test estimation
- Test case design
- Manual execution
- Defect reporting
- Retesting
- Documentation (FRS, Test Strategy, Test Plan, Reports)

11. Deliverables

- Functional Requirements Specification (FRS)
- Test Strategy Document
- Test Plan
- Test Scenarios
- Test Cases
- Test Execution Report
- Defect Report
- Traceability Matrix
- Final Test Summary Report

If you want this added to the PDF or inserted into the existing FRS document, tell me and I'll generate a new polished version.

8. Tools Required

- Test case management: Excel, Google Sheets, Jira (optional)
- Defect tracking: Jira, Bugzilla, or Excel
- Browser developer tools

9. Risk & Mitigation Strategies

Risk	Impact	Mitigation
Demo site downtime	High	Test during active hours, maintain local setup
Unstable data on demo	Medium	Avoid relying on stored data; re-create test data
Browser incompatibility	Medium	Test on multiple browsers
Incorrect environment configuration	High	Document setup steps clearly

10. Exit Criteria

- All high and medium severity defects resolved.
- Test cases executed with ≥95% pass rate.
- No unresolved blockers.
- Test summary report approved.

11. Approvals

Role	Name	Signature	Date
Test Lead	Kedarraje Gaikwad		15/11/2025
Project Manager			
QA Manager			

End of Document