OpenCart – QA Testing Project Documentation

---

1. Project Overview

1.1 Project Name:

OpenCart E-Commerce Application – Manual, API, and Database Testing

1.2 Purpose of Testing

The purpose of this project is to validate the functionality, performance, and reliability of the OpenCart application running on a local server. The testing scope includes:

- Manual functional testing of web features
- API testing using Postman
- Database testing using MySQL
- Validation of end-to-end user workflows

1.3 Test Environment

Component	Version / Details
Application	OpenCart (Localhost)
Web Server	XAMPP / WAMP
Database	MySQL (phpMyAdmin)
API Tool	Postman
Browser	Chrome / Firefox
OS	Windows / macOS

---

2. Scope of Testing

2.1 In-Scope

- User registration & login
- Product browsing, search, and categories
- Shopping cart & checkout
- Payment methods
- Order placement
- Email notifications (local SMTP testing)
- Admin panel functionalities
- API endpoints (customers, orders, products)
- Database validation (CRUD operations)

2.2 Out-of-Scope

- Performance / Load testing
- Security penetration testing
- Mobile app testing

---

3. Test Approach

3.1 Manual Testing

Perform functional and UI testing following these types:

- Smoke Testing
- Functional Testing
- Regression Testing
- Cross-browser Testing
- Boundary & Negative Test Cases

---

4. Test Scenarios (Manual Testing)

4.1 User Module

Scenario	Expected Result
Register new user	Email verification or account creation success
Login with valid credentials	User logs in
Login with invalid credentials	Error displayed

4.2 Product Module

Scenario	Expected Result
View product details	Product images, description, price visible
Click alternate images	Main image updates
Add product to cart	Cart updates with item

4.3 Checkout Module

Scenario	Expected Result
Select payment method	All methods displayed
Place an order	Order success + email sent

---

5. Bug Reports (Sample)

Include issues such as:

- Alternate images not working
- Discount price not shown
- Invalid quantity accepted
- Payment method missing
- Confirmation email not received
- Duplicate customer names in DB

---

6. API Testing Using Postman

6.1 API Test Scope

- Customer API
- Product API
- Order API
- Authentication API

6.2 Sample API Test Cases

1. Create Customer (POST)

Endpoint: /index.php?route=api/customer/create
Request Body: JSON
Expected Result: 200 OK – customer created

2. Get Product List (GET)

Endpoint: /index.php?route=api/product
Expected Result: List of products returned

3. Add Product to Cart (POST)

Endpoint: /index.php?route=api/cart/add
Expected Result: Product added to cart successfully

4. Create Order (POST)

Endpoint: /index.php?route=api/order/add
Expected Result: Order ID generated

---

7. Database Testing Using MySQL

7.1 DB Scope

- Validate table structure
- Verify data mapping
- Check duplicate records
- Perform CRUD testing

7.2 Important Tables

- oc_customer
- oc_order
- oc_product
- oc_category
- oc_cart

7.3 Sample SQL Queries

1. Check Duplicate Customer (firstname, lastname, email)

```
SELECT c1.customer_id, c1.firstname, c1.lastname, c1.email
FROM oc_customer c1
JOIN oc_customer c2
ON c1.customer_id <> c2.customer_id
AND c1.firstname = c2.firstname
AND c1.lastname = c2.lastname
AND c1.email = c2.email;
```

2. Verify Product Count

```
SELECT COUNT(*) FROM oc_product;
```

3. Validate Customer Order Mapping

```
SELECT customer_id, order_id FROM oc_order;
```

4. Check Orders With No Products

```
SELECT * FROM oc_order WHERE order_id NOT IN (SELECT order_id FROM oc_order_product);
```

---

8. Test Metrics

Metric	Description
Total Test Cases	xxx
Passed	xxx
Failed	xxx
Blocked	xxx
Severity	Critical / Major / Minor
Priority	High / Medium / Low

---

9. Conclusion

The OpenCart manual, API, and database testing ensured:

- Functional accuracy
- Data integrity
- Valid user flows
- Reliable API performance

All critical bugs were logged, and retesting/regression was performed.