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, reliability, and data integrity of the OpenCart application running on a local environment. The testing includes:

- Manual functional testing
- API testing using Postman
- Database testing using MySQL
- End-to-end workflow validation

1.3 Test Environment

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

---

2. Scope of Testing

2.1 In-Scope

- User registration & login
- Product browsing, search, categories
- Shopping cart & checkout
- Payment methods
- Order placement
- Email notifications (SMTP local)
- Admin panel functionalities
- API endpoints (Users, Posts, Comments, Todos)
- Database validation (CRUD, FK relations, duplicates)

2.2 Out-of-Scope

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

---

3. Test Approach

3.1 Manual Testing Types

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

---

4. Manual Test Scenarios

4.1 User Module

- Register new user
- Login with valid credentials
- Login with invalid credentials

4.2 Product Module

- View product details
- Click alternate images
- Add product to cart

4.3 Checkout Module

- Select payment method
- Place order successfully

---

5. Bug Reports (Sample)

- Alternate product images not working
- Discount price not displayed
- Invalid quantity accepted
- Missing payment methods
- Confirmation email not received
- Duplicate customer names in database

---

6. API Testing Using Postman

6.1 API Test Scope

- Users API
- Posts API
- Comments API
- To-Do API
- Authentication (Token-based)

6.2 Sample API Test Cases

1. **Get Users List (GET)**
   Endpoint: `https://gorest.co.in/public/v2/users`
   Expected Result: 200 OK – List of users returned.

2. **Get User with Invalid ID (GET)**
   Endpoint: `https://gorest.co.in/public/v2/users/99999999`
   Expected Result: 404 Not Found – "Resource not found".

3. **Create New User (POST)**
   Endpoint: `https://gorest.co.in/public/v2/users`
   Request Body (JSON):

```
{
  "name": "Test User",
  "gender": "female",
  "email": "test.user@email.com",
  "status": "active"
}
```

   Expected Result: 201 Created – User created.

4. **Create User with Invalid Email (POST)**
   Endpoint: `https://gorest.co.in/public/v2/users`
   Request Body (JSON):

```
{
  "name": "Bad Email",
  "gender": "male",
  "email": "bademail",
  "status": "active"
}
```

   Expected Result: 422 Unprocessable Entity – Invalid email.

5. **Update Existing User (PUT)**
   Endpoint: `https://gorest.co.in/public/v2/users/{user_id}`
   Request Body (JSON):

```
{
  "name": "Updated Name"
}
```

   Expected Result: 200 OK – User updated.

6. **Create New Post (POST)**
   Endpoint: `https://gorest.co.in/public/v2/posts`
   Request Body (JSON):

```
{
  "user_id": 1234,
  "title": "My new post",
  "body": "Post content here"
}
```

   Expected Result: 201 Created – Post created.

7. **Create New To-Do Item (POST)**
   Endpoint: `https://gorest.co.in/public/v2/todos`
   Request Body (JSON):

```
{
  "user_id": 94699,
  "title": "DEPI",
  "due_on": "2025-11-09T00:00:00.000+05:30",
  "status": "pending"
}
```

   Expected Result: 201 Created – To-Do created.

---

7. Database Testing Using MySQL

7.1 DB Scope

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

7.2 Important Tables

Table Name	Description
oc_customer	Stores customer data (firstname, lastname, email, etc.)
oc_order	Stores all order information including customer and totals
oc_product	Stores product info such as price, quantity, and status
oc_category	Holds category information for product grouping
oc_order_product	Links products to orders (order line items)
oc_address	Contains customer address details
oc_product_to_category	Junction table linking products to categories

7.3 Sample SQL Queries

1. **Customers Missing Firstname or Lastname**

```
SELECT * FROM oc_customer
WHERE firstname = '' OR firstname IS NULL
   OR lastname = '' OR lastname IS NULL;
```

2. **Customers Without Addresses**

```
SELECT c.customer_id
FROM oc_customer c
LEFT JOIN oc_address a ON c.customer_id = a.customer_id
WHERE a.address_id IS NULL;
```

3. **Duplicate Customers**

```
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;
```

4. **Orders Without Products**

```
SELECT o.order_id
FROM oc_order o
LEFT JOIN oc_order_product op ON o.order_id = op.order_id
WHERE op.order_id IS NULL;
```

5. **Products Without Categories**

```
SELECT p.product_id
FROM oc_product p
LEFT JOIN oc_product_to_category pc ON p.product_id = pc.product_id
WHERE pc.product_id IS NULL;
```

6. **Orders Linked to Deleted Customers**

```
SELECT o.order_id, o.customer_id
FROM oc_order o
LEFT JOIN oc_customer c ON o.customer_id = c.customer_id
WHERE o.customer_id <> 0 AND c.customer_id IS NULL;
```

7. **Zero or Negative Stock Products**

```
SELECT product_id, quantity
FROM oc_product
WHERE quantity <= 0;
```

8. Test Metrics

Metric	Total
Total Test Cases	426
Passed	354
Failed	72
Severity	Critical / Major / Minor
Priority	High / Medium / Low

---

9. Conclusion

Description	Value
Overall Pass Rate	83.1%
Highest Performing Module	API Users / API Posts / API Comments / API Todo (100% Pass)
Lowest Performing Module	Admin Work (57.3% Pass)
Recommendations	Focus on Admin Work module improvements and failed test cases in Cart Management and Product Page