Project Title

Manual, API & Database Testing of an E-Commerce Web Application

1. Introduction

This project focuses on testing an e-commerce web application manually along with basic API and database validation.
The goal of this project is to ensure that the main user flows such as login, product selection, cart management, and order placement are working correctly and no critical issues exist before release.

This project was done after completing a beginner-level SQA project to gain more practical and real-world testing experience.

2. Application Overview

Application Name

E-Commerce Web Application

Application Description

The application allows users to register, log in, browse products, add items to cart, place orders, and view order history.
Admin users can view customer orders but cannot modify them in this scope.

3. User Roles

- Customer
- Admin (limited access)

4. Scope of Testing

In Scope

- User Registration & Login
- Product Listing & Search
- Add to Cart
- Cart Update & Remove

- Checkout & Order Placement
- Order History
- Session & Security validation
- API testing (basic)
- Database verification (basic)

Out of Scope

- Real payment gateway
- Delivery partner system
- Refund processing

5. Types of Testing Performed

- Functional Testing
- Negative Testing
- Boundary Value Analysis
- Regression Testing
- API Testing using Postman
- Database Testing using SQL
- Basic Security Testing

6. Test Environment

Item	Description
OS	Windows 10
Browser	Google Chrome
Testing Type	Manual
API Tool	Postman
Database	MySQL

Item	Description
Test Data	Dummy users & products

7. Test Scenarios (Human-written)

1. User registers with valid and invalid data
2. User logs in using correct and incorrect credentials
3. Products load correctly on homepage
4. Search returns relevant products
5. Filter works within selected price range
6. User adds product to cart
7. User updates quantity in cart
8. User removes product from cart
9. Checkout without login
10. Order placement after login
11. Duplicate order submission
12. Session expiry during checkout
13. Order history accuracy
14. Unauthorized admin access
15. Browser back button after logout

8. Test Cases (Realistic & Natural)

AUTHENTICATION

TC_01
Scenario: Register with valid data
Steps:
Enter valid name, email, password → Click Register
Expected Result: User account created successfully
Status: Pass

TC_02
Scenario: Register with existing email
Expected Result: Proper error message shown
Status: Pass

TC_03
Scenario: Login with valid credentials
Expected Result: User redirected to homepage
Status: Pass

TC_04
Scenario: Login with wrong password
Expected Result: Error message shown
Actual: Generic message displayed
Status: Fail

PRODUCT & SEARCH

TC_05
Scenario: View product list
Expected Result: Products load with name, price, image
Status: Pass

TC_06
Scenario: Search valid product
Expected Result: Relevant products shown
Status: Pass

TC_07
Scenario: Search invalid keyword
Expected Result: No results message
Status: Pass

CART

TC_08
Scenario: Add product to cart
Expected Result: Product added successfully
Status: Pass

TC_09
Scenario: Increase quantity beyond stock

Expected Result: Stock warning shown
Actual: Quantity increases
Status: Fail

TC_10
Scenario: Remove product from cart
Expected Result: Product removed
Status: Pass

CHECKOUT & ORDER

TC_11
Scenario: Checkout without login
Expected Result: Redirect to login page
Status: Pass

TC_12
Scenario: Place order successfully
Expected Result: Order placed confirmation shown
Status: Pass

TC_13
Scenario: Double click place order
Expected Result: Only one order created
Actual: Multiple orders created
Status: Fail

SECURITY & SESSION

TC_14
Scenario: Access dashboard URL without login
Expected Result: Redirect to login
Status: Pass

TC_15
Scenario: Logout and use browser back
Expected Result: User remains logged out
Status: Pass

9. API TESTING (Postman – Human Style)

Login API

- Method: POST
- Status Code: 200 (Valid login)
- Status Code: 401 (Invalid login)
- Verified response message & token

Product List API

- Method: GET
- Verified product count and response time

Order API

- Method: POST
- Verified order created only once (Bug found)

10. DATABASE TESTING (Basic)

```
SELECT * FROM users WHERE email='testuser@gmail.com';
```

```
SELECT * FROM orders WHERE user_id=102;
```

Verified:

- User created correctly
- Order amount matches cart total
- Duplicate orders found for same request

11. BUG REPORTS (Natural, Not AI-polished)

BUG_01

Module: Login
Title: Incorrect error message for wrong password
Severity: Medium
Priority: High
Status: Open

BUG_02

Module: Cart
Title: Cart allows quantity more than stock
Severity: High
Priority: Critical
Status: Open

BUG_03

Module: Checkout
Title: Duplicate orders created on multiple clicks
Severity: Critical
Priority: Critical
Status: Open

12. Test Summary Report

Item	Count
Total Test Cases	15
Passed	11
Failed	4
Critical Bugs	1
Testing Status	Not Ready for Production

13. Conclusion

Manual, API, and database testing were performed on the e-commerce application.
Although most core functionalities are working, critical issues were identified in the cart and checkout modules. These issues should be fixed before production release.

14. Interview Talking Points (VERY HUMAN)

- Followed STLC properly
- Designed test cases based on real user behavior
- Found business-critical bugs
- Used Postman for API validation
- Used SQL to verify backend data