**Test Case Design Approach & Core Module Functionality**

**Introduction**

When designing test cases for WafiCommerce, it was important to first understand the core functionality of each module. By defining what each module is supposed to do, we could ensure that our test cases cover all critical workflows, business logic, and system interactions.

Additionally, while writing test cases, we focused on a thoughtful approach—choosing scenarios that test not only normal operations but also edge cases, system failures, and data consistency across modules.

**Product Module**

**Core Functionality**

The Product Module is the foundation of WafiCommerce. It manages product creation, inventory tracking, pricing, availability, and batch updates. The module ensures that businesses can add, modify, and remove products efficiently while maintaining accurate stock levels.

**Thought Process**

* Focused on ensuring smooth product creation and updates, while preventing duplicate entries.
* Verified how stock levels change when products are purchased or sold.
* Included scenarios to test bulk updates, ensuring large data handling works correctly.
* Checked how pricing changes impact sales and purchase calculations.
* Added a case for product deactivation, ensuring discontinued products cannot be sold.

**Sales Module**

**Core Functionality**

The Sales Module handles transactions between customers and the business. It ensures that stock is deducted correctly, payments are processed, discounts and taxes are applied, and invoices are generated.

**Thought Process**

* Ensured that sales transactions update stock correctly, preventing overselling.
* Verified that sales can only be made if stock is available.
* Checked how system handles simultaneous sales to prevent negative stock.
* Ensured the system enforces customer credit limits before allowing transactions.

**Sales Order Module**

**Core Functionality**

The Sales Order Module allows customers to place orders before they are invoiced. Unlike direct sales, sales orders do not immediately deduct stock, but they help businesses manage upcoming transactions.

**Thought Process**

* Ensured that creating a new sales order does not impact inventory.
* Verified that users can edit submitted sales orders before processing.
* Checked how the system handles expired or canceled orders.
* Ensured that submitted sales orders correctly appear in the sales order list.

**Purchase Module**

**Core Functionality**

The Purchase Module is responsible for procurement from suppliers. It ensures that stock is updated correctly when new products are received, supplier accounts are managed, and financial transactions are recorded accurately.

**Thought Process**

* Ensured that purchases correctly increase stock, preventing inventory miscalculations.
* Tested how the system handles duplicate purchase orders, ensuring no unintentional reorders.
* Added test cases for supplier management, ensuring blacklisted suppliers cannot be selected.
* Checked the purchase approval process, ensuring unauthorized purchases cannot proceed.
* Verified how tax and pricing updates impact purchase records.

**Purchase Return Module**

**Core Functionality**

The Purchase Return Module allows businesses to return goods to suppliers, process refunds, and adjust stock accordingly. This is essential for handling defective, expired, or unwanted inventory.

**Thought Process**

* Ensured that returns deduct stock correctly without affecting unrelated transactions.
* Verified that supplier balances update correctly after returns.
* Checked how the system prevents duplicate return requests.
* Tested returns for products under warranty and ensured proper tracking.