Industry Problem Statement

Inventory & Warehouse Management System (Python)

Business Background

A retail distribution company wants to build a Python-based system to manage its warehouse operations, including:

- Product master data
- Stock inwards and outwards
- Reorder level monitoring
- Inventory valuation

The system should reflect **real-world inventory workflows** used in warehouses and retail supply chains.

Task 1: Capture Product Details (Input Validation)

Objective

Collect and validate product information.

Requirements

Write a program to accept:

- Product ID
- Product Name
- Category (Electronics / Grocery / Apparel / Furniture)
- Unit Price

Business Rules

- Product name must not be empty

- Unit price must be greater than 0
- Category must be valid

Expected Outcome

Validated product master record.

Task 2: Initialize Opening Stock

Objective

Record starting inventory.

Requirements

- Accept opening stock quantity

Business Rules

- Quantity must be ≥ 0

Expected Outcome

Initial stock level recorded.

Task 3: Stock Inward Entry

Objective

Handle incoming inventory.

Requirements

- Accept inward quantity
- Update stock level

Business Rules

- Inward quantity must be positive

Formula

```
Updated Stock = Current Stock + Inward Quantity
```

Expected Outcome

Stock updated after purchase receipt.

Task 4: Stock Outward Entry

Objective

Handle outgoing inventory (sales/dispatch).

Requirements

- Accept outward quantity

Business Rules

- Outward quantity must be ≤ current stock

Formula

```
Updated Stock = Current Stock - Outward Quantity
```

Expected Outcome

Accurate stock reduction.

Task 5: Current Stock Status Check

Objective

Display live stock availability.

Output

- Product Name
- Available Quantity

- Stock Status (In Stock / Low Stock / Out of Stock)

Business Rules

- Low Stock if quantity < 10

Task 6: Reorder Level Monitoring

Objective

Prevent stockouts.

Requirements

- Accept reorder level

Business Rules

- If stock ≤ reorder level → reorder alert

Expected Outcome

Clear reorder notification.

Task 7: Inventory Valuation

Objective

Calculate total inventory value.

Formula

```
Inventory Value = Unit Price × Available Quantity
```

Expected Outcome

Accurate stock valuation for accounting.

Task 8: Stock Transaction Log

Objective

Maintain stock movement history.

Requirements

- Store inward and outward transactions in a list
- Each record should include:
  - Transaction type
  - Quantity
  - Date

Expected Outcome

Auditable transaction trail.

Task 9: Procedural Inventory Summary

Objective

Generate a quick inventory report.

Summary Should Include

- Product ID & Name
- Category
- Unit Price
- Available Stock
- Inventory Value

Task 10: Multiple Product Handling

Objective

Scale system for multiple SKUs.

Requirements

- Store multiple products in a dictionary
- Product ID as key

Task 11: Product Class Design (OOP)

Objective

Model product as a real-world object.

Create class `Product` with:

Attributes

- product_id
- name
- category
- unit_price
- stock_quantity

Task 12: Stock Management Methods

Objective

Encapsulate stock logic.

Methods

- add_stock()
- remove_stock()
- check_stock_status()

Task 13: Reorder Check Method

Objective

Automate reorder alerts.

Method

- check_reorder_level()

Task 14: Inventory Valuation Method

Objective

Encapsulate valuation logic.

Method

- calculate_inventory_value()

Task 15: Final Inventory Report Generation

Objective

Generate a professional inventory report.

Output Format (Example)

```
Product ID        : P210
Product Name      : Wireless Mouse
Category          : Electronics
Unit Price        : ₹1,200
Available Stock   : 25
Stock Status      : In Stock
Reorder Alert     : No
Inventory Value   : ₹30,000
```