Project Document: Restaurant Menu and Order System

Course: Programming For Business (Python)

Instructor: Riyan Athar
Objective: Develop a Python program to manage a restaurant's menu, process customer orders, track inventory, and generate sales reports.

Project Overview

This project simulates a restaurant management system where users (staff) can:

- Add, update, and remove dishes from the menu.
- Take customer orders and calculate bills.
- Track low-stock ingredients and apply discounts.
- Generate sales and inventory reports.

By completing this project, students will practice:

- Python data structures (lists, dictionaries).
- File handling (optional for data persistence).
- Functions, loops, and user input validation.

Project Requirements

1. Menu Management

Add New Dish

- Prompt the user to enter:
    - Dish name (e.g., "Margherita Pizza").
    - Category (Appetizer, Main Course, Dessert, Beverage).
    - Price (float).
    - Ingredients (list, e.g., ["flour", "cheese", "tomatoes"]).

- Store dishes in a structured format (e.g., list of dictionaries).

Update Dish

- Allow the user to:
    - Modify a dish’s price or ingredients.
    - Search for dishes by name or category.

Remove Dish

- Delete a dish from the menu after confirming its existence.

---

2. Order Processing

Take Orders

- Let the user input:
    - Table number (e.g., "Table 5").
    - Dishes ordered (with quantities).
- Calculate the total bill (including tax, e.g., 16%).

Low-Stock Alerts

- Flag dishes if any ingredient is "low stock" (e.g., <5 units remaining).
- Example:
    - Warning: "Margherita Pizza" is low on "cheese" (3 left)!

Discounts

- Apply discounts:
    - Seasonal (e.g., "10% off all Desserts in December").
    - Special promotions (e.g., "Buy 1 Appetizer, get 1 free").

---

3. Reporting & Analysis

Sales Report

- Display:
    - Total revenue.

    - Best-selling dishes.
    - Least popular dishes.

Inventory Report

- List all ingredients and their remaining quantities.
- Highlight items needing restocking.

Forecasting

- Predict ingredient demand for the next week based on current sales trends.

Project Deliverables

1. Source Code

- A single Python file (restaurant_system.py) with:
    - Functions for each feature (e.g., add_dish(), take_order()).
    - Clear comments explaining the logic.
    - Input validation (e.g., prevent negative prices).

2. Project Report (1–2 Pages)

- **Description**: Explain the system’s purpose and key features.
- **Sample Outputs**: Screenshots/text of:
    - Adding a dish.
    - Placing an order.
    - Generating a sales report.
- **Challenges & Solutions**: Briefly describe any coding hurdles and how you resolved them.

3. Bonus (Optional)

- **File Handling**: Save menu/orders to a .txt or .csv file.
- **GUI**: Use tkinter for a simple graphical interface.

Example Workflow

Staff Menu:

```
1. Add Dish    2. Update Dish    3. Remove Dish

4. Take Order  5. View Reports   6. Exit
```

Order Screen:

```
Table 3 Orders:

- 2x Margherita Pizza ($10.99 each)

- 1x Garlic Bread ($4.50)

Subtotal: $26.48

Tax (16%): $4.24

Total: $30.72
```

Grading Criteria

Task	Points
Menu Management	30
Order Processing	30
Reporting	20
Code Quality/Comments	10
Report/Output Samples	10
Total	100

Submission Instructions

1. Submit restaurant_system.py and the project report as a ZIP file.
2. Deadline: **05/12/2025.**

Notes for Students:

- Use dictionaries/lists to store dishes/orders.
- Test edge cases (e.g., invalid input, empty menu).
- Ask Instructor if stuck!