**Café Management System in Python**

* **Objective:**

The objective of this project is to build a simple and interactive **Café Management System** using Python that allows a user to: View a café menu. Place orders for multiple food items. Specify quantities for each item. Automatically calculate and display the bill. Summarize the order with total amount due. This system aims to simplify the ordering and billing process in small cafés or food outlets without requiring a graphical interface or database.

* **Key Features:**

1. **Interactive Menu Display:** Shows a list of available items with serial numbers and prices.
2. **Multi-Item Ordering:** Allows customers to order more than one item at a time.
3. **Custom Quantity Input:** Users can input how many units of each item they want.
4. **Order Storage in Dictionary:** Orders are dynamically stored in a dictionary for processing and billing.
5. **Real-Time Bill Generation:** Calculates subtotal per item and total amount of the entire order.
6. **Order Summary/Bill Printing:** Prints a clear breakdown of items, quantities, prices, and the grand total.

* **System Functionality (How It Works):**

**1. Display Menu:** The café menu is stored in a Python dictionary, with item IDs as keys and details like name and price as values.

menu = {

1: {"name": "Coffee", "price": 50},

2: {"name": "Tea", "price": 30},

...

}

A function display\_menu() prints the menu in a user-friendly format.

**2. Take Orders:** Users are asked to enter the item number and quantity. Input validation ensures the item exists.Orders are stored in a second dictionary order which accumulates item names, prices, and quantities.

order[item\_id] = {

"name": menu[item\_id]["name"],

"price": menu[item\_id]["price"],

"quantity": qty

}

**3. Generate Bill:** The display\_bill() function. Iterates through the order dictionary. Multiplies price by quantity for each item.Sums totals and prints an invoice-like summary.

**4. Exit Option:** The user can type 0 to finish the ordering process and generate the final bill.

**Sample Output Flow:**

Welcome to Python Café!

====== Café Menu ======

1. Coffee - ₹50

2. Tea - ₹30

3. Sandwich - ₹80

4. Burger - ₹120

5. Pasta - ₹150

6. Pizza - ₹250

========================

Enter item number to order (0 to finish): 2

Enter quantity for Tea: 3

Enter item number to order (0 to finish): 6

Enter quantity for Pizza: 1

Enter item number to order (0 to finish): 0

======= 🧾 Order Summary =======

Tea x 3 = ₹90

Pizza x 1 = ₹250

Total Amount: ₹340

Thank you for visiting our café! ☕

* **Possible Extensions / Future Enhancements:**
* GUI Interface: Add a graphical interface for better user experience.
* Database Integration: Store orders and menu items in a database for persistence.
* Login System: Separate cashier and admin roles.
* Dynamic Menu Management: Add/edit/delete items from the menu via admin panel.
* Export Bill to PDF/Print: Option to generate and print receipts.
* Inventory Tracking: Track stock levels of each item.
* Tax and Discount Handling: Add GST, service charge, and promo codes.
* **Conclusion:**

The **Café Management System in Python** is a beginner-friendly project that effectively demonstrates the use of: Python data structures like dictionaries. Input/output operations. Conditional logic and loops. Modular coding with functions. It can be used as a foundational step towards developing more advanced retail and point-of-sale (POS) applications.