

# **CAFE MANAGEMENT SYSTEM**

**(SWEET BLISS)**

**FULL PROJECT REPORT**

**Created By:**

**AYONIJA TRIPATHI**

**REG NO. - 25BCE10679**

# 1. Problem Description

Running a café or bakery requires managing several things such as:

- Menu management
- Taking multiple customer orders
- Billing
- Tracking sales analytics
- Managing expenses
- Storing customer feedback

Most small cafés still use manual paperwork which is slow, error-prone, and difficult to maintain.

**So the objective of this project is to create a simple, terminal-based Café Management System using Python that automates these tasks.**

The system uses **CSV files** for data storage and includes analytics features using **NumPy** and **Matplotlib**.

# 2. Abstract

This project is a **Python-based Café and Bakery Management System** designed to streamline café operations. It provides features like:

- Displaying menu and menu charts
- Taking multiple-item customer orders
- Auto-generating bills
- Maintaining total sales data
- Recording daily expenses
- Displaying sales analytics graphs

- Accepting customer feedback

The system is lightweight, console-based, and uses **CSV files** for persistent storage, making it easy to use and modify.

## 3. Project Features

### 1. Menu Management

- Loads menu data from `menu.csv`
- Displays menu in the terminal
- Shows menu as a horizontal bar graph using Matplotlib

### 2. Multi-Item Order System

- Customer can order any number of items in one order
- Automatically calculates prices
- Stores order entries is saved permanently in `orders.csv`

### 3. Bill Generator

- Generates a detailed bill with:
  - Order ID
  - Item names
  - Quantity
  - Line totals
  - Final total amount

### 4. Sales Analytics

Uses NumPy for:

- Total sales
- Average order value
- Maximum order value

Displays sales chart (item-wise quantities) using Matplotlib.

## 5. Expense Tracker

- Add expenses (butter, sugar, etc.)
- Store them in `expenses.csv`
- Compute total monthly expenses

## 6. Customer Feedback Module

Stores:

- Customer name
- Rating
- Feedback is saved in `feedback.csv`

# 4. Technology Stack

## Programming Language

- Python 3

## Libraries Used

- `csv` – file handling
- `numpy` – analytics & calculations
- `matplotlib` – graphs & visualizations

## Data Storage

- CSV files (`menu.csv`, `orders.csv`, `expenses.csv`, `feedback.csv`)

# 5. System Architecture

## Modules in the project

1. **main.py** – It shows the menu to the user when the program starts (1–9 options). It keeps the program running inside a loop until the user selects Exit.
2. **menu.py** – Reads the menu from `menu.csv`. Shows menu items in the terminal in neat format .Uses matplotlib to create a horizontal bar graph of menu items vs. their prices.
3. **orders.py** – Lets the customer order more than one item at the same time.
4. **bill.py** – Prints a clean and readable bill for the customer.
5. **analytics.py** – It tells you how much you earned and what sells the most.
6. **expenses.py** – It helps track how much money the café is spending.
7. **feedback.py** – It stores customer reviews so the café can improve later.

# 7. Screenshots / Outputs

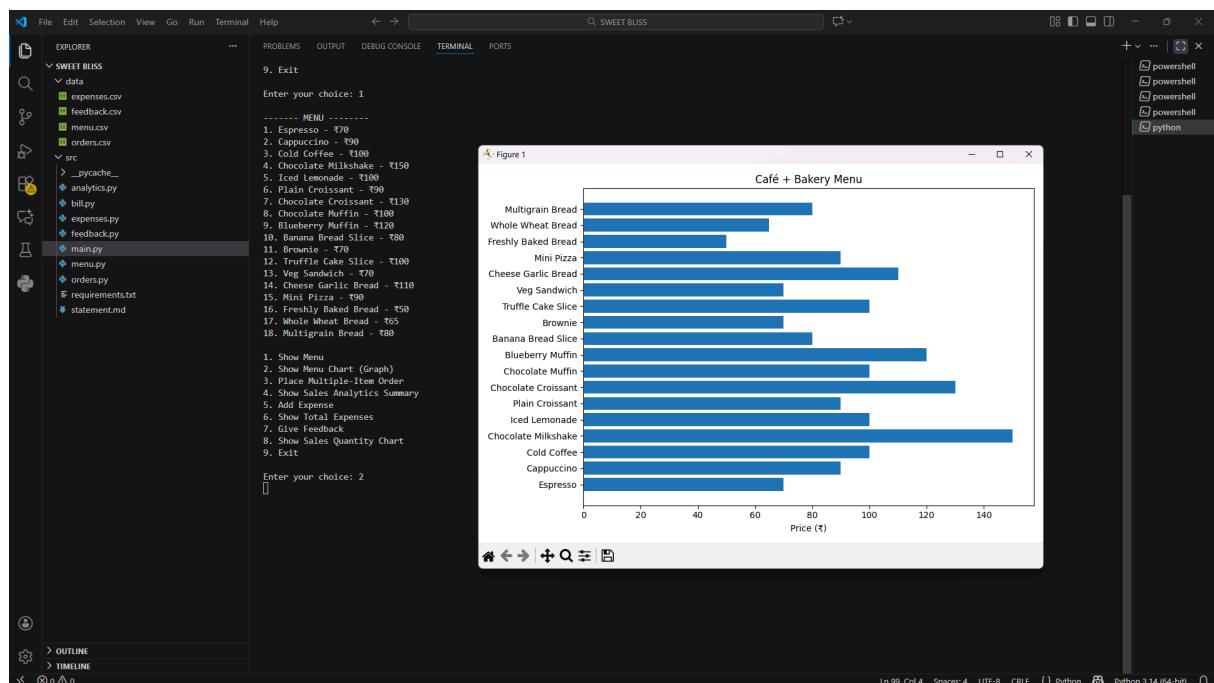
You will need to take screenshots from your terminal:

- Menu display

The screenshot shows a terminal window titled "SWEET BLISS". The terminal displays a menu for "WELCOME TO SWEET BLISS CAFE!" with various options numbered 1 through 9. Below the menu is a detailed list of items with their names and prices. The terminal prompt "Enter your choice: " is visible at the bottom.

```
----- WELCOME TO SWEET BLISS CAFE! -----  
1. Show Menu  
2. Show Menu Chart (Graph)  
3. Place Multiple-Item Order  
4. Show Sales Analytics Summary  
5. Add Expense  
6. Show Total Expenses  
7. Give Feedback  
8. Show Sales Quantity Chart  
9. Exit  
----- MENU -----  
1. Espresso - ₹70  
2. Cappuccino - ₹90  
3. Cold Coffee - ₹100  
4. Chocolate Milkshake - ₹150  
5. Iced Lemonade - ₹100  
6. Freshly Baked Bread - ₹90  
7. Chocolate Croissant - ₹130  
8. Chocolate Muffin - ₹100  
9. Blueberry Muffin - ₹120  
10. Banana Bread Slice - ₹80  
11. Brownie - ₹70  
12. Truffle Cake Slice - ₹100  
13. Veg Sandwich - ₹70  
14. Cheese Garlic Bread - ₹110  
15. Mini Pizza - ₹90  
16. Freshly Baked Bread - ₹50  
17. Whole Wheat Bread - ₹65  
18. Multigrain Bread - ₹80  
1. Show Menu  
2. Show Menu Chart (Graph)  
3. Place Multiple-Item Order  
4. Show Sales Analytics Summary  
5. Add Expense  
6. Show Total Expenses  
7. Give Feedback  
8. Show Sales Quantity Chart  
9. Exit  
Enter your choice: |
```

- Menu bar graph



- Order input & Bill Generation

```

File Edit Selection View Go Run Terminal Help ← → SWEET BLISS PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
SWEET BLISS
data
expenses.csv
feedback.csv
menu.csv
orders.csv
pycache_
analytics.py
bill.py
expenses.py
main.py
orders.py
requirements.txt
statement.md

Enter your choice: 3
Enter multiple items for this order.
Type 0 to finish adding items.

Item ID (0 to stop): 6
Quantity: 1

Item ID (0 to stop): 5
Quantity: 2

Item ID (0 to stop): 6
Quantity: 1

Item ID (0 to stop): 5
Quantity: 2

Item ID (0 to stop): 5
Quantity: 1

Item ID (0 to stop): 2
Quantity: 2

Item ID (0 to stop): 0
Order placed successfully!
----- BILL -----
Order ID : 137
-----
Flat Caramel Macchiato x2 - T$20
Iced Latte x2 - T$20
Cappuccino x1 - T$20
-----
TOTAL AMOUNT : T$80
-----


1. Show Menu
2. Show Menu Chart (Graph)
3. Place Multiple-Item Order
4. Show Sales Analytics Summary
5. Add Expense
6. Show Total Expenses
7. Give Feedback
8. Show Sales Quantity Chart
9. Exit

Enter your choice: 1

```

- Sales Summary

```

File Edit Selection View Go Run Terminal Help ← → SWEET BLISS PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
SWEET BLISS
data
expenses.csv
feedback.csv
menu.csv
orders.csv
pycache_
analytics.py
bill.py
expenses.py
main.py
orders.py
requirements.txt
statement.md

Enter your choice: 4
----- SALES SUMMARY -----
----- SALES SUMMARY -----
----- SALES SUMMARY -----
Total Sales Amount : T$18110
Total Sales Amount : T$18110
Average Order Value: T$1569.14
Highest Single Sale: T$15000
Highest Single Sale: T$15000

1. Show Menu
2. Show Menu Chart (Graph)
3. Place Multiple-Item Order
4. Show Sales Analytics Summary
5. Add Expense
6. Show Total Expenses
7. Give Feedback
8. Show Sales Quantity Chart
9. Exit

Enter your choice: 1

```

- Expense Adding

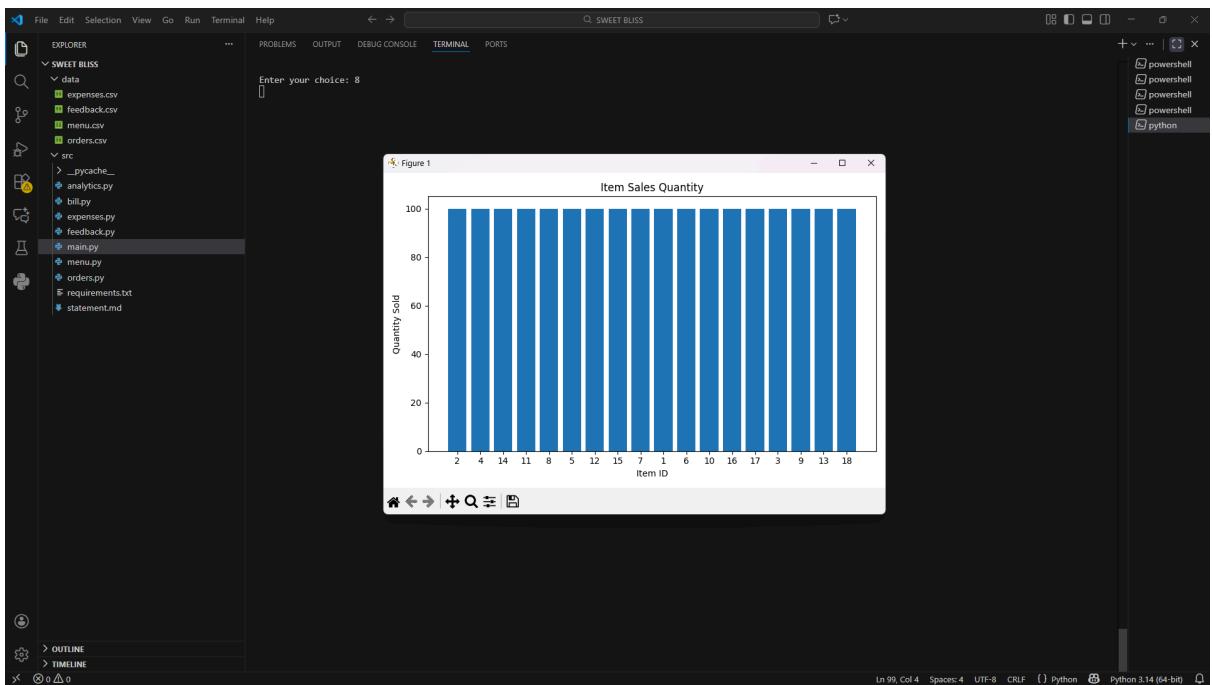
The screenshot shows the VS Code interface with the terminal tab active. The terminal window displays a Python script's interaction with the user:

```
1. Show Menu
2. Show Menu Chart (Graph)
3. Place Multiple-Item Order
4. Show Sales Analytics Summary
5. Add Expense
6. Show Total Expenses
7. Give Feedback
8. Show Sales Quantity Chart
9. Exit

Enter your choice: 5
Expense name: other
Amount: 500
Expense added.
```

The terminal also shows the script's menu again and a prompt for the user to enter a choice.

- Sales graph



- Feedback entry

```

SWEET BLISS
Customer name: Mehek
Rating (1-5): 5
Comments: please add more bread options
Thank you for your feedback!
1. Show Menu
2. Show Menu Chart (Graph)
3. Place Multiple-Item Order
4. Show Sales Analytics Summary
5. Add Expense
6. Show Total Expenses
7. Enter Feedback
8. Show Sales Quantity Chart
9. Exit
Enter your choice: 7

```

## 9. Conclusion

The Café Management System is an efficient tool for small cafés and bakeries to automate daily operations. It provides accurate billing, maintains sales and expense records, and supports decision-making using analytics and visualizations.

The project demonstrates good use of Python fundamentals, file handling, libraries, modular programming, and real-world application logic.

## 10. Future Enhancements

- Add GUI using Tkinter or PyQt5
- Add user login & admin dashboard
- Connect to SQL database instead of CSV
- Add PDF bill generation
- Add inventory management

- Add monthly financial reports

## 11. References

- Python official documentation
- NumPy documentation
- Matplotlib documentation
- Classroom materials