RESTAURANT MENU ORDERING SYSTEM

By Team: Slayers

Mentor: Garvit Jain

PROJECT DOCUMENTATION

Team Members

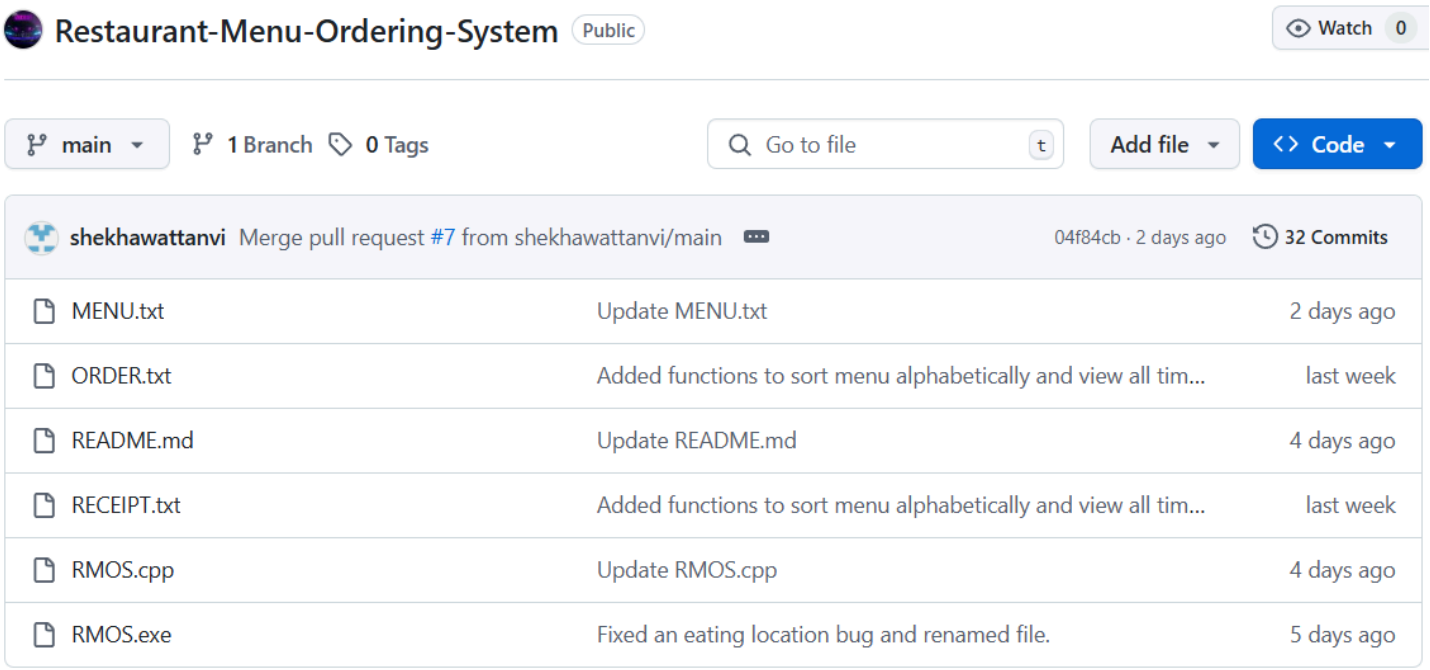
- Prince Kumar Gupta (EEE'28): Github LinkedIn
- Tanvi Shekhawat (CSE'28): Github LinkedIn
- Shiv Yadav (ECC'28): Github LinkedIn

GitHub Repository Required

Project: Github Repository

- C++ source file(s) (RMOS.cpp)
- Data files (MENU.txt)
- README.md with:
  - Project summary
  - How to run the program
  - Sample inputs/outputs

This encourages best practices in version control and organized code submission



Project Overview

This project is a simple food ordering system made in C++ using Visual Studio Code. It was created during the SPARK Summer Project and gives a basic restaurant-like experience through a console screen.

Users can view the menu, place or update orders, see their bills, and check order history. It uses basic programming ideas like classes, file storage, and neat code structure. This helped us learn how real apps work and how to build them step by step.

What it does:

- Displays full menu with Item ID, Name & Price.
- Allows user to place order from menu.
- Helps user to search & sort menu items (by price or alphabetical order).
- Lets admin securely add, modify & delete menu items.
- Generates receipts and saves all orders.

How does it work?

- View menu and place order with quantity selection using Item ID, Name & Price.
- Search menu items by name.
- Sort menu by price (low to high & high to low) or alphabetical order.
- Provide security to features like Add, Edit & Delete menu items.
- Add new items in menu with auto-generated ID.
- Remove items in menu using their ID.
- Modify an item's ID, name or price.
- Generate receipts and view full order history.

System Features

- OOP (Object-Oriented Programming): Classes with encapsulation.
- File Handling (ifstream, ofstream) Reading/Writing menu, orders, receipts & order history.
- Vectors to replicate menu items.
- Structures for clean organization of menu and order data.
- Sorting & Searching menu by price or alphabetical order.
- Loops and Conditional Statements: for, while, continue, break, nested if-else.
- Input Validation (empty, invalid or duplicate inputs).
- STL (vectors, sort functions with custom comparator logic).
- Timestamps (for bills using ctime header file)
- Password-protected admin features (add/edit/delete).
- Data storage in RECEIPT.txt and ORDER.txt.
- Clean & Modular Programming.
- Github Submission with README file and sample.

Key Features

- Main Menu

```
|| WELCOME TO RESTAURANT MENU ORDERING SYSTEM ||

Choose an option:
1) View menu and place order.
2) Search item in menu.
3) Sort menu (price low to high).
4) Sort menu (price high to low).
5) Sort menu (alphabetical order).
6) Settings.

Enter 1/2/3/4/5/6:
```

- MENU.txt: Browse through the menu before placing an order.

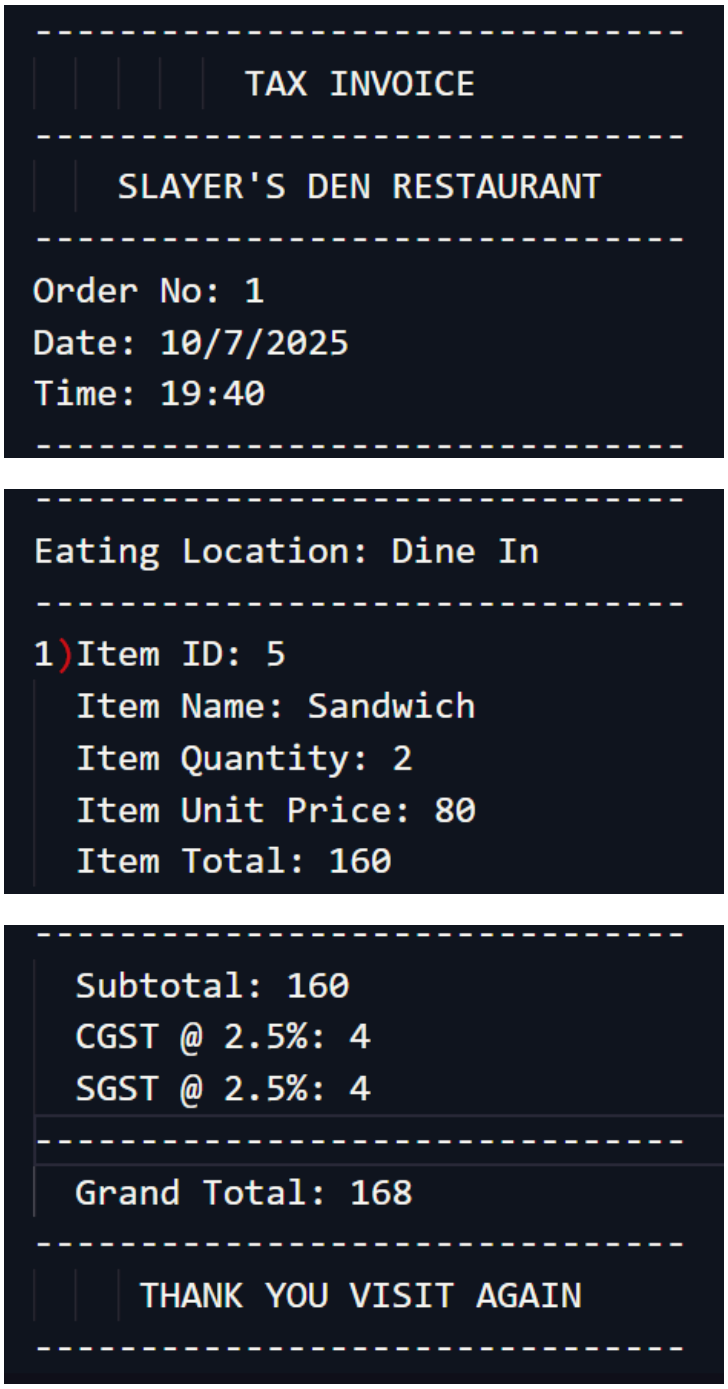
```
1   1    Water_Bottle        20
2   2    Cold_Coffee         60
3   3    Oreo_Shake          100
4   4    Mojito              80
5   5    Sandwich            80
6   6    French_Fries        60
7   7    Burger              80
8   8    Pizza               100
9   9    Pasta               90
10  10   Chowmein            80
11  11   Chole_Bhature       120
12  12   Pav_Bhaji           80
13  13   Dal_Makhani         180
```

14	14	Palak_Paneer	180
15	15	Sahi_Paneer	200
16	16	Malai_Kofta	200
17	17	Roti	15
18	18	Laccha_Parantha	40
19	19	Naan	50
20	20	Tandoori_Roti	40
21	21	Panipuri	20
22	22	Samosa	20
23	23	Kachori	15
24	24	Vadapav	30
25	25	Pyaaz_Kachori	30
26	26	Mirchi_Bada	20

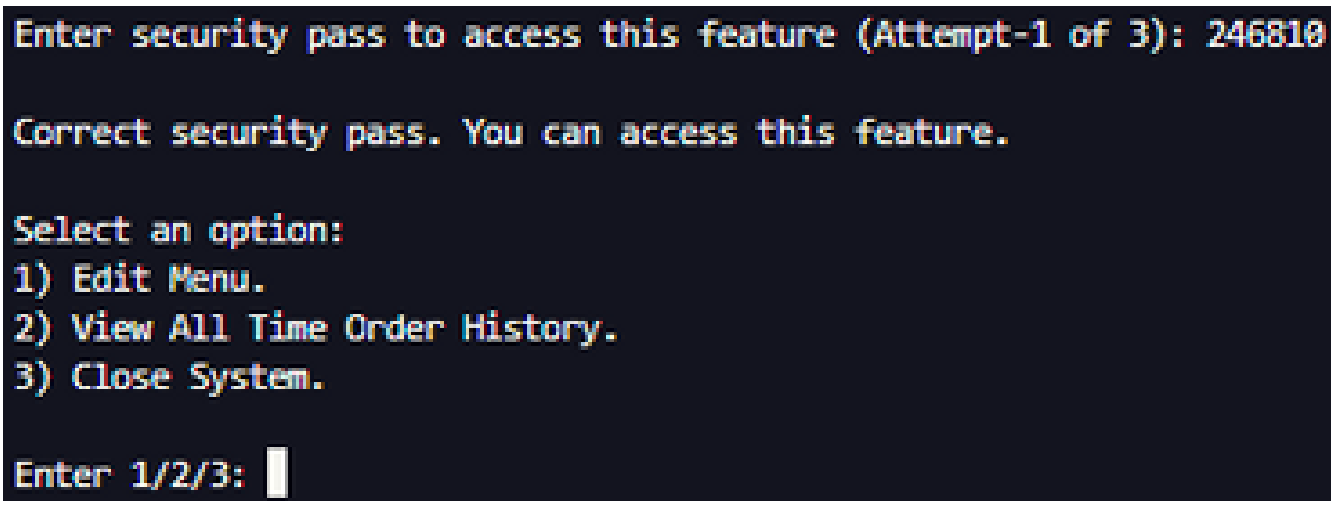
- Place an order: Provide Item ID, Quantity, and Eating Location to place an order.

```
Do you want to give order(Y/N): y

Give your order:

Enter eating location (in/out): in
Enter item ID: 5
Enter quantity of Sandwich : 2

Do you want to add more items? (y/n): n

Items added successfully. Do you want to confirm your order? (p to place order, c to cancel): p

Your order is successfully placed! Please wait a moment, your order is being prepared...
```

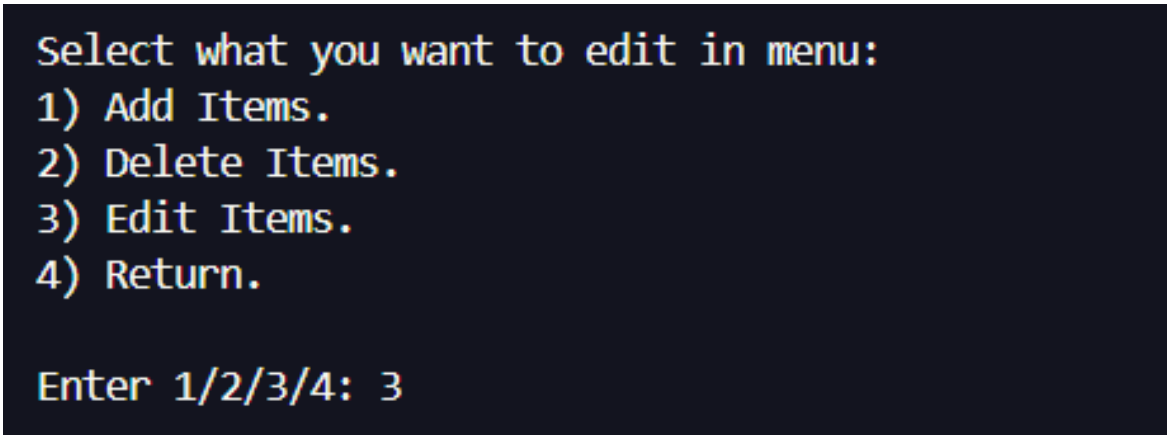
- RECEIPT.txt: Generate receipt for every order and save to text file.



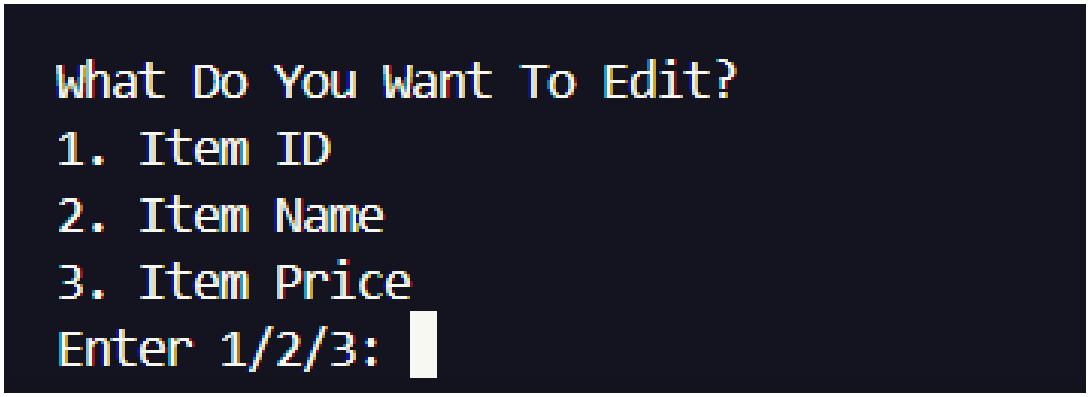
- Settings: To access this feature, enter correct security pass in 3 attempts.



- Modify menu: Add, Delete & Edit menu items.



- Edit Menu Items: Edit Item ID, Name & Price by using Item ID.



What did we learn?

We built a Restaurant Menu Ordering System as part of the SPARK Summer Program to simulate a digital food ordering experience through a C++ console application. The system lets users browse a categorized menu, place and manage orders, generate bills, and track past orders. It's designed using core programming concepts like object-oriented principles, file handling, and modular architecture

This project offered practical exposure to core C++ programming through real-world problem-solving. Key learnings include:

- How to design a user-friendly, menu-driven system using C++
- Practical implementation of file handling for saving and retrieving data

- Use of object-oriented principles to keep the code clean and modular
- Importance of team collaboration, version control, and clear documentation

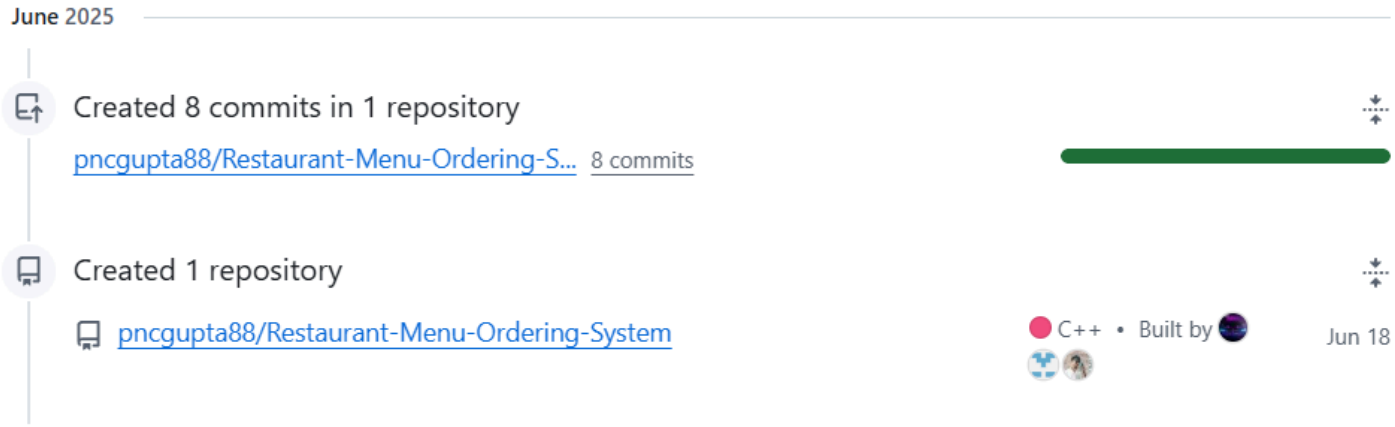
Purpose

This project was inspired by the need to make restaurant operations smoother—especially in places where handwritten orders and manual billing are still used. We wanted a simple and efficient system that could manage food items, handle orders, and produce error-free bills, all in one place.

Contributions

Member 1: Prince Kumar Gupta

- Developed Main Menu along with functions to:
  - View menu and place order
  - Sort menu by:
    - Price (Low to High)
    - Price (High to Low)
    - Alphabetical Order
  - Add new items to menu
  - Generate RECEIPT.txt
  - View all time order history from ORDER.txt
- Made README.md

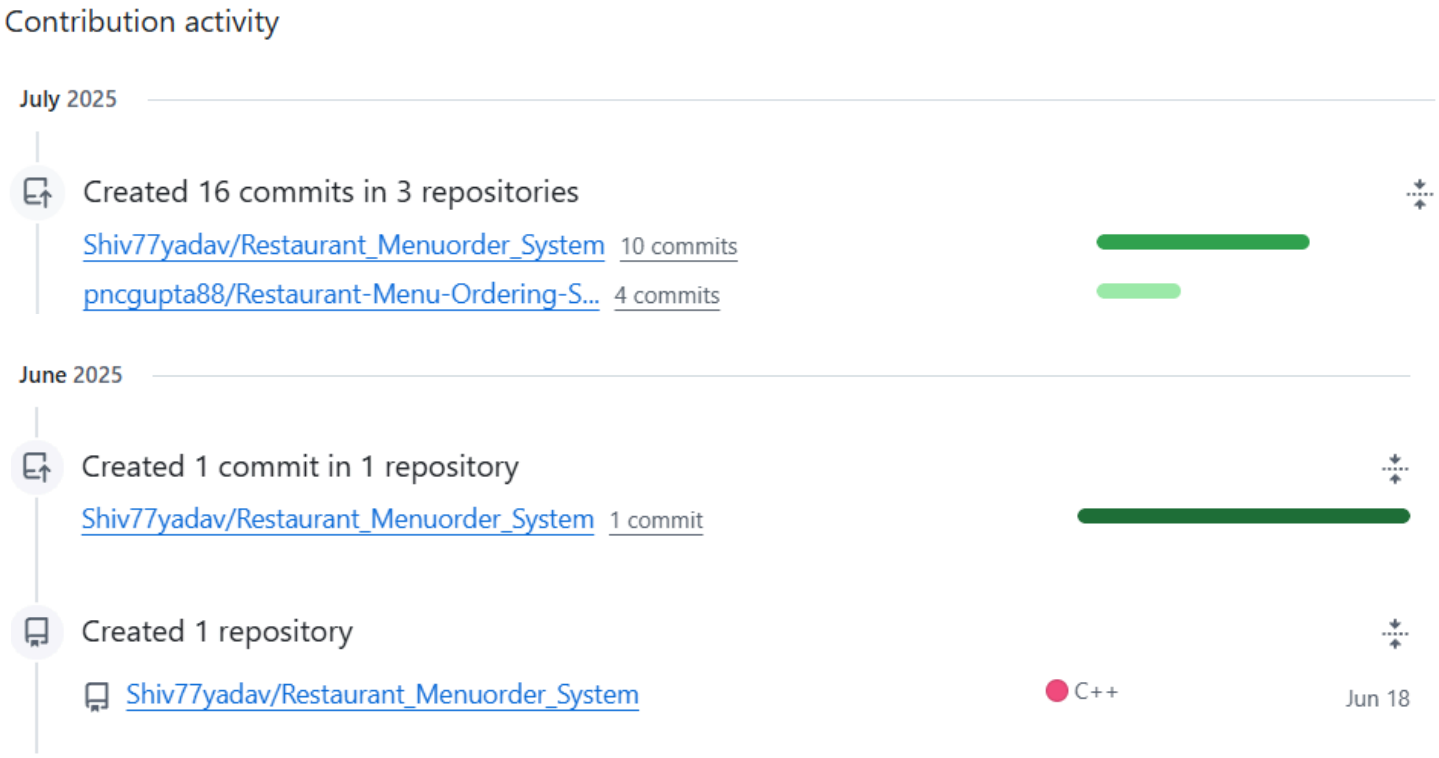


Member 2: Tanvi Shekhawat

- Developed the code for deleting menu items by using item ID with input validation and duplicate checks.
- Developed the code for modifying menu items with input validation and duplicate checks.
- Made MENU.txt with Item ID, Name & Price.
- Made Project Report.

Member 3: Shiv Yadav

- Developed the code to Search items in menu.
- Made README.md.
- Made Project Report.



Future scope

In future versions, this system can be extended with:

- It can be developed into a full-fledged website or mobile app for cafés or restaurants
- With an easy-to-use design and clear setup, this system can really help small restaurants and cafés shift to digital ordering and billing in a smooth way
- Real-time menu updates, table-wise order tracking, and instant billing can enhance its usefulness
- A backend powered by MySQL or Firebase can manage menu items, orders, and reports efficiently

Tasks

Task 1: System Planning & Structure Design

- Requirement Analysis
- Class Diagram Design
- Function Listing
- Defining Data Members and Member Functions

Task 2: Product Management Module

- Class with constructor/destructor
- File read/write (binary or text)
- Input validation
- Display product list from file

Task 3: Customer & Billing Module

- Customer and Invoice class design
- Bill generation logic
- File handling for invoice storage
- Basic mathematics (totals, discounts, etc.)

Task 4: Final Integration & Menu System

- Switch-case or loop-based menu
- Smooth flow from one module to another
- Code organization and commenting
- Final testing and debugging

Task 5: Documentation & Submission

- Final project polish
- Documentation
- Proper code indentation
- Screenshots (optional but recommended)