PROJECT REPORT

ONLINE FOOD ORDERING SYSTEM

BY Yash Dhondiyal

SAP ID:- 590024289

INTRODUCTION

The **Online Food Ordering System** is a console-based application developed in the C programming language.
The project simulates a basic restaurant ordering experience where users can:

- View menu items
- Add items to a cart
- View the cart at any time
- Generate a final bill with GST
- Exit the system

It uses several fundamental concepts of C programming such as:

- Loops
- Conditionals
- Functions
- Structures
- Arrays
- Strings
- Basic arithmetic
- Modular program design

OBJECTIVES

The main objectives of this project are:

1. To design a simple command-line food ordering interface.
2. To practice the use of **structures**, **arrays**, and **functions**.
3. To implement a cart system that stores ordered items.
4. To calculate subtotal, GST, and final payable amount.
5. To demonstrate structured & modular program design.
6. To enhance problem-solving skills in C.

PROJECT FEATURES

Menu Display

Shows the list of available food items, their IDs, and price.

Add Item to Cart

User selects an item using its ID and specifies quantity.

View Cart

Shows all added items along with calculated totals.

Checkout

Calculates:

- Subtotal
- GST (5%)
- Final payable amount

Exit

Gracefully exits the application.

DATA STRUCTURES USED

Structure: Food

Stores menu details:

```
struct Food {
    int id;
    char name[50];
    float price;
};
```

Structure: CartItem

Stores items added to cart:

```
struct CartItem {
    char name[50];
    float price;
    int quantity;
};
```

Why Structures?

Because food items and cart entries contain multiple types of data (string + float + int). Structures help group them logically.

FUNCTIONAL MODULES

➤ displayMenu()

Prints the food menu.

➤ addToCart(int id, int qty)

Adds selected item + quantity into the cart array.

➤ viewCart()

Displays all ordered items with calculated item totals.

➤ checkout()

Calculates:

- Subtotal
- GST = 5% of subtotal
- Total bill

➤ main()

Controls program flow using a menu-driven loop.

FLOW OF PROGRAM

1. Start program
2. Display main menu
3. User selects choice
4. According to choice:
    - Show menu
    - Add to cart
    - View cart
    - Checkout
5. Loop continues until user chooses to exit

CODE LISTING

Users > yash > Desktop > C Yash_590024289_c_project.c > addToCart(int, int)

```c
#include <stdio.h>
#include <string.h>
#define MAX_ITEMS 50
struct Food {
    int id;
    char name[50];
    float price;
};
struct CartItem {
    char name[50];
    float price;
    int quantity;
};
struct Food menu[] = {
    {1, "Pizza", 199},
    {2, "Burger", 99},
    {3, "Pasta", 149},
    {4, "French Fries", 79},
    {5, "Cold Coffee", 89}
};
int menuSize = 5;
struct CartItem cart[MAX_ITEMS];
int cartCount = 0;
void displayMenu() {
    printf("\n--------- MENU ----------\n");
    for (int i = 0; i < menuSize; i++) {
        printf("%d. %s - Rs %.2f\n", menu[i].id, menu[i].name, menu[i].price);
    }
    printf("-------------------------\n");
}
void addToCart(int id, int qty) {
    if (id < 1 || id > menuSize) {
        printf("Invalid item ID!\n");
        return;
    }
    strcpy(cart[cartCount].name, menu[id - 1].name);
    cart[cartCount].price = menu[id - 1].price;
    cart[cartCount].quantity = qty;
    cartCount++;
    printf("%s (x%d) added to cart!\n", menu[id - 1].name, qty);
}
void viewCart() {
    if (cartCount == 0) {
        printf("\nYour cart is empty!\n");
        return;
    }
    printf("\n--------- YOUR CART ---------\n");
    float subtotal = 0;
    for (int i = 0; i < cartCount; i++) {
        float itemTotal = cart[i].price * cart[i].quantity;
        printf("%s (x%d) - Rs %.2f\n", cart[i].name, cart[i].quantity, itemTotal);
        subtotal += itemTotal;
    }
```

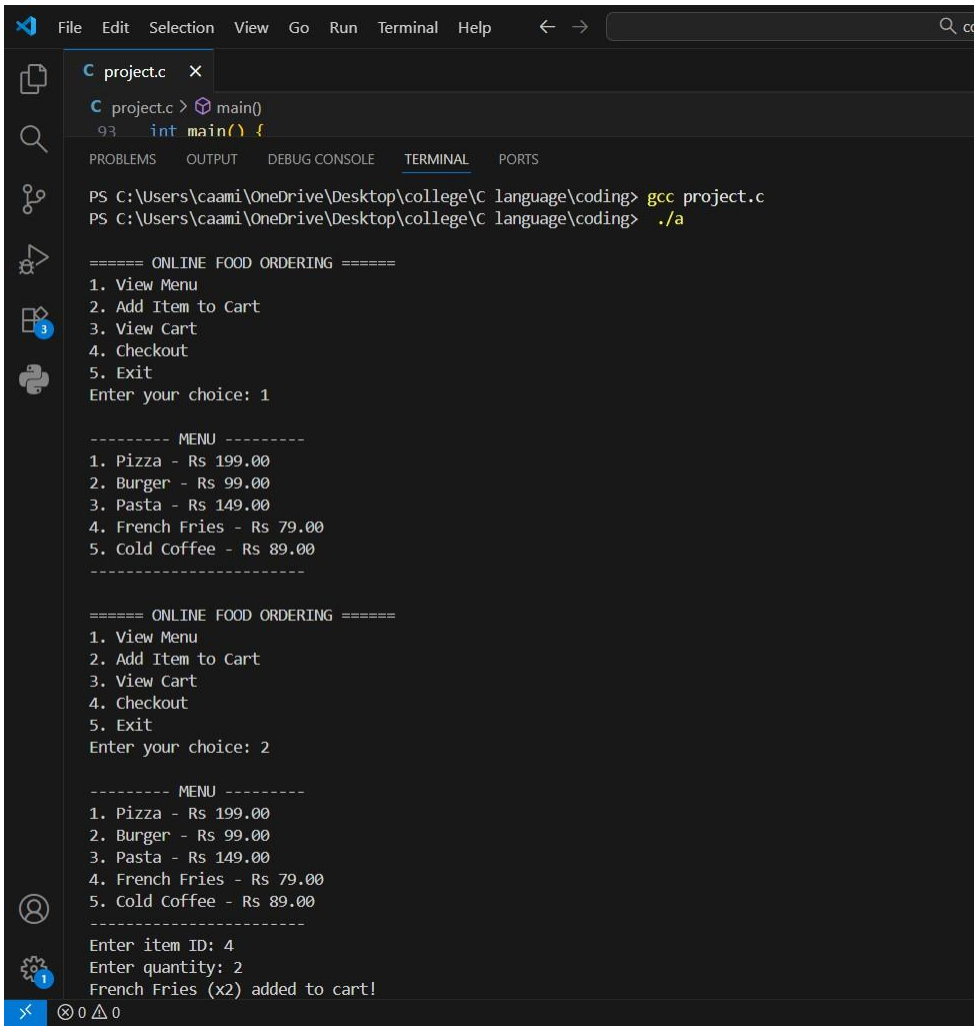
Users > yash > Desktop > C Yash_590024289_c_project.c > addToCart(int, int)

```c
void viewCart() {
    for (int i = 0; i < cartCount; i++) {

    printf("-----------------------------\n");
    printf("Subtotal: Rs %.2f\n", subtotal);
}
void checkout() {
    if (cartCount == 0) {
        printf("\nCart is empty! Cannot checkout.\n");
        return;
    }
    float subtotal = 0;
    for (int i = 0; i < cartCount; i++) {
        subtotal += cart[i].price * cart[i].quantity;
    }
    float gst = subtotal * 0.05;   // 5% GST
    float total = subtotal + gst;
    printf("\n--------- BILL ----------\n");
    printf("Subtotal: Rs %.2f\n", subtotal);
    printf("GST (5%%): Rs %.2f\n", gst);
    printf("TOTAL: Rs %.2f\n", total);
    printf("-------------------------\n");
    printf("\nThank you for ordering!\n");
}
int main() {
    int choice, id, qty;
    while (1) {
        printf("\n===== ONLINE FOOD ORDERING =====\n");
        printf("1. View Menu\n");
        printf("2. Add Item to Cart\n");
        printf("3. View Cart\n");
        printf("4. Checkout\n");
        printf("5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch (choice) {
            case 1:
                displayMenu();
                break;
            case 2:
                printf("Enter item ID: ");
                scanf("%d", &id);
                printf("Enter quantity: ");
                scanf("%d", &qty);
                addToCart(id, qty);
                break;
            case 3:
                viewCart();
                break;
            case 4:
                checkout();
                break;
            case 5:
```

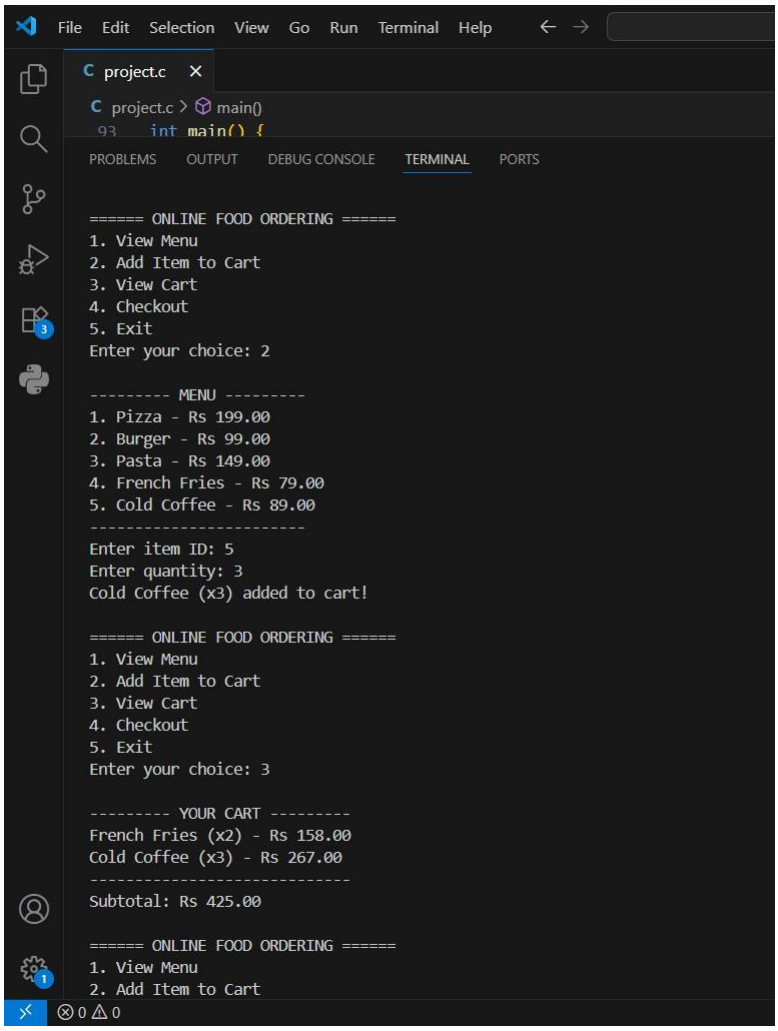
```c
int main() {
    int choice, id, qty;
    while (1) {
        printf("\n===== ONLINE FOOD ORDERING =====\n");
        printf("1. View Menu\n");
        printf("2. Add Item to Cart\n");
        printf("3. View Cart\n");
        printf("4. Checkout\n");
        printf("5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch (choice) {
            case 1:
                displayMenu();
                break;
            case 2:
                printf("Enter item ID: ");
                scanf("%d", &id);
                printf("Enter quantity: ");
                scanf("%d", &qty);
                addToCart(id, qty);
                break;
            case 3:
                viewCart();
                break;
            case 4:
                checkout();
                break;
            case 5:
                printf("Exiting...\n");
                return 0;
            default:
                printf("Invalid choice! Try again.\n");
        }
    }
    return 0;
}
```

SAMPLE OUTPUT


File Edit Selection View Go Run Terminal Help
project.c
project.c > main()
int main() {
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\caami\OneDrive\Desktop\college\C language\coding> gcc project.c
PS C:\Users\caami\OneDrive\Desktop\college\C language\coding> ./a
====== ONLINE FOOD ORDERING ======
1. View Menu
2. Add Item to Cart
3. View Cart
4. Checkout
5. Exit
Enter your choice: 1
--------- MENU ---------
1. Pizza - Rs 199.00
2. Burger - Rs 99.00
3. Pasta - Rs 149.00
4. French Fries - Rs 79.00
5. Cold Coffee - Rs 89.00

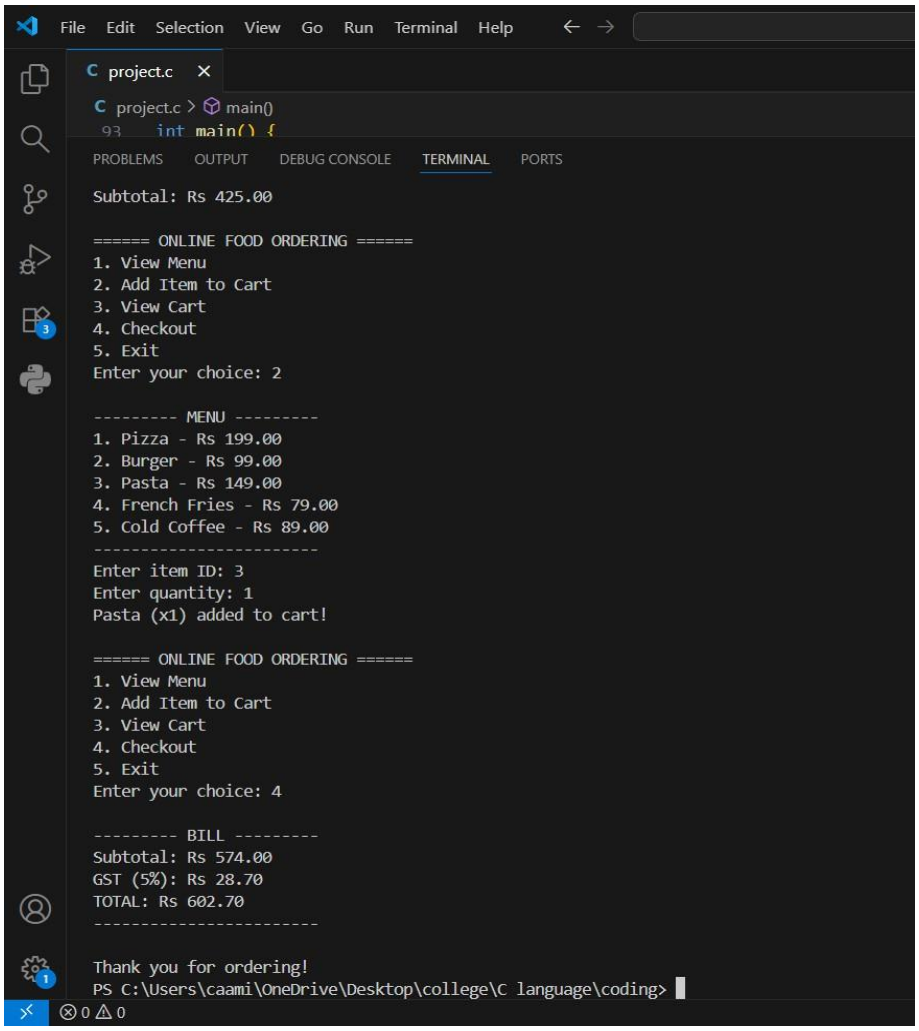
====== ONLINE FOOD ORDERING ======
1. View Menu
2. Add Item to Cart
3. View Cart
4. Checkout
5. Exit
Enter your choice: 2
--------- MENU ---------
1. Pizza - Rs 199.00
2. Burger - Rs 99.00
3. Pasta - Rs 149.00
4. French Fries - Rs 79.00
5. Cold Coffee - Rs 89.00

Enter item ID: 4
Enter quantity: 2
French Fries (x2) added to cart!
0 0


File Edit Selection View Go Run Terminal Help
project.c
project.c > main()
93 int main() {
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
====== ONLINE FOOD ORDERING ======
1. View Menu
2. Add Item to Cart
3. View Cart
4. Checkout
5. Exit
Enter your choice: 2
--------- MENU ---------
1. Pizza - Rs 199.00
2. Burger - Rs 99.00
3. Pasta - Rs 149.00
4. French Fries - Rs 79.00
5. Cold Coffee - Rs 89.00

Enter item ID: 5
Enter quantity: 3
Cold Coffee (x3) added to cart!
====== ONLINE FOOD ORDERING ======
1. View Menu
2. Add Item to Cart
3. View Cart
4. Checkout
5. Exit
Enter your choice: 3
--------- YOUR CART ---------
French Fries (x2) - Rs 158.00
Cold Coffee (x3) - Rs 267.00

Subtotal: Rs 425.00
====== ONLINE FOOD ORDERING ======
1. View Menu
2. Add Item to Cart
0 0



RESULT

The program successfully simulates a simple online food ordering system where users can:

- Select food items
- Add multiple items
- Manage their cart
- Generate bill with GST

It works logically and accurately with user-friendly prompts.

CONCLUSION

This Online Food Ordering System demonstrates how C programming concepts can be applied to build real-world style applications.
The program is modular, readable, and effectively uses:

- Structures
- Arrays
- Functions
- Loops
- Conditional statements