Subject: C Programming

FINAL PROJECT

RESTAURANT MANAGEMENT SYSTEM

BTech CSE 1st Year (2025-2026)

Batch: - 68

Name & SAP_ID

Shivani_590028517

Khyati_590028513 Bharadwaj

Faculty:-
Vinod Kumar

Date of submission
28/11/25

Abstract

This project implements a fully functional Restaurant Management System using the C programming language.
The purpose of this system is to automate essential restaurant operations such as menu management, order placement, availability checking, table assignment, occasion booking, and bill generation. Customers can view categorized menu items, check availability status, order multiple dishes, and receive system-generated suggestions if a selected item is unavailable. The system also supports function bookings, allowing the user to input event details, apply decoration charges, and add custom menu items for special occasions. This project demonstrates fundamental programming concepts including:

- **Array manipulation**
- **Modular design using functions**
- **Conditional branching**
- **Loops**
- **Character string operations**
- **Structured billing calculations**

Problem Definition

Restaurants frequently need a simple, reliable, and fast computerized system to manage orders and generate bills. Manual systems often lead to errors, delays, and difficulty in managing item availability. Therefore, the following key requirements were identified:

System Requirements:

1. **Display menu items categorized as:**

   Fast Food, Beverages ,Thalis

2. **Check the availability of each menu item before order confirmation.**
3. **Provide automatic suggestions for alternative dishes if the selected item is unavailable.**
4. **Accept multiple orders with specified quantities.**
5. **Generate a detailed bill, including:**

   *** Total amount**

   *** CGST (9%)**

   *** SGST (9%)**

   *** Final payable amount**

6. **Record essential customer information:**

    * **Name**

    * **Contact number**

    * **Address**

7. **Automatically assign a table using a simple hashing technique.**

8. **Handle special functions or events, including:**

    * **Event date and time**

    * **Number of attendees**

    * **Decoration charges**

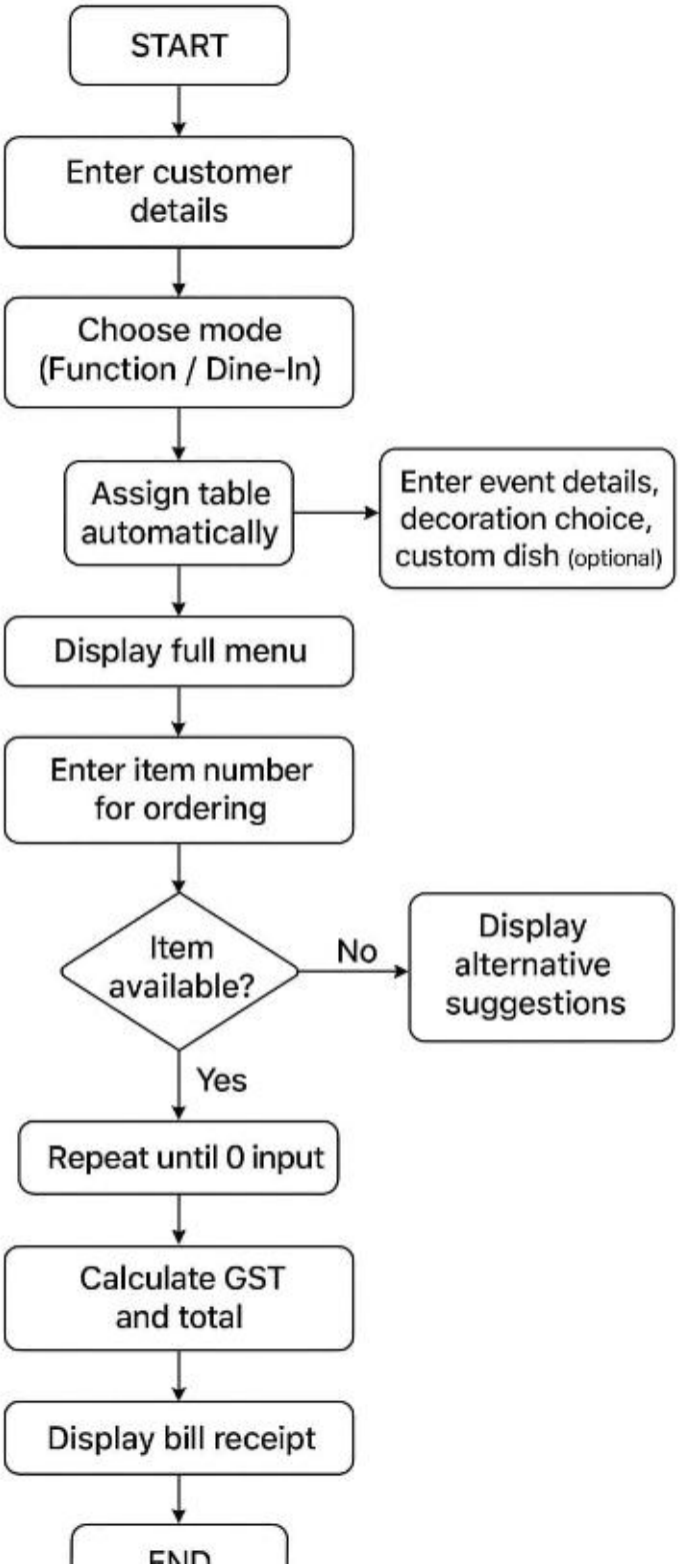
    * **Custom special dishes**

Need for the Project

Most beginner-friendly C projects lack real-world scenarios. This system aims to bridge that gap by providing a structured, easy-to-understand solution that implements actual restaurant processes without databases or external libraries.

System Design

Flowchart



Algorithm

Step 1: Define arrays for items, categories, availability, prices, and ordered quantities.

Step 2: Accept customer details using formatted input operations.

Step 3: Ask the customer to select between dine-in or event booking.

Step 4: * If booking a function:

- **Take date, time, people count, decoration preference, and optional custom dish.**
- **Assign table number using assignTable().**

Step 5: Call showMenu() to display menu items.

Step 6: Start order-taking loop:

- **Accept item number**
- **If unavailable → call suggestAlternatives()**
- **Else accept quantity and update bill**

Step 7: Compute taxes:

- **CGST = 9% and SGST = 9%**

Step 8: Display complete bill with all order details

Step 9: End program.

Implementation Details

a) Menu Display

This function categorizes and prints menu items with their availability in a neatly formatted structure.

```
void showMenu(char items[][30], int price[], int available[], int category[], int N){
    for(int i = 0; i < N; i++){
        printf("%2d. %-20s Rs %3d %s\n",
            i+1,
            items[i],
            price[i],
            available[i] ? "(Available)" : "(Not Available)"
        );
    }
}
```

b) Table Allocation

A simple hashing technique is used to generate table numbers dynamically.

```
int assignTable(char name[], char address[]){
    int len1 = strlen(name);
    int len2 = strlen(address);
    return (len1 + len2) % 20 + 1;   // ensures table number between 1 and 20
}
```

c) Alternative Suggestions

If an ordered item is unavailable, this function finds up to 5 alternatives from the same category first, then from others.

```
void suggestAlternatives(int index, char items[][30], int available[],
                         int category[], int price[], int N){
    for(int i=0; i<N; i++){
        if(i != index && available[i] && category[i] == category[index]){
            printf(" - %s (Rs %d)\n", items[i], price[i]);
        }
    }
}
```

d) GST Billing

GST is calculated using fixed percentages

```
float cgst = total * 0.09f;
float sgst = total * 0.09f;
float finalBill = total + cgst + sgst;
```

Testing & Result

Test Case 1: Ordering Available Item

Input: Pizza, quantity = 2
Expected Output: Correct computation of item amount
Result: Successful

Test Case 2: Unavailable Item

Input: Burger (marked unavailable)
Expected Output: Display of suggested alternatives
Result: The system displayed relevant suggestions based on category

---

Test Case 3: Special Occasion Mode

Input: Occasion = Yes

Special dish name = "Cake"

Price = 500

Output = Cake is 19^{th} item in menu

Result: Successful

Test Case 4: GST Accuracy

Input: Total = 300

Expected Output:

```
CGST = 27
SGST = 27
Final = 354
```

Result: Correct calculation

Overall Result

All major features—menu display, ordering, table assignment, event booking, custom items, and billing—performed correctly and consistently during testing.

Problems Faced By Our Group

1. No Option to Modify or Cancel Order

Once the order is entered, the user cannot change quantity or remove item.If they type something wrong, they must restart the program.

2. Fixed Table Allocation Rules

Table is allotted strictly based on number of people (e.g., 3 people = table of 4).No option to choose preferred location (window seat, AC, non-AC, outdoor, etc.)

3. No Error Handling for Wrong Input

If the user enters characters instead of numbers, the program may behave unexpectedly.Example: entering "a" instead of "2" may cause loop errors or wrong output.

4. System Does Not Support Multiple Orders at Once

Only one order can be handled per program execution.Real restaurants process multiple table orders simultaneously.

5. No Permanent Record

All bookings and orders are erased once the program ends.No history for returning customers.

Conclusion

The Restaurant Management System developed in C successfully demonstrates a basic yet effective way to automate restaurant operations. This project allows the staff to manage menu display, check item availability, suggest alternatives for unavailable dishes, process orders, handle special occasions, assign tables automatically, and generate detailed bills including applicable taxes (CGST and SGST).

The system showcases the practical use of core C programming concepts such as arrays, loops, conditional statements, functions, and string manipulation. Additionally, the modular structure of the program ensures that each task is handled by a separate function, making the code easier to maintain and extend.

Future Work

File Handling and Database Integration

Implement persistent storage to save customer orders, bills, and inventory data, which allows retrieval for future referenceand analytics.

Admin Login and Authentication

Introduce secure login for restaurant staff to manage menu, check table availability, and modify item details.

Inventory Management

Automatically update stock levels when items are ordered and provide notifications for low inventory.

Real-Time Table Tracking

Implement a dynamic table management system to track table availability and reservations in real time.

Graphical User Interface (GUI)

Replace the console-based interface with a graphical interface to make the system more interactive and visually appealing.

Snip of Code

```c
#include <stdio.h>
#include <string.h>

// ======== FUNCTION DECLARATIONS ========
void showMenu(char items[][30], int price[], int available[], int category[], int N);
int assignTable(char name[], char address[]);
void suggestAlternatives(int index, char items[][30], int available[], int category[], int price[], int N);

// ======================================
//              MAIN
// ======================================
int main() {
    int N = 18;

    char items[18][30] = {
        "Pizza","Burger","Pasta","Momos","French Fries","Sandwich","Hot Dog","Cold Drink",
        "Coffee","Tea","Lassi","Soda","Veg Thali","Non-Veg Thali","Paneer Thali","Chole Bhature",
        "Rajasthani Thali","Masala Dosa"
    };

    int category[18] = {
        0,0,0,0,0,0,0,2,   // first 8 = fast food (0) except "Cold Drink" labelled 2 for beverage
        2,2,2,2,           // four beverages = 2
        1,1,1,1,           // four thalis = 1
        1,0                // "Rajasthani Thali" = 1, "Masala Dosa" treated as fast food = 0
    };

    int available[18] = {
        1,0,1,1,1,1,0,1,
        1,1,0,1,
        1,1,1,0,
        0,1
```

```c

    int price[18] = {
        180,90,140,110,70,80,85,40,
        60,30,80,40,150,200,180,120,
        250,90
    };

    int orderedQty[18];
    for(int i = 0; i < N; i++) {
        orderedQty[i] = 0;
    }

    // ===== CUSTOMER DETAILS =====
    char name[50], address[100], contact[20], date[20], timeSlot[20];
    printf("Enter Customer Name: ");
    scanf(" %[^\n]", name);

    printf("Enter Address: ");
    scanf(" %[^\n]", address);

    printf("Enter Contact Number: ");
    scanf("%s", contact);

    // ===== CHOOSE MODE =====
    int mode;
    printf("\nDo you want to (1) BOOK the restaurant for a function OR (2) Just DINE-IN? Enter 1 or 2: ");
    scanf("%d", &mode);

```

```c
    int people = 0;
    float perPlateRate = 0.0f;
    int tableNumber = 0;
    int addedCustomItem = 0;
    char customItem[30] = "";
    int customPrice = 0;
    int decoration = 0;
    float decorCharge = 0.0f;

    switch(mode) {
        case 1:
            // Booking mode
            printf("Enter Date of function (DD/MM/YYYY): ");
            scanf("%s", date);
            printf("Enter Time of function (HH:MM): ");
            scanf("%s", timeSlot);

            printf("For how many people will attend? ");
            scanf("%d", &people);

            printf("Do you want decoration/themes? (yes=1 / no=0): ");
            scanf("%d", &decoration);
            if(decoration == 1) {
                printf("Decoration charge: Rs 2000 flat + Rs 150 per plate theme extra.\n");
                decorCharge = 2000 + (150 * people);
            }

            printf("Enter Per Plate Rate for the full course (Rs): ");
            scanf("%f", &perPlateRate);
```

```c
            printf("Do you want to add any special dish for this occasion? (yes=1 / no=0): ");
            scanf("%d", &addedCustomItem);
            if(addedCustomItem == 1) {
                printf("Enter Special Item Name: ");
                scanf(" %[^\n]", customItem);
                printf("Enter Price of this special item: ");
                scanf("%d", &customPrice);
            }

            tableNumber = assignTable(name, address);
            printf("\n>> Booking Confirmed!\n");
            printf("Date: %s   Time: %s\n", date, timeSlot);
            printf("People: %d   Table No: %d\n", people, tableNumber);
            if(decoration == 1) {
                printf("Decoration charge: Rs %.2f\n", decorCharge);
            }
            printf("Per Plate Course Rate: Rs %.2f\n", perPlateRate);
            if(addedCustomItem) {
                printf("Special Dish: %s at Rs %d\n", customItem, customPrice);
            }
            printf("---------------------------------\n");
            break;

        case 2:
            // Dine-in mode
            printf("Enter Date (DD/MM/YYYY): ");
            scanf("%s", date);
```

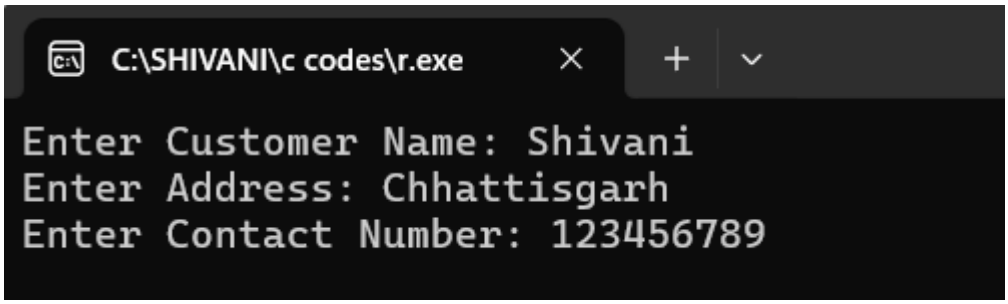
```
            scanf("%s", date);
            printf("Enter Time (HH:MM): ");
            scanf("%s", timeSlot);

            tableNumber = assignTable(name, address);
            printf("\n>> Table Booked for Dine-In\n");
            printf("Date: %s   Time: %s\n", date, timeSlot);
            printf("Table No: %d\n", tableNumber);
            printf("-----------------------------------\n");
            break;

        default:
            printf("Invalid mode selected. Exiting.\n");
            return 1;
    }

    // ===== SHOW MENU =====
    showMenu(items, price, available, category, N);
    if(addedCustomItem && mode == 1) {
        printf("\n--- SPECIAL OCCASION ITEM ---\n");
        printf("   19. %-20s Rs %d (Available)\n", customItem, customPrice);
    }

    // ===== ORDERING =====
    int choice, qty;
    float total = 0.0f;
    while(1) {
        printf("\nEnter item number to order (0 to finish): ");
        scanf("%d", &choice);

        if(choice == 0) break;
```

```
    // ===== BILLING =====
    if(mode == 1) {
        float bookingCharge = perPlateRate * people;
        total = bookingCharge;
        if(decoration == 1) total += decorCharge;
        if(addedCustomItem) total += (customPrice);  // assuming one unit
    }

    float cgst = total * 0.09f;
    float sgst = total * 0.09f;
    float finalBill = total + cgst + sgst;

    printf("\n=========================================\n");
    printf("                BILL RECEIPT             \n");
    printf("=========================================\n");
    printf("Customer: %s\nDate: %s Time: %s\nContact: %s\n", name, date, timeSlot, contact);
    printf("Address : %s\nTable No: %d\n", address, tableNumber);
    if(mode == 1) {
        printf("People  : %d\n", people);
    }
    printf("\n-------------- ORDER SUMMARY --------------\n");
    printf("%-3s %-22s %-6s %-8s\n", "No", "Item", "Qty", "Amount");

    if(mode == 2) {
        for(int i = 0; i < N; i++) {
            if(orderedQty[i] > 0) {
                printf("%-3d %-22s %-6d Rs %-7d\n", i+1, items[i], orderedQty[i], orderedQty[i] * price[i]);
            }
        }
        if(addedCustomItem && mode == 1) {
            printf("%-3d %-22s %-6d Rs %-7d\n", 19, customItem, 1, customPrice);
        }
```

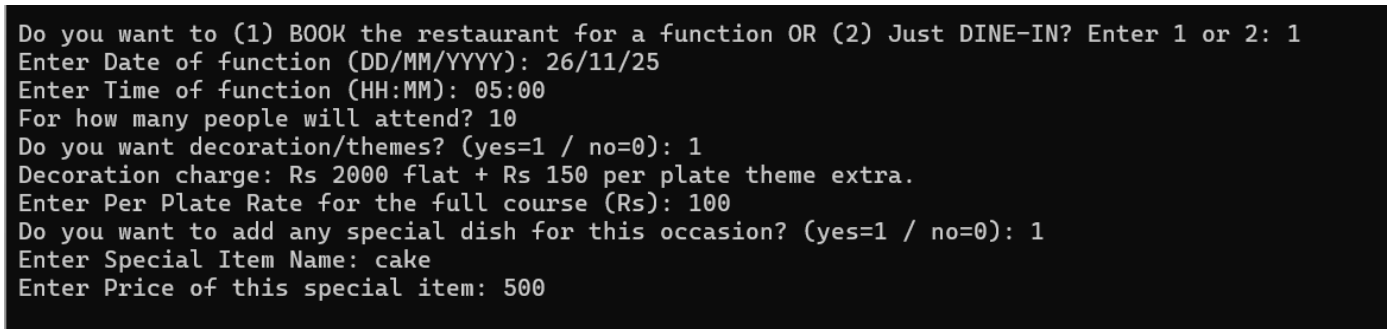
```
        printf("-----------------------------------------\n");
        printf("Total (without GST) : Rs %.2f\n", total);
    } else {
        // booking summary
        printf("Full-course for %d people @ Rs %.2f per plate: Rs %.2f\n", people, perPlateRate, perPlateRate * people);
        if(decoration == 1) printf("Decoration Charges      : Rs %.2f\n", decorCharge);
        if(addedCustomItem) printf("Special Dish Charge     : Rs %d\n", customPrice);
        printf("-----------------------------------------\n");
        printf("Total (without GST) : Rs %.2f\n", total);
    }

    printf("CGST (9%%)           : Rs %.2f\n", cgst);
    printf("SGST (9%%)           : Rs %.2f\n", sgst);
    printf("-----------------------------------------\n");
    printf("FINAL BILL          : Rs %.2f\n", finalBill);
    printf("=========================================\n");
    printf("        THANK YOU! PLEASE VISIT AGAIN       \n");
    printf("=========================================\n");

    return 0;
}

// ====================================
//            FUNCTIONS
// ====================================


void showMenu(char items[][30], int price[], int available[], int category[], int N) {
    printf("\n================ MENU =================\n");

    printf("\n--- FAST FOOD ---\n");
    for(int i = 0; i < N; i++) {
        if(category[i] == 0) {
            printf("%2d. %-20s Rs %3d %s\n", i+1, items[i], price[i],
                   available[i] ? "(Available)" : "(Not Available)");
        }
    }

    printf("\n--- BEVERAGES ---\n");
    for(int i = 0; i < N; i++) {
        if(category[i] == 2) {
            printf("%2d. %-20s Rs %3d %s\n", i+1, items[i], price[i],
                   available[i] ? "(Available)" : "(Not Available)");
        }
    }

    printf("\n--- THALIS ---\n");
    for(int i = 0; i < N; i++) {
        if(category[i] == 1) {
            printf("%2d. %-20s Rs %3d %s\n", i+1, items[i], price[i],
                   available[i] ? "(Available)" : "(Not Available)");
        }
    }

    printf("=========================================\n");
}
```

APPENDIX

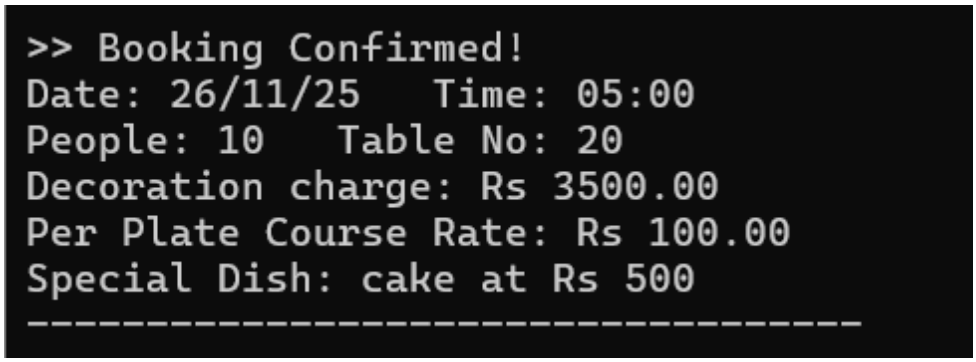
Customer Details



Booking / Dine in



Booking Details



Menu


C:\SHIVANI\c codes\r.exe
========== MENU ==========
--- FAST FOOD ---
1. Pizza Rs 180 (Available)
2. Burger Rs 90 (Not Available)
3. Pasta Rs 140 (Available)
4. Momos Rs 110 (Available)
5. French Fries Rs 70 (Available)
6. Sandwich Rs 80 (Available)
7. Hot Dog Rs 85 (Not Available)
18. Masala Dosa Rs 90 (Available)
--- BEVERAGES ---
8. Cold Drink Rs 40 (Available)
9. Coffee Rs 60 (Available)
10. Tea Rs 30 (Available)
11. Lassi Rs 80 (Not Available)
12. Soda Rs 40 (Available)
--- THALIS ---
13. Veg Thali Rs 150 (Available)
14. Non-Veg Thali Rs 200 (Available)
15. Paneer Thali Rs 180 (Available)
16. Chole Bhature Rs 120 (Not Available)
17. Rajasthani Thali Rs 250 (Not Available)
--- SPECIAL OCCASION ITEM ---
19. cake Rs 500 (Available)

Taking Order

```
Enter item number to order (0 to finish): 4
Enter quantity: 2
Added 2 x Momos

Enter item number to order (0 to finish): 7

SORRY! Hot Dog is NOT available.
Here are alternatives:
 - Pizza (Rs 180)
 - Pasta (Rs 140)
 - Momos (Rs 110)
 - French Fries (Rs 70)
 - Sandwich (Rs 80)

Enter item number to order (0 to finish): 5
Enter quantity: 12
Added 12 x French Fries

Enter item number to order (0 to finish): 13
Enter quantity: 5
Added 5 x Veg Thali

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

==========================================
```

Bill

```
==========================================
                BILL RECEIPT
==========================================
Customer: Shivani
Date: 26/11/25 Time: 05:00
Contact: 123456789
Address : Chhattisgarh
Table No: 20
People  : 10

-------------- ORDER SUMMARY --------------
No  Item                    Qty    Amount
Full-course for 10 people @ Rs 100.00 per plate: Rs 1000.00
Decoration Charges      : Rs 3500.00
Special Dish Charge     : Rs 500
------------------------------------------
Total (without GST) : Rs 5000.00
CGST (9%)           : Rs 450.00
SGST (9%)           : Rs 450.00
------------------------------------------
FINAL BILL          : Rs 5900.00
==========================================
        THANK YOU! PLEASE VISIT AGAIN
==========================================
```

References

Balagurusamy, E. *Programming in ANSI C*, Tata McGraw-Hill Education.

GeeksforGeeks. "C Programming Language Tutorials." https://www.geeksforgeeks.org/c-programming-language/

College Lecture Notes on Programming in C, School of Computer Science.

Kernighan, B. W., & Ritchie, D. M. *The C Programming Language*, 2nd Edition, Prentice Hall, 1988.

Tutorials Point. "C Programming – Basics and Advanced Concepts." https://www.tutorialspoint.com/cprogramming/index.htm

THANK YOU