PROGRAMMING FUNDAMENTALS [CT-175]

PROJECT REPORT

BILLING SYSTEM

GROUP MEMBERS:

- SHEES ALI [CT-88]
- TAQI HAIDER [CT-92]
- NAVEED IQBAL [CT-51]

DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

1. PROBLEM STATEMENT:

Create a command line application using C programming language which is use to display a restaurant receipt and also receive the related tax and other information. Inventory management system is also included to help the user to manage the items.

2. PROJECT DESCRIPTION:

User are provided with different option at the start of the program .

- Generate Invoice:-

  User can generate the invoice by entering different information.

- Show all Invoices:-

  Admin can see all the invoices and access all the records.

- Search Invoice:-

  Admin can recall all the invoices by mentioning the user invoice.

- Inventory Management:-

  We also integrated inventory management system for the items through which we provide the user the ability to create and modify items. Modifications can only be made by an admin user.

- Exit:-

  Program Exits.

3. DESCRIPTION OF FUNCTIONALITIES:

The most challenging part we faced while coding is firstly to decide that which approach should we go for, we have used multiple approaches so that anyone can easily understand our code, we have implemented different structures and functions and fully utilize arrays to their maximum potential in order to make our code easily readable . We believe in writing clean code that’s why we have adhered to the best practices where ever possible. We also faced difficulty in opening file in our code. Initially it was bit hard for us but we managed it as it was necessary to use file, without that our code will remain incomplete. Admin authentication and inventory management took most of the time.

4. INDIVIDUAL CONTRIBUTIONS:

Most of the work was done in group but some individual contributions are:

SYED MUHAMMAD SHEES ALI [CT-088]:

- Implemented admin authentication.
- Debugged the code thoroughly and code syntax.
- Implemented inventory management.

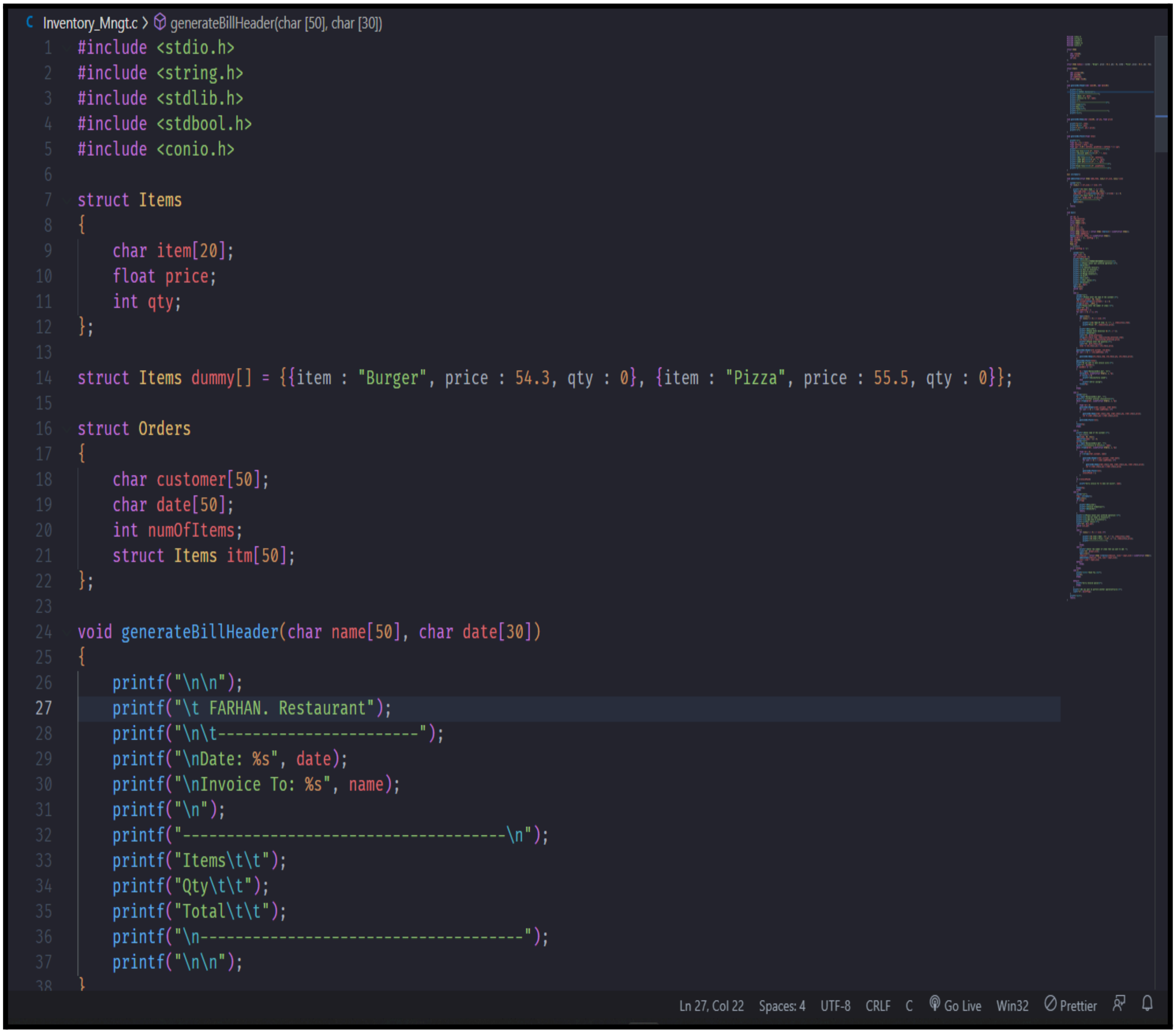
MUHAMMAD TAQI HAIDER [CT-092]:

- Implemented Invoice generation.
- Main idea of the project.
- Project report.

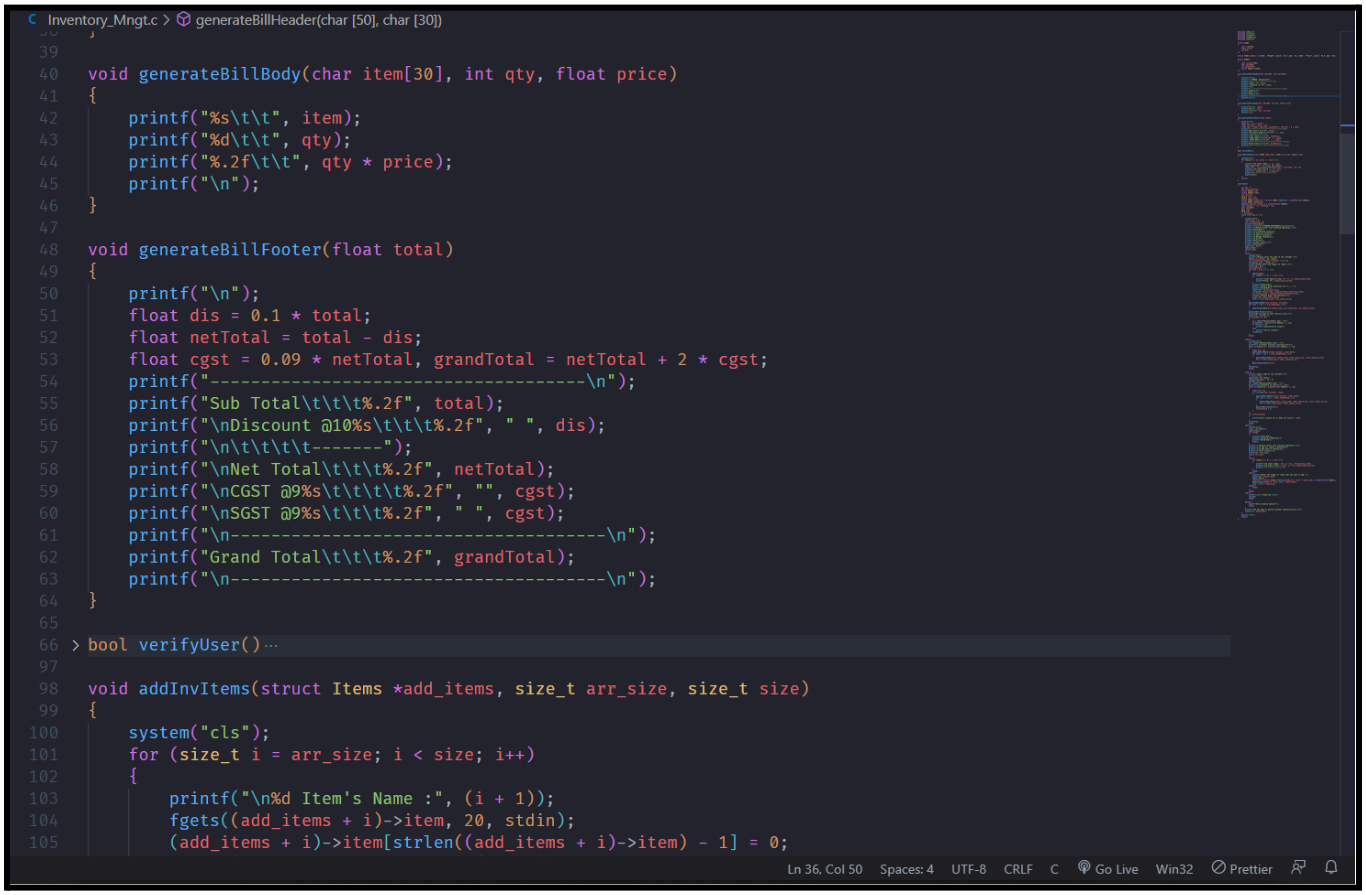
NAVEED IQBAL [CT-051]:

- Manage functions for invoice generation.
- Contributed in file handling.
- Use Structures and switch statement.

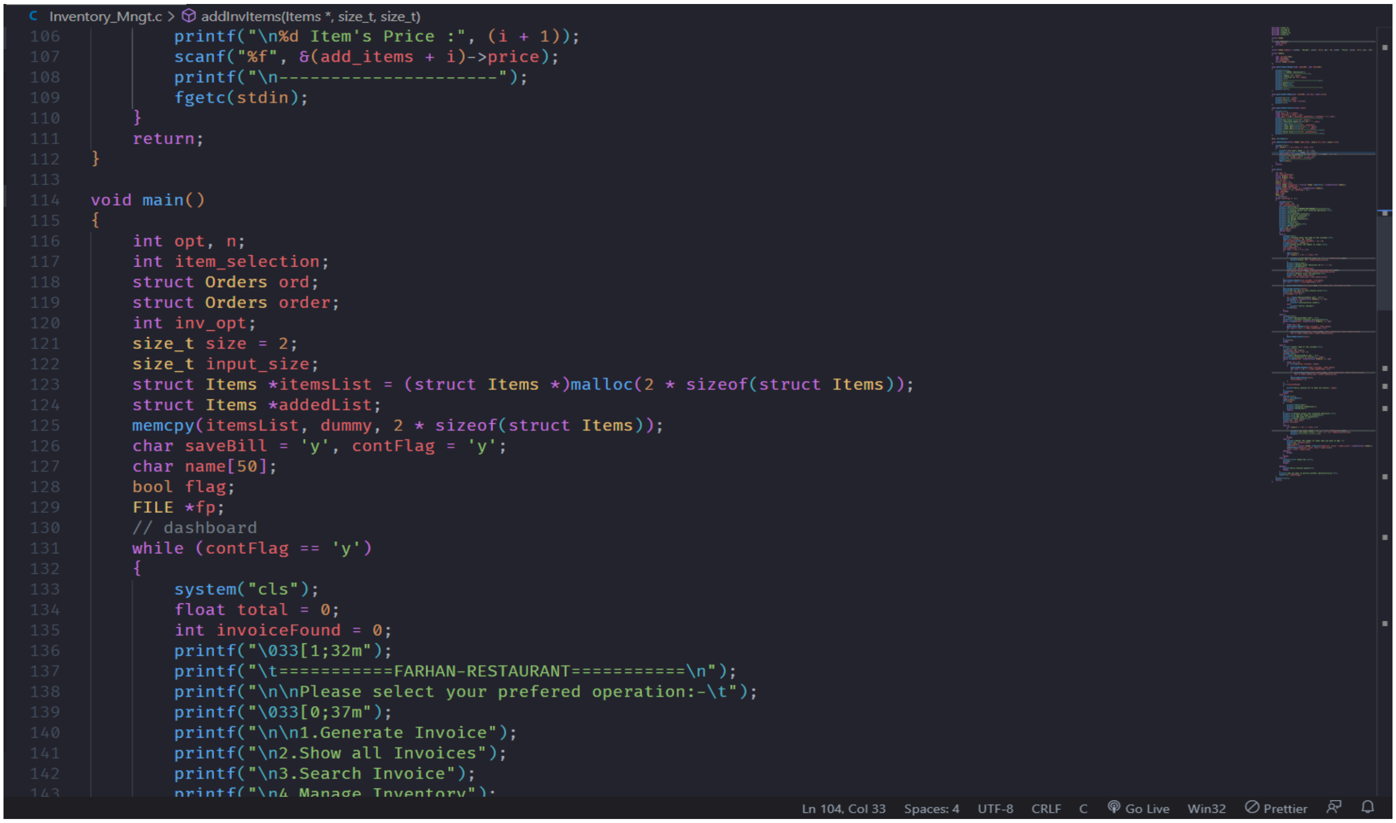
5. SOURCE CODE:



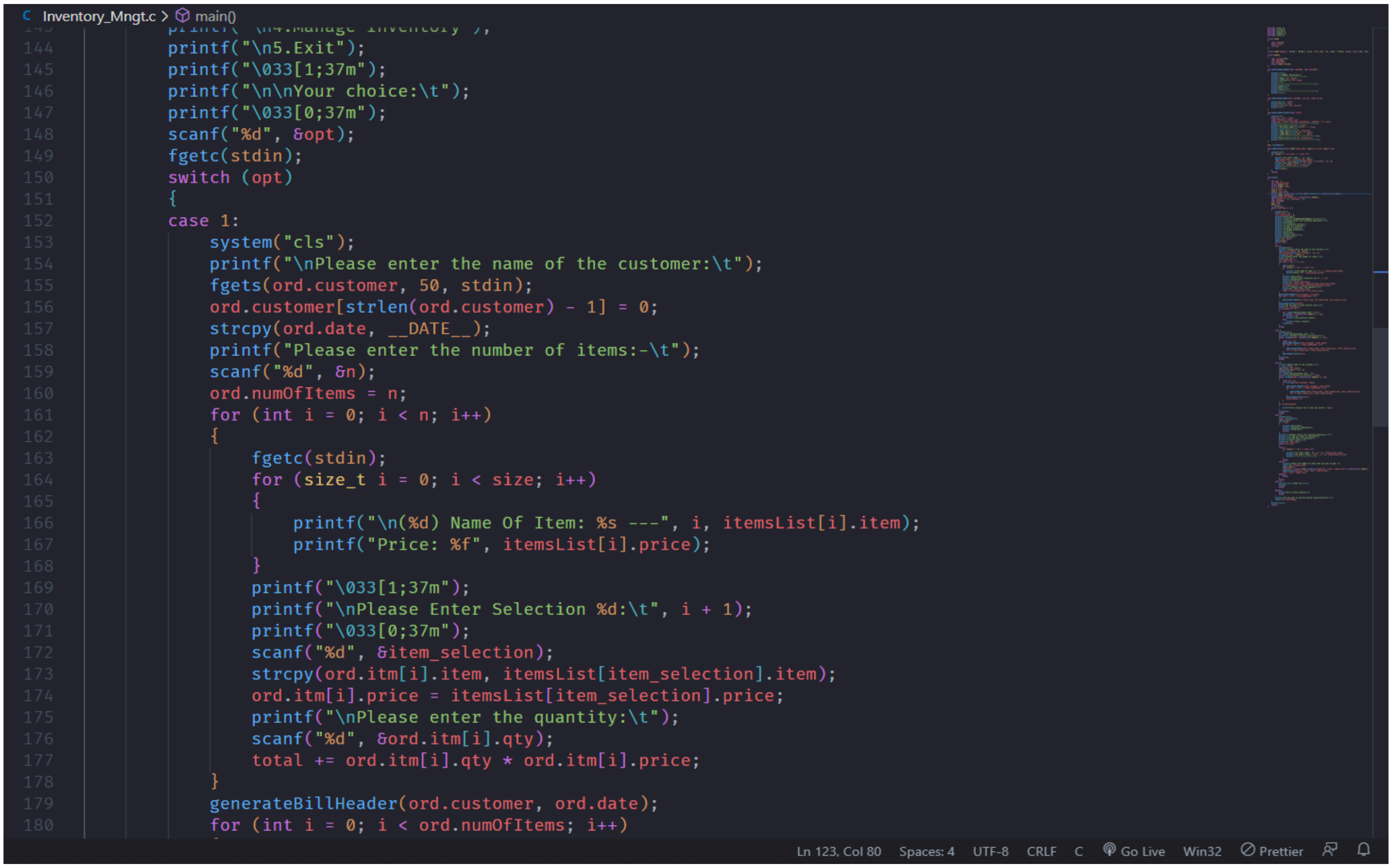
```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <stdbool.h>
#include <conio.h>

struct Items
{
    char item[20];
    float price;
    int qty;
};

struct Items dummy[] = {{item : "Burger", price : 54.3, qty : 0}, {item : "Pizza", price : 55.5, qty : 0}};

struct Orders
{
    char customer[50];
    char date[50];
    int numOfItems;
    struct Items itm[50];
};

void generateBillHeader(char name[50], char date[30])
{
    printf("\n\n");
    printf("\t FARHAN. Restaurant");
    printf("\n\t-----------------------");
    printf("\nDate: %s", date);
    printf("\nInvoice To: %s", name);
    printf("\n");
    printf("-------------------------------------\n");
    printf("Items\t\t");
    printf("Qty\t\t");
    printf("Total\t\t");
    printf("\n-------------------------------------");
    printf("\n\n");
}
```



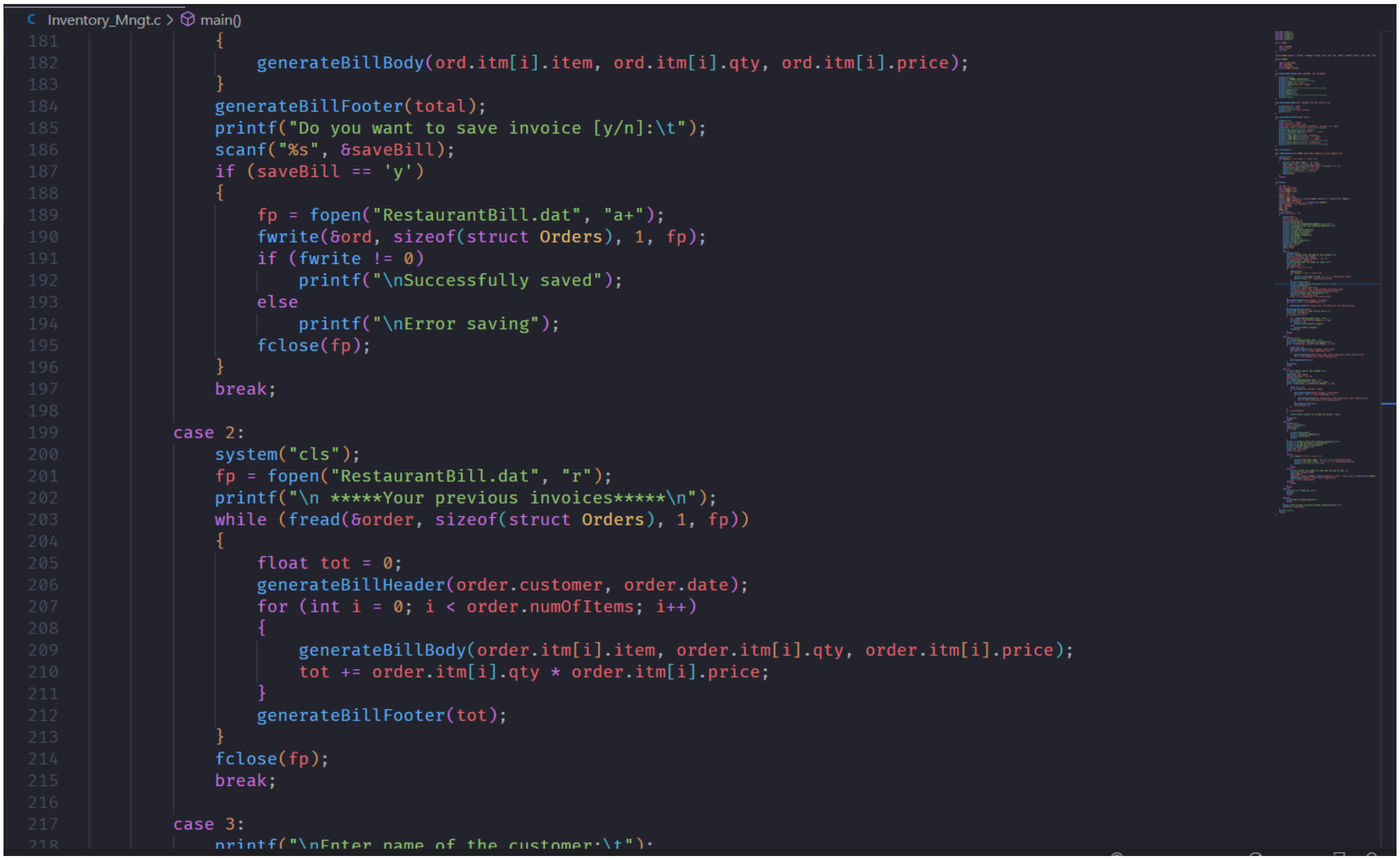
```c

void generateBillBody(char item[30], int qty, float price)
{
    printf("%s\t\t", item);
    printf("%d\t\t", qty);
    printf("%.2f\t\t", qty * price);
    printf("\n");
}

void generateBillFooter(float total)
{
    printf("\n");
    float dis = 0.1 * total;
    float netTotal = total - dis;
    float cgst = 0.09 * netTotal, grandTotal = netTotal + 2 * cgst;
    printf("-------------------------------------\n");
    printf("Sub Total\t\t\t%.2f", total);
    printf("\nDiscount @10%s\t\t\t%.2f", " ", dis);
    printf("\n\t\t\t\t-------");
    printf("\nNet Total\t\t\t%.2f", netTotal);
    printf("\nCGST @9%s\t\t\t\t%.2f", "", cgst);
    printf("\nSGST @9%s\t\t\t%.2f", " ", cgst);
    printf("\n-------------------------------------\n");
    printf("Grand Total\t\t\t%.2f", grandTotal);
    printf("\n-------------------------------------\n");
}

> bool verifyUser()…

void addInvItems(struct Items *add_items, size_t arr_size, size_t size)
{
    system("cls");
    for (size_t i = arr_size; i < size; i++)
    {
        printf("\n%d Item's Name :", (i + 1));
        fgets((add_items + i)->item, 20, stdin);
        (add_items + i)->item[strlen((add_items + i)->item) - 1] = 0;
```



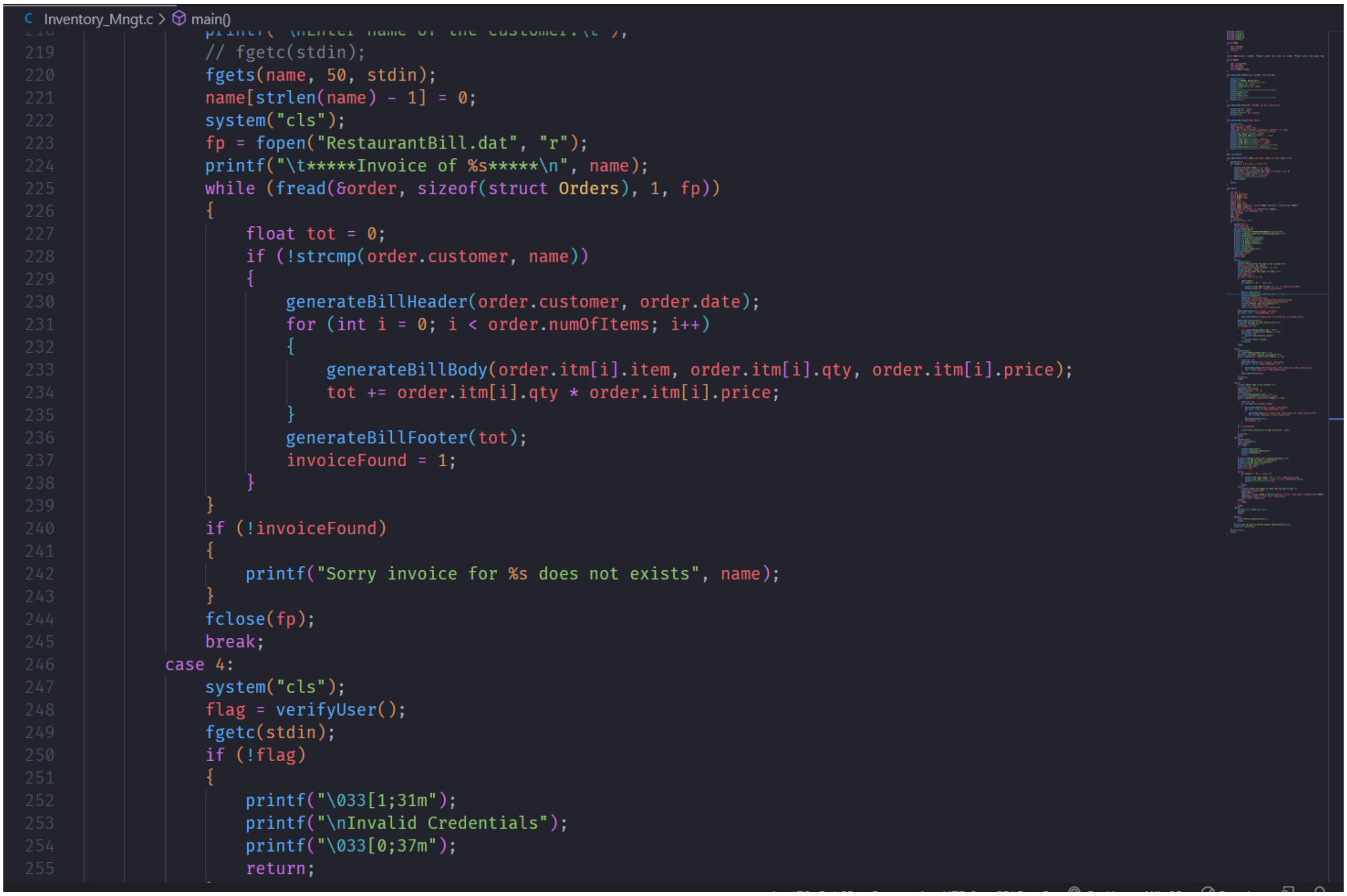
```c
        printf("\n%d Item's Price :", (i + 1));
        scanf("%f", &(add_items + i)->price);
        printf("\n----------------------");
        fgetc(stdin);
    }
    return;
}

void main()
{
    int opt, n;
    int item_selection;
    struct Orders ord;
    struct Orders order;
    int inv_opt;
    size_t size = 2;
    size_t input_size;
    struct Items *itemsList = (struct Items *)malloc(2 * sizeof(struct Items));
    struct Items *addedList;
    memcpy(itemsList, dummy, 2 * sizeof(struct Items));
    char saveBill = 'y', contFlag = 'y';
    char name[50];
    bool flag;
    FILE *fp;
    // dashboard
    while (contFlag == 'y')
    {
        system("cls");
        float total = 0;
        int invoiceFound = 0;
        printf("\033[1;32m");
        printf("\t==========FARHAN-RESTAURANT==========\n");
        printf("\n\nPlease select your prefered operation:-\t");
        printf("\033[0;37m");
        printf("\n\n1.Generate Invoice");
        printf("\n2.Show all Invoices");
        printf("\n3.Search Invoice");
        printf("\n4.Manage Inventory");
```



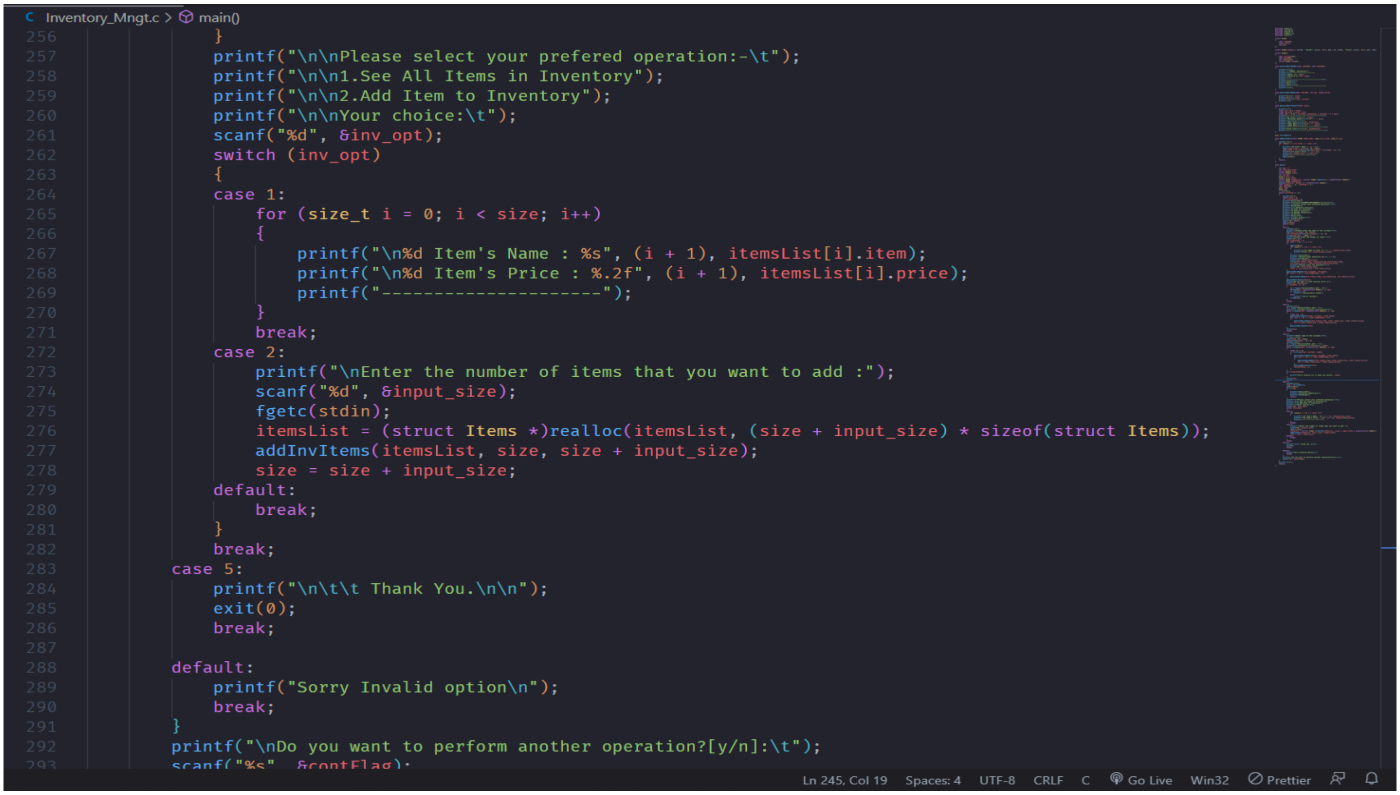
```c
        printf("\n4.Manage Inventory");
        printf("\n5.Exit");
        printf("\033[1;37m");
        printf("\n\nYour choice:\t");
        printf("\033[0;37m");
        scanf("%d", &opt);
        fgetc(stdin);
        switch (opt)
        {
        case 1:
            system("cls");
            printf("\nPlease enter the name of the customer:\t");
            fgets(ord.customer, 50, stdin);
            ord.customer[strlen(ord.customer) - 1] = 0;
            strcpy(ord.date, __DATE__);
            printf("Please enter the number of items:-\t");
            scanf("%d", &n);
            ord.numOfItems = n;
            for (int i = 0; i < n; i++)
            {
                fgetc(stdin);
                for (size_t i = 0; i < size; i++)
                {
                    printf("\n(%d) Name Of Item: %s ---", i, itemsList[i].item);
                    printf("Price: %f", itemsList[i].price);
                }
                printf("\033[1;37m");
                printf("\nPlease Enter Selection %d:\t", i + 1);
                printf("\033[0;37m");
                scanf("%d", &item_selection);
                strcpy(ord.itm[i].item, itemsList[item_selection].item);
                ord.itm[i].price = itemsList[item_selection].price;
                printf("\nPlease enter the quantity:\t");
                scanf("%d", &ord.itm[i].qty);
                total += ord.itm[i].qty * ord.itm[i].price;
            }
            generateBillHeader(ord.customer, ord.date);
            for (int i = 0; i < ord.numOfItems; i++)
```



```c
            {
                generateBillBody(ord.itm[i].item, ord.itm[i].qty, ord.itm[i].price);
            }
            generateBillFooter(total);
            printf("Do you want to save invoice [y/n]:\t");
            scanf("%s", &saveBill);
            if (saveBill == 'y')
            {
                fp = fopen("RestaurantBill.dat", "a+");
                fwrite(&ord, sizeof(struct Orders), 1, fp);
                if (fwrite != 0)
                    printf("\nSuccessfully saved");
                else
                    printf("\nError saving");
                fclose(fp);
            }
            break;

        case 2:
            system("cls");
            fp = fopen("RestaurantBill.dat", "r");
            printf("\n *****Your previous invoices*****\n");
            while (fread(&order, sizeof(struct Orders), 1, fp))
            {
                float tot = 0;
                generateBillHeader(order.customer, order.date);
                for (int i = 0; i < order.numOfItems; i++)
                {
                    generateBillBody(order.itm[i].item, order.itm[i].qty, order.itm[i].price);
                    tot += order.itm[i].qty * order.itm[i].price;
                }
                generateBillFooter(tot);
            }
            fclose(fp);
            break;

        case 3:
            printf("\nEnter name of the customer:\t");
```



```c
            printf("\nEnter name of the customer:\t");
            // fgetc(stdin);
            fgets(name, 50, stdin);
            name[strlen(name) - 1] = 0;
            system("cls");
            fp = fopen("RestaurantBill.dat", "r");
            printf("\t*****Invoice of %s*****\n", name);
            while (fread(&order, sizeof(struct Orders), 1, fp))
            {
                float tot = 0;
                if (!strcmp(order.customer, name))
                {
                    generateBillHeader(order.customer, order.date);
                    for (int i = 0; i < order.numOfItems; i++)
                    {
                        generateBillBody(order.itm[i].item, order.itm[i].qty, order.itm[i].price);
                        tot += order.itm[i].qty * order.itm[i].price;
                    }
                    generateBillFooter(tot);
                    invoiceFound = 1;
                }
            }
            if (!invoiceFound)
            {
                printf("Sorry invoice for %s does not exists", name);
            }
            fclose(fp);
            break;
        case 4:
            system("cls");
            flag = verifyUser();
            fgetc(stdin);
            if (!flag)
            {
                printf("\033[1;31m");
                printf("\nInvalid Credentials");
                printf("\033[0;37m");
                return;
```



```c
            }
            printf("\n\nPlease select your prefered operation:-\t");
            printf("\n\n1.See All Items in Inventory");
            printf("\n\n2.Add Item to Inventory");
            printf("\n\nYour choice:\t");
            scanf("%d", &inv_opt);
            switch (inv_opt)
            {
            case 1:
                for (size_t i = 0; i < size; i++)
                {
                    printf("\n%d Item's Name : %s", (i + 1), itemsList[i].item);
                    printf("\n%d Item's Price : %.2f", (i + 1), itemsList[i].price);
                    printf("---------------------");
                }
                break;
            case 2:
                printf("\nEnter the number of items that you want to add :");
                scanf("%d", &input_size);
                fgetc(stdin);
                itemsList = (struct Items *)realloc(itemsList, (size + input_size) * sizeof(struct Items));
                addInvItems(itemsList, size, size + input_size);
                size = size + input_size;
            default:
                break;
            }
            break;
        case 5:
            printf("\n\t\t Thank You.\n\n");
            exit(0);
            break;

        default:
            printf("Sorry Invalid option\n");
            break;
        }
        printf("\nDo you want to perform another operation?[y/n]:\t");
        scanf("%s", &contFlag);
```