# C PROGRAMMING PROJECT REPORT

- **Project Title:** MINI SHOPPING BILLING SYSTEM
- **Course:** Programming in C (B.Tech 1st Semester)
- **Course Code:** CSEG1041_5
- **Submitted By:** DHRUVI SINGH , 590021818
- **Submitted To:** MOHSIN F.DAR, SOCS
- **Date of Submission:** 02/02/2025

## 1. Problem Statement

**In small shops, billing is often done manually, which can cause calculation errors, time delays, and difficulty maintaining records. The Mini Shopping Billing System automates billing by allowing users to add items with quantity and price, and then generates an accurate bill, reducing errors and saving time.**

## 2. Objective of the Project

- **Develop a simple billing system using C programming.**
- **Enable adding items with their name, price, and quantity.**
- **Automatically calculate the total amount for all items.**
- **Display a clear itemized bill with totals.**
- **Minimize human errors in billing and speed up checkout.**

## 3. Software / Tools Used

- **Operating System: Mac**
- **Compiler: gcc compiler , vs code**

## 4.Algorithm

### main.c

### 1)Start the program.

**2Declare variable choice.**
**3)Repeat the following steps:**

**A-> Display the menu:**

- **Add Item**

. **View bill**

•

**B -> Ask the user to enter a choice.**

**C -> Read choice.**

**D -> If choice = 1**
**→ Call addItem().**

**E -> Else if choice = 2**
**→ Call viewBill().**

**F -> Else if choice = 3**
**→ Display exit message.**

**G -> Else**
**→ Display “Invalid option”.**

**4.Continue the loop until choice = 3.**

**5.End the program.**

## shop.c
## ADD ITEM (addItem function)

**1.Open file bill.txt in append mode.**

**2.If the file cannot be opened:**

**■ Display error message.**

**■ Exit function.**

**3.Create an Item structure variable.**

**4.Ask the user to enter:**

**■ Item name**
**■ Price**
**■ Quantity**

**5.Store these values in the structure.**

**6.Close the file.**

**7.Display “Item added successfully”.**

**8.End function.**

## VIEW BILL (viewBill function)

**1.Open file bill.txt in read mode.**

**2.If the file cannot be opened:**

**■ Display “No bill found!”**

**■ Exit function.**

**3.Declare:**

**■ An Item structure variable**

**■ A variable total = 0**

**4.Display bill table heading.**

**5.Read each item from the file until EOF:**

**■ Calculate amount = price × quantity**

**■ Display item details in table format**

**Add amount to total**

**After reading all items:**

**Display a separator line**

**Display “Total Amount = total”**

**Close the file.**

**End function.**

**6. After reading all items:**

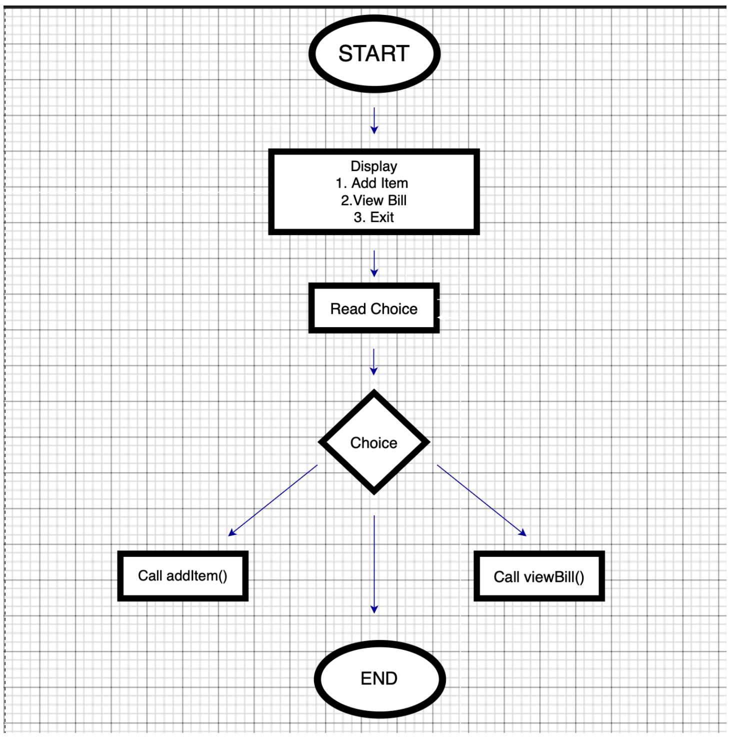
- ■ **Display a separator line**
- ■ **Display "Total Amount = total”**

**7.Close the file.**

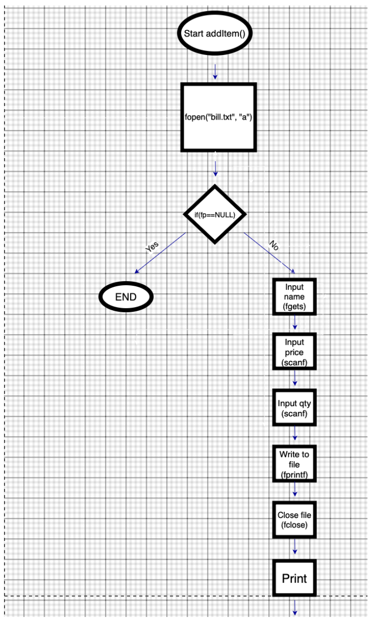
**8.End function.**

## 5. Flowchart

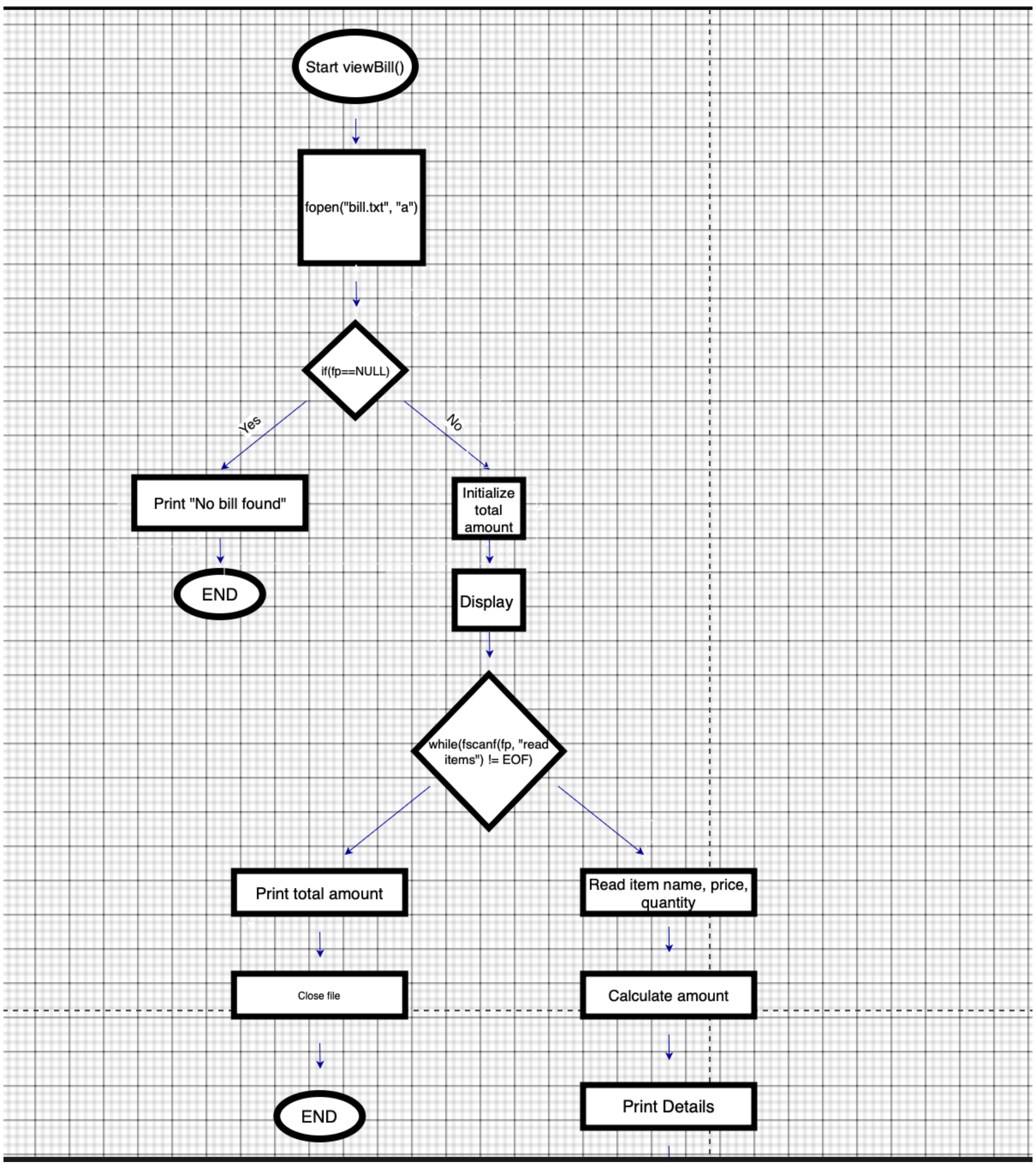
### (main.c)



### addItem()



# viewBil()


Start viewBill()
fopen("bill.txt", "a")
if(fp==NULL)
Yes
No
Print "No bill found"
END
Initialize total amount
Display
while(fscanf(fp, "read items") != EOF)
Print total amount
Close file
END
Read item name, price, quantity
Calculate amount
Print Details

# 7. Source Code

## main.c

```
#include<stdio.h>
#include<stdlib.h>
#include "shop.h"
int main() {
   int choice;
   do{
    printf("\n=== MINI SHPPING BILLING SYSTEM ===\n");
    printf("1. Add Item\n");
    printf("2. View bill\n");
    printf("3. Exit\n");
    printf("Enter your choice: ");
    scanf("%d", &choice)

    switch(choice){
        case 1:
            addItem();
            break;
        case 2:
            viewBill();
            break;
        case 3:
            printf("Exiting...\n");
            break;
        default:
            printf("Invalid optional!  try again.\n");
    }
   } while(choice != 3);
    return 0;
}
```

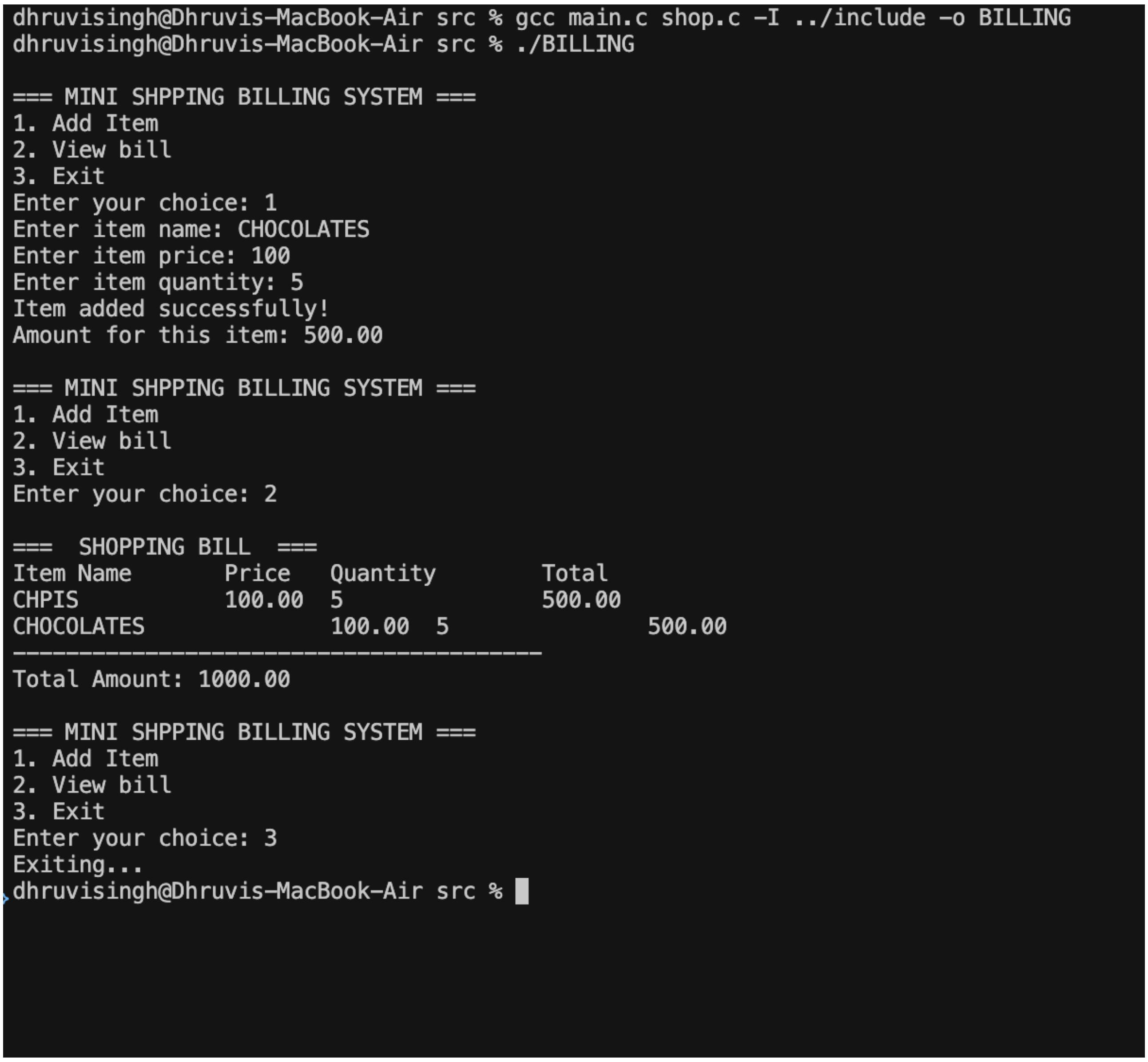
# shop.c

```c
#include <stdio.h>
#include <stdlib.h>
#include "shop.h"
struct item {
    char name[50];
    float price;
    int quantity;
};
void addItem() {
    FILE *fp;
    fp = fopen("bill.txt", "a");
    if (fp == NULL)
    {
        printf("Error opening file!\n");
        return;
    }
    struct item item;
    float amt;
    printf("Enter item name: ");
    scanf("%s", item.name);
    printf("Enter item price: ");
    scanf("%f", &item.price);
    printf("Enter item quantity: ");
    scanf("%d", &item.quantity);
    amt=item.price * item.quantity;
    fprintf(fp, "%s %.2f %d\n", item.name, item.price,
item.quantity);
    fclose(fp);
    printf("Item added successfully!\n");
    printf("Amount for this item: %.2f\n", amt);
}
void viewBill() {
    FILE *fp;
    fp = fopen("bill.txt", "r");
    if (fp == NULL) {
        printf("No items in the bill.\n");
        return;
    }
    struct item item;
    float total = 0;
    printf("\n===  SHOPPING BILL  ===\n");
    printf("Item Name\tPrice\tQuantity\tTotal\n");
    while (fscanf(fp, "%s %f %d", item.name, &item.price,
&item.quantity) != EOF) {
        float amt = item.price * item.quantity;
        printf("%s\t\t%.2f\t%d\t\t%.2f\n", item.name,
item.price, item.quantity, amt);
        total += amt;
    }
    printf("----------------------------------------\n");
    printf("Total Amount: %.2f\n", total);
    fclose(fp);
}
```

## shop.h

```
#ifndef SHOP_H
#define SHOP_H
void addItem();
void viewBill();
#endif
```

## OUTPUT



```
dhruvisingh@Dhruvis-MacBook-Air src % gcc main.c shop.c -I ../include -o BILLING
dhruvisingh@Dhruvis-MacBook-Air src % ./BILLING

=== MINI SHPPING BILLING SYSTEM ===
1. Add Item
2. View bill
3. Exit
Enter your choice: 1
Enter item name: CHOCOLATES
Enter item price: 100
Enter item quantity: 5
Item added successfully!
Amount for this item: 500.00

=== MINI SHPPING BILLING SYSTEM ===
1. Add Item
2. View bill
3. Exit
Enter your choice: 2

===  SHOPPING BILL  ===
Item Name       Price   Quantity        Total
CHPIS           100.00  5               500.00
CHOCOLATES              100.00  5               500.00
----------------------------------------
Total Amount: 1000.00

=== MINI SHPPING BILLING SYSTEM ===
1. Add Item
2. View bill
3. Exit
Enter your choice: 3
Exiting...
dhruvisingh@Dhruvis-MacBook-Air src % 
```

## 8. Conclusion

The Mini Shopping Billing System successfully automates the billing process for a small shop by enabling users to add multiple items and generate a detailed bill with totals. This system reduces manual errors, speeds up billing, and helps maintain an electronic record of sales.

## 9. Future Enhancements

In the future, the Mini Shopping Billing System can be enhanced by adding inventory management to automatically update stock, improving the user interface with better menus and validations, introducing customer records and discount features, providing options to save and print bills, and implementing a secure login system. These upgrades will make the system more efficient, user-friendly, and suitable for real-world shop management.