Restaurant Billing System

(with menu and categories)

Submitted by: Vatsal Chawla

SAP ID: 590027526

Submitted to: Dr. Prashant Trivedi

Abstract

This project is a console-based application written in C to help restaurants manage their billing process.

- It displays a categorized menu (Starters, Main Course, etc.).
- It calculates bills automatically based on user selection.
- It saves a permanent record of the transaction in a text file.

The system is designed to be simple, fast, and accurate.

Problem Definition

Manual Billing Issues:

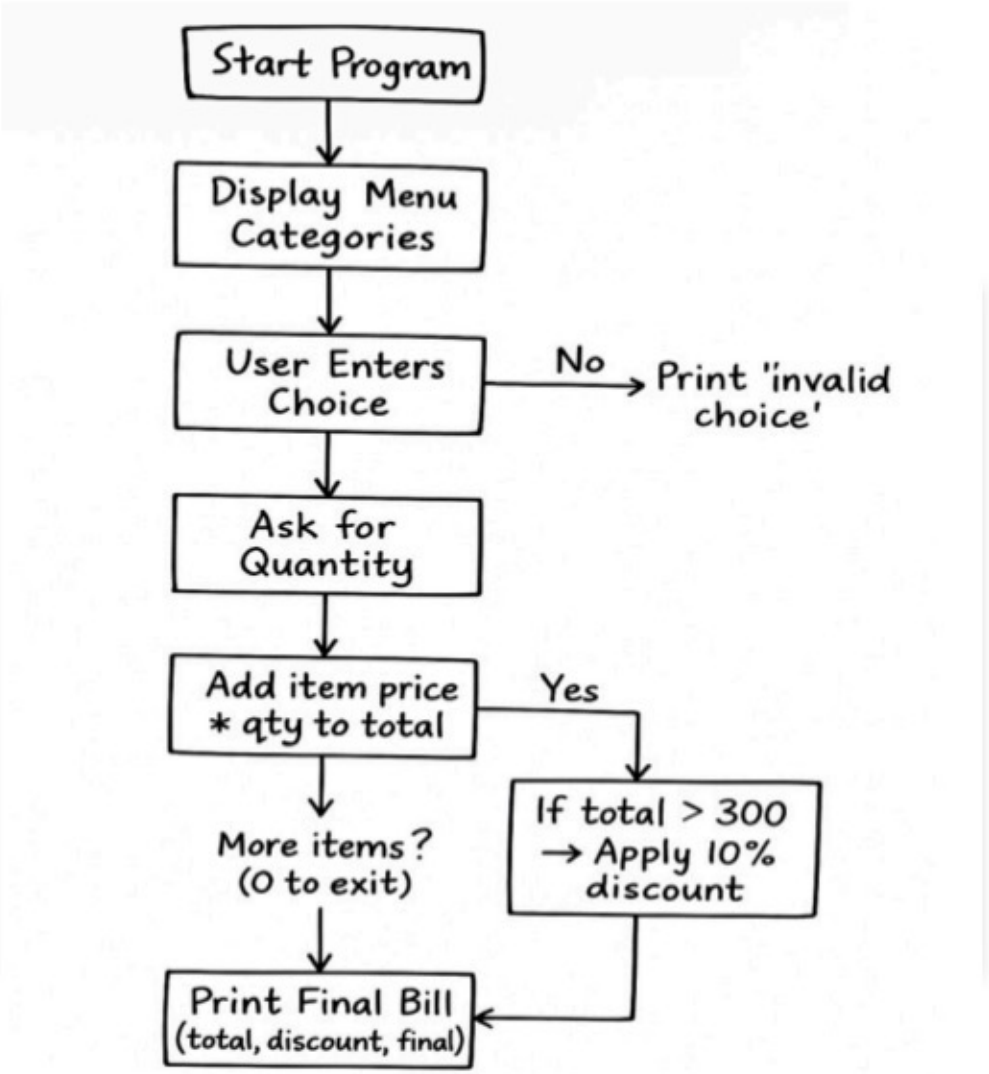
- Handwritten receipts are prone to calculation errors.
- It is slow and inefficient during rush hours.
- Paper records are hard to maintain and easily lost.

Solution: An automated C program that eliminates math errors and saves data digitally.

System Design

Step-by-Step Logic:

1. **Initialize:** Load menu data into structures.
2. **Display:** Show menu with prices.
3. **Input:** Ask user for Item Number and Quantity.
4. **Calculate:** Add (Price x Qty) to Total.
5. **Repeat:** Continue until user enters '0'.
6. **Output:** Print Final Bill & Save to File.



Implementation: Data Structure

```
struct menu {
    char itemname[50];
    float price;
    char category[20];
};
```

Structure Definition:

We use a `struct` to group data for each food item. This keeps our code organized.

- `itemname`: The name (e.g., "Dosa").
- `price`: The cost (e.g., 80.00).
- `category`: Grouping (e.g., "Main").

Implementation: Display Menu

```
printf("  ***MENU****\n");

printf("\nSTARTERS\n");
for(int i = 0; i <= 1; i++)
printf("%d. %-15s Rs %.2f\n", i+1, m[i].itemname, m[i].price);
```

Function Logic:

- The function loops through specific ranges of the array to print items category by category.
- %-15s is used for left-alignment padding, ensuring the output looks like a neat table.

Implementation: Taking Orders

```
float takeorder(struct menu m[]) {
int ch, qty;
float total = 0;

while(1){
    printf("\nEnter your choice (0 to exit): "); //user input
    scanf("%d", &ch);
```

Looping for Input:

- An infinite `while(1)` loop allows the user to keep adding items.
- The loop breaks only when the user enters `0`.
- The total is updated in real-time: `Price * Quantity`.

Implementation: File Handling

```
void savebill(float total) {

float discount =0;
if(total >= 300)
discount=total * 0.10;

float final = total -discount; //final amount compu

FILE *fptr;
fptr=fopen("bill.txt", "w");   //file opened file i

if(fptr == NULL) {
printf("file doesnt exist\n");
return;
}
riting into file
fprintf(fptr, "---- BILL ----\n");
fprintf(fptr, "Total amount : %.2f\n" ,total);
fprintf(fptr, "Discount     : %.2f\n" , discount);
fprintf(fptr, "Final amount : %.2f\n"  , final);
fprintf(fptr, "\nThank you for visiting!\n");

fclose(fptr);//file closed
```

Saving Data:

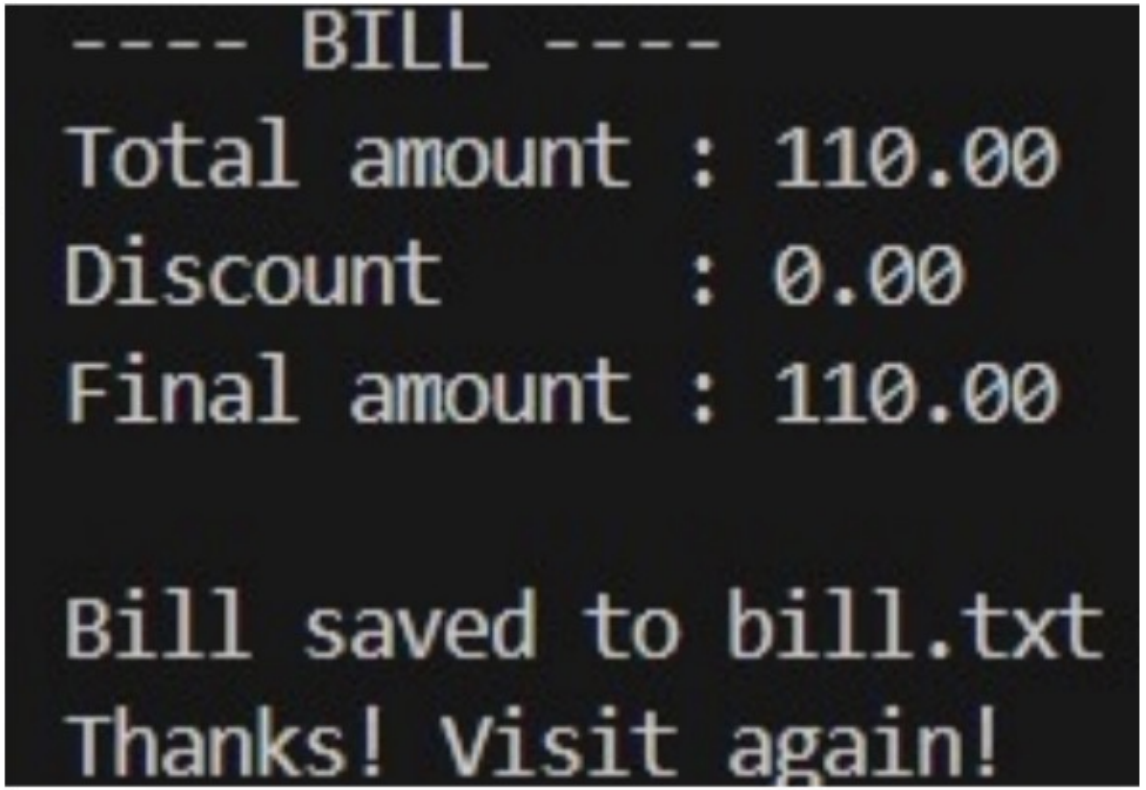
- `fopen` opens (or creates) "bill.txt" in write mode.
- `fprintf` writes the calculated data into the file instead of the screen.
- This ensures we have a permanent record of the sale.

Testing & Results (Case 1: No Discount)

Test Scenario:

We ordered items below the discount limit (Rs. 300).

Order : 2 Samosas and 1 tea
Total < 300

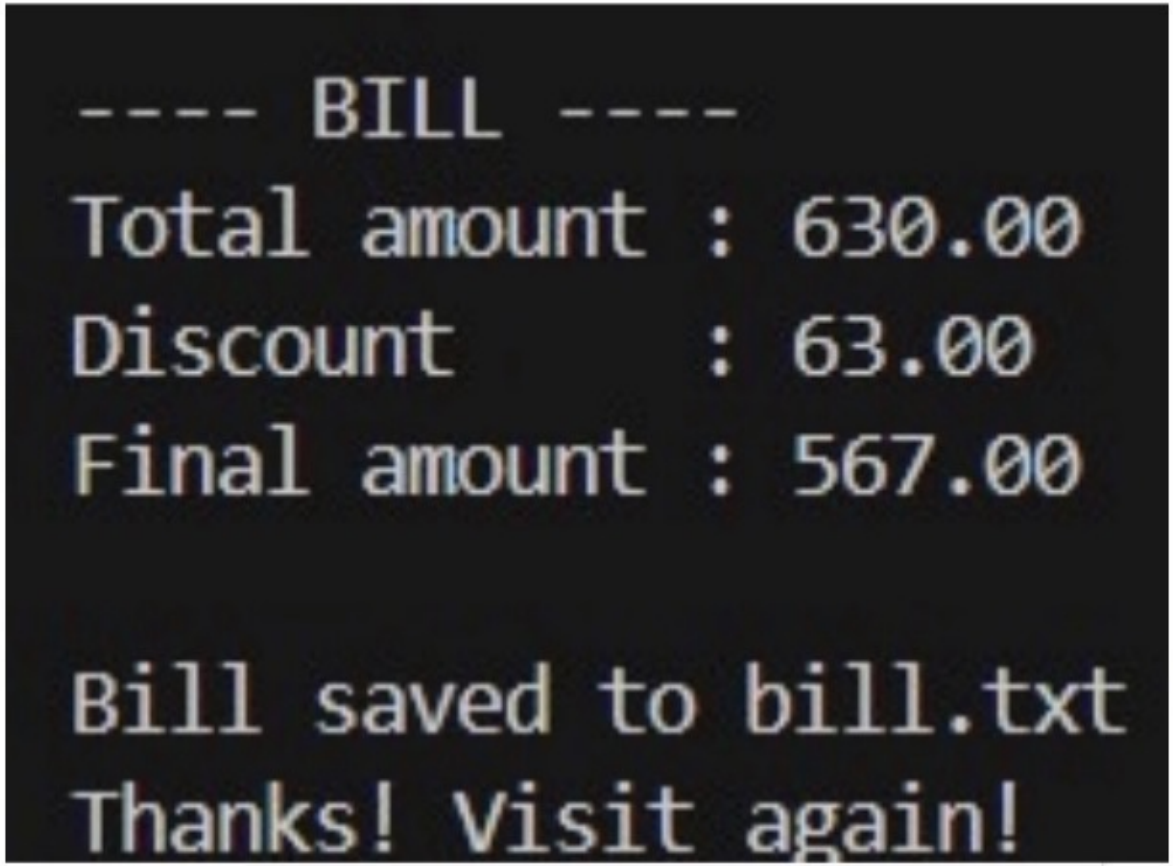


Testing & Results (Case 2: With Discount)

Test Scenario:

We ordered items above the discount limit (Rs. 300) to check the 10% discount logic.

Order : 3 Paneer rolls and 2 Icecreams
Total > 300



Conclusion & Future Vision

Conclusion

- The project works correctly and is easy to use.
- It calculates the total bill and applies discounts accurately.
- It successfully saves the bill to a text file.
- This project helped me learn how to use C structures and file handling.

Future Improvements

- **Add Login:** Create a password system for security.
- **Search Option:** Allow finding items by name instead of just numbers.
- **Edit Menu:** Let the user add or delete food items from the list.
- **Better Design:** Make the program look colorful or use a proper window interface.

References

I utilized the following resources to complete this project:

- **Textbook:** *"Let Us C"* by Yashavant Kanetkar.
- **Class Notes:** Programming Fundamentals lecture notes.