

Restaurant Billing System

(with menu categories and discounts)

SUBMITTED BY

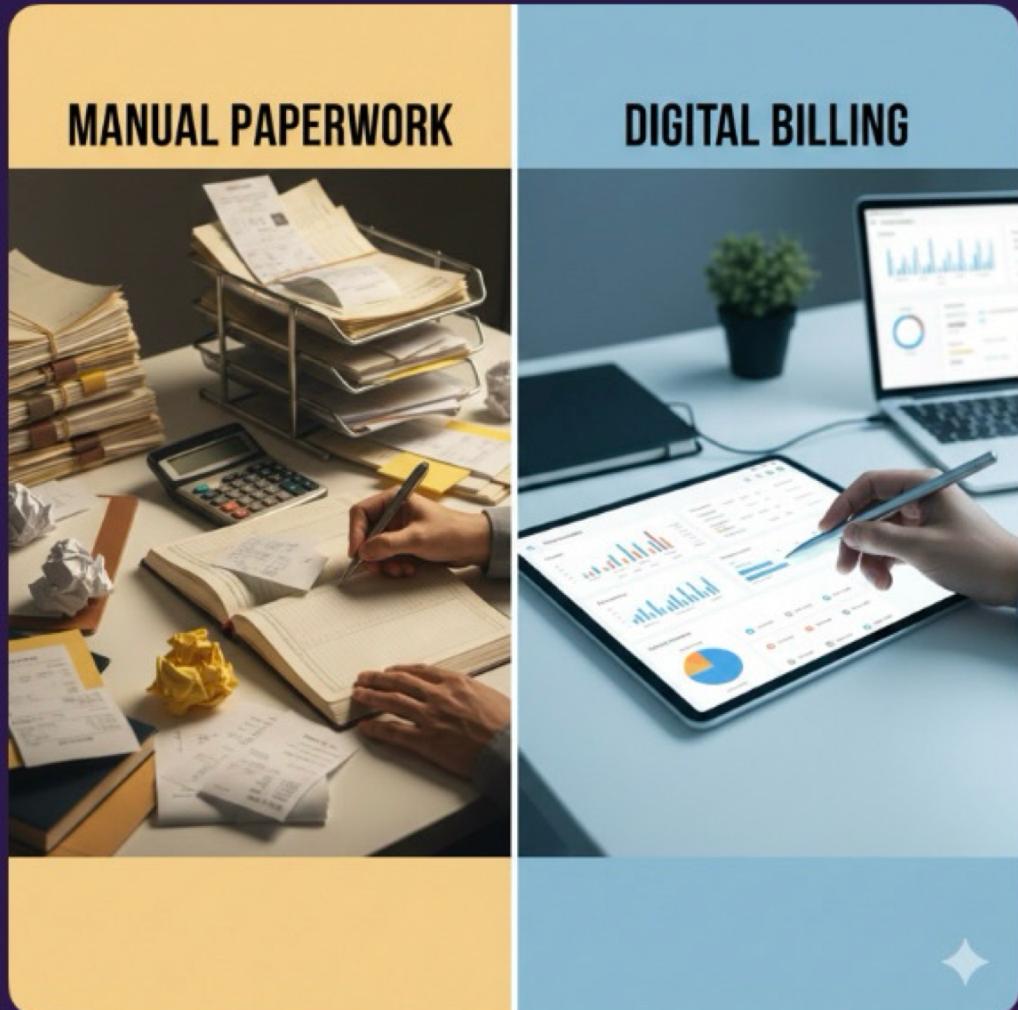
Vatsal Chawla
SAP ID: 590027526

SUBMITTED TO

Dr. Prashant Trivedi

2. Abstract

- Manual billing on paper is very slow.
- It creates a mess in busy restaurants.
- A **digital program** was created to fix this.
- It categorizes the menu (Starters, Drinks).
- It calculates bills automatically.
- It gives a 10% discount for bills over 300 INR.



3. Problem Definition

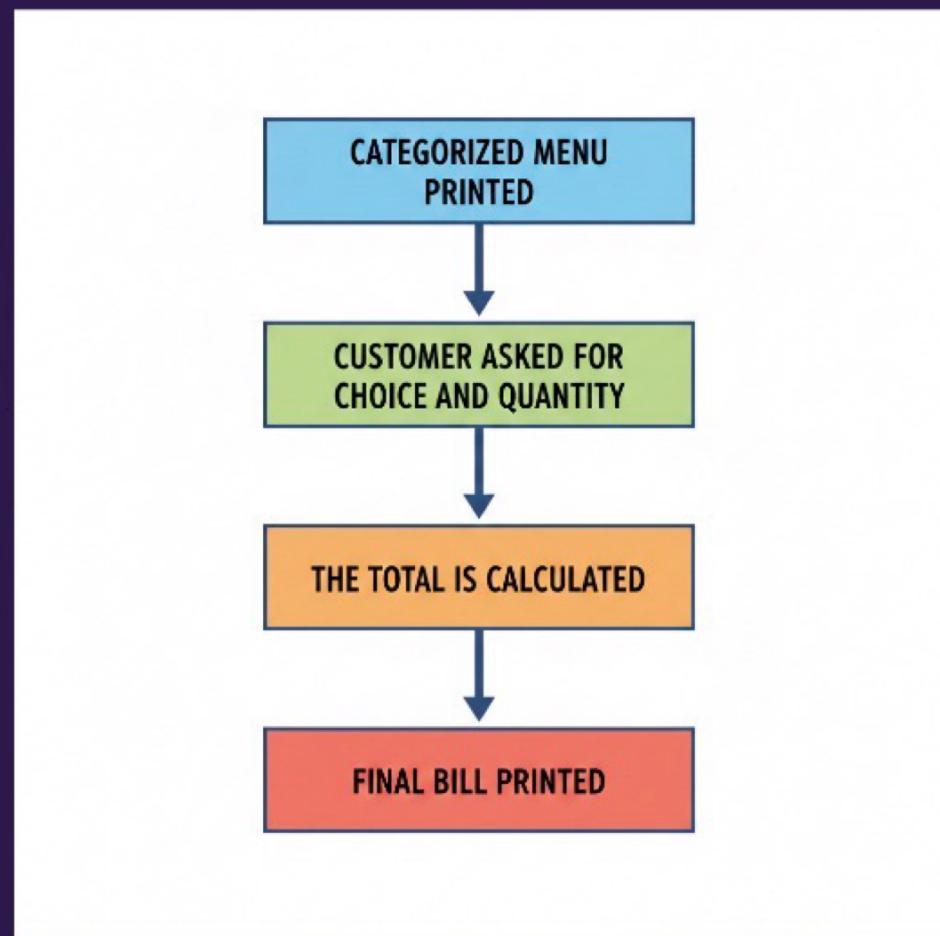
- **Long Wait Times:** Customers wait too long for hand-written bills.
- **Math Errors:** Mistakes happen when adding numbers manually.
- **Lost Records:** Paper slips get lost or damaged easily.
- **No Tracking:** It is hard to know total sales at the end of the day.



4. System Design

How the Program is Designed

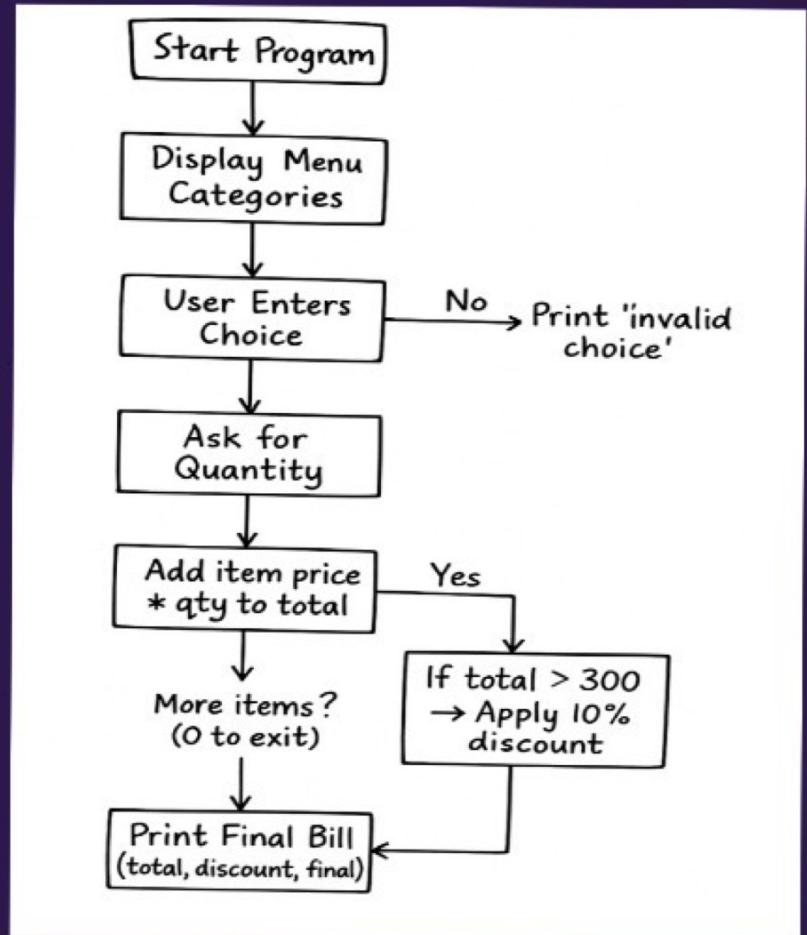
- A plan was made before writing code to ensure structure.
- The system loads the menu into memory for quick access.
- It uses a simple loop to accept multiple orders from a single table.
- The core job is to check the total amount and apply a discount rule if needed.



5. Algorithm and Flowchart

Steps

- Start the program.
- Show the menu to the user.
- Ask for **Item Choice** and **Quantity**.
- Check if the choice is valid (1-12).
- Add the cost to the Total.
- If Total > 300, apply **10% Discount**.
- Print the Final Bill.



6. Implementation: Data Definition (Structs)

Using 'Structs' for Items

- The C struct feature was used to group data.
- It holds the **Name**, **Price**, and **Category** together for each food item.
- This makes the code clean and easy to manage for 12 menu items.

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

7. Implementation: Menu Display (The Printing Part)

Printing the Menu

- A simple 'for' loop iterates through the 12 items stored in the 'struct' array.
- The 'printf' function displays the index number, item name, and price.
- This gives the user a clear, categorized list of available choices.

```
//printing menu
printf(" *** MENU *** \n");
//printing starters
printf("\n--- STARTERS ---\n");
for(int i = 0; i<=1 ;i++){
    printf("%d. %15srupees %.2f\n",i+1, m[i].itemname, m[i].price);}

//printing maincourse
printf("\n--- MAINCOURSE---\n");
for(int i=2; i <=6 ; i++){
    printf("%d. %15s Rs %.2f\n", i+1, m[i].itemname , m[i].price);}

//printing drinks
printf("\n--- DRINKS ---\n");
for(int i =7 ; i <= 9; i++){
    printf("%d. %15s Rs %.2f\n", i+1, m[i].itemname , m[i].price) ;
}
//printing extras
printf("\n--- EXTRAS ---\n");
for(int i = 10; i<= 11; i++){
    printf("%d. %15s Rs %.2f\n", i+1, m[i].itemname ,m[i].price);}
```

8. Implementation: Input Loop & Validation

Input Management and Safety Check

- The `while(1)` loop runs until the user signals the end of the order (inputting '`0`').
- Input variables `'ch'` (choice) and `'qty'` (quantity) capture the order details.
- A crucial `'if'` check validates that the item choice is within the valid range (1-12) to prevent crashes.
- If input is invalid, `'continue'` forces the loop to restart without adding to the bill.

```
//taking order
while(1){
    printf("\n enter your choice (t
    scanf("%d" ,&ch);

    if(ch == 0){           //loop ends
        break;
    }

    if(ch<1||ch>12){      //checkin
        printf("invalid choice\n");
        continue;
    }
}
```

9. Implementation: Final Bill & Discount

Discount and Final Output

- Once the ordering loop breaks, the program executes the financial logic.
- The 10% discount is applied ONLY if the accumulated `total` is 300 INR or higher.
- The final payable amount is calculated as `total - discount`.
- A detailed receipt is printed using `printf`, showing the total, discount applied, and final amount.

```
//calculating discount
if(total>=300){
    discount = total* 0.10;
}
//final bill calculated
final =total - discount;
//final bill printed with other details and a greeting
printf("\n ---- BILL ---- \n");
printf("total amount : %.2f\n" , total);
printf("discount: %.2f\n",discount);
printf("final amount :%.2f\n", final);

printf("\nthanks! visit again");
```

10. Testing and Results

- **Input Validation:** The program successfully rejected non-menu item choices.
- **Discount Test 1 (Below Threshold):** Total 290 INR resulted in 0 INR discount.
(Correct)
- **Discount Test 2 (Above Threshold):** Total 310 INR resulted in 31 INR discount.
(Correct)
- **Accuracy:** All item quantities and totals were added precisely.

```
----- BILL -----  
total amount : 290.00  
discount: 0.00  
final amount : 290.00
```

```
----- BILL -----  
total amount : 310.00  
discount: 31.00  
final amount : 279.00
```

II. Conclusion and Future Work

Summary

- Manual calculation errors are removed.
- Billing is now fast and accurate.
- Code structure is clean and organized.
- Discount logic works perfectly.

Future Scope

- Save bills to a file.
- Add a graphical interface (GUI).
- Add tax calculations.

12. References

- Class notes and course material
- Programming in C by Balagurusamy
- Online C programming tutorials (YouTube)