A menu-driven C program for a mart billing system handles customer invoices, including the process of generating invoices, saving them to a binary file, and retrieving them based on customer names.

### **Functionalities:**

- 1. A menu screen displaying the operations.
- 2. Generating new invoices.
- 3. View all the invoices.
- 4. View the invoice of a particular customer based on the customer's name.
- 5. Handling Invalid Inputs

#### **Source Code:**

```
#include<stdio.h>
#include<time.h>
#include<string.h>
#include<stdlib.h>

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

struct orders {
    char customer[50];
    char date[50];
    int numOfItems;
    struct items itm[50];
};
```

```
// Functions to generate bills
void generateBillHeader(char name[50], char date[30]) {
  printf("\n\n");
  printf("\t FRIENDS MART");
  printf("\n\t -----");
  printf("\nDate: %s", date);
  printf("\nInvoice To: %s", name);
  printf("\n");
  printf("-----\n");
  printf("Items\t\t\t\t");
  printf("Qty\t\t\t");
  printf("Total\t\t\t");
  printf("\n-----");
  printf("\n\n");
void generateBillBody(char item[20], int qty, float price) {
  printf("%s\t\t\t", item);
  printf("%d\t\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; // netTotal + cgst + sgst
  printf("-----\n");
  printf("Sub Total\t\t\%.2f", total);
  printf("\nDiscount @10%%\t\t\t%.2f", dis);
  printf("\n\t\t\t\----");
  printf("\nNet Total\t\t\t\%.2f", netTotal);
  printf("\nCGST @9%%\t\t\%.2f", cgst);
```

```
printf("\nSGST @9%%\t\t\%.2f", cgst);
  printf("\n----");
  printf("\nGrand Total\t\t\t%.2f", grandTotal);
  printf("\n----\n");
}
int main() {
  int opt, n, i;
  struct orders ord;
  struct orders order;
  char saveBill = 'y', contFlag = 'y';
  char name[50];
  FILE *fp;
  // Dashboard
  while (contFlag == 'y')  {
    system("cls");
    float total = 0;
    int invoiceFound = 0;
    printf("\t======FRIENDS MART======");
    printf("\n\nPlease select your preferred operation");
    printf("\n\n1.Generate Invoice");
    printf("\n2.Show all Invoices");
    printf("\n3.Search Invoice");
    printf("\n4.Exit");
    printf("\n\nYour choice:\t");
    fflush(stdin);
    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); //getting name
         ord.customer[strlen(ord.customer) - 1] = '\0';
```

```
time_t t=time(NULL);
          struct tm date=*localtime(&t);
          int yr=date.tm year+1900;
          int mon=date.tm mon+1;
          int d=date.tm mday;
          char year[10];
          char month[10];
          char day[10];
          char fulldate[20];
          snprintf(year, sizeof(year), "%d", yr);
          snprintf(month, sizeof(month), "%d/", mon);
          snprintf(day, sizeof(day), "%d/", d);
          strcpy(fulldate,day);
          strcat(fulldate,month);
          strcat(fulldate,year);
          strcpy(ord.date, fulldate); //getting current date
          printf("\nPlease enter the number of items:\t");
          fflush(stdin);
          scanf("%d", &n);
          ord.numOfItems = n;
          for (i = 0; i < n; i++) {
//
              fgetc(stdin);
             fflush(stdin);
                                              printf("\n\n");
            printf("Please enter the item %d name:\t", i + 1);
            fgets(ord.itm[i].item, 50, stdin);
            ord.itm[i].item[strlen(ord.itm[i].item) - 1] = '\0';
            printf("Please enter the quantity:\t");
            fflush(stdin);
                                              scanf("%d", &ord.itm[i].qty);
            printf("Please enter the unit price:\t");
            fflush(stdin);
                                              scanf("%f", &ord.itm[i].price);
            total += ord.itm[i].qty * ord.itm[i].price;
```

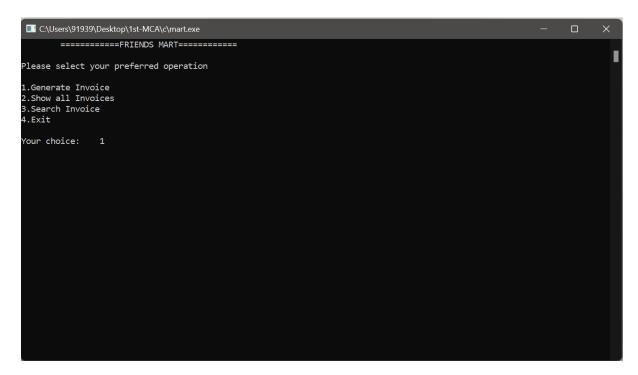
```
}
generateBillHeader(ord.customer, ord.date);
for (i = 0; i < ord.numOfItems; i++) {
  generateBillBody(ord.itm[i].item, ord.itm[i].qty, ord.itm[i].price);
}
generateBillFooter(total);
printf("\nDo you want to save the invoice [y/n]:\t");
fflush(stdin);
                           scanf("%s", &saveBill);
                           while(saveBill)
{
                                     if (saveBill == 'y' \parallel saveBill == 'Y') \{
            fp = fopen("MartBill.bin", "ab");
            if(fp==NULL){
                  printf("Error Accessing File:\n");
                  break;
                                              }
                                              else{
                                                        int flag=0;
                     flag=fwrite(&ord, sizeof(struct orders), 1, fp);
                      if (flag)
                        printf("\nSuccessfully saved");
                     else
                        printf("\nError saving");
                     fclose(fp);
                   break;
                  }
         }
         else if(saveBill == 'n' \parallel saveBill=='N'){
                  break;
                                     else{
```

```
printf("\nINVALID INPUT(please enter again)");
                                               printf("\nDo you want to save the invoice [y/n]:\t");
                    fflush(stdin);
                                               scanf("%s", &saveBill);
                                      }
 }
case 2:
  system("cls");
  fp = fopen("MartBill.bin", "rb");
  printf("\n *****Your Previous Invoices****\n");
  while (fread(&order, sizeof(struct orders), 1, fp)) {
    float tot = 0;
    generateBillHeader(order.customer, order.date);
    for (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("Enter the name of the customer:\t");
  fgets(name, 50, stdin);
  name[strlen(name) - 1] = 0;
  system("cls");
  fp = fopen("MartBill.bin", "rb");
  printf("\t*****Invoice of %s*****", name);
  while (fread(&order, sizeof(struct orders), 1, fp)) {
    float tot = 0;
    if (!strcmp(order.customer, name)) {
```

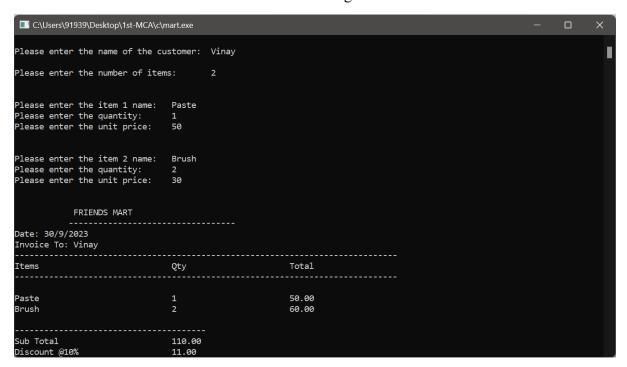
```
generateBillHeader(order.customer, order.date);
            for (i = 0; i < order.numOfItems; i++) {
               generateBillBody(order.itm[i].item, order.itm[i].qty, order.itm[i].price);
               tot += order.itm[i].qty * order.itm[i].price;
            }
            generateBillFooter(tot);
            invoiceFound = 1;
          }
       }
       if (!invoiceFound) {
          printf("\n\nSorry, the invoice for %s does not exist", name);
       fclose(fp);
       break;
    case 4:
       printf("\n\t Bye Bye :)\n\n");
       exit(0);
       break;
    default:
       printf("Sorry invalid option");
       break;
  }
  printf("\nDo you want to perform another operation?[y/n]:\t");
  fflush(stdin);
               scanf("%s", &contFlag);
}
printf("\n\t Bye Bye :)\n\n");
return 0;
```

}

# **OUTPUTS:**



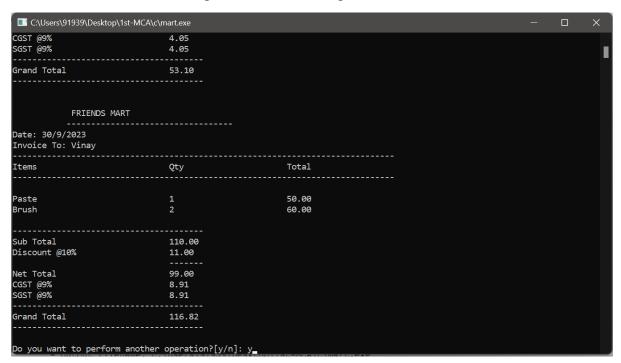
#### 1. Menu Page



2. Invoice Generation Page

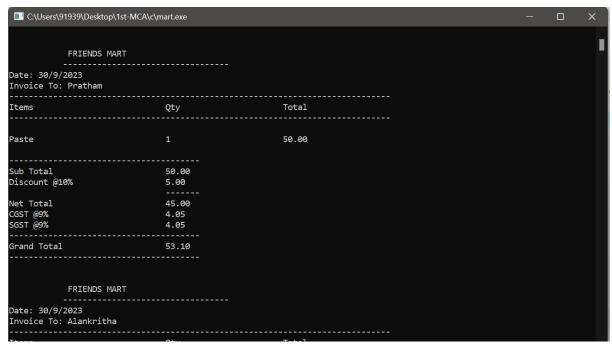


3. Invoice is generated and waiting for confirmation to save



4. Invoice is saved and waiting for the response to perform any other operation

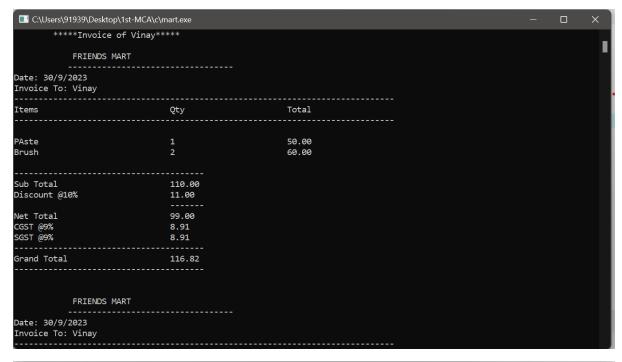
FRIENDS MART 					
ems	Qty	Total			
ste rush	1 2	50.00 60.00			
b Total	110.00				
scount @10%	11.00				
t Total	99.00				
ST @9%	8.91				
ist @9% 	8.91				
and Total	116.82				

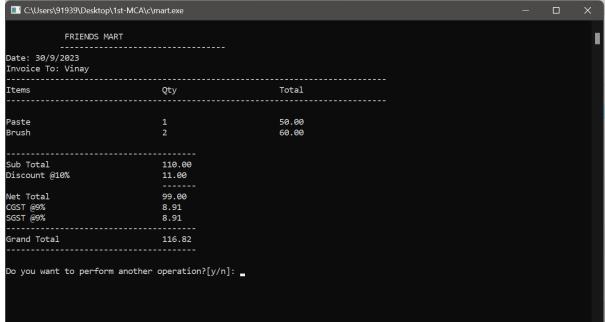




5. Viewing all invoices

6. Searching invoices of customer





7. All invoices of the customer

## 8. Entering an invalid input

9. Exiting the program