SUPERMARKET BILLING SYSTEM

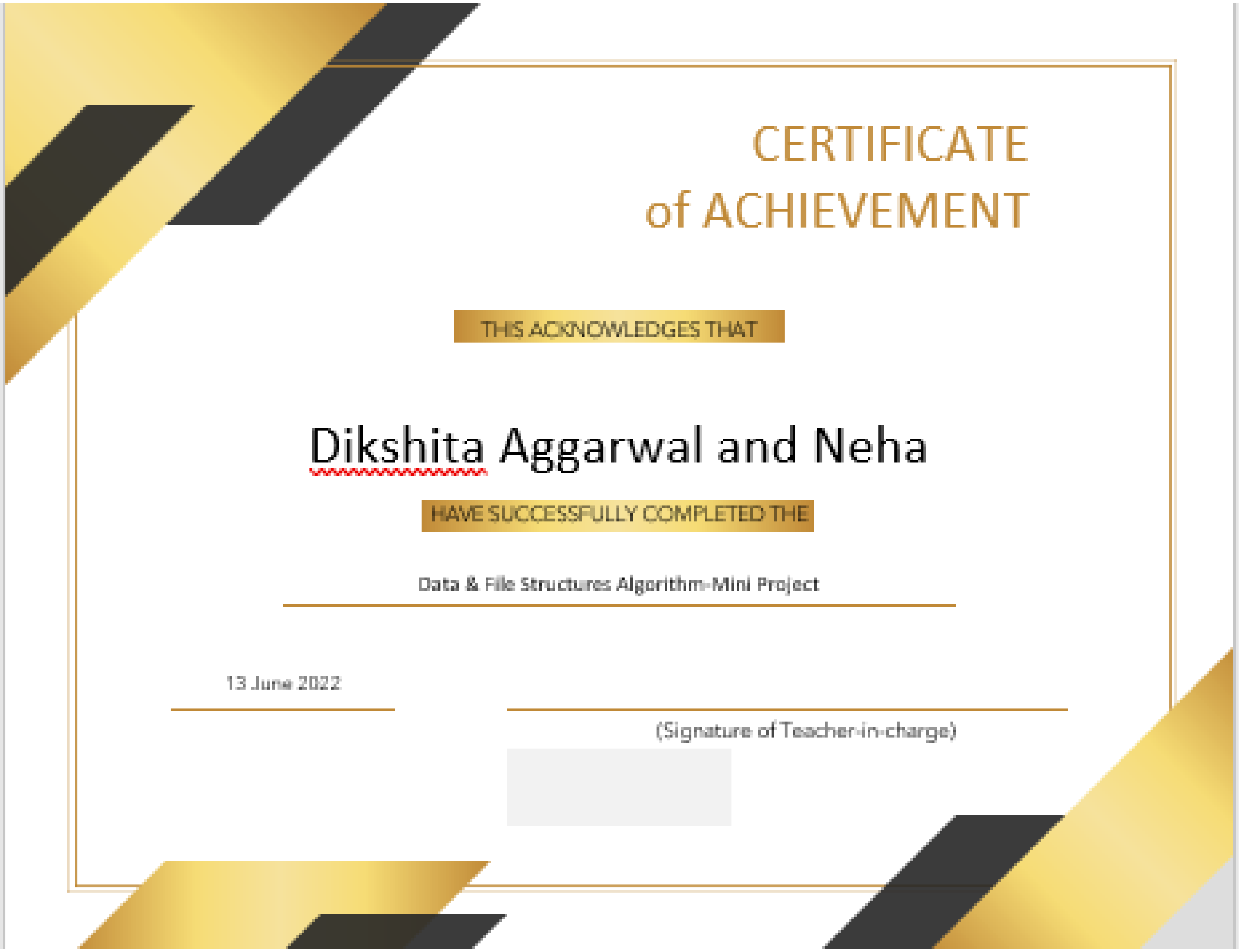
In C language

Submitted To:

Dr. M. Syamala Devi

Submitted By:

Dikshita(12) & Neha(27)
MCA I-Morning
2021-24



CERTIFICATE
of ACHIEVEMENT

THIS ACKNOWLEDGES THAT

Dikshita Aggarwal and Neha

HAVE SUCCESSFULLY COMPLETED THE

Data & File Structures Algorithm-Mini Project

13 June 2022

(Signature of Teacher-in-charge)

ACKNOWLEGDEMENT

It is with great pleasure that we find myself penning down these lines to express my sincere gratitude to various people who helped me along in completing this project.

The nurturing and positive environment provided by our teachers at the University encouraged us to work hard and perform well. Thanks to the guidance and mentorship provided by Syamala ma’am that we were able to complete it on time and with utmost sincerity.

We’d also like to thank our parents and peers. They’ve been immensely helpful and supportive throughout the project.

ABSTRACT

Supermarkets tend to use a software for their stock and to store bills. This is an abstract model of the billing system they use. In this application, we allow store keepers to enter the details of the products available in their store, to view them, to delete them, and prepare bills.

The **features** of this software include:

- The supermarket owner can perform various operations like:
  - Maintaining an inventory (or details of the stock available in the store)
  - Add products to the inventory
  - Delete or Modify products from the inventory
  - Display the products in the inventory

- The customer can perform operations like:
  - Add items to bill
  - Calculate the Total amount payable
  - Display the bill of the corresponding customer.

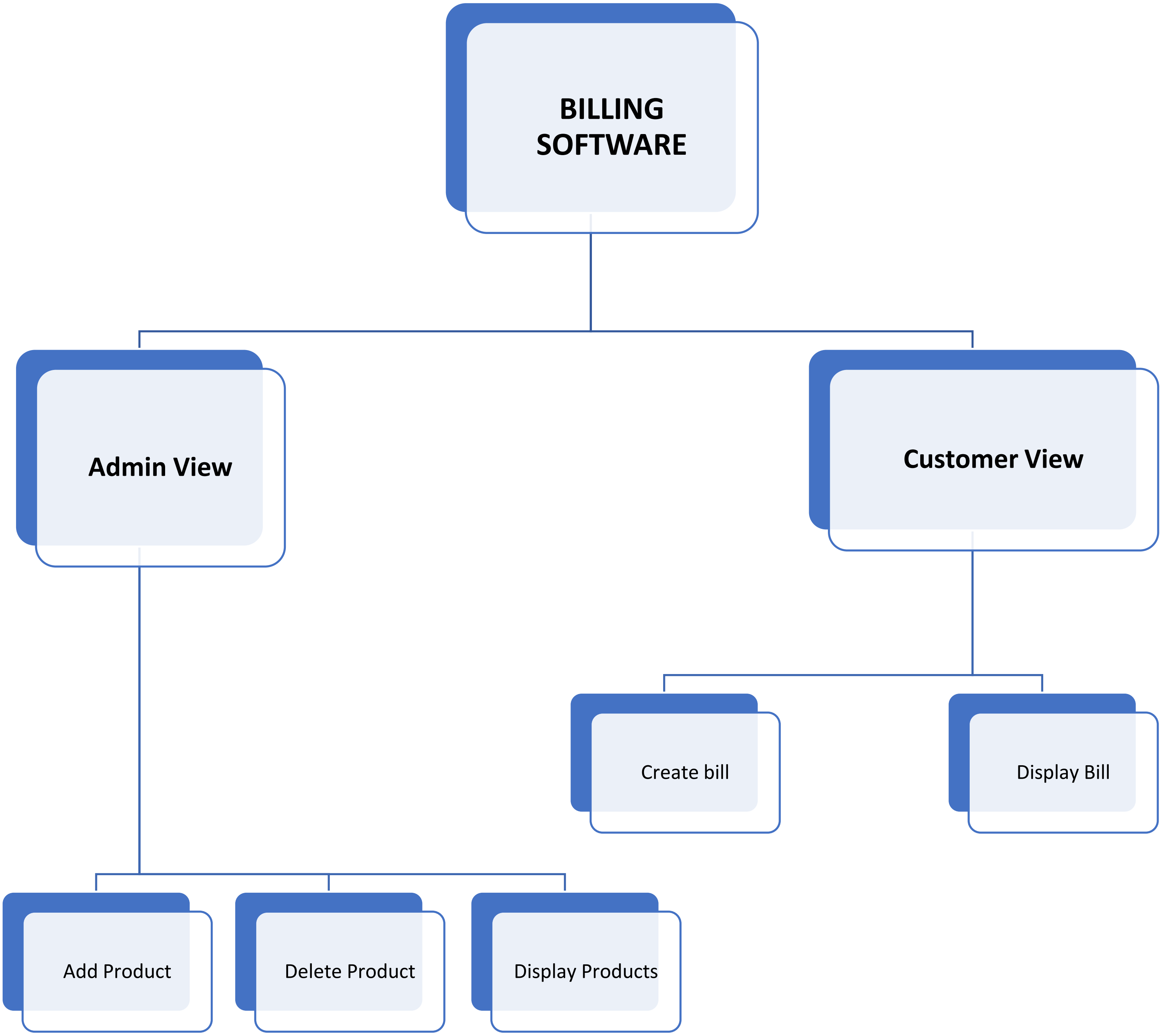
TABLE OF CONTENTS

INTRODUCTION

The purpose of creating a Supermarket Billing system is to keep track of the inventory of the store so that every time a customer tries to buy something the stock of that particular reduces by the quantity purchased by the user and is later decremented in the stock as well. Hence, it helps in not only maintaining an inventory or to create bills, it helps in keeping a close eye on the stock. The owner can any time display the inventory to check for the availability of products.

If the owner doesn’t use this software for billing and maintaining inventories then s/he will have to make the bills by hand and memorize the stock availability. It’s difficult to keep of the inventory if you have large number of items available in the store and a huge sale. Using the software, you can easily check the products available and help the customers with it. This not only saves time but also increases the efficiency with which the store operates.

PROJECT DESIGN



ADMIN MODULES:

Before entering admin view, the user has to enter a password to have access to the database. The admin view is used to create an inventory of the stock and store information of the products available in the Supermarket.

The following modules are there in the Admin view:

1. Add Product
2. Delete Product
3. Display Product

CUSTOMER MODULES:

The following modules are there in the Admin view:

1. Make Bill
2. Display Bill

FEATURES:

- To avoid unauthorized access to the inventory, admin view is password protected
- 2 files are used to store the number of products and the inventory
- Data Structures used: Arrays

DATABASE DESIGN:

There are 2 files used in this system:

- **Quantity.txt:** It is a simple text file that stores nothing but the number of products stored in the inventory. This is used in either read or write mode. Whenever used in read mode, it is used to know the number of products available, and when used in write mode, it's used to add or reduce the quantity on addition or deletion of products.

- **Records.csv:** A CSV file, aka comma separated values file, is used to make an excel file that stores details in a tabular form. This file is used to store details about each product like the product name, price and quantity available. The file is opened in read mode to display the inventory and in write mode, to add or modify product details stored in inventory.

HEADER FILES:

- **stdio.h**: It stands for Standard Input Output. It has the information related to input/output functions.
  Example: *printf(), scanf(), getc(), putc(), fopen(), fclose(), remove(), fflush().*

- **conio.h**: It is mostly used by MS-DOS compilers to provide console input and output.
  Example: *clrscr()*, *cgets()*, *getch()*, *getche()*, *kbhit()*, *putch(), cputs()*

- **string.h**: It defines one variable type(size_t), one macro(NULL), and various functions for manipulating arrays of characters.
  Example: *strcmp(), strncmp(), strcat(), strncat(), strcpy(), strlen()*

- **ctype.h**: It declares a set of functions to classify (and transform) individual characters.
  Example: *isupper(), isalpha(), isdigit(), islower(), tolower(), toupper()*

- **stdlib.h**: header of the general purpose standard library of C programming language which includes functions involving memory allocation, process control, conversions and others.
  Example: *malloc(), free, atoi(), abort(), exit(), abs(), system("cls")*

IMPLEMENTATION

- **SOFTWARE REQUIREMENTS**
    - Operating System: DOS based
    - Language: C language
    - Compiler: Dev C++

- **HARDWARE REQUIREMENTS**
    - RAM: 8 GB (baseline)
    - Secondary Memory:
        - 1.81 MB(for project)
        - 69 MB(for compiler)
    - Processor: i3 or i5 (minimum)

- **DETAILS**
    - Files Associated:
        - Main file: BILLING SOFTWARE.C
        - Executable file: BILLING SOFTWARE.exe
        - Text File: Quantity.txt
        - Inventory: Records.csv
    - Password: admin (for admin view)

- **PROGRAM CODE**

```
#include<stdio.h>
#include<string.h>
#include<ctype.h>
#include<conio.h>
#include<stdlib.h>
#include<graphics.h>
#include<time.h>

int n; //used for Qauntity.txt
int ti=0; //customer index


//This structure stores details of products available at the
supermarket
struct item
{
        int productno;
        char productname[20];
        int quantity;
        int price;
}item;

//Stores customer's cart
struct customer
{
        int productno;
        char productname[20];
        int quantity;
        int price;
        int amount;
}cst[100];

//Function to add products in inventory and file
void create()
{
        int i=0,rec_n;
        FILE *fp,*fpq;
        fp=fopen("Records.csv","a");
        fpq=fopen("Quantity.txt","w");
        if(fp==NULL)
        {
            printf("Error\n");
        }
        printf("\n\n\n\n\t");
        printf("Enter the Number of Records: ");
        scanf("%d",&rec_n);
        fscanf(fpq,"%d",&n);
        n=n+rec_n;
        fprintf(fpq,"%d",n);
        printf("\n");
        fclose(fpq);
        while(i<rec_n)
        {
            printf("\t");
                printf("Enter Product Code:");
                scanf("%d",&item.productno);
                printf("\t");
                printf("Enter Product Name:");
                scanf("%s",item.productname);
                printf("\t");
                printf("Enter Quantity:");
                scanf("%d",&item.quantity);
                printf("\t");
                printf("Enter Price:");
                scanf("%d",&item.price);
                printf("\n");
                i++;
                fprintf(fp,"\n%d %s %d
%d",item.productno,item.productname,item.quantity,item.price);
        }
        fclose(fp);
}


void display()
{
        int record,read;
        read=0;
        FILE *fp;
        fp=fopen("Records.csv","r");
        record=0;
        if(feof(fp))
        {
                printf("Error in Opening Files\n");
        }
        printf("\t");
        printf("=================================================
===============\n");
        printf("\t");
        printf("|Product Number\t|Product
Name\t|Quantity\t|Price\t\n");
        printf("\t");
        printf("=================================================
===============\n");
        while(!feof(fp))
        {
                read=fscanf(fp,"%d %s %d
%d\n",&item.productno,item.productname,&item.quantity,&item.pri
ce);
                if(read==4){
                        record++;
                        printf("\n\t
%14d|\t%8s|\t%8d|\t%d|\n",item.productno,item.productname,ite
m.quantity,item.price);
        printf("\n");
       }

                if(read!=4 && !feof(fp)){
                        printf("\n\n\tFormat incorrect...\n\n\n");
                fclose(fp);
                return;
            }
        }
        fclose(fp);
}

void edit()
{
        FILE *fp;
        FILE *fptr;
        int h;
        int found=0;
        int n=0;
        fp=fopen("Records.csv","r");
        fptr=fopen("temp1.csv","w");
        printf("\t");
        printf("Enter the Product Number to Delete: ");
        scanf("%d",&h);
        printf("\n");
        while(!(feof(fp)))
        {
            n++;
            fscanf(fp,"%d",&item.productno);
            if(item.productno==h)
            {
                found=1;
                fscanf(fp,"%s",item.productname);
                fscanf(fp,"%d",&item.quantity);
                fscanf(fp,"%d",&item.price);

                printf("\t");
                printf("Product has been Successfully Deleted\n");
            }
            else
            {
                fscanf(fp,"%s",item.productname);
                fscanf(fp,"%d",&item.quantity);
                fscanf(fp,"%d",&item.price);
                fprintf(fptr,"%d %s %d %d
",item.productno,item.productname,item.quantity,item.price);
            }
        }
        fclose(fptr);
        fclose(fp);
        if(found==0)
        {
          printf("\t");
            printf("Product no Found");
            printf("\n");
            printf("\n");
        }
        remove("Records.csv");
        rename("temp1.csv","Records.csv");
}
```

```
void bill()
{
        int na=0;  //used for product number in scanf function
        int a=0;    //quantity of a product
        int cont=0; //counter
        int k=0; //continue
        int r=0;
        int w=0;
        int ch=0;
        int l=0; //line counter in file
        int bi=0;
        int ln=0; //repition of product number
        int read;
    printf("\n\t");
        l3:
        printf("Enter the Product Number: ");
        scanf("%d",&na);
        printf("\n");
        FILE *fp;
        fp=fopen("Records.csv","r+");
        FILE *fptr;
        fptr=fopen("temp.csv","w");
        if(fp==NULL || fptr==NULL){
                printf("Error opening file");
                return;
        }
        while(!(feof(fp)))
        {
```

```
        l++;
        fscanf(fp,"%d %s %d
%d\n",&item.productno,item.productname,&item.quantity,&item.pri
ce);
        if(item.productno == na)
        {
            ln++;
            if(ln==2)
            break;

            printf("Enter the Quantity: ");
            scanf("%d",&a);
            if(a <= item.quantity)
            {
                cst[ti].productno=item.productno;
                strcpy(cst[ti].productname,item.productname);
                cst[ti].quantity=a;
                cst[ti].price=item.price;
                item.quantity=item.quantity-a;
                ti++;

            }
            else
            {
                printf("Sorry Out of Stock\n");
                bi++;

            }
        }
        else
        {
            cont++;
            fprintf(fptr,"%d ",item.productno);
            fprintf(fptr,"%s ",item.productname);
            fprintf(fptr,"%d ",item.quantity);
            fprintf(fptr,"%d \n",item.price);
            continue;
        }
        fprintf(fptr,"%d ",item.productno);
        fprintf(fptr,"%s ",item.productname);
        fprintf(fptr,"%d ",item.quantity);
        fprintf(fptr,"%d \n",item.price);
    }
    if(l==cont)
    printf("Error:Choose Properly\n");
    fclose(fptr);
    fclose(fp);
    remove("Records.csv");
    rename("temp.csv","Records.csv");
    ln=0;
    printf("Do you Want to Shop More:\nPress 1 for More\nPress 2 to Exit\t");
    scanf("%d",&k);
    if(k==1)
    goto l3;
}
```

```
void invoice()
{
        int j;
        int o;
        float gt=0;
        if(ti==0)
        {
                printf("There are No Products in your Cart");
                printf("\n");
        }
        printf("\n\n");
        printf("****************************INVOICE************
**************");
        printf("\n\n");
        printf("product number\t| product name\t| price\t| quantity\t|
amount\t|\n");
        printf("**************************************************
**************\n");
        for(o=0;o<ti;o++)
        {
                cst[o].amount=0;
                cst[o].amount=((cst[o].price)*(cst[o].quantity));
                gt=gt+cst[o].amount;
        }
        for(j=0;j<ti;j++)
        {
                printf("%6d\t      |%9s   \t  | %3d\t|%3d\t       |
%3d\n",cst[j].productno,cst[j].productname,cst[j].price,cst[j].quantity
,cst[j].amount);
        }
        printf("\nAmount Payable:%f",gt);
        printf("\n");
        printf("Thank you for Shopping");
        printf("\n");
        printf("Visit Again");
        printf("\n");
        int key;
        printf("Press any key to return to main menu\t");
        scanf("%d",&key);
}


int main()
{
        //FRONT PAGE
        int gd=DETECT, gm;
        initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
        unsigned int x_hours=0;
        unsigned int x_minutes=0;
        unsigned int x_seconds=0;
        unsigned int x_milliseconds=0;
        unsigned int
totaltime=0,count_down_time_in_secs=0,time_left=0;
```

```
        clock_t x_startTime,x_countTime;
        count_down_time_in_secs=30;
        x_startTime=clock();  // start clock
    time_left=count_down_time_in_secs-x_seconds;
        //setting x,y coordinates to the center
        int x_coord=150;
        int y_coord=0;


        //setting text style
        settextstyle(8,0,4);
        //print text
        outtextxy(x_coord,y_coord,"SUPERMARKET BILLING SYSTEM");
        settextstyle(3,0,2);
        y_coord+=50;
        x_coord+=200;
        outtextxy(x_coord,y_coord,"by Dikshita and Neha");
        y_coord+=30;
        x_coord=0;
        //increment timer upto 30 seconds and then close front page
        while (time_left>0)
        {
                x_countTime=clock();
                x_milliseconds=x_countTime-x_startTime;
                x_seconds=(x_milliseconds/(CLOCKS_PER_SEC))-
(x_minutes*60);
                x_minutes=(x_milliseconds/(CLOCKS_PER_SEC))/60;
                x_hours=x_minutes/60;

                time_left=count_down_time_in_secs-x_seconds;
        }

        closegraph();


        // MAIN SYSTEM
        char ch,password[10],q[10]="admin";
        printf("\n");
        int i;
        int s=0; //choice for view
        int y=0; //choice for admin view
        //MENU


    q:
        printf("\n\t"); //defining label 'q' to go back to main menu
without having to reenter code again
    printf("*****************************");
    printf("\n");
    printf("\t");
    printf("Please select an option: ");
    printf("\n");
    printf("\t");
    printf("*****************************");
    printf("\n\n");
    printf("\t");
        printf("01. ADMIN VIEW");
        printf("\n\t");
        printf("02. CUSTOMER VIEW");
        printf("\n\t");
        printf("03. EXIT APP");
        printf("\n");
        printf("\n\t");
        printf("Please Enter an Option: ");
        scanf("%d",&s);

        switch(s)
        {
            case 1: //ADMIN VIEW
                        printf("\tEnter Password:\t");
           for(i=0;i<5;i++)
           {
             ch = getch();
             password[i] = ch;
             ch = '*' ;
              printf("%c",ch);
                        }
           password[i]='\0';

           //check if password is correct
           if(strcmp(password,q))
           {
                system("cls");
        printf("\n\t");
        printf("Wrong Password Please Try Again");
        printf("\n\n");
        goto q;
    }
    //If password correct, enter Admin view
    else
       system("cls");
       goto l3;
       l3: printf("\n\t");
       printf("Access Granted\n");
       printf("\n\n");
       printf("\t");
       printf("****************************");
       printf("\n");
       printf("\t");
       printf("Please Select an Option: ");
       printf("\n");
       printf("\t");
       printf("****************************");
       printf("\n\n");
                printf("\t01.ADD PRODUCTS TO INVENTORY");
                printf("\n\t");
                printf("02.DELETE PRODUCTS IN INVENTORY");
                printf("\n\t");
                printf("03.DISPLAY PRODUCTS IN INVENTORY");
                printf("\n\t");
                printf("04.GO BACK TO MAIN MENU");
        printf("\n");
        printf("\n");
        printf("\t");
        printf("Please Enter an Option: ");
        scanf("%d",&y);
        switch(y)
           {
         case 1:
              system("cls");
                        create();
                        system("cls");
                        printf("\t");
                        printf("Records are Created");
                        printf("\n\n");
           goto q;

                   case 2:
                        system("cls");
                        edit();
                goto q;
                   case 3:
                        system("cls");
                        display();
       goto q;
    case 4:
                        system("cls");
                     goto q;
     default :
```

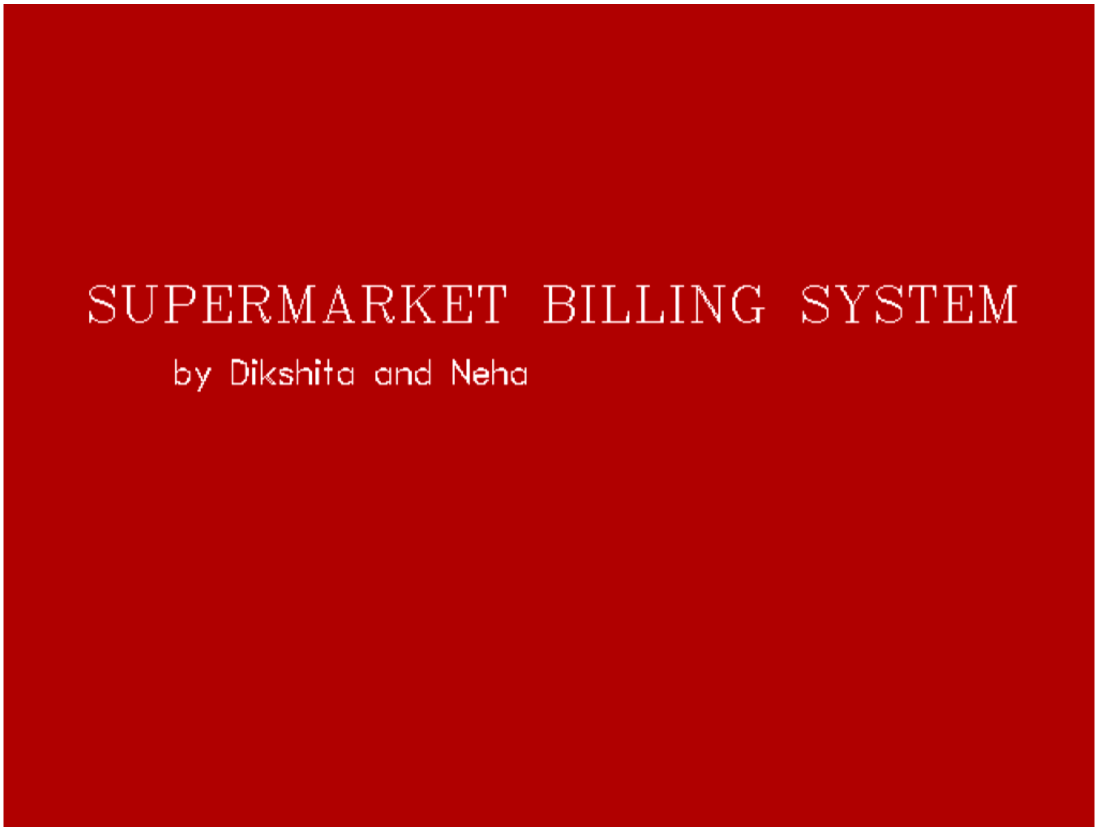
```
                system("cls");
                                        printf("error\n");
                                 goto q;
}
                break;



            case 2:
               system("cls");
                    printf("\n");
    printf("\t");
    printf("****************************");
    printf("\n");
    printf("\t");
    printf("Please Select an Option: ");
    printf("\n");
    printf("\t");
    printf("****************************");
    printf("\n");
    printf("\t");
    printf("01.MAKE BILL");
    printf("\n\t");
    printf("02.SHOW BILL");
    printf("\n\t");
    printf("03.MAIN MENU");
    printf("\n\n\n\t");
    printf("Please Enter an Option: ");
    scanf("%d",&y);
```

```
printf("\n");
switch(y)
          {
    case 1 :
                   system("cls");
                                            bill();
                                            system("cls");
                  goto q;

    case 2 :
                   system("cls");
                                            invoice();
                                            system("cls");
            goto q;

    case 3 :
                                            goto q;

    default :
                                            printf("Error\n");
            goto q;
   }
           break;

      case 3: //EXIT
                   printf("\tExiting...\n");
                   break;
```

```
                default: //INVALID VIEW
                        printf("\tInvalid Choice\n");
                    break;
        }
        return 0;
}
```

TESTING AND VALIDATION

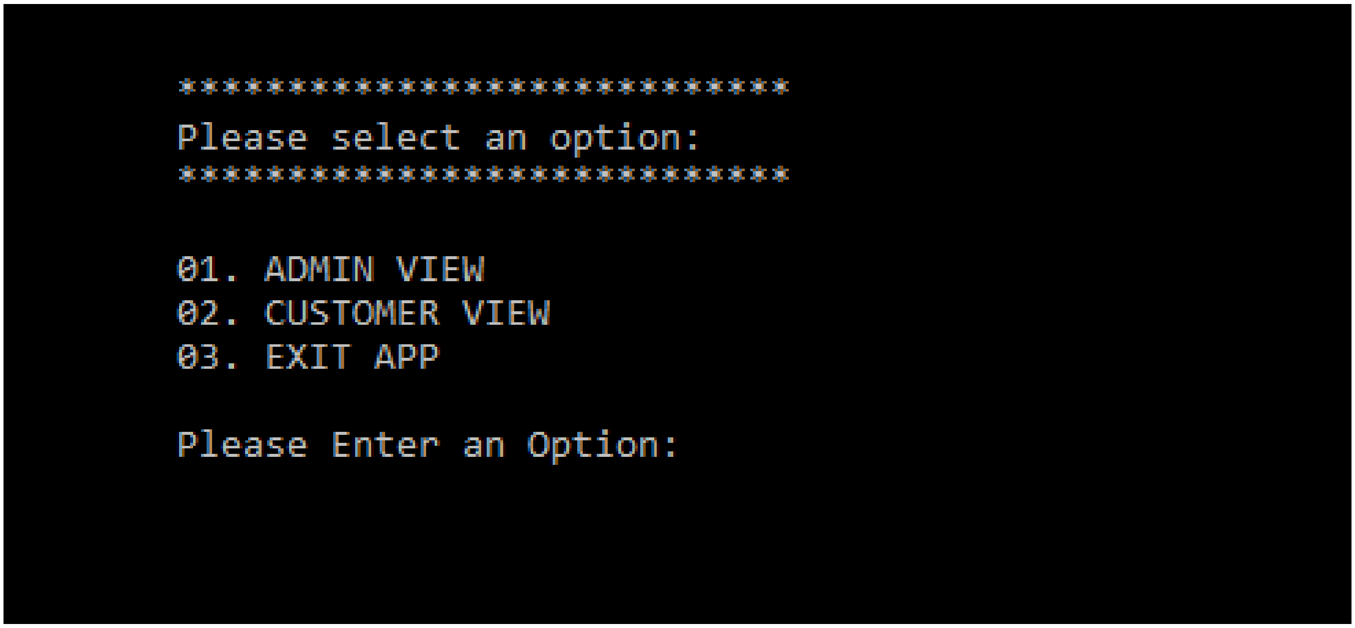
MAIN PAGE



SUPERMARKET BILLING SYSTEM

by Dikshita and Neha

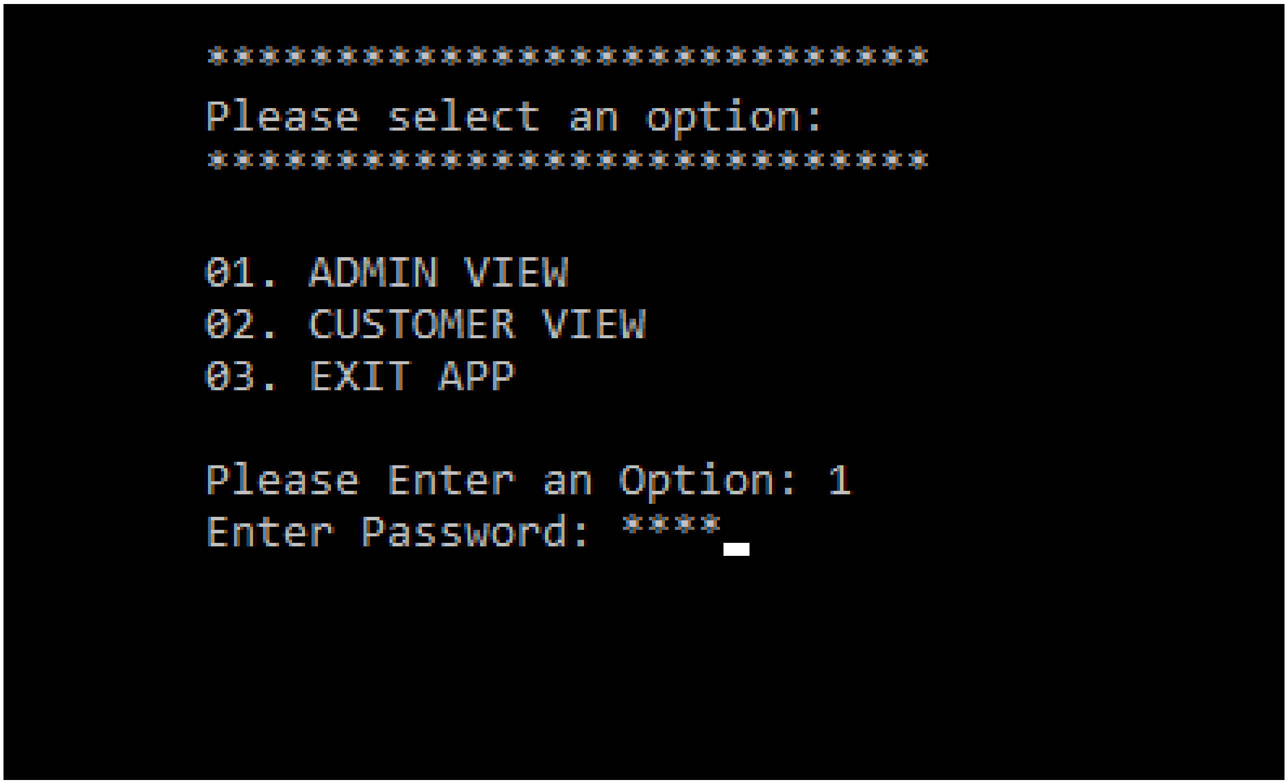
After 30 seconds……



```
****************************
Please select an option:
****************************


01. ADMIN VIEW
02. CUSTOMER VIEW
03. EXIT APP

Please Enter an Option:
```

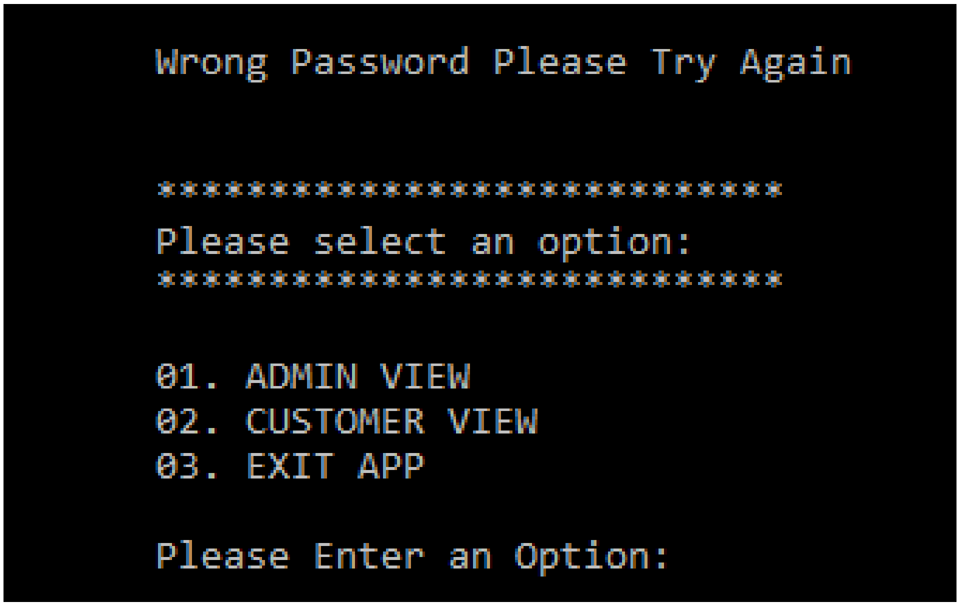
ADMIN VIEW

- ENTER PASSWORD:



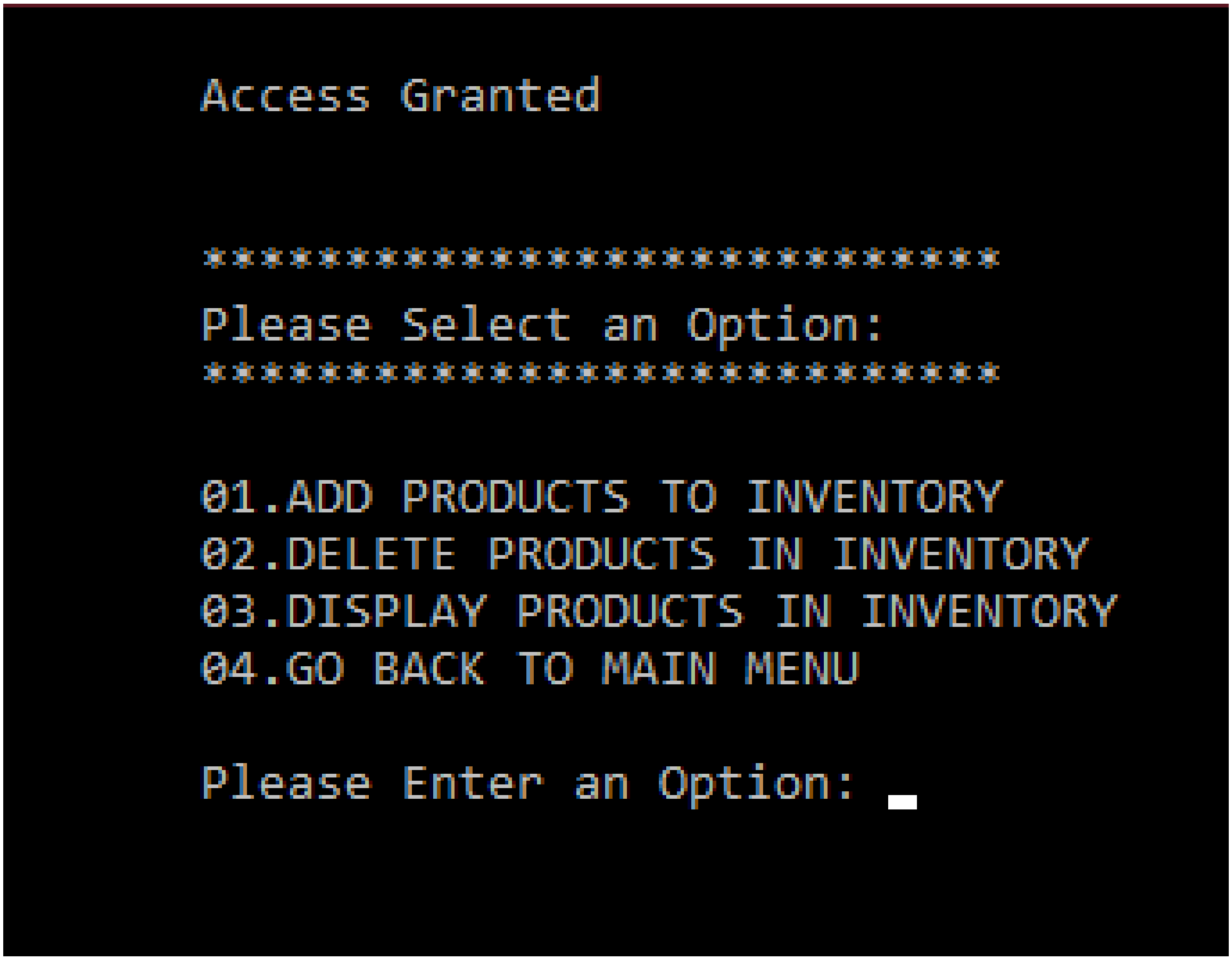
```
****************************
Please select an option:
****************************


01. ADMIN VIEW
02. CUSTOMER VIEW
03. EXIT APP

Please Enter an Option: 1
Enter Password: ****_
```

- INCORRECT PASSWORD



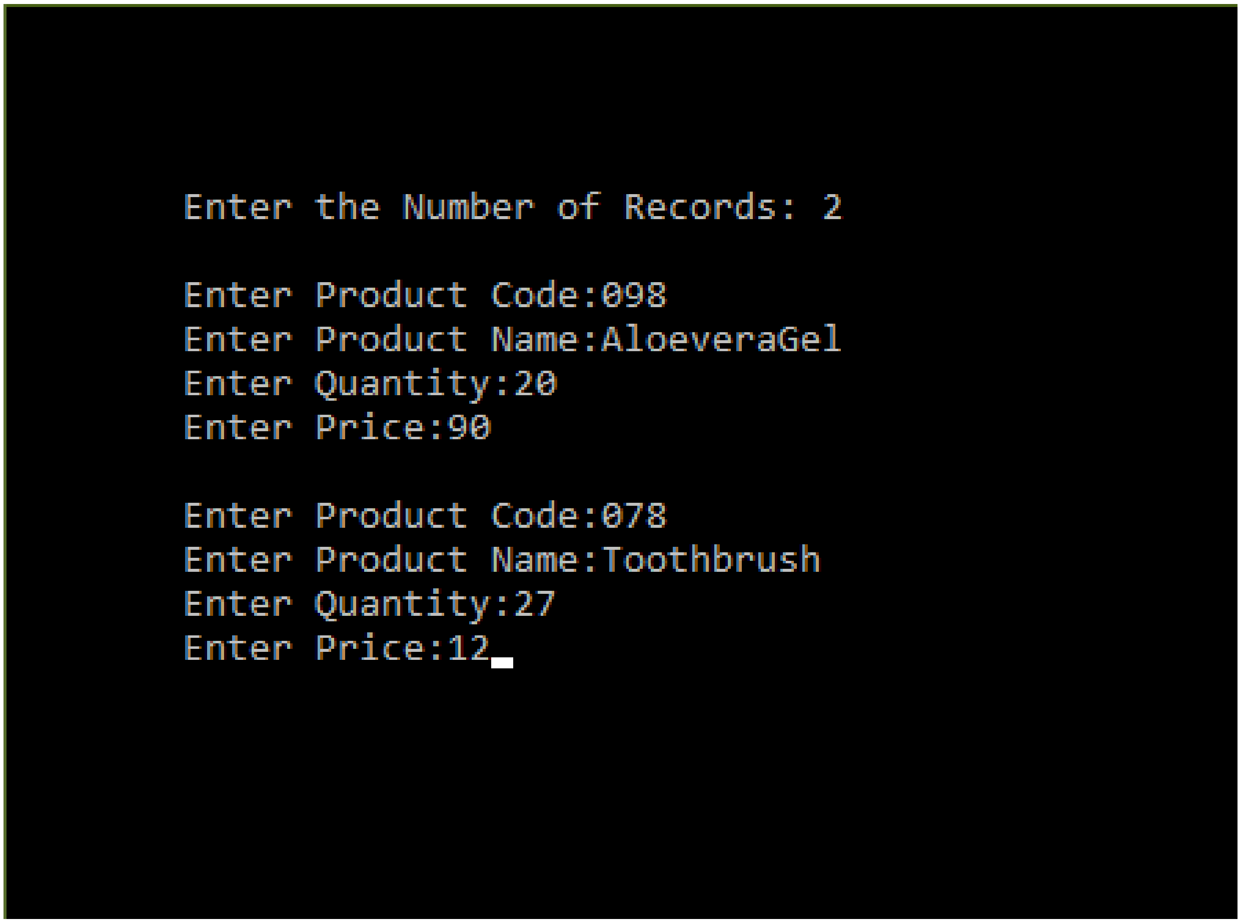
```
Wrong Password Please Try Again


****************************
Please select an option:
****************************


01. ADMIN VIEW
02. CUSTOMER VIEW
03. EXIT APP

Please Enter an Option:
```

- <u>IF VALID PASSWORD IS ENTERED, ADMIN MENU IS OPENED:</u>



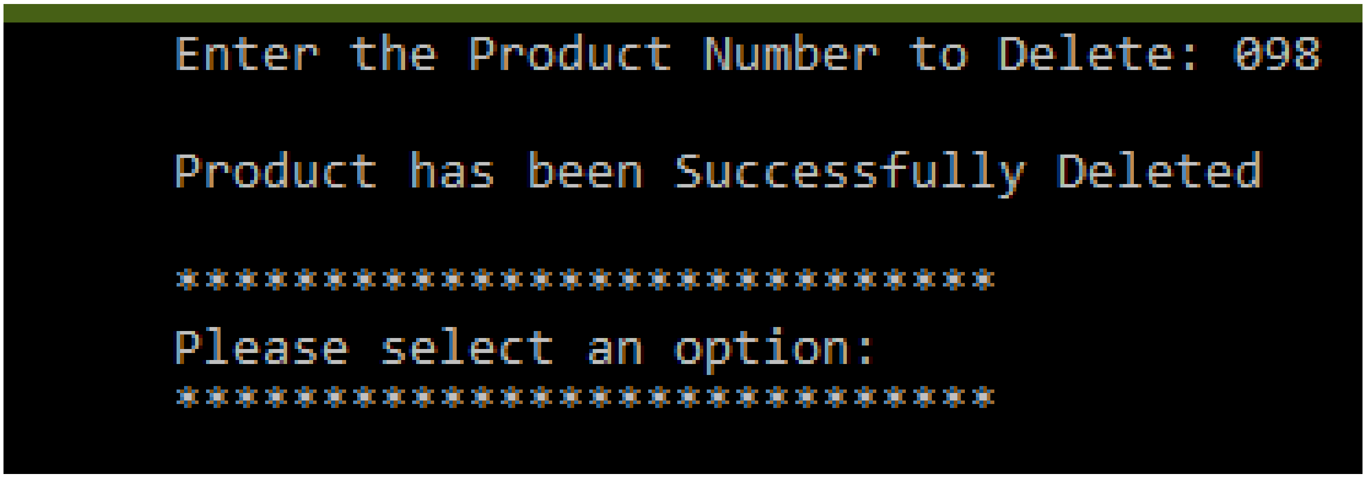
```
Access Granted


****************************
Please Select an Option:
****************************


01.ADD PRODUCTS TO INVENTORY
02.DELETE PRODUCTS IN INVENTORY
03.DISPLAY PRODUCTS IN INVENTORY
04.GO BACK TO MAIN MENU

Please Enter an Option: _
```

- ADD PRODUCTS IN INVENTORY



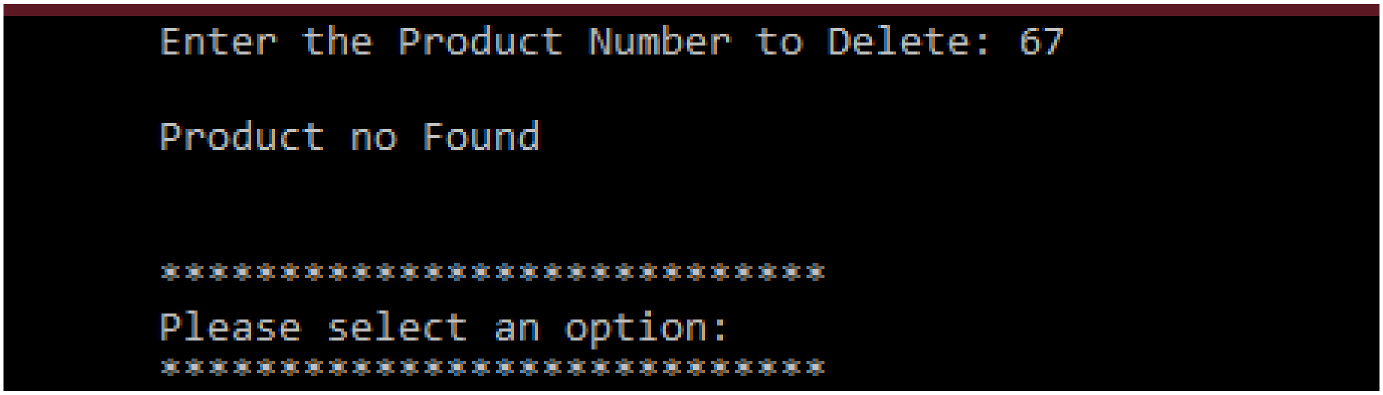
```
Enter the Number of Records: 2

Enter Product Code:098
Enter Product Name:AloeveraGel
Enter Quantity:20
Enter Price:90

Enter Product Code:078
Enter Product Name:Toothbrush
Enter Quantity:27
Enter Price:12_
```

- DELETE PRODUCTS FROM INVENTORY



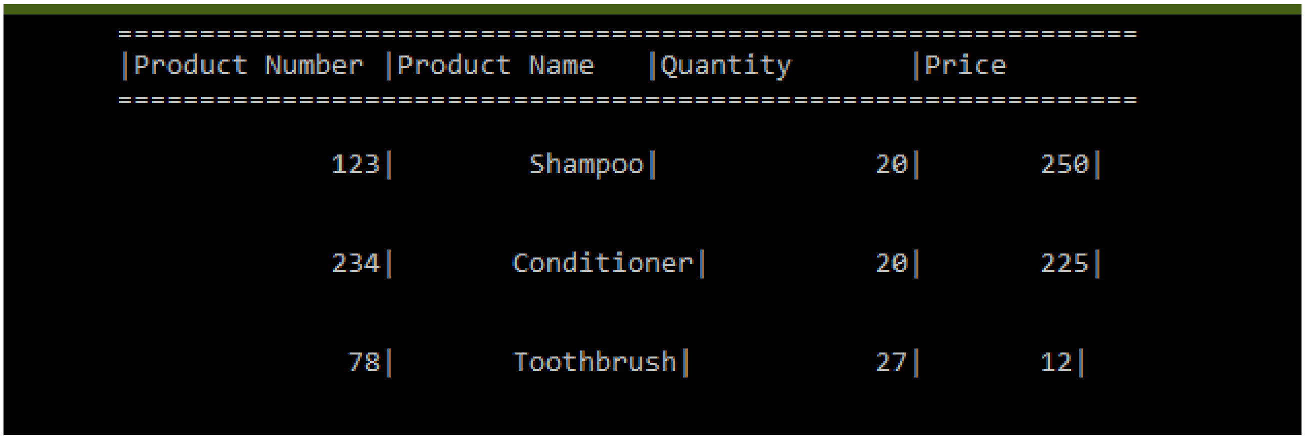
```
Enter the Product Number to Delete: 098

Product has been Successfully Deleted

*****************************

Please select an option:
*****************************
```

- <u>IF WRONG PRODUCT ID ENTERED:</u>



```
Enter the Product Number to Delete: 67

Product no Found


****************************

Please select an option:
****************************
```

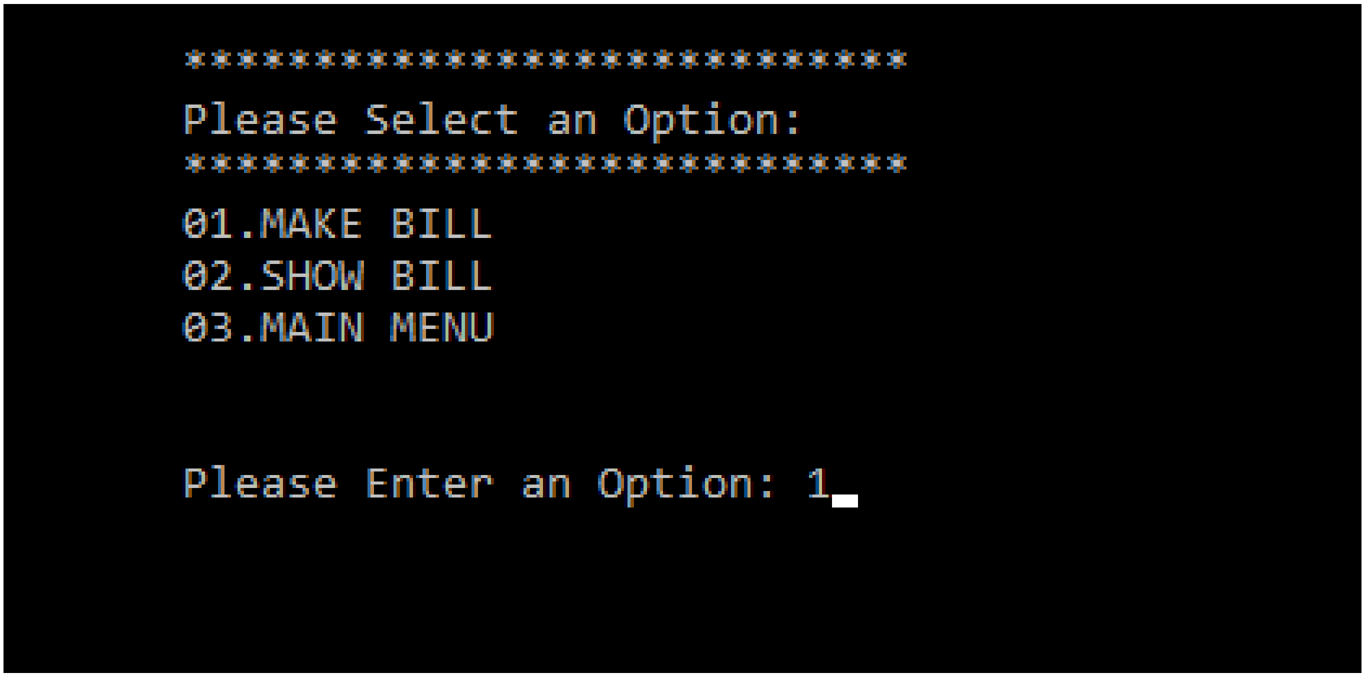
- <u>SHOW PRODUCTS IN INVENTORY</u>



```
============================================================
|Product Number |Product Name  |Quantity      |Price
============================================================

             123|       Shampoo|             20|       250|


             234|      Conditioner|          20|       225|


              78|      Toothbrush|           27|       12|
```

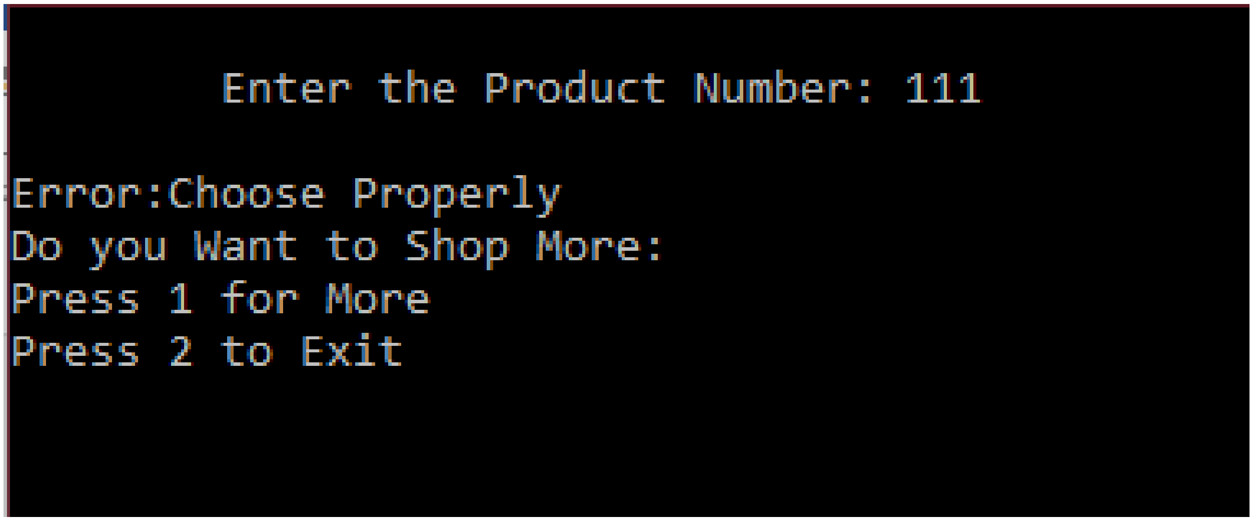
CUSTOMER VIEW:

- CUSTOMER MENU



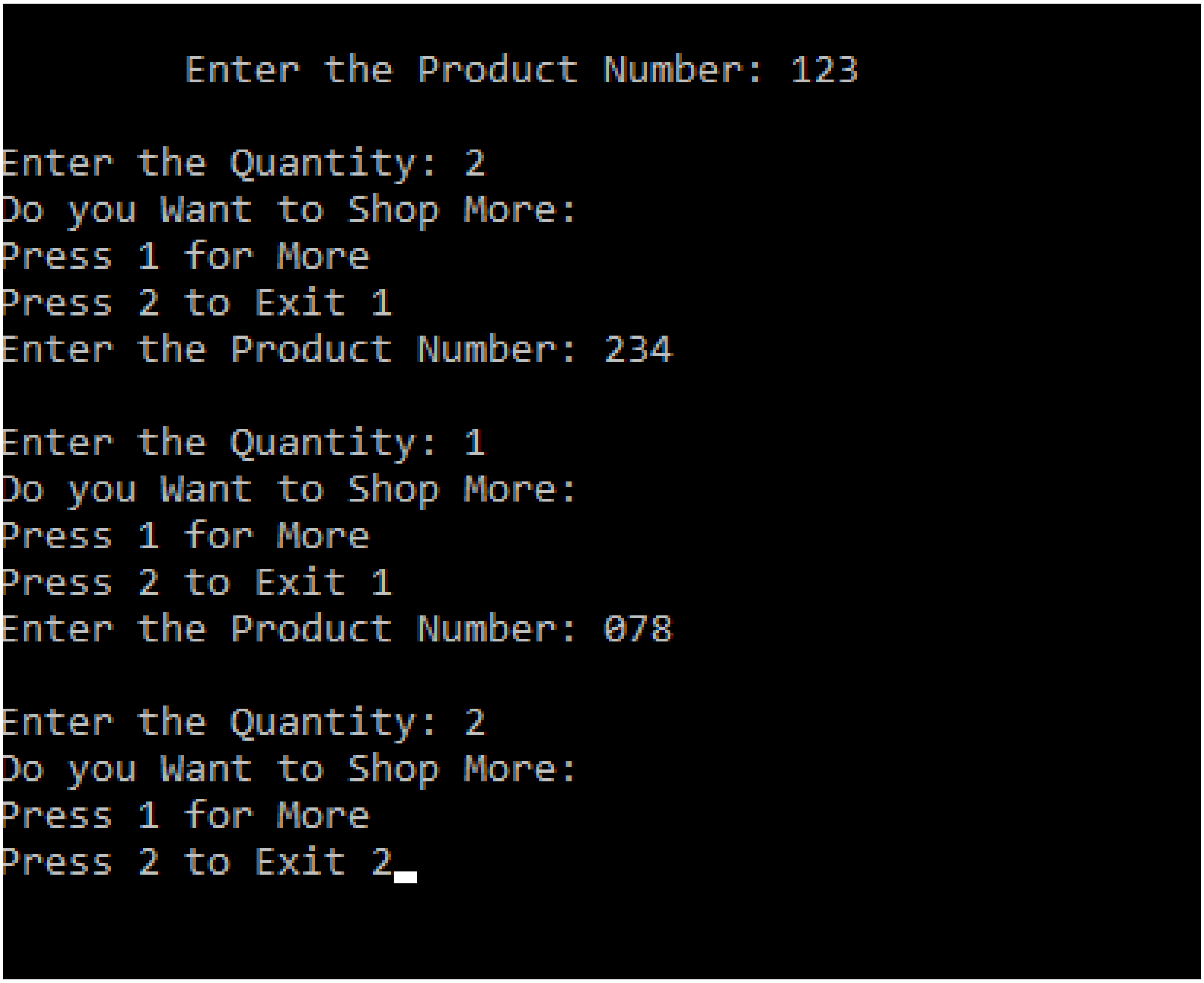
```
****************************
Please Select an Option:
****************************
01.MAKE BILL
02.SHOW BILL
03.MAIN MENU


Please Enter an Option: 1_
```

- INCORRECT PRODUCT ID IN MAKE BILL



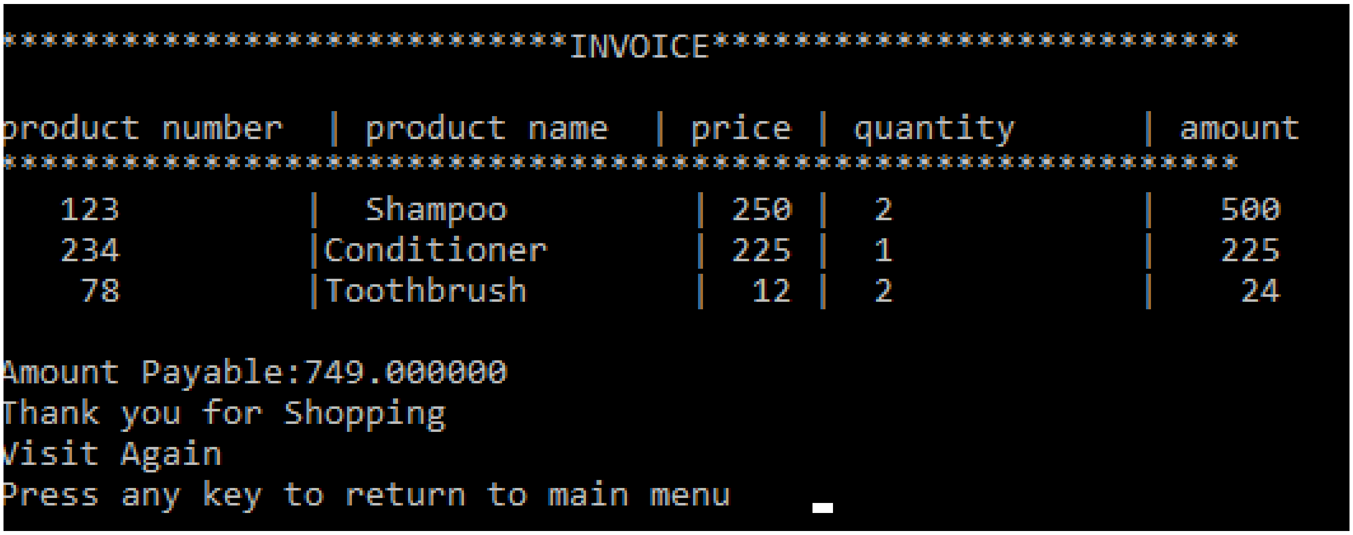
```
        Enter the Product Number: 111

Error:Choose Properly
Do you Want to Shop More:
Press 1 for More
Press 2 to Exit
```

- MAKE BILL



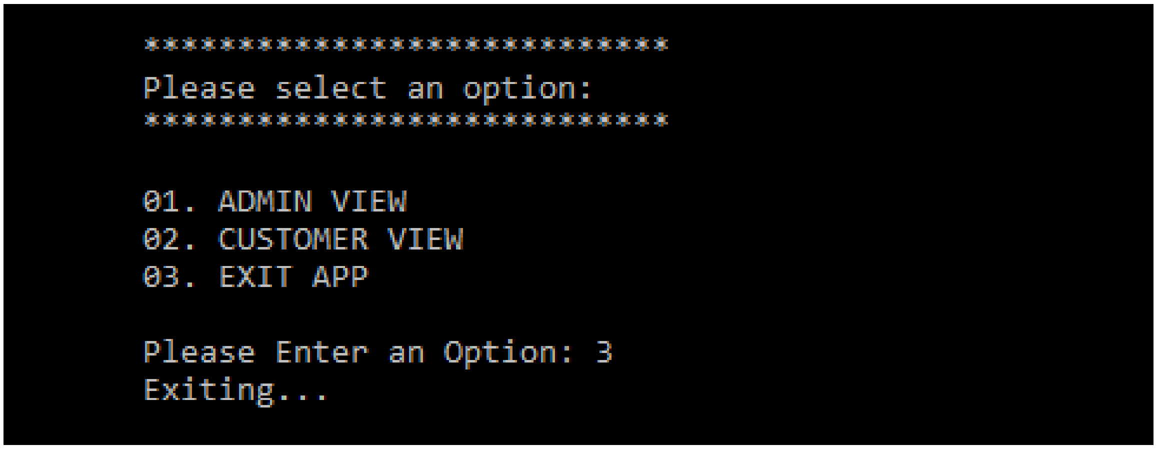
```
        Enter the Product Number: 123

Enter the Quantity: 2
Do you Want to Shop More:
Press 1 for More
Press 2 to Exit 1
Enter the Product Number: 234

Enter the Quantity: 1
Do you Want to Shop More:
Press 1 for More
Press 2 to Exit 1
Enter the Product Number: 078

Enter the Quantity: 2
Do you Want to Shop More:
Press 1 for More
Press 2 to Exit 2
```

- SHOW BILL



```
****************************INVOICE**************************

product number  | product name   | price | quantity      | amount
**************************************************************
   123         |  Shampoo          | 250 |  2            |   500
   234         |Conditioner        | 225 |  1            |   225
    78         |Toothbrush         |  12 |  2            |    24

Amount Payable:749.000000
Thank you for Shopping
Visit Again
Press any key to return to main menu
```

EXITING



```
****************************
Please select an option:
****************************


01. ADMIN VIEW
02. CUSTOMER VIEW
03. EXIT APP

Please Enter an Option: 3
Exiting...
```

CONCLUSION

This was our submission for the Data and File Structures using C-Mini Project. Developing this project was not as easy as we assumed it would be, but we learnt a lot of things while building this system.

We faced a lot of issues in setting up Graphics using Dev C++ as it doesn’t support the graphics header file that is supported in Turbo C++.

We got stuck while reading the Records file because earlier it was a .txt file and it displayed the details incorrectly. Even when we converted the file into the .csv format, some problems were faced but they were resolved with a lot of brainstorming and help from our friends and family.

All in all, we’d like to thank our teachers for this learning opportunity. It was fun building this software!

REFERENCES

- https://www.geeksforgeeks.org/
- https://stackoverflow.com/
- https://www.youtube.com/
- https://www.ibm.com/docs/en/i/7.1?topic=programming
- https://www.quora.com/
- https://sourceforge.net/
- https://www.educative.io/