// Computerised Billing System

//----------------------------------------------------------------------------

// Header Files Included

//----------------------------------------------------------------------------

#include<stdio.h>

#include<iostream.h>

#include<stdlib.h>

#include<dos.h>

#include<string.h>

#include<fstream.h>

#include<ctype.h>

#include<graphics.h>

#include<conio.h>

//Function prototype declaration

void listallm();

void listallp();

void products\_enter();

void members\_enter();

void readp();

void readm();

void billing();

void sales();

void modifyp();

void modifym();

void instructions();

//----------------------------------------------------------------------------

// Function for GRAPHICS slide (cover page)

//----------------------------------------------------------------------------

int gmode,gdriver,r;

//----------------------------------------------------------------------------

// Structure for the products' records

//----------------------------------------------------------------------------

struct products\_record

{

float rate;

int stock;

int no;

char name[20];

}; //end of structure

//Global variable

products\_record ob1,ob2; //declaring objects of the structure

fstream pro1,pro2; //declaring a file stream

//----------------------------------------------------------------------------

// Structure for members' records

//----------------------------------------------------------------------------

struct members\_record

{

int no;

char name[20];

char add[50];

long int tel;

long int card;

float amt ;

}; //end of structure

members\_record m1,m2; //creating objects

fstream mem1,mem2; //declaring streams;

//----------------------------------------------------------------------------

//structure for store opertaions

//----------------------------------------------------------------------------

struct tables

{

int no;

int qty;

char name[20];

float rate;

float total;

};

tables n1,n2;

int i=0;

float discount=0.0;

//Welcome screen of computerized billing system

void welcome()

{

gdriver=DETECT; //request auto detection

initgraph(&gdriver,&gmode,"c:\\turboc3\\bgi"); //initialising graph

setbkcolor(11);

setcolor(4);

settextstyle(SANS\_SERIF\_FONT,HORIZ\_DIR,5);

settextstyle(7,HORIZ\_DIR,7);

moveto(60,100);

outtext("COMPUTERISED");

moveto(160,210);

outtext("BILLING ");

moveto(160,320);

outtext("SYSTEM");

delay(700);

setcolor(12);

int i=0;

while(i<=2\*getmaxx()-460)

{

line(0,i,i,0);

i++;

delay(5);

}

setcolor(YELLOW);

for (r=150;r<230;r+=15)

{ circle (315,239,r);

delay(25);

}

settextstyle(8,HORIZ\_DIR,1);

setcolor(GREEN);

delay(5);

moveto(245,160);

outtext("Submitted By :");

moveto(185,120);

outtext(" ARYAN RAJ , SHREYANSH GUPTA , SAMYAK BARNWAL");

moveto(240,250);

setcolor(GREEN);

outtext("Guided By : ");

moveto(240,270);

outtext("PRIYANKA GABA");

moveto(10,450);

cin.get();

closegraph(); //closing graphics screen

}

//----------------------------------------------------------------------------

// Function for screen setting

//----------------------------------------------------------------------------

void border()

{ textbackground(BLACK);

clrscr();

textcolor(RED);

for (int i=2;i<80;i++) //drawing horizontal lines

{

gotoxy(i,3);

cprintf("-");

gotoxy(i,23);

cprintf("-");

}

for (i=4;i<23;i++) //drawing vertical lines

{

gotoxy(2,i);

cprintf("|");

gotoxy(79,i);

cprintf("|");

}

}

//----------------------------------------------------------------------------

// Main program

//----------------------------------------------------------------------------

void main()

{

clrscr();

void highvideo(); //setting the intensity of text charaters to high

welcome(); // calling the function for making the cover page

remove("sale.dat");

int n;

char l='y';

do

{

textcolor(BLUE);

border(); //making the border of the screen

textcolor(GREEN);

gotoxy(30,5);

cprintf("MAIN MENU");

for (int z=29;z<=40;z++)

{

gotoxy(z,6);

cprintf("-");

}

gotoxy(20,7);

cprintf("1:ADD NEW PRODUCTS");

gotoxy(20,8);

cprintf("2:ADD NEW MEMBERS");

gotoxy(20,9);

cprintf("3:VIEW AN EXISTING PRODUCT RECORD ");

gotoxy(20,10);

cprintf("4:VIEW AN EXISTING MEMBER'S RECORD ");

gotoxy(20,11);

cprintf("5:BILLING ");

gotoxy(20,12);

cprintf("6:TODAY'S SALES ");

gotoxy(20,13);

cprintf("7:MODIFY PRODUCT RECORD ");

gotoxy(20,14);

cprintf("8:MODIFY MEMBER'S RECORD ");

gotoxy(20,15);

cprintf("9:INSTRUCTIONS ");

gotoxy(20,16);

cprintf("10:LIST ALL PRODUCTS ");

gotoxy(20,17);

cprintf("11:LIST ALL MEMBERS ");

gotoxy(20,18);

cprintf("0:EXIT");

gotoxy(20,20);

cprintf("Enter your choice:");

cin>>n;

switch(n)

{

case 1: products\_enter();

break;

case 2: members\_enter();

break;

case 3: readp();

break;

case 4: readm();

break;

case 5: billing();

break;

case 6: sales();

break;

case 7: modifyp();

break;

case 8: modifym();

break;

case 9: instructions();

break;

case 10: listallp();

break;

case 11: listallm();

break;

case 0: l='n';

}

} while (l=='y');

clrscr();

}

//----------------------------------------------------------------------------

// Function for getting the products' records

//----------------------------------------------------------------------------

void products\_enter()

{

int q=0;

char l='y';

pro1.open("products.dat",ios::in); //opening file in stream

pro1.seekg(0,ios::end); //determining end of file

q=pro1.tellg()/sizeof(products\_record); //finding size of file

q+=1255;

pro1.close();

pro1.open("products.dat" ,ios::app);

do

{

textcolor(LIGHTGREEN);

clrscr();

border();

textcolor(LIGHTGREEN);

gotoxy(28,2);

cprintf("ENTERING PRODUCT RECORDS ");

gotoxy(15,6);

cprintf("Name :");

gets(ob1.name);

if (!ob1.name[0]) //to undo entering if name is not entered

return;

gotoxy(15,7);

cprintf("Stock :");

cin>>ob1.stock;

gotoxy(15,8);

cprintf("Rate (Rs.):");

cin>>ob1.rate;

ob1.no = q++;

gotoxy(15,9);

cprintf("Number :");

cout<<ob1.no;

pro1.write((char\*)&ob1,sizeof(products\_record));

gotoxy(10,15);

cprintf("Do you want to enter more (Y/N) ");

l=getch();

} while (tolower(l)=='y');

pro1.close();

}

//--------------------------------------------------------------

//Function for searching in the file

//--------------------------------------------------------------

products\_record products\_search(int no)

{

fstream pro1; //declaring stream

pro1.open("products.dat" ,ios::in||ios::nocreate); //opening the file

while(pro1)

{

pro1.read((char\*)&ob1,sizeof(products\_record));

//reading from file

if (ob1.no==no)

return ob1; //returning the searched record

}

pro1.close();

ob1.no=0;

return ob1;

}

//----------------------------------------------------------------------------

//Function for modifying records

//----------------------------------------------------------------------------

void products\_modify(int no)

{

ob2.no=0;

pro1.open("products.dat",ios::in); //opening the file

pro2.open("product1.dat",ios::app); //opening another file

pro1.seekg(0,ios::beg); //locating beggining of file

pro1.read((char\*)&ob1,sizeof(products\_record));

//reading from file

while (!pro1.eof() ) //testing for end of file

{

if (ob1.no!=no )

pro2.write((char\*)&ob1,sizeof(products\_record));

//writing in file

else

ob2=ob1;

pro1.read((char\*)&ob1,sizeof(products\_record));

//reading from file

}

//displaying previous reccords and entering new records

if (ob2.no)

{

gotoxy(13,7);

cprintf("CURRENT RECORDS ARE ");

gotoxy(15,8);

cprintf("NAME :");

puts(ob2.name);

gotoxy(15,9);

cprintf("RATE :(Rs.)");

cout<<ob2.rate;

gotoxy(15,10);

cprintf("STOCK :");

cprintf("%d",ob2.stock);

gotoxy(13,12);

cprintf("ENTER NEW PRODUCT RECORDS");

gotoxy(15,13);

cprintf("NAME :");

ob1.no=ob2.no;

gets(ob1.name);

if (!isalnum(ob1.name[0])) //testing for an entry

strcpy(ob1.name,ob2.name); //retaining previous name when no entry

gotoxy(15,14);

cprintf("RATE :(Rs.)");

cin>>ob1.rate;

if (!ob1.rate)

ob1.rate=ob2.rate;

gotoxy(15,15);

cprintf("STOCK :");

cin>>ob1.stock;

if (!ob1.stock)

ob1.stock=ob2.stock;

pro2.write((char\*)&ob1,sizeof(products\_record)); //writing in file

}

else

{

gotoxy(20,9);

cprintf("NO ENTRY\a");

}

pro1.close(); //closing file

pro2.close(); //closing file

remove ("products.dat"); //deleting file

rename ("product1.dat","products.dat"); //renaming file

}

//----------------------------------------------------------------------------

// Function to enter members' records

//----------------------------------------------------------------------------

void members\_enter()

{

int m;

char l='y';

//determining number of enteries in the file

mem1.open("members.dat",ios::in);

mem1.seekg(0,ios::end);

m=mem1.tellg()/sizeof(members\_record);

mem1.close();

m+=10;

clrscr();

mem1.open("members.dat",ios::app); //opening file

do

{

textcolor(LIGHTCYAN);

clrscr();

border();

textcolor(LIGHTCYAN);

gotoxy(28,2);

cprintf("ENTERING MEMBER RECORDS ");

gotoxy(15,6);

cprintf("Name :");

gets(m1.name);

if (!m1.name[0])

return; //undo entering when no name is entered

gotoxy(15,7);

cprintf("Card number :");

cin>>m1.card;

gotoxy(15,8);

cprintf("Address :");

gets(m1.add);

gotoxy(15,9);

cprintf("Tel. :");

cin>>m1.tel;

gotoxy(15,10);

cprintf("Amount Deposited (Rs.):");

cin>>m1.amt;

if (m1.amt<=100)

{

gotoxy(15,11);

cprintf("Amount less\a");

gotoxy(15,12);

cprintf("Try Again");

goto end;

}

m1.no = m++;

gotoxy(15,11);

cprintf("Membership number :");

cout<<m1.no;

mem1.write((char\*)&m1,sizeof(members\_record)); //writing in file

end:

gotoxy(10,15);

cprintf("Do you want to enter more (Y/N) ");

l=getch();

} while (tolower(l)=='y');

mem1.close();

}

//----------------------------------------------------------------------------

// Function for searching in the file

//----------------------------------------------------------------------------

members\_record members\_search(int no)

{

fstream mem2; //declaring stream

mem2.open("members.dat",ios::in); //opening file

while (mem2)

{

mem2.read((char\*)&m1,sizeof(members\_record)); //reading from file

if (m1.no==no)

return m1; //returning the searched record

}

mem2.close();

m1.no=0;

return m1;

}

//----------------------------------------------------------------------------

// Function to modify members' records

//----------------------------------------------------------------------------

void members\_modify(int no)

{

m2.no=0;

mem1.open("members.dat",ios::in); //opening a file

mem2.open("member1.dat",ios::app); //opening another file

mem1.seekg(0,ios::beg);

mem1.read((char\*)&m1,sizeof(members\_record));

while (!mem1.eof()) //testing for end of file

{

if (m1.no!=no)

mem2.write((char\*)&m1,sizeof(members\_record));

else

m2=m1;

mem1.read((char\*)&m1,sizeof(members\_record));

}

//displaying current records and entering new records

if (m2.no)

{

gotoxy(13,7);

cprintf("CURRENT RECORDS ARE ");

gotoxy(15,8);

cprintf("NAME :");

puts(m2.name);

gotoxy(15,9);

cprintf("CARD NUMBER :");

cout<<m2.card;

gotoxy(15,10);

cprintf("ADDRESS :");

puts(m2.add);

gotoxy(15,11);

cprintf("TELEPHONE :");

cout<<m2.tel;

gotoxy(15,12);

cprintf("AMOUNT :(Rs.)");

cout<<m2.amt;

gotoxy(13,14);

cprintf("ENTER NEW RECORDS");

gotoxy(15,15);

cprintf("NAME :");

m1.no=m2.no;

gets(m1.name);

if (!m1.name[0])

strcpy(m1.name,m2.name);

gotoxy(15,16);

cprintf("ADDRESS :");

gets(m1.add);

if (!m1.add[0])

strcpy(m1.add,m2.add);

gotoxy(15,17);

cprintf("CARD NUMBER :");

cin>>m1.card;

if (!m1.card)

m1.card=m2.card;

gotoxy(15,18);

cprintf("TELEPHONE :");

cin>>m1.tel;

if (!m1.tel)

m1.tel=m2.tel;

gotoxy(15,19);

cprintf("AMOUNT ADDED :(Rs.)");

cin>>m1.amt;

m1.amt+=m2.amt;

mem2.write((char\*)&m1,sizeof(members\_record));

}

else

{

gotoxy(20,10);

cprintf("NO ENTRY\a");

}

mem1.close(); //closing file

mem2.close(); //closing file

remove ("members.dat"); //removing file

rename ("member1.dat","members.dat"); //renaming file

}

//----------------------------------------------------------------------------

// Function for reading product records

//----------------------------------------------------------------------------

void readp()

{

char l='y';

do

{

textcolor(LIGHTBLUE);

clrscr();

border();

textcolor(LIGHTBLUE);

gotoxy(28,2);

cprintf("VIEWING PRODUCT RECORDS ");

gotoxy(15,6);

cprintf("Enter product number:");

int no;

cin>>no;

ob1=products\_search(no); //calling for search in the file

// displaying records

if (ob1.no)

{

gotoxy(18,9);

cprintf("The record is");

gotoxy(20,10);

cprintf(" Number :");

cout<<ob1.no;

gotoxy(20,11);

cprintf(" Stock :");

cout<<ob1.stock;

gotoxy(20,12);

cprintf(" Name : ");

puts(ob1.name);

gotoxy(20,13);

cprintf(" Rate :(Rs.)");

cout<<ob1.rate;

}

else

{

gotoxy(20,10);

cprintf("NO ENTRY \a");

}

gotoxy(15,16);

cprintf("Any more records desired (Y/N)");

l=getch();

} while(tolower(l)=='y');

}

//----------------------------------------------------------------------------

// Function to pruduce screen for 'modify product records'

//----------------------------------------------------------------------------

void modifyp()

{

char l='y';

do

{

textcolor(LIGHTBLUE);

clrscr();

border();

textcolor(LIGHTBLUE);

gotoxy(25,2);

cprintf("MODIFYING A PRODUCT RECORD");

gotoxy(15,5);

cprintf("Enter the product number:");

int no;

cin>>no;

products\_modify(no); //calling for modifications

gotoxy(13,20);

cprintf("Any more modifications desired(Y/N)");

l=getch();

} while (tolower(l)=='y');

}

//----------------------------------------------------------------------------

// Function to read members' records

//----------------------------------------------------------------------------

void readm()

{

char l='y';

do

{

textcolor(LIGHTCYAN);

clrscr();

border();

textcolor(LIGHTCYAN);

gotoxy(25,2);

cprintf("VIEWING A MEMBER'S RECORD ");

gotoxy(15,6);

cprintf("Enter membership number:");

int no;

cin>>no;

m1=members\_search(no); //calling for search

//displaying records

if (m1.no)

{

gotoxy(18,9);

cprintf("The record is");

gotoxy(20,10);

cprintf("Number :");

cout<<m1.no;

gotoxy(20,11);

cprintf("Name :");

puts(m1.name);

gotoxy(20,12);

cprintf("Card number :");

cout<<m1.card;

gotoxy(20,13);

cprintf("Address :");

puts(m1.add);

gotoxy(20,14);

cprintf("Telephone :");

cout<<m1.tel;

gotoxy(20,15);

cout<<"Amount :(Rs.)"<<m1.amt;

}

else

{

gotoxy(17,12);

cprintf("NO ENTRY\a ");

}

gotoxy(15,18);

cprintf("Any more records desired (Y/N)");

l=getch();

} while(tolower(l)=='y');

}

//----------------------------------------------------------------------------

// Function to display screen for 'modify members' records

//----------------------------------------------------------------------------

void modifym()

{

char l='y';

do

{

textcolor(LIGHTCYAN);

clrscr();

border();

gotoxy(25,2);

cprintf("MODIFYING MEMBER'S RECORDS ");

gotoxy(15,5);

cprintf("Enter the membership number:");

int no;

cin>>no;

members\_modify(no); //calling for modifications

gotoxy(13,22);

cprintf("Any more modifications desired(Y/N)");

l=getch();

} while(tolower(l)=='y');

}

//----------------------------------------------------------------------------

// Function for creating 'sale.dat'

//----------------------------------------------------------------------------

// Function for billing

//----------------------------------------------------------------------------

void billing()

{

textcolor(LIGHTRED);

clrscr();

border();

textcolor(LIGHTRED);

fstream b1; //declaring stream

b1.open("sale.dat",ios::app); //opening file

gotoxy(30,2);

cprintf("BILLING ");

gotoxy(15,7);

cprintf("Are you a member(Y/N):");

float less=1.0;

char m;

int no;

long card;

m=getche();

if (tolower(m)=='y')

{

gotoxy(18,9);

cprintf("Enter membership number:");

cin>>no;

gotoxy(18,10);

cprintf("Enter card number:");

cin>>card;

m1=members\_search(no);

if (card!=m1.card) //checking for authenticity of the details

{

gotoxy(20,12);

cprintf("Incorrect \a");

getch();

return; //undoing billing process

}

else

if (m1.amt>100)

{

less=0.95;

m1.amt-=5;

}

else

{

gotoxy(20,12);

cprintf("Amount Less\a");

gotoxy(15,15);

cprintf("Continue with normal billing(Y/N)");

m=getch();

if (tolower(m)=='n')

return;

}

//decreasing the members' amount if the details are correct

fstream t3,t4;

t3.open("members.dat",ios::in);

t4.open("member1.dat",ios::app);

t3.seekg(0,ios::beg);

t3.read((char\*)&m2,sizeof(members\_record));

while (!t3.eof())

{

if (m2.no!=m1.no)

t4.write((char\*)&m2,sizeof(members\_record));

else

t4.write((char\*)&m1,sizeof(members\_record));

t3.read((char\*)&m2,sizeof(members\_record));

}

t3.close();

t4.close();

remove("members.dat");

rename("member1.dat","members.dat");

}

clrscr();

float total=0.0;

textcolor(LIGHTGREEN);

gotoxy(60,1);

cprintf("PRESS 0 :Exit Billing");

textcolor(LIGHTRED);

gotoxy(30,1);

cprintf("BILLING");

i=0;

void table1(); //function declaration for making table

beg: //giving line a name for further reference

table1(); //calling function for making table

gotoxy(4,6+i);

cin>>n1.no;

ob1=products\_search(n1.no); //searching for product record

gotoxy(56,6+i);

float output1(products\_record); //declaring a nested function

if (n1.no>0)

{

if(ob1.no>0)

{

cin>>n1.qty;

strcpy(n1.name,ob1.name);

n1.rate=ob1.rate;

n1.total=output1(ob1);

total+=n1.total;

fstream t3,t4;

t3.open("sale.dat",ios::in);

t4.open("sale1.dat",ios::app);

t3.seekg(0,ios::beg);

int qty=n1.qty;

n1.qty=n1.total/n1.rate;

char test='y';

t3.read((char\*)&n2,sizeof(tables));

while (!t3.eof())

{

if (n1.no==n2.no)

{

n2.qty+=n1.qty;

n2.total+=n1.total;

test='n';

}

t4.write((char\*)&n2,sizeof(tables));

t3.read((char\*)&n2,sizeof(tables));

}

if (test=='y')

{

t4.seekg(0,ios::end);

t4.write((char\*)&n1,sizeof(tables));

}

t3.close();

t4.close();

remove("sale.dat");

rename("sale1.dat","sale.dat");

n1.qty=qty;

if (n1.total)

{

//reducing the products' stock

fstream temp3,temp4;

temp3.open("products.dat",ios::in);

temp4.open("product1.dat",ios::app);

temp3.seekg(0,ios::beg);

temp3.read((char\*)&ob1,sizeof(products\_record));

while (!temp3.eof())

{

if (ob1.no==n1.no)

ob1.stock-=n1.qty;

temp4.write((char\*)&ob1,sizeof(products\_record));

temp3.read((char\*)&ob1,sizeof(products\_record));

}

temp3.close();

temp4.close();

remove("products.dat");

rename("product1.dat","products.dat");

}

i++;

goto beg;

}

else

if (ob1.no==0)

{

gotoxy(10,6+i);

cprintf("No entry\a");

i++;

goto beg;

}

}

else

if (n1.no==0)

{

textcolor(WHITE);

for (int j=0;j<81;j++) //ending table

{

gotoxy(j,6+i);

cprintf("-");

}

}

textcolor(LIGHTRED);

gotoxy(5,9+i);

cprintf("Number of items = ");

cout<<i;

gotoxy(5,10+i);

cprintf("Grand total = Rs.");

cout<<total;

//giving discount

if (less!=1)

{

discount+=0.05\*total;

gotoxy(5,12+i);

cprintf("Discount = Rs.");

printf("%.2f",0.05\*total);

gotoxy(5,13+i);

cprintf("Net total = Rs.");

printf("%.2f",less\*total);

}

b1.close();

getch();

}

//----------------------------------------------------------------------------

// nested function of 'billing'

//----------------------------------------------------------------------------

float output1(products\_record ob1)

{

if (ob1.no==0) //if no entry then return to billing

return 0;

float stotal;

stotal=ob1.rate\*n1.qty; //determining the cost of the particular item

//putting the values on the screen

gotoxy(4,6+i);

cout<<ob1.no;

gotoxy(10,6+i);

puts(ob1.name);

gotoxy(38,6+i);

cout<<ob1.rate;

gotoxy(43,6+i);

cout<<" ";

gotoxy(56,6+i);

if (n1.qty>ob1.stock) //checking for the item being in stock

{

gotoxy(50,6+i);

cprintf("Out of stock\a");

return 0;

}

cout<<n1.qty;

gotoxy(68,6+i);

cout<<stotal;

gotoxy(74,6+i);

cout<<" ";

return stotal;

}

//----------------------------------------------------------------------------

// Function for making table

//----------------------------------------------------------------------------

void table1()

{

textcolor(WHITE);

//drawing vertical lines

for (int a=1;a<=80;a++)

{

gotoxy(a,2);

cprintf("\_");

gotoxy(a,5);

cprintf("-");

}

//drawing horizontal lines(always)

for (a=3;a<=5;a++)

{

gotoxy(1,a);

cprintf("|");

gotoxy(9,a);

cprintf("|");

gotoxy(31,a);

cprintf("|");

gotoxy(49,a);

cprintf("|");

gotoxy(64,a);

cprintf("|");

gotoxy(80,a);

cprintf("|");

}

/\* drawing horizontal lines

(depending upon the s.no. of item in purchase list)\*/

gotoxy(1,6+i);

cprintf("|");

gotoxy(9,6+i);

cprintf("|");

gotoxy(31,6+i);

cprintf("|");

gotoxy(49,6+i);

cprintf("|");

gotoxy(64,6+i);

cprintf("|");

gotoxy(80,6+i);

cprintf("|");

textcolor(LIGHTRED);

//giving column titles

gotoxy(2,3);

cprintf("Product");

gotoxy(14,3);

cprintf("Product name ");

gotoxy(35,3);

cprintf("Rate (Rs.)");

gotoxy(53,3);

cprintf("Quantity");

gotoxy(67,3);

cprintf("Total (Rs.)");

gotoxy(2,4);

cprintf("number");

gotoxy(53,4);

cprintf("(Kgs/pcs)");

}

//----------------------------------------------------------------------------

// Function for viewing the day's sale

//----------------------------------------------------------------------------

void sales()

{

textcolor(LIGHTRED);

clrscr();

textcolor(LIGHTRED);

gotoxy(28,1);

cprintf("TODAY'S SALES");

float sales=0.0;

fstream sal1;

sal1.open("sale.dat",ios::in);

i=0;

sal1.read((char\*)&n1,sizeof(n1));

while(!sal1.eof() && n2.no!=n1.no)

{

table1();

gotoxy(4,6+i);

cout<<n1.no;

gotoxy(10,6+i);

puts(n1.name);

gotoxy(38,6+i);

cout<<n1.rate;

gotoxy(43,6+i);

cout<<" ";

gotoxy(56,6+i);

cout<<n1.qty;

gotoxy(68,6+i);

cout<<n1.total;

sales+=n1.total;

gotoxy(74,6+i);

cout<<" ";

i++;

n2.no=n1.no;

sal1.read((char\*)&n1,sizeof(n1));

}

textcolor(WHITE);

gotoxy(1,6+i);

for(int n=1;n<=80;n++)

cprintf("-");

textcolor(LIGHTRED);

gotoxy(5,8+i);

cprintf("Grand total = Rs.");

cout<<sales;

gotoxy(5,10+i);

cprintf("Discount = Rs.");

cout<<discount;

gotoxy(5,11+i);

cprintf("Net total = Rs.");

cout<<(sales-discount);

getch();

}

//----------------------------------------------------------------------------

// Instructions

//----------------------------------------------------------------------------

// Function for help on entering records

void enter()

{

border();

gotoxy(5,1);

cprintf("Entering Records");

gotoxy(5,7);

textcolor(11);

cprintf("For Products");

textcolor(13);

gotoxy(5,10);

cprintf("This part of the program helps the store management to ");

cprintf("add more");

gotoxy(5,9);

cprintf("products to their store. The existing product records are");

cprintf(" not ");

gotoxy(5,10);

cprintf("affected by this option. The new record(s) are appended at");

cprintf(" the end");

gotoxy(5,11);

cprintf("of the file containing such records namely ");

textcolor(12);

cprintf("'products.dat'");

textcolor(13);

cprintf(". This");

gotoxy(5,12);

cprintf("option automatically assigns a product number to the product");

cprintf(" which");

gotoxy(5,13);

cprintf("must be remembered for any product reference.");

gotoxy(5,15);

textcolor(11);

cprintf("For members");

textcolor(13);

gotoxy(5,17);

cprintf("This option works just like its counterpart for products. It");

cprintf(" stores");

gotoxy(5,18);

cprintf("records in the file ");

textcolor(12);

cprintf("'members.dat'");

textcolor(13);

cprintf(" and assigns a membership number ");

gotoxy(5,19);

cprintf("for every new entry. For further details see the help section");

cprintf(" on ");

textcolor(12);

gotoxy(5,20);

cprintf("'Membership Rules'.");

textcolor(13);

getch();

}

//Function for help on viewing records

void view()

{

border();

gotoxy(30,5);

cprintf("Viewing Records");

gotoxy(10,7);

cprintf("This option enables the user to view the latest records of ");

gotoxy(10,8);

cprintf("either products or members. It helps in operations like ");

gotoxy(10,9);

cprintf("stock checking and member verification. The viewing is");

gotoxy(10,10);

cprintf("only possible with the product/membership numbers.");

getch();

}

//Function for help on modifying records

void modify()

{

border();

gotoxy(30,5);

cprintf("Modifying Records");

gotoxy(10,7);

cprintf("This option helps in altering the records in the files.");

gotoxy(10,8);

cprintf("It can be used for deleting the entries as well. In ");

gotoxy(10,9);

cprintf("case of products either the record could be changed to");

gotoxy(10,10);

cprintf("some new entry or its stock could be made zero. For members");

gotoxy(10,11);

cprintf("deletion is only possible by overwriting with a new entry.");

getch();

}

// Function for help on billing process

void procedure()

{

border();

gotoxy(30,5);

cprintf("Billing Procedure");

gotoxy(10,7);

cprintf("The actual billing procedure consists of two sections.");

gotoxy(10,8);

cprintf("One checks if the buyer is a member or not and accordingly");

gotoxy(10,9);

cprintf("performs the necessary tasks. The other section consists of ");

gotoxy(10,10);

cprintf("billing. The user is asked to enter the product number and the ");

gotoxy(10,11);

cprintf("quantity bought. The computer calculates the total itself and ");

gotoxy(10,12);

cprintf("also the discount if necessary. In order to exit the procedure");

gotoxy(10,13);

cprintf("and view the final total, the user should enter the product number");

gotoxy(10,14);

cprintf("as '0'.");

getch();

}

// Function for help on Membership details

void rules()

{

border();

gotoxy(30,5);

cprintf("Membership Rules");

gotoxy(5,7);

cprintf("The program lets its subscribers (stores) maintain a");

cprintf(" member's list");

gotoxy(5,8);

cprintf("in order to help them in increasing their sales for a");

cprintf(" marginal cut");

gotoxy(5,9);

cprintf("in their profits. The program sets minimum membership ");

cprintf("amount as Rs.100");

gotoxy(5,10);

cprintf("which must be paid at the time of registration. The member");

cprintf(" will be ");

gotoxy(5,11);

cprintf("given a membership number and a membership card (to be");

cprintf("collected ");

gotoxy(5,12);

cprintf("after a week by reference to the membership number and ");

cprintf("other");

gotoxy(5,13);

cprintf("particulars). In case a member loses the membership card ");

cprintf(", he/she ");

gotoxy(5,14);

cprintf("must register a complaint with the store by submitting a");

cprintf(" written");

gotoxy(5,15);

cprintf("application with all the particulars. He/She can collect ");

cprintf("his/her ");

gotoxy(5,16);

cprintf("remaining money by showing proofs of the particulars.");

gotoxy(5,17);

cprintf("The membership allows a person to avail a special discount");

cprintf(" of 5%");

gotoxy(5,18);

cprintf("over every purchase. Rs.5 will be subtracted from the ");

cprintf("member's account");

gotoxy(5,19);

cprintf("over every purchase. The account will be rendered ");

cprintf("inaccessible if the ");

gotoxy(5,20);

cprintf("amount will be less than Rs.100 and the member would be");

cprintf(" expected to ");

gotoxy(5,21);

cprintf("deposit more money for continuing the membership. The store");

cprintf(" will also");

gotoxy(5,22);

cprintf("give away attractive gifts to a lucky member once every year.");

getch();

}

// Functiom for help on today's sales

void today()

{

border();

gotoxy(30,5);

cprintf("Today's sales");

gotoxy(10,7);

cprintf("This option shows the total sales in terms of both quantity");

gotoxy(10,8);

cprintf("and amount per item. This is stored in a file namely ");

gotoxy(10,9);

cprintf("'sale.dat' which is modified after every buying during the");

gotoxy(10,10);

cprintf("billing process.");

getch();

}

//Function for displaying help index

void instructions()

{

clrscr();

int n;

char l='y';

do

{

textcolor(13);

border();

textcolor(13);

gotoxy(30,5);

cprintf("HELP INDEX");

for (int z=29;z<=39;z++)

{

gotoxy(z,6);

cprintf("-");

}

gotoxy(20,7);

cprintf("1:ADD RECORDS");

gotoxy(20,8);

cprintf("2:VIEWING RECORDS ");

gotoxy(20,9);

cprintf("3:MODIFYING RECORDS");

gotoxy(20,10);

cprintf("4:BILLING PROCEDURE");

gotoxy(20,11);

cprintf("5:TODAY'S SALES ");

gotoxy(20,12);

cprintf("6:MEMBERSHIP DETAILS");

gotoxy(20,13);

cprintf("0:BACK TO MAIN MENU");

gotoxy(21,16);

cprintf("Enter your choice:");

cin>>n;

switch(n)

{

case 1: enter();

break;

case 2: view();

break;

case 3: modify();

break;

case 4: procedure();

break;

case 5: today();

break;

case 6: rules();

break;

case 0: l='n';

}

} while (l=='y');

}

void listallp()

{

clrscr();

border();

fstream pro1; //declaring stream

pro1.open("products.dat" ,ios::in||ios::nocreate); //opening the file

gotoxy(29,2);

cprintf("The records are");

gotoxy(9,5);

cprintf(" Number ");

gotoxy(26,5);

cprintf(" Stock ");

gotoxy(43,5);

cprintf(" Name ");

gotoxy(59,5);

cprintf(" Rate :(Rs.)");

gotoxy(3,6);

for(i=1;i<=75;i++)

cprintf("-");

i=8;

while(pro1.read((char\*)&ob1,sizeof(products\_record)))

{

gotoxy(11,i);

cout<<ob1.no;

gotoxy(28,i);

cout<<ob1.stock;

gotoxy(44,i);

puts(ob1.name);

gotoxy(62,i);

cout<<ob1.rate;

i++;

}

pro1.close();

getch();

}

void listallm()

{

clrscr();

border();

fstream mem1; //declaring stream

mem1.open("members.dat" ,ios::in||ios::nocreate); //opening the file

gotoxy(29,2);

cprintf("The records are");

gotoxy(3,5);

cprintf(" Number ");

gotoxy(11,5);

cprintf(" Name ");

gotoxy(21,5);

cprintf(" Card No. ");

gotoxy(35,5);

cprintf(" Address ");

gotoxy(49,5);

cprintf(" Telephone ");

gotoxy(63,5);

cprintf(" Amount (Rs.) ");

gotoxy(3,6);

for(i=1;i<=75;i++)

cprintf("-");

i=8;

while(mem1.read((char\*)&m1,sizeof(members\_record)))

{

gotoxy(6,i);

cout<<m1.no;

gotoxy(12,i);

cout<<m1.name;

gotoxy(24,i);

cout<<m1.card;

gotoxy(37,i);

puts(m1.add);

gotoxy(51,i);

cout<<m1.tel;

gotoxy(66,i);

cout<<m1.amt;

i++;

}

mem1.close();

getch();

}