**……………………………………………………………**

**……………………………………… Project Report**

**On**

**CANTEEN MANAGEMENT SYSTEM**

**……………………… CLASS—XII**

**ACKNOWLEDGEMENT**

**I would like to convey my heartful thanks to (Computer**

**Science) who always gave valuable suggestions & guidance for completion of my project.**

**He helped me to understand & remember important details of the project. My project has been a success only because of his guidance.**

**I am especially indented & I am also beholden to my friends. And finally I**

**thank to the members of my family for their support & encouragement.**

**CERTIFICATE**

**This is to certify that ……………………….**

**of ………. of**

**………………………………………………………**

**….. has completed his project under my supervision. He has taken proper care & shown sincerity in completion of this project.**

**I certify that this project is up to my expectation & as per the guideline issued by CBSE.**

***………………………………..***

***(Computer Science faculty )***

***Content***

1. ***Introduction***
2. ***Source Code***
3. ***Output***
4. ***Requirement***
5. ***Advantages and disadvantages***
6. ***Conclusion***
7. ***System design***



This project **CANTEEN MANAGEMENT SYSTEM** includes some facilities of customer's and products search, display, modification, delete etc. This software searches the client data which is store in the record and then gives permission to place the order by the customer.

The software used for small scale canteens which have limited customers and products and wants to maintain the records.



//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// HEADER FILE USED IN PROJECT

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include<conio.h> #include<stdio.h> #include<process.h> #include<fstream.h> #include<iomanip.h> #include<string.h> #include<ctype.h> struct order

{

int prodid1;

char pname1[50]; char compy1[50]; int qty1;

float price1,dis1;

}o1[50];

int orderk=0;

void middleadminmenu();

void copyme(int k,order order1[],int q1,int &c2); void intromain();

int getproduct(); int getcustomers(); void display\_all();

void display\_all\_cust(); void prod\_tabular();

void modify\_record(int n); void delete\_record(int n); void againopenandclose();

void againopenandclosecust(); int search(int p);

void changeqty(int pr1,int q11);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// CLASS USED IN PROJECT

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*customer information\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* class customer

{

int cust\_id;

char cname[25]; char address[35]; char phno[15]; public:

void modifycust\_data(int n1,char nm[15],char add[15],char q[15]);

int getcustid()

{return cust\_id;}

char \*getcustnm()

{ return cname;}

char \*getcustadd()

{ return address;}

char \*getphno()

{ return phno;}

void cust\_input(int custid)

{

gotoxy(5,7); cout<<"======================================================================";

gotoxy(10,8); cout<<"CUST NO:";

cust\_id=custid; gotoxy(24,8);

cout<<cust\_id; gotoxy(35,8); cout<<"NAME OF CUST:";

gotoxy(54,8); cin>>cname; gotoxy(10,9); cout<<"ADDRESS:";

gotoxy(22,9); cin>>address; gotoxy(10,11); cout<<"PHONE NO.:";

gotoxy(25,11); cin>>phno; gotoxy(5,12);

cout<<"=====================================================================";

}

void show\_cust()

{

gotoxy(5,7); cout<<"======================================================================";

gotoxy(10,8); cout<<"CUST NO:";

gotoxy(25,8); cout<<cust\_id; gotoxy(35,8); cout<<"NAME OF CUST:";

gotoxy(54,8); cout<<cname; gotoxy(10,9); cout<<"ADDRESS:";

gotoxy(25,9); cout<<address; gotoxy(10,10); cout<<"PHONE NO.:";

gotoxy(25,10); cout<<phno; gotoxy(5,12);

cout<<"=====================================================================";

}

void showallcust(int c)

{

gotoxy(1,c); cout<<cust\_id<<setw(17)<<cname<<setw(15)<<address<<setw(30)<<phno;

}

void showcustdatamulti()

{

gotoxy(5,7); cout<<"======================================================================";

gotoxy(10,8); cout<<"CUST NO:";

gotoxy(30,8); cout<<cust\_id; gotoxy(40,8);

cout<<"NAME OF CUSTD:";

gotoxy(60,8); cout<<cname; gotoxy(10,9); cout<<"ADDRESS:";

gotoxy(30,9); cout<<address; gotoxy(10,10); cout<<"PHONE NO.:";

gotoxy(30,10); cout<<phno; gotoxy(5,11);

cout<<"=====================================================================";

}

};

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*modify customer details

void customer::modifycust\_data(int n1,char nm[15],char add[15],char q[15])

{

char tmpnm[40],tmpnm2[40],tmpnm3[15]; gotoxy(5,14);

cout<<"===================WANT TO MODIFY ===============================";

gotoxy(10,15); cout<<"CUST NO:";

cust\_id=n1; gotoxy(25,15); cout<<cust\_id; gotoxy(40,15); strcpy(cname,nm); cout<<"NAME OF CUST:";

gotoxy(60,15); cout<<cname; gotoxy(10,17);

cout<<"Want to change the name of customer"; gotoxy(50,17);

int flag=0; while(1)

{

gets(tmpnm); if(strlen(tmpnm)!=0)

{

flag=1;

break;

}

if(strlen(tmpnm)==0)

{ flag=0; break;

}

}

if(flag==1)

{ strcpy(cname,tmpnm);

}

else

{

}

gotoxy(1,18); strcpy(address,add);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* NAME TO BE MODIFY cout<<"CUSTOMER ADDRESS:";

gotoxy(20,18); cout<<address; gotoxy(45,18);

cout<<"Want to change the address"; gotoxy(70,18);

flag=0; while(1)

{

gets(tmpnm2); if(strlen(tmpnm2)!=0)

{

flag=1;

break;

}

if(strlen(tmpnm2)==0)

{ flag=0; break;

}

}

if(flag==1)

{ strcpy(address,tmpnm2);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*COMPANY NAME TO BE MODIFIED ENDS HERE gotoxy(5,19);

strcpy(phno,q);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* phone no. TO BE MODIFY cout<<"CUSTOMER PHONE NO.:";

gotoxy(20,18); cout<<phno; gotoxy(45,18);

cout<<"Want to change the phone no."; gotoxy(70,18);

flag=0; while(1)

{

gets(tmpnm3); if(strlen(tmpnm3)!=0)

{

flag=1;

break;

}

if(strlen(tmpnm3)==0)

{ flag=0; break;

}

}

if(flag==1)

{ strcpy(phno,tmpnm3);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*MODIFIED ENDS HERE

gotoxy(5,20); cout<<"=====================================================================";

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*class function customer modify ends here

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*customer class ends here\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*function to add the records in a file void write\_customer()

{

ofstream objoff; customer cobj;

objoff.open("customer.dat",ios::out|ios::app); clrscr();

intromain();

int r=getcustomers(); if(r>100) //1000

{

r=1; // r=100

}

cobj.cust\_input(r); objoff.write((char\*)&cobj,sizeof(customer)); objoff.close();

gotoxy(10,20);

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CUSTOMER RECORD SAVED\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* "; cin.ignore();

getch();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*function to add the record in a file ends here

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*check the customer number already given or not\*\*\*\*\*\*

int getcustomers()

{

ifstream objiff; customer cust; int count=0;

objiff.open("customer.dat",ios::binary); objiff.seekg(0,ios::beg);

if(!objiff)

{

cout<<"File could not be open !! Press any Key..."; getch();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*temporary hiding these lines while(objiff.read((char \*) &cust, sizeof(customer)))

{

count++;

}

//\*\*\*\*\*\*\*\*\*\*\*jump to the last line objiff.seekg(count-sizeof(cust),ios::beg); objiff.read((char \*) &cust, sizeof(customer)); count=cust.getcustid();

count++; objiff.close();

return count;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*check the customer number ends here

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// function to read specific record from file

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* void display\_cust\_sp(int n)

{

ifstream objfp; customer cust; int flag=0;

objfp.open("customer.dat",ios::binary); if(!objfp)

{

cout<<"File could not be open !! Press any Key...";

getch(); return;

}

while(objfp.read((char\*)&cust,sizeof(customer)))

{

if(cust.getcustid()==n)

{

clrscr(); intromain(); cust.show\_cust(); flag=1;

}

}

objfp.close(); if(flag==0)

cout<<"\n\nrecord not exist"; getch();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*customer record sepecific\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// DISPLAY ALL THE CUSTOMER TABULAR FORM

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* void cust\_tabular()

{

int r=0,col=10;

customer cust; ifstream inFile;

inFile.open("customer.dat",ios::binary); if(!inFile)

{

cout<<"File could not be open !! Press any Key...";

getch(); return;

}

display\_all\_cust();

while(inFile.read((char \*) &cust, sizeof(customer)))

{

if(r<=12)

{ r++;

cust.showallcust(col); col++;

}else

{

gotoxy(20,30);

cout<<"--------------press any key ";

getch();

clrscr(); display\_all\_cust(); col=10;

r=0;

}

}

inFile.close(); getch();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*tabular forms ends\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//function to display all the records of product

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* void display\_all\_cust()

{

clrscr(); intromain(); gotoxy(1,5);

cout<<" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CUSTOMER DETAILS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"; gotoxy(1,6);

cout<<"================================================================================";

gotoxy(1,7); cout<<"CUST.NO"<<setw(10)<<"NAME"<<setw(15)<<"ADDRESS"<<setw(30)<<"PHONE NO";

gotoxy(1,8); cout<<"================================================================================";

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DISPLAY ALL ENDS HERE\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// MODIFY customer RECORD

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* void modify\_cust\_record(int n)

{

customer cust,temp;

char tmpnm[50],tmpaddress[50]; ifstream inFile;

int fpos1=-1; inFile.open("customer.dat",ios::binary); if(!inFile)

{

cout<<"File could not be open !! Press any Key...";

getch(); return;

}

int flag=0;

while(inFile.read((char \*) &cust, sizeof(customer)))

{

if(cust.getcustid()==n)

{ clrscr();

intromain(); cust.showcustdatamulti(); flag=1;

}

}

inFile.close();

if(flag==0)

cout<<"\n\nrecord not exist";

else

{

//\*\*\*\*\*\*\*modifying the records starts here fstream File;

File.open("customer.dat",ios::binary|ios::in|ios::out); if(!File)

{

cout<<"File could not be open !! Press any Key..."; getch();

return;

}

while(File.read((char \*) &cust, sizeof(customer)))

{

if(cust.getcustid()==n)

{ fpos1=(int)File.tellg(); break;

}

}

File.seekp(fpos1-sizeof(customer),ios::beg); strcpy(tmpnm,cust.getcustnm()); strcpy(tmpaddress,cust.getcustadd());

char q1[15]; strcpy(q1,cust.getphno()); gotoxy(1,12);

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"; gotoxy(1,13);

cout<<"================ENTER NEW VALUES FOR THE RECORDS GIVEN ABOVE=================";

temp.modifycust\_data(n,tmpnm,tmpaddress,q1); File.write((char \*) &temp, sizeof(customer)); File.close();

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// DELETE THE RECORD OF THE customer AVAILABLE

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* void deletecust\_record(int n)

{

customer cust;

ifstream inFile; inFile.open("customer.dat",ios::binary); if(!inFile)

{

cout<<"File could not be open !! Press any Key..."; getch();

return;

}

int flag=0;

while(inFile.read((char \*) &cust, sizeof(customer)))

{

if(cust.getcustid()==n)

{ clrscr();

intromain(); cust.showcustdatamulti(); flag=1;

}

}

inFile.close(); char ch; if(flag==0)

cout<<"\n\nrecord not exist";

else

{

//\*\*\*\*\*\*\*deletion of the records starts from here gotoxy(1,15);

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"; gotoxy(5,16);

cout<<"======DO YOU WANT TO DELETE THE RECORDS GIVEN ABOVE[YES(Y) OR NO (N)========";

gotoxy(2,17); cin>>ch;

if (toupper(ch)=='Y')

{

ofstream outFile; outFile.open("Temp2.dat",ios::binary); ifstream objiff("customer.dat",ios::binary); objiff.seekg(0,ios::beg);

while(objiff.read((char \*) &cust, sizeof(customer)))

{

if(cust.getcustid()!=n)

{

outFile.write((char \*) &cust, sizeof(customer));

}

}

outFile.close(); objiff.close(); remove("customer.dat");

rename("Temp2.dat","customer.dat"); againopenandclosecust(); gotoxy(30,20);

cout<<"----------------------------Record Deleted ";

}

}

getch();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*delete record ends\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* void againopenandclosecust()

{

ifstream inFile;

customer cust; inFile.open("customer.dat",ios::binary); if(!inFile)

{

getch(); return;

}

while(inFile.read((char \*) &cust, sizeof(customer)))

{

}

inFile.close();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*search the customer\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* int searchcust(int p)

{

customer cust; int tmprt=0; ifstream inFile;

inFile.open("customer.dat",ios::binary); if(!inFile)

{

cout<<"File could not be open !! Press any Key..."; getch();

return -1;

}

int flag=0;

while(inFile.read((char \*) &cust, sizeof(customer)))

{

if(cust.getcustid()==p)

{ clrscr();

intromain(); cust.showcustdatamulti(); flag=1; tmprt=(int)inFile.tellg(); break;

}

}

inFile.close(); if(flag==0)

return 1;

//cout<<"\n\nrecord not exist";

else

{

}

}

return tmprt;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*customer information ends\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* class product

{

int prodid;

char name[50]; char company[50]; int qty;

float price,dis; public:

product()

{

qty=0; price=0; dis=0;

}

void modifydata(int n1,char snm[15],char companynm[15],int q);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* void showdatamulti()

{

gotoxy(5,7); cout<<"======================================================================";

gotoxy(10,8); cout<<"PROD NO:";

gotoxy(30,8); cout<<prodid; gotoxy(40,8); cout<<"NAME OF PROD:";

gotoxy(60,8); cout<<name; gotoxy(10,9);

cout<<"COMPANY:";

gotoxy(30,9); cout<<company; gotoxy(40,9); cout<<"PROD PRICE:";

gotoxy(60,9); cout<<price; gotoxy(10,10); cout<<"QUANTITY:";

gotoxy(30,10); cout<<qty; gotoxy(40,10); cout<<"DISCOUNT";

gotoxy(60,10); cout<<dis; gotoxy(5,11);

cout<<"=====================================================================";

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void create\_prod(int rn1)

{

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* gotoxy(5,7);

cout<<"======================================================================";

gotoxy(10,8); cout<<"PROD NO:";

prodid=rn1; gotoxy(24,8); cout<<prodid; gotoxy(35,8);

cout<<"NAME OF PROD:";

gotoxy(54,8); cin>>name; gotoxy(10,9); cout<<"COMPANY:";

gotoxy(22,9); cin>>company; gotoxy(35,9); cout<<"QUANTITY";

gotoxy(54,9); cin>>qty; gotoxy(10,11);

cout<<"PROD PRICE";

gotoxy(25,11); cin>>price; gotoxy(35,11); cout<<"DISCOUNT";

gotoxy(45,11); cin>>dis; gotoxy(5,12);

cout<<"=====================================================================";

}

void show\_prod()

{

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* gotoxy(5,7);

cout<<"======================================================================";

gotoxy(10,8); cout<<"PROD NO:";

//rollno=rn1; gotoxy(25,8); cout<<prodid; gotoxy(35,8);

cout<<"NAME OF PROD:";

gotoxy(54,8); cout<<name; gotoxy(10,9); cout<<"COMPANY:";

gotoxy(25,9); cout<<company; gotoxy(35,9); cout<<"PROD PRICE:";

gotoxy(54,9); cout<<price; gotoxy(10,10); cout<<"DISCOUNT:";

gotoxy(25,10); cout<<dis<<"%"; gotoxy(35,10); cout<<"QUANTITY:";

gotoxy(54,10); cout<<qty; gotoxy(5,11);

cout<<"=====================================================================";

}

//-----------------------show data tabular form----------------------

void showall(int c)

{

gotoxy(1,c);

cout<<prodid<<setw(17)<<name<<setw(11)<<company<<setw(10)<<"Rs."<<price<<setw(15)<<qty<<setw(12)<<dis<<"

%";

}

//

int retpno()

{return prodid;}

ends here

float retprice()

{return price;}

char\* getcompany()

{ return company;} char\* getname()

{return name;}

int getqty()

{ return qty;} float retdis()

{return dis;}

void setqty(int q21)

{

qty=q21;

}

}; //class ends here

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// global declaration for stream object, object

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*modify product

void product::modifydata(int n1,char snm[15],char companynm[15],int q)

{

char tmpnm[40],tmpnm2[40]; gotoxy(5,14);

cout<<"===================WANT TO MODIFY ===============================";

gotoxy(10,15); cout<<"PROD NO:";

prodid=n1;

gotoxy(25,15); cout<<prodid; gotoxy(40,15); strcpy(name,snm); cout<<"NAME OF PROD:";

gotoxy(60,15); cout<<name; gotoxy(10,17);

cout<<"Want to change the name of product"; gotoxy(50,17);

int flag=0; while(1)

{

gets(tmpnm); if(strlen(tmpnm)!=0)

{

flag=1;

break;

}

if(strlen(tmpnm)==0)

{ flag=0; break;

}

}

if(flag==1)

{ strcpy(name,tmpnm);

}

else

{

}

gotoxy(1,18); strcpy(company,companynm);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*COMPANY NAME TO BE MODIFY cout<<"COMPANY NAME:";

gotoxy(20,18); cout<<company; gotoxy(35,18);

cout<<"Want to change the COMPANY"; gotoxy(65,18);

flag=0; while(1)

{

gets(tmpnm2); if(strlen(tmpnm2)!=0)

{

flag=1;

break;

}

if(strlen(tmpnm2)==0)

{ flag=0; break;

}

}

if(flag==1)

{ strcpy(company,tmpnm2);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*COMPANY NAME TO BE MODIFIED ENDS HERE gotoxy(5,19);

cout<<"\*\*\*\*\*\*\*\*MODIFY\*\*PRODUCT PRICE - QUANTITY - DISCOUNT \*\*\*\*\*\*\*\*\*\*\*";

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* gotoxy(5,20);

cout<<"PRICE:";

gotoxy(15,20); float tmppr=0; flag=-1; while(1)

{

cin>>tmppr; if(tmppr!=0)

{

flag=1;

break;

}else

{ flag=0; break;

}

}

if(flag==1)

{ price=tmppr;

}

else

{

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* gotoxy(22,20);

cout<<"INCREASE QUANTITY:";

gotoxy(45,20); int tmpqty=0;

cin>>tmpqty; qty=q; qty=qty+tmpqty;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* gotoxy(50,20);

cout<<"DISCOUNT:";

gotoxy(63,20); float tmpdis=0; cin>>tmpdis; dis=tmpdis;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* gotoxy(5,22);

cout<<"=====================================================================";

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*class function outside fstream fp;

product pr;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// function to write in file

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* void write\_book()

{

fp.open("product.dat",ios::out|ios::app); clrscr();

intromain();

int rnn=getproduct(); if(rnn>100)

{

rnn=1;

}

pr.create\_prod(rnn); fp.write((char\*)&pr,sizeof(product)); fp.close();

gotoxy(10,20);

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*PRODUCTS RECORD SAVED\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* "; cin.ignore();

getch();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*check the product number already given or not\*\*\*\*\*\* int getproduct()

{

ifstream objiff; product st;

int count=0; objiff.open("product.dat",ios::binary); objiff.seekg(0,ios::beg);

if(!objiff)

{

cout<<"File could not be open !! Press any Key..."; getch();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*temporary hiding these lines while(objiff.read((char \*) &st, sizeof(product)))

{

count++;

}

//\*\*\*\*\*\*\*\*\*\*\*jump to the last line objiff.seekg(count-sizeof(st),ios::beg); objiff.read((char \*) &st, sizeof(product)); count=st.retpno();

count++; objiff.close();

return count;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// function to read specific record from file

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* void display\_sp(int n)

{

int flag=0; fp.open("product.dat",ios::in); if(!fp)

{

cout<<"File could not be open !! Press any Key...";

getch(); return;

}

while(fp.read((char\*)&pr,sizeof(product)))

{

if(pr.retpno()==n)

{

clrscr(); intromain(); pr.show\_prod(); flag=1;

}

}

fp.close(); if(flag==0)

cout<<"\n\nrecord not exist"; getch();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// function before place order and then call place order function

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int before\_order()

{ int f=-1,num=0; customer cust; clrscr();

intromain(); gotoxy(10,8);

cout<<"\*\*\*\*\*ENTER THE CUST ID TO BE SEARCHED:";

gotoxy(55,9); cin>>num;

ifstream inFile;

inFile.open("customer.dat",ios::binary); if(!inFile)

{

cout<<"File could not be open !! Press any Key..."; getch();

return -1;

}

while(inFile.read((char \*) &cust, sizeof(customer)))

{

if(cust.getcustid()==num)

{ clrscr();

intromain(); cust.showcustdatamulti(); f=1;

//tmprt=(int)inFile.tellg(); break;

}

}

inFile.close(); return f;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// function to place order and generating bill for PRODUCT PURCHASED

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void place\_order()

{

order o1[50]; int c=0,pr1=0;

float amt=0,damt=0,total=0,ttaxt=0; int k=0,q1;

char ch='Y';

int value=before\_order(); if(value==1)

{

clrscr(); prod\_tabular(); clrscr(); intromain(); gotoxy(30,4);

cout<<"============================";

gotoxy(30,5);

cout<<" PLACE YOUR ORDER";

gotoxy(30,6); cout<<"============================";

do

{

clrscr();

gotoxy(5,10);

cout<<"ENTER THE PRODUCT NO: ";

gotoxy(55,10); cin>>pr1; k=search(pr1); if(k>0)

{

gotoxy(5,12);

cout<<"Enter the Quantity:"; gotoxy(28,12);

cin>>q1; changeqty(pr1,q1); copyme(k,o1,q1,c);

}else

{

}

clrscr();

cout<<"PRODUCT not found";

gotoxy(10,20);

cout<<"Do you want purchase more (Yes[ y or Y ] or NO [n or N])"; gotoxy(10,21);

cin>>ch;

} while(ch=='y' || ch=='Y');

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*while ends by these lines

clrscr(); gotoxy(20,20);

cout<<"Thank You For Placing The Order ";

getch();

clrscr(); intromain(); gotoxy(13,5);

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*INVOICE\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"; gotoxy(1,6);

cout<<"PR.No."<<setw(12)<<"NAME"<<setw(10)<<"Qty"<<setw(15)<<"Price"<<setw(13)<<"Amount"<<setw(23)<<"Am ount- discount";

gotoxy(3,7);

cout<<" ";

int yy=8;

for(int x=0;x<c;x++)

{

gotoxy(1,yy); cout<<o1[x].prodid1; gotoxy(10,yy); cout<<o1[x].pname1; gotoxy(27,yy); cout<<o1[x].qty1; gotoxy(40,yy); cout<<"Rs."<<o1[x].price1; gotoxy(50,yy); amt=o1[x].qty1\*o1[x].price1; cout<<"Rs."<<amt; damt=amt-o1[x].dis1; gotoxy(65,yy); cout<<"Rs."<<damt; total+=damt;

//ttaxt+=o1[x].tax1;

}else

{

yy++;

}

ttaxt=5; gotoxy(3,yy);

cout<<" ";

yy++; gotoxy(25,yy); cout<<"TOTAL:";

gotoxy(35,yy); cout<<total; yy++; gotoxy(25,yy); cout<<"TAX%:";

gotoxy(36,yy); cout<<"+"<<ttaxt; yy++; gotoxy(25,yy);

cout<<" ";

yy++; gotoxy(25,yy);

cout<<"NET TOTAL:";

gotoxy(35,yy); cout<<(total+((ttaxt\*total)/100)); yy++;

gotoxy(3,yy);

cout<<" ";

clrscr(); gotoxy(20,30);

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*YOUR ID IS WRONG \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*change at the above getch();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*fuction to copy all record to a structure

void copyme(int k2,order order1[50],int q1,int &c2)

{

ifstream objiff2("product.dat",ios::binary); product bk1;

objiff2.seekg(k2-sizeof(product)); objiff2.read((char\*)&bk1,sizeof(product)); strcpy(order1[c2].pname1,bk1.getname()); strcpy(order1[c2].compy1,bk1.getcompany());

order1[c2].dis1=bk1.retdis(); order1[c2].price1=bk1.retprice();

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*copy record\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* order1[c2].qty1=q1;

c2++;

objiff2.close();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// INTRODUCTION FUNCTION

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* void intro()

{

gotoxy(4,2);

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CANTEEN-MANAGEMENT\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"; gotoxy(12,4); cout<<"===========================================================";

gotoxy(5,5); cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*S=Y=S=T=E=M\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"; gotoxy(50,10);

cout<<"PROJECT:";

gotoxy(50,12);

cout<<"MADE BY : SNEHIT CHANDRA";

gotoxy(10,14);

cout<<"---------------SCHOOL : DOON PUBLIC SCHOOL ";

getch();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// ADMINSTRATOR MENU1 FUNCTION

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* void admin\_menu1()

{

clrscr(); char ch2;

int num;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* clrscr();

intromain(); gotoxy(20,6);

cout<<"=================ADMIN MENU====================";

gotoxy(22,7);

cout<<"1.CREATE CUSTOMERS DETAILS";

gotoxy(22,8);

cout<<"2.DISPLAY ALL CUSTOMERS DETAILS";

gotoxy(22,9);

cout<<"3.SEARCH RECORD(QUERY) ";

gotoxy(22,10);

cout<<"4.MODIFY CUSTOMERS RECORDS";

gotoxy(22,11);

cout<<"5.DELETE CUSTOMERS RECORDS";

gotoxy(22,12);

cout<<"6.BACK TO MAIN MENU";

gotoxy(18,13);

cout<<"Please Enter Your Choice (1-6) "; gotoxy(55,13);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ch2=getche();

switch(ch2)

{

case '1': clrscr();

write\_customer(); break;

case '2': cust\_tabular(); break;

case '3':

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* clrscr(); intromain(); gotoxy(10,8);

cout<<"\*\*\*\*\*ENTER THE CUST ID TO BE SEARCHED:";

gotoxy(55,9); cin>>num; display\_cust\_sp(num);

break;

case '4':

clrscr();

intromain(); gotoxy(10,8);

cout<<"\*\*\*\*\*ENTER THE CUST ID TO BE SEARCHED AND MODIFY:";

gotoxy(55,9); cin>>num;

modify\_cust\_record(num);

break; case '5':

clrscr();

intromain(); gotoxy(10,8);

cout<<"\*\*\*\*\*ENTER THE CUST ID TO BE SEARCHED AND TO DELETE:";

gotoxy(55,9); cin>>num;

deletecust\_record(num); break;

case '6': break; default:cout<<"\a";admin\_menu1();

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*MENU 1 CUSTOMERS ENDS HERE\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// ADMINSTRATOR MENU2 FUNCTION

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* void admin\_menu()

{

clrscr(); char ch2;

int num;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* clrscr();

intromain(); gotoxy(20,6);

cout<<"=================ADMIN MENU====================";

gotoxy(22,7); cout<<"1.CREATE PRODUCTS";

gotoxy(22,8);

cout<<"2.DISPLAY ALL PRODUCTS AVAILABEL";

gotoxy(22,9);

cout<<"3.SEARCH RECORD(QUERY) ";

gotoxy(22,10); cout<<"4.MODIFY PRODUCTS";

gotoxy(22,11); cout<<"5.DELETE PRODUCTS";

gotoxy(22,12);

cout<<"6.BACK TO MAIN MENU";

gotoxy(18,13);

cout<<"Please Enter Your Choice (1-6) "; gotoxy(55,13);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ch2=getche();

switch(ch2)

{

case '1': clrscr();

write\_book(); break;

case '2': prod\_tabular();//display\_all(); break;

case '3':

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* clrscr(); intromain(); gotoxy(10,8);

cout<<"\*\*\*\*\*ENTER THE PROD ID TO BE SEARCHED:";

gotoxy(55,9); cin>>num; display\_sp(num);

break;

case '4':

clrscr();

intromain();

gotoxy(10,8);

cout<<"\*\*\*\*\*ENTER THE PROD ID TO BE SEARCHED AND MODIFY:";

gotoxy(55,9); cin>>num; modify\_record(num);

break; case '5':

clrscr(); intromain(); gotoxy(10,8);

cout<<"\*\*\*\*\*ENTER THE PROD ID TO BE SEARCHED AND TO DELETE:";

gotoxy(55,9); cin>>num;

delete\_record(num); break;

case '6': break; default:cout<<"\a";admin\_menu();

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THE MAIN FUNCTION OF PROGRAM

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* void main()

{

char ch; intro(); do

{

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*TEMPORARY\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* clrscr();

intromain(); gotoxy(20,6);

cout<<"=================MAIN MENU====================";

gotoxy(28,7);

cout<<"01. PRODUCTS REPORT GENERATOR";

gotoxy(28,8);

cout<<"02. ADMINISTRATOR";

gotoxy(28,9); cout<<"03. EXIT";

gotoxy(20,10); cout<<"==============================================";

gotoxy(25,12);

cout<<"Please Select Your Option (1-3) "; gotoxy(30,14);

ch=getche(); switch(ch)

{

case '1': clrscr();

orderk=0; place\_order(); getch(); break;

case '2': middleadminmenu();

break; case '3':exit(0); default :cout<<"\a";

}

}while(ch!='3');

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*main intro void intromain()

{ clrscr();

gotoxy(1,2);

cout<<"\*\*\*\*\*\*\*\*\*\*CANTEEN\*\*\*\*\*\*MANAGEMENT\*\*\*\*SYSTEM\*\*\*\*\*\*PROJECT\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"; gotoxy(1,3);

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// DISPLAY ALL THE PRODUCT TABULAR FORM

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* void prod\_tabular()

{

int r=0,col=10;

product st; ifstream inFile;

inFile.open("product.dat",ios::binary); if(!inFile)

{

cout<<"File could not be open !! Press any Key...";

getch(); return;

}

display\_all();

while(inFile.read((char \*) &st, sizeof(product)))

{

if(r<=12)

{ r++;

st.showall(col); col++;

}else

{

gotoxy(20,30);

cout<<"--------------press any key ";

getch();

clrscr(); display\_all(); col=10;

r=0;

}

}

inFile.close(); getch();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*tabular forms ends\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*tabulars forms headings\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//function to display all the records of product

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* void display\_all()

{

clrscr(); intromain(); gotoxy(1,5);

cout<<" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*PRODUCTS DETAILS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"; gotoxy(1,6);

cout<<"================================================================================";

gotoxy(1,7); cout<<"PROD.NO"<<setw(10)<<"NAME"<<setw(15)<<"COMPANY"<<setw(15)<<"PRICE"<<setw(15)<<"QUANTITY"<< setw(10)<<"DISCOUNT";

gotoxy(1,8); cout<<"================================================================================";

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// MODIFY RECORD

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* void modify\_record(int n)

{

product st,temp;

char tmpnm[50],tmpcompany[50]; ifstream inFile;

int fpos=-1; inFile.open("product.dat",ios::binary); if(!inFile)

{

cout<<"File could not be open !! Press any Key...";

getch(); return;

}

int flag=0;

while(inFile.read((char \*) &st, sizeof(product)))

{

if(st.retpno()==n)

{ clrscr();

intromain(); st.showdatamulti(); flag=1;

}

}

inFile.close();

if(flag==0)

cout<<"\n\nrecord not exist";

else

{

//\*\*\*\*\*\*\*modifying the records starts here fstream File;

File.open("product.dat",ios::binary|ios::in|ios::out); if(!File)

{

cout<<"File could not be open !! Press any Key..."; getch();

return;

}

while(File.read((char \*) &st, sizeof(product)))

{

if(st.retpno()==n)

{ fpos=(int)File.tellg(); break;

}

}

File.seekp(fpos-sizeof(product),ios::beg); strcpy(tmpnm,st.getname()); strcpy(tmpcompany,st.getcompany());

int q1=st.getqty(); gotoxy(1,12);

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"; gotoxy(1,13);

cout<<"================ENTER NEW VALUES FOR THE RECORDS GIVEN ABOVE=================";

temp.modifydata(n,tmpnm,tmpcompany,q1); File.write((char \*) &temp, sizeof(product)); File.close();

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// DELETE THE RECORD OF THE PRODUCTC NOT AVAILABLE

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* void delete\_record(int n)

{

product st;

ifstream inFile; inFile.open("product.dat",ios::binary); if(!inFile)

{

cout<<"File could not be open !! Press any Key..."; getch();

return;

}

int flag=0;

while(inFile.read((char \*) &st, sizeof(product)))

{

if(st.retpno()==n)

{ clrscr();

intromain(); st.showdatamulti(); flag=1;

}

}

inFile.close(); char ch; if(flag==0)

cout<<"\n\nrecord not exist";

else

{

//\*\*\*\*\*\*\*deletion of the records starts from here gotoxy(1,15);

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"; gotoxy(5,16);

cout<<"======DO YOU WANT TO DELETE THE RECORDS GIVEN ABOVE[YES(Y) OR NO (N)========";

gotoxy(2,17); cin>>ch;

if (toupper(ch)=='Y')

{

ofstream outFile; outFile.open("Temp1.dat",ios::binary); ifstream objiff("product.dat",ios::binary); objiff.seekg(0,ios::beg); while(objiff.read((char \*) &st, sizeof(product)))

{

if(st.retpno()!=n)

{

outFile.write((char \*) &st, sizeof(product));

}

}

outFile.close(); objiff.close(); remove("product.dat");

rename("Temp1.dat","product.dat"); againopenandclose(); gotoxy(30,20);

cout<<"----------------------------Record Deleted ";

}

}

getch();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*delete record ends\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* void againopenandclose()

{

ifstream inFile;

product st; inFile.open("product.dat",ios::binary); if(!inFile)

{

getch(); return;

}

while(inFile.read((char \*) &st, sizeof(product)))

{

}

inFile.close();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*search the product\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* int search(int p)

{

product st; int tmprt=0;

ifstream inFile;

inFile.open("product.dat",ios::binary); if(!inFile)

{

cout<<"File could not be open !! Press any Key..."; getch();

return -1;

}

int flag=0;

while(inFile.read((char \*) &st, sizeof(product)))

{

break;

if(st.retpno()==p)

{ clrscr();

intromain(); st.showdatamulti(); flag=1; tmprt=(int)inFile.tellg();

}

}

inFile.close(); if(flag==0)

return 1;

//cout<<"\n\nrecord not exist";

else

{

}

}

return tmprt;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*change quantity\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* void changeqty(int pr1,int q11)

{

product st; int fpos=-1; fstream File;

File.open("product.dat",ios::binary|ios::in|ios::out); if(!File)

{

cout<<"File could not be open !! Press any Key..."; getch();

return;

}

while(File.read((char \*) &st, sizeof(product)))

{

if(st.retpno()==pr1)

{ fpos=(int)File.tellg(); break;

}

}

File.seekp(fpos-sizeof(product),ios::beg); int q1=st.getqty();

q1=q1-q11; st.setqty(q1);

File.write((char \*) &st, sizeof(product)); File.close();

}

void middleadminmenu()

{

char ch; do

{

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*TEMPORARY\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* clrscr();

intromain(); gotoxy(20,6);

cout<<"===========CUSTOMER'S-PRODUCT'S MENU===========";

gotoxy(28,7);

cout<<"01. CUSTOMER'S MENU";

gotoxy(28,8);

cout<<"02. PRODUCT'S MENU";

gotoxy(28,9);

cout<<"03. BACK TO MAIN";

gotoxy(20,10); cout<<"==============================================";

gotoxy(25,12);

cout<<"Please Select Your Option (1-3) "; gotoxy(30,14);

ch=getche(); switch(ch)

{

case '1': admin\_menu1();

break; case '2': admin\_menu();

break; case '3': break;

default :cout<<"\a";admin\_menu();

}

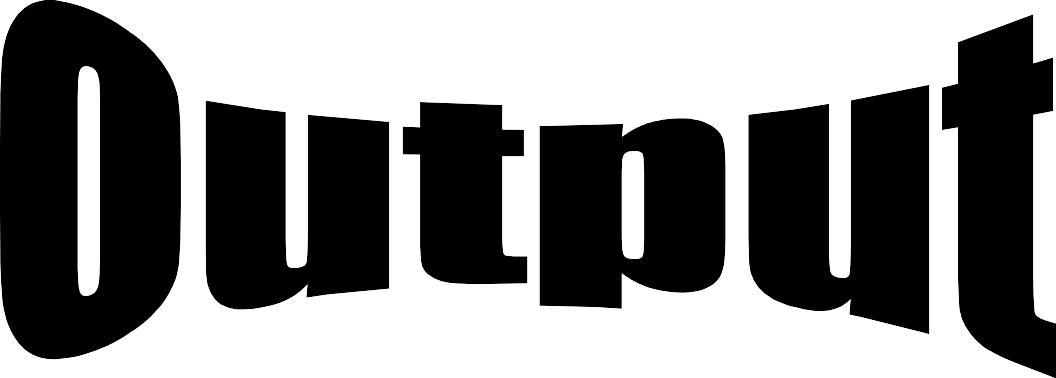
}while(ch!='3');

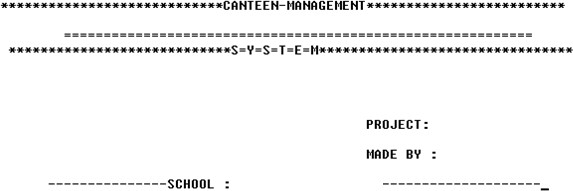
}

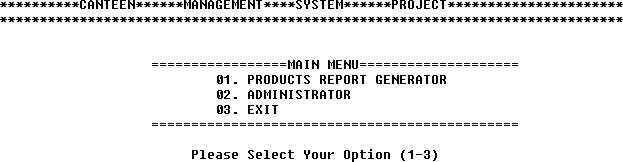
//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

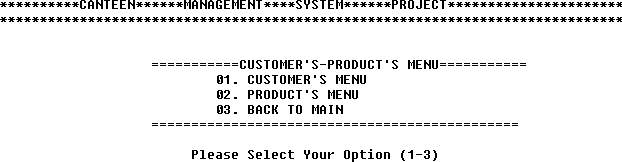
// END OF PROJECT

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

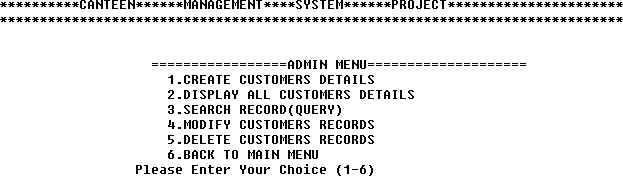








### if choice is 1:



\*\*\*\*\*\*\*\*\*\*CAHTEEH\*\*\*\*\*\*MAHAGEMEHT\*\*\*\*SYSTEM\*\*\*\*\*\*PROJECT\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CUST HO: ADDRESS:

PHONE **HO.:**

2

nanital

8659235689

HAME OF CUST: xyz

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CUSTOMER RECORD SAUED\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*CAHTEEH\*\*\*\*\*\*MAHAGEMEHT\*\*\*\*SYSTEM\*\*\*\*\*\*PROJECT\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CUSTOMER DETAILS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CUST.HO HAME ADDRESS PHONE HO

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | abc | haldwani | 8956235623 |
| 2 | xyz | nanital | 8659235689 |

\*\*\*\*\*\*\*\*\*\*CAHTEEH\*\*\*\*\*\*MAHAGEMEHT\*\*\*\*SYSTEM\*\*\*\*\*\*PROJECT\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

|  |  |  |
| --- | --- | --- |
| \*\*\*\*\*EHTER | THE CUST ID TO BE SEARCHED: | 2 |
| CUST HO: ADDRESS:  PHONE HO.: | 2 HAME OF CUST:  nanital  8659235689 | xyz |

\*\*\*\*\*EHTER THE CUST ID TO BE SEARCHED AHD MODIFY:

2\_



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CUST HO: ADDRESS: PHONE HO.:

2 HAHE OF CUSTD: nanital

8659235689

*xyz*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

================ENTER HEW UALUES FOR THE RECORDS GIUEH ABOUE=================

-------------------WANT TO HODIFY -------------------------------

CUST HO: 2 HAHE OF CUST: *xyz*

Want to change the name of customer



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

=================ADHIH HEHU==================== 1.CREATE PRODUCTS

* 1. DISPLAY ALL PRODUCTS AUAILABEL
  2. SEARCH RECORD(QUERY)
  3. HODIFY PRODUCTS
  4. DELETE PRODUCTS
  5. BACK TO HAIN HEHU

Please Enter Your Choice (1-6)

PROD HO: COHPAHY:

2

parle

HAHE OF PROD: QUANTITY

popins

500

PROD PRICE 2

DISCOUNT

1.2

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*PRODUCTS RECORD SAUED\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

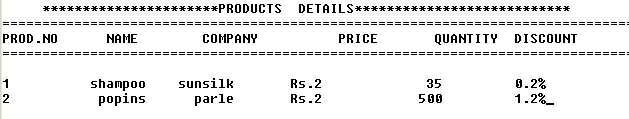
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*PRODUCTS DETAILS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* PROD.HO HAHE COHPAHY PRICE QUANTITY DISCOUNT

|  |  |  |
| --- | --- | --- |
| 1 shampoo sunsilk Rs.2  \*\*\*\*\*ENTER THE PROD ID TO BE | 35  SEARCHED: | 0.2% |
|  |  | 2\_ |
| PROD HO: 2 HAHE OF PROD: | popins |  |
| COHPAHY: parle PROD PRICE: | 2 |  |
| DISCOUNT: 1.2% QUANTITY: | 500 |  |

### product report generator:



if found



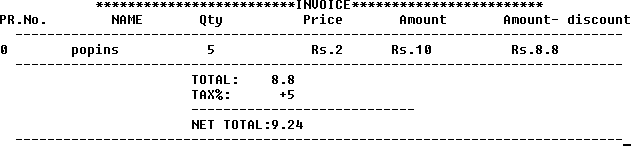












# REQUIREMENTS

## HARDWARE REQUIRED

* Printer, to print the required documents of the project
* Compact Drive
* Processor : intel
* Ram : 512 MB or more than 512MB
* Harddisk : 80 GB or more than 80GB.

## SOFTWARE REQUIRED

* Operating system : Windows XP
* Turbo C++, for execution of program and Ms word, for presentation of output.

# ADVANTAGES

* Customer does not have to wait in long queues.
* Customer does not need to worry about buying products.
* This software reduces paper work.
* It is easy to handle customer’s record and products records for future
* This software saves the time.
* Information of the customer stores permanently.

# DISADVANTAGES

* This system suitable for only small canteens.
* Online facility is not available.

# CONCLUSION

This software is efficient in maintaining customer’s details and can easily perform operations on customer’s records and also works to handles the information of the products available in a canteen. This software also reduces the work load of the canteens.

In future, this system can launch web site for easy online canteen facilities.

# SYSTEM DESIGN

**Then we began with the design phase of the system. System design is a solution, a “HOW TO” approach to the creation of a new system. It translates system requirements into ways by which they can be made operational. It is a translational from a user oriented document to a document oriented programmers. For that, it provides the understanding and procedural details necessary for the implementation. Here we use Flowchart to supplement the working of the new system. The system thus made should be reliable, durable and above all should have least possible maintenance costs. It should overcome all the drawbacks of the Old existing system and most important of all meet the user requirements.** START

* + 1. generate the invoice 2.administrator

3.exit

IF CHOICE = 1 IF CHOICE = 2 IF CHOICE = 3 THEN

IT WORKS FOR ADDING DELETING MODIFYING SEARCHING THE RECORD OF CUSTOMER AND PRODUCTS

IT GENERATE THE INVOICE AFTER THE PRODUCT PURCHASED BY THE CUSTOMERS

QUIT

STOP

BIBLIOGRAPHY

The great help from our faculty members and my project guide that led the successful completion of the project. Besides that, I took the help of some books and websites to develop the project. They are : --

Sumita Arora – Computer Science with C++

E. Balagurusami – C++ Robert Lafore – Turbo C++