

**PROJECT REPORT ON**

**“HOTEL MANAGEMENT”**

**MD. SAIFUL ISLAM SURAJ DANSENA**

For the partial fulfillment of the XII CBSE Practical Examination

Computer Science session 2016-2017

**Under the guidance of:**

**MR. NILESH MAHORE**

**(PGT Computer Science)**

****

**KENDRIYA VIDYALAYA NO.1 KALPAKKAM**

**CERTIFICATE**

This is to certify that the project titled

“HOTEL MANAGEMENT” is successfully developed and completed by SAIFUL AND SURAJ student of Class XII under the guidance of MR. NILESH MAHORE (PGT Computer Science) in C++.

We developed this project in the session 2016-2017 for the partial fulfillment of the Central Board of Secondary Education Practical Examination 2016 as a part of practical under the subject Computer Science.

This project has not been submitted for any other course, degree or diploma.

**PGT COMPUTER SCIENCE:**

**Mr. Nilesh Mahore**

**KendriyaVidyalaya No.1 Kalpakkam**

**PRINCIPAL:**

**Shri V Sivaji**

**Kendriya Vidyalaya No.1 Kalpakkam**

**ACKNOWLEDGEMENT**

I am thankful to our teacher Mr. Nilesh Mahore (PGT Computer Science), KendriyaVidyalaya No 1, Kalpakkam for his valuable practical and theoretical guidance and constant encouragement that have been a source of inspiration to my entire project.

I am also very thankful to our Principal Sir, Shri V Sivaji who have provide me all the equipments and arrangements which helped me a lot to complete my project within the due time.

I am also thankful to my parents for their blessings and allow me to complete my project by supporting in all manners. Finally, my thanks to all the friends without their support and help it is not possible to develop my project successfully.

**MD. SAIFUL ISLAM SURAJ DANSENA**

Class XII

KendriyaVidyalaya No.1, Kalpakkam

**“HOTEL MANAGEMENT”**

**INTRODUCTION**

It is reliable software. Whenever the user wants to retrieve any information from the database, it returns with the correct information. Provided, the information stored is a correct one.

    As mentioned in the requirements, the Hospital Management system involves the following function or process  
Hotel Management system consists of four modules. They are  
    Module 1: ENQUIRY  
    Module 2: RESERVATION   
    Module 3: STATUS  
    Module 4: MODIFICATION  
    Module 5: DELETION  
    Module 6: EXIT  
  
  **Enquiry Module:**  
    In this module the user enquires the detail of the rooms available.  
  **Reserveation Module:**  
    In this module the user reserves the room.  
  **Status Module:**  
    In this module the user is checking the status of the hotel.  
  **Modification Module:**  
    In this module the user modifies the detail of the customer.  
  **Deletion Module:**  
    In this module the user deletes the detail of the customer.  
  **Exit:**  
    In this module the user can quit from the project.

**SOFTWARE REQUIREMENTS:**

 Compiler : Turbo C++ (or) Dev C++

Operating system:    Any Operating System (Window or Linux)

**HARDWARE REQUIREMENTS:**

    Processor     :     3.06 GHZ operating speed.  
    Hard Disk    :      80GB Capacity.  
    RAM           :       256MB  
    Monitor       :       CRT 17’  
    Mouse         :      Not required  
    Keyboard    :       Alphanumeric keys.

     The specified hardware requirements are the minimum requirements that are needed to implement in the project.

**SOURCE CODE OF THE PROGRAM**

//HOTEL MANAGEMENT SYSTEM//

//TAJ HOTEL//

#include<iostream.h>

#include<fstream.h>

#include<string.h>

#include<stdlib.h>

#include<stdio.h>

#include<conio.h>

int cnt=0,sno=0;

class hotel

{

public:

int choice,x,y,m,m1,a,b,i,dr,rno,s,d,j,k;

int sr;

char p1[25],r[25],r1[25],temp[28],ar,rc,n[25],n1[25],o[25],o1[25],p[25];

char p4[25],p5[25],r4[25],r5[25],vchoice,mf,n4[25],n5[25],o4[25],o5[25];

hotel()

{

sr = 0;

dr = 0;

}

int ret\_single();

int ret\_double();

void wel();

void menu();

void enquiry();

void reser();

void re\_single();

void re\_double();

void status();

void st\_single();

void st\_double();

void modifi();

void mo\_single();

void mo\_double();

void deletion();

void de\_single();

void de\_double();

};

int hotel::ret\_single()

{

cnt = 0;

fstream f1;

f1.open("one.dat",ios::in|ios::binary);

f1.seekg(0,ios::beg);

while(f1.read((char\*)this,sizeof(hotel)))

{

cnt = cnt + 1;

}

return cnt;

}

int hotel::ret\_double()

{

cnt = 0;

fstream f1;

f1.open("double.dat",ios::in|ios::binary);

f1.seekg(0,ios::beg);

while(f1.read((char\*)this,sizeof(hotel)))

{

cnt = cnt + 1;

}

return cnt;

}

//WELCOME//

void hotel:: wel()

{

textbackground(GREEN);

textcolor(WHITE);

cout<<"\n\n\n";

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

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

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

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

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

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

cout<<"\n\n\n\n";

cout<<"\t\tSAIFUL \t\t\t\t\tSURAJ";

cout<<"\n\n Press Any Key To Continue... ";

getch();

menu();

}

// MENU FUNCTION //

void hotel::menu()

{

clrscr();

cout<<"\n\n\n\n\n\t\t\t\t\tMENU:"<<endl;

cout<<"\t1. ENQUIRY"<<endl;

cout<<"\t2. RESERVATION "<<endl;

cout<<"\t3. STATUS"<<endl;

cout<<"\t4. MODIFICATION"<<endl;

cout<<"\t5. DELETION"<<endl;

cout<<"\t6. EXIT"<<endl;

cout<<"\n\n\n\n\t\t\tENTER YOUR CHOICE....";

cin>>choice;

switch(choice)

{

case 1:

enquiry();

case 2:

reser();

case 3:

status();

case 4:

modifi();

case 5:

deletion();

case 6:

exit(0);

break;

default:

{

cout<<"\n\n\nILLEGAL OPERATION...."<<endl;

cout<<"ENTER AGAIN";

getch();

menu();

}

}

}

// ENQUIRY //

void hotel::enquiry()

{

int a,d,tmp;

char ch,c,n;

clrscr();

cout<<"\n\t\t\t\tENQUIRY SESSION"<<endl;

cout<<" \t\t\t\t------- -------"<<endl;

cout<<"\n\tCHECK ROOM DETAILS FOR"<<endl;

cout<<"\n\tS : FOR SINGLE ROOM"<<endl;

cout<<"\n\tD : FOR DOUBLE ROOM"<<endl;

cout<<"\n\t\tENTER YOUR CHOICE"<<endl;

cin>>n;

switch(n)

{

case 'S':

tmp=ret\_single();

d=50-tmp;

tmp=0;

cout<<"\nSINGLE ROOM LEFTOUT : "<<d<<endl;

getch();

menu();

break;

case 'D':

tmp = ret\_double();

a=50-tmp;

tmp=0;

cout<<"\nDOUBLE ROOM LEFTOUT : "<<a<<endl;

getch();

menu();

}

if(ch=='N')

{

menu();

}

else

gotoxy(30,48);

cout<<"\nILLEGAL CHOICE.... \nMAKE SURE CAPS LOCK IS ON";

getch();

menu();

}

//RESERVATION//

void hotel::reser()

{

char a,b;

clrscr();

cout<<"\n\t\t\tRESERVATION COUNTER";

cout<<"\n\t\t\t-------------------\n\n"<<endl;

cout<<"\nDo You Want To Reserve a Room? (Y/N) \n";

cin>>a;

if(a=='Y')

{

cout<<"\n\nS :: SINGLE ROOM"<<endl;

cout<<"\n\nD :: DOUBLE ROOM"<<endl;

cin>>b;

switch(b)

{

case 'S':

re\_single();

break;

case 'D':

re\_double();

break;

}

}

else if(a=='N')

menu();

else

cout<<"\n\nILLEGAL CHOICE...... \nMAKE SURE CAPS LOCK IS ON";

getch();

menu();

}

//RESERVATION\_SINGLE ROOM//

void hotel::re\_single()

{

int ltno;

char ch;

fstream f;

clrscr();

f.open("one.dat",ios::in|ios::app|ios::binary);

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

if(f.tellg()==0)

ltno=1;

else

{

f.seekg(f.tellg()-sizeof(hotel),ios::beg);

f.read((char\*)this,sizeof(hotel));

ltno=sr+1;

}

f.seekp(0,ios::end);

sr=ltno;

cout<<"\n\t\t\tRESERVATION";

cout<<"\n\t\t\t------------\n\n"<<endl;

cout<<"\n\nROOM NO.:" <<sr<< " \nAVAILABLE";

cout<<"\n\nEnter the following details";

cout<<"\n\n CUSTOMER NAME :";

gets(n);

cout<<"\n\n ADDRESS :";

gets(r);

cout<<"\n\n CHECK IN DATE :";

gets(o);

cout<<"\n\n CHECK OUT DATE :";

gets(p);

f.write((char\*)this,sizeof(hotel));

f.close();

cout<<"\n\nROOM NO.:"<<sr<< " IS RESERVED\n\n";

cout<<"\n\n\n\nDo You Want To Reserve Another Room? (Y/N)";

cin>>ch;

if(ch=='Y')

{

reser();

}

if(ch=='N')

{

menu();

}

else

cout<<"\n\nILLEGAL CHOICE...... \NMAKE SURE CAPS LOCK IS ON";

getch();

menu();

}

//RESERVATION\_DOUBLE ROOM//

void hotel::re\_double()

{

int ltno;

char ch;

fstream f;

clrscr();

f.open("double.dat",ios::in|ios::app|ios::binary);

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

if(f.tellg()==0)

ltno=1;

else

{

f.seekg(f.tellg()-sizeof(hotel),ios::beg);

f.read((char\*)this,sizeof(hotel));

ltno=dr+1;

}

f.seekp(0,ios::end);

dr=ltno;

cout<<"\n\t\t\tRESERVATION";

cout<<"\n\t\t\t------------\n\n"<<endl;

cout<<"\n\nROOM NO.:" <<dr<< " \nAVAILABLE";

cout<<"\n\nEnter the following details";

cout<<"\n\n CUSTOMER NAME :";

gets(n1);

cout<<"\n\n ADDRESS :";

gets(r1);

cout<<"\n\n CHECK IN DATE :";

gets(o1);

cout<<"\n\n CHECK OUT DATE :";

gets(p1);

f.write((char\*)this,sizeof(hotel));

f.close();

cout<<"\n\n\nROOM NO.:"<<dr<< " \nRESERVED";

cout<<"\n\n\n\nDo You Want To Reserve Another Room? (Y/N)";

cin>>ch;

if(ch=='Y')

{

reser();

}

if(ch=='N')

{

menu();

}

else

cout<<"\n\nILLEGAL CHOICE..... \nMAKE SURE CAPS LOCK IS ON";

getch();

menu();

}

//MODIFICATION//

void hotel::modifi()

{

char a,b;

clrscr();

cout<<"\n\t\t\tMODIFICATION";

cout<<"\n\t\t\t------------\n\n"<<endl;

cout<<"Do You Want To Modify A Room? (Y/N)\n";

cin>>a;

if(a=='Y')

{

cout<<"\n\nS :: SINGLE ROOMS\n\n";

cout<<"\n\nR :: DOUBLE ROOMS\n\n";

cin>>b;

switch(b)

{

case 'S':

mo\_single();

break;

case 'D':

mo\_double();

break;

}

}

else if(a=='N')

menu();

else

cout<<"\n\nILLEGAL CHOICE.... \nMAKE SURE CAPS LOCK IS ON";

getch();

menu();

}

//MODIFICATION\_SINGLE ROOM//

void hotel::mo\_single()

{

clrscr();

fstream f;

int no,found,i,j;

char wish;

clrscr();

found=0;

cout<<"\n\n\t\t\t\tMODIFICATION";

cout<<"\n \t\t\t\t------------"<<endl;

cout<<"\nENTER ROOM NUMBER TO MODIFY:";

cin>>no;

f.open("one.dat",ios::in|ios::out|ios::binary);

while(f.read((char\*)this,sizeof(hotel)))

{

if(no==sr)

{

found=1;

cout<<"\n\nCUSTOMER NAME :"<<n;

cout<<"\n\nNEW CUSTOMER NAME :";

gets(n);

cout<<"\n\nADDRESS :"<<r;

cout<<"\n\nENTER NEW ADDRESS :";

gets(r);

cout<<"\n\nCHECK IN DATE :"<<o;

cout<<"\n\nENTER NEW CHECK IN DATE :";

gets(o);

cout<<"\n\nCHECK OUT DATE :"<<p;

cout<<"\n\nENTER NEW CHECK OUT DATE :";

gets(p);

cout<<"DETAILS ARE MODIFIED"<<endl;

f.seekg(f.tellg()-sizeof(hotel),ios::beg);

f.write((char\*)this,sizeof(hotel));

}

}

if(!found)

cout<<"\nROOM IS VACANT";

f.close();

getch();

menu();

}

//MODIFICATION\_DOUBLE ROOM//

void hotel::mo\_double()

{

fstream f1;

int num,found1;

clrscr();

found1=0;

cout<<"\n\n\t\t\t\tMODIFICATION";

cout<<"\n \t\t\t\t------------"<<endl;

cout<<"\nENTER ROOM NUMBER TO MODIFY:";

cin>>num;

f1.open("double.dat",ios::in|ios::out|ios::binary);

while(f1.read((char\*)this,sizeof(hotel)))

{

if(num==dr)

{

found1=1;

cout<<"\n\nCUSTOMER NAME :"<<n;

cout<<"\n\nNEW CUSTOMER NAME :";

gets(n);

cout<<"\n\nADDRESS :"<<r;

cout<<"\n\nENTER NEW ADDRESS :";

gets(r);

cout<<"\n\nCHECK IN DATE :"<<o;

cout<<"\n\nENTER NEW CHECK IN DATE :";

gets(o);

cout<<"\n\nCHECK OUT DATE :"<<p;

cout<<"\n\nENTER NEW CHECK OUT DATE :";

gets(p1);

getch();

cout<<"\t\nDETAILS ARE MODIFIED"<<endl;

f1.seekg(f1.tellg()-sizeof(hotel),ios::beg);

f1.write((char\*)this,sizeof(hotel));

}

}

if(!found1)

cout<<"\nROOM IS VACANT";

f1.close();

getch();

menu();

getch();

}

//STATUS//

void hotel::status()

{

char a,b;

clrscr();

cout<<"\n\t\t\tROOM STATUS\n";

cout<<" \t\t\t------\n\n";

cout<<"WANT TO KNOW ROOM STATUS? (Y/N)\n";

cin>>a;

if(a=='Y')

{

cout<<"\n\nS :: SINGLE ROOM";

cout<<"\n\nD :: DOUBLE ROOM";

cin>>b;

switch(b)

{

case 'S':

st\_single();

break;

case 'D':

st\_double();

break;

}

}

else if(a=='N')

menu();

else

{

cout<<"\n\nILLEGAL CHOICE..... \nMAKE SURE CAPS LOCK IS ON";

}

getch();

menu();

}

//STATUS\_SINGLE ROOM//

void hotel::st\_single()

{

fstream f1;

int na,found5,q;

clrscr();

found5=0;

clrscr();

f1.open("one.dat",ios::in|ios::binary);

f1.seekg(0,ios::beg);

cout<<"\n\t\t\tSTATUS\n";

cout<<" \t\t\t------"<<endl;

cout<<"\n\n\n\t\tRESERVATION DETAILS\n\n";

while(f1.read((char\*)this,sizeof(hotel)))

{

found5=1;

cout<<"\nROOM NUMBER : "<<sr<<endl;

cout<<"\nCUSTOMER'S NAME : "<<n<<endl;

}

if(!found5)

cout<<"ROOM'S ARE NOT RESERVED ";

cout<<"\n\n\n CUSTOMERS STATUS \n\n"<<endl;

cout<<" 1. Details"<<endl;

cout<<" 2. Menu"<<endl;

cout<<"Enter ur Choice"<<endl;

cin>>q;

if(q==1)

{

getch();

f1.close();

clrscr();

found5=0;

cout<<"\nENTER ROOM NUMBER TO VIEW:";

cin>>na;

f1.open("one.dat",ios::in|ios::binary);

f1.seekg(0,ios::beg);

while(f1.read((char\*)this,sizeof(hotel)))

{

if(na==sr)

{

found5=1;

cout<<"\n\n\nCUSTOMER NAME :"<<n;

cout<<"\n\n\nADDRESS :"<<r;

cout<<"\n\n\nCHECK IN DATE :"<<o;

cout<<"\n\n\nCHECK OUT DATE :"<<p;

}

}

if(!found5)

cout<<"\nROOM IS VACANT";

f1.close();

getch();

}

if(q==2);

{

menu();

}

}

//STATUS\_DOUBLE ROOM//

void hotel::st\_double()

{

fstream f1;

int na,found5,q;

clrscr();

found5=0;

clrscr();

f1.open("double.dat",ios::in|ios::binary);

f1.seekg(0,ios::beg);

cout<<"\n\t\t\t STATUS"<<endl;

cout<<" \t\t\t ------"<<endl;

cout<<"\n\n\n\t\tRESERVATION DETAILS\n\n";

while(f1.read((char\*)this,sizeof(hotel)))

{

found5=1;

cout<<"\n \tROOM NUMBER\t"<<dr;cout<<"\tCUSTOMER'S NAME\t"<<n1<<endl;

}

if(!found5)

cout<<"ROOM'S ARE NOT RESERVED ";

cout<<"\n\n\n CUSTOMERS STATUS \n\n"<<endl;

cout<<" 1. DETAILS"<<endl;

cout<<" 2. MENU"<<endl;

cout<<"ENTER CHOICE..."<<endl;

cin>>q;

if(q==1)

{

getch();

f1.close();

clrscr();

found5=0;

cout<<"\nENTER THE NUMBER TO VIEW:\n";

cin>>na;

f1.open("double.dat",ios::in|ios::binary);

f1.seekg(0,ios::beg);

while(f1.read((char\*)this,sizeof(hotel)))

{

if(na==dr)

{

found5=1;

cout<<"\n\n\nCUSTOMER NAME :"<<n1;

cout<<"\n\n\nADDRESS :"<<r1;

cout<<"\n\n\nCHECK IN DATE :"<<o1;

cout<<"\n\n\nCHECK OUT DATE :"<<p1;

}

}

if(!found5)

cout<<"\nROOM IS VACANT";

f1.close();

getch();

}

if(q==2);

{

menu();

}

}

//DELETION//

void hotel::deletion()

{

char a,b;

int cost;

clrscr();

cout<<"\n\t\t\tVACATING A ROOM"<<endl;

cout<<" \t\t\t---------------"<<endl;

cout<<"Do You Want To Vacate A Room? (Y/N)\n";

cin>>a;

if(a=='Y')

{

cout<<"\n\nS :: SINGLE ROOM"<<endl;

cout<<"\n\nD :: DOUBLE ROOM"<<endl;

cin>>b;

switch(b)

{

case 'S':

de\_single();

break;

case 'D':

de\_double();

break;

}

}

else if(a=='N')

menu();

else

cout<<"\n\nILLEGAL CHOICE..... \n MAKE SURE CAPS LOCK IS ON";

getch();

menu();

}

//DELETION\_SINGLE ROOM//

void hotel::de\_single()

{

fstream f1,f2;

int no,found;

clrscr();

found=0;

f1.open("one.dat",ios::in|ios::binary);

f2.open("tp.dat",ios::out);

cout<<"\n\n\t\t\tVACTING A ROOM";

cout<<"\n \t\t\t--------------\n\n";

cout<<"\nENTER THE SINGLE ROOM NUMBER TO BE VACATED:";

cin>>no;

f1.read((char\*)this,sizeof(hotel));

while(!f1.eof())

{

if(sr==no)

found=1;

else

f2.write((char\*)this,sizeof(hotel));

f1.read((char\*)this,sizeof(hotel));

}

if(!found)

cout<<"\nROOM IS VACANT ";

f1.close();

f2.close();

remove("one.dat");

rename("tp.dat","one.dat");

if(found==1)

cout<<"\n ROOM NUMBER " <<no<< " IS VACATED \n" ;

}

//DELETION\_DOUBLE ROOM//

void hotel::de\_double()

{

fstream f1,f2;

int no,found;

clrscr();

found=0;

f1.open("double.dat",ios::in|ios::binary);

f2.open("tp.dat",ios::out);

cout<<"\n\n\t\t\tVACATING A ROOM";

cout<<"\n \t\t\t---------------\n\n";

cout<<"\nENTER THE DOUBLE ROOM NUMBER TO BE VACATED:";

cin>>no;

f1.read((char\*)this,sizeof(hotel));

while(!f1.eof())

{

if(dr==no)

found=1;

else

f2.write((char\*)this,sizeof(hotel));

f1.read((char\*)this,sizeof(hotel));

}

if(!found)

cout<<"\nROOM IS VACANT ";

f1.close();

f2.close();

remove("double.dat");

rename("tp.dat","double.dat");

if(found==1)

cout<<"\nROOM NUMBER " <<no<< "IS VACATED \n" ;

}

//MAIN FUNCTION//

void main()

{

clrscr();

hotel a;

a.wel();

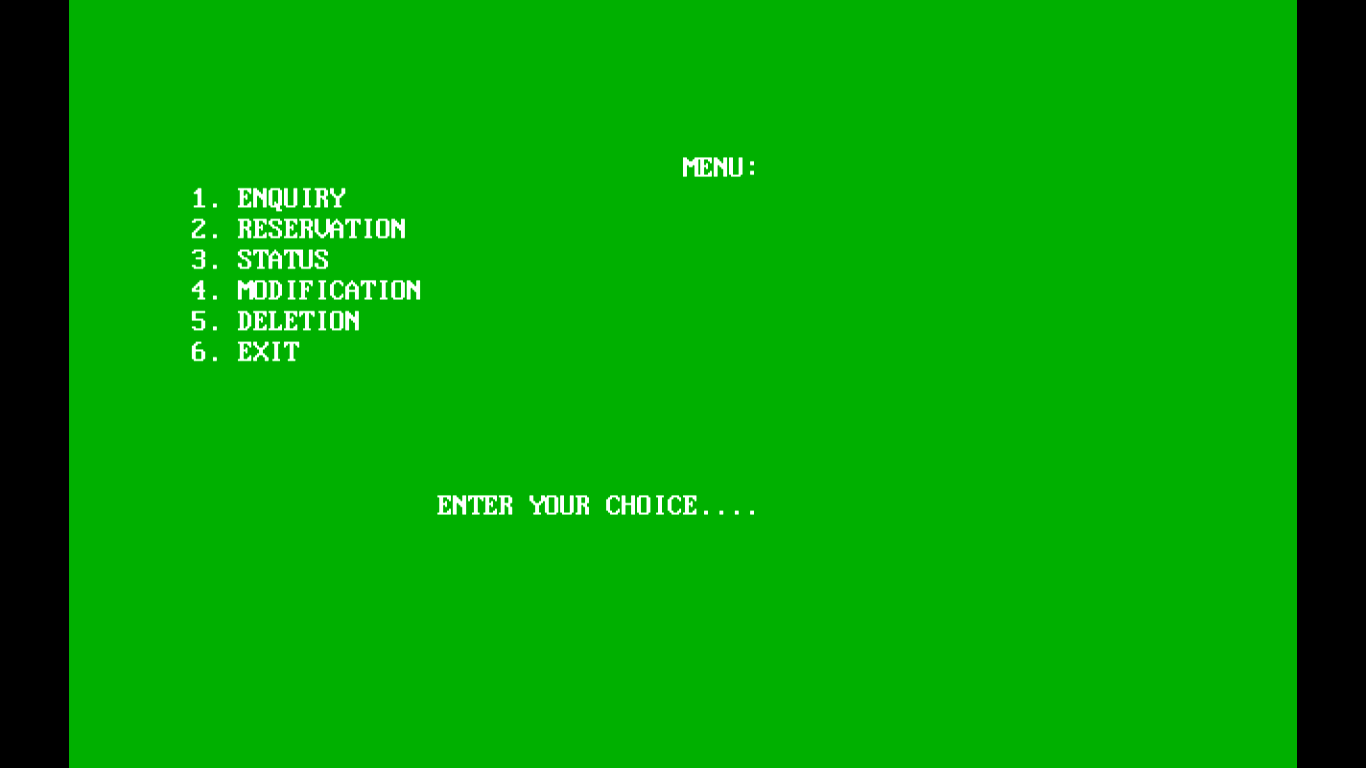
a.menu();

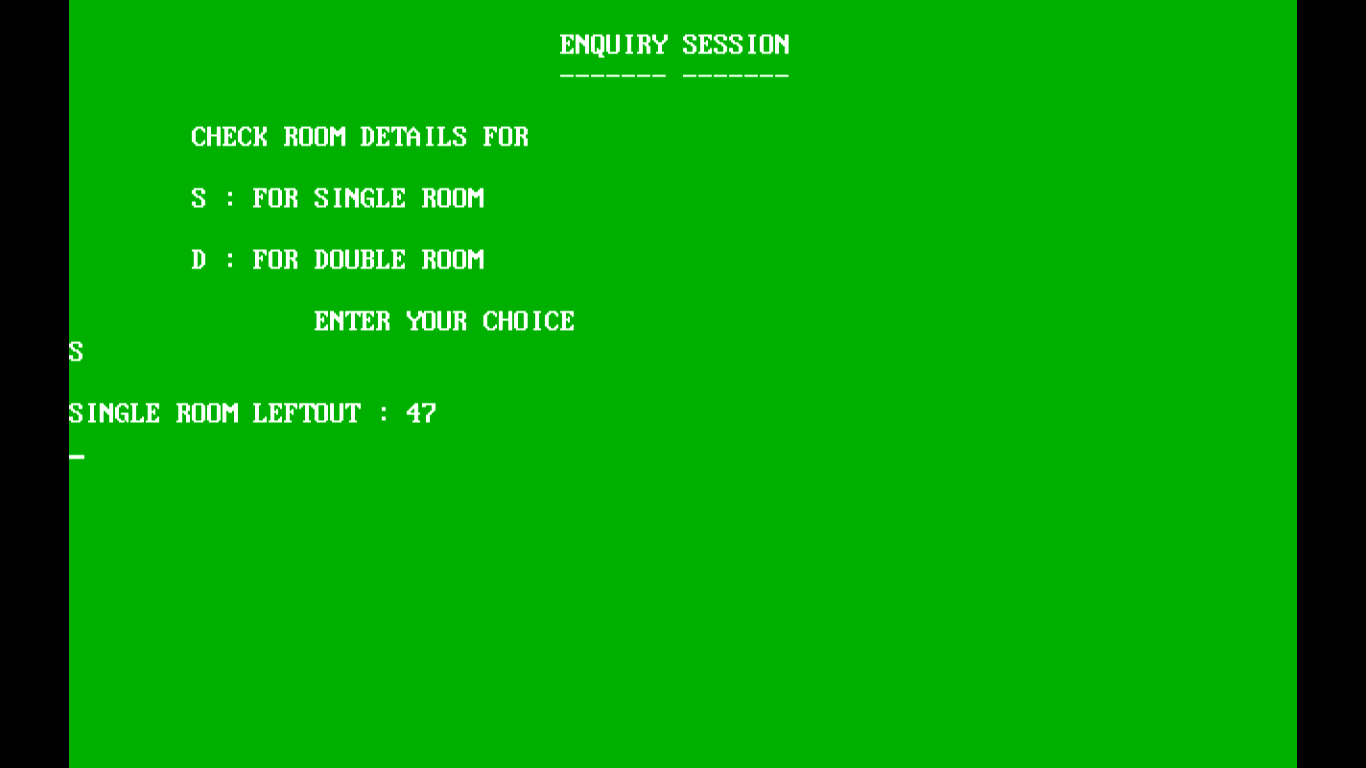
getch();

}

**SCREENSHOTS**

****

****



****

****

****

**MERITS AND DEMERITS:**

**MERITS**

1. The cost of maintenance of the software is very low.
2. The size of the software is less. This means, it is a portable one.
3. If any request is given, the time taken for the response is low. This makes the software more efficient.

**DEMERITS**

1. There is a chance of software corrupt.
2. There are chances of mistakes to take place.

**SHORTCOMINGS**

This project is designed mainly to ensure that it makes *Hotel management* easier and saves time.

It is user friendly, simple and effective.

**FUTURE ENHANCEMENT**

This project would be more useful and effective for *Hotel management.* It will make Hotel management work load easier.

The system provides excellent security of data at every level of user system interaction and also provide reliable storage and backup facilities.

**BIBLIOGRAPHY**

1. **Class XII Computer Science textbook**
2. **www.google.com**