Hotel Management System

Version 1.0 (2016-2017)

Computer Science Project

Developed By

Sanyam Agrawal

Sanket Maheshwari

Delhi Public School, R.K.Puram

www.dpsrkp.net

Index

Certificate

This is to certify that this Hotel Management System Computer Science project is developed by **Sanyam Agrawal** and **Sanket Maheshwari** under my supervision in the computer lab of Delhi Public School, R.K.Puram in the session 2016-2017. The work done by them is original.

Romi Sharma

Computer Science Teacher

Date: _____

Acknowledgement

I would like to express my sincere gratitude to my computer teacher Mrs. Romi Sharma for her vital support, guidance and encouragement without which this project would not come forth from my side. Who helped me completing the project by giving ideas,thoughts and made this project easy and accurate.

I wish to thank my parents for their undivided support and interest who inspired me and encouraged me to go my own way, without which I would be unable to complete my project.

Reference

1.Classnotes

Introduction

The project that we've worked upon is a **Hotel Management System,** based on C++ Programming.

Purpose : To build a Hotel Management System that can be used by the Manager as well as the Guests to avail the following **Features** :

Manager

- Adding Rooms
- Deleting Rooms
- Displaying Room Details
- Modifying Rooms
- Sorting Rooms
- Searching Rooms
- Changing Password

Guest

- Checking In
- Checking Out
- Displaying Room Details
- Adding Services

The concepts that are employed in this project are:

- Classes and Objects
- Data Structures
- File Handling

The whole program operates in a single run and we've created menus which we can toggle between.

Merits

- User Friendly
- Manager/Guest Mode
- Manager Mode Secured through Changeable Password
- Drop Down Menu System
- Quick toggle between different Menus
- Invalid Inputs Not Accepted
- Deluxe/Economy Rooms
- Sorting Records by Name/Room No.
- 24/7 Meal & Laundry Services

Limitations

- No Unique Code for every Guest
- Any Guest can be checked out by any User
- No feature of Advance Booking

Hope You like our Project.

Sanyam Agrawal

Sanket Maheshwari

XII G

Source Code

```
/*
Project Title: Hotel Management System
Version : 1.0
Developed By : Sanyam Agrawal, Sanket Maheshwari
School : DPS RK Puram
*/
```

```
#include<fstream.h>
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
#include<dos.h>
#include<string.h>
#include<iomanip.h>
#include<ctype.h>
```

```
class Hotel                          //Contains Booked Rooms
{
  char name[20];                     //Guest Name
  char age[4];                       //Guest Age
  char gen;                          //Guest Gender
  char city[50];                     //Guest Address
  char phno[15];
  float stay;                        //Duration of Stay
  int rno;                           //Room No.
  char rtype;                        //Room Type
  int charge;                        //Hotel Charges
  date datein;                       //Check-In Date
  date dateout;                      //Check-Out Date
```

```
  date currdate;                    //Current Date
  char in[10];
  char out[10];
  int rent;                         //Hotel Rent
  int laundry;                      //Laundry Counter
  int meals;                        //Meal Counter
  int laundrybill;                  //Laundry Bill
  int mealbill;                     //Meals Bill
  public :


  void In(int ro, char rt);           //To Enter Guest Data


  void Charges(int);           //To Calculate Current Charges


  void Out(int c=0);               //To Display Guest Details


  void Out2(int i)                 //To Display Guest Details
  {


cout<<endl<<setw(3)<<i<<setw(13)<<rno<<setw(23)<<name<<setw(14)<<(int
)datein.da_day<<"."<<(int)datein.da_mon<<"."<<datein.da_year;

  }


  Hotel()               //To Initialize Hotel class variables
  {
    charge=0;
    rent=0;
    meals=0;
    laundry=0;
    mealbill=0;
    laundrybill=0;
  }
```

```
void Bill()                     //To Calculate Current Bill
{
  gotoxy(2,12);
  cout<<" Bill :- ";
  gotoxy(2,14);
  cout<<" Rent            : "<<rent;
  gotoxy(2,15);
  cout<<" Meals           : "<<mealbill;
  gotoxy(2,16);
  cout<<" Laundry         : "<<laundrybill;
}


char* retname()                 //To return Guest Name
{ return name; }


char* retcity()                 //To return Guest Address
{ return city; }


char retrtype()                 //To return Room Type
{ return rtype; }


int retrno()                    //To return Room No.
{ return rno; }


int retcharge()                 //To return Charges
{ return charge; }


float retstay()                 //To return Days of Stay
{ return stay; }
```

```
  void AMeal();                         //To Add Meal

  void ALaundry();                      //To Add Laundry

  void crno(int n)
  { rno = n; }

  };


class Rooms                             //Contains All Rooms
{
  int rno;                              //Room No.
  char rtype;                           //Room Type
  char rtyp[10];                        //Room Type
  char name[20];                        //Guest Name
  char status[10];                      //Room Status
  int rent;                             //Room Rent
  date datein;                          //Check-In Date
  date dateout;                         //Check-Out Date
  int stay;                             //Days of Stay
  public :
  void In();                            //To input Room Details

  void Out(int i)             //To display Room details' headers
  {

cout<<endl<<setw(3)<<i<<setw(13)<<rno<<setw(21)<<rtyp<<setw(15)<<rent
<<setw(22)<<status;
  }

  void ChangeN(char S[])                //To change Guest Name
  {
    strcpy(name, S);
  }

  void ChangeS(char S[])                //To change Room Status
  {
    strcpy(status, S);
```

```
  }

  Rooms()                 //To initialize Rooms Class Variables
  {
    strcpy(name, "");
  }

  int retrno()                          //To return Room No.
  { return rno; }

  char* retname()                       //To return Guest Name
  { return name; }

  char* retstatus()                     //To return Room Status
  { return status; }

  char retrtype()                       //To return Room Type
  { return rtype; }

  int retrent()
  { return rent; }
};



void Hotel::Charges(int c)
{
  getdate(&currdate);
  stay=1 + (int)currdate.da_day - (int)datein.da_day +
30*((int)currdate.da_mon - (int)datein.da_mon);

  if(rtype == 'D'|| rtype == 'd')
   charge = rent + mealbill + laundrybill;
```

```
  else
   charge = rent + mealbill + laundrybill;
  if(c == 1)
  {
    gotoxy(2,10);
    cout<<"Charges           : "<<"Rs. "<<charge<<endl;
  }
}


void Hotel::AMeal()
{
  meals++;
  if(rtype=='E')
   mealbill = meals*400;
  else
   mealbill = meals*250;
}

void Hotel::ALaundry()
{
  laundry++;
  if(rtype=='D')
   laundrybill = laundry*200;
  else
   laundrybill = laundry*100;
}

void Alter(int c, int n, char S[])     //Alters Guest Name\Room
Status
{
  fstream fil;
  fil.open("Rooms.dat", ios::binary | ios::out | ios::in);
  Rooms R;
  int pos;

  while(fil.read((char*)&R, sizeof(R)))
  {
    if(R.retrno() == n)
```

```
    {
      if(c==1)
       R.ChangeN(S);
      else if(c==2)
       R.ChangeS(S);
      pos = fil.tellg();
      fil.seekp(pos - sizeof(R));
      fil.write((char*)&R, sizeof(R));
    }
  }
  fil.close();
}

void Hotel::In(int ro, char rt)
{
  fstream fil;
  fil.open("Rooms.dat, ios::binary | ios::in);
  Rooms R;
  char ch;
  int c;
  rno=ro;
  rtype=rt;
  cout<<endl<<"Name            : ";
  gets(name);
  Alter(1,rno,name);
  do
  {
    c=0;
    cout<<"Age             : ";
    gets(age);
    for(int i=0; i<strlen(age); i++)
    {
      if((int)age[i]<48 || (int)age[i]>57)
       c++;
      if(strlen(age)>3 || strlen(age)>=3 && age[0] >= '2')
       c++;
    }
    if(c)
     cout<<"Invalid Age!"<<endl;
```

```
  }while(c);
  do
  {
    cout<<"Gender (M/F)    : ";
    gen=getche();
    cout<<endl;
  }while(gen!='M' && gen!='m' && gen!='F' && gen!='f');
  cout<<"Address         : ";
  gets(city);


  do
  {
    c=0;
    cout<<"Mobile No.      :";
    gets(phno);
    if(strlen(phno) != 10)
     c++;
    for(int i=0; i<strlen(phno); i++)
    {
      if((int)phno[i]<48 || (int)phno[i]>57)
       c++;
    }
    if(c)
    cout<<"Invalid Mobile No."<<endl;
  }while(c);

  while(fil.read((char*)&R, sizeof(R)))
   if(R.retrno == rno)
    rent = R.retrent();
  cout<<endl;
  getdate(&datein);
  fil.close();
}


void Hotel::Out(int c)
{
  cout<<endl<<endl;
```

```
  cout<<"Room Number       : "<<rno<<endl;
  cout<<"Room Type         : ";
  if(rtype=='e' || rtype=='E')
   cout<<"Economy"<<endl;
  else if(rtype=='d' || rtype=='D')
   cout<<"Deluxe"<<endl;
  cout<<endl<<"Name              : "<<name<<endl;
  cout<<"Gender            : "<<gen<<endl;
  cout<<"Age               : "<<age<<endl;
  cout<<"Address           : "<<city<<endl;
  cout<<"Mobile No.        : "<<phno<<endl;


  cout<<"Date of Check In  :
"<<(int)datein.da_day<<"."<<(int)datein.da_mon<<"."<<datein.da_year<<
endl;
  if(c==1)
  {
    getdate(&dateout);
    cout<<"Date of Check Out :
"<<(int)dateout.da_day<<"."<<(int)dateout.da_mon<<"."<<dateout.da_yea
r<<endl;
    Charges(0);
    cout<<"Days of Stay      : "<<stay<<endl;
    Charges(1);
    gotoxy(7, 15);
    cout<<endl<<"  B : Bill "<<endl;
  }
}


void Rooms::In()
{
  fstream fil;
  fil.open("Rooms.dat", ios::binary | ios::in);
  Rooms R;
  int c;

  do
```

```
  {
    c=0;
    cout<<endl<<endl<<"Room No.      : ";
    cin>>rno;
    while(fil.read((char*)&R, sizeof(R)))
    if(R.retrno() == rno)
     c++;
    if(c)
     cout<<"Room Already Present!"<<endl;
  }while(c);

  do
  {
    cout<<"Room Type(D/E): ";
    rtype=getche();
    cout<<endl;
    if(rtype != 'd' && rtype != 'D' && rtype != 'e' && rtype != 'E')
     cout<<"Invalid Input!";
  }while(rtype != 'd' && rtype != 'D' && rtype != 'e' && rtype !=
'E');

  if(rtype == 'd' || rtype == 'D')
   strcpy(rtyp, "Deluxe");
  else if(rtype == 'e' || rtype == 'E')
   strcpy(rtyp, "Economy");

  cout<<"Rent          : ";
  cin>>rent;
  strcpy(status, "Vacant");
  fil.close();
}


void M()                                          //Displays EXIT
option
{
  textcolor(8);
  gotoxy(35,25);
  cout<<"Esc to EXIT";
```

```
  textcolor(15);
}


void Transfer()                                  //Transfers file
content
{
  fstream fil1, fil2;
  fil1.open("temp.dat", ios::binary | ios::in);
  fil2.open("Hotel.dat", ios::binary | ios::out);
  Hotel H;
  while(fil1.read((char*)&H, sizeof(H)))
  fil2.write((char*)&H, sizeof(H));
  fil1.close();
  fil2.close();
}

void CurrCharges()                              //Calculates Current
Bill
{
  fstream fil;
  fil.open("Hotel.dat", ios::binary | ios::in);
  Hotel H;
  int n;
  gotoxy(2,4);
  cout<<"Room No. : ";
  cin>>n;
  while(fil.read((char*)&H, sizeof(H)))
  if(H.retrno() == n)
  {
    H.Charges(1);
    H.Bill();
  }
  fil.close();
}


void RTransfer()                                 //Transfers File
Content
```

```
{
  fstream fil1, fil2;
  fil1.open("temp.dat", ios::binary | ios::in);
  fil2.open("Rooms.dat", ios::binary | ios::out);
  Rooms R;
  while(fil1.read((char*)&R, sizeof(R)))
  fil2.write((char*)&R, sizeof(R));
  fil1.close();
  fil2.close();
}

void RAdd()                                          //Adds a Room
{
  fstream fil;
  fil.open("Rooms.dat", ios::binary | ios::app);
  Rooms R;
  char ch;
  cout<<"\n     Enter Room Details :- \n";
  do
  {
    clrscr();
    R.In();
    fil.write((char*)&R, sizeof(R));
    clrscr();
    gotoxy(32,12);
    cout<<"Room Added!";
    gotoxy(32,14);
    cout<<"Add More Rooms? (Y/N) \n";
    ch=getch();
    cout<<endl;
  }while(ch=='Y' || ch=='y');
  fil.close();
}


void RDelete()                                       //Deletes Room
{
  fstream fil1, fil2;
  int del=0;
```

```
fil1.open("Rooms.dat", ios::binary | ios::in);
fil2.open("temp.dat", ios::binary | ios::out);
Rooms R;
int rno;
gotoxy(32,12);
cout<<"Enter Room No. : ";
cin>>rno;


while(fil1.read((char*)&R, sizeof(R)))
{
  if(R.retrno() != rno)
   fil2.write((char*)&R, sizeof(R));
  else
  {
    del++;
    gotoxy(35,14);
    cout<<"Room Deleted!";
  }
}
fil1.close();
fil2.close();
if(!del)
{
  clrscr();
  gotoxy(32,12);
  cout<<"Room not present!"<<endl;
}
else
  RTransfer();


del=0;
fil1.open("Hotel.dat", ios::binary | ios::in);
fil2.open("temp.dat", ios::binary | ios::out);
Hotel H;
while(fil1.read((char*)&H, sizeof(H)))
{
  if(H.retrno()!=rno)
```

```
    fil2.write((char*)&H, sizeof(H));
    else
    {
      del++;
    }
  }
  fil1.close();
  fil2.close();
  if(del)
   Transfer();
}

void RDisplay(int c)                                    //Displays All
Rooms
{
  int i=1;
  fstream fil;
  fil.open("Rooms.dat", ios::binary | ios::in);
  Rooms R;
  cout<<setw(5)<<"S.No."<<setw(15)<<"Room No."<<setw(18)<<"Room
Type"<<setw(14)<<"Rent"<<setw(22)<<"Status"<<endl;
  while(fil.read((char*)&R, sizeof(R)))
  {
    if(c==2 && strcmp(R.retstatus(),"Vacant")==0 || c==1)
    {
      if(!(i%9))
      {
        gotoxy(28,22);
        cout<<"Press any key to view next ";
        getch();
        clrscr();
        cout<<setw(5)<<"S.No."<<setw(15)<<"Room No."<<setw(18)<<"Room
Type"<<setw(14)<<"Rent"<<setw(22)<<"Status"<<endl;
      }
      R.Out(i++);
      cout<<endl;
    }
  }
  fil.close();
```

```
}

void Create()                                                    //Books a
Room
{
  fstream fil1,fil2;
  fil1.open("Hotel.dat", ios::binary | ios::app);
  Hotel H;
  fil2.open("Rooms.dat", ios::binary | ios::in);
  Rooms R;
  char ch,c,rt;
  int rno,check=0;

cout<<endl<<endl<<"                                  ---Available
Rooms---"<<endl<<endl;
  RDisplay(2);
  gotoxy(31,18);
  cout<<"Press Enter to proceed ";
  M();
  do
  { c=getch(); }while(c != 13 && c != 27);
  clrscr();
  if(c==13)
  {
    gotoxy(32,12);
    cout<<"Room No : ";
    cin>>rno;
    clrscr();
    gotoxy(2,2);
    while(fil2.read((char*)&R, sizeof(R)))
    if(R.retrno() == rno && strcmpi(R.retstatus(),"Vacant") == 0)
    {
      rt=R.retrtype();
      check++;
    }
    fil2.seekg(0);
    if(check)
    {
      cout<<"\n     Enter Guest Details :- \n";
```

```
      H.In(rno,rt);
      clrscr();
      gotoxy(34,12);
      cout<<"Room Booked! ";
      gotoxy(30,15);
      cout<<"Press Enter to Continue! "<<endl;
      do
      { c=getch(); }while(c != 13);

      fil1.write((char*)&H , sizeof(H));
      fil2.close();
      Alter(2, rno, "Occupied");
      cout<<endl;
    }
    else
    {
      gotoxy(30,12);
      cout<<"Room Not Available"<<endl;
      M();
      getch();
    }
  }
  fil1.close();
}


void Display()                                  //Displays Occupied
Rooms
{
  int i=1;
  fstream fil;
  fil.open("Hotel.dat", ios::binary | ios::in);
  Hotel H;
  cout<<setw(5)<<"S.No."<<setw(15)<<"Room No."<<setw(22)<<"Guest
Name"<<setw(18)<<"Check In"<<endl;
  while(fil.read((char*)&H, sizeof(H)))
  {
    if(!(i%9))
    {
```

```
      gotoxy(28,22);
      cout<<"Press any key to view next ";
      getch();
      clrscr();
      cout<<setw(5)<<"S.No."<<setw(15)<<"Room No."<<setw(22)<<"Guest
Name"<<setw(18)<<"Check In"<<endl;
    }
    H.Out2(i++);
    cout<<endl;
  }
  fil.close();
}


void Searchname()                          //Searches Room By Guest
Name
{
  char name[20];
  int found = 0, check = 0;
  char c;
  gotoxy(34,12);
  cout<<"Guest Name : ";
  gets(name);
  clrscr();

  fstream fil1, fil2;
  fil1.open("Hotel.dat", ios::binary | ios::in);
  fil2.open("Rooms.dat", ios::binary | ios::in);
  Rooms R;
  Hotel H;

  gotoxy(2,2);
  cout<<setw(5)<<"S.No."<<setw(15)<<"Room No."<<setw(18)<<"Room
Type"<<setw(14)<<"Rent"<<setw(22)<<"Status"<<endl;
  while(fil2.read((char*)&R, sizeof(R)))
  if(strcmpi(R.retname(), name)==0)
  {
    R.Out(1);
    check++;
```

```
  }

  if(check)
  {
    gotoxy(28,18);
    cout<<"Press Enter for Details!";
      c=getch();
    if(c == 13)
    {
      while(fil1.read((char*)&H, sizeof(H)) && found==0)
      {
        if(strcmpi(H.retname(), name)==0)
        {
          gotoxy(28,18);
          cout<<"                                                    ";
          found++;
          gotoxy(2,4);
          H.Out();
        }
      }
    }
  }
  else
  {
    clrscr();
    gotoxy(33,14);
    cout<<"Room Not Present!"<<endl;
  }


  fil1.close();
  fil2.close();
}


void Searchroom()                          //Searches Room By Room
No.
{
  fstream fil1, fil2;
```

```
  fil1.open("Hotel.dat", ios::binary | ios::in);
  fil2.open("Rooms.dat", ios::binary | ios::in);
  int A, found=0, check=0;
  Hotel H;
  Rooms R;
  int n;
  char c;

  gotoxy(34,12);
  cout<<"Room No. : ";
  cin>>n;
  clrscr();

  cout<<setw(5)<<"S.No."<<setw(15)<<"Room No."<<setw(18)<<"Room
Type"<<setw(14)<<"Rent"<<setw(22)<<"Status"<<endl;
  while(fil2.read((char*)&R, sizeof(R)))
  if(R.retrno() == n)
  {
    R.Out(1);
    check++;
  }

  if(check)
  {
    gotoxy(28,18);
    cout<<"Press Enter for Details!";
      c=getch();
    if(c == 13)
    {
      while(fil1.read((char*)&H, sizeof(H)) && found==0)
      {
        if(H.retrno()==n)
        {
          gotoxy(28,18);
          cout<<"                                                  ";
          found++;
          gotoxy(2,4);
          H.Out();
        }
```

```
      }
    }
    if(!found)
    {
      clrscr();
      gotoxy(34,18);
      cout<<"Room Vacant!";
    }
  }
  else
  {
    gotoxy(33,14);
    cout<<"Room Not Present!"<<endl;
  }


  fil1.close();
  fil2.close();
}


void Sortname()                                //Sorts Files By Guest
Name
{
  fstream fil;
  fil.open("Hotel.dat", ios::binary | ios::in | ios::out);
  Hotel HJ,HJP1;
  fil.seekg(0, ios::end);
  int NOR = fil.tellg()/sizeof(HJ);
  for(int I=0; I<NOR-1; I++)
  {
    for(int J=0; J<NOR-I-1; J++)
    {
      fil.seekg(J*sizeof(HJ));
      fil.read((char*)&HJ, sizeof(HJ));
      fil.read((char*)&HJP1, sizeof(HJP1));
      if(strcmpi(HJ.retname(), HJP1.retname())>0)
      {
        fil.seekp(J*sizeof(HJ));
```

```
        fil.write((char*)&HJP1, sizeof(HJP1));
        fil.write((char*)&HJ, sizeof(HJ));
      }
    }
  }
  fil.close();
  gotoxy(35,12);
  cout<<"Records Sorted!";
}


void Sortroom()                                  //Sorts Files By Room
No.
{
  fstream fil;
  fil.open("Rooms.dat", ios::binary | ios::in | ios::out);
  Rooms HJ,HJP1;
  fil.seekg(0, ios::end);
  int NOR = fil.tellg()/sizeof(HJ);
  for(int I=0; I<NOR-1; I++)
  {
    for(int J=0; J<NOR-I-1; J++)
    {
      fil.seekg(J*sizeof(HJ));
      fil.read((char*)&HJ, sizeof(HJ));
      fil.read((char*)&HJP1, sizeof(HJP1));
      if(HJ.retrno() > HJP1.retrno())
      {
        fil.seekp(J*sizeof(HJ));
        fil.write((char*)&HJP1, sizeof(HJP1));
        fil.write((char*)&HJ, sizeof(HJ));
      }
    }
  }
  fil.close();
  gotoxy(35,12);
  cout<<"Records Sorted!";
}
```

```
char Delete1()                                          //Checks Out
{
  fstream fil1, fil2, fil3;
  int del=0, check=0;
  fil1.open("Hotel.dat", ios::binary | ios::in);
  fil2.open("temp.dat", ios::binary | ios::out);
  fil3.open("Rooms.dat", ios::binary | ios::in);
  Hotel H;
  Rooms R;
  int rno;
  char c;
  gotoxy(33,12);
  cout<<"Enter Room No. : ";
  cin>>rno;
  clrscr();

  while(fil3.read((char*)&R, sizeof(R)))
   if(R.retrno() == rno )
    check++;
  fil3.close();
  if(check)
  {
    while(fil1.read((char*)&H, sizeof(H)))
    {
      if(H.retrno()!=rno)
       fil2.write((char*)&H, sizeof(H));
      else
      {
        del++;
        H.Out(1);
        M();
        gotoxy(35,18);
        c=getch();
        clrscr();
        if(c=='B' || c=='b')
        H.Bill();
        Alter(2, rno, "Vacant");
      }
```

```
    }
  }
  else
  {
    gotoxy(33,14);
    cout<<"Room Not Present! ";
  }
  fil1.close();
  fil2.close();
  if(!del && check)
  {
    gotoxy(33,14);
    cout<<"Room Already Vacant!"<<endl;
  }
  else
    Transfer();
  return(c);
}


void Add(int c)                                    //Adds Services
{
  fstream fil;
  fil.open("Hotel.dat", ios::binary | ios::in | ios::out);
  Hotel H;
  int rno,n,pos,f=0;

  gotoxy(2,4);
  cout<<"Enter Room No. : ";
  cin>>rno;
  if((rno>=100 && rno<=120 || rno>=200 && rno<=250))
  {
    while(fil.read((char*)&H, sizeof(H)))
    {
      pos=fil.tellg();
      if(H.retrno()==rno)
      {
        fil.seekp(pos - sizeof(H));
        gotoxy(35,12);
```

```
        if(c==1)
        {
          H.AMeal();
          cout<<"Meal Added! ";
        }
        else if(c==2)
        {
          H.ALaundry();
          cout<<"Laundry Sent! ";
        }
        fil.write((char*)&H, sizeof(H));
        f++;
      }
    }
  }
  else
    cout<<"Invalid Room! ";
  if(!f && (rno>=100 && rno<=120 || rno>=200 && rno<=250))
   cout<<"Room vacant! "<<endl;
}



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



const char VL=179,HL=196,TR=191,BL=192,BR=217,TL=218;


void DispXY(int X,int Y,char CH)                //Displays a
Character
{
  gotoxy(X,Y);//conio.h
  cprintf("%c",CH);
}
```

```
void DispXY(int X,int Y,char CH[])                  //Displays a
String
{
  gotoxy(X,Y);//conio.h
  cprintf("%s",CH);
}

void COLOR(int TC,int BC)                           //Changes Colors
{
  textcolor(TC);
  textbackground(BC);
}

void BOX(int C1,int R1,int C2,int R2,int c=0)       //Draws A BOX
{
  DispXY(C1,R1,'É');
  DispXY(C2,R1,'»');
  DispXY(C1,R2,'È');
  DispXY(C2,R2,'¼');
  for (int I=C1+1;I<C2;I++)
  {
    DispXY(I,R1,'Í');
    DispXY(I,R2,'Í');
    if(c==1)
     delay(3);
  }
  for (I=R1+1;I<R2;I++)
  {
    DispXY(C1,I,'º');
    DispXY(C2,I,'º');
    if(c==1)
     delay(10);
  }
}

int VMENU(int C,int R, char M[][20],int NOO)        //Vertical Menu
{
  for (int I=0;I<NOO;I++)
```

```
    DispXY(C,R+I+2,M[I]);
  int Sel=0, Exit=0;
  char CH;
  do
  {
    textbackground(9);
    DispXY(C,R+Sel+2,M[Sel]);
    textbackground(0);
    int TSel=Sel;
    CH=getch();
    switch(CH)
    {
      case 71 : Sel=0; break;        //Home
      case 79 : Sel=NOO-1; break;    //End
      case 72 : if(Sel==0)           //Up
                 Sel=NOO-1;
                else
                 Sel--;
                break;
      case 80 : if(Sel==NOO-1)       //Dn
                 Sel=0;
                else
                 Sel++;
                break;
      case 27 : Sel=-1;              //Escape
      case 13 : Exit++;              //Enter
    }
    DispXY(C,R+TSel+2,M[TSel]);
  }while (Exit==0);
  return Sel;
}

int HMENU(int C,int R, char M[][20],int NOO)       //Horizontal Menu
{
  for (int I=0;I<NOO;I++)
    DispXY(C+14*I+2,R,M[I]);
  int Sel=0, Exit=0;
  char CH;
  do
```

```
  {
    textbackground(9);
    DispXY(C+14*Sel+2,R,M[Sel]);
    textbackground(0);
    int TSel=Sel;
    CH=getch();
    switch(CH)
    {
      case 71 : Sel=0;break;        //Home
      case 79 : Sel=NOO-1;break;    //End
      case 75 : if(Sel==0)          //Lt
                 Sel=NOO-1;
                else
                 Sel--;
                break;
      case 77 : if(Sel==NOO-1)      //Rt
                 Sel=0;
                else
                 Sel++;
                break;
      case 27 : Sel=-1;             //Escape
      case 13 : Exit++;             //Enter
    }
    if(!Exit)
     DispXY(C+14*TSel+2,R,M[TSel]);
  }while (Exit==0);
  return Sel;
}

char* getpass1(char s[], int R, int C)              //Inputs Password
{
  char p[25],c;
  int i=0;
  M();
  gotoxy(R,C);
  cout<<s;
  do
  {
    c=getch();
```

```
    if(c!=27)
    {
      cout<<"*";
      p[i]=c;
      i++;
    }
  }while(c!=13 && c!=27 && i<4);
  p[i]='\0';
  return(p);
}



void ChngPass()                                    //Changes Password
{
  fstream fil, filt;
  char P[25], PCheck[25], PNew1[25], PNew2[25];
  int c=0;


  fil.open("Password.txt", ios::in);
  fil.getline(P,25);
  fil.close();
  strcpy(PCheck, getpass1("Old Password         : ", 25, 13));
  if(strcmp(PCheck, P)==0)
  {
    do
    {
      strcpy(PNew1, getpass1("New Password        : ", 25, 15));
      strcpy(PNew2, getpass1("Verify New Password : ", 25, 17));
      clrscr();
      gotoxy(35,15);
      if(strcmp(PNew1,PNew2)==0)
      {
        cout<<"Password Changed";
        fil.open("Password.txt", ios::out);
        fil<<PNew2<<endl;
        c++;
      }
```

```
      else
      {
        gotoxy(33,13);
        cout<<"Passwords do not match! ";
        gotoxy(33,15);
        cout<<"Press any key to enter again";
        getch();
        clrscr();
      }
    }while(c==0);
  }
  else
  {
    gotoxy(35,18);
    cout<<"Wrong Password!";
  }
  fil.close();
}


void Welcome(int c=0)                                  //Welcomes User
{
  textcolor(1);
  if(c==1)
   BOX(1,1,79,24,1);
  else
    BOX(1,1,79,24,0);
  textcolor(15);


  char W[][100]={" ÉÍÍ»  ÉÍÍ»
",
    " º  º  º  º
",
    " º  º  º  º ÉÍÍÍÍÍ» ÉÍÍÍÍÍ» ÉÍÍÍÍÍÍ»  ÉÍ»
",
    " º  ÈÍÍ¼  º º ÉÍ» º ÈÍ» ÉÍ¼ º ÉÍÍÍÍ¼  º º
",
    " º  ÉÍÍ»  º º º º º   º º   º ÈÍÍ»    º º
```

```
",
    " º  º  º  º º º º º   º º   º ÉÍÍ¼    º º
",
    " º  º  º  º º ÈÍ¼ º   º º   º ÈÍÍÍÍ»  º ÈÍÍÍ»
",
    " ÈÍÍ¼  ÈÍÍ¼ ÈÍÍÍÍÍ¼   ÈÍ¼   ÈÍÍÍÍÍÍ¼  ÈÍÍÍÍÍ¼
",
    "
",
    "
",
    " ÉÍÍÍÍÍÍÍÍÍ»
",
    " º  ÉÍÍÍ»  º
",
    " º  º   º  º ÉÍÍÍÍÍ» ÉÍÍÍÍÍ» ÉÍÍÍÍÍ» ºÍÍÍÍÍ» ÉÍÍ» ÉÍÍÍÍÍ»
ÉÍÍÍÍÍ»  ",
    " º  ÈÍÍÍ¼  º º ÉÍ» º º ÉÍ» º º ÉÍ» º º ÉÍ» º º  º º ÉÍÍÍ¼ º
ÉÍÍÍÍ¼  ",
    " º  ÉÍÍÍÍÍÍ¼ º º º º º ÈÍ¼ º º º º º º º º º º  º º ÈÍÍÍ» º ÈÍÍ»
",
    " º  º        º ÈÍ¼ º º ÍÍÍÍ¼ º ÈÍ¼ º º º º º º  º ÈÍÍÍ» º º ÉÍÍ¼
",
    " º  º        º ÉÍ» º º ÉÍ» º º ÉÍ» º º ÈÍ¼ º º  º ÉÍÍÍ¼ º º
ÈÍÍÍÍ»  ",
    " ÈÍÍ¼        ÈÍ¼ ÈÍ¼ ÈÍ¼ ÈÍ¼ ÈÍ¼ ÈÍ¼ ºÍÍÍÍÍ¼ ÈÍÍ¼ ÈÍÍÍÍÍ¼
ÈÍÍÍÍÍÍ¼  "
  };


  char A[]="[ Hotel Management System ]";
  gotoxy(28,1);
  textcolor(14);
  for(int i=0; i<strlen(A); i++)
  {
    cprintf("%c", A[i]);
    if(c==1)
     delay(25);
  }
```

```
  if(c==1)
   delay(300);
  textcolor(15);
  for(i=0; i<18; i++)
  {
    gotoxy(2,2+i);
    cprintf("%s", W[i]);
    if(c==1)
     delay(10);
  }
  if(c==1)
   delay(1000);
  textcolor(15);
}

void Thanks()                                          //Thanks User
{
  clrscr();
  textmode(C40);
  gotoxy(9,13);
  cprintf("Thank You for Visiting!");
    textcolor(15);
  getch();
}


void main()                                            //Main Function
{
  clrscr();
  Welcome(1);
  //Menu Headers
  char CMenu[][20] = {" Check In ", " Check Out ", " Current Bill ",
"
Services ", " Main Menu "};
  char CSMenu[][20] = {" Meal "," Laundry "};
  char MMenu[][20] = {" Manage ", " Sort ", " Search ", " Modify ", "
Change Password "};
```

```
  char MMenu1[][20] = {" Add Room ", " All Rooms ", " Occupied Rooms
", " Delete Room "};
  char MMenu2[][20] = {" By Name ", " By Room "};
  char MMenu3[][20] = {" By Name ", " By Room "};
  char MMenu4[][20] = {" Current Bill ", " Add Meal ", " Add Laundry
"};
  char Menu[][20] = {" Manager ", " Guest "};
  char c=' ';
  char Pass[25], P[25];
  int MOption, Option, SOption,e;


  fstream fil;
  //fil.open("Password.txt", ios::out); fil<<"pass"; fil.close();
  fil.open("Password.txt", ios::in);
  fil.getline(P,25);  fil.close();

  do
  {
    Welcome();
    e=0;
    M();
    BOX(28,21,40,23);
    BOX(41,21,53,23);
    MOption=HMENU(28,22,Menu,2);

    if(MOption == 1)                              //Guest Mode
    {
      clrscr();
      do
      {
        M();
        BOX(1,1,79,24);
        Option=HMENU(2,2,CMenu,5);
        switch(Option)
        {
          case 0 : clrscr(); Create();
                   break;
          case 1 : clrscr(); c = Delete1();
```

```
                while(c!=27)
                c=getch();
                break;
      case 2 : clrscr(); CurrCharges(); M();
                do
                { c=getch(); }while(c!=27);
                break;
      case 3 : BOX(43,3,55,6); SOption=VMENU(45,2,CSMenu,2);
                switch(SOption)
                {
                  case 0 : clrscr(); Add(1); M();
                           do
                           { c=getch(); }while(c!=27);
                           break;
                  case 1 : clrscr(); Add(2); M();
                           do
                           { c=getch(); }while(c!=27);
                }
                break;
      case 4 : Option = -1;
    }
    clrscr();
  }while (Option != -1);
}
else if(MOption == 0)                          //Manager Mode
{
  clrscr();
  BOX(29, 13, 47, 15);
  gotoxy(30, 14);
  strcpy(Pass,getpass1(" Password : ", 30, 14));
  if(strcmp(Pass,"")==0)
  {
    e=-1;
    clrscr();
  }

  fil.open("Password.txt", ios::in);
  fil.getline(P,25);  fil.close();
```

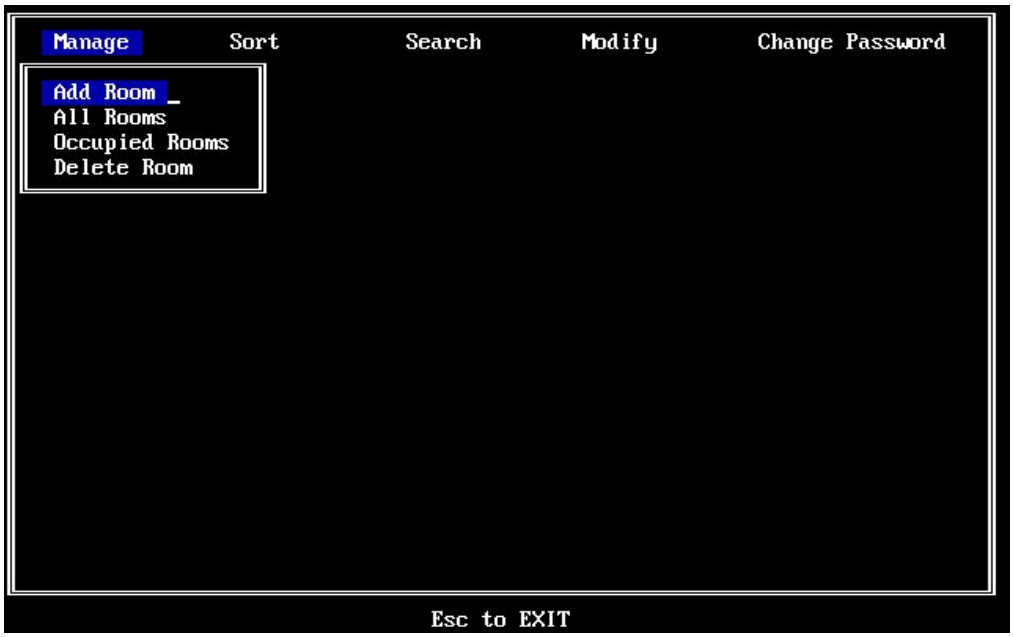
```
if(strcmp(P,Pass)==0)
{
  clrscr();
  do
  {
    M();
    BOX(1,1,79,24);
    Option=HMENU(2,2,MMenu,5);
    switch(Option)
    {
      case 0 : BOX(2,3,21,8);
               SOption=VMENU(4,2,MMenu1,4);
               switch(SOption)
               {
                 case 0 : clrscr(); RAdd();
                          break;
                 case 1 : clrscr(); RDisplay(1); M();
                          do
                          { c=getch(); }while(c!=27);
                          break;
                 case 2 : clrscr(); Display(); M();
                          do
                          { c=getch(); }while(c!=27);
                          break;
                 case 3 : clrscr(); RDelete(); M();
                          do
                          { c=getch(); }while(c!=27);
               }
               break;
      case 1 : BOX(17,3,29,6);
               SOption=VMENU(19,2,MMenu2,2);
               switch(SOption)
               {
                 case 0 : clrscr(); Sortname(); M();
                          do
                          { c=getch(); }while(c!=27);
                          break;
                 case 1 : clrscr(); Sortroom(); M();
                          do
```

```
                    { c=getch(); }while(c!=27);
                    break;
          }
          break;
  case 2 : BOX(30,3,42,6);
          SOption=VMENU(32,2,MMenu3,2);
          switch(SOption)
          {
            case 0 : clrscr(); Searchname(); M();
                    do
                    { c=getch(); }while(c!=27);
                    break;
            case 1 : clrscr(); Searchroom(); M();
                    do
                    { c=getch(); }while(c!=27);
          }
          break;
  case 3 : BOX(42,3,58,7);
          SOption=VMENU(44,2,MMenu4,3);
          switch(SOption)
          {
            case 0 : clrscr(); CurrCharges(); M();
                    do
                    { c=getch(); }while(c!=27);
                    break;
            case 1 : clrscr(); Add(1); M();
                    do
                    { c=getch(); }while(c!=27);
                    break;
            case 2 : clrscr(); Add(2); M();
                    do
                    { c=getch(); }while(c!=27);
                    break;
          }
          break;
  case 4 : clrscr(); ChngPass(); M();
          do
          { c=getch(); }while(c!=27);
}
```

```
        clrscr();
      }while (Option!=-1);
    }
    else
    {
      gotoxy(27,18);
      if(e!=-1)
      {
        cout<<"Incorrect Password!";
        M();
      }
      else
      {
        gotoxy(30,19);
        cout<<"Press Esc once again!";
      }
      do
      { c=getch(); }while(c!=27);
      clrscr();
    }
  }
}while(MOption!=-1);
Thanks(); }
```

Output Screen


Password : ***_
Esc to EXIT


Manage
Sort
Search
Modify
Change Password
Add Room
All Rooms
Occupied Rooms
Delete Room
Esc to EXIT

```
Room No.    : 129
Room Type(D/E): D
Rent        : 5000
```

S.No.	Room No.	Room Type	Rent	Status
1	100	Deluxe	1000	Vacant
2	200	Economy	500	Vacant
3	150	Deluxe	2000	Vacant
4	125	Deluxe	5000	Vacant
5	129	Deluxe	5000	Vacant

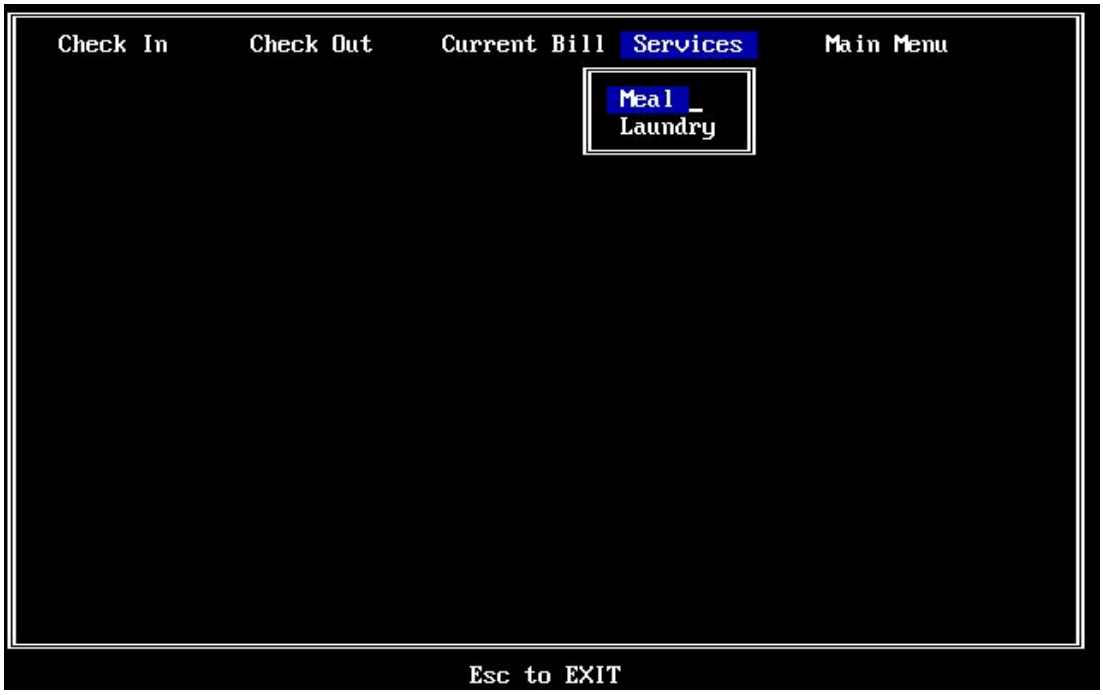
Esc to EXIT_

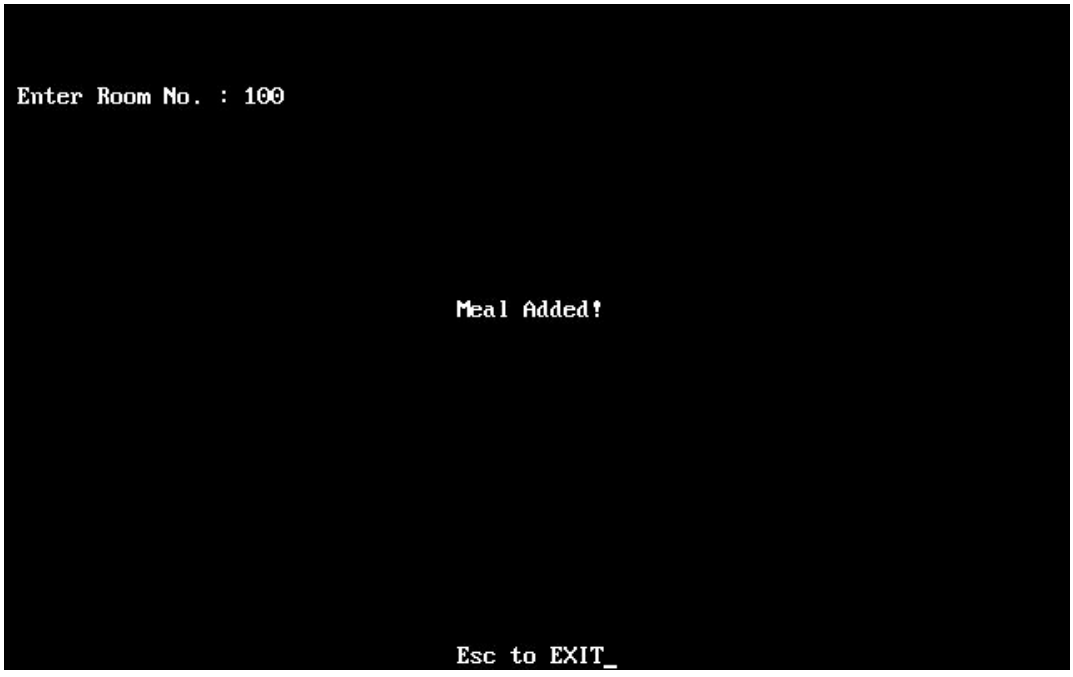

---Available Rooms---
S.No. Room No. Room Type Rent Status
1 100 Deluxe 1000 Vacant
2 200 Economy 500 Vacant
3 150 Deluxe 2000 Vacant
4 125 Deluxe 5000 Vacant
5 129 Deluxe 5000 Vacant
Press Enter to proceed
Esc to EXIT_


Check In
Check Out
Current Bill
Services
Main Menu
Esc to EXIT

```
    Enter Guest Details :-

Name            : Sam
Age             : 10
Gender (M/F)    : m
Address         : Delhi
Mobile No.      :9876543210
```

```
                    Room Booked!

                Press Enter to Continue!
```


Check In
Check Out
Current Bill
Services
Main Menu
Meal
Laundry
Esc to EXIT

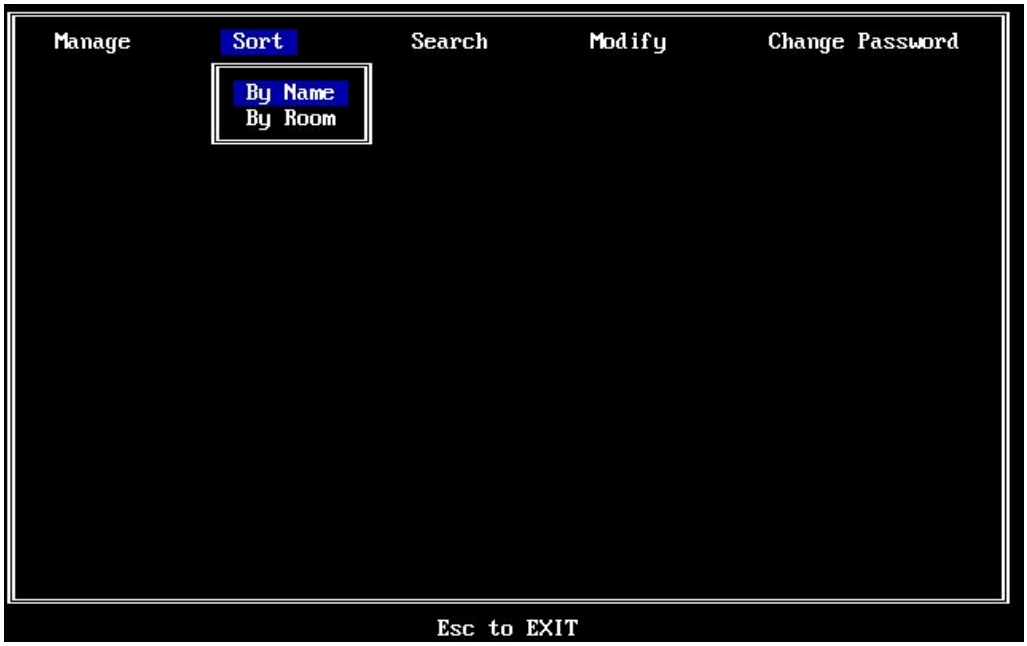

Enter Room No. : 100
Meal Added!
Esc to EXIT

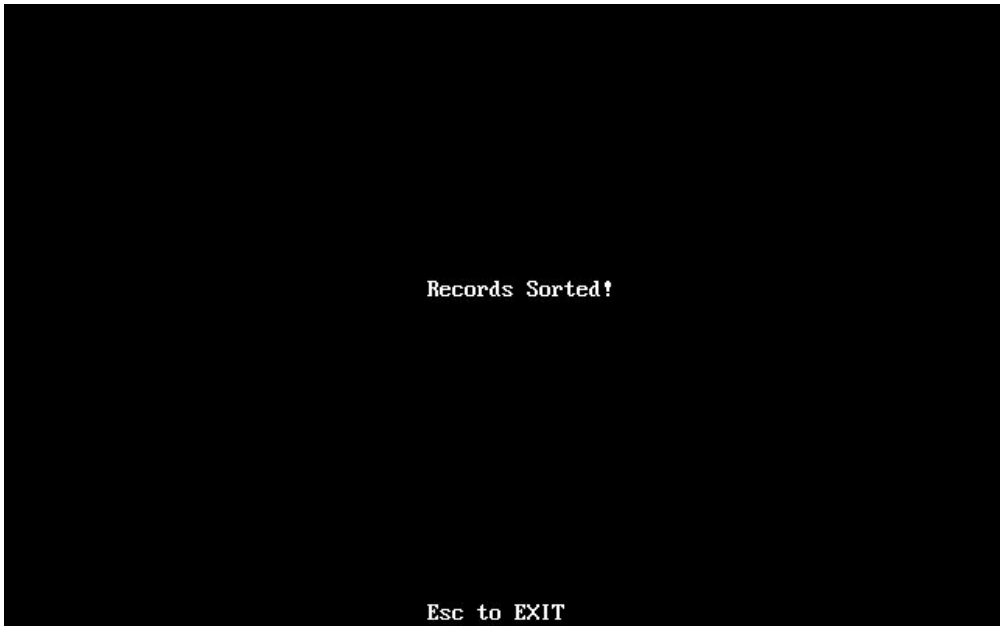
```
Enter Room No. : 100

                              Laundry Sent!

                              Esc to EXIT_
```

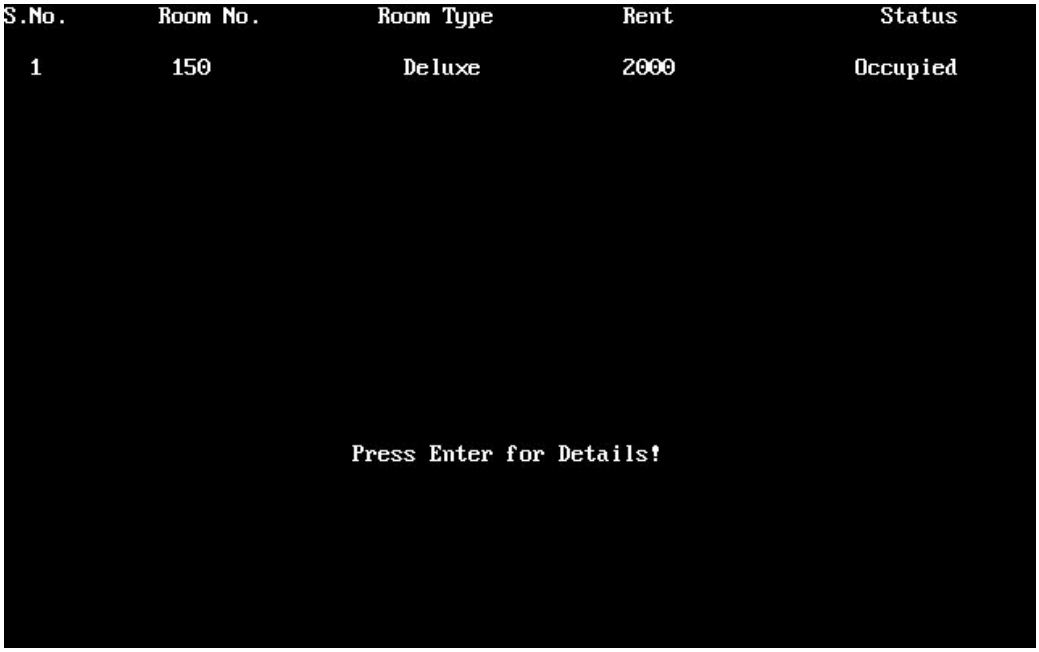
```
Room No. : 100

                                 Bill :-
Charges           : Rs. 3350
                                 Rent            : 3000
                                 Meals           : 250
                                 Laundry         : 100

                              Esc to EXIT
```

```
Room Number       : 125
Room Type         : Deluxe

Name              : Guest
Gender            : f
Age               : 20
Address           : Bombay
Date of Check In  : 9.12.2016
Date of Check Out : 9.12.2016
Days of Stay      : 1
 Charges           : Rs. 3000


  B : Bill




                              Esc to EXIT
```

S.No.	Room No.	Guest Name	Check In
1	150	Sam	9.12.2016
2	201	Sammy	9.12.2016


Manage
Sort
Search
Modify
Change Password
By Name
By Room
Esc to EXIT

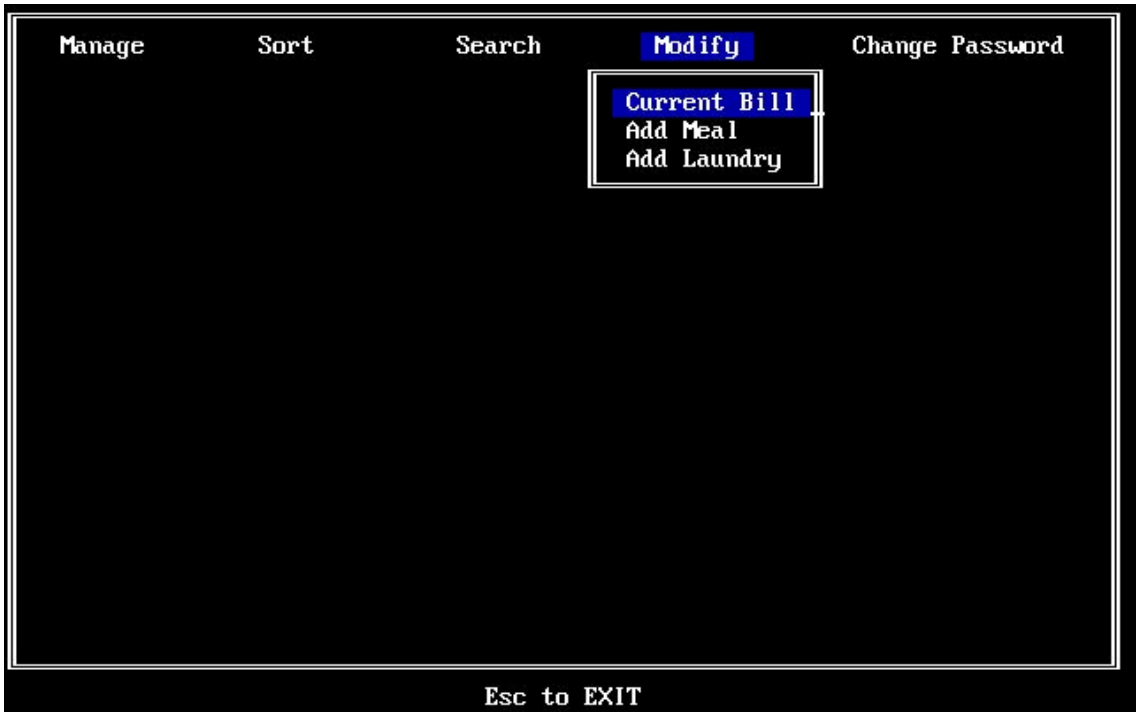

Records Sorted!
Esc to EXIT

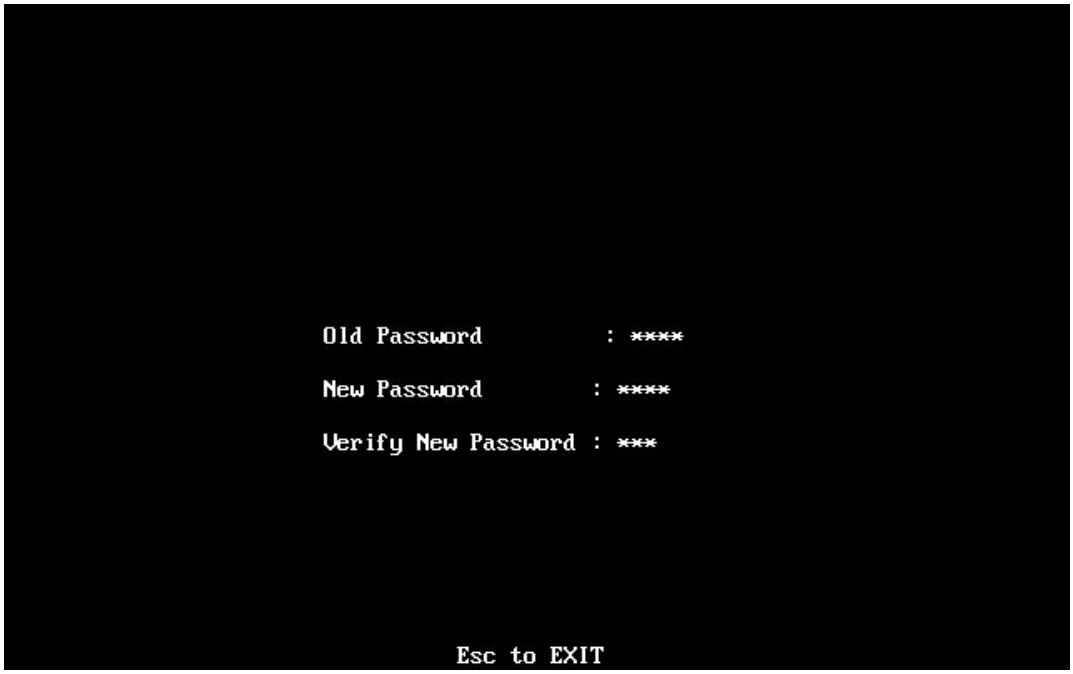

S.No.
Room No.
Room Type
Rent
Status
1
150
Deluxe
2000
Occupied
Press Enter for Details!

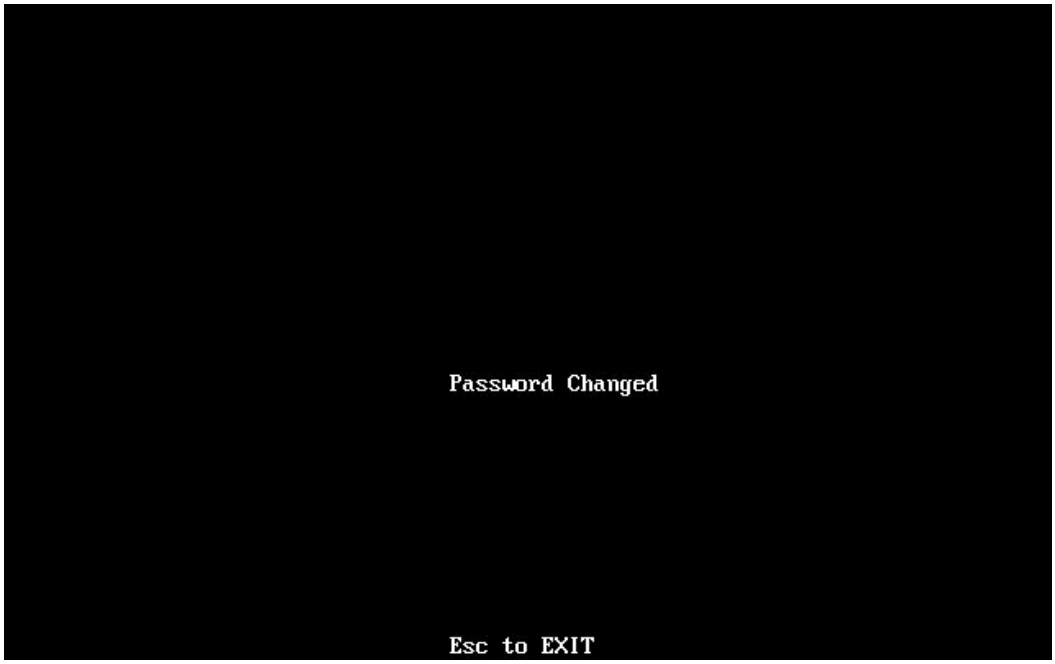

Room Number : 200
Room Type : Economy
Name : sanket
Gender : m
Age : 12
Address : jhgfd
Date of Check In : 9.12.2016
Esc to EXIT_

```
    Enter Guest Details :-

Name            : sammy'
Age             : 12345
Invalid Age!
Age             : 1234
Invalid Age!
Age             : 120
Gender (M/F)    : m
Address         : Bombay
Mobile No.      :9999888877776666
Invalid Mobile No.
Mobile No.      :987654321234
Invalid Mobile No.
Mobile No.      :9876543210_
```

```
 S.No.      Room No.        Room Type        Rent              Status

  1         200              Deluxe          2000             Occupied

Room Number       : 200
Room Type         : Deluxe

Name              : Sam
Gender            : m
Age               : 10
Address           : Delhi
Mobile No.        : 9876543210
Date of Check In  : 11.12.2016

                                Esc to EXIT
```


Manage
Sort
Search
Modify
Change Password
Current Bill
Add Meal
Add Laundry
Esc to EXIT


Old Password : ****
New Password : ****
Verify New Password : ***
Esc to EXIT


Password Changed
Esc to EXIT


Thank You for Visiting!_

Hardware & Software Requirement

Hardware Requirement

Intel core/i3/i5/i7 or any equivalent

With at least 256 MB RAM

2 MB free space on Hard Disk

Color Monitor/LCD

Operating System & Compiler

MS Windows Turbo C++ 3.1 Compiler