INDIAN SCHOOL SOHAR

YEAR: 2018-2019

**INFORMATICS PRACTICES PROJECT**

**Hotel Management**

**System**



Done by

Name: KRITHIKHA BALAMURUGAN

INTRODUCTION

HOTEL PHOENIX is a 5 star hotel. This application is designed to serve the purpose as a hotel management system.This application :

* Is user-friendly and takes the minimal work of the user from the front-end in the fastest way possible.
* The functionality is in favour of the user and it does not concern the user with the back end work.The user’s work is saved continually
* Its design is aesthetic and decent in view.Java tables are displayed in the forms for a quick review by the user.
* Shortcut keys are provided for buttons using the mnemonic property.
* Tool tips are displayed for users to help them in the front end interface.
* JCalender is used in the application.
* All the panels are linked to the mains form through ‘CARD LAYOUT’.The **CardLayout** class manages the components in such a way that only one component is visible at a time. It treats each component as a **card** in the container. Only one **card** is visible at a time, and the container acts as a stack of **cards**.

Hotel Phoenix has 3 types of rooms –

* Single
* Double
* Deluxe

It has the well provided following services –

* Swimming Pool
* Gym
* Laundry

The software used by HOTEL PHOENIX has –

LOGIN –

The frame consists of authorization and authentication of specified users who can carry out the process of booking.

MAIN FRAME –

It consists of a main panel that contains various buttons which directs to various panels through ‘card layout’.

ROOM INFORMATION & CHECK-IN –

The panel has details about the status and other details of various rooms which serve as initial stand by in the process of booking. This panel serves as a means for the selection of room number. User can filter the rooms as per the choice of the customer.

The customer details are entered and saved through this panel as a part of checking in.

UPDATE –

This provides the privilege to add and delete records from the room information table using buttons like “add” and “delete” and provides necessary details through a java code.

CHECKOUT –

This frame serves as the terminating step in the booking process.Here the final bill is calculated.

On entering the room number, all the information regarding that particular room and the customer information will be displayed including the number of days, total rent and other recreational costs.

REPORT–

This panel generates all the information regarding the customers between the ‘from’ and ‘to’ dates.

It also calculates the total amount collected in this period and the total number of customers who have checked in at this period.

BACK –

In this user-friendly GUI, the user is provided with a back button at all times to get back to the home frame and resume our work.

EXIT –

It exits the user from the application.

SYSTEM AND SOFTWARE USED

OS Name – Microsoft Windows 10 Home

System Name DESKTOP–BLE6CMQ

System Type –x64-based PC

Java Platform used–NETBEANS IDE 8.2

MySQL version used–MYSQL 5.5

SOURCE CODE

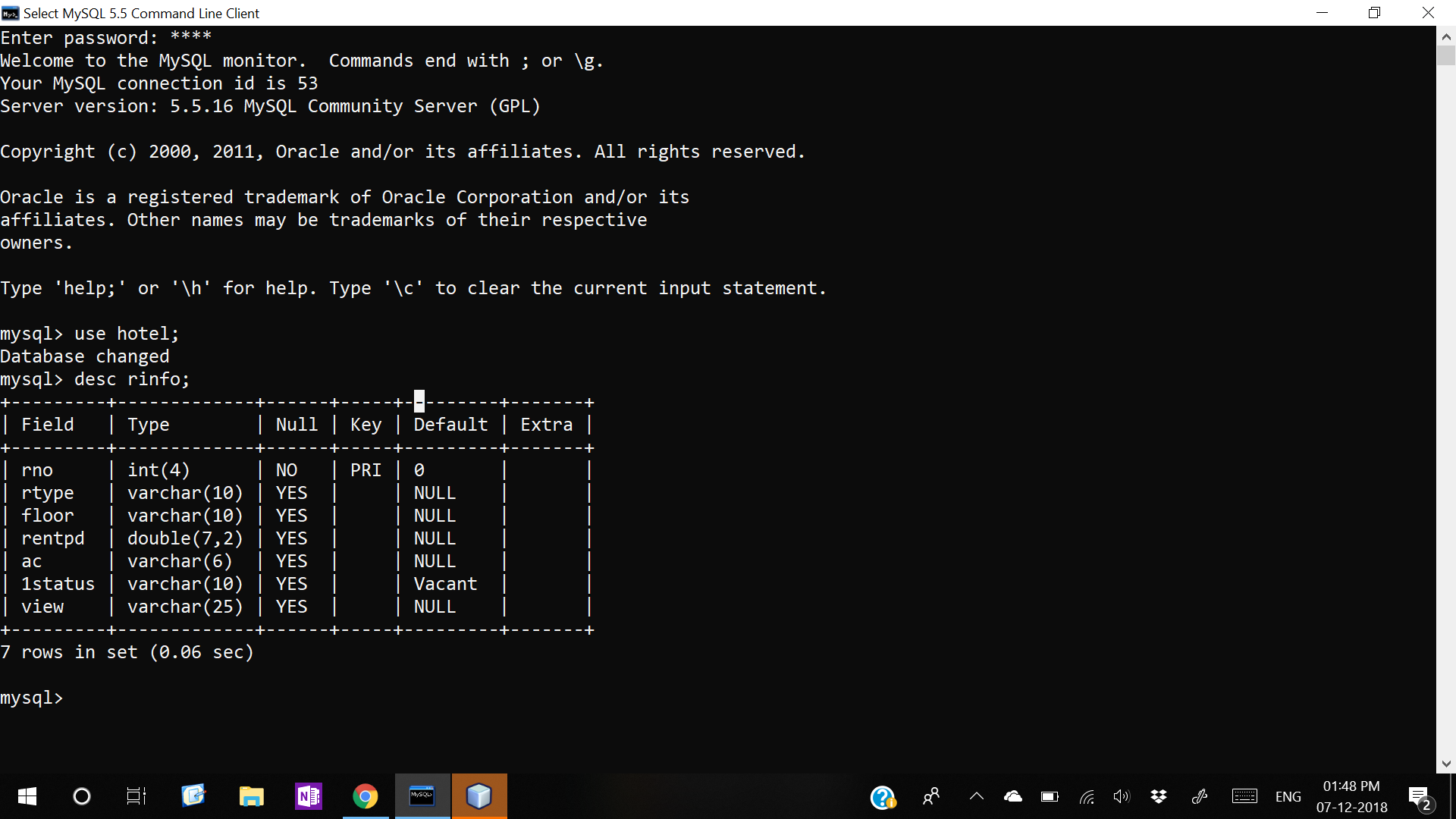
MySQL (Tables Created) -

Room Information Table

The name of the room information table is rinfo. It gives the room details, i.e., the room number, type, floor, rent per day, view.

THE SQL QUERY TO CREATE THE TABLE IS:

Create table room(rno int(4) primary key, rtype varchar(10), floor varchar(10), rentpd double(7,2), ac varchar(6), 1status varchar(10) default ‘vacant’, view varchar(25));

TABLE STRUCTURE:

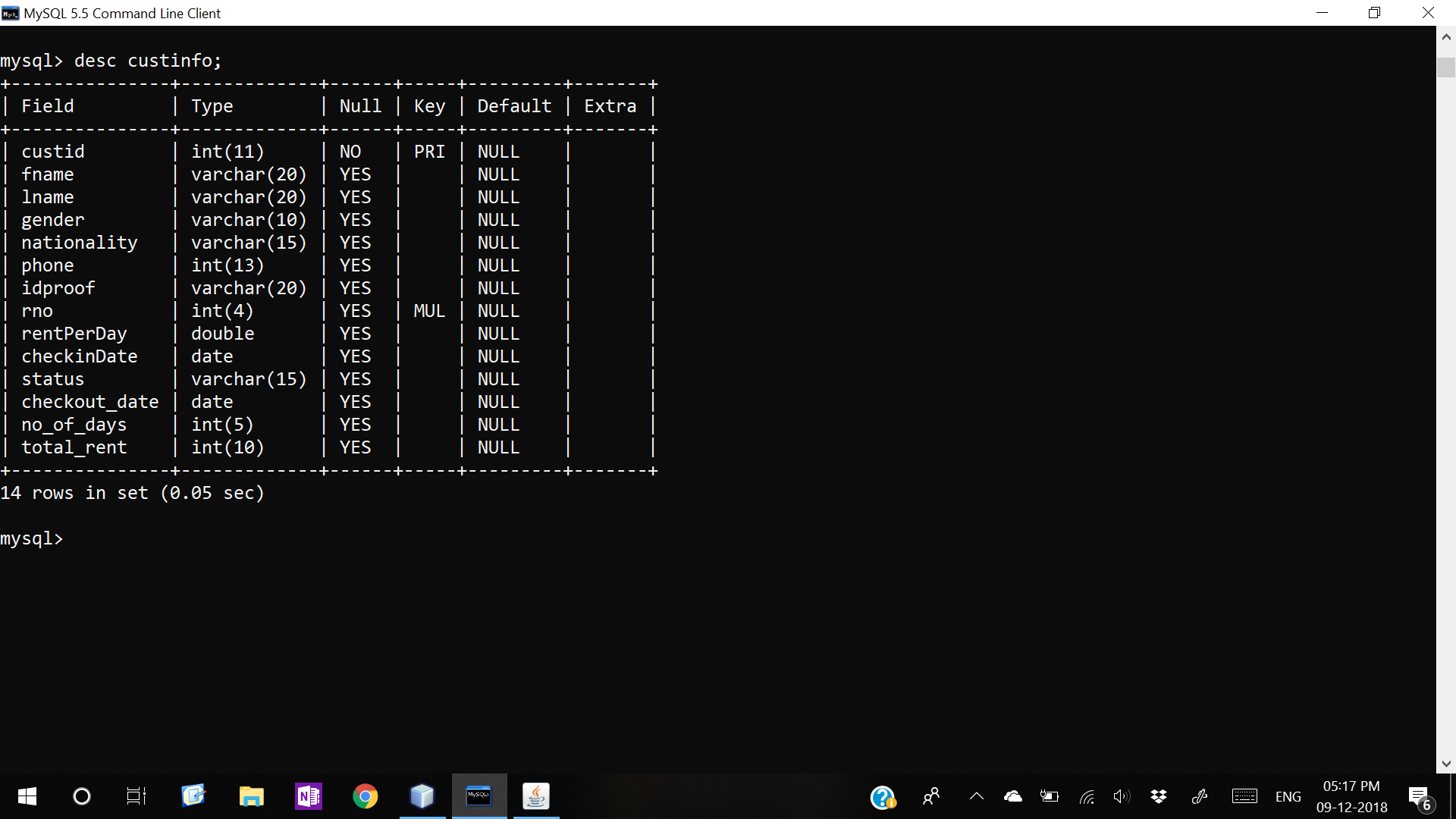
Customer Information Table

The name of the room information table is custinfo.It stores the details about the customer’s details including the checkin date and checkout date.

THE SQL QUERY TO CREATE THE TABLE IS:

Create table custinfo(custid int(11) primary key, fname varchar(20), lname varchar(20), gender varchar(10), nationality varchar(15), phone int(13), idproof varchar(20), rno int(4), rentPerDay double, checkinDate date, status varchar(15), checkout\_date date, no\_of\_days int(5), total\_rent int(10), foreign key(rno) references rinfo(rno));

TABLE STRUCTURE:

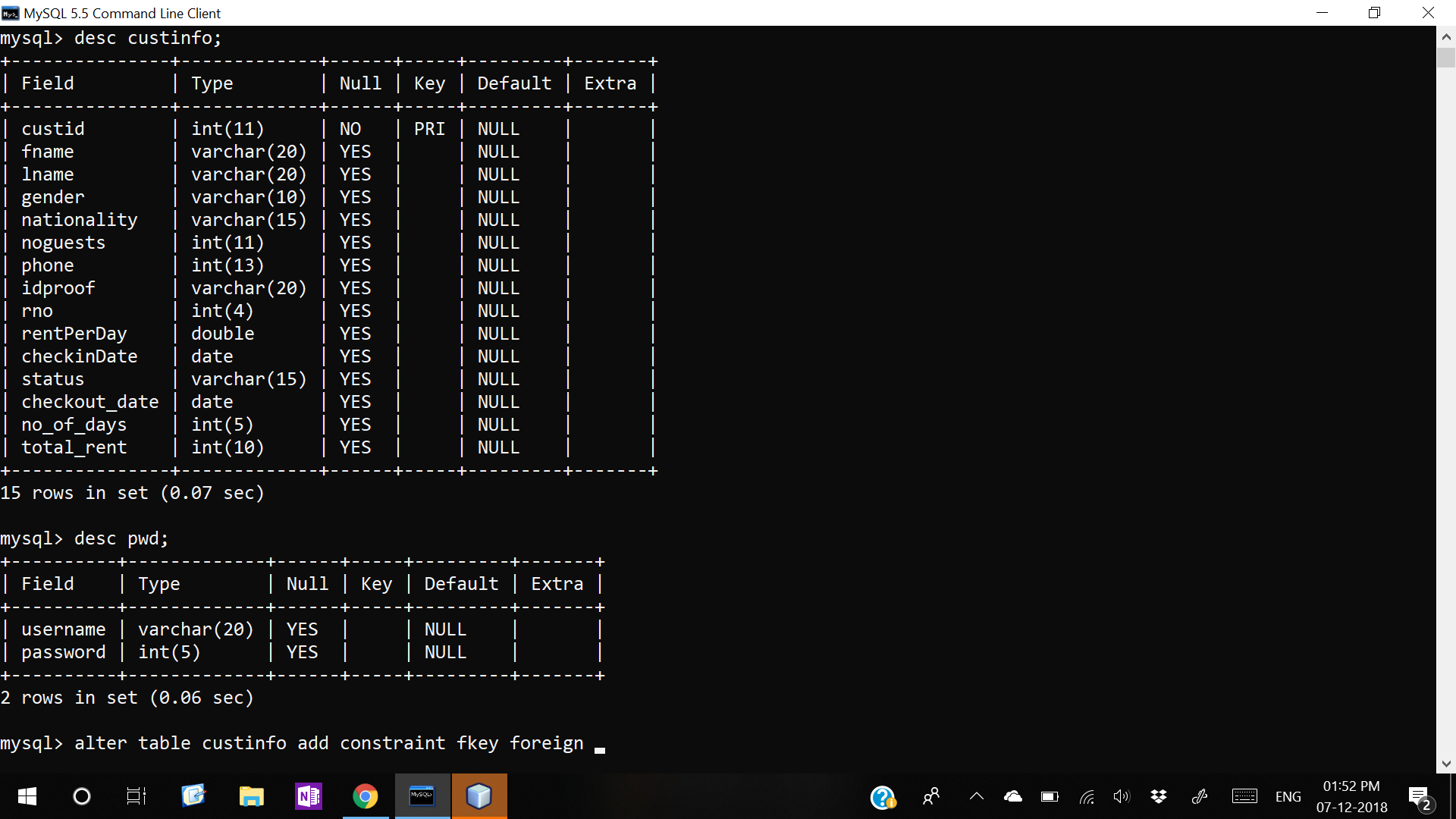
****

Password Table

The name of the password table is pwd. It stores the password of each authorized user.

THE SQL QUERY TO CREATE THE TABLE IS:

Create table pwd(username varchar(20), password int(5));

TABLE STRUCTURE:

MAIN FRAME



Source Code

**//Import statements**

package hotelmanagement;

import java.awt.Color;

import java.awt.Font;

import java.awt.Graphics;

import java.awt.Graphics2D;

import java.sql.Connection;

import java.sql.DriverManager;

import javax.swing.JOptionPane;

//for moving text at the top of the frame-like marquee effect

public class main extends javax.swing.JFrame {

int x=0,y=200;

public void paint(Graphics g)

{super.paint(g);

Graphics2D g2=(Graphics2D)g;

Font f=new Font("Georgia Pro Cond Black",Font.BOLD,80);

g2.setFont(f);

g2.setColor(Color.WHITE);

g2.drawString("Hotel PHEONIX",x,y);// for moving text coding

try{Thread.sleep(50);}

catch(Exception e){}

x+=10;

if(x>this.getWidth())

x=0;

repaint();}

Connection con; //setting up the driver

void opencon(){

try{

Class.forName("java.sql.Driver");

con=DriverManager.getConnection("jdbc:mysql://localhost/hotel","root","pass");

}

catch (Exception e){

JOptionPane.showMessageDialog(this,e.getMessage());

}

}

public main() {

initComponents();

opencon();

}

**//formWindowOpened**

opencon();

jPanel2.removeAll();

jPanel2.repaint();

jPanel2.revalidate();

jPanel2.add(home);

jPanel2.repaint();

jPanel2.revalidate();

**//roominfoActionPerformed**

jPanel2.removeAll();

jPanel2.repaint();

jPanel2.revalidate();

jPanel2.add(roominfo1); //roominfo1 panel under the card layout

jPanel2.repaint();

jPanel2.revalidate();

**//checkoutActionPerformed**

jPanel2.removeAll();

jPanel2.repaint();

jPanel2.revalidate();

jPanel2.add(checkout1); //checkout1 panel under the card layout

jPanel2.repaint();

jPanel2.revalidate();

**//reportActionPerformed**

jPanel2.removeAll();

jPanel2.repaint();

jPanel2.revalidate();

jPanel2.add(report1); //report1 panel under the card layout

jPanel2.repaint();

jPanel2.revalidate();

**//updateActionPerformed**

jPanel2.removeAll();

jPanel2.repaint();

jPanel2.revalidate();

jPanel2.add(update1); //update1 panel under the card layout

jPanel2.repaint();

jPanel2.revalidate();

**//backActionPerformed**

// user friendly button which helps the user to return back from any panel at any time

jPanel2.removeAll();

jPanel2.repaint();

jPanel2.revalidate();

jPanel2.add(home);

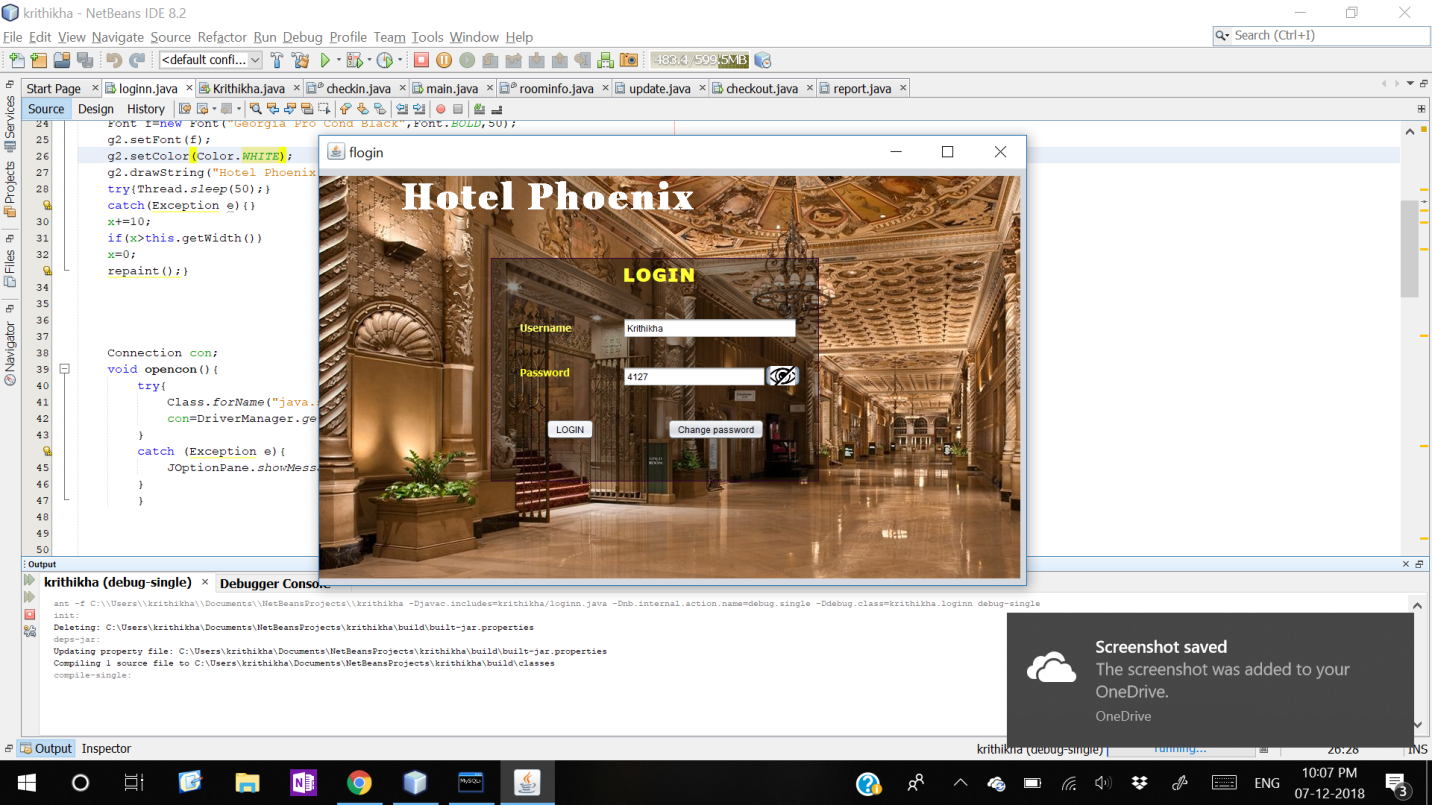
jPanel2.repaint();

jPanel2.revalidate();

**//exitActionPerformed**

System.exit(0); //exits the application

LOGIN FRAME



LOGIN FRAME DETAILS -

|  |  |
| --- | --- |
| CONTROL | CONTROL NAME |
| Username (Text Field) | plogin |
| Password (Password Field) | pass |
| LOGIN (Button) | login |
| CHANGE PASSWORD (Button) | change |
| Show password (Toggle Button) | eye |

Source Code

**//Import statements**

package hotelmanagement;

import java.awt.Color;

import java.awt.Font;

import java.awt.Graphics;

import java.awt.Graphics2D;

import java.sql.\*;

import javax.swing.ImageIcon;

import javax.swing.JOptionPane;

//for moving text like marquee behaviour

public class loginn extends javax.swing.JFrame {

int x=0,y=100;

public void paint(Graphics g)

{super.paint(g);

Graphics2D g2=(Graphics2D)g;

Font f=new Font("Georgia Pro Cond Black",Font.BOLD,30);

g2.setFont(f);

g2.setColor(Color.WHITE);

g2.drawString("Hotel Phoenix ",x,y);

try{Thread.sleep(50);}

catch(Exception e){

JOptionPane.showMessageDialog(this,e.getMessage());

}

x+=10;

if(x>this.getWidth())

x=0;

repaint();}

Connection con;

void opencon(){

try{

Class.forName("java.sql.Driver"); con=DriverManager.getConnection("jdbc:mysql://localhost/hotel","root","pass");

}

catch (Exception e){

JOptionPane.showMessageDialog(this,e.getMessage());

}

}

String login(String u,String pwd){

String q="";

try{ String c="Select \* from pwd";

if(u.equals("")||pwd.equals(""))

{q="";}

else{

Statement st=con.createStatement();

ResultSet rs=st.executeQuery(c);

while(rs.next()){

String a=rs.getString("username");

String p=rs.getString("password");

if(a.equalsIgnoreCase(u) && p.equalsIgnoreCase(pwd)){

q="yes";

break; }

else

q="Your username and password do not match. Please try again.";

}

}

}

catch(Exception e){

JOptionPane.showMessageDialog(this,"Error in connection");

}

return q; }

public loginn() {

initComponents();

plogin.setBackground(new Color(0,0,0,105));

}

**// formWindowOpened**

opencon();

**//loginActionPerformed**

String u=user.getText();

String pwd=new String(pass.getPassword());

String q=login(u,pwd);

if(q.equals(""))

JOptionPane.showMessageDialog(null,"Please enter your username and password to login");

else if (q!="yes")

JOptionPane.showMessageDialog(this,q);

else if(q.equals("yes")){

main frame=new main();

frame.setVisible(true);

this.dispose();}

}

//this is the toogle button to reveal and unreveal the password

**//eyeActionPerformed**

if(eye.isSelected()){

pass.setEchoChar((char)0);

ImageIcon img=new ImageIcon("C:\\Users\\krithikha\\Desktop\\ip project\\closeeye.jpg"); eye.setIcon(img);

}

else{

pass.setEchoChar('•');

ImageIcon img=new ImageIcon("C:\\Users\\krithikha\\Desktop\\ip project\\eye.jpg");

eye.setIcon(img);

}

}

//to update your password after authentication

**//changeActionPerformed**

try{

String nw;

String confirm;

String query;

String u=user.getText();

String pwd=new String(pass.getPassword());

String q=login(u,pwd);

if(q.equals(""))

JOptionPane.showMessageDialog(null,"Enter your username and password in the feilds");}

else if(q!=("yes")){

JOptionPane.showMessageDialog(null,"Your username and password do not match. You need to enter correct information to reset your password");}

else if (q.equals("yes")){

nw= JOptionPane.showInputDialog(null,"Enter new password");

confirm=JOptionPane.showInputDialog(null,"Confirm your password");

if(nw.equals(confirm)){

query="Update pwd set password='"+nw+"'where username='"+u+"'"; Statement stmt=con.createStatement();

stmt.executeUpdate(query);

JOptionPane.showMessageDialog(null,&quot;Your password is changed&quot;);

}

else

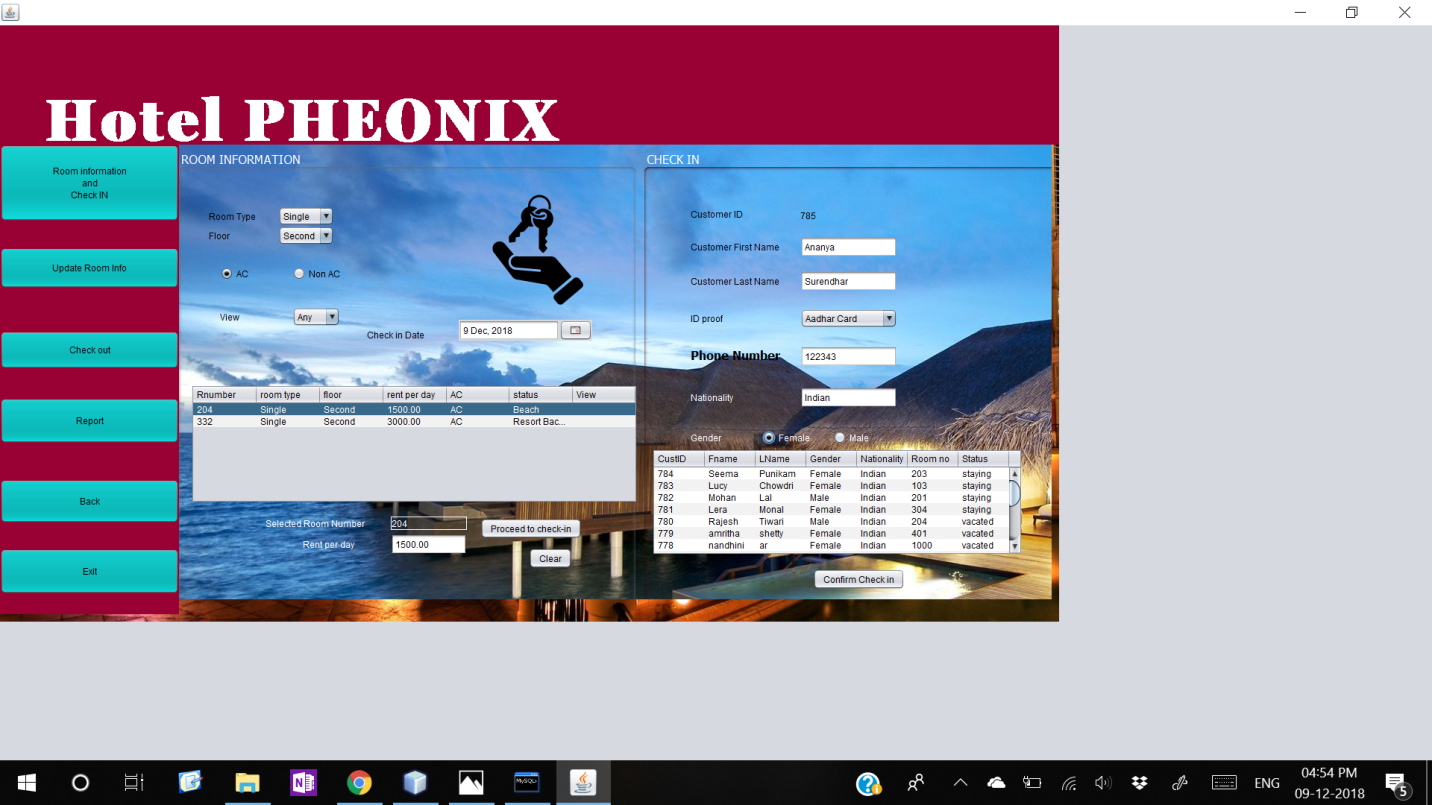
JOptionPane.showMessageDialog(null,"The password and its confirmation do not match");

}}

catch(Exception e){

JOptionPane.showMessageDialog(null,e.getMessage());}

}

 ROOM INFORMATION & CHECKIN

UPDATE DETAILS -

|  |  |
| --- | --- |
| **CONTROL** | **CONTROLNAME** |
| **TEXTFIELDS**  Customer ID  Customer First Name  Customer Last name  Phone  Nationality  Selected Room Number  Rent per day | cid  fname  lname  phone  nat  rn  rpd |
| **RADIOBUTTON**  Ac  Non-Ac  Male  Female | Ac  nonac  male  female |
| **COMBOBOX**  View  Room Type  Floor  ID proof | View  rtype  floor  proof |
| **TABLE**  Room information  Customer information | Tbl  cus |
| **DATECHOOSER**  Check in Date | date |
| **BUTTONS**  Clear  Proceed to Check-in  Confirm Check in | Clear  Proceed  checkin |

Source Code

**//Import statements**

import java.awt.Color;

import java.sql.\*;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.Date;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

Connection method

Connection con; String q;

void opencon(){

try{

Class.forName("java.sql.Driver");

con=DriverManager.getConnection("jdbc:mysql://localhost/hotel","root","pass");

}

catch (Exception e){

JOptionPane.showMessageDialog(this,e.getMessage());

}}

**Filter method:**

//to filter the records according to the requirements of the customer

void filter(){

try{

q="Select \* from rinfo where 1status='Vacant'";

String s=(String)rtype.getSelectedItem();

if(s!="Any")

q+=" and rtype like '"+s+"'";

String n=(String)floor.getSelectedItem();

if(n!="Any")

q+=" and floor like '"+n+"'";

if (ac.isSelected())

q+="and ac like 'ac'";

if (nonac.isSelected())

q+="and ac like 'Non-Ac'";

String v=(String)view.getSelectedItem();

if(v!="Any")

q+="and view like '"+v+"'";

Statement stmt=(Statement)con.createStatement();

ResultSet rs=stmt.executeQuery(q);

DefaultTableModel dtm= (DefaultTableModel)tbl.getModel();

while(dtm.getRowCount()>0)

dtm.removeRow(0);

String a,c,f,r,ac,vw;

while(rs.next()){

a=rs.getString(1);

c=rs.getString(2);

f=rs.getString(3);

r=rs.getString(4);

ac=rs.getString(5);

vw=rs.getString(7);

Object[] obj={a,c,f,r,ac,vw};

dtm.addRow(obj);

}

} catch(Exception e){ JOptionPane.showMessageDialog(this,e.getMessage());

}}

**//viewItemStateChanged**

filter();

//proceed button is to proceed to enter the customer information after choosing the room

**//proceedActionPerformed**

pane2.setVisible(true); //pane2 the check in confirmation panel

try{

int c = 0;

String q2="Select max(custid) from custinfo";

Statement stmt=con.createStatement();

ResultSet rs=stmt.executeQuery(q2);

while(rs.next()){

c = rs.getInt(1);

c+=1;

cid.setText(""+c);

}}

catch(Exception e){JOptionPane.showMessageDialog(this,e.getMessage());

}

//to get the rent per day and room number of the selected row in the text field

**//tblMouseClicked**

DefaultTableModel dtm= (DefaultTableModel)tbl.getModel();

int selectedRowIndex=tbl.getSelectedRow();

rn.setText(dtm.getValueAt(selectedRowIndex, 0).toString());

rpd.setText(dtm.getValueAt(selectedRowIndex,3).toString());

**//floorItemStateChanged**

filter();

**//rtypeItemStateChanged**

filter();

**//nonacActionPerformed**

filter();

**//nonacItemStateChanged**

filter();

**//acActionPerformed**

filter();

**//formMouseEntered**

if(date.getDate()==null)

date.setDate(new Date()); //this sets today’s date to the datechooser named date

try{

String q3="Select \* from custinfo order by custid desc";

Statement stmt3=con.createStatement();

ResultSet rs=stmt3.executeQuery(q3);

DefaultTableModel dtm=(DefaultTableModel)cus.getModel();

while(dtm.getRowCount()>0)

dtm.removeRow(0);

while(rs.next())

dtm.addRow(new Object[]{rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5),

rs.getString(8)});

} catch(Exception e){JOptionPane.showMessageDialog(this,e.getMessage());

}

//confirms the checkin and enters all the values to the sql and java table

**//checkinActionPerformed**

DateFormat fmt = new SimpleDateFormat("yyyy-MM-dd");

try{

int l=phone.getText().length();

if(l>10 || l==0){

JOptionPane.showMessageDialog(this,"Invalid Phone number");

phone.setText("");

}

String a=cid.getText();

String b=fname.getText();

String aa=lname.getText();

String c=(male.isSelected())?("Male"):("Female");

String d=nat.getText();

String h=proof.getSelectedItem().toString();

int i=Integer.parseInt(rn.getText());

String j=rpd.getText();

String f=phone.getText();

String k = fmt.format(this.date.getDate());

String q1="Insert into custinfo values('"+a+"','"+b+"','"+aa+"','"+c+"','"+d+"','"+f+"','"+h+"',"+i+",'"+j+"','"+k+"','occupi ed',null, null,null)";

Statement stmt=con.createStatement();

stmt.executeUpdate(q1);

String q2="Update rinfo set 1status='occupied' where rno="+i;

Statement stmt2=con.createStatement();

stmt2.executeUpdate(q2);

JOptionPane.showMessageDialog(this,"Successfully checked in!");

rtype.setSelectedIndex(0); //to clear all contents after check in

floor.setSelectedIndex(0);

ac.setSelected(true);

rn.setText("");

rpd.setText("");

fname.setText("");

lname.setText("")

proof.setSelectedIndex(0);

phone.setText("");

nat.setText("Indian");

male.setSelected(true);

tbl.removeAll();

}

catch(Exception e){JOptionPane.showMessageDialog(this,e.getMessage());

}

**//clearActionPerformed**

rtype.setSelectedIndex(0);

floor.setSelectedIndex(0);

ac.setSelected(true);

rn.setText("");

rpd.setText("");

fname.setText("");

lname.setText("")

proof.setSelectedIndex(0);

phone.setText("");

nat.setText("Indian");

male.setSelected(true);

pane2.setVisible(false);

tbl.removeAll();

//to enter values into JAVA table

**//cusFocusGained**

try{

String q3="Select \* from custinfo order by custid desc";

Statement stmt3=con.createStatement();

ResultSet rs=stmt3.executeQuery(q3);

DefaultTableModel dtm=(DefaultTableModel)cus.getModel();

while(dtm.getRowCount()>0)

dtm.removeRow(0);

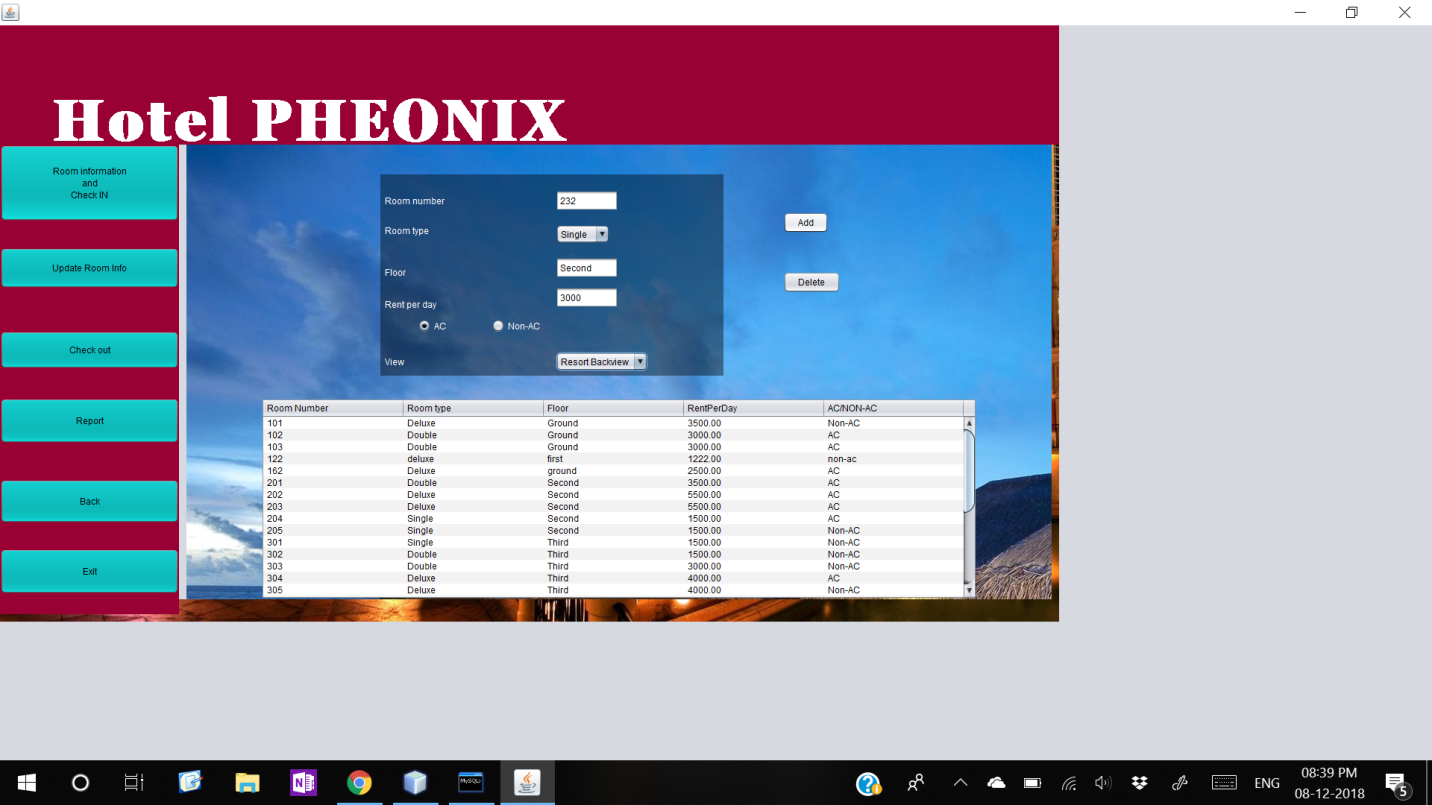
while(rs.next())

dtm.addRow(new Object[]{rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5),rs.getString(8)});

}

catch(Exception e){JOptionPane.showMessageDialog(this,e.getMessage());

}

 UPDATE ROOM INFO

UPDATE DETAILS -

|  |  |
| --- | --- |
| **CONTROL** | **CONTROL NAME** |
| **TEXTFIELDS**  Room number  Floor  Rent Per day | rno  floor  rent |
| **RADIOBUTTON**  ac  non-ac | ac  nonac |
| **BUTTON**  Add  Delete | add  delete |
| **TABLE** | tb |
| **COMBOBOX**  Room type  view | rtype  view |
|  |  |
|  |  |

Source Code

**//Import statements**

import java.awt.Color;

import java.awt.HeadlessException;

import java.sql.\*;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

Connection method

Connection con; String q;

void opencon(){

try{

Class.forName("java.sql.Driver");

con=DriverManager.getConnection("jdbc:mysql://localhost/hotel","root","pass");

}

catch (Exception e){

JOptionPane.showMessageDialog(this,e.getMessage());

}}

public update() {

initComponents();

opencon();

panel.setBackground(new Color(0,0,0,105)); //to set a translucent background

}

//to add rooms in the hotel application

**//addActionPerformed**

try{

String a="";

String stat="Vacant";

int r=Integer.parseInt(rno.getText());

String rt=(String)rtype.getSelectedItem();

String flr=floor.getText();

int rpd=Integer.parseInt(rent.getText());

String v=(String)view.getSelectedItem();

if(ac.isSelected())

a="AC";

if(nonac.isSelected())

a="Non-AC";

String q1="Insert into rinfo values( "+r+",'"+rt+"','"+flr+"',"+rpd+",'"+a+"','"+stat+"','"+v+"')";

Statement stmt=con.createStatement();

stmt.executeUpdate(q1);

JOptionPane.showMessageDialog(this,"Record successfully added!");

rno.setText("");

floor.setText("");

rent.setText("");

ac.setSelected(“true”);

rtype.setSelectedValue="Single";

}

catch(Exception e){JOptionPane.showMessageDialog(this,e.getMessage());

}

//to delete rooms in the hotel application

**//deleteActionPerformed**

try{

String rno=JOptionPane.showInputDialog("Enter room number to be deleted");

String q2="Delete from rinfo where rno="+rno;

Statement stmt=con.createStatement();

stmt.executeUpdate(q2);

JOptionPane.showMessageDialog(this,"Record with rno="+rno+" succesfully deleted");

}

catch(Exception e){JOptionPane.showMessageDialog(this,e.getMessage());

}

//to enter values into JAVA table

**//formFocusGained**

try{

String q="Select \* from rinfo";

Statement stmt=con.createStatement();

ResultSet rs=stmt.executeQuery(q);

DefaultTableModel dtm=(DefaultTableModel)tb.getModel();

while(dtm.getRowCount()>0)

dtm.removeRow(0);

while(rs.next())

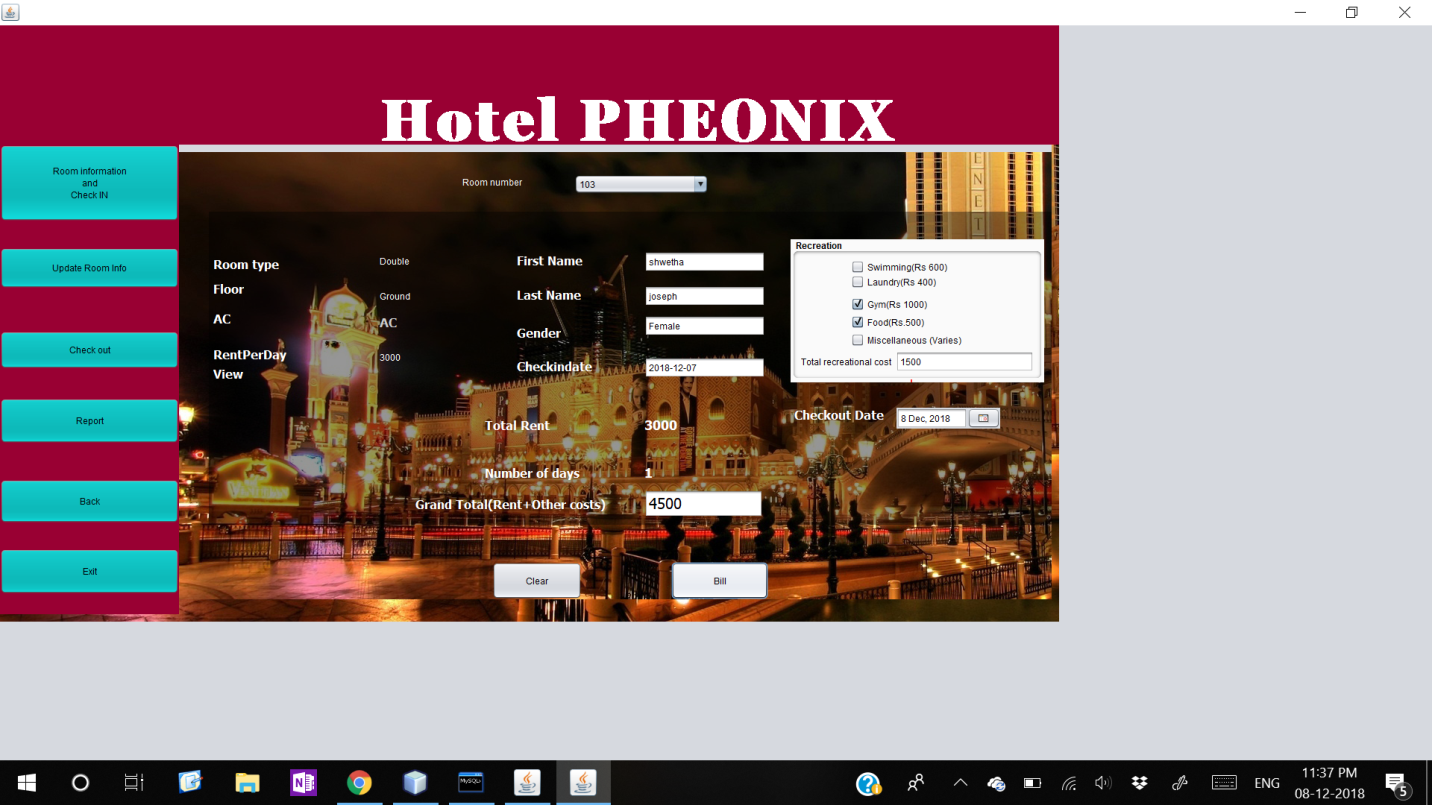
dtm.addRow(new Object[]{rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5)});

}

catch(Exception e){JOptionPane.showMessageDialog(this,e.getMessage());

}

CHECKOUT



|  |  |
| --- | --- |
| **CONTROL** | **CONTROL NAME** |
| **TEXTFIELDS**  First Name  Last Name  Gender  ChechinDate  Total recreational cost  Grand Total | fname  lname  Gen  chkdate  rectotal  gtot |
| **BUTTONS**  Clear  Bill | clear  bill |
| **COMBOBOX**  Room Number | cb |
| **LABEL**  Room type  Floor  AC  Rent Per Day  View | rtype  floor  ac  rentpd  view |
| **CHECKBOX**  Swimming  Laundry  Gym  Food  Miscellaneous | swim  lndry  gym  food  mis |
| **DATECHOOSER**  Checkout Date | dtc |
|  |  |
|  |  |

Source Code

**//Import statements**

import com.toedter.calendar.JMonthChooser;

import java.awt.Color;

import java.sql.\*;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.Calendar;

import java.util.Date;

import javax.swing.JOptionPane;

**Connection method:**

   Connection con;

   void opencon(){

       try{

           Class.forName("java.sql.Driver");

           con=DriverManager.getConnection("jdbc:mysql://localhost/hotel","root","pass");

       }

       catch (Exception e){

           JOptionPane.showMessageDialog(this,e.getMessage());

       }

}

**Chkout method:**

   Date cdt;

     void chkout(){

         try{

           DateFormat fmt = new SimpleDateFormat("yyyy-MM-dd");

       String d1 = fmt.format(this.dtc.getDate());

     String rno= (String) cb.getSelectedItem();

       String type="";

        String flr="";

        String acond="";

        String vw="";

        String fn="";

        String ln="";

        String g="";

        int rpd=0;

       String q="Select \* from rinfo natural join custinfo where custinfo.rno='"+rno+"' and 1status ='occupied'";

      Statement stmt =con.createStatement();

              ResultSet rs=stmt.executeQuery(q);

              if(rs.next()){

               type=rs.getString(2);

               flr=rs.getString(3);

               rpd=rs.getInt(4);

              acond=rs.getString(5);

              vw=rs.getString(7);

              cdt= rs.getDate(16);

              fn= rs.getString(9);

              ln=rs.getString(10);

               g=rs.getString(11);

              rtype.setText(type);

              floor.setText(flr);

              ac.setText(acond);

              view.setText(vw);

              rentpd.setText(""+rpd);

              chkdate.setText(""+cdt);

              fname.setText(fn);

              lname.setText(ln);

               gen.setText(g);

      }

              else{

              JOptionPane.showMessageDialog(this,"Please select the room number for checkout");

              }

                                  }

       catch(Exception e){

           JOptionPane.showMessageDialog(this,e.getMessage());

       }

     }

//To calculate the days between checkindate and checkout date

**Days method:**

     void days(){

Date d2 = dtc.getDate();                                   //checkin date

long diffdys = d2.getTime() - cdt.getTime();

            long dys= (long)( diffdys / (1000 \* 60 \* 60 \* 24));                                 //milliseconds to days

                  days.setText(""+dys);

     }

//to calculate other recreational costs apart from rent

**Rec method:**

      int rc=0;

void rec(){

         if(swm.isSelected())

   rc+=600;

if(lndry.isSelected())

   rc+=400;

if(gym.isSelected())

   rc+=1000;

if(food.isSelected())

   rc+=500;

rectotal.setText(""+rc);

     }

public  checkout() {

       initComponents();

       opencon();

                bg.setBackground(new Color(0,0,0,105));

                rec.setBackground(new Color(250,250,250));

}

//cbItemStateChanged

   chkout();

}

cbFocusGained {

           if(dtc.getDate()==null)

           dtc.setDate(new Date()); //today’s date

                             try{

         String q="Select \* from rinfo where 1status='occupied'";

         Statement stmt =con.createStatement();

          ResultSet rs=stmt.executeQuery(q);

           while (rs.next()){

           int r=rs.getInt(1);

           cb.addItem(""+r);

}         }

       catch(Exception e){            JOptionPane.showMessageDialog(this,e.getMessage());

}

                       //to calculate the grand total bill inclusive of rent and orecreational costs

**//BillActionPerfomed**

  days();

      try{

   int c = 0;

   String  rno=(String)cb.getSelectedItem();

   DateFormat fmt = new SimpleDateFormat("yyyy-MM-dd");

   String d1 = fmt.format(this.dtc.getDate());

int rpd=Integer.parseInt(rentpd.getText());

int no=Integer.parseInt(days.getText());

int tot=rpd\*no ;

rent.setText(""+tot);

int rcost=Integer.parseInt(rectotal.getText());

int bill=tot+rcost;

gtot.setText(""+bill):

String q2="Update custinfo set        status='vacated',checkout\_date='"+d1+"',"+"no\_of\_days="+no+",total\_rent="+tot+" where rno='"+rno+"' and status='staying'";

               String q3="Update rinfo set 1status='vacant' where rno='"+rno+"'";

               Statement stmt2=con.createStatement();

               Statement stmt3=con.createStatement();

               stmt2.executeUpdate(q2);

               stmt3.executeUpdate(q3);

            JOptionPane.showMessageDialog(this,"The Total amount to be paid is "+bill);

            JOptionPane.showMessageDialog(this,"Successfuly checked out");

       }

       catch(Exception e){  JOptionPane.showMessageDialog(this,e.getMessage());

   }

 //formFocusGained

                if(dtc.getDate()==null)

                dtc.setDate(new Date());

                 days();

     // clearActionPerformed

       cb.setSelectedIndex(0);

       rtype.setText("");

       floor.setText("");

       rentpd.setText("");

       view.setText("");

       ac.setText("");

       fname.setText("");

       lname.setText("");

       gen.setText("");

       chkdate.setText("");

       swm.setSelected(false);

       lndry.setSelected(false);

       gym.setSelected(false);

       food.setSelected(false);

       mis.setSelected(false);

       rectotal.setText("");

        rent.setText("");

        days.setText("");

        gtot.setText("");

   }

  private void swmItemStateChanged(java.awt.event.ItemEvent evt) {

       rec();

   }

  //lndryItemStateChanged

:

       rec();

//gymItemStateChanged

       rec();

//foodItemStateChanged

               rec();

//miscellaneous costs like breakage and other are to be specified

   //misItemStateChanged

    if(mis.isSelected()){

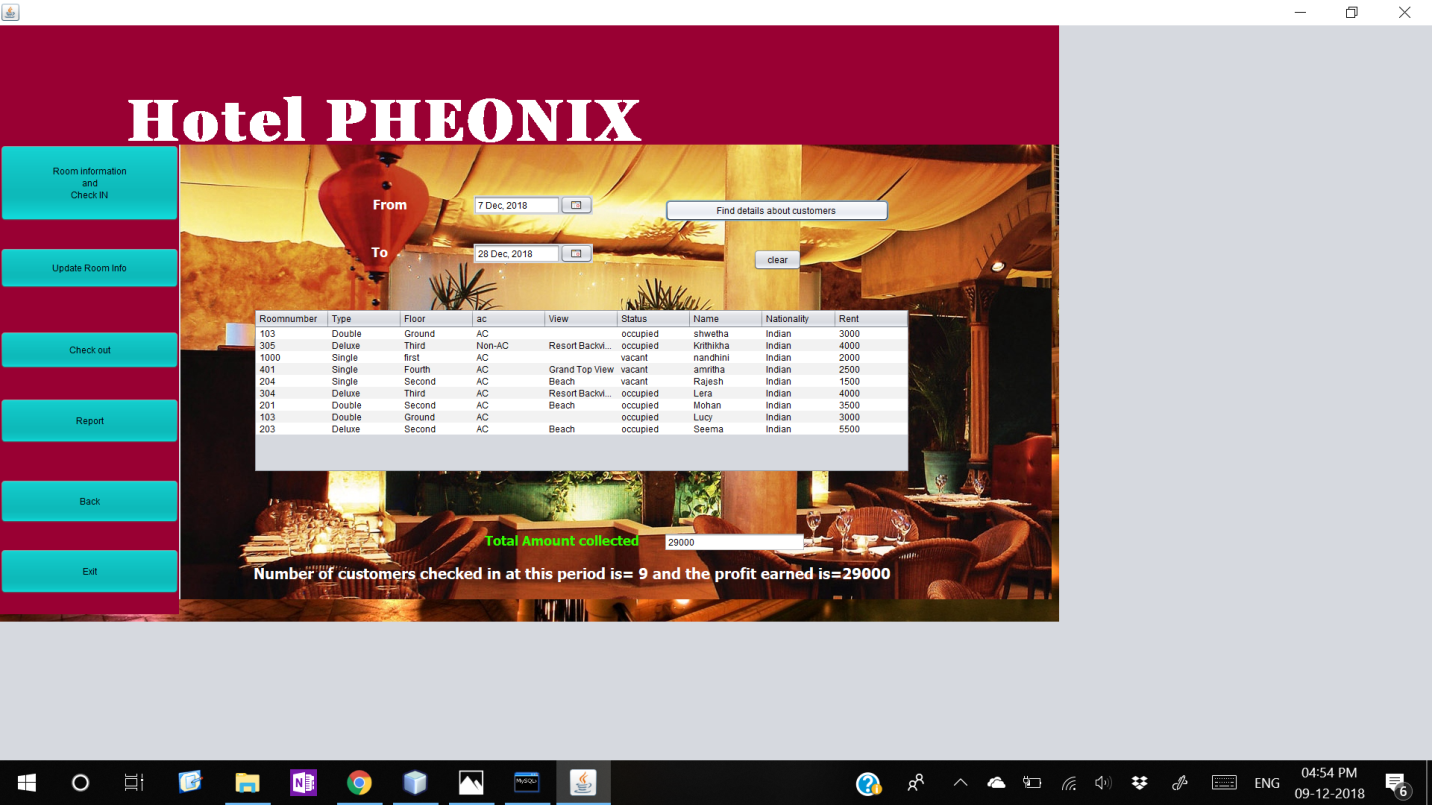
   String m= JOptionPane.showInputDialog(this,"Enter miscellaneous charges-");

   int mn=Integer.parseInt(m);

   int r1=rc+mn;

   rectotal.setText(""+r1);}

REPORT



|  |  |
| --- | --- |
| **CONTROL** | **CONTROL NAME** |
| **DATECHOOSER**  From  To | from  to |
| **BUTTON**  Find  clear | find  clear |
| **TEXTFIELD**  Total Amount Collected | profit |
| **TABLE** | report |



Source Code

**//Import statements**

import java.awt.Color;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.Date;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

**Connection method:**

Connection con;

void opencon(){

try{

Class.forName("java.sql.Driver");

con=DriverManager.getConnection("jdbc:mysql://localhost/hotel","root","pass");

}

catch (Exception e){

JOptionPane.showMessageDialog(this,e.getMessage());

}

}

**Find method:**

//to find the number of days between the from and to checkin-Dates

String dt1,dt2;

void find(String d1,String d2){

try{

String q="Select \* from rinfo natural join custinfo where checkinDate between '"+d1+"' and '"+d2+"' ";

Statement stmt=con.createStatement();

ResultSet rs=stmt.executeQuery(q);

DefaultTableModel dtm=(DefaultTableModel)report.getModel();

while(dtm.getRowCount()>0)

dtm.removeRow(0);

String id,f,rno,rt,a,r,b,c,t;

int rent;

while(rs.next()){

r=rs.getString(1); //rno

id=rs.getString(2); //rtype

f=rs.getString(3); //floor

rent=rs.getInt(4); //rent of room

rt=rs.getString(5); //ac

a=rs.getString(6); //1status

v=rs.getString(7); //view from room

b=rs.getString(9); //fname

c=rs.getString(12); //nationality

Object nr[]={r,id,f,rt,v,a,b,c,rent,};

dtm.addRow(nr);}

int p=0;

String q2="Select sum(rentperday) from custinfo where checkindate between '"+dt1+"' and '"+dt2+" ' ";

Statement stmt2=con.createStatement();

ResultSet rs2=stmt.executeQuery(q2);

while(rs2.next())

p=rs2.getInt(1);

profit.setText(""+p);

int n=report.getRowCount();

JOptionPane.showMessageDialog(this,"Number of customers in this period is= "+n+" and the profit earned is="+p);

}

catch(Exception e){

JOptionPane.showMessageDialog(this,e.getMessage());

}

}

public report() {

opencon();

initComponents();

}

**//findActionPerformed**

DateFormat fmt = new SimpleDateFormat("yyyy-MM-dd");

dt1 = fmt.format(this.from.getDate());

dt2= fmt.format(this.to.getDate());

find(dt1,dt2);

**//clearActionPerformed**

from.setCalendar(null);

to.setCalendar(null);

report.removeAll();

profit.setText("");

BIBLIOGRAPHY

* <https://images.google.com>
* <https://stackoverflow.com>
* https://docs.oracle.com
* <https://www.geeksforgeeks.org>
* Informatics Practices Class XII (Sumita Arora)