

Appendix C: Source Code

The source code for the product is given below. It must be noted that the code highlighted in grey was automatically generated by NetBeans for the GUI components.

LoginPage Class:

```
*Hospital Management System
*IDE - Netbeans
*Platform - Mac
*/
import java.awt.event.KeyEvent;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JRadioButton;
import javax.swing.JToggleButton;

public class LoginPage extends javax.swing.JFrame {
    /**
     * Creates new form LoginPage
     */
    public LoginPage() {
        initComponents();
    }
    String username;
    String password;

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    BEGIN:initComponents
    private void initComponents() {

        buttonGroup1 = new javax.swing.ButtonGroup();
```

```
buttonGroup2 = new javax.swing.ButtonGroup();
jMenuItem1 = new javax.swing.JMenuItem();
jMenu1 = new javax.swing.JMenu();
jPanel1 = new javax.swing.JPanel();
jPanel2 = new javax.swing.JPanel();
jLabel1 = new javax.swing.JLabel();
jPanel4 = new javax.swing.JPanel();
jLabel2 = new javax.swing.JLabel();
jPanel6 = new javax.swing.JPanel();
jLabel3 = new javax.swing.JLabel();
Userid = new javax.swing.JTextField();
Password1 = new javax.swing.JPasswordField();
jButtonLogin = new javax.swing.JButton();
```

```
jMenuItem1.setText("jMenuItem1");
```

```
jMenu1.setText("jMenu1");
```

```
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle("Login Page");
```

```
jPanel1.setBackground(new java.awt.Color(153, 153, 153));
jPanel1.setBorder(new javax.swing.border.MatteBorder(null));
```

```
jPanel2.setBackground(new java.awt.Color(102, 102, 102));
```

```
jPanel2.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED, new java.awt.Color(102, 102, 102), new java.awt.Color(102, 102, 102), new java.awt.Color(0, 0, 0), new java.awt.Color(102, 102, 102)));
```

```
jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/LOGO.png"))); // NOI18N
```

```
javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(225, 225, 225)
                .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 499, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(178, 178, 178))
        );
jPanel2Layout.setVerticalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addGap(225, 225, 225)
            .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 499, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(178, 178, 178))
        );
```

```

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(jLabel1)
        .addContainerGap())
    );

```

```

jLabel2.setFont(new java.awt.Font("Kohinoor Bangla", 1, 13)); // NOI18N
jLabel2.setText("USER ID :");

```

```

javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);
jPanel4.setLayout(jPanel4Layout);
jPanel4Layout.setHorizontalGroup(
    jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel4Layout.createSequentialGroup()
            .addContainerGap(23, Short.MAX_VALUE)
            .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 76,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap())
        );
jPanel4Layout.setVerticalGroup(
    jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel4Layout.createSequentialGroup()
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(jLabel2)
            .addContainerGap())
        );

```

```

jLabel3.setFont(new java.awt.Font("Kohinoor Bangla", 1, 13)); // NOI18N
jLabel3.setText("PASSWORD:");

```

```

javax.swing.GroupLayout jPanel6Layout = new javax.swing.GroupLayout(jPanel6);
jPanel6.setLayout(jPanel6Layout);
jPanel6Layout.setHorizontalGroup(
    jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel6Layout.createSequentialGroup()
            .addGap(14, 14, 14)
            .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 76,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(15, Short.MAX_VALUE))
        );
jPanel6Layout.setVerticalGroup(
    jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel6Layout.createSequentialGroup()
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel6Layout.createSequentialGroup()

```

```

        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(jLabel3)
        .addContainerGap()
    );

```

```

    Userid.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            UseridActionPerformed(evt);
        }
    });

```

```

    Password1.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            Password1ActionPerformed(evt);
        }
    });
    Password1.addKeyListener(new java.awt.event.KeyAdapter() {
        public void keyPressed(java.awt.event.KeyEvent evt) {
            Password1KeyPressed(evt);
        }
    });

```

```

    jButtonLogin.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/signin.png"))); // NOI18N
    jButtonLogin.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
    jButtonLogin.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButtonLoginActionPerformed(evt);
        }
    });

```

```

    javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
    jPanel1.setLayout(jPanel1Layout);
    jPanel1Layout.setHorizontalGroup(
        jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel2, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(201, 201, 201)

```

```

                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                    .addComponent(jPanel6, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

```

```

        .addGap(49, 49, 49)

    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    G)
        .addComponent(Userid, javax.swing.GroupLayout.DEFAULT_SIZE, 375,
Short.MAX_VALUE)
        .addComponent>Password1))
        .addGap(176, 176, 176))
        .addGroup(jPanel1Layout.createSequentialGroup())
        .addGap(365, 365, 365)
        .addComponent(jButtonLogin, javax.swing.GroupLayout.PREFERRED_SIZE, 163,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );
    jPanel1Layout.setVerticalGroup(
        jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup())
        .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(77, 77, 77)

    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    G)
        .addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(Userid, javax.swing.GroupLayout.PREFERRED_SIZE, 32,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(50, 50, 50)

    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    G)
        .addComponent(jPanel6, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent>Password1, javax.swing.GroupLayout.PREFERRED_SIZE, 34,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)
        .addComponent(jButtonLogin, javax.swing.GroupLayout.PREFERRED_SIZE, 68,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(55, Short.MAX_VALUE))
    );

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

```

```

);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
);

```

```

pack();
} // </editor-fold> // GEN-END: initComponents

```

```

public void login()
{
    if(username.trim().equals("")){
        JOptionPane.showMessageDialog(rootPane, "Enter Your Username to Login", "Empty
Username ", 2);
    }
    else if(password.trim().equals("")){
        JOptionPane.showMessageDialog(rootPane, "Enter Your Password to Login", "Empty
Password", 2);
    }
    else{
        try {
            Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/"
                + "java_hospital_db?serverTimezone=UTC", "root", "");
            String sql = "SELECT * FROM `users` WHERE `username`=? AND `password`=?";
            PreparedStatement ps = con.prepareStatement(sql);
            ps.setString(1, Userid.getText());
            ps.setString(2, Password1.getText());
            ResultSet rs = ps.executeQuery();
            if(rs.next()){
                String s1=rs.getString("type");
                if ( s1.equals("Admin")){
                    JOptionPane.showMessageDialog(null, "Logged in as "+ s1);
                    mainForm adminform = new mainForm();
                    adminform.setVisible(true);
                    adminform.pack();
                    adminform.setLocationRelativeTo(null);
                    adminform.setExtendedState(JFrame.MAXIMIZED_BOTH);
                    this.dispose();
                }else {
                    JOptionPane.showMessageDialog(null, "Logged in as "+ s1);
                    mainForm1 userform = new mainForm1();
                    userform.setVisible(true);
                    userform.pack();
                    userform.setLocationRelativeTo(null);
                    userform.setExtendedState(JFrame.MAXIMIZED_BOTH);
                    this.dispose();
                }
            }
        } catch (SQLException ex) {
            JOptionPane.showMessageDialog(null, "Database Error: " + ex.getMessage());
        }
    }
}

```

```

        }
    }else{
        JOptionPane.showMessageDialog(rootPane, "Wrong User ID or Password",
"Login Error", 2);
    }
    con.close();
    }
    catch (SQLException e) {
        Logger.getLogger(LoginPage.class.getName()).log(Level.SEVERE, null, e);
    }
    }
}
private void jButtonLoginActionPerformed(java.awt.event.ActionEvent evt) {
    login();
}

private void Password1KeyPressed(java.awt.event.KeyEvent evt) {
    if(evt.getKeyCode()== KeyEvent.VK_ENTER)
    {
        login();
    }
}
// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JPasswordField Password1;
private javax.swing.JTextField Userid;
private javax.swing.ButtonGroup buttonGroup1;
private javax.swing.ButtonGroup buttonGroup2;
private javax.swing.JButton jButtonLogin;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JMenu jMenu1;
private javax.swing.JMenuItem jMenuItem1;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel4;
private javax.swing.JPanel jPanel6;
// End of variables declaration//GEN-END:variables
}

```

Registration Class

```

import java.sql.Connection;
import java.sql.Date;
import java.sql.DriverManager;

```

```

import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Vector;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import net.proteanit.sql.DbUtils;
/**
 *
 * @author Abhi
 */
public class Registration extends javax.swing.JFrame {

    /**
     * Creates new form Patient1
     */
    public Registration() {
        initComponents();
        patient_table();
        AutoID();
    }

    Connection con;
    PreparedStatement pst;
    ResultSet rs;
    String pid;
    String pname;
    String psex;
    String pphone;
    String paddress;
    java.util.Date stringDate;

    private void variables()
    {
        pid= ID_Label.getText();
        pname=jTextFieldName.getText().toString();
        psex="";
        if (jRadioButtonMale.isSelected()){
            psex="Male";
        }
        else if (jRadioButtonFemale.isSelected()){
            psex="Female";
        }
    }

```



```

    }
    pphone=jTextFieldContactNumber.getText();
    paddress=jTextAreaAddress.getText();
    stringDate=jDateChooserDOB.getDate();
}

public void Connect(){
    try{
        Connection con=
DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serverTimez
one=UTC", "root", "");
    }catch (SQLException ex) {
        Logger.getLogger(Registration.class.getName()).log(Level.SEVERE, null, ex);
    }
}

private void AutoID(){ try {
    Connect();
    String sql = "SELECT MAX(Patient_ID) FROM `patientsinfo`";
    pst=con.prepareStatement(sql);
    rs=pst.executeQuery();
    rs.next();
    rs.getString("MAX(Patient_ID)");
    if (rs.getString("MAX(Patient_ID)")==null)
    {
        ID_Label.setText("PS001");
    }else
    {
        long id =
Long.parseLong(rs.getString("MAX(Patient_ID)").substring(2,rs.getString("MAX(Patient_ID)")
.length()));
        id ++;
        ID_Label.setText("PS"+String.format("%03d", id));
    }
} catch (SQLException ex) {
    Logger.getLogger(Patients.class.getName()).log(Level.SEVERE, null, ex);
}
}

private void patient_table(){
    try {
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serverTimez
one=UTC", "root", "");
        PreparedStatement pst;
        ResultSet rs;
        pst=con.prepareStatement("select * from patientsinfo ");
    }
}

```

```

rs=pst.executeQuery();
ResultSetMetaData Rsm= rs.getMetaData();
int c;
c=Rsm.getColumnCount();
DefaultTableModel df = (DefaultTableModel)jTable1.getModel();
df.setRowCount(0);
while (rs.next()){
    Vector v2= new Vector();
    for(int i = 1; i<=c; i++)
    {
        v2.add(rs.getString("Patient_ID"));
        v2.add(rs.getString("Patient_Name"));
        v2.add(rs.getString("Patient_Sex"));
        v2.add(rs.getString("Patient_DOB"));
        v2.add(rs.getString("Patient_Phone"));
        v2.add(rs.getString("Patient_Address"));
    }
    df.addRow(v2);
}
} catch (SQLException ex) {
    Logger.getLogger(Registration.class.getName()).log(Level.SEVERE, null, ex);
}
}

```

// saving a new record from text fields

```

private void save(){
    variables();
    if(
        pname.trim().equals("") || pname.trim().equals("Enter Patient Name") ||
        psex.trim().equals("") || pphone.trim().equals("Enter Phone Number") ||
        pphone.trim().equals("") || paddress.trim().equals("") )
    {
        JOptionPane.showMessageDialog(null, "All fields are compulsory "
            +" Kindly check all fields again","Error",JOptionPane.ERROR_MESSAGE);
    }else if (stringDate==null)
    {
        JOptionPane.showMessageDialog(null, "All fields are compulsory "
            +" Choose a valid date","Error",JOptionPane.ERROR_MESSAGE);
    }else
    {
        try {
            Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/"
                + "java_hospital_db?serverTimezone=UTC", "root", "");
            PreparedStatement pst;
            ResultSet rs;
            java.sql.Date sqlDate=new java.sql.Date(jDateChooserDOB.getDate().getTime());
            pst=con.prepareStatement("INSERT INTO `patientsinfo`(`Patient_ID`,`

```

```

        + " `Patient_Name`, `Patient_Sex`, `Patient_DOB`, "
        + "`Patient_Phone`, `Patient_Address`) VALUES (?, ?, ?, ?, ?, ?)");
pst.setString(1, ID_Label.getText());
pst.setString(2, jTextFieldName.getText());
pst.setString(3, psex);
pst.setDate(4, sqlDate);
pst.setString(5, jTextFieldContactNumber.getText());
pst.setString(6, paddress);
int i = pst.executeUpdate();
if(i>0){
    JOptionPane.showMessageDialog(null, "New Patient "+ pname+ " Saved
Successfully");
    patient_table();
    AutoID();
    clear();
}
else {
    JOptionPane.showMessageDialog(null, pname+" 's Information not saved
successfully");
}
} catch (SQLException e) {
    Logger.getLogger(Patients.class.getName()).log(Level.SEVERE, null, e);
    JOptionPane.showMessageDialog(null, pname+" 's Information not saved
successfully");
}

}
}
//deleting the selected row's record
private void delete(){
    String pid=ID_Label.getText();
    String pname= jTextFieldName.getText();
    try {
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serverTimez
one=UTC", "root", "");
        PreparedStatement pst;
        ResultSet rs;
        java.sql.Date sqlDate=new java.sql.Date(jDateChooserDOB.getDate().getTime());
        pst=con.prepareStatement("delete from `patientsinfo` where Patient_ID=?");
        pst.setString(1, pid);
        int i = pst.executeUpdate();
        if(i>0){
            JOptionPane.showMessageDialog(null, "Patient "+ pname+ " Information Deleted
Successfully");
            patient_table();
            AutoID();

```

```

        clear();
    }
    else {
        JOptionPane.showMessageDialog(null," Information not deleted successfully");
    }
} catch (SQLException e) {
    Logger.getLogger(Patients.class.getName()).log(Level.SEVERE, null, e);
    JOptionPane.showMessageDialog(null," Information not deleted successfully");
}
}

// updating the table
private void update(){
    variables();
    if(
        pname.trim().equals("") || pname.trim().equals("Enter Patient Name") ||
        psex.trim().equals("") || pphone.trim().equals("Enter Phone Number") ||
        pphone.trim().equals("") || paddress.trim().equals("") )
    {
        JOptionPane.showMessageDialog(null, "All fields are compulsory "
            +" Kindly check all fields again","Error",JOptionPane.ERROR_MESSAGE);
    }else if (stringDate==null)
    {
        JOptionPane.showMessageDialog(null, "All fields are compulsory "+
            " Choose a valid date","Error",JOptionPane.ERROR_MESSAGE);
    }else
    {
        try {
            Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/"
                + "java_hospital_db?serverTimezone=UTC", "root", "");
            PreparedStatement pst;
            ResultSet rs;
            java.sql.Date sqlDate=new java.sql.Date(jDateChooserDOB.getDate().getTime());
            pst=con.prepareStatement("Update `patientsinfo` set Patient_Name = ?, "
                + "`Patient_Sex` = ?, `Patient_DOB`=?, `Patient_Phone`=?,
                `Patient_Address`=? where Patient_ID=?");
            pst.setString(1, pname);
            pst.setString(2, psex);
            pst.setDate(3, sqlDate);
            pst.setString(4, pphone);
            pst.setString(5, paddress);
            pst.setString(6, pid);
            int i = pst.executeUpdate();
            if(i>0){
                JOptionPane.showMessageDialog(null,"Patient "+ pname+ " Information
Updated Successfully");
                patient_table();
                AutoID();
            }
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null,"Error "+ e.getMessage());
        }
    }
}

```

```

        clear();
    }else {
        JOptionPane.showMessageDialog(null, pname+" 's Information not updated
successfully");
    }
    } catch (SQLException e) {
        Logger.getLogger(Patients.class.getName()).log(Level.SEVERE, null, e);
        JOptionPane.showMessageDialog(null, pname+" 's Information not updated
successfully");
    }
    }
}
//clearing the records
private void clear()
{
    AutoID(); // Automatically generating patient id
    jButtonSave.setEnabled(true);
    jTextFieldName.setText("");
    buttonGroup1.clearSelection();
    jTextFieldContactNumber.setText("");
    jTextAreaAddress.setText("");
    jDateChooserDOB.setDate(null);
}

private void jButtonDeleteMouseClicked(java.awt.event.MouseEvent evt) {
    delete();
}

private void jButtonSaveActionPerformed(java.awt.event.ActionEvent evt) {
    save();
}

private void jButtonClearActionPerformed(java.awt.event.ActionEvent evt) {
    clear();
}

private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
    try {
        DefaultTableModel d1 = (DefaultTableModel)jTable1.getModel();
        int SelectIndex = jTable1.getSelectedRow();
        ID_Label.setText(d1.getValueAt(SelectIndex, 0).toString());
        jTextFieldName.setText(d1.getValueAt(SelectIndex, 1).toString());
        String psex= (d1.getValueAt(SelectIndex, 2).toString());
        if (psex=="Male") {
            jRadioButtonMale.setSelected(true);

```

```

        jRadioButtonFemale.setSelected(false);
    }
    else {
        jRadioButtonMale.setSelected(false);
        jRadioButtonFemale.setSelected(true);

    }
    String dateValue = (d1.getValueAt(SelectIndex, 3).toString());
    java.util.Date date = new SimpleDateFormat("yyyy-MM-dd").parse(dateValue);
    jDateChooserDOB.setDate(date);
    jTextFieldContactNumber.setText(d1.getValueAt(SelectIndex, 4).toString());
    jTextAreaAddress.setText(d1.getValueAt(SelectIndex, 5).toString());
    jButtonSave.setEnabled(false);
} catch (ParseException ex) {
    Logger.getLogger(Registration.class.getName()).log(Level.SEVERE, null, ex);
}

}

private void jButtonUpdateMouseClicked(java.awt.event.MouseEvent evt) {
    update();
}

private void jLabelHome1MouseClicked(java.awt.event.MouseEvent evt) {
    this.dispose();
    new mainForm1().setVisible(true);
}

private void jLabelPatients1MouseClicked(java.awt.event.MouseEvent evt) {
    this.dispose();
    new Registration().setVisible(true);
}

private void jLabelLogs4MouseClicked(java.awt.event.MouseEvent evt) {
    this.dispose();
    new Visit().setVisible(true);
}

private void jLabelLogout1MouseClicked(java.awt.event.MouseEvent evt) {
    this.dispose();
    new LoginPage().setVisible(true);
}

private void jLabelLogs1MouseClicked(java.awt.event.MouseEvent evt) {
    this.dispose();
    new CheckIN().setVisible(true);
}

```

```
private void jLabelLogs6MouseClicked(java.awt.event.MouseEvent evt) {  
    this.dispose();  
    new ServiceSlip().setVisible(true);  
}
```

```
private void jLabelLogs9MouseClicked(java.awt.event.MouseEvent evt) {  
    this.dispose();  
    new Bill().setVisible(true);  
}
```

```
private void jLabelLogs5MouseClicked(java.awt.event.MouseEvent evt) {  
    this.dispose();  
    new CheckOUT().setVisible(true);  
}
```

```
}
```

```
// Variables declaration - do not modify
```

```
private javax.swing.JLabel ID_Label;  
private javax.swing.ButtonGroup buttonGroup1;  
private javax.swing.ButtonGroup buttonGroup2;  
private javax.swing.ButtonGroup buttonGroup3;  
private javax.swing.JButton jButtonClear;  
private javax.swing.JButton jButtonDelete;  
private javax.swing.JButton jButtonSave;  
private javax.swing.JButton jButtonUpdate;  
private com.toedter.calendar.JDateChooser jDateChooserDOB;  
private javax.swing.JLabel jLabel1;  
private javax.swing.JLabel jLabel10;  
private javax.swing.JLabel jLabel11;  
private javax.swing.JLabel jLabel12;  
private javax.swing.JLabel jLabel13;  
private javax.swing.JLabel jLabel14;  
private javax.swing.JLabel jLabel3;  
private javax.swing.JLabel jLabel4;  
private javax.swing.JLabel jLabel9;  
private javax.swing.JLabel jLabelHome1;  
private javax.swing.JLabel jLabelLogout1;  
private javax.swing.JLabel jLabelLogs1;  
private javax.swing.JLabel jLabelLogs4;  
private javax.swing.JLabel jLabelLogs5;  
private javax.swing.JLabel jLabelLogs6;  
private javax.swing.JLabel jLabelLogs9;  
private javax.swing.JLabel jLabelPatients1;  
private javax.swing.JPanel jPanel10;  
private javax.swing.JPanel jPanel11;
```

```

private javax.swing.JPanel jPanel12;
private javax.swing.JPanel jPanel13;
private javax.swing.JPanel jPanel14;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel4;
private javax.swing.JPanel jPanel5;
private javax.swing.JPanel jPanel7;
private javax.swing.JPanel jPanel8;
private javax.swing.JPanel jPanelHeader;
private javax.swing.JPanel jPanelMenu1;
private javax.swing.JRadioButton jRadioButtonFemale;
private javax.swing.JRadioButton jRadioButtonMale;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JSeparator jSeparator10;
private javax.swing.JSeparator jSeparator11;
private javax.swing.JSeparator jSeparator12;
private javax.swing.JSeparator jSeparator19;
private javax.swing.JSeparator jSeparator2;
private javax.swing.JSeparator jSeparator6;
private javax.swing.JSeparator jSeparator7;
private javax.swing.JSeparator jSeparator9;
private javax.swing.JTable jTable1;
private javax.swing.JTextArea jTextAreaAddress;
private javax.swing.JTextField jTextFieldContactNumber;
private javax.swing.JTextField jTextFieldName;
// End of variables declaration
}

```

Visit Class

```

import java.awt.Color;
import java.awt.event.KeyEvent;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.text.SimpleDateFormat;
import java.util.Vector;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.DefaultListModel;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

```



```

public class Visit extends javax.swing.JFrame {

    /**
     * Creates new form Visit
     */
    public Visit() {
        initComponents();
        Disable();
        LoadDoctor();
        LoadOrganization();
        AutoID();
        jButtonok2.setEnabled(false);

    }

    String oo;
    Connection con;
    // initializing several preparedstatement variables
    PreparedStatement pst;
    ResultSet rs;
    String vno;
    PreparedStatement pst1;
    ResultSet rs1;
    PreparedStatement pst2;
    ResultSet rs2;
    PreparedStatement pst3;
    ResultSet rs3;
    public void Connect(){
        try {
            Class.forName("com.mysql.jdbc.Driver");
            con=
DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serverTimez
one=UTC", "root", "");

        }catch (SQLException ex) {
            Logger.getLogger(Patients.class.getName()).log(Level.SEVERE, null, ex);
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(Registration.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    public void Disable()
    {
        jTextFieldDOB.setEnabled(false);
        jTextFieldSex.setEnabled(false);
        jTextFieldContactNumber.setEnabled(false);
        jTextAreaAddress.setEnabled(false);
    }

```

```

public void visit_table(){
    try {
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serverTimez
one=UTC", "root", "");
        SimpleDateFormat dateformat = new SimpleDateFormat("yyyy-MM-dd");
        String date = dateformat.format(jDateChooser.getDate());
        PreparedStatement pst= con.prepareStatement("SELECT * FROM `visit` where
Visit_Date= ?");
        pst.setString(1, date);
        ResultSet rs=pst.executeQuery();
        ResultSetMetaData Rsm= rs.getMetaData();
        int c;
        c=Rsm.getColumnCount();
        DefaultTableModel df = (DefaultTableModel)jTable1.getModel();
        df.setRowCount(0);
        while (rs.next())
        { Vector v2= new Vector();
            for(int i = 1; i<=c; i++)
            {
                v2.add(rs.getString("Visit_No"));
                String pid=rs.getString("Patient_ID");
                pst1= con.prepareStatement("SELECT * FROM `patientsinfo` WHERE Patient_ID=
?");
                pst1.setString(1, pid);
                rs1=pst1.executeQuery();
                if (rs1.next()==true)
                {
                    v2.add(rs1.getString("Patient_Name"));
                }
                String did=rs.getString("Doctor_ID");
                pst2= con.prepareStatement("SELECT * FROM `doctors` WHERE Doctor_ID= ?");
                pst2.setString(1, did);
                rs2=pst2.executeQuery();
                if (rs2.next()==true)
                {
                    v2.add(rs2.getString("Doctor_Name"));
                }
                String oid = rs.getString("Organization_ID");
                pst3= con.prepareStatement("SELECT * FROM organizations where
Organization_ID= ?");
                pst3.setString(1, oid);
                rs3=pst3.executeQuery();
                if (rs3.next()==true)
                {
                    v2.add(rs3.getString("Organisation_Name"));
                }
            }
        }
    }
}

```

```

        v2.add(rs.getString("Visit_Reason"));
    }
    df.addRow(v2);
}
} catch (SQLException ex) {
    Logger.getLogger(Registration.class.getName()).log(Level.SEVERE, null, ex);
}
}

```

```

public void AutoNo(){

    try {
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serverTimez
one=UTC", "root", "");
        SimpleDateFormat dateformat = new SimpleDateFormat("yyyy-MM-dd");
        String date = dateformat.format(jDateChooser.getDate());
        String sql = "SELECT MAX(Visit_No) FROM `visit` where Visit_Date= ?";
        pst=con.prepareStatement(sql);
        pst.setString(1, date);
        rs=pst.executeQuery();
        rs.next();
        rs.getString("MAX(Visit_No)");
        if (rs.getString("MAX(Visit_No)")==null)
        {
            NO_Label.setText("1");
        }else
        {
            int number = rs.getInt("MAX(Visit_No)");
            number++;
            NO_Label.setText(String.valueOf(number));
        }
    }catch (SQLException ex) {
        Logger.getLogger(Patients.class.getName()).log(Level.SEVERE, null, ex);
    }
}

public void AutoID(){

    try {
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serverTimez
one=UTC", "root", "");
        String sql = "SELECT MAX(Visit_ID) FROM `visit`";
        pst=con.prepareStatement(sql);

```

```

rs=pst.executeQuery();
rs.next();
rs.getString("MAX(Visit_ID)");

if (rs.getString("MAX(Visit_ID)")==null)
{
    ID_Label.setText("VS001");
}else
{
    long id =
Long.parseLong(rs.getString("MAX(Visit_ID)").substring(2,rs.getString("MAX(Visit_ID)").length()));
    id ++;
    ID_Label.setText("VS"+String.format("%03d", id));
}
} catch (SQLException ex) {
    Logger.getLogger(Patients.class.getName()).log(Level.SEVERE, null, ex);
}
}

```

```

public class Patient{
    String id;
    String name;

    public Patient (String id, String name)
    {
        this.id = id;
        this.name = name;
    }
    public String toString()
    {
        return name;
    }
}

public class Patient1{
    String id;
    String name;

    public Patient1 (String id, String name)
    {
        this.id = id;
        this.name = name;
    }
    public String toString()
    {
        return id;
    }
}

```

```
}
```

```
String[] idarray = new String [50];
```

```
public void LoadPatient()
```

```
{
```

```
    try {
```

```
        Connection con =
```

```
DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serverTimezone=UTC", "root", "");
```

```
        PreparedStatement pst;
```

```
        ResultSet rs;
```

```
        pst=con.prepareStatement("select * from patientsinfo where Patient_name =?");
```

```
        pst.setString(1, jTextField1.getText());
```

```
        rs= pst.executeQuery();
```

```
        PatientList.removeAll();
```

```
        PatientList.setModel(new DefaultListModel());
```

```
        int x=0;
```

```
        while (rs.next())
```

```
        {
```

```
            DefaultListModel model = (DefaultListModel) PatientList.getModel();
```

```
            model.add(x, "Name:"+rs.getString(2)+" ID:"+rs.getString(1));
```

```
            idarray[x]=rs.getString(1);
```

```
            x=x+1;
```

```
        }
```

```
    } catch (SQLException ex) {
```

```
        Logger.getLogger(Visit.class.getName()).log(Level.SEVERE, null, ex);
```

```
    }
```

```
}
```

```
public class Organization{
```

```
    String id;
```

```
    String name;
```

```
    public Organization (String id, String name)
```

```
    {
```

```
        this.id = id;
```

```
        this.name = name;
```

```
    }
```

```
    public String toString()
```

```
    {
```

```
        return name;
```

```
    }
```

```
}
```

```
public void LoadOrganization()
```

```
{
```

```

    try {
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serverTimez
one=UTC", "root", "");
        PreparedStatement pst;
        ResultSet rs;
        pst=con.prepareStatement("select*from organizations");
        rs= pst.executeQuery();
        organizationname.removeAll();
        while (rs.next())
        {
            organizationname.addItem(rs.getString(2));
        }
    } catch (SQLException ex) {
        Logger.getLogger(Visit.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

```

public class Doctor{
    String id;
    String name;

    public Doctor (String id, String name)
    {
        this.id = id;
        this.name = name;
    }

    public String toString()
    {
        return name;
    }
}

```

```

public void LoadDoctor()
{
    try {
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serverTimez
one=UTC", "root", "");
        PreparedStatement pst;
        ResultSet rs;
        pst=con.prepareStatement("select*from doctors");
        rs= pst.executeQuery();
        doctorname.removeAll();
        while (rs.next())
        {

```

```

        doctorname.addItem(new Doctor(rs.getString(1),rs.getString(2)));
    }
} catch (SQLException ex) {
    Logger.getLogger(Visit.class.getName()).log(Level.SEVERE, null, ex);
}
}

private void jDateChooserMouseClicked(java.awt.event.MouseEvent evt) {
    jButtonok1.setEnabled(true);
}

private void jLabelHome1MouseClicked(java.awt.event.MouseEvent evt) {
    this.dispose();
    new mainForm1().setVisible(true);
}

private void jLabelPatients1MouseClicked(java.awt.event.MouseEvent evt) {
    this.dispose();
    new Registration().setVisible(true);
}

private void jLabelLogs4MouseClicked(java.awt.event.MouseEvent evt) {
    this.dispose();
    new Visit().setVisible(true);
}

private void jLabelLogout1MouseClicked(java.awt.event.MouseEvent evt) {
    this.dispose();
    new LoginPage().setVisible(true);
}

private void jLabelLogs1MouseClicked(java.awt.event.MouseEvent evt) {
    this.dispose();
    new CheckIN().setVisible(true);
}

private void jLabelLogs6MouseClicked(java.awt.event.MouseEvent evt) {
    this.dispose();
    new ServiceSlip().setVisible(true);
}

private void jLabelLogs9MouseClicked(java.awt.event.MouseEvent evt) {
    this.dispose();
    new Bill().setVisible(true);
}

private void jLabelLogs5MouseClicked(java.awt.event.MouseEvent evt) {

```

```

        this.dispose();
        new CheckOUT().setVisible(true);
    }

    private void jTextField1KeyPressed(java.awt.event.KeyEvent evt) {
        if(evt.getKeyCode() == KeyEvent.VK_ENTER)
        {
            LoadPatient();
            jButtonok2.setEnabled(true);
        }
    }
}

```

Checkin Class

```

import java.awt.Color;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.text.SimpleDateFormat;
import java.util.Vector;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

public class CheckIN extends javax.swing.JFrame {

    /**
     * Creates new form CheckIN
     */
    public CheckIN() {
        initComponents();
        Disable();
        indoor_table();
    }

    public void Disable()
    {
        jButtonSave.setEnabled(false);
        jTextFieldDOB.setEnabled(false);
        jTextFieldSex.setEnabled(false);
        jTextFieldContactNumber.setEnabled(false);
        jTextFieldAddress.setEnabled(false);
        jTextFieldReason.setEnabled(false);
        jTextFieldDoctor.setEnabled(false);
    }
}

```



```

jTextFieldOrganization.setEnabled(false);

}

public class Patient{
    String id;
    String name;
    public Patient (String id, String name)
    {
        this.id = id;
        this.name = name;
    }
    public String toString()
    {
        return name;
    }
}

public void LoadPatient()
{
    try {
        Connection con =
java.sql.DriverManager.getConnection("jdbc:mysql://localhost:3306/"
        + "java_hospital_db?serverTimezone=UTC", "root", "");
        PreparedStatement pst;
        ResultSet rs;
        SimpleDateFormat dateformat = new SimpleDateFormat("yyyy-MM-dd");
        String date = dateformat.format(jDateChooser.getDate());
        pst=con.prepareStatement("select * from visit where Visit_Date=?");
        pst.setString(1, date);
        rs= pst.executeQuery();
        PatientName.removeAllItems();
        while (rs.next())
        {
            String pid= rs.getString("Patient_ID");
            PreparedStatement pst1;
            ResultSet rs1;
            pst1= con.prepareStatement("SELECT * FROM `patientsinfo` WHERE Patient_ID=
?");
            pst1.setString(1, pid);
            rs1=pst1.executeQuery();
            if (rs1.next()==true)
            {
                PatientName.addItem(new Patient(rs.getString(1),rs1.getString(2)));
            }
        }
    } catch (SQLException ex) {
        Logger.getLogger(Visit.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

```

    }
}
public class Room{
    String id;
    String name;

    public Room (String id, String name)
    {
        this.id = id;
        this.name = name;
    }

    public String toString()
    {
        return id;
    }

}

public void LoadRoom()
{

    try {
        Connection con =
java.sql.DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serv
erTimezone=UTC", "root", "");
        PreparedStatement pst=con.prepareStatement("select * from rooms where
Room_Status=?");
        pst.setString(1, "Vacant");
        ResultSet rs= pst.executeQuery();
        Roombox.removeAllItems();
        while (rs.next())
        {
            Roombox.addItem(new Room(rs.getString(1),rs.getString(2)));
        }

    } catch (SQLException ex) {
        Logger.getLogger(Visit.class.getName()).log(Level.SEVERE, null, ex);
    }
}

public void indoor_table(){
    try {
        Connection con =
java.sql.DriverManager.getConnection("jdbc:mysql://localhost:3306/"
+ "java_hospital_db?serverTimezone=UTC", "root", "");
    }
}

```

```

PreparedStatement pst;
ResultSet rs;
pst=con.prepareStatement("Select* from rooms");
rs=pst.executeQuery();
ResultSetMetaData Rsm= rs.getMetaData();
int c;
c=Rsm.getColumnCount();
DefaultTableModel df = (DefaultTableModel)jTable1.getModel();
df.setRowCount(0);
while (rs.next())
{
    Vector v2= new Vector();
    for(int i = 1; i<=c; i++)
    {
        v2.add(rs.getString("Room_ID"));
        v2.add(rs.getString("Room_Type"));

        String Status = rs.getString("Room_Status");
        if (Status=="Vacant")
        {
            v2.add("No Patient");
        }else
        {
            PreparedStatement pst1;
            ResultSet rs1;
            pst1=con.prepareStatement("SELECT * FROM `indoor` WHERE Room_ID=? AND
Checkout_Date=?");
            pst1.setString(1, rs.getString("Room_ID"));
            pst1.setString(2, "Pending");
            rs1=pst1.executeQuery();
            if (rs1.next()==false)
            {
                v2.add("No Patient");
            }else
            {
                String pid= rs1.getString("Visit_ID");

                PreparedStatement pst2;
                ResultSet rs2;
                pst2=con.prepareStatement("SELECT * FROM `visit` WHERE Visit_ID=?");
                pst2.setString(1, pid);
                rs2=pst2.executeQuery();
                rs2.next();
                String ppid = rs2.getString("Patient_ID");
                PreparedStatement pst3;
                ResultSet rs3;
                pst3=con.prepareStatement("Select* from patientsinfo where Patient_ID=?");

```

```

        pst3.setString(1, ppid);
        rs3=pst3.executeQuery();
        rs3.next();
        v2.add(rs3.getString("Patient_Name"));
    }
}
df.addRow(v2);
}
}
} catch (SQLException ex) {
    Logger.getLogger(CheckIN.class.getName()).log(Level.SEVERE, null, ex);
}
}
}

```

```

private void jButtonSaveActionPerformed(java.awt.event.ActionEvent evt) {

```

```

    try {
        Connection con =
java.sql.DriverManager.getConnection("jdbc:mysql://localhost:3306/"
        + "java_hospital_db?serverTimezone=UTC", "root", "");
        PreparedStatement pst;
        ResultSet rs;
        PreparedStatement pst1;
        ResultSet rs1;
        SimpleDateFormat dateformat = new SimpleDateFormat("yyyy-MM-dd");
        String date = dateformat.format(jDateChooser.getDate());
        Room r = (Room)Roombox.getSelectedItem();
        String sql = "SELECT MAX(Indoor_ID) FROM `indoor`";
        pst1=con.prepareStatement(sql);
        rs1=pst1.executeQuery();
        rs1.next();
        rs1.getString("MAX(Indoor_ID)");
        String IndoorID;
        if (rs1.getString("MAX(Indoor_ID)")==null)
        {
            IndoorID ="1";
        }
        else
        {
            int number = rs1.getInt("MAX(Indoor_ID)");
            number++;
            IndoorID= String.valueOf(number);
        }
        pst=con.prepareStatement("INSERT INTO `indoor`(`Indoor_ID`, `Visit_ID`, "
        + "Checkin_Date,Checkout_Date, Room_ID) VALUES (?, ?, ?, ?, ?)");
        pst.setString(1,IndoorID);
        pst.setString(2,ID_Label1.getText());
    }
}

```

```

        pst.setString(3, date);
        pst.setString(4, "Pending");
        pst.setString(5, r.id);
        int i = pst.executeUpdate();
        if(i>0){
            JOptionPane.showMessageDialog(null, "Check-In Successful");
            ID_Label1.setText("___");
            jTextFieldReason.setText("");
            jTextFieldDOB.setText("");
            jTextFieldSex.setText("");
            jTextFieldContactNumber.setText("");
            jTextAreaAddress.setText("");
            jTextFieldDoctor.setText("");
            jTextFieldOrganization.setText("");
            PatientName.setSelectedIndex(-1);
            Roombox.setSelectedIndex(-1);
            PreparedStatement pst2;
            ResultSet rs2;
            pst2=con.prepareStatement("UPDATE `rooms` SET `Room_Status`= ? WHERE
Room_ID=?");
            pst2.setString(1, "Active");
            pst2.setString(2, r.id);
            int j = pst2.executeUpdate();
            if(j>0){
                indoor_table();
            }
        }
        else {
            JOptionPane.showMessageDialog(null, "Information not saved successfully");
        }
    } catch (SQLException e) {
        Logger.getLogger(Patients.class.getName()).log(Level.SEVERE, null, e);
        JOptionPane.showMessageDialog(null, " 's Information not saved successfully");
    }
}

```

```

}

```

```

private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {

```

```

}

```

```

private void jButton2MouseClicked(java.awt.event.MouseEvent evt) {

```

```
}
```

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    jButtonSave.setEnabled(true);  
    LoadRoom();  
    try{  
        PreparedStatement pst;  
        ResultSet rs;  
        Patient p = (Patient)PatientName.getSelectedItem();  
        Connection con =  
java.sql.DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serv  
erTimezone=UTC", "root", "");  
        pst= con.prepareStatement("SELECT * FROM `patientsinfo` WHERE Patient_Name  
=?");  
        pst.setString(1, p.name);  
        rs=pst.executeQuery();  
        rs.next();  
        String pid=rs.getString("Patient_ID");  
        PreparedStatement pst1;  
        ResultSet rs1;  
        SimpleDateFormat dateformat = new SimpleDateFormat("yyyy-MM-dd");  
        String date = dateformat.format(jDateChooser.getDate());  
        pst1= con.prepareStatement("SELECT * FROM `visit` WHERE Patient_ID =? AND  
Visit_Date=?");  
        pst1.setString(1, pid);  
        pst1.setString(2, date);  
        rs1=pst1.executeQuery();  
        if(rs1.next()==true)  
        {  
            ID_Label1.setText(rs1.getString("Visit_ID"));  
        }  
        jTextFieldSex.setText(rs.getString("Patient_Sex"));  
        jTextFieldDOB.setText(rs.getString("Patient_DOB"));  
        jTextFieldContactNumber.setText(rs.getString("Patient_Phone"));  
        jTextAreaAddress.setText(rs.getString("Patient_Address"));  
        String did=rs1.getString("Doctor_ID");  
        PreparedStatement pst2;  
        ResultSet rs2;  
        pst2= con.prepareStatement("SELECT * FROM `doctors` WHERE Doctor_ID =?");  
        pst2.setString(1, did);  
        rs2=pst2.executeQuery();  
        rs2.next();  
        jTextFieldDoctor.setText(rs2.getString("Doctor_Name"));  
        String oid=rs1.getString("Organization_ID");
```

```

        PreparedStatement pst3;
        ResultSet rs3;
        pst3= con.prepareStatement("SELECT * FROM `organizations` WHERE
Organization_ID =?");
        pst3.setString(1, oid);
        rs3=pst3.executeQuery();
        rs3.next();
        jTextFieldOrganization.setText(rs3.getString("Organisation_Name"));
        jTextFieldReason.setText(rs1.getString("Visit_Reason"));
    } catch (SQLException ex) {
        Logger.getLogger(Visit.class.getName()).log(Level.SEVERE, null, ex);
    }
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    java.util.Date stringDate=jDateChooser.getDate();
    if (stringDate==null)
    {
        JOptionPane.showMessageDialog(null, "Choose a valid
date","Error",JOptionPane.ERROR_MESSAGE);
    }
    else{
        LoadPatient();
    }

}

```

ServiceSlip Class

```

import java.awt.Color;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;

/**
 *
 * @author
 */
public class ServiceSlip extends javax.swing.JFrame {

```

```

/**
 * Creates new form ServiceSlip
 */
public ServiceSlip() {
    initComponents();
    disable();
}

Connection con;
public void Connect(){
    try {
        Class.forName("com.mysql.jdbc.Driver");
        con=
java.sql.DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serv
erTimezone=UTC", "root", "");

    }

    catch (SQLException ex) {
        Logger.getLogger(Patients.class.getName()).log(Level.SEVERE, null, ex);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(Registration.class.getName()).log(Level.SEVERE, null, ex);
    }
}

public void disable(){
    jButtonopd.setEnabled(false);
    jButtonindoor.setEnabled(false);
    jTextFieldDOB.setEnabled(false);
    jTextFieldContactNumber.setEnabled(false);
    jTextFieldSex.setEnabled(false);
    jTextFieldAddress.setEnabled(false);
    jTextFieldDoctor.setEnabled(false);
    jTextFieldRoom.setVisible(false);
    jLabel21.setVisible(false);

}

public class Service{
    String id;
    String name;

    public Service (String id, String name)
    {
        this.id = id;
        this.name = name;
    }
}

```



```

        public String toString()
        {
            return name;
        }

    }

    public void LoadService()
    {
        try {
            Connection con =
java.sql.DriverManager.getConnection("jdbc:mysql://localhost:3306/"
            + "java_hospital_db?serverTimezone=UTC", "root", "");
            PreparedStatement pst;
            ResultSet rs;

            pst=con.prepareStatement("select * from servicesinfo");
            rs= pst.executeQuery();
            ServiceBox.removeAllItems();
            while (rs.next())
            {
                ServiceBox.addItem(rs.getString(2));
            }
        } catch (SQLException ex) {
            Logger.getLogger(Visit.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    public void LoadService1()
    {
        try {
            Connection con =
java.sql.DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serv
erTimezone=UTC", "root", "");
            PreparedStatement pst;
            ResultSet rs;
            pst=con.prepareStatement("select * from servicesinfo where Service_Type= ?");
            pst.setString(1, "General");
            rs= pst.executeQuery();
            ServiceBox.removeAllItems();
            while (rs.next())
            {
                ServiceBox.addItem(rs.getString(2));
            }
        } catch (SQLException ex) {
            Logger.getLogger(Visit.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

```

```

}
public class Patient{
    String id;
    String name;

    public Patient (String id, String name)
    {
        this.id = id;
        this.name = name;
    }

    public String toString()
    {
        return name;
    }

}

public void LoadPatient1()
{
    try {
        Connection con =
java.sql.DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serv
erTimezone=UTC", "root", "");
        PreparedStatement pst;
        ResultSet rs;
        SimpleDateFormat dateformat = new SimpleDateFormat("yyyy-MM-dd");
        String date = dateformat.format(jDateChooser.getDate());
        pst=con.prepareStatement("select * from indoor where Checkout_Date=?");
        pst.setString(1, "Pending");
        rs= pst.executeQuery();
        PatientName.removeAllItems();
        while (rs.next())
        {
            try {
                String date1 = rs.getString("Checkin_Date");
                Date d3 = dateformat.parse(date);
                Date d4 = dateformat.parse(date1);
                if (d3.compareTo(d4)>=0){
                    PreparedStatement pst2 = con.prepareStatement("select * from visit where
Visit_ID=?");
                    pst2.setString(1, rs.getString("Visit_ID"));
                    ResultSet rs2= pst2.executeQuery();
                    rs2.next();

                    PreparedStatement pst3 = con.prepareStatement("select * from patientsinfo
where Patient_ID=?");
                    pst3.setString(1, rs2.getString("Patient_ID"));

```

```

        ResultSet rs3= pst3.executeQuery();
        rs3.next();
        PatientName.addItem(new
Patient(rs3.getString("Patient_ID"),rs3.getString("Patient_Name")));
    }
    } catch (ParseException ex) {
        Logger.getLogger(ServiceSlip.class.getName()).log(Level.SEVERE, null, ex);
    }
}
} catch (SQLException ex) {
    Logger.getLogger(Visit.class.getName()).log(Level.SEVERE, null, ex);
}
}

public void LoadPatient()
{
    try {
        Connection con =
java.sql.DriverManager.getConnection("jdbc:mysql://localhost:3306/"
+ "java_hospital_db?serverTimezone=UTC", "root", "");
        PreparedStatement pst;
        ResultSet rs;
        SimpleDateFormat dateformat = new SimpleDateFormat("yyyy-MM-dd");
        String date = dateformat.format(jDateChooser.getDate());
        pst=con.prepareStatement("select * from visit where Visit_Date=?");
        pst.setString(1, date);
        rs= pst.executeQuery();
        PatientName.removeAllItems();
        while (rs.next())
        {
            String vid=rs.getString("Visit_ID");
            PreparedStatement pst2 = con.prepareStatement("Select * from indoor where
Visit_ID =? ");
            pst2.setString(1,vid);
            ResultSet rs2=pst2.executeQuery();
            rs2.next();
            if (rs2.next()==false)
            {
                String pid= rs.getString("Patient_ID");
                PreparedStatement pst1;
                ResultSet rs1;
                pst1= con.prepareStatement("SELECT * FROM `patientsinfo` WHERE Patient_ID=
?");
                pst1.setString(1, pid);
                rs1=pst1.executeQuery();
                if (rs1.next()==true)
                {

```

```

        PatientName.addItem(new Patient(rs.getString(1),rs1.getString(2)));
    }
}
} catch (SQLException ex) {
    Logger.getLogger(Visit.class.getName()).log(Level.SEVERE, null, ex);
}
}private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

    java.util.Date stringDate=jDateChooser.getDate();
    if (stringDate==null)
    {
        JOptionPane.showMessageDialog(null, "Choose a valid
date","Error",JOptionPane.ERROR_MESSAGE);

    }
    else{
        jButtonopd.setEnabled(true);
        jButtonindoor.setEnabled(true);
        jLabel12.setForeground(Color.black);
        jLabel13.setForeground(Color.black);
        jLabel21.setForeground(Color.black);
        jLabel9.setForeground(Color.black);
        jLabel22.setForeground(Color.black);
        jLabel20.setForeground(Color.black);
        jLabel18.setForeground(Color.black);
        jLabel19.setForeground(Color.black);
    }

}

private void jButtonopdMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
}

private void jTextFieldDOBActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void jTextFieldSexActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void jTextFieldDoctorActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

```

```

private void jTextFieldRoomActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void jButtonOpdActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    LoadPatient();
    LoadService1();
    jButtonOpd.setEnabled(false);
    jButtonIndoor.setEnabled(false);
}

private void ServiceBoxActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void jButtonIndoorActionPerformed(java.awt.event.ActionEvent evt) {
    LoadService();
    LoadPatient1();
    jLabel21.setVisible(true);
    jTextFieldRoom.setVisible(true);
    jTextFieldRoom.setEnabled(false);
    jButtonOpd.setEnabled(false);
    jButtonIndoor.setEnabled(false);
}

private void jButtonSaveActionPerformed(java.awt.event.ActionEvent evt) {
try {
    Connection con =
java.sql.DriverManager.getConnection("jdbc:mysql://localhost:3306/"
        + "java_hospital_db?serverTimezone=UTC", "root", "");
    PreparedStatement pst;
    ResultSet rs;
    PreparedStatement pst1;
    ResultSet rs1;
    SimpleDateFormat dateformat = new SimpleDateFormat("yyyy-MM-dd");
    String date = dateformat.format(jDateChooser.getDate());
    Service s = (Service)ServiceBox.getSelectedItem();
    String sql = "SELECT MAX(Patientservice_ID) FROM `patientservice`";
    pst1=con.prepareStatement(sql);
    rs1=pst1.executeQuery();
    rs1.next();
    rs1.getString("MAX(Patientservice_ID)");
    String PatientserviceID;
    if (rs1.getString("MAX(Patientservice_ID)")==null)
    {

```

```

        PatientserviceID ="1";
    }
else
{
    int number = rs1.getInt("MAX(Patientservice_ID)");
    number++;
    PatientserviceID= String.valueOf(number);
}

pst=con.prepareStatement("INSERT INTO `patientservice`(`Patientservice_ID`, "
    + "`Visit_ID`, Service_Date, Service_ID) VALUES (?, ?, ?, ?)");
pst.setString(1,PatientserviceID);
pst.setString(2,ID_Label1.getText());
pst.setString(3, date);
pst.setString(4, s.id);
int i = pst.executeUpdate();
if(i>0){
    JOptionPane.showMessageDialog(null,"Service Alotted Successfully");
    ID_Label1.setText("___");
    jTextFieldDoctor.setText("");
    jTextFieldDOB.setText("");
    jTextFieldSex.setText("");
    jTextFieldContactNumber.setText("");
    jTextAreaAddress.setText("");
    jTextFieldRoom.setText("");
    PatientName.setSelectedIndex(-1);
    ServiceBox.setSelectedIndex(-1);
    disable();

}
else {
    JOptionPane.showMessageDialog(null, "Information not saved successfully");
}
} catch (SQLException e) {
    Logger.getLogger(Patients.class.getName()).log(Level.SEVERE, null, e);
    JOptionPane.showMessageDialog(null, " 's Information not saved successfully");
}

}

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
    try {

```

```

        Connection con =
java.sql.DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serv
erTimezone=UTC", "root", "");
        PreparedStatement pst2 = con.prepareStatement("select * from patientsinfo where
Patient_Name=?");
        pst2.setString(1, PatientName.getSelectedItemAt().toString());
        ResultSet rs2= pst2.executeQuery();
        while (rs2.next()){
            String pid = rs2.getString("Patient_ID");
            SimpleDateFormat dateformat = new SimpleDateFormat("yyyy-MM-dd");
            String date = dateformat.format(jDateChooser.getDate());
            PreparedStatement pst;
            pst= con.prepareStatement("SELECT * FROM `visit` WHERE Patient_ID=?");
            pst.setString(1, pid);
            ResultSet rs=pst.executeQuery();
            while (rs.next()==true)
            {
                PreparedStatement pst3 = con.prepareStatement("SELECT * FROM `indoor` WHERE
Visit_ID=? AND Checkout_Date=?");
                pst3.setString(1, rs.getString("Visit_ID"));
                pst3.setString(2, "Pending");
                ResultSet rs3 = pst3.executeQuery();
                if (rs3.next()==true)
                {
                    ID_Label1.setText(rs3.getString("Visit_ID"));
                    jTextFieldRoom.setText(rs3.getString("Room_ID"));
                    PreparedStatement pst5 = con.prepareStatement("Select * from visit where
Visit_ID=?");
                    pst5.setString(1,rs3.getString("Visit_ID") );
                    ResultSet rs5 = pst5.executeQuery();
                    rs5.next();
                    if ("1".equals(rs5.getString("Doctor_ID")))
                    {
                        jTextFieldDoctor.setText("Dr. Ajay Gupta");
                    }
                    else if ("2".equals(rs5.getString("Doctor_ID")))
                    {
                        jTextFieldDoctor.setText("Dr. Amita Gupta");
                    }
                    else if ("3".equals(rs5.getString("Doctor_ID")))
                    {
                        jTextFieldDoctor.setText("Dr. Pradeep Tinna");
                    }
                    PreparedStatement pst6 = con.prepareStatement("Select * from patientsinfo where
Patient_ID=?");
                    pst6.setString(1,rs5.getString("Patient_ID") );

```

```

        ResultSet rs6 = pst6.executeQuery();
        rs6.next();

        jTextFieldDOB.setText(rs6.getString("Patient_DOB"));
        jTextFieldSex.setText(rs6.getString("Patient_Sex"));
        jTextFieldContactNumber.setText(rs6.getString("Patient_Phone"));
        jTextAreaAddress.setText(rs6.getString("Patient_Address"));
    }
    else if (rs3.next()==false)
    {

        PreparedStatement pst4= con.prepareStatement("SELECT * FROM `visit` WHERE
Patient_ID=? AND Visit_Date=?");
        pst4.setString(1, pid);
        pst4.setString(2,date);
        ResultSet rs4=pst4.executeQuery();
        rs4.next();
        ID_Label1.setText(rs4.getString("Visit_ID"));

        if ("1".equals(rs4.getString("Doctor_ID")))
        {
            jTextFieldDoctor.setText("Dr. Ajay Gupta");
        }
        else if ("2".equals(rs4.getString("Doctor_ID")))
        {
            jTextFieldDoctor.setText("Dr. Amita Gupta");
        }
        else if ("3".equals(rs4.getString("Doctor_ID")))
        {
            jTextFieldDoctor.setText("Dr. Pradeep Tinna");
        }

        PreparedStatement pst6 = con.prepareStatement("Select * from patientsinfo where
Patient_ID=?");
        pst6.setString(1,rs4.getString("Patient_ID") );
        ResultSet rs6 = pst6.executeQuery();
        rs6.next();

        jTextFieldDOB.setText(rs6.getString("Patient_DOB"));
        jTextFieldSex.setText(rs6.getString("Patient_Sex"));
        jTextFieldContactNumber.setText(rs6.getString("Patient_Phone"));
        jTextAreaAddress.setText(rs6.getString("Patient_Address"));

    }

}
}
}

```



```

    } catch (SQLException ex) {
        Logger.getLogger(ServiceSlip.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

Checkout Class

```

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Vector;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author
 */
public class CheckOUT extends javax.swing.JFrame {

    /**
     * Creates new form CheckOUT
     */
    public CheckOUT() {
        initComponents();
        indoor_table();
        Disable();
    }

    public void Disable(){
        jButtonSave.setEnabled(false);
    }

    public void indoor_table(){
        try {
            Connection con =
java.sql.DriverManager.getConnection("jdbc:mysql://localhost:3306/"
            + "java_hospital_db?serverTimezone=UTC", "root", "");
            PreparedStatement pst;
            ResultSet rs;

```

```

pst=con.prepareStatement("Select* from rooms where Room_Status=?");
pst.setString(1, "Active");
rs=pst.executeQuery();
ResultSetMetaData Rsm= rs.getMetaData();
int c;
c=Rsm.getColumnCount();
DefaultTableModel df = (DefaultTableModel)jTable4.getModel();
df.setRowCount(0);
while (rs.next())
{

    Vector v2= new Vector();
    for(int i = 1; i<=c; i++)
    {
        v2.add(rs.getString("Room_ID"));
        v2.add(rs.getString("Room_Type"));
        String Status = rs.getString("Room_Status");
        if (Status=="Vacant")
        {
            df.removeRow(i);
        }
        else
        {
            PreparedStatement pst1;
            ResultSet rs1;
            String pid = rs.getString("Room_ID");
            pst1=con.prepareStatement("SELECT * FROM `indoor` WHERE Room_ID=? AND
Checkout_Date=?");
            pst1.setString(1, pid);
            pst1.setString(2, "Pending");
            rs1=pst1.executeQuery();
            rs1.next();
            String ppid= rs1.getString("Visit_ID");
            PreparedStatement pst2;
            ResultSet rs2;
            pst2=con.prepareStatement("SELECT * FROM `visit` WHERE Visit_ID=?");
            pst2.setString(1, ppid);
            rs2=pst2.executeQuery();
            rs2.next();
            String pppid = rs2.getString("Patient_ID");
            PreparedStatement pst3;
            ResultSet rs3;
            pst3=con.prepareStatement("Select* from patientsinfo where Patient_ID=?");
            pst3.setString(1, pppid);
            rs3=pst3.executeQuery();
            rs3.next();
            v2.add(rs3.getString("Patient_Name"));

```

```

        }df.addRow(v2);
    }
}
} catch (SQLException ex) {
    Logger.getLogger(CheckIN.class.getName()).log(Level.SEVERE, null, ex);
}
}
}

```

```

private void jButtonSaveActionPerformed(java.awt.event.ActionEvent evt) {

    try {
        Connection con =
java.sql.DriverManager.getConnection("jdbc:mysql://localhost:3306/"
            + "java_hospital_db?serverTimezone=UTC", "root", "");
        PreparedStatement pst;
        ResultSet rs;
        PreparedStatement pst1;
        ResultSet rs1;
        SimpleDateFormat dateformat = new SimpleDateFormat("yyyy-MM-dd");
        String date = dateformat.format(jDateChooser.getDate());
        DefaultTableModel d1 = (DefaultTableModel)jTable4.getModel();
        int SelectIndex = jTable4.getSelectedRow();
        if (jTable4.getSelectedRow() == -1)
        {
            JOptionPane.showMessageDialog(null, "Chose a date and a room", "Error", HEIGHT);
        }
        else{
            String rid = d1.getValueAt(SelectIndex, 0).toString();
            String sql= "Select * from indoor where Room_ID=? AND Checkout_Date=?";
            PreparedStatement pst2=con.prepareStatement(sql);
            pst2.setString(1, rid);
            pst2.setString(2, "Pending");
            ResultSet rs2=pst2.executeQuery();
            rs2.next();
            try {
                Date d3 = dateformat.parse(date);
                Date d4 = dateformat.parse(rs2.getString("Checkin_Date"));
                if (d3.compareTo(d4) >= 0){
                    pst1=con.prepareStatement("Update indoor set Checkout_Date= ? where
Room_ID=? and Checkout_Date=?");
                    pst1.setString(1, date);
                    pst1.setString(2, rid);
                    pst1.setString(3, "Pending");
                    int i = pst1.executeUpdate();
                    pst=con.prepareStatement("UPDATE `rooms` SET `Room_Status`=? WHERE
`Room_ID`=?");
                    pst.setString(1, "Vacant");

```

```

        pst.setString(2, rid);
        int j = pst.executeUpdate();
        indoor_table();
        JOptionPane.showMessageDialog(null, "Check-out Successful");
        DOB.setText("");
        Phone.setText("");
        Sex.setText("");
        Address.setText("");
        Doctor.setText("");
        Org.setText("");
        Reason.setText("");
        ID_Label12.setText("");
        ID_Label.setText("");
        jLabel4.setText("");
    }
    else {
        JOptionPane.showMessageDialog(null, "Select a date after Check-in Date", "Error",
HEIGHT);
    }
}
}
    catch (ParseException ex) {
        Logger.getLogger(CheckOUT.class.getName()).log(Level.SEVERE, null, ex);
    }
}

} catch (SQLException e) {
    Logger.getLogger(Patients.class.getName()).log(Level.SEVERE, null, e);
    JOptionPane.showMessageDialog(null, "'s Information not saved successfully");
}
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    java.util.Date stringDate=jDateChooser.getDate();
    if (stringDate==null)
    {
        JOptionPane.showMessageDialog(null, "Choose a valid
date", "Error",JOptionPane.ERROR_MESSAGE);
    }else
    {
        jButtonSave.setEnabled(true);
    }
}

private void jTable4MouseClicked(java.awt.event.MouseEvent evt) {

    try {
        Connection con =
java.sql.DriverManager.getConnection("jdbc:mysql://localhost:3306/"
+ "java_hospital_db?serverTimezone=UTC", "root", "");

```

```

PreparedStatement pst;
ResultSet rs;
DefaultTableModel d1 = (DefaultTableModel)jTable4.getModel();
int SelectIndex = jTable4.getSelectedRow();
String name= d1.getValueAt(SelectIndex, 2).toString();
String id = d1.getValueAt(SelectIndex, 0).toString();
jLabel4.setText(name);
String sql = "Select Visit_ID from indoor where Room_ID=? AND Checkout_Date=?";
pst= con.prepareStatement(sql);
pst.setString(1, id);
pst.setString(2, "Pending");
rs=pst.executeQuery();
rs.next();
String vid= rs.getString("Visit_ID");
ID_Label12.setText(vid);

```

```

PreparedStatement pst1 = con.prepareStatement( "SELECT * FROM `visit` WHERE
`Visit_ID`=?");
pst1.setString(1, vid);
ResultSet rs1=pst1.executeQuery();
rs1.next();
String pid= rs1.getString("Patient_ID");
String did= rs1.getString("Doctor_ID");
String oid= rs1.getString("Organization_ID");
Reason.setText(rs1.getString("Visit_Reason"));
String sql2 = "Select * from patientsinfo where Patient_ID=?";

```

```

PreparedStatement pst2= con.prepareStatement(sql2);
pst2.setString(1, pid);
ResultSet rs2 = pst2.executeQuery();
rs2.next();
String sql3 = "Select * from doctors where Doctor_ID=?";

```

```

PreparedStatement pst3= con.prepareStatement(sql3);
pst3.setString(1, did);
ResultSet rs3 = pst3.executeQuery();
rs3.next();
String sql4 = "Select * from organizations where Organization_ID=?";

```

```

PreparedStatement pst4= con.prepareStatement(sql4);
pst4.setString(1, oid);
ResultSet rs4 = pst4.executeQuery();
rs4.next();
DOB.setText(rs2.getString("Patient_DOB"));
Phone.setText(rs2.getString("Patient_Phone"));
Sex.setText(rs2.getString("Patient_Sex"));
Address.setText(rs2.getString("Patient_Address"));

```

```

        Doctor.setText(rs3.getString("Doctor_Name"));
        Org.setText(rs4.getString("Organisation_Name"));
    } catch (SQLException ex) {
        Logger.getLogger(CheckOUT.class.getName()).log(Level.SEVERE, null, ex);
    }
}
}
}

```

Bill Class

```

import java.awt.print.PrinterException;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.time.temporal.ChronoUnit;
import java.util.Date;
import java.util.Vector;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

public class Bill extends javax.swing.JFrame {
    /**
     * Creates new form Bill
     */
    public Bill() {
        initComponents();
        AutoID();
    }

    Connection con;
    PreparedStatement pst;
    ResultSet rs;

    public void Connect(){
        try {
            Class.forName("com.mysql.jdbc.Driver");

```

```

        con=
java.sql.DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serv
erTimezone=UTC", "root", "");
    }
    catch (SQLException ex) {
        Logger.getLogger(Patients.class.getName()).log(Level.SEVERE, null, ex);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(Registration.class.getName()).log(Level.SEVERE, null, ex);
    }
}

public class Patient{
    String id;
    String name;

    public Patient (String id, String name)
    {
        this.id = id;
        this.name = name;
    }
    public String toString()
    {
        return name;
    }
}

public void LoadPatient1()
{
    try {
        Connect();
        SimpleDateFormat dateformat = new SimpleDateFormat("yyyy-MM-dd");
        String date = dateformat.format(jDateChooser.getDate());
        pst=con.prepareStatement("select * from indoor where Checkout_Date=?");
        pst.setString(1, date);
        rs= pst.executeQuery();
        PatientName.removeAllItems();
        while (rs.next())
        {
            try {
                String date1 = rs.getString("Checkin_Date");
                Date d3 = dateformat.parse(date);
                Date d4 = dateformat.parse(date1);
                if (d3.compareTo(d4)>=0){
                    PreparedStatement pq=con.prepareStatement("Select*from payments where
Visit_ID=?");
                    pq.setString(1, rs.getString("Visit_ID"));
                    ResultSet rr = pq.executeQuery();
                    if (rr.next()==false){

```

```

        PreparedStatement pst2 = con.prepareStatement("select * from visit where
Visit_ID=?");
        pst2.setString(1, rs.getString("Visit_ID"));
        ResultSet rs2= pst2.executeQuery();
        rs2.next();
        PreparedStatement pst3 = con.prepareStatement("select * from patientsinfo
where Patient_ID=?");
        pst3.setString(1, rs2.getString("Patient_ID"));
        ResultSet rs3= pst3.executeQuery();
        rs3.next();
        PatientName.addItem(new
Patient(rs3.getString("Patient_ID"),rs3.getString("Patient_Name")));
    }
}
} catch (ParseException ex) {
    Logger.getLogger(ServiceSlip.class.getName()).log(Level.SEVERE, null, ex);
}
}
}
catch (SQLException ex) {
    Logger.getLogger(Visit.class.getName()).log(Level.SEVERE, null, ex);
}
}
}
public void LoadPatient()
{
    try {
        Connect();
        SimpleDateFormat dateformat = new SimpleDateFormat("yyyy-MM-dd");
        String date = dateformat.format(jDateChooser.getDate());
        pst=con.prepareStatement("select * from visit where Visit_Date=?");
        pst.setString(1, date);
        rs= pst.executeQuery();
        PatientName.removeAllItems();
        while (rs.next())
        {
            PreparedStatement pq=con.prepareStatement("Select*from payments where
Visit_ID=?");
            pq.setString(1, rs.getString("Visit_ID"));
            ResultSet rr = pq.executeQuery();
            if (rr.next()==false){
                String vid=rs.getString("Visit_ID");
                PreparedStatement pst2 = con.prepareStatement("Select * from indoor where
Visit_ID =? ");
                pst2.setString(1,vid);

                ResultSet rs2=pst2.executeQuery();
                rs2.next();
            }
        }
    }
}

```



```

        if (rs2.next()==false)
        {
            String pid= rs.getString("Patient_ID");
            PreparedStatement pst1;
            ResultSet rs1;
            pst1= con.prepareStatement("SELECT * FROM `patientsinfo` WHERE Patient_ID=
?");
            pst1.setString(1, pid);
            rs1=pst1.executeQuery();
            if (rs1.next()==true)
            {
                PatientName.addItem(new Patient(rs.getString(1),rs.getString(2)));
            }
        }
    }
} catch (SQLException ex) {
    Logger.getLogger(Visit.class.getName()).log(Level.SEVERE, null, ex);
}
}

```

```

public void bill(){

```

```

    String total = ID_Label3.getText();
    DefaultTableModel model= new DefaultTableModel ();
    model = (DefaultTableModel)jTable1.getModel();

```

```

    bill.setText(bill.getText()+"*****\n");
    bill.setText(bill.getText()+"**      Sarvodaya Hospital Bill No. "+ID_Label.getText()+"
**\n");

```

```

    bill.setText(bill.getText()+"*****\n");
    bill.setText(bill.getText()+"Sno."+ "\t" + "Type"+ "\t" + "Charges"+ "\n");
    int x= 1;
    for (int i=0; i<model.getRowCount();i++)
    {
        String sno = Integer.toString(x);
        String type = (String)model.getValueAt(i, 1);
        String amount = (String)model.getValueAt(i, 2);
        bill.setText(bill.getText()+sno+"\t" + type+"\t"+amount+"\n");
        x=x+1;
    }
    bill.setText(bill.getText()+"\n");
    bill.setText(bill.getText()+"\n");
    bill.setText(bill.getText()+"Total Amount Rs. "+ID_Label3.getText()+"\n");
    bill.setText(bill.getText()+"Paid Rs. "+ID_Label3.getText()+"\n");

```

```

        bill.setText(bill.getText()+"Balance Rs. "+" 0"+"\\n"+"\\n");

bill.setText(bill.getText()+"*****\\n");
        bill.setText(bill.getText()+"                THANK YOU                "+"\\n");

bill.setText(bill.getText()+"*****\\n");
    }
    int total=0;
    int totaldiscount;
    int opdcharges;
    int indoorcharges;
    public void CalculateOPD(){
        try {
            Connect();
            SimpleDateFormat dateformat = new SimpleDateFormat("yyyy-MM-dd");
            String date = dateformat.format(jDateChooser.getDate());
            PreparedStatement pst= con.prepareStatement("SELECT * FROM `visit` where
Visit_ID= ?");
            pst.setString(1, ID_Label2.getText());
            ResultSet rs=pst.executeQuery();
            if (rs.next() ==true) {
                if ("1".equals(rs.getString("Free_Followup")))
                {
                    opdcharges=0;
                }
                else
                {
                    String did = rs.getString("Doctor_ID");
                    PreparedStatement pst1= con.prepareStatement("Select * from doctors where
Doctor_ID=?");
                    pst1.setString(1, did);
                    ResultSet rs1 = pst1.executeQuery();
                    rs1.next();
                    opdcharges = Integer.parseInt(rs1.getString("Fee"));
                    total = total+opdcharges;
                }
            }

        }catch (SQLException ex) {
            Logger.getLogger(Registration.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    public void CalculateIndoor(){
        try{
            Connect();
            PreparedStatement pst1= con.prepareStatement("Select * from indoor where
Visit_ID=?");

```

```

        pst1.setString(1, ID_Label2.getText());
        ResultSet rs1 = pst1.executeQuery();
        if (rs1.next()==true)
        {
            String rid = rs1.getString("Room_ID");
            PreparedStatement pst2= con.prepareStatement("Select * from rooms where
Room_ID=?");
            pst2.setString(1, rid);
            ResultSet rs2 = pst2.executeQuery();
            rs2.next();
            String roomcharges = rs2.getString("Room_Charges");
            DateTimeFormatter dtf = DateTimeFormatter.ofPattern("yyyy-MM-dd");
            String inputString1 = rs1.getString("Checkin_Date");
            String inputString2 = rs1.getString("Checkout_Date");
            LocalDate date1 = LocalDate.parse(inputString1, dtf);
            LocalDate date2 = LocalDate.parse(inputString2, dtf);
            long daysBetween = ChronoUnit.DAYS.between(date1, date2);
            int roomday = Integer.parseInt(roomcharges);
            indoorcharges = (int) (roomday*daysBetween);
            total = total + indoorcharges;
        }
        else {
            indoorcharges=0;
        }
    }catch (SQLException ex) {
        Logger.getLogger(Registration.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

```

public void bill_table(){

    try {
        Connect();
        pst= con.prepareStatement("SELECT * FROM `visit` where Visit_ID= ?");
        pst.setString(1, ID_Label2.getText());
        rs=pst.executeQuery();
        rs.next();
        ResultSetMetaData Rsm= rs.getMetaData();
        int c;
        c=Rsm.getColumnCount();
        DefaultTableModel df = (DefaultTableModel)jTable1.getModel();
        df.setRowCount(0);
    }
}

```

```

//opd charges
Vector v2= new Vector();
v2.add("1");
v2.add("Opd");
String opdcharges1 = Integer.toString(opdcharges);
v2.add(opdcharges1);
df.addRow(v2);
//indoor charges
Vector v3= new Vector();
v3.add("2");
v3.add("Indoor");
String indoorcharges1 = Integer.toString(indoorcharges);
v3.add(indoorcharges1);
df.addRow(v3);
// service charges
int x = 3;
PreparedStatement pst6 = con.prepareStatement("Select * from patientservice
where Visit_ID=?");
pst6.setString(1, ID_Label2.getText());
ResultSet rs6 = pst6.executeQuery();
while (rs6.next())
{
    Vector v4= new Vector();
    for(int i = 1; i<=1; i++)
    {
        v4.add(x);
        PreparedStatement pst7 =con.prepareStatement("Select* from servicesinfo
where Service_ID= ?");
        pst7.setString(1, rs6.getString("Service_ID"));
        ResultSet rs7=pst7.executeQuery();
        rs7.next();
        v4.add(rs7.getString("Service_Name"));
        v4.add(rs7.getString("Service_Charges"));
        int service = Integer.parseInt(rs7.getString("Service_Charges"));
        total = total+ service;
        x ++;
    }
    df.addRow(v4);
}
//discount
Vector v5= new Vector();
v5.add(x);
v5.add("Discount");
String oid = rs.getString("Organization_ID");
PreparedStatement pst8= con.prepareStatement("Select * from organizations
where Organization_ID=?");
pst8.setString(1, oid);

```

```

        ResultSet rs8= pst8.executeQuery();
        rs8.next();
        String discount = rs8.getString("Discount");
        int disc = Integer.parseInt(discount);
        int totaldisc = total*disc;
        totaldiscount = totaldisc/100;
        String discount1 = Integer.toString(totaldiscount);
        v5.add(discount1);
        int amount= total-totaldiscount;
        String amount1 = Integer.toString(amount);
        ID_Label3.setText(amount1);
        total=0;
        totaldiscount=0;
        df.addRow(v5);
    } catch (SQLException ex) {
        Logger.getLogger(Registration.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

```

public void AutoID(){

    try {
        Connection con =
java.sql.DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serv
erTimezone=UTC", "root", "");
        String sql = "SELECT MAX(Payment_ID) FROM `payments`";
        pst=con.prepareStatement(sql);
        rs=pst.executeQuery();
        rs.next();
        rs.getString("MAX(Payment_ID)");
        if (rs.getString("MAX(Payment_ID)")==null)
        {
            ID_Label.setText("BS001");
            jComboBox1.setSelectedIndex(-1);
        }else
        {
            long id =
Long.parseLong(rs.getString("MAX(Payment_ID)").substring(2,rs.getString("MAX(Payment_I
D)").length()));
            id ++;

            ID_Label.setText("BS"+String.format("%03d", id));
            jComboBox1.setSelectedIndex(-1);
        }
    }
    catch (SQLException ex) {

```



```

    }} bill();}} catch (SQLException ex) {
        Logger.getLogger(ServiceSlip.class.getName()).log(Level.SEVERE, null, ex);
    }
}

private void jButtonindoorActionPerformed(java.awt.event.ActionEvent evt) {
    java.util.Date stringDate=jDateChooser.getDate();
    if (stringDate==null)
    {
        JOptionPane.showMessageDialog(null, "Choose a valid
date","Error",JOptionPane.ERROR_MESSAGE);

    } else{
        LoadPatient1();
        jButtonPreview.setEnabled(true);
        bill.setText("");
        ID_Label2.setText("");
        ID_Label3.setText("");
    }
}

private void jButtonopdActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    java.util.Date stringDate=jDateChooser.getDate();
    if (stringDate==null)
    {
        JOptionPane.showMessageDialog(null, "Choose a valid
date","Error",JOptionPane.ERROR_MESSAGE);
    }else{
        LoadPatient();
        jButtonPreview.setEnabled(true);
        bill.setText("");
        ID_Label2.setText("");
        ID_Label3.setText("");
    }
}

private void jButtonopdMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
}

private void jButtonPrintActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        bill.print();
        bill.setText("");
    } catch (PrinterException ex) {
        Logger.getLogger(Bill.class.getName()).log(Level.SEVERE, null, ex);
    }
}
}

```

```

private void jButtonSave1ActionPerformed(java.awt.event.ActionEvent evt) {
    Connection con;
    try {
        con =
java.sql.DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serv
erTimezone=UTC", "root", "");
        SimpleDateFormat dateformat = new SimpleDateFormat("yyyy-MM-dd");
        String date = dateformat.format(jDateChooser.getDate());

        PreparedStatement ps = con.prepareStatement("INSERT INTO
`payments`(`Payment_ID`, `Payment_Amount`, `Visit_ID`, `Payment_Mode`,
`Payment_Date`) VALUES (?, ?, ?, ?, ?)");
        ps.setString(1, ID_Label.getText());
        ps.setString(2, ID_Label3.getText());
        ps.setString(3, ID_Label2.getText());
        ps.setString(4, jComboBox1.getSelectedItem().toString());
        ps.setString(5, date);
        int i = ps.executeUpdate();

        if (i>0){
            JOptionPane.showMessageDialog(null, "Payment Information Saved Successfully");
            AutoID();
            ID_Label3.setText("");
            ID_Label2.setText("");
            jComboBox1.setSelectedIndex(-1);

            jTable1.removeAll();
        }
    } catch (SQLException ex) {
        Logger.getLogger(Bill.class.getName()).log(Level.SEVERE, null, ex);
    }
}

private void jButtonSave2ActionPerformed(java.awt.event.ActionEvent evt) {
    try
    {
        String s = bill.getText();
        File f = new File("Bill.txt");
        FileWriter fw = new FileWriter(f);
        fw.write(s);
        fw.close();
        JOptionPane.showMessageDialog(null, "Bill Saved in File Bill.txt");
        bill.setText("");
    }catch(IOException ioe) {
        System.out.println("Exception Caught : " +ioe.getMessage());
        JOptionPane.showMessageDialog(null, "Bill NOT Saved");
    } } }

```


MainForm Class

```
import java.sql.PreparedStatement;
import javax.swing.JOptionPane;
import java.sql.Connection;
import java.sql.Date;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Vector;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import net.proteanit.sql.DbUtils;

public class mainForm extends javax.swing.JFrame {

    /**
     * Creates new form mainForm
     */
    public mainForm() {
        initComponents();
        doctor_table();
        AutoID();

    }
    Connection con;
    PreparedStatement pst;
    ResultSet rs;

    public void Connect(){
        try {
            Class.forName("com.mysql.jdbc.Driver");
            con=
DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serverTimez
one=UTC", "root", "");

        }

        catch (SQLException ex) {
            Logger.getLogger(Patients.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
```

```

    } catch (ClassNotFoundException ex) {
        Logger.getLogger(Registration.class.getName()).log(Level.SEVERE, null, ex);
    }
}

public void AutoID(){

    try {

        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serverTimezone=UTC", "root", "");
        String sql = "SELECT MAX(Doctor_ID) FROM `doctors`";
        pst=con.prepareStatement(sql);
        rs=pst.executeQuery();
        rs.next();
        rs.getString("MAX(Doctor_ID)");

        if (rs.getString("MAX(Doctor_ID)")==null)
        {
            ID_Label.setText("1");
        }

        else
        {
            int number = rs.getInt("MAX(Doctor_ID)");
            number++;
            ID_Label.setText(String.valueOf(number));
        }
    }
    catch (SQLException ex) {
        Logger.getLogger(Patients.class.getName()).log(Level.SEVERE, null, ex);
    }
}

public void doctor_table(){

    try {
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serverTimezone=UTC", "root", "");
        PreparedStatement pst;
        ResultSet rs;
        pst=con.prepareStatement("select * from doctors");
        rs=pst.executeQuery();
        ResultSetMetaData Rsm= rs.getMetaData();
        int c;
        c=Rsm.getColumnCount();
    }
}

```

```

DefaultTableModel df = (DefaultTableModel)jTable1.getModel();
df.setRowCount(0);

while (rs.next()){
    Vector v2= new Vector();
    for(int i = 1; i<=c; i++)
    {
        v2.add(rs.getString("Doctor_ID"));
        v2.add(rs.getString("Doctor_Name"));
        v2.add(rs.getString("Specialization"));
        v2.add(rs.getString("Qualification"));
        v2.add(rs.getString("Fee"));
        v2.add(rs.getString("Phone"));
    }
    df.addRow(v2);
}
} catch (SQLException ex) {
    Logger.getLogger(Registration.class.getName()).log(Level.SEVERE, null, ex);
}
}

private void jButtonDeleteMouseClicked(java.awt.event.MouseEvent evt) {

    String pid=ID_Label.getText();
    String pname= jTextFieldName.getText();

    try {

        Connection con =
        DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serverTimezone=UTC", "root", "");
        PreparedStatement pst;
        ResultSet rs;

        pst=con.prepareStatement("delete from `doctors` where Doctor_ID=?");

        pst.setString(1, pid);
        int i = pst.executeUpdate();
        if(i>0){
            JOptionPane.showMessageDialog(null,"Doctor "+ pname+ " Information Deleted Successfully");
            doctor_table();
            AutoID();
            jButtonSave.setEnabled(true);
            jTextFieldName.setText("");
            jTextFieldSpecialization.setText("");
            jTextFieldQualification.setText("");

```

```

        jTextFieldfee.setText("");
        jTextFieldContactNumber.setText("");

    }
    else {
        JOptionPane.showMessageDialog(null," Information not deleted successfully");
    }
} catch (SQLException e) {
    Logger.getLogger(Patients.class.getName()).log(Level.SEVERE, null, e);
    JOptionPane.showMessageDialog(null," Information not deleted successfully");
}

}

private void jButtonSaveActionPerformed(java.awt.event.ActionEvent evt) {

    String pid= ID_Label.getText();
    String pname=jTextFieldName.getText().toString();
    String pfee=jTextFieldfee.getText();
    String pspl=jTextFieldSpecialization.getText();
    String pqlf=jTextFieldQualification.getText();
    String pphone=jTextFieldContactNumber.getText();

    if(
        pname.trim().equals("") ||
        pfee.trim().equals("") || pphone.trim().equals("") || pspl.trim().equals("") ||
        pqlf.trim().equals("") )
    {
        JOptionPane.showMessageDialog(null, "All fields are compulsory "+" Kindly check all
        fields again","Error",JOptionPane.ERROR_MESSAGE);

    }

    else
    {
        try {

            Connection con =
            DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serverTimez
            one=UTC", "root", "");
            PreparedStatement pst;
            ResultSet rs;

            pst=con.prepareStatement("INSERT INTO `doctors`(`Doctor_ID`, `Doctor_Name`,
            `Specialization`, `Qualification`, `Fee`, `Phone`) VALUES (?, ?, ?, ?, ?, ?)");

```

```

        pst.setString(1, ID_Label.getText());
        pst.setString(2, jTextFieldName.getText());
        pst.setString(3, pspl);
        pst.setString(4, pqlf);
        pst.setString(5, pfee);
        pst.setString(6, pphone);

        int i = pst.executeUpdate();
        if(i>0){
            JOptionPane.showMessageDialog(null,"New Doctor "+ pname+ " Saved
Successfully");
            doctor_table();
            AutoID();
            jButtonSave.setEnabled(true);
            jTextFieldName.setText("");
            jTextFieldSpecialization.setText("");
            jTextFieldQualification.setText("");
            jTextFieldfee.setText("");
            jTextFieldContactNumber.setText("");

        }
        else {
            JOptionPane.showMessageDialog(null, pname+" 's Information not saved
successfully");
        }
    } catch (SQLException e) {
        Logger.getLogger(Patients.class.getName()).log(Level.SEVERE, null, e);
        JOptionPane.showMessageDialog(null, pname+" 's Information not saved
successfully");
    }
}

private void jButtonUpdateMouseClicked(java.awt.event.MouseEvent evt) {
{
    String pid= ID_Label.getText();
    String pname=jTextFieldName.getText().toString();
    String pfee=jTextFieldfee.getText();
    String pspl=jTextFieldSpecialization.getText();
    String pqlf=jTextFieldQualification.getText();
    String pphone=jTextFieldContactNumber.getText();
    if(pname.trim().equals("") ||
        pfee.trim().equals("") || pphone.trim().equals("") || pspl.trim().equals("") ||
        pqlf.trim().equals("") )
    {
        JOptionPane.showMessageDialog(null, "All fields are compulsory "+" Kindly check all
fields again","Error",JOptionPane.ERROR_MESSAGE);
    }
}
}

```

```

    }else
    {
        try {
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serverTimez
one=UTC", "root", "");
            PreparedStatement pst;
            ResultSet rs;
            pst=con.prepareStatement("Update `doctors` set Doctor_Name = ?,
`Specialization` = ?, `Qualification`=?, `Fee`=?, `Phone`=? where Doctor_ID=?");

            pst.setString(1, jTextFieldName.getText());
            pst.setString(2, pspl);
            pst.setString(3, pqIf);
            pst.setString(4, pfee);
            pst.setString(5, pphone);
            pst.setString(6, pid);
            int i = pst.executeUpdate();
            if(i>0){
                JOptionPane.showMessageDialog(null,"Patient "+ pname+ " Information
Updated Successfully");
                doctor_table();
                AutoID();
                jButtonSave.setEnabled(true);
                jTextFieldName.setText("");
                jTextFieldSpecialization.setText("");
                jTextFieldQualification.setText("");
                jTextFieldfee.setText("");
                jTextFieldContactNumber.setText("");
            }
            else {
                JOptionPane.showMessageDialog(null, pname+" 's Information not updated
successfully");
            }
        } catch (SQLException e) {
            Logger.getLogger(Patients.class.getName()).log(Level.SEVERE, null, e);
            JOptionPane.showMessageDialog(null, pname+" 's Information not updated
successfully");
        }
    }
}
}
}

```

```

private void jButtonClearActionPerformed(java.awt.event.ActionEvent evt) {

```

```

    AutoID();
    jButtonSave.setEnabled(true);

```

```

        jTextFieldName.setText("");
        jTextFieldSpecialization.setText("");
        jTextFieldQualification.setText("");
        jTextFieldfee.setText("");
        jTextFieldContactNumber.setText("");
    }

    private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {

        DefaultTableModel d1 = (DefaultTableModel)jTable1.getModel();
        int SelectIndex = jTable1.getSelectedRow();
        ID_Label.setText(d1.getValueAt(SelectIndex, 0).toString());
        jTextFieldName.setText(d1.getValueAt(SelectIndex, 1).toString());
        jTextFieldSpecialization.setText(d1.getValueAt(SelectIndex, 2).toString());
        jTextFieldQualification.setText(d1.getValueAt(SelectIndex, 3).toString());
        jTextFieldfee.setText(d1.getValueAt(SelectIndex, 4).toString());
        jTextFieldContactNumber.setText(d1.getValueAt(SelectIndex, 5).toString());
        jButtonSave.setEnabled(false);

    }

```

Organizations Class

```

import java.sql.PreparedStatement;
import javax.swing.JOptionPane;
import java.sql.Connection;
import java.sql.Date;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Vector;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import net.proteanit.sql.DbUtils;

public class Organizations extends javax.swing.JFrame {

    /**
     * Creates new form Organizations
     */
    public Organizations() {
        initComponents();
    }

```

```

        organization_table();
        AutoID();
    }

    Connection con;
    PreparedStatement pst;
    ResultSet rs;

    public void Connect(){
        try {
            Class.forName("com.mysql.jdbc.Driver");
            con=
DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serverTimez
one=UTC", "root", "");

        }

        catch (SQLException ex) {
            Logger.getLogger(Patients.class.getName()).log(Level.SEVERE, null, ex);
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(Registration.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    public void AutoID(){

        try {

            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serverTimez
one=UTC", "root", "");
            String sql = "SELECT MAX(Organization_ID) FROM `organizations`";
            pst=con.prepareStatement(sql);
            rs=pst.executeQuery();
            rs.next();
            rs.getString("MAX(Organization_ID)");

            if (rs.getString("MAX(Organization_ID)")==null)
            {
                ID_Label.setText("1");
            }

            else
            {
                int number = rs.getInt("MAX(Organization_ID)");
                number++;
                ID_Label.setText(String.valueOf(number));
            }
        }
    }

```



```

    }
}
catch (SQLException ex) {
    Logger.getLogger(Patients.class.getName()).log(Level.SEVERE, null, ex);
}
}

public void organization_table(){

    try {
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serverTimezone=UTC", "root", "");
        PreparedStatement pst;
        ResultSet rs;
        pst=con.prepareStatement("select * from organizations");
        rs=pst.executeQuery();

        ResultSetMetaData Rsm= rs.getMetaData();
        int c;
        c=Rsm.getColumnCount();
        DefaultTableModel df = (DefaultTableModel)jTable1.getModel();
        df.setRowCount(0);

        while (rs.next())
        {

            Vector v2= new Vector();

            for(int i = 1; i<=c; i++)
            {
                v2.add(rs.getString("Organization_ID"));
                v2.add(rs.getString("Organisation_Name"));
                v2.add(rs.getString("Discount"));

            }

            df.addRow(v2);

        }

    } catch (SQLException ex) {
        Logger.getLogger(Registration.class.getName()).log(Level.SEVERE, null, ex);
    }
}

private void jButtonDeleteMouseClicked(java.awt.event.MouseEvent evt) {

```

```

String pid=ID_Label.getText();
String pname= jTextFieldName.getText();

try {

    Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serverTimez
one=UTC", "root", "");
    PreparedStatement pst;
    ResultSet rs;

    pst=con.prepareStatement("delete from `organizations` where Organization_ID=?");

    pst.setString(1, pid);
    int i = pst.executeUpdate();
    if(i>0){
        JOptionPane.showMessageDialog(null,"Organization "+ pname+ " Information
Deleted Successfully");
        organization_table();
        AutoID();
        jButtonSave.setEnabled(true);
        jTextFieldName.setText("");
        jTextFieldDiscount.setText("");
        jTextFieldQualification.setText("");

    }
    else {
        JOptionPane.showMessageDialog(null," Information not deleted successfully");
    }
} catch (SQLException e) {
    Logger.getLogger(Patients.class.getName()).log(Level.SEVERE, null, e);
    JOptionPane.showMessageDialog(null," Information not deleted successfully");
}
}

private void jButtonSaveActionPerformed(java.awt.event.ActionEvent evt) {

String pid= ID_Label.getText();
String pname=jTextFieldName.getText().toString();

String pspl=jTextFieldDiscount.getText();

if(
    pname.trim().equals("")||
    pspl.trim().equals("") )

```

```

{
    JOptionPane.showMessageDialog(null, "All fields are compulsory "+" Kindly check all
fields again", "Error", JOptionPane.ERROR_MESSAGE);

}

else
{
    try {

        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serverTimez
one=UTC", "root", "");
        PreparedStatement pst;
        ResultSet rs;

        pst=con.prepareStatement("INSERT INTO `organizations`(`Organization_ID`,
`Organisation_Name`, `Discount`) VALUES (?, ?, ?)");

        pst.setString(1, ID_Label.getText());
        pst.setString(2, jTextFieldName.getText());
        pst.setString(3, pspl);

        int i = pst.executeUpdate();
        if(i>0){
            JOptionPane.showMessageDialog(null, "New Organization "+ pname+ " Saved
Successfully");
            organization_table();
            AutoID();
            jButtonSave.setEnabled(true);
            jTextFieldName.setText("");
            jTextFieldDiscount.setText("");
            jTextFieldQualification.setText("");

        }
        else {
            JOptionPane.showMessageDialog(null, pname+" 's Information not saved
successfully");
        }
    } catch (SQLException e) {
        Logger.getLogger(Patients.class.getName()).log(Level.SEVERE, null, e);
        JOptionPane.showMessageDialog(null, pname+" 's Information not saved
successfully");
    }

}
}

```

```

}

private void jButtonUpdateMouseClicked(java.awt.event.MouseEvent evt) {
{
    String pid= ID_Label.getText();
    String pname=jTextFieldName.getText().toString();

    String pspl=jTextFieldDiscount.getText();

    if(
        pname.trim().equals("") ||
        pspl.trim().equals("")
    )
    {
        JOptionPane.showMessageDialog(null, "All fields are compulsory "+" Kindly check
all fields again", "Error", JOptionPane.ERROR_MESSAGE);

    }

    else
    {
        try {

            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serverTimez
one=UTC", "root", "");
            PreparedStatement pst;
            ResultSet rs;

            pst=con.prepareStatement("Update `organizations` set Organisation_Name = ?,
`Discount` = ? where Organization_ID=?");

            pst.setString(1, jTextFieldName.getText());
            pst.setString(2, pspl);

            pst.setString(3, pid);

            int i = pst.executeUpdate();
            if(i>0){
                JOptionPane.showMessageDialog(null, "Organization "+ pname+ " Information
Updated Successfully");
                organization_table();
                AutoID();
                jButtonSave.setEnabled(true);
                jTextFieldName.setText("");
                jTextFieldDiscount.setText("");
                jTextFieldQualification.setText("");
            }
        }
    }
}
}

```

```

    }
    else {
        JOptionPane.showMessageDialog(null, pname+" 's Information not updated
successfully");
    }
} catch (SQLException e) {
    Logger.getLogger(Patients.class.getName()).log(Level.SEVERE, null, e);
    JOptionPane.showMessageDialog(null, pname+" 's Information not updated
successfully");
}
}
}
}

```

```

private void jButtonClearActionPerformed(java.awt.event.ActionEvent evt) {

```

```

    AutoID();
    jButtonSave.setEnabled(true);
    jTextFieldName.setText("");
    jTextFieldDiscount.setText("");
    jTextFieldQualification.setText("");

```

```

}

```

```

private void jTextFieldDiscountActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

```

```

private void jTextFieldQualificationActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

```

```

private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {

```

```

    DefaultTableModel d1 = (DefaultTableModel)jTable1.getModel();
    int SelectIndex = jTable1.getSelectedRow();
    ID_Label.setText(d1.getValueAt(SelectIndex, 0).toString());
    jTextFieldName.setText(d1.getValueAt(SelectIndex, 1).toString());
    jTextFieldDiscount.setText(d1.getValueAt(SelectIndex, 2).toString());
    jTextFieldQualification.setText("4");

```

```

    jButtonSave.setEnabled(false);
}

```

Services Class:

```
import java.sql.PreparedStatement;
import javax.swing.JOptionPane;
import java.sql.Connection;
import java.sql.Date;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Vector;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import net.proteanit.sql.DbUtils;

public class Services extends javax.swing.JFrame {

    /**
     * Creates new form Services
     */
    public Services() {
        initComponents();
        AutoID();
        service_table();
    }
    Connection con;
    PreparedStatement pst;
    ResultSet rs;

    public void Connect(){
        try {
            Class.forName("com.mysql.jdbc.Driver");
            con=
DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serverTimez
one=UTC", "root", "");

        }

        catch (SQLException ex) {
            Logger.getLogger(Patients.class.getName()).log(Level.SEVERE, null, ex);
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(Registration.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
```

```

    }
}
public void AutoID(){

    try {

        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serverTimez
one=UTC", "root", "");
        String sql = "SELECT MAX(Service_ID) FROM `servicesinfo`";
        pst=con.prepareStatement(sql);
        rs=pst.executeQuery();
        rs.next();
        rs.getString("MAX(Service_ID)");

        if (rs.getString("MAX(Service_ID)")==null)
        {
            ID_Label1.setText("1");
        }

        else
        {
            int number = rs.getInt("MAX(Service_ID)");
            number++;
            ID_Label1.setText(String.valueOf(number));
        }
    }
    catch (SQLException ex) {
        Logger.getLogger(Patients.class.getName()).log(Level.SEVERE, null, ex);
    }
}

public void service_table(){

    try {
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serverTimez
one=UTC", "root", "");
        PreparedStatement pst;
        ResultSet rs;
        pst=con.prepareStatement("select * from servicesinfo");
        rs=pst.executeQuery();
        ResultSetMetaData Rsm= rs.getMetaData();
        int c;
        c=Rsm.getColumnCount();
        DefaultTableModel df = (DefaultTableModel)jTable1.getModel();
        df.setRowCount(0);
        while (rs.next())

```

```

        {
            Vector v2= new Vector();
            for(int i = 1; i<=c; i++)
            {
                v2.add(rs.getString("Service_ID"));
                v2.add(rs.getString("Service_Name"));
                v2.add(rs.getString("Service_Type"));
                v2.add(rs.getString("Service_Charges"));
            }
            df.addRow(v2);
        }
    } catch (SQLException ex) {
        Logger.getLogger(Registration.class.getName()).log(Level.SEVERE, null, ex);
    }
}

private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {

    DefaultTableModel d1 = (DefaultTableModel)jTable1.getModel();
    int SelectIndex = jTable1.getSelectedRow();
    ID_Label1.setText(d1.getValueAt(SelectIndex, 0).toString());
    jTextFieldName1.setText(d1.getValueAt(SelectIndex, 1).toString());
    PreparedStatement pst;
    try {
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serverTimez
one=UTC", "root", "");
        pst = con.prepareStatement("Select * from servicesinfo where Service_ID=?");
        pst.setString(1, d1.getValueAt(SelectIndex, 0).toString());
        ResultSet rs = pst.executeQuery();
        rs.next();
        String description = rs.getString("Service_Description");
        jTextFieldDescription.setText(description);
        jTextFieldfee.setText(d1.getValueAt(SelectIndex, 3).toString());
        String type =d1.getValueAt(SelectIndex, 2).toString();
        if (type == "Indoor")
        {
            TypeBox.setSelectedIndex(0);
        }else
        {
            TypeBox.setSelectedIndex(1);
        }
        jButtonSave.setEnabled(false);
    } catch (SQLException ex) {
        Logger.getLogger(Services.class.getName()).log(Level.SEVERE, null, ex);
    }
}
}

```



```

private void jTextFieldDescriptionActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void jButtonClearActionPerformed(java.awt.event.ActionEvent evt) {

    AutoID();
    jButtonSave.setEnabled(true);
    jTextFieldName1.setText("");
    jTextFieldDescription.setText("");
    jTextFieldfee.setText("");
}

private void jButtonUpdateMouseClicked(java.awt.event.MouseEvent evt) {
{
    String pid= ID_Label1.getText();
    String pname=jTextFieldName1.getText().toString();
    String pspl=jTextFieldDescription.getText();
    String ptype = TypeBox.getSelectedItem().toString();
    String pfee = jTextFieldfee.getText();
    if(pname.trim().equals("") || pspl.trim().equals("") || pfee.trim().equals(""))
    {
        JOptionPane.showMessageDialog(null, "All fields are compulsory "+" Kindly check
all fields again", "Error", JOptionPane.ERROR_MESSAGE);
    }else
    {
        try {
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serverTimez
one=UTC", "root", "");
            PreparedStatement pst;
            ResultSet rs;
            pst=con.prepareStatement("UPDATE `servicesinfo` SET
`Service_Name`=?, `Service_Description`=?, `Service_Charges`=?, `Service_Type`=? WHERE
`Service_ID`=?"");
            pst.setString(1, pname);
            pst.setString(2, pspl);
            pst.setString(3, pfee);
            pst.setString(4, ptype);
            pst.setString(5, pid);
            int i = pst.executeUpdate();
            if(i>0){
                JOptionPane.showMessageDialog(null, "Service "+ pname+ " Information
Updated Successfully");
                service_table();
                AutoID();
                jButtonSave.setEnabled(true);
            }
        } catch (SQLException ex) {
            JOptionPane.showMessageDialog(null, "Error: " + ex.getMessage());
        }
    }
}
}

```

```

        jTextFieldName1.setText("");
        jTextFieldDescription.setText("");
        jTextFieldfee.setText("");
    }else {
        JOptionPane.showMessageDialog(null, pname+" 's Information not updated
successfully");
    }
    } catch (SQLException e) {
        Logger.getLogger(Patients.class.getName()).log(Level.SEVERE, null, e);
        JOptionPane.showMessageDialog(null, pname+" 's Information not updated
successfully");
    }

    }
}
}

```

```

private void jButtonSaveActionPerformed(java.awt.event.ActionEvent evt) {

    String pid= ID_Label1.getText();
    String pname=jTextFieldName1.getText().toString();
    String pspl=jTextFieldDescription.getText();
    String ptype = TypeBox.getSelectedItem().toString();
    String pfee = jTextFieldfee.getText();
    if(
        pname.trim().equals("") ||
        pspl.trim().equals("") || pfee.trim().equals(""))
    {
        JOptionPane.showMessageDialog(null, "All fields are compulsory "+" Kindly check all
fields again", "Error",JOptionPane.ERROR_MESSAGE);

    }

    else
    {
        try {

            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serverTimez
one=UTC", "root", "");
            PreparedStatement pst;
            ResultSet rs;
            pst=con.prepareStatement("INSERT INTO `servicesinfo`(`Service_ID`,
`Service_Name`, `Service_Description`, Service_Charges, Service_Type) VALUES (?, ?, ?, ?, ?)");
            pst.setString(1, ID_Label1.getText());
            pst.setString(2, jTextFieldName1.getText());
            pst.setString(3, pspl);

```

```

        pst.setString(4, pfee);
        pst.setString(5, ptype);
        int i = pst.executeUpdate();
        if(i>0){
            JOptionPane.showMessageDialog(null,"New Service "+ pname+ " Saved
Successfully");
            service_table();
            AutoID();
            jButtonSave.setEnabled(true);
            jTextFieldName1.setText("");
            jTextFieldDescription.setText("");
            jTextFieldfee.setText("");
        }
        else {
            JOptionPane.showMessageDialog(null, pname+" 's Information not saved
successfully");
        }
    } catch (SQLException e) {
        Logger.getLogger(Patients.class.getName()).log(Level.SEVERE, null, e);
        JOptionPane.showMessageDialog(null, pname+" 's Information not saved
successfully");
    }
}
}

```

```

private void jButtonDeleteMouseClicked(java.awt.event.MouseEvent evt) {

    String pid= ID_Label1.getText();
    String pname=jTextFieldName1.getText().toString();
    String pspl=jTextFieldDescription.getText();
    String ptype = TypeBox.getSelectedItem().toString();
    String pfee = jTextFieldfee.getText();

    try {
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serverTimez
one=UTC", "root", "");
        PreparedStatement pst;
        ResultSet rs;
        pst=con.prepareStatement("delete from `servicesinfo` where Service_ID=?");
        pst.setString(1, pid);
        int i = pst.executeUpdate();
        if(i>0){
            JOptionPane.showMessageDialog(null,"Service "+ pname+ " Information Deleted
Successfully");
            service_table();

```

```

        AutoID();
        jButtonSave.setEnabled(true);
        jTextFieldName1.setText("");
        jTextFieldDescription.setText("");
        jTextFieldfee.setText("");
    }
    else {
        JOptionPane.showMessageDialog(null," Information not deleted successfully");
    }
} catch (SQLException e) {
    Logger.getLogger(Patients.class.getName()).log(Level.SEVERE, null, e);
    JOptionPane.showMessageDialog(null," Information not deleted successfully");
}
}

```

Users Class

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import net.proteanit.sql.DbUtils;

public class Users extends javax.swing.JFrame {

    public Users() {
        initComponents();
        update_table();
    }

    public void update_table(){
        PreparedStatement pst;
        ResultSet rs;
        try {
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serverTimezone=UTC", "root", "");
            String sql = "SELECT `username`, `password`, `type`, `name` FROM `users`";
            pst=con.prepareStatement(sql);
            rs=pst.executeQuery();
            jTable1.setModel(DbUtils.resultSetToTableModel(rs));
        } catch (SQLException ex) {
            Logger.getLogger(Patients.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}

```

```

    }
    private void jButtonSaveActionPerformed(java.awt.event.ActionEvent evt) {

        String pid = jTextFieldUserID.getText().toString();
        String pname = jTextFieldName.getText().toString();
        String ptype = "";
        if (jRadioButtonAdmin.isSelected()){
            ptype = "Admin";
        }
        else if (jRadioButtonUser.isSelected()){
            ptype = "User";
        }
        String ppassword = jTextFieldPassword.getText();
        try {
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serverTimez
one=UTC", "root", "");
            PreparedStatement pst;
            ResultSet rs;
            String sql = "INSERT INTO `users`(`username`, `password`, `type`, `name`) VALUES
(?,?,?,?)";
            pst = con.prepareStatement(sql);
            pst.setString(1, pid);
            pst.setString(2, ppassword);
            pst.setString(3, ptype);
            pst.setString(4, pname);
            int i = pst.executeUpdate();
            if(i >= 1){
                JOptionPane.showMessageDialog(null, "New User " + pname + " Saved
Successfully");
                update_table();
            }
            else {
                JOptionPane.showMessageDialog(null, pname + "'s Information not saved
successfully");
            }
        } catch (SQLException e) {
            Logger.getLogger(Patients.class.getName()).log(Level.SEVERE, null, e);
            JOptionPane.showMessageDialog(null, pname + "Information not saved
successfully");
        }
    }
    private void jButtonClearActionPerformed(java.awt.event.ActionEvent evt) {
        jTextFieldUserID.setText("");
        jTextFieldName.setText("");
        jRadioButtonAdmin.setSelected(false);
        jRadioButtonUser.setSelected(false);
    }

```

```
jTextFieldPassword.setText("");
}
```

Payments Class

```

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.text.SimpleDateFormat;
import java.time.format.DateTimeFormatter;
import java.util.Vector;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

public class Payments extends javax.swing.JFrame {

    public Payments() {
        initComponents()
    }

    public void indoor_table(String r){
        try {
            Connection con =
                java.sql.DriverManager.getConnection("jdbc:mysql://localhost:3306/"
                    + "java_hospital_db?serverTimezone=UTC", "root", "");
            PreparedStatement pst;
            ResultSet rs;
            pst=con.prepareStatement("select * from payments WHERE Payment_DATE >= ?
            AND Payment_Date <=?");
            pst.setString(1, r+"-01-01");
            pst.setString(2, r+"-12-31");
            rs=pst.executeQuery();
            ResultSetMetaData Rsm= rs.getMetaData();
            int c;
            c=Rsm.getColumnCount();
            DefaultTableModel df = (DefaultTableModel)jTable1.getModel();
            df.setRowCount(0);
            while (rs.next())
            {
                Vector v2= new Vector();
                for(int i = 1; i<=c; i++)
                {
                    v2.add(rs.getString("Payment ID"));
                }
            }
        }
    }
}

```

```

        v2.add(rs.getString("Payment_Amount"));
        v2.add(rs.getString("Visit_ID"));
        v2.add(rs.getString("Payment_Mode"));
        v2.add(rs.getString("Payment_Date"));
    }
    df.addRow(v2);
}
int total = 0;
DefaultTableModel model= new DefaultTableModel ();
model = (DefaultTableModel)jTable1.getModel();
for (int o=0; o<model.getRowCount();o++)
{
    String amount = (String)model.getValueAt(o, 1);
    int amount1 = Integer.parseInt(amount);
    total = total+amount1;
}
String totalamount = Integer.toString(total);
jLabel5.setText("Rs. "+totalamount);
} catch (SQLException ex) {
    Logger.getLogger(Registration.class.getName()).log(Level.SEVERE, null, ex);
}
}

```

```

public void indoor_table1(){
    String month = null;
    try {
        int i = jComboBox1.getSelectedIndex();
        for (int j=0;j<=11;j++)
        {
            if (i==j)
            {
                int m = i+1;
                month = String.format("%02d", m);
            }
        }
        String r =jTextField1.getText();
        Connection con =
        java.sql.DriverManager.getConnection("jdbc:mysql://localhost:3306/java_hospital_db?serv
        erTimezone=UTC", "root", "");
        PreparedStatement pst;
        ResultSet rs;
        pst=con.prepareStatement("select * from payments WHERE Payment_DATE >= ?
        AND Payment_Date <=?");
        pst.setString(1, r+"-"+month+"-01");
        pst.setString(2, r+"-"+month+"-31");
        rs=pst.executeQuery();
        ResultSetMetaData Rsm= rs.getMetaData();
    }
}

```

```

int c;
c=Rsm.getColumnCount();
DefaultTableModel df = (DefaultTableModel)jTable1.getModel();
df.setRowCount(0);
while (rs.next())
{
    Vector v2= new Vector();
    for(int k = 1; k<=c; k++)
    {
        v2.add(rs.getString("Payment_ID"));
        v2.add(rs.getString("Payment_Amount"));
        v2.add(rs.getString("Visit_ID"));
        v2.add(rs.getString("Payment_Mode"));
        v2.add(rs.getString("Payment_Date"));
    }
    df.addRow(v2);
}
int total = 0;
DefaultTableModel model= new DefaultTableModel ();
model = (DefaultTableModel)jTable1.getModel();
for (int o=0; o<model.getRowCount();o++)
{
    String amount = (String)model.getValueAt(o, 1);
    int amount1 = Integer.parseInt(amount);
    total = total+amount1;
}
String totalamount = Integer.toString(total);
jLabel5.setText("Rs. "+totalamount);
} catch (SQLException ex) {
    Logger.getLogger(Registration.class.getName()).log(Level.SEVERE, null, ex);
}
}

```

```

private void jButtononeyearlyActionPerformed(java.awt.event.ActionEvent evt) {

```

```

    String r = jTextField1.getText();
    if (r=="")
    {
        JOptionPane.showMessageDialog(null, "Enter a valid year",
"Error",JOptionPane.ERROR_MESSAGE);
    }else {
        indoor_table(r);
    }
}

```

```

private void jButtonmonthlyActionPerformed(java.awt.event.ActionEvent evt) {
    indoor_table1();
}

```