



JAVA FULL-STACK DEVELOPMENT COURSE JAVA SWING PROJECT

HOSPITAL MANAGEMENT SYSTEM

Project Report submitted in Partial fulfillment requirement for the Proficient
Certificate Course

Done By

SANTHIYA C K

MCA(2023-2025)

Under the Guidance of

RANJINI S

S.NO	TOPIC	PG.NO
1	ABOUT THE PROJECT	4
2	JAVA SWING EXPLANATION	6
3	HOME PAGE	7
4	ADMIN PAGE <ul style="list-style-type: none"> i. Admin_Login ii. Admin_main iii. Admin_Add_Doctor iv. Admin_Delete_Doctor v. Admin_View_Doctor vi. Admin_Add_Staff vii. Admin_Delete_Staff viii. Admin_View_Staff ix. Admin_View_Patient x. Admin_Calculate_Expenses 	8
5	DOCTOR PAGE <ul style="list-style-type: none"> i. Doctor_Login ii. Doctor_Main iii. Doctor_View_Patient_Appointment iv. Doctor_View_Patient_Details v. Doctor_Predict_Patient_Details 	22
6	STAFF PAGE <ul style="list-style-type: none"> i. Staff_Login ii. Staff_Main iii. Staff_Add_Patient iv. Staff_View_Patient v. Staff_Delete_Patient vi. Staff_Checkout_Patient vii. Staff_View_Checkout 	28

7	PATIENT PAGE i. Patient_Registration ii. Patient_Login iii. Patient_Main iv. Patient_View_Appointment	38
8	CONCLUSION	44

ABOUT THE PROJECT

The **Hospital Management System** is a Java Swing-based desktop application developed to simplify and automate the administrative and medical functions of hospitals and clinics. It offers an efficient and organized approach to managing essential workflows such as patient registration, doctor management, diagnosis recording, billing, and patient discharge. The system supports multiple user roles, primarily Admins and Doctors, with role-based access to ensure security and functional separation. Admins can manage hospital data, including adding or removing doctors, handling patient records, processing billing, and managing patient checkout. Doctors, on the other hand, can log in to access patient details and add disease or diagnosis records. The application features a user-friendly interface built using Java Swing components, ensuring ease of navigation and interaction. Its design is suited for small to mid-sized healthcare facilities and offers a practical solution to replace traditional paperwork-based management.

Key Features:

User Authentication System

- Login system for Admin and Doctor roles

- Ensures secure and restricted access

- Redirects users to role-specific dashboards

Admin Dashboard

- Access to manage doctors, patients, and billing

- Centralized control for all hospital activities

- Easy navigation to all modules

Doctor Management

- Add new doctors with personal and professional details

- View a list of all registered doctors

- Remove doctor records when needed

Patient Registration

- Register new patients with essential details

- Maintain and update patient database

View patient information on demand

Diagnosis Entry

Doctors can add diagnosis and treatment details

View and update existing diagnosis records

Link diagnosis to patient profiles

Billing System

Generate patient bills based on treatments and services

View billing history for each patient

Admin can modify or delete billing entries

Patient Checkout

Record patient discharge date and summary

Finalize and save discharge process

Link billing and diagnosis to checkout

Project Prerequisites:

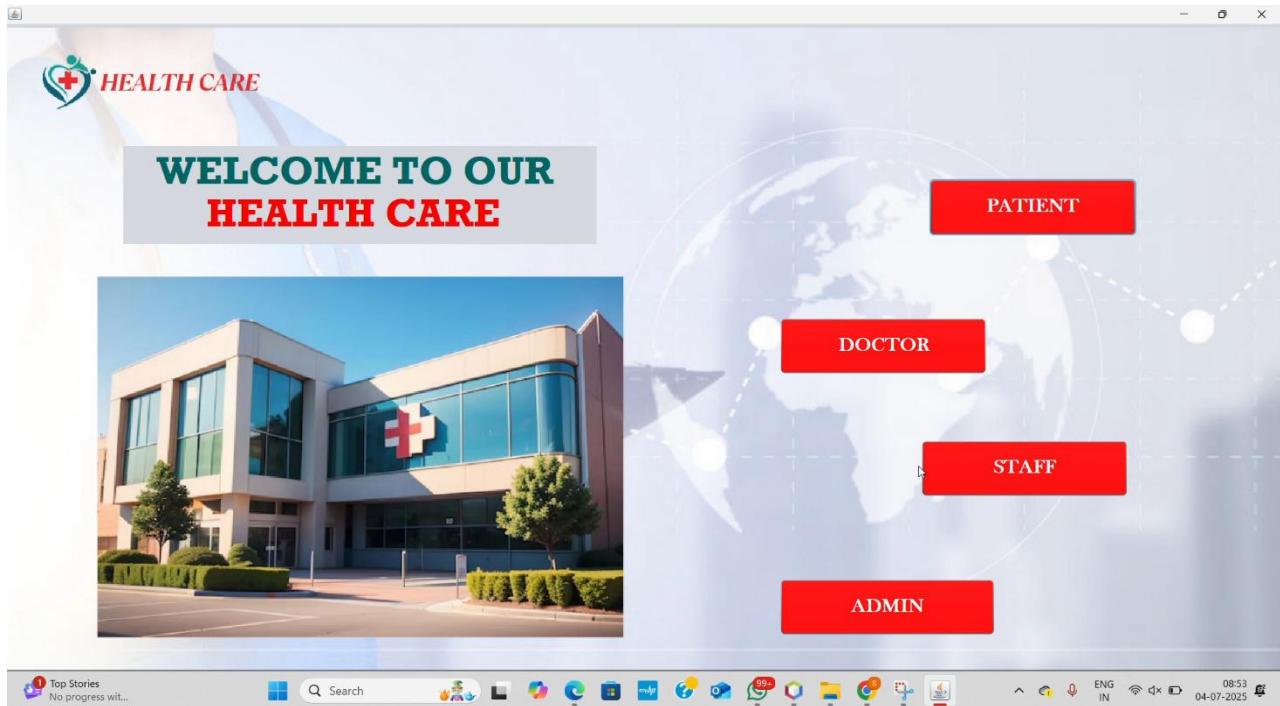
To build this project, we will need the following libraries:

- NetBeans IDE- Java Swing (via NetBeans GUI Builder)
- Java Swing- To build a front-end (UI)
- MySQL Connector – To connect with the Database

JAVA SWING

- Java Swing is a part of Java's Standard Library used for developing **Graphical User Interface (GUI)** applications. It provides a set of components for building rich desktop applications that are **platform-independent** and **highly customizable**.
- Swing is built on top of the Abstract Window Toolkit (AWT) and follows a **lightweight component architecture**, allowing it to run on any system with Java installed.
- The primary purpose of Java Swing is to enable the development of **interactive and user-friendly desktop applications**. It supports a wide range of GUI components such as buttons, text fields, labels, tables, combo boxes, panels, and windows, which are essential for creating forms, dashboards, and navigation menus.
- **Platform Independence:** Swing components are written entirely in Java, making them run consistently across different operating systems.
- **Lightweight Components:** Unlike AWT, Swing components do not rely heavily on native GUI components, making them faster and more flexible.
- **Customizable Look and Feel:** Swing supports pluggable look-and-feel designs, allowing developers to change the appearance of components easily.
- **Rich Set of Features:** Swing offers advanced components like JTable, JTree, JTabbedPane, and JScrollPane, which are not available in AWT.
- **Event Handling Mechanism:** Java Swing provides a robust event-handling model, allowing programs to respond to user actions like mouse clicks, key presses, or menu selections.
- **Integration with IDEs:** Swing is fully supported by IDEs like **NetBeans**, which offers drag-and-drop functionality to speed up GUI design and development.

HOME PAGE:



CODE:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    Admin_Login v=new Admin_Login();
    v.setVisible(true);
    this.setVisible(false);           // TODO add your handling code here:
}

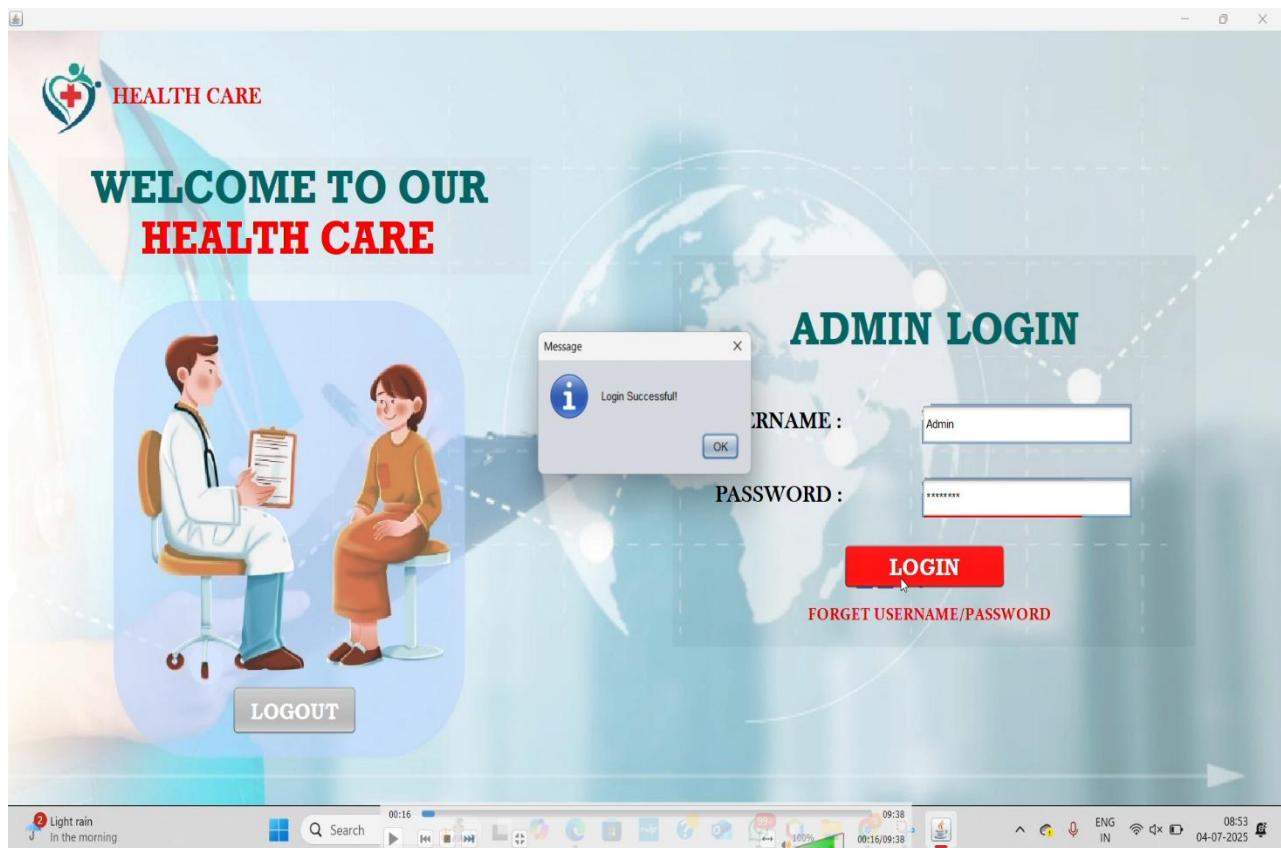
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    Patient_Login v=new Patient_Login();
    v.setVisible(true);
    this.setVisible(false);           // TODO add your handling code here:
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    Staff_Login v=new Staff_Login();
    v.setVisible(true);
    this.setVisible(false);           // TODO add your handling code here:
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    Doctor_Login v=new Doctor_Login();
    v.setVisible(true);
    this.setVisible(false);           // TODO add your handling code here:
}
```

1) ADMIN PAGE:

ADMIN LOGIN:



CODE:

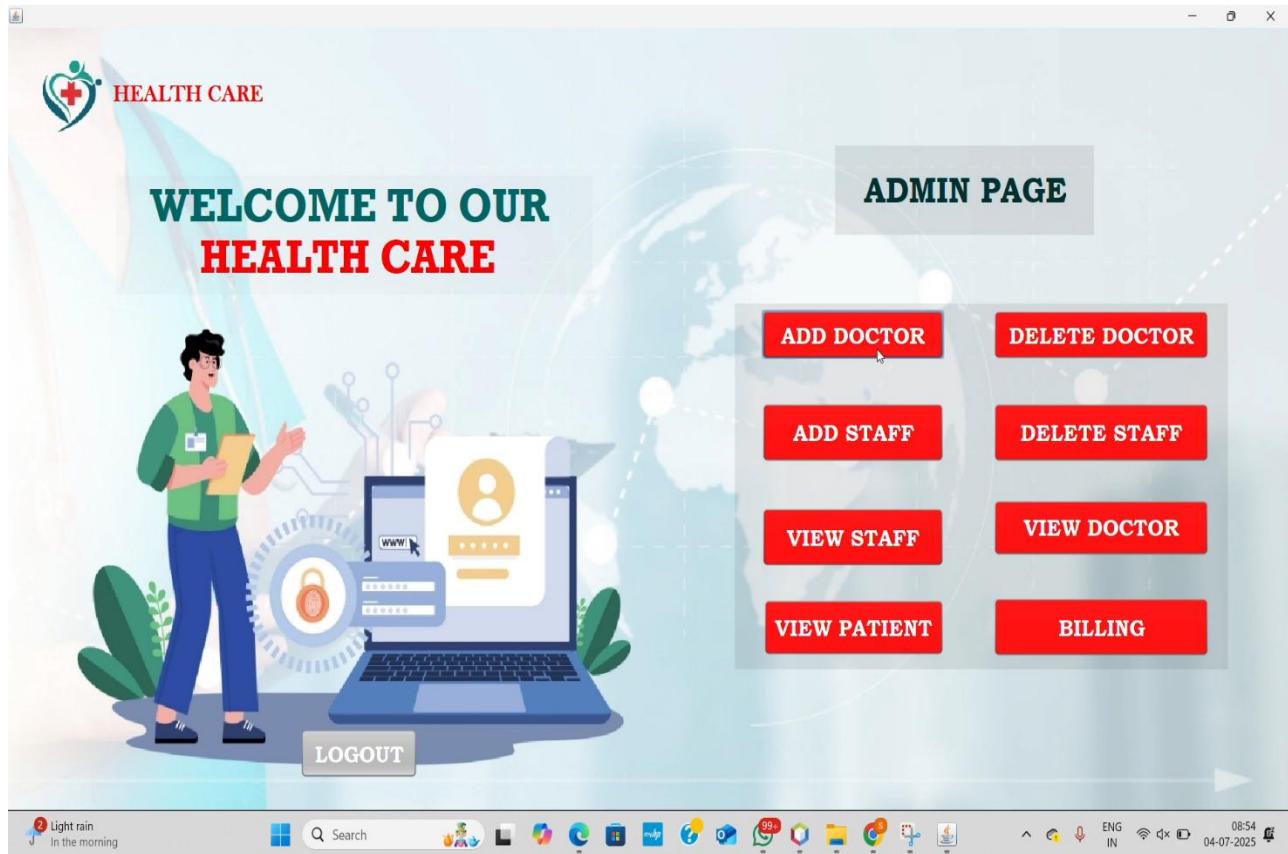
```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    String username = Username.getText();
    String password = new String(Password.getPassword());

    if (username.equals("Admin") && password.equals("Admin123"))
    {
        JOptionPane.showMessageDialog(this, "Login Successful!");
        // Open dashboard or new form here
    } else {
        JOptionPane.showMessageDialog(this, "Invalid username or password", "Login Failed", JOptionPane.ERROR_MESSAGE);
    }

    Admin_Main v=new Admin_Main();
    v.setVisible(true);
    this.setVisible(false);
    // TODO add your handling code here:
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    Home v=new Home();
    v.setVisible(true);
    this.setVisible(false);      // TODO add your handling code here:
}
```

ADMIN_MAIN:



CODE:

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    Doctor_Delete v=new Doctor_Delete();
    v.setVisible(true);
    this.setVisible(false);           // TODO add your handling code here:
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    Doctor_Add v=new Doctor_Add();
    v.setVisible(true);
    this.setVisible(false);           // TODO add your handling code here:
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    Staff_Add v=new Staff_Add();
    v.setVisible(true);
    this.setVisible(false);           // TODO add your handling code here:
}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    Staff_Delete v=new Staff_Delete();
    v.setVisible(true);
    this.setVisible(false);           // TODO add your handling code here:
}

private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {
    Patient_View v=new Patient_View();
    v.setVisible(true);
    this.setVisible(false);           // TODO add your handling code here:
}

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
    Doctor_View v=new Doctor_View();
    v.setVisible(true);
    this.setVisible(false);           // TODO add your handling code here:
}
```

```

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
    Staff_View v=new Staff_View();
    v.setVisible(true);
    this.setVisible(false);           // TODO add your handling code here:
}

private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {
    Admin_Billing v=new Admin_Billing();
    v.setVisible(true);
    this.setVisible(false);           // TODO add your handling code here:
}

private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {
    Home v=new Home();
    v.setVisible(true);
    this.setVisible(false);           // TODO add your handling code here:
}

```

ADMIN_ADD_DOCTOR:



CODE:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
Admin_Main v=new Admin_Main();
v.setVisible(true);
this.setVisible(false); // TODO add your handling code here:
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
String n = Name.getText();
System.out.println(n);

String ph=PhoneNo.getText();
System.out.println(ph);

String g = (String)Gender.getSelectedItem();
System.out.println(g);

String dob=new java.text.SimpleDateFormat("dd-MM-yyyy").format(DOB.getDate());
System.out.println(dob);

String ag=Age.getText();
System.out.println(ag);

String e=E_Mail.getText();
System.out.println(e);

String s = (String)Specialist.getSelectedItem();
System.out.println(s);

String ad=Address.getText();
System.out.println(ad);

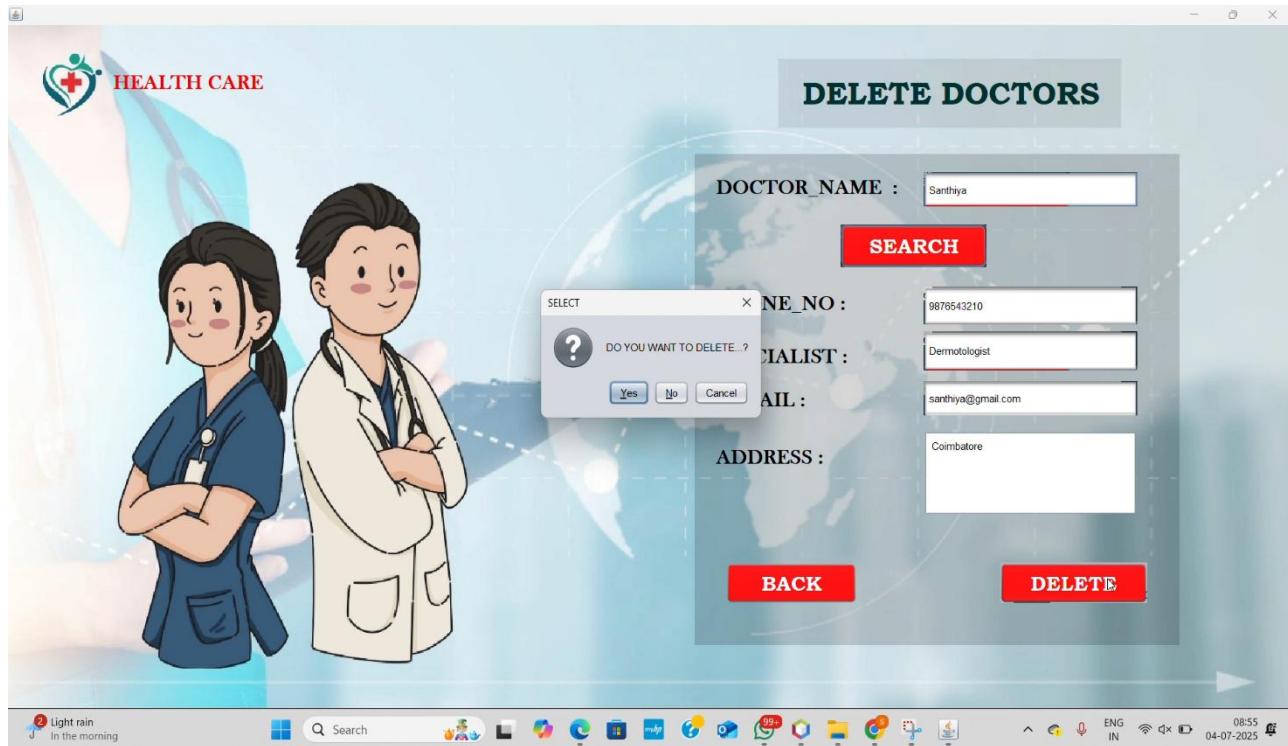
PreparedStatement ps;

String query="insert into add_doctor (Name,PhoneNo,Gender,DOB,Age,Specialist,E_Mail,Address)values(?,?,?,?,?,? ,?,?)";
try{
    ps=MyConnection1.getConnection().prepareStatement(query);
    ps.setString(1,n);
    ps.setString(2,ph);
    ps.setString(3,g);
    ps.setString(4,dob);
    ps.setString(5,ag);
    ps.setString(6,e);
    ps.setString(7,s);
    ps.setString(8,ad);

    if(ps.executeUpdate(>0){
        JOptionPane.showMessageDialog(null, "Add Doctor Details Successfully...");
    }
}
catch(Exception es){
    System.out.println(es);
}
Admin_Main v=new Admin_Main();
v.setVisible(true);
this.setVisible(false);
// TODO add your handling code here:
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
Home v=new Home();
v.setVisible(true);
this.setVisible(false); // TODO add your handling code here:
}
```

ADMIN_DELETE_DOCTOR:



CODE:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    Admin_Main v=new Admin_Main();
    v.setVisible(true);
    this.setVisible(false);           // TODO add your handling code here:
}

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

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    String s=Name.getText();
    System.out.println(s);
    PreparedStatement ps;
    ResultSet rs;
    try{
        String query="SELECT PhoneNo,E_Mail,Specialist,Address FROM add_doctor WHERE Name=?";
        ps=MyConnection1.getConnection().prepareStatement(query);
        ps.setString(1, s);
        rs=ps.executeQuery();
        if(rs.next()){
            PhoneNo.setText(rs.getString("PhoneNo"));
            E_Mail.setText(rs.getString("E_Mail"));
            Specialist.setText(rs.getString("Specialist"));
            Address.setText(rs.getString("Address"));
        }
        else{
            JOptionPane.showMessageDialog(null, "DOCTOR_NAME is INVALID!!!!");
        }
    }
    catch(Exception e){
        System.out.println(e);
    } // TODO add your handling code here:
}
```

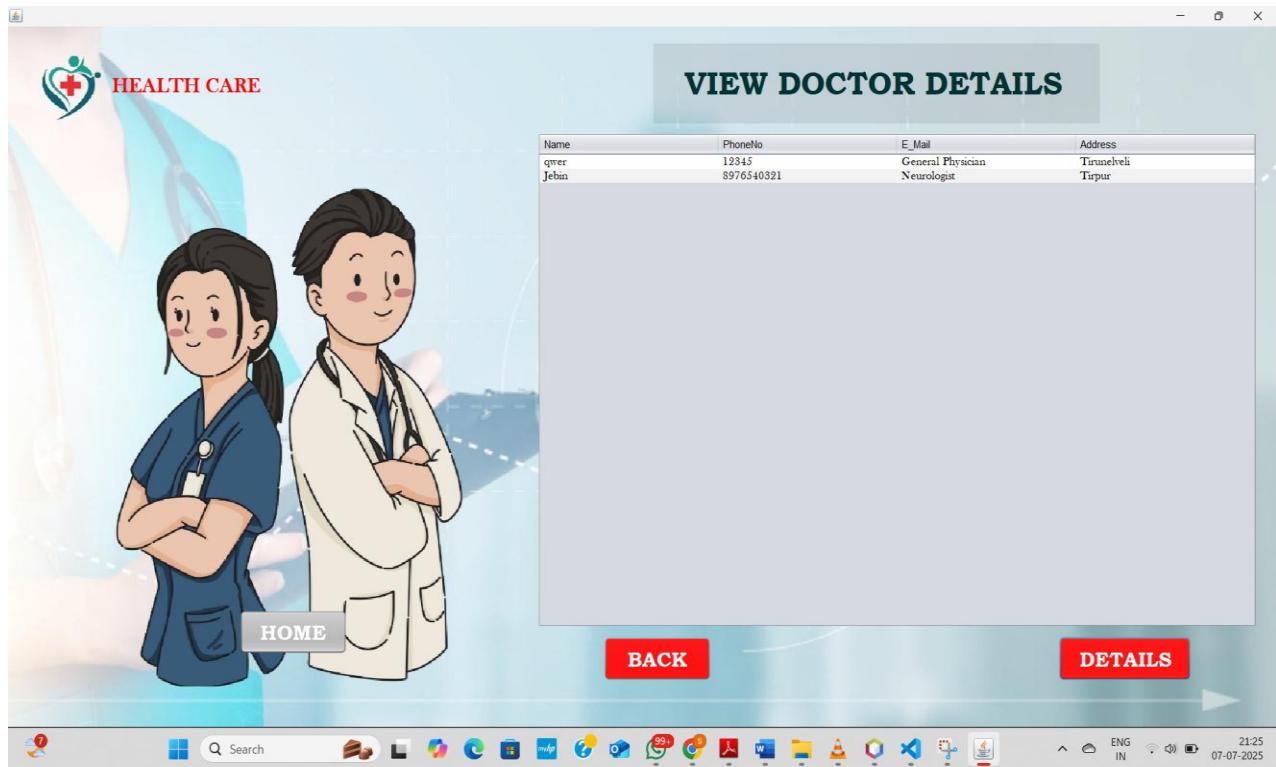
```

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    String s=Name.getText();
    int a= JOptionPane.showConfirmDialog(null, "DO YOU WANT TO DELETE...?", "SELECT", JOptionPane.YES_NO_CANCEL_OPTION);
    PreparedStatement ps;
    if(a==0){
        String query="DELETE FROM add_doctor WHERE Name=?";
        try{
            ps=MyConnection1.getConnection().prepareStatement(query);
            ps.setString(1,s);
            if(ps.executeUpdate(>0)){
                JOptionPane.showMessageDialog(null, "Doctor Details are Deleted Successfully...");
            }
            else{
                JOptionPane.showMessageDialog(null, "Doctor Details are Not Deleted");
            }
        }
        catch(Exception ex){
            System.out.println(ex);
        }
    }
    Admin_Main v=new Admin_Main();
    v.setVisible(true);
    this.setVisible(false);
    // TODO add your handling code here
}

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

```

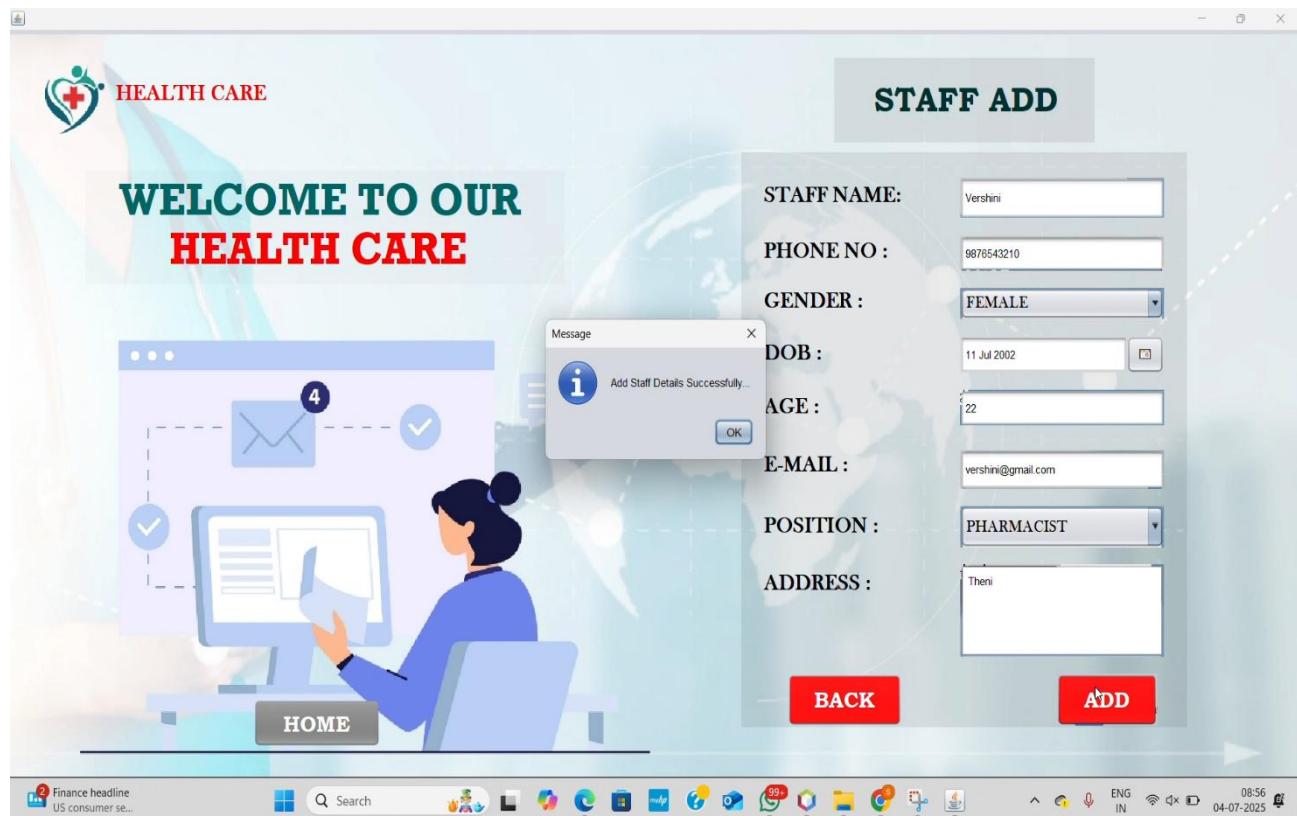
ADMIN_VIEW_DOCTORS:



CODE:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    PreparedStatement ps;  
    ResultSet rs;  
    String query="SELECT Name,PhoneNo,Specialist,Address FROM add_doctor";  
    try{  
        ps=MyConnection1.getConnection().prepareStatement(query);  
        rs=ps.executeQuery();  
        Table.setModel(DbUtils.resultSetToTableModel(rs));  
    }  
    catch(Exception e){  
        System.out.println(e);  
    }  
    // TODO add your handling code here:  
}  
  
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    Admin_Main v=new Admin_Main();  
    v.setVisible(true);  
    this.setVisible(false);  
    // TODO add your handling code here:  
}  
  
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
    Home v=new Home();  
    v.setVisible(true);  
    this.setVisible(false);  
    // TODO add your handling code here:  
}
```

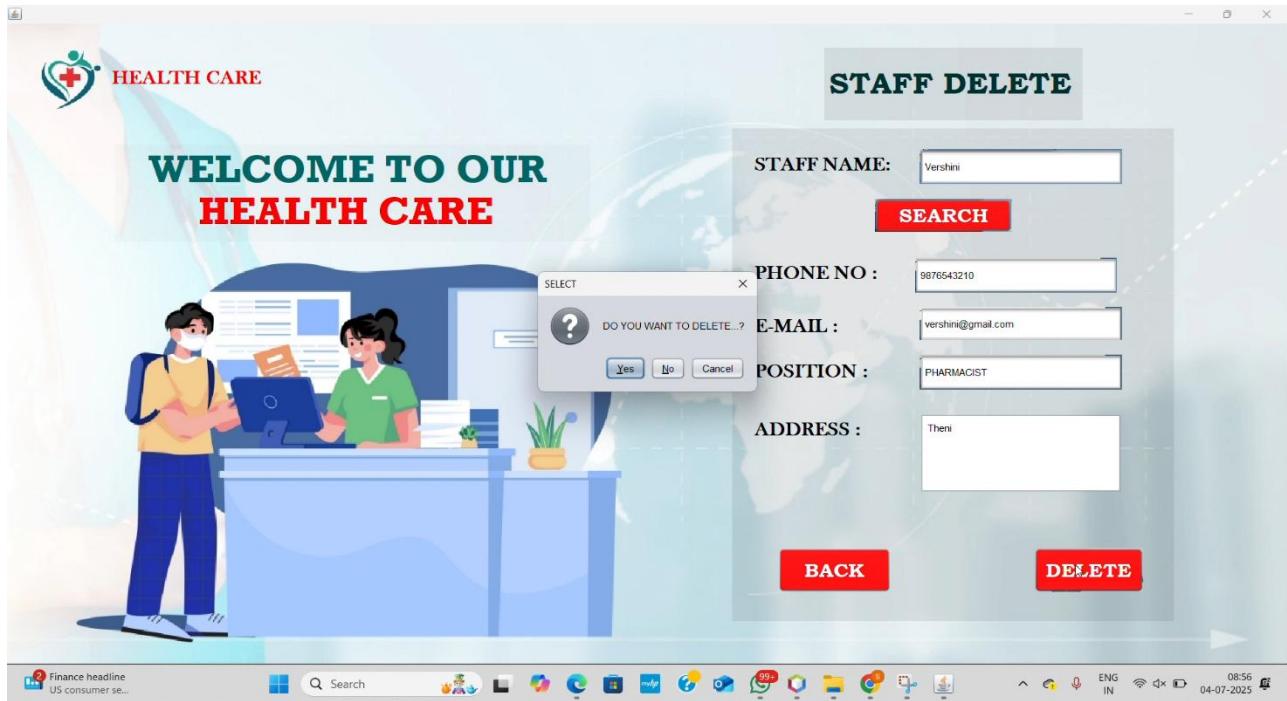
ADMIN_ADD_STAFF:



CODE:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
String n = Name.getText();  
System.out.println(n);  
  
String ph=PhoneNo.getText();  
System.out.println(ph);  
  
String g = (String)Gender.getSelectedItem();  
System.out.println(g);  
  
String dob=new java.text.SimpleDateFormat("dd-MM-yyyy").format(DOB.getDate()));  
System.out.println(dob);  
  
String ag=Age.getText();  
System.out.println(ag);  
  
String e=E_Mail.getText();  
System.out.println(e);  
  
String p = (String)Position.getSelectedItem();  
System.out.println(p);  
  
String ad=Address.getText();  
System.out.println(ad);  
  
PreparedStatement ps;  
  
String query="insert into add_staff (Name,PhoneNo,Gender,DOB,Age,E_Mail,Positions,Address)values(?,?,?,?,?, ?,?,?)";  
try{  
    ps=MyConnection1.getConnection().prepareStatement(query);  
    .....  
    ps.setString(1,n);  
    ps.setString(2,ph);  
    ps.setString(3,g);  
    ps.setString(4,dob);  
    ps.setString(5,ag);  
    ps.setString(6,e);  
    ps.setString(7,p);  
    ps.setString(8,ad);  
  
    if(ps.executeUpdate(>0){  
        JOptionPane.showMessageDialog(null, "Add Staff Details Successfully...");  
    }  
}  
catch(Exception es){  
    System.out.println(es);  
}  
Admin_Main v=new Admin_Main();  
v.setVisible(true);  
this.setVisible(false);  
// TODO add your handling code here:  
}  
  
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
Admin_Main v=new Admin_Main();  
v.setVisible(true);  
this.setVisible(false); // TODO add your handling code here:  
}  
  
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
Home v=new Home();  
v.setVisible(true);  
this.setVisible(false); // TODO add your handling code here:  
}
```

ADMIN_DELETE_STAFF:



CODE:

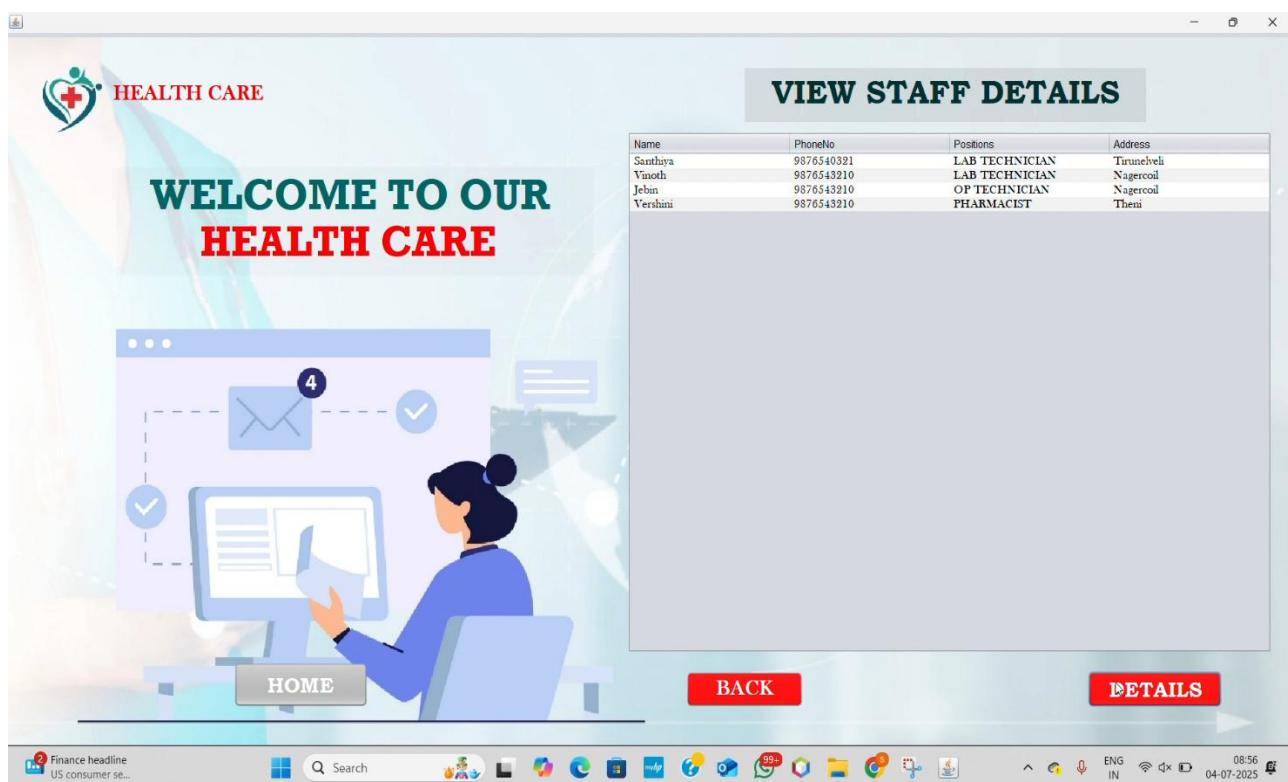
```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
    String s=Name.getText();  
    System.out.println(s);  
    PreparedStatement ps;  
    ResultSet rs;  
    try{  
        String query="SELECT PhoneNo,E_Mail,Positions,Address FROM add_staff WHERE Name=?";  
        ps=MyConnection1.getConnection().prepareStatement(query);  
        ps.setString(1, s);  
        rs=ps.executeQuery();  
        if(rs.next()){  
            PhoneNo.setText(rs.getString(1));  
            E_Mail.setText(rs.getString(2));  
            Position.setText(rs.getString(3));  
            Address.setText(rs.getString(4));  
        }  
        else{  
            JOptionPane.showMessageDialog(null, "Staff Name is INVALID!!!");  
        }  
    }  
    catch(Exception e){  
        System.out.println(e);  
    }  
    // TODO add your handling code here:  
}
```

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
String s=Name.getText();
int a=JOptionPane.showConfirmDialog(null, "DO YOU WANT TO DELETE...?", "SELECT", JOptionPane.YES_NO_CANCEL_OPTION);
PreparedStatement ps;

if(a==0){
    String query="DELETE FROM add_staff WHERE Name=?";
    try{
        ps=MyConnection1.getConnection().prepareStatement(query);
        ps.setString(1,s);
        if(ps.executeUpdate(>0)){
            JOptionPane.showMessageDialog(null, "Staff Details are Deleted Successfully...");
        }
        else{
            JOptionPane.showMessageDialog(null, "Staff Details are Not Deleted");
        }
    }
    catch(Exception ex){
        System.out.println(ex);
    }
}
Admin_Main v=new Admin_Main();
v.setVisible(true);
this.setVisible(false); // TODO add your handling code here:
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
Admin_Main v=new Admin_Main();
v.setVisible(true);
this.setVisible(false); // TODO add your handling code here:
}
```

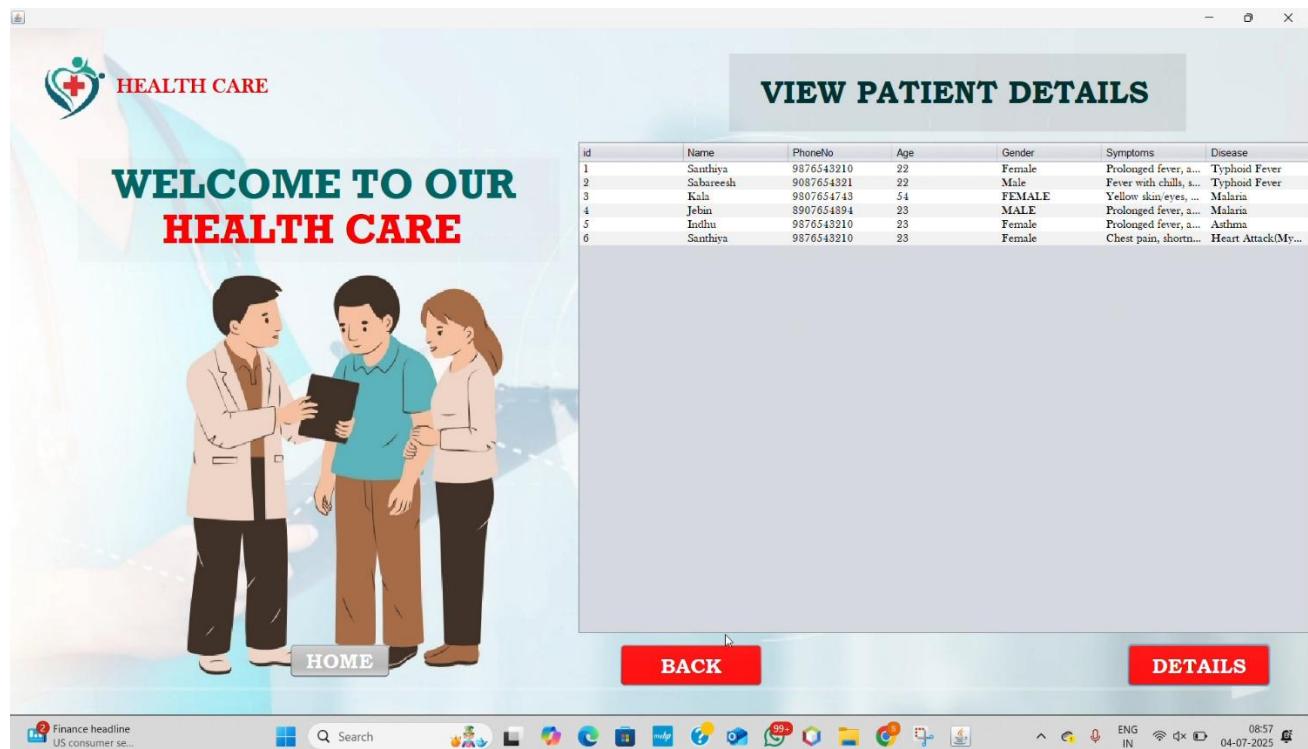
ADMIN_VIEW_STAFF:



CODE:

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
Home v=new Home();  
v.setVisible(true);  
this.setVisible(false);           // TODO add your handling code here:  
}  
  
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
Admin_Main v=new Admin_Main();  
v.setVisible(true);  
this.setVisible(false);           // TODO add your handling code here:  
}  
  
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
PreparedStatement ps;  
ResultSet rs;  
String query="SELECT Name, PhoneNo, Positions, Address FROM add_staff";  
try{  
    ps=MyConnection1.getConnection().prepareStatement(query);  
    rs=ps.executeQuery();  
    Table.setModel(DbUtils.resultSetToTableModel(rs));  
}  
catch(Exception e){  
    System.out.println(e);  
}           // TODO add your handling code here:  
}  
...  
...
```

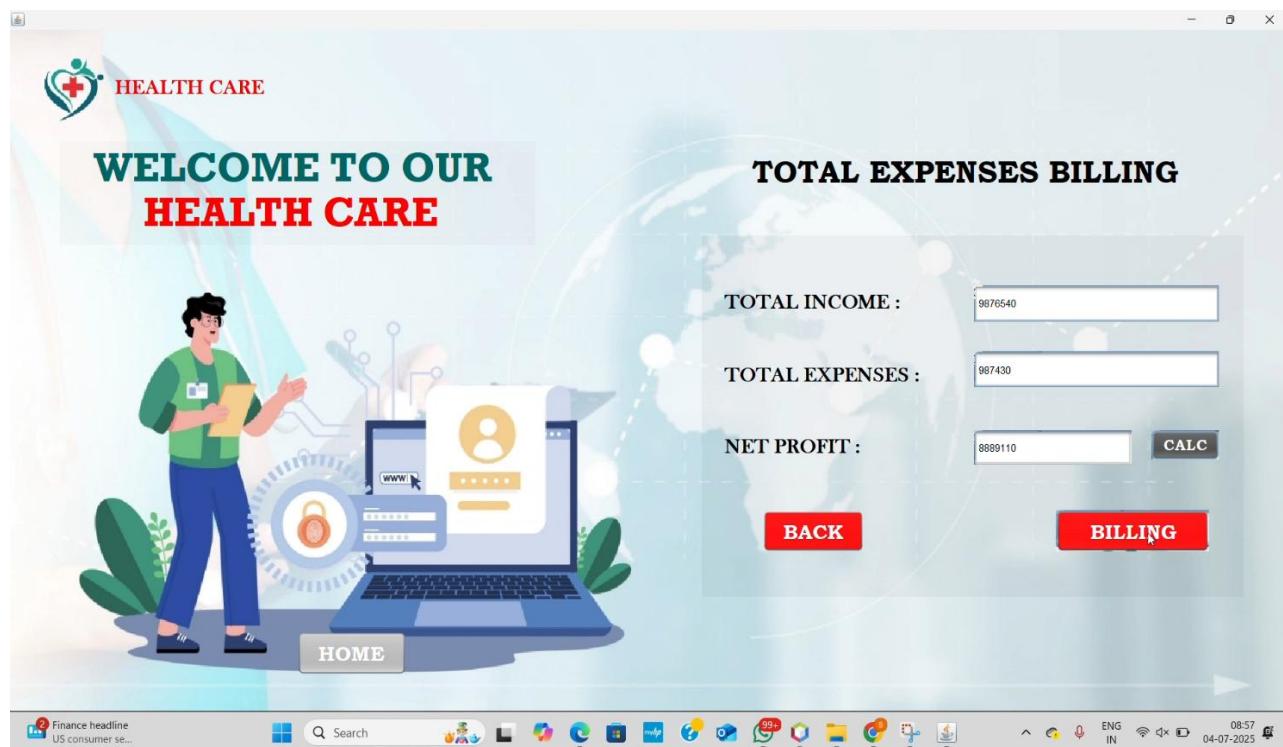
ADMIN_VIEW_PATIENT:

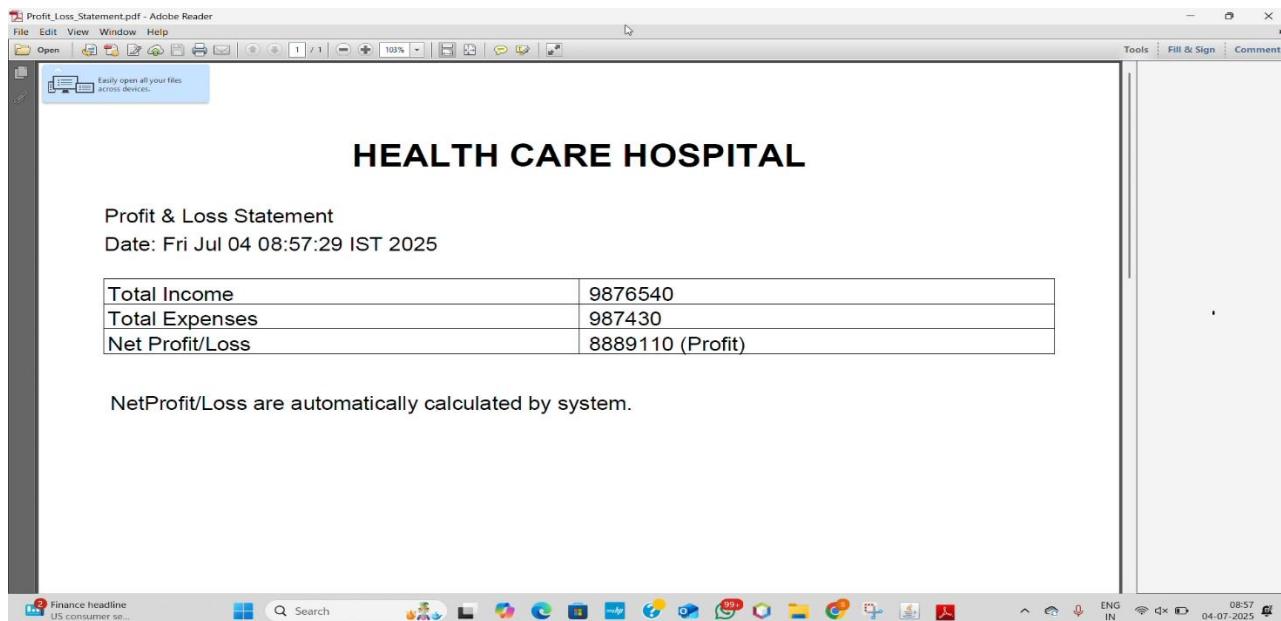


CODE:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    PreparedStatement ps;  
    ResultSet rs;  
    String query="SELECT * FROM add_patient_disease";  
    try{  
        ps=MyConnection1.getConnection().prepareStatement(query);  
        rs=ps.executeQuery();  
        Table.setModel(DbUtils.resultSetToTableModel(rs));  
    }  
    catch(Exception e){  
        System.out.println(e);  
    } // TODO add your handling code here:  
}  
  
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    Admin_Main v=new Admin_Main();  
    v.setVisible(true);  
    this.setVisible(false); // TODO add your handling code here:  
}  
  
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
    Home v=new Home();  
    v.setVisible(true);  
    this.setVisible(false); // TODO add your handling code here:  
}
```

ADMIN_CALCULATE_EXPENSES:





CODE:

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
try {
    int income = Integer.parseInt(TotalIncome.getText().trim());
    int expenses = Integer.parseInt(TotalExpenses.getText().trim());
    int profit = income - expenses;

    NetProfit.setText(String.valueOf(profit));
} catch (NumberFormatException e) {
    JOptionPane.showMessageDialog(null, "Please enter valid numbers for income and expenses.");
} // TODO add your handling code here:
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
try {
    String fileName = "Profit_Loss_Statement.pdf";

    int income = Integer.parseInt(TotalIncome.getText().trim());
    int expenses = Integer.parseInt(TotalExpenses.getText().trim());
    int profit = income - expenses;

    Document doc = new Document();
    PdfWriter.getInstance(doc, new FileOutputStream(fileName));
    doc.open();

    Paragraph title = new Paragraph("HEALTH CARE HOSPITAL",
        FontFactory.getFont(FontFactory.HELVETICA_BOLD, 20, BaseColor.BLACK));
    title.setAlignment(Element.ALIGN_CENTER);
    doc.add(title);

    doc.add(new Paragraph("\nProfit & Loss Statement"));
    doc.add(new Paragraph("Date: " + new java.util.Date().toString() + "\n\n"));

    PdfPTable table = new PdfPTable(2);
    table.setWidthPercentage(100);
}

```

```

        table.addCell("Total Income");
        table.addCell("₹ " + income);

        table.addCell("Total Expenses");
        table.addCell("₹ " + expenses);

        table.addCell("Net Profit/Loss");
        table.addCell("₹ " + profit + (profit >= 0 ? " (Profit)" : " (Loss)"));

        doc.add(table);

        doc.add(new Paragraph("\n NetProfit/Loss are automatically calculated by system."));
        doc.close();

        JOptionPane.showMessageDialog(null, "Profit & Loss PDF generated successfully.");

        // Optional: Open the PDF
        if (Desktop.isDesktopSupported()) {
            Desktop.getDesktop().open(new File(fileName));
        }

    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Error generating PDF: " + e.getMessage());
    }
}

Admin_Main v=new Admin_Main();
v.setVisible(true);
this.setVisible(false);
// TODO add your handling code here:
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    Admin_Main v=new Admin_Main();
    v.setVisible(true);
    this.setVisible(false);           // TODO add your handling code here:
}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    Home v=new Home();
    v.setVisible(true);
    this.setVisible(false);           // TODO add your handling code here:
}

```

2) DOCTOR PAGE:

DOCTOR_LOGIN:



CODE:

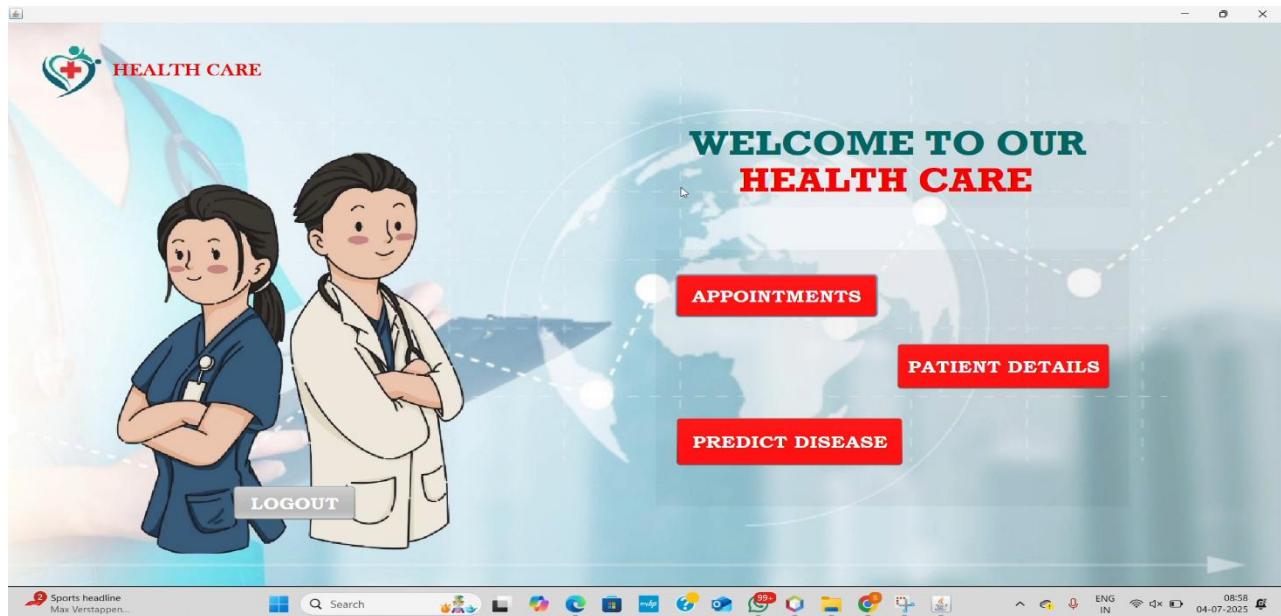
```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
String username = Doctor_Name.getText();  
String password = new String(Password.getPassword());  
  
PreparedStatement ps;  
ResultSet rs;  
  
String query = "SELECT * FROM add_doctor WHERE Name = ?";  
  
try {  
    ps = MyConnection1.getConnection().prepareStatement(query);  
    ps.setString(1, username);  
  
    rs = ps.executeQuery();  
  
    if (rs.next()) {  
        if (password.equals("Doctor123")) {  
            JOptionPane.showMessageDialog(this, "Login Successful!");  
            Doctor_Main v = new Doctor_Main();  
            v.setVisible(true);  
            this.setVisible(false);  
        } else {  
            JOptionPane.showMessageDialog(this, "Invalid password", "Login Failed", JOptionPane.ERROR_MESSAGE);  
        }  
    } else {  
        JOptionPane.showMessageDialog(this, "Username not found", "Login Failed", JOptionPane.ERROR_MESSAGE);  
    }  
}  
}  
catch (Exception ex) {  
    System.out.println(ex);  
}  
// TODO add your handling code here:  
}
```

```

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    Home v = new Home();
    v.setVisible(true);
    this.setVisible(false);           // TODO add your handling code here:
}

```

DOCTOR_MAIN:



CODE:

```

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
Patient_Details v = new Patient_Details();
    v.setVisible(true);
    this.setVisible(false);           // TODO add your handling code here:
}

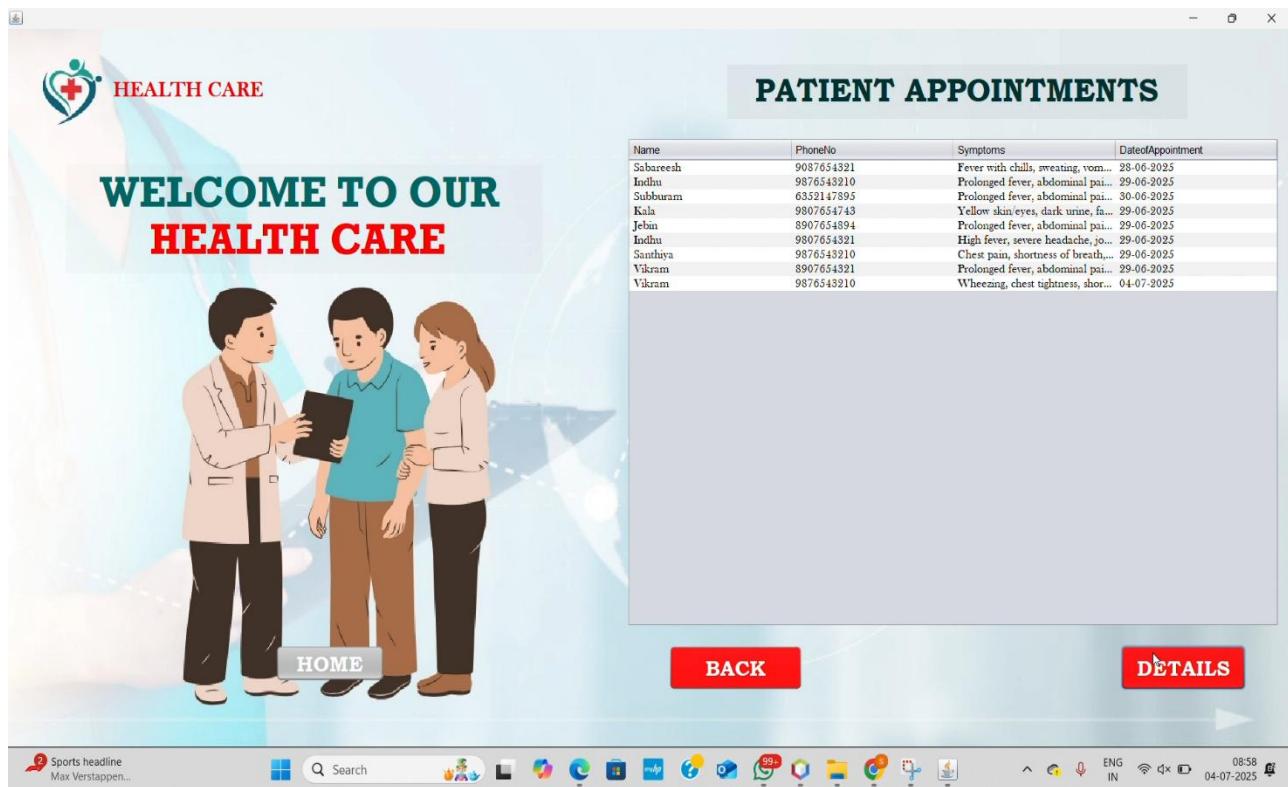
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    Home v = new Home();
    v.setVisible(true);
    this.setVisible(false);           // TODO add your handling code here:
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
Doctor_Appointment v = new Doctor_Appointment();
v.setVisible(true);
this.setVisible(false);           // TODO add your handling code here:
}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
Add_Patient_Disease v = new Add_Patient_Disease();
v.setVisible(true);
this.setVisible(false);           // TODO add your handling code here:
}

```

DOCTOR_VIEW_PATIENT_APPOINTMENT:



CODE:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    PreparedStatement ps;  
    ResultSet rs;  
    String query="SELECT Name,PhoneNo,Symptoms,DateofAppointment FROM register";  
    try{  
        ps=MyConnection1.getConnection().prepareStatement(query);  
        rs=ps.executeQuery();  
        Table.setModel(DbUtils.resultSetToTableModel(rs));  
    }  
    catch(Exception e){  
        System.out.println(e);  
    } // TODO add your handling code here:  
}  
  
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    Doctor_Main v=new Doctor_Main();  
    v.setVisible(true);  
    this.setVisible(false); // TODO add your handling code here:  
}  
  
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
    Home v=new Home();  
    v.setVisible(true);  
    this.setVisible(false); // TODO add your handling code here:  
}
```

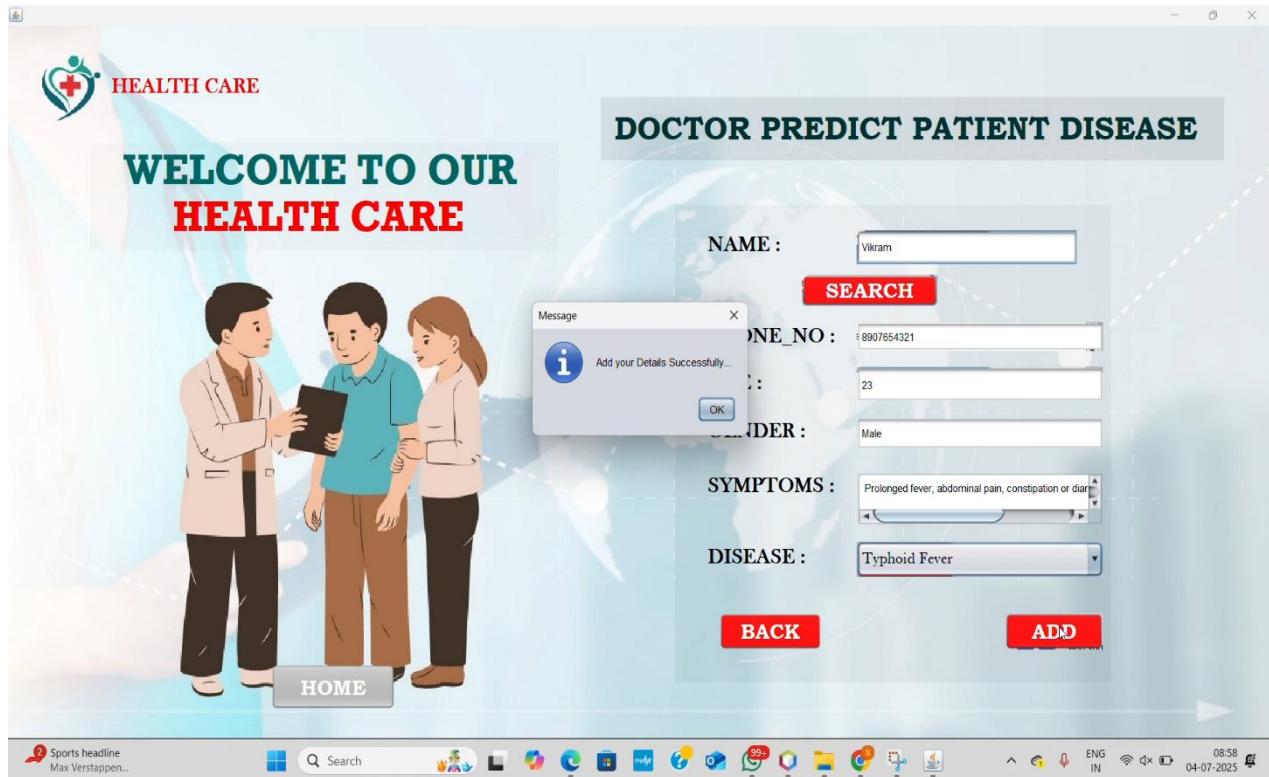
DOCTOR_VIEW_PATIENT_DETAILS:



CODE:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    PreparedStatement ps;  
    ResultSet rs;  
    String query="SELECT Name, Symptoms FROM register";  
    try{  
        ps=MyConnection1.getConnection().prepareStatement(query);  
        rs=ps.executeQuery();  
        Table.setModel(DbUtils.resultSetToTableModel(rs));  
    }  
    catch(Exception e){  
        System.out.println(e);  
    }  
    // TODO add your handling code here:  
}  
  
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    Doctor_Main v=new Doctor_Main();  
    v.setVisible(true);  
    this.setVisible(false);  
    // TODO add your handling code here:  
}
```

DOCTOR_PREDICT_PATIENT_DISEASE:



CODE:

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    String n = Name.getText();
    System.out.println(n);

    String ph=PhoneNo.getText();
    System.out.println(ph);

    String ag=Age.getText();
    System.out.println(ag);

    String g=Gender.getText();
    System.out.println(g);

    String sy=Symptoms.getText();
    System.out.println(sy);

    String d = (String)Disease.getSelectedItem();
    System.out.println(d);

    PreparedStatement ps;

    String query="insert into add_patient_disease (Name,PhoneNo,Age,Gender,Symptoms,Disease)values(?,?,?,?,?,?)";
    try{
        ps=MyConnection1.getConnection().prepareStatement(query);
        ps.setString(1,n);
        ps.setString(2,ph);
        ps.setString(3,ag);
        ps.setString(4,g);
        ps.setString(5,sy);
        ps.setString(6,d);

        if(ps.executeUpdate(>0){
            JOptionPane.showMessageDialog(null, "Add your Details Successfully...");
        }
    }
```

```

        catch(Exception es){
            System.out.println(es);
        }
        Doctor_Main v=new Doctor_Main();
        v.setVisible(true);
        this.setVisible(false);      // TODO add your handling code here:
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        String s=Name.getText();
        System.out.println(s);
        PreparedStatement ps;
        ResultSet rs;
        try{
            String query="SELECT PhoneNo, Age, Gender, Symptoms FROM register WHERE Name=?";
            ps=MyConnection1.getConnection().prepareStatement(query);
            ps.setString(1, s);
            rs=ps.executeQuery();
            if(rs.next()){
                PhoneNo.setText(rs.getString(1));
                Age.setText(rs.getString(2));
                Gender.setText(rs.getString(3));
                Symptoms.setText(rs.getString(4));
            }
            else{
                JOptionPane.showMessageDialog(null, "Patient_Name is INVALID!!!!");
            }
        }
        catch(Exception e){
            System.out.println(e);
        }           // TODO add your handling code here:
    }

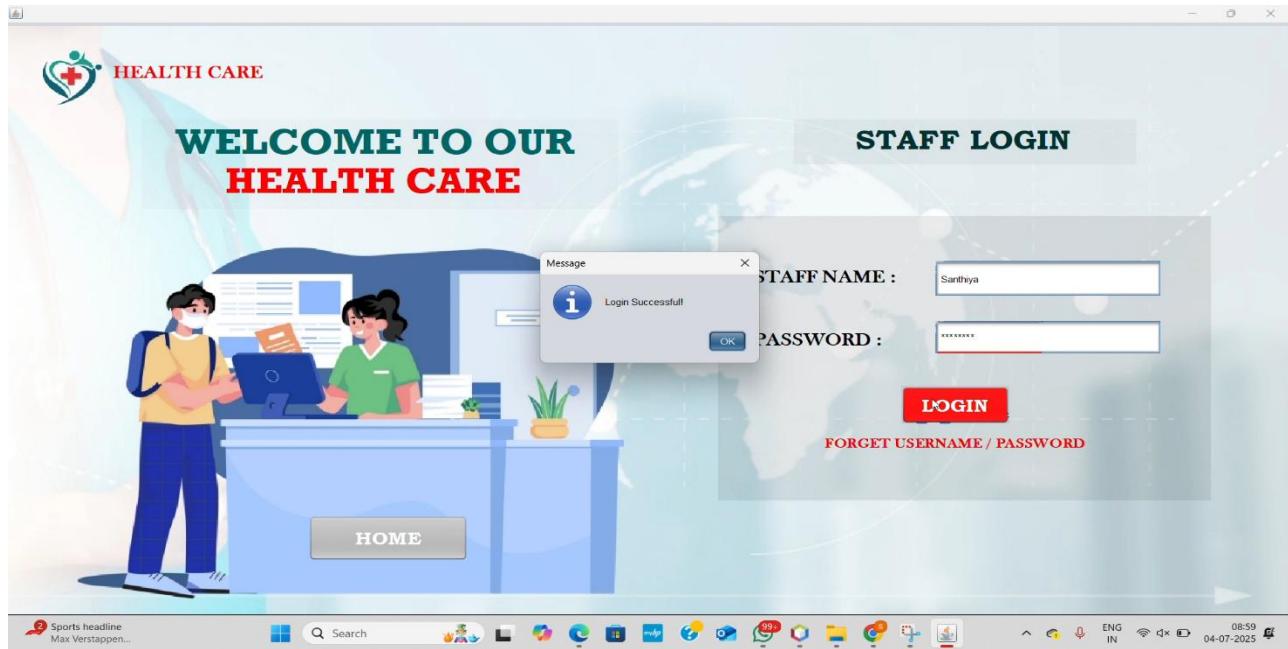
    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
Doctor_Main v=new Doctor_Main();
        v.setVisible(true);
        this.setVisible(false);      // TODO add your handling code here:
    }

    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
Home v=new Home();
        v.setVisible(true);
        this.setVisible(false);      // TODO add your handling code here:
    }

```

3) STAFF PAGE:

STAFF_LOGIN:



CODE:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
String username = Name.getText();
String password = new String(Password.getPassword());

PreparedStatement ps;
ResultSet rs;

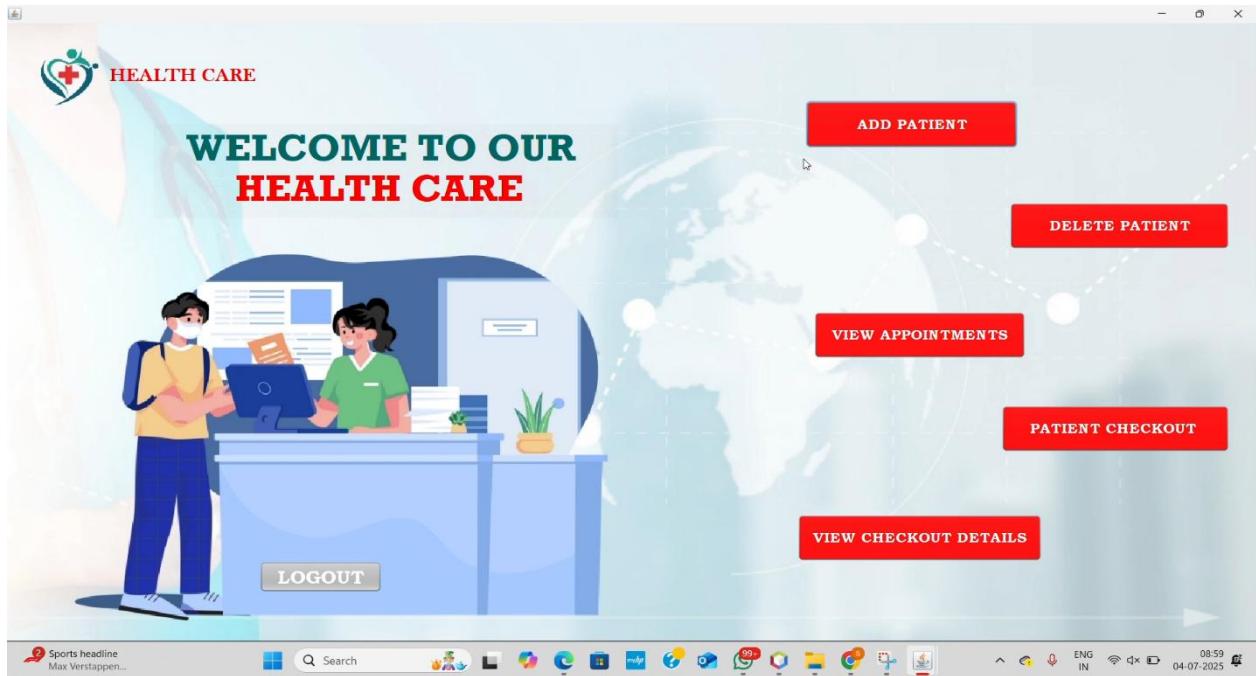
String query = "SELECT * FROM add_staff WHERE Name = ?";

try {
    ps = MyConnection1.getConnection().prepareStatement(query);
    ps.setString(1, username);

    rs = ps.executeQuery();

    if (rs.next()) {
        if (password.equals("Staff123")) {
            JOptionPane.showMessageDialog(this, "Login Successful!");
            Doctor_Main v = new Doctor_Main();
            v.setVisible(true);
            this.setVisible(false);
        } else {
            JOptionPane.showMessageDialog(this, "Invalid password", "Login Failed", JOptionPane.ERROR_MESSAGE);
        }
    } else {
        JOptionPane.showMessageDialog(this, "Username not found", "Login Failed", JOptionPane.ERROR_MESSAGE);
    }
} catch (Exception ex) {
    System.out.println(ex);
}
Staff_Main v=new Staff_Main();
v.setVisible(true);
this.setVisible(false);
// TODO add your handling code here:
}
```

STAFF_MAIN:



CODE:

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    Patient_Delete v=new Patient_Delete();  
    v.setVisible(true);  
    this.setVisible(false);           // TODO add your handling code here:  
}  
  
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
    Staff_View_Appointment v=new Staff_View_Appointment();  
    v.setVisible(true);  
    this.setVisible(false);           // TODO add your handling code here:  
}  
  
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    Patient_Add v=new Patient_Add();  
    v.setVisible(true);  
    this.setVisible(false);           // TODO add your handling code here:  
}  
  
private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {  
    Home v=new Home();  
    v.setVisible(true);  
    this.setVisible(false);           // TODO add your handling code here:  
}  
  
private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {  
    Patient_Checkout v=new Patient_Checkout();  
    v.setVisible(true);  
    this.setVisible(false);           // TODO add your handling code here:  
}  
  
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {  
    Checkout_Details v=new Checkout_Details();  
    v.setVisible(true);  
    this.setVisible(false);           // TODO add your handling code here:  
}
```

STAFF_ADD_PATIENT:



CODE:

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
String n = Patient_Name.getText();  
System.out.println(n);  
  
String ag = Age.getText();  
System.out.println(ag);  
  
String g = (String)Gender.getSelectedItem();  
System.out.println(g);  
  
String ph=PhoneNo.getText();  
System.out.println(ph);  
  
String e=E_Mail.getText();  
System.out.println(e);  
  
String ad=Address.getText();  
System.out.println(ad);  
  
String date = new java.text.SimpleDateFormat("dd-MM-yyyy").format(DateofAppointment.getDate());  
System.out.println(date);  
  
String sy = (String)Symptoms.getSelectedItem();  
System.out.println(sy);  
  
PreparedStatement ps;  
  
if (n.isEmpty() || ag.isEmpty() || ph.isEmpty() || e.isEmpty() || ad.isEmpty()) {  
JOptionPane.showMessageDialog(null,  
"Name, Age, Phone Number, E-Mail, and Address are required fields.\nPlease fill them out.",  
"Validation Error",  
JOptionPane.WARNING_MESSAGE  
);  
return; // Stop further execution  
}
```

```

String query="insert into register (Name,Age,Gender,PhoneNo,E_Mail,Address,DateofAppointment,Symptoms)values(?, ?, ?, ?, ?, ?, ?, ?)";
try{
    ps=MyConnection1.getConnection().prepareStatement(query);
    ps.setString(1,n);
    ps.setString(2,ag);
    ps.setString(3,g);
    ps.setString(4,ph);
    ps.setString(5,e);
    ps.setString(6,ad);
    ps.setString(7,date);
    ps.setString(8,sy);

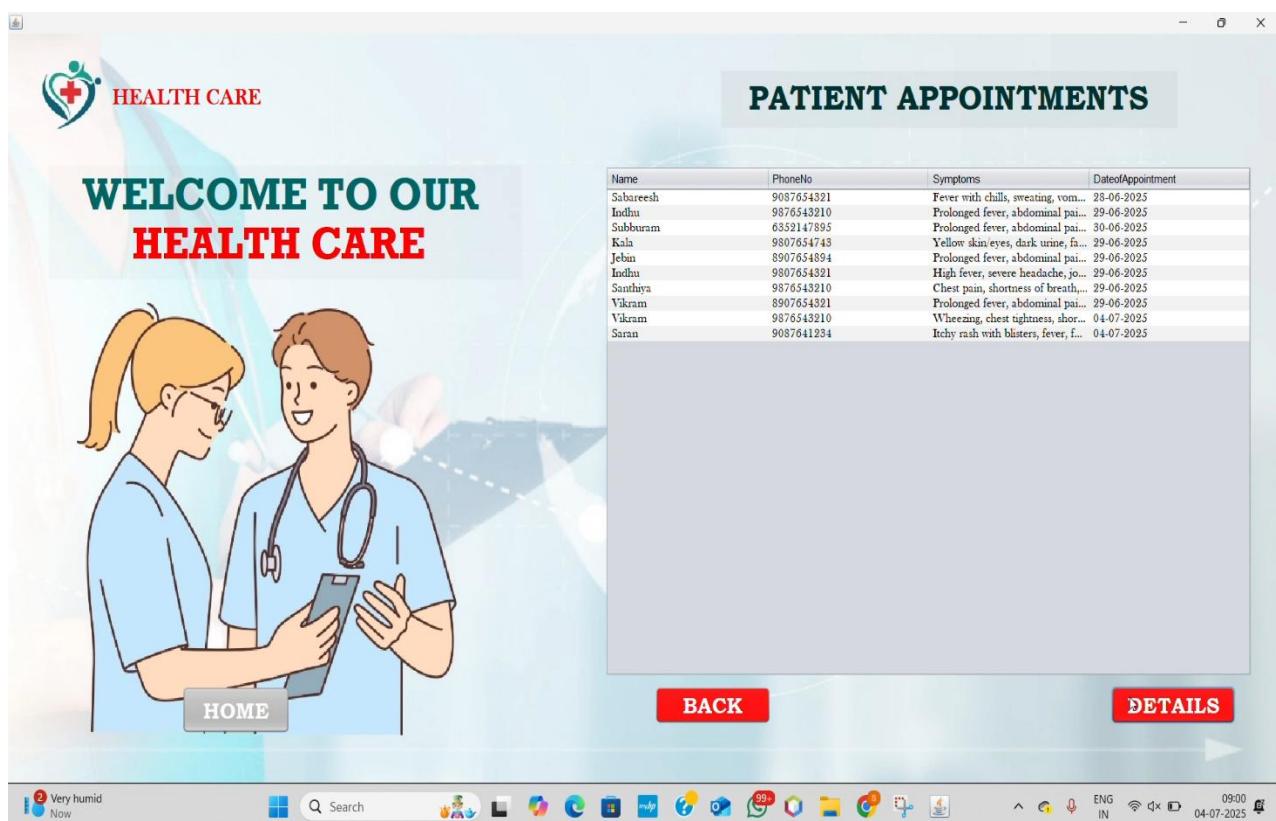
    if(ps.executeUpdate(>0){
        JOptionPane.showMessageDialog(null, "Registration done Successfully");
    }
}
catch(Exception ex){
    System.out.println(ex);
}
Staff_Main v=new Staff_Main();
v.setVisible(true);
this.setVisible(false); // TODO add your handling code here:
}

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

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
Staff_Main v=new Staff_Main();
v.setVisible(true);
this.setVisible(false); // TODO add your handling code here:
}

```

STAFF_VIEW_PATIENT:



CODE:

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
    Home v=new Home();  
    v.setVisible(true);  
    this.setVisible(false);           // TODO add your handling code here:  
}  
  
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    PreparedStatement ps;  
    ResultSet rs;  
    String query="SELECT Name,PhoneNo,Symptoms,DateofAppointment FROM register";  
    try{  
        ps=MyConnection1.getConnection().prepareStatement(query);  
        rs=ps.executeQuery();  
        Table.setModel(DbUtils.resultSetToTableModel(rs));  
    }  
    catch(Exception e){  
        System.out.println(e);  
    }           // TODO add your handling code here:  
}  
  
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    Staff_Main v=new Staff_Main();  
    v.setVisible(true);  
    this.setVisible(false);           // TODO add your handling code here:  
}
```

STAFF_DELETE_PATIENT:



CODE:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
String s=Name.getText();
System.out.println(s);
PreparedStatement ps;
ResultSet rs;
try{
    String query="SELECT PhoneNo,Gender,Address,E_Mail,Symptoms,DateofAppointment FROM register WHERE Name=?";
    ps=MyConnection1.getConnection().prepareStatement(query);
    ps.setString(1, s);
    rs=ps.executeQuery();
    if(rs.next()){
        PhoneNo.setText(rs.getString(1));
        Gender.setText(rs.getString(2));
        Address.setText(rs.getString(3));
        E_Mail.setText(rs.getString(4));
        Symptoms.setText(rs.getString(5));
        DateofAppointment.setText(rs.getString(6));
    }
    else{
        JOptionPane.showMessageDialog(null, "NAME is INVALID!!!");
    }
}
catch(Exception e){
    System.out.println(e);
} // TODO add your handling code here:
}

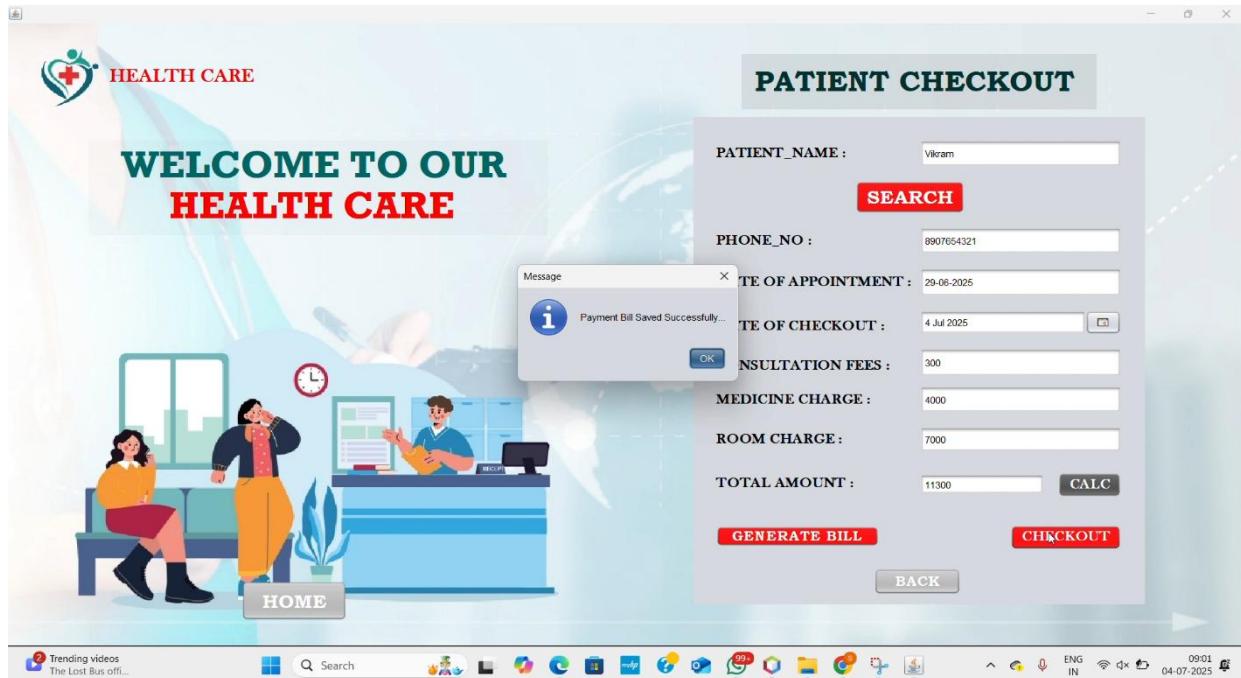
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
String s=Name.getText();
int a=JOptionPane.showConfirmDialog(null, "DO YOU WANT TO DELETE...?", "SELECT", JOptionPane.YES_NO_CANCEL_OPTION);
PreparedStatement ps;

if(a==0){
    String query="DELETE FROM register WHERE Name=?";
    try{
        ps=MyConnection1.getConnection().prepareStatement(query);
        ps.setString(1,s);
        if(ps.executeUpdate(>0){
            JOptionPane.showMessageDialog(null, "Details are Deleted Successfully...");
        }
        else{
            JOptionPane.showMessageDialog(null, "Details are Not Deleted");
        }
    }
    catch(Exception ex){
        System.out.println(ex);
    }
}
Staff_Main v=new Staff_Main();
v.setVisible(true);
this.setVisible(false); // TODO add your handling code here:
}

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

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
Staff_Main v=new Staff_Main();
v.setVisible(true);
this.setVisible(false); // TODO add your handling code here:
}
```

STAFF_CHECKOUT_PATIENT:



CODE:

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
try {  
    // 1. Get TextField Values  
    String n = Name.getText().trim();  
    String ph = PhoneNo.getText().trim();  
    String ag = DateofAppointment.getText().trim(); // example: 26-06-2025  
    Date checkoutDate = DateofCheckout.getDate(); // from JDateChooser  
    String sy = Consultationfees.getText().trim();  
    String mc = MedicalCharge.getText().trim();  
    String rc = RoomCharge.getText().trim();  
    String ta = TotalAmount.getText().trim();  
  
    // 2. Convert "dd-MM-yyyy" String to java.sql.Date  
    SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy");  
    Date parsedDate = sdf.parse(ag);  
    java.sql.Date sqlAppDate = new java.sql.Date(parsedDate.getTime());  
  
    // 3. Convert JDateChooser Date to java.sql.Date  
    java.sql.Date sqlCheckOutDate = new java.sql.Date(checkoutDate.getTime());  
  
    // 4. Prepare and execute SQL Insert  
    String query = "INSERT INTO patient_checkout "  
        + "(Name, PhoneNo, DateofAppointment, DateofCheckout, Consultationfees, MedicalCharge, Roomcharge, TotalAmount) "  
        + "VALUES (?, ?, ?, ?, ?, ?, ?, ?)";  
  
    PreparedStatement ps = MyConnection1.getConnection().prepareStatement(query);  
    ps.setString(1, n);  
    ps.setString(2, ph);  
    ps.setDate(3, sqlAppDate); // set DATE properly  
    ps.setDate(4, sqlCheckOutDate); // set DATE properly  
    ps.setInt(5, Integer.parseInt(sy));  
    ps.setInt(6, Integer.parseInt(mc));  
    ps.setInt(7, Integer.parseInt(rc));  
    ps.setInt(8, Integer.parseInt(ta));
```

```

        if (ps.executeUpdate() > 0) {
            JOptionPane.showMessageDialog(null, "Payment Bill Saved Successfully...");
        }

    } catch (java.text.ParseException pe) {
        JOptionPane.showMessageDialog(null, "Invalid date format! Use dd-MM-yyyy.");
    } catch (SQLException se) {
        JOptionPane.showMessageDialog(null, "SQL Error: " + se.getMessage());
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Unexpected Error: " + e.getMessage());
    } // TODO add your handling code here:
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
String s=Name.getText();
System.out.println(s);
PreparedStatement ps;
ResultSet rs;
try{
    String query="SELECT PhoneNo,DateofAppointment FROM register WHERE Name=?";
    ps=MyConnection1.getConnection().prepareStatement(query);
    ps.setString(1, s);
    rs=ps.executeQuery();
    if(rs.next()){
        PhoneNo.setText(rs.getString(1));
        DateofAppointment.setText(rs.getString(2));
    }
    else{
        JOptionPane.showMessageDialog(null, "Name is INVALID!!!!");
    }
}
catch(Exception e){
    System.out.println(e);
} // TODO add your handling code here:
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
try {
    int cf = Integer.parseInt(Consultationfees.getText().trim());
    int mc = Integer.parseInt(MedicalCharge.getText().trim());
    int rc = Integer.parseInt(RoomCharge.getText().trim());

    int total = cf + mc + rc;
    TotalAmount.setText(String.valueOf(total));
} catch (NumberFormatException e) {
    JOptionPane.showMessageDialog(null, "Please enter valid numeric values!");
} // TODO add your handling code here:
}

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
try {
    // 1. Prepare file name
    String fileName = "Patient_Bill_" + Name.getText().trim() + ".pdf";

    // 2. Create a new PDF document
    Document doc = new Document();
    PdfWriter.getInstance(doc, new FileOutputStream(fileName));
    doc.open();

    // 3. Add hospital header
    Paragraph header = new Paragraph("HEALTH CARE HOSPITAL", FontFactory.getFont(FontFactory.HELVETICA_BOLD, 20, BaseColor.BLACK));
    header.setAlignment(Element.ALIGN_CENTER);
    doc.add(header);
    doc.add(new Paragraph("Patient Billing Summary\n\n"));

    // 4. Create billing table
    PdfPTable table = new PdfPTable(2);
    table.setWidthPercentage(100);

    table.addCell("Patient Name");
    table.addCell(Name.getText());
}

```

```

table.addCell("Phone No");
table.addCell(PhoneNo.getText());

table.addCell("Date of Appointment");
table.addCell(DateofAppointment.getText());

table.addCell("Date of Checkout");
table.addCell(new java.text.SimpleDateFormat("dd-MM-yyyy").format(DateofCheckout.getDate()));

table.addCell("Consultation Fees");
table.addCell(Consultationfees.getText());

table.addCell("Medicine Charge");
table.addCell(MedicalCharge.getText());

table.addCell("Room Charge");
table.addCell(RoomCharge.getText());

table.addCell("Total Amount");
table.addCell(TotalAmount.getText());

// 5. Add table and thank you message
doc.add(table);
doc.add(new Paragraph("\nThank you for visiting our hospital.\nYou will Get well soon!"));

// 6. Close the document
doc.close();

// 7. Show message and open file
JOptionPane.showMessageDialog(null, "Bill PDF generated successfully as: " + fileName);

if (Desktop.isDesktopSupported()) {
    Desktop.getDesktop().open(new File(fileName));
}

} catch (Exception e) {
    JOptionPane.showMessageDialog(null, "Error generating bill: " + e.getMessage());
}
Staff_Main v=new Staff_Main();
v.setVisible(true);
this.setVisible(false);

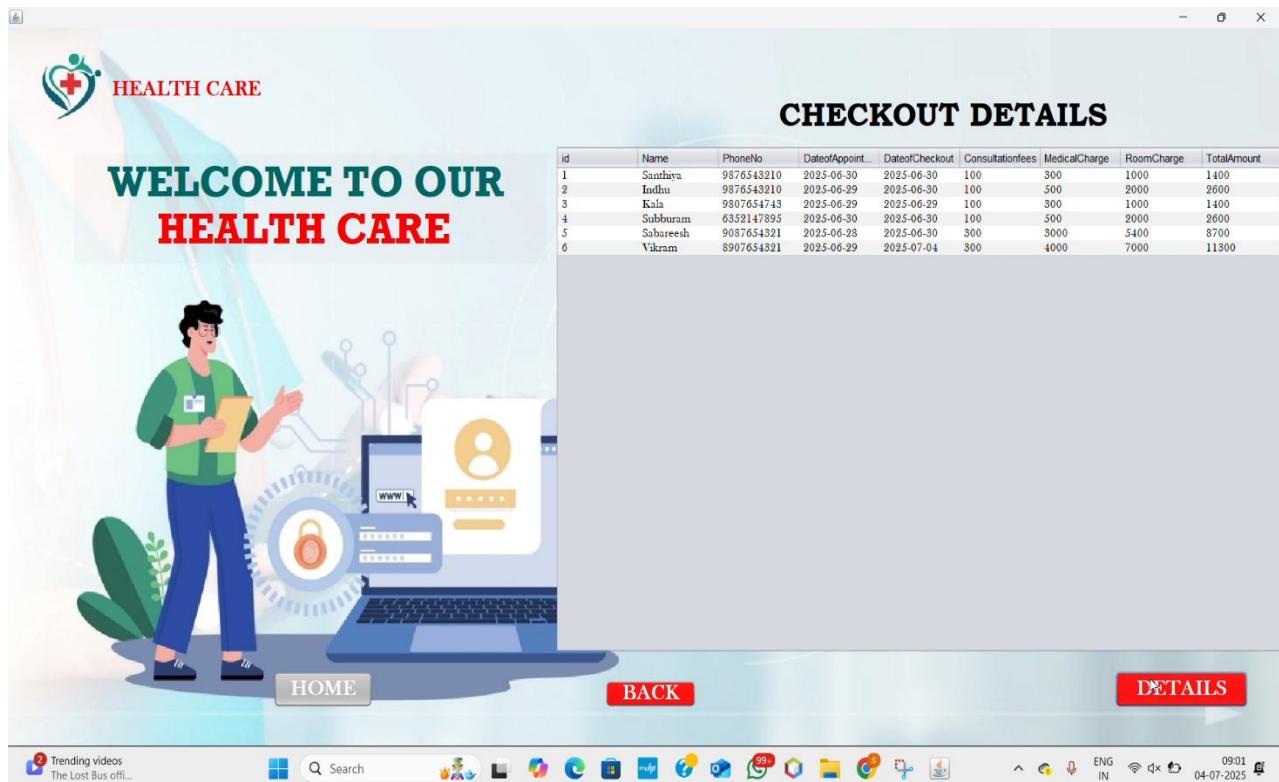
// TODO add your handling code here:
}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
Home v=new Home();
v.setVisible(true);
this.setVisible(false);           // TODO add your handling code here:
}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
Staff_Main v=new Staff_Main();
v.setVisible(true);
this.setVisible(false);           // TODO add your handling code here:
}

```

STAFF_VIEW_CHECKOUT_DETAILS:

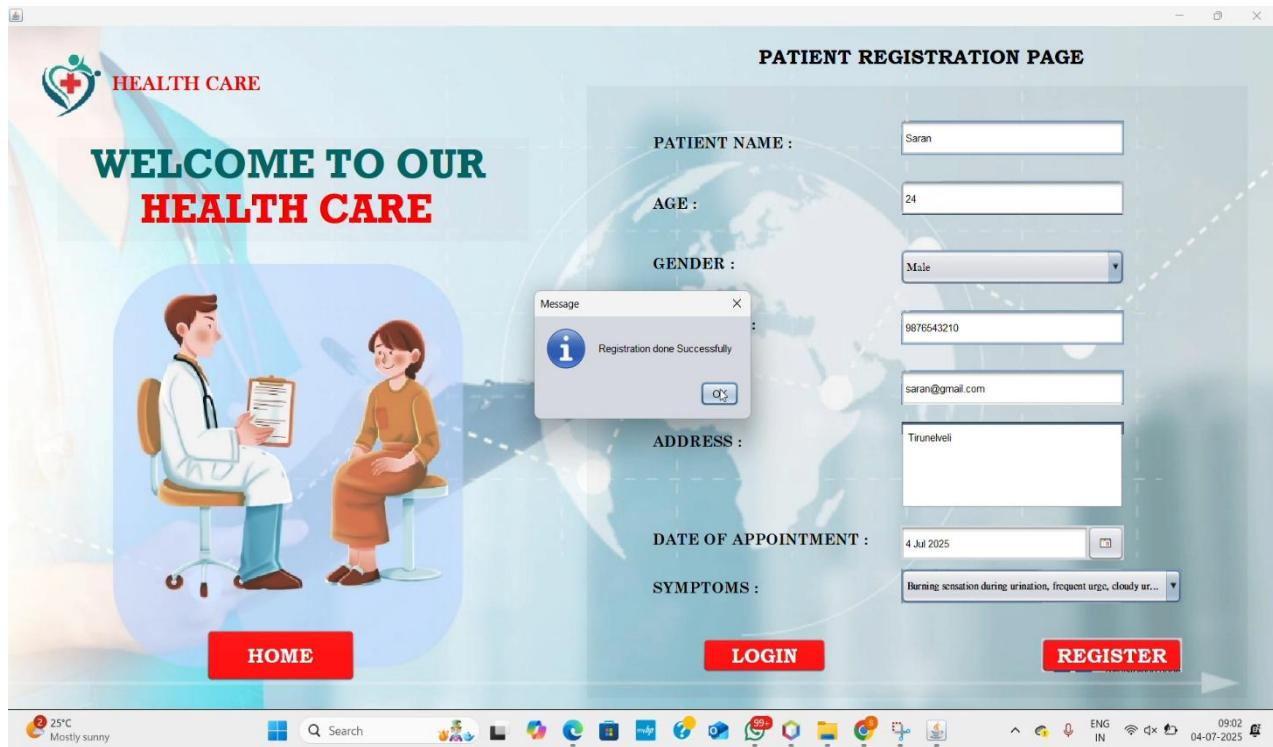


CODE:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    PreparedStatement ps;  
    ResultSet rs;  
    String query="SELECT * FROM patient_checkout";  
    try{  
        ps=MyConnection1.getConnection().prepareStatement(query);  
        rs=ps.executeQuery();  
        Table.setModel(DbUtils.resultSetToTableModel(rs));  
    }  
    catch(Exception e){  
        System.out.println(e);  
    } // TODO add your handling code here:  
}  
  
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    Staff_Main v=new Staff_Main();  
    v.setVisible(true);  
    this.setVisible(false); // TODO add your handling code here:  
}  
  
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
    Home v=new Home();  
    v.setVisible(true);  
    this.setVisible(false); // TODO add your handling code here:  
}
```

4) PATIENT PAGE:

PATIENT_REGISTRATION:



CODE:

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    String n = Name.getText();  
    System.out.println(n);  
  
    String ag = Age.getText();  
    System.out.println(ag);  
  
    String g = (String)Gender.getSelectedItem();  
    System.out.println(g);  
  
    String ph=Phone_No.getText();  
    System.out.println(ph);  
  
    String e=E_Mail.getText();  
    System.out.println(e);  
  
    String ad=Address.getText();  
    System.out.println(ad);  
  
    String date = new java.text.SimpleDateFormat("dd-MM-yyyy").format(DateofAppointment.getDate());  
    System.out.println(date);  
  
    String sy = (String)Symptoms.getSelectedItem();  
    System.out.println(sy);  
  
    PreparedStatement ps;  
  
    if (n.isEmpty() || ag.isEmpty() || ph.isEmpty() || e.isEmpty() || ad.isEmpty()) {  
        JOptionPane.showMessageDialog(null,  
            "Name, Age, Phone Number, E-Mail, and Address are required fields.\nPlease fill them out.",  
            "Validation Error",  
            JOptionPane.WARNING_MESSAGE  
    };  
    return; // Stop further execution  
}
```

```

String query="insert into register (Name,Age,Gender,PhoneNo,E_Mail,Address,DateofAppointment,Symptoms)values(?,?,?,?,?,?,?,?,?)";
try{
    ps=MyConnection1.getConnection().prepareStatement(query);
    ps.setString(1,n);
    ps.setString(2,ag);
    ps.setString(3,g);
    ps.setString(4,ph);
    ps.setString(5,e);
    ps.setString(6,ad);
    ps.setString(7,date);
    ps.setString(8,sy);

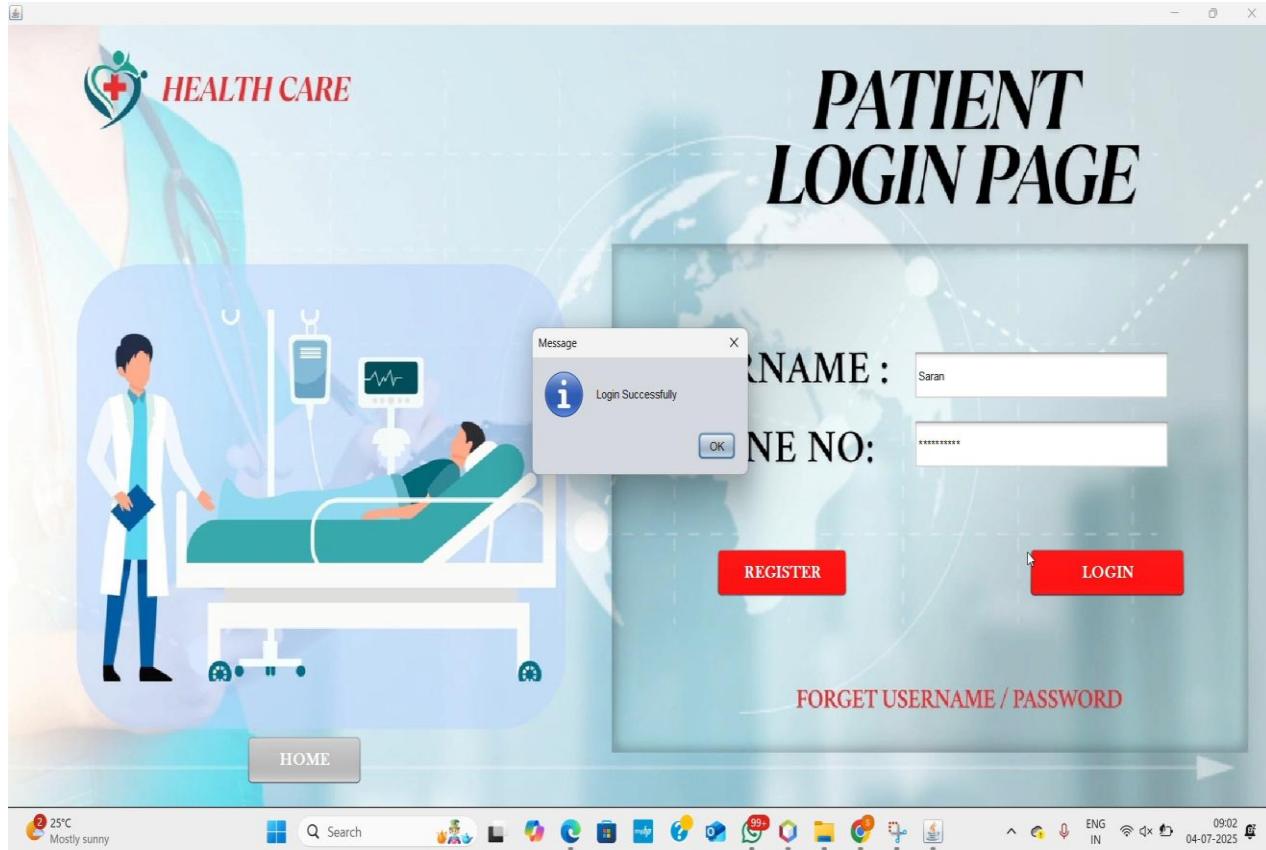
    if(ps.executeUpdate(>0){
        JOptionPane.showMessageDialog(null, "Registration done Successfully");
    }
}
catch(Exception ex){
    System.out.println(ex);
}
Patient_Login v=new Patient_Login();
v.setVisible(true);
this.setVisible(false);
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
Patient_Login v=new Patient_Login();
v.setVisible(true);
this.setVisible(false);           // TODO add your handling code here:
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
Home v=new Home();
v.setVisible(true);
this.setVisible(false);           // TODO add your handling code here:
}

```

PATIENT_LOGIN:



CODE:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    String n = Name.getText();
    String p = new String(Password.getPassword());

    PreparedStatement ps;
    ResultSet rs;

    String query = "SELECT * FROM register WHERE Name = ? AND PhoneNo = ?";

    try {
        ps = MyConnection1.getConnection().prepareStatement(query);
        ps.setString(1, n);
        ps.setString(2, p);

        rs = ps.executeQuery();

        if (rs.next()) {
            // Save to session
            Session.name = n;
            Session.phoneno = p;
            JOptionPane.showMessageDialog(null, "Login Successfully");

            // Open appointment view
            Patient_Appointment pa = new Patient_Appointment();
            pa.setVisible(true);
            this.setVisible(false);
        } else {
            JOptionPane.showMessageDialog(null, "Invalid Name or Phone Number.");
        }
    } catch (Exception e) {
        System.out.println(e);
    }

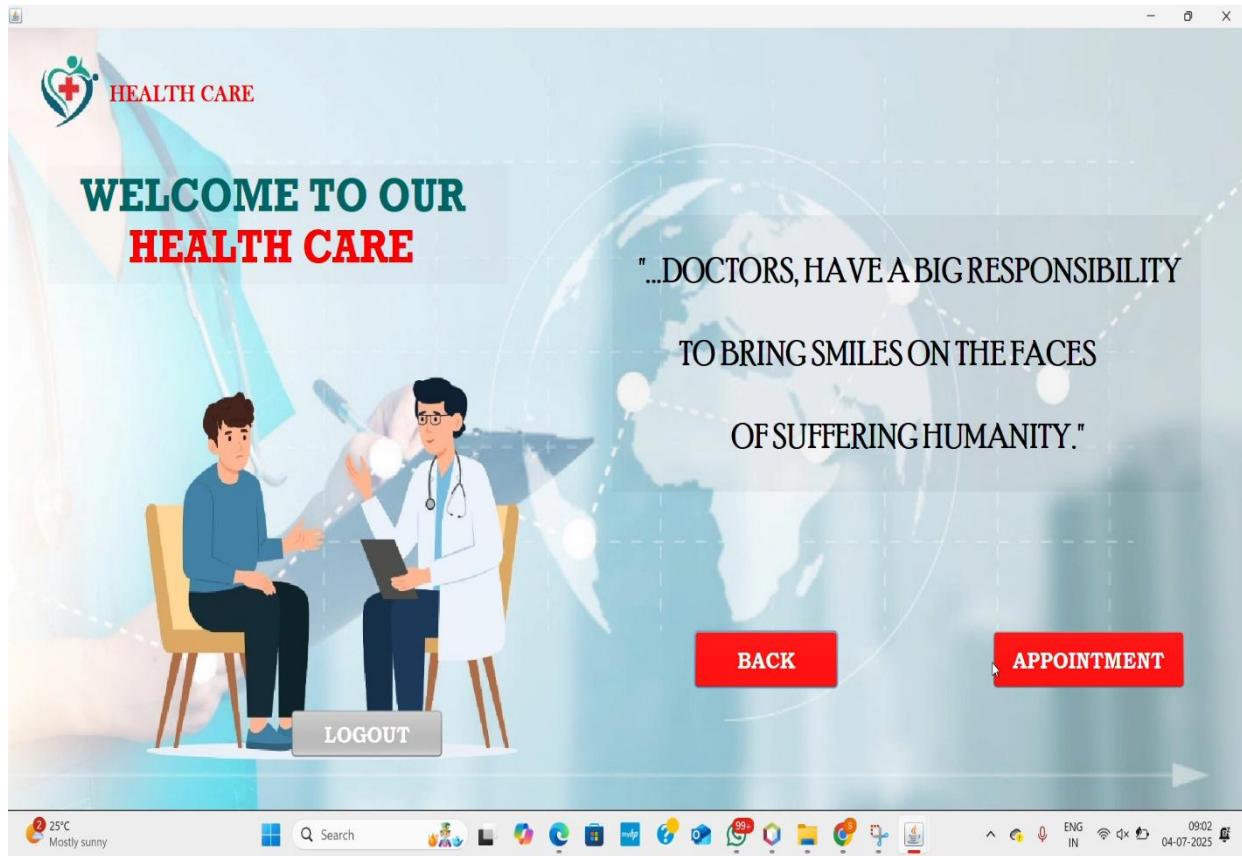
    Patient_Main v=new Patient_Main();
    v.setVisible(true);
    this.setVisible(false);

    // TODO add your handling code here:
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    Patient_Registration v=new Patient_Registration();
    v.setVisible(true);
    this.setVisible(false);           // TODO add your handling code here:
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    Home v=new Home();
    v.setVisible(true);
    this.setVisible(false);           // TODO add your handling code here:
}
```

PATIENT_MAIN:



CODE:

```
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {  
    Patient_Appointment v=new Patient_Appointment();  
    v.setVisible(true);  
    this.setVisible(false);           // TODO add your handling code here:  
}  
  
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
    Patient_Login v=new Patient_Login();  
    v.setVisible(true);  
    this.setVisible(false);           // TODO add your handling code here:  
}  
  
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    Home v=new Home();  
    v.setVisible(true);  
    this.setVisible(false);           // TODO add your handling code here:  
}
```

PATIENT_VIEW_APPOINTMENT:



CODE:

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    PreparedStatement ps;  
    ResultSet rs;  
  
    String query = "SELECT * FROM register WHERE Name = ? AND PhoneNo = ?";  
  
    try {  
        ps = MyConnection1.getConnection().prepareStatement(query);  
        ps.setString(1, Session.name);  
        ps.setString(2, Session.phoneno);  
  
        rs = ps.executeQuery();  
        Table.setModel(DbUtils.resultSetToTableModel(rs));  
    } catch (Exception e) {  
        System.out.println(e);  
    }  
    // TODO add your handling code here:  
}  
  
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    Home v=new Home();  
    v.setVisible(true);  
    this.setVisible(false); // TODO add your handling code here:  
}  
  
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
    Patient_Main v=new Patient_Main();  
    v.setVisible(true);  
    this.setVisible(false); // TODO add your handling code here:  
}
```

CONCLUSION:

The Hospital Management System is a user-friendly desktop application built using Java Swing. It aims to streamline hospital operations and reduce manual workload. The project handles core hospital functions like patient registration, doctor management, diagnosis entry, billing, and discharge. The system supports multiple user roles (Admin and Doctor) with role-based access control. Its GUI interface makes it easy for staff to interact and perform their tasks efficiently. The use of Java Swing allows for a responsive and modular design. The application improves data accuracy and ensures faster decision-making. It minimizes paperwork and organizes medical records effectively. Admins can easily monitor doctors and manage patient histories. Doctors can view and update patient details and diagnoses quickly. Billing features provide accurate and quick invoice generation. The system is scalable and can be extended with additional modules. It is suitable for small to mid-sized hospitals or clinics. The project also provides a solid foundation for students learning Java GUI development. With improvements, it can be integrated with databases, reports, or even online services. Overall, the system is secure, structured, and practical. It boosts operational efficiency and enhances patient care. This project demonstrates good software engineering practices in Java. It is ideal as both a working tool and an academic project.

**THANK
YOU!!!**