

# HOSPITAL MANAGEMENT SYSTEM

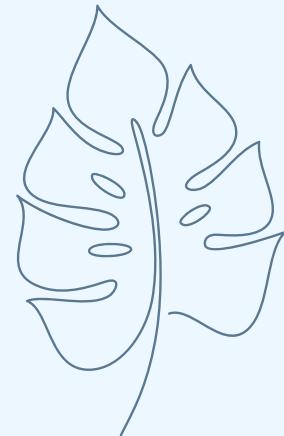
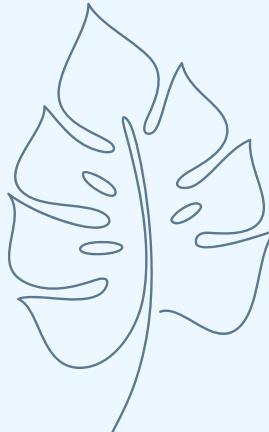
SAKTHI MS 220701240

SHANMUGA PRIYA RAANJANI SH 220701262

# Overview

- ★ OBJECTIVE
- ★ ABSTRACT
- ★ ER DIAGRAM
- ★ ENTITIES
- ★ TECH STACK
- ★ CODE
- ★ SNAPSHOT

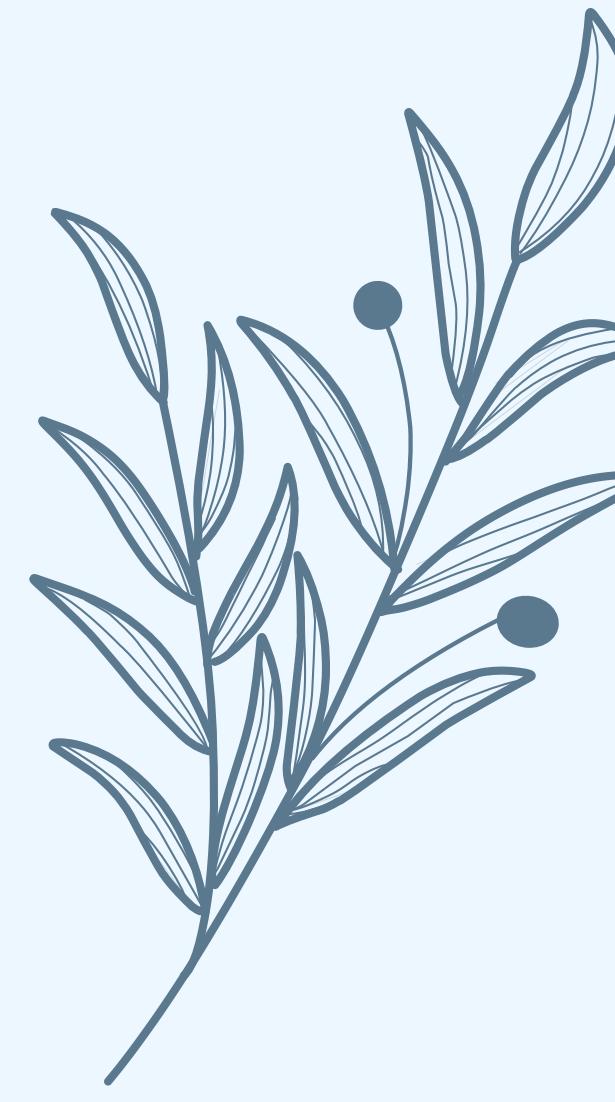
# OBJECTIVES



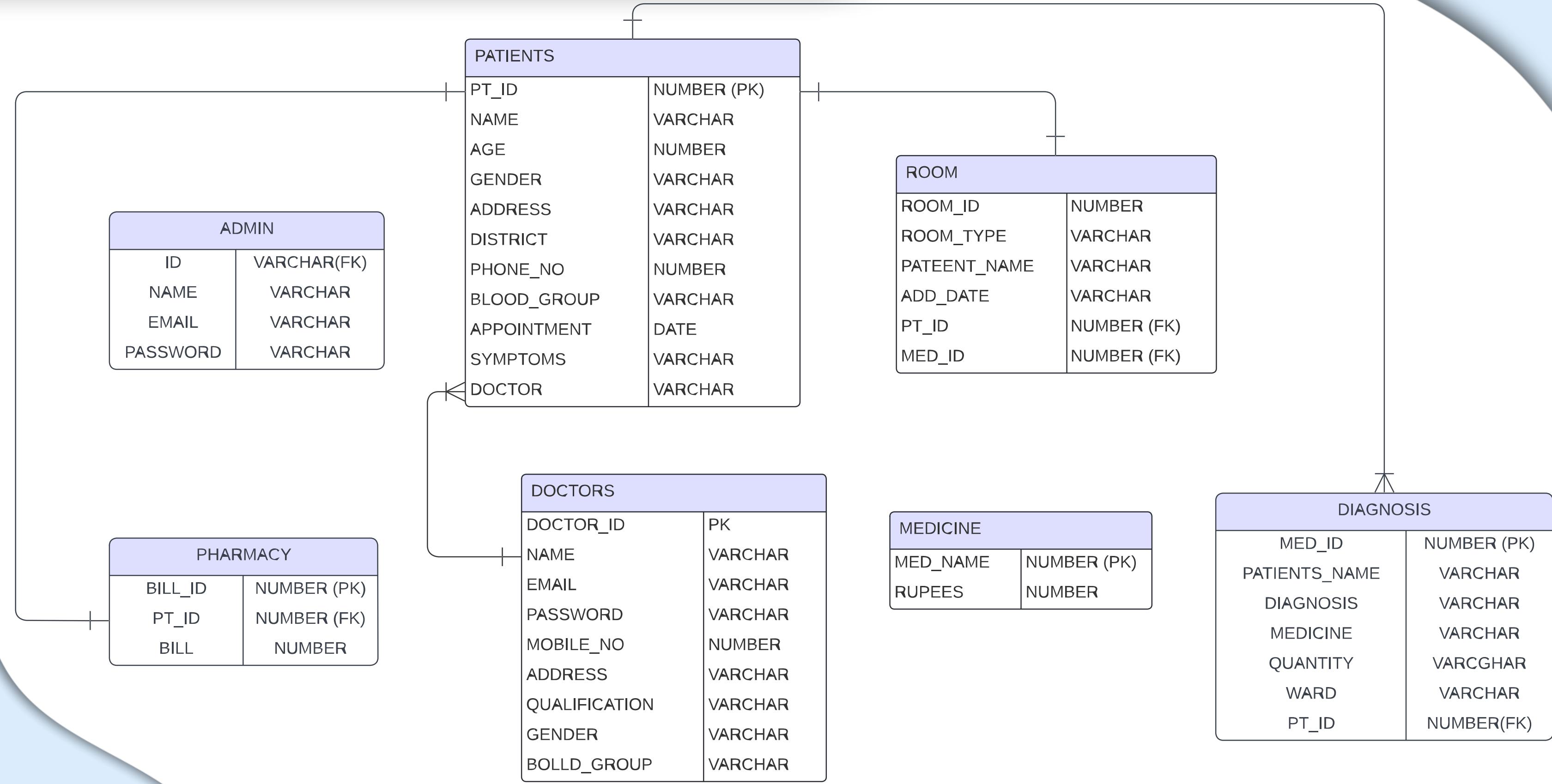
The objective of the Hospital Management System project is to streamline the recording and management of patient-doctor interactions, diagnoses, and treatments. By maintaining comprehensive databases, the system aims to efficiently store and retrieve patient histories, doctor details, and room allotments based on illness severity. Users are provided unique IDs for reference, ensuring secure access to relevant information. Through user-friendly interfaces, the project facilitates easy access to data for administrators, doctors, and patients alike, ultimately enhancing the overall efficiency and accessibility of hospital management processes.

# ABSTRACT

- The hospital management system maintains a collection of databases of various data like patient history, the details of the patients and doctors, regarding the status of available doctors, virtual pharmacy etc., and the primary aim of the project is to record all details of the doctor- patient diagnosis and treatment and store the data.
- Each user is given a unique id for the reference purpose, and there is even a provision for the allotment of rooms based on the intensity of severity of the illness, suffered by the patient.
- The details of the patient and the doctor can be accessed by the admin. Also, the doctor can access a patient's medical history.
- And, the patients can access the information of the available doctors.
- This project offers a method to make the process of the hospital management system easy and available at any time.
- It is user- friendly and overcomes the complex difficulty faced by the manual mode of data storage, and provides a simple method that can be understood by all.



# ER DIAGRAM



# ENTITIES

1

## ADMIN MODULE:

- Manage department of hospitals, patients, doctors, pharmacy, nurse etc.
- Watch transaction reports of the patient payment.
- Can view Doctor's details and Patients details.
- Administrator has the access to entire database of Hospital Management System.

2

## DOCTOR MODULE:

- Manage patient account opening and updating.
- Can view own profile.
- Records contains the personal information such as name, email, phone number, address, gender etc. Educational details lie qualification, specialization.
- Can view patient's details.
- Create prescription for patients

# ENTITIES

3

## PHARMACY MODULE:

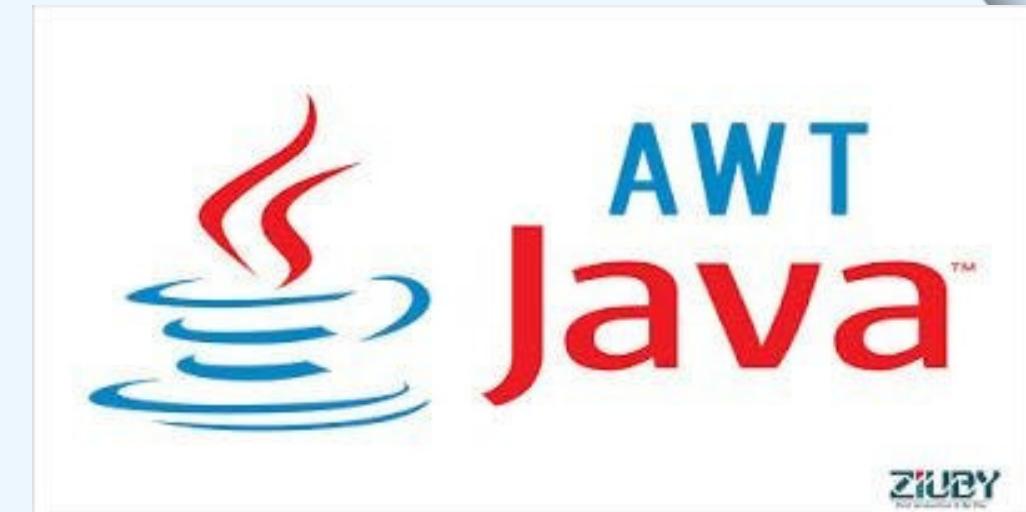
- Record contains the patients name, ID, patient diagnosis.
- Maintains medicines, keep record of the stocks.
- Watch prescription of patient and provide medication accordingly.
- Also manages the total hospitalization and medication bill of the patient.

4

## PATIENT MODULE:

- Patients can know about the doctors.
- Records include patient medical histories, prescribed medications, test results and appointment histories.
- Records contain PII (Personal Identifiable Information) such as name date of birth, address, etc.
- Patients can view medication history, operation history.
- Can view admit history like bed, ward etc.

# TECH STACK



# CODE - DOCTOR

## 1 DOCTOR LOGIN

```
lbl_heading=new JLabel("Login Form");
lbl_heading.setFont(new Font("Algerian", Font.BOLD, 25));
lbl_heading.setHorizontalAlignment(SwingConstants.CENTER);
lbl_user=new JLabel("Username");
lbl_user.setFont(new Font("Bodoni MT",Font.BOLD,20));
lbl_password=new JLabel("Password");
lbl_password.setFont(new Font("Bodoni MT",Font.BOLD,20));
txt_user=new JTextField();
jpf_password=new JPasswordField();
jpf_password.setEchoChar('*');
```

## 2 DOCTOR VIEW

```
lbl_heading=new JLabel("VIEW THE DETAILS");
lbl_heading.setFont(new Font("Algerian",Font.BOLD,30));
lbl_id=new JLabel("ID");
lbl_id.setFont(new Font("Bodoni MT",Font.BOLD,20));
lbl_name=new JLabel("Name");
lbl_name.setFont(new Font("Bodoni MT",Font.BOLD,20));
lbl_email=new JLabel("Email");
lbl_email.setFont(new Font("Bodoni MT",Font.BOLD,20));
lbl_password=new JLabel("Password");
lbl_password.setFont(new Font("Bodoni MT",Font.BOLD,20));
lbl_mobile=new JLabel("Mobile");
lbl_mobile.setFont(new Font("Bodoni MT",Font.BOLD,20));
```

# CODE - PATIENT

1

## PATIENT SQL CONNECTIVITY

```
try {
    Class.forName("oracle.jdbc.driver.OracleDriver");
    Connection con = DatabaseOperation.getConnection();
    Statement st = con.createStatement();
    String query = "select * from patient where id=" + id + "";
    ResultSet res = st.executeQuery(query);
    while (res.next()) {
        c_details[0][0] = (res.getString("Name"));
        c_details[0][1] = (res.getString("Age"));
        c_details[0][2] = (res.getString("Bloodgrp"));
        c_details[0][3] = (res.getDate("Appointment").toString());
        c_details[0][4] = (res.getString("Doctor"));
        t_sym.setText(res.getString("Symptoms"));
    }
    con.setAutoCommit(true);
} catch (HeadlessException | ClassNotFoundException |
    NumberFormatException | SQLException ex) {
    JOptionPane.showMessageDialog(null, ex.toString(), "Error",
        JOptionPane.ERROR_MESSAGE);
```

2

## PATIENT DIAGNOSIS

```
@Override
public void actionPerformed(ActionEvent e) {
    String id = t_pid.getText();
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection con =
            DatabaseOperation.getConnection();
        Statement st = con.createStatement();
        String query = "select * from patient where id=" +
            id + "";
        ResultSet res = st.executeQuery(query);
        if (!res.next()) {
            JOptionPane.showMessageDialog(null,
                "Patient details not registered");
        } else {
            new PDetails(id);
            setVisible(false);
        }
        con.setAutoCommit(true);
    } catch (HeadlessException |
        ClassNotFoundException | NumberFormatException |
        SQLException ex) {
        JOptionPane.showMessageDialog(null,
            ex.toString(), "Error", JOptionPane.ERROR_MESSAGE);
    }
```

# CODE-ADMIN

1

## ROOM DETAILS VIEW

```
try {
    Connection con = DatabaseOperation.getConnection();
    PreparedStatement pstm = con.prepareStatement("SELECT
        * FROM room");
    ResultSet Rs = pstm.executeQuery();
    while(Rs.next()){
        model.addRow(new Object[]{Rs.getInt("roomid"),
        Rs.getString("PatientName"),Rs.getString("Roomtype")});
    }
} catch (Exception e) {
    System.out.println(e.getMessage());
}
```

2

## PATIENTS DETAILS VIEW

```
try {
    Connection con=DatabaseOperation.getConnection();
    Statement st = con.createStatement();
    String query = "select * from patient where id='"+ id + "'";
    ResultSet res = st.executeQuery(query);
    if (!res.next()) {
        JOptionPane.showMessageDialog(null, "Patient Details not
        Registered.");
    } else {
        query = "select * from patient where id='"+ id + "'";
        res = st.executeQuery(query);
        while (res.next()) {
            id_txt.setText(id);
            name_txt.setText(res.getNString("name"));
            age_txt.setText(res.getNString("age"));
            phno_txt.setText(res.getNString("ph_no"));
            address_txt.setText(res.getNString("address"));
            dist_jcb.setText(res.getNString("district"));
            apt_date.setText(res.getDate("appointment").toString());
            bldgrp_jcb.setText(res.getNString("bloodgrp"));
            doctName_jcb.setText(res.getNString("doctor"));
            symp_txt.setText(res.getNString("symptoms"));
            gender.setText(res.getNStrin
```

# CODE - ADMIN

3

## DOCTOR DETAILS VIEW

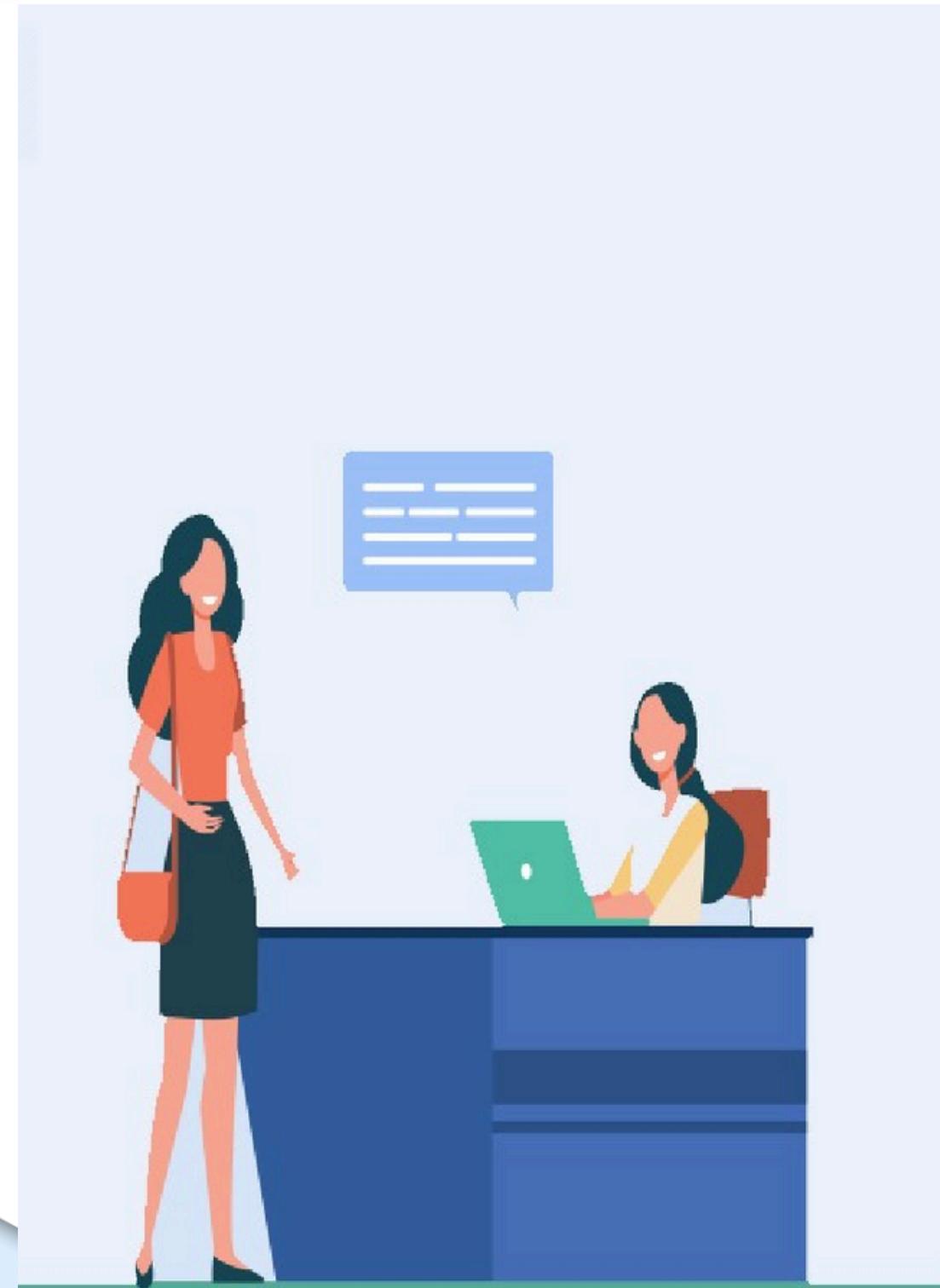
```
Connection con=DatabaseOperation.getConnection();
String query="select
id,name,email,password,mobile,address,qualification,gender,bloodgroup,dateofjoining
from doctor_details where id=?";
PreparedStatement pstmt=con.prepareStatement(query);
pstmt.setString(1,int_id);
ResultSet rst=pstmt.executeQuery();
if(rst.next()){
txt_id.setText(rst.getString("id"));
//txt_id.setText("int_id");
//txt_id.setText("id");
txt_name.setText(rst.getString("name"));
txt_email.setText(rst.getString("email"));
txt_password.setText(rst.getString("password"));
txt_mobile.setText(rst.getString("mobile"));
txt_address.setText(rst.getString("address"));
txt_qualification.setText(rst.getString("qualification"));
txt_gender.setText(rst.getString("gender"));
txt_blood_group.setText(rst.getString("bloodgroup"));}
```

4

## ADMIN LOGIN

```
try
{
if(!Pattern.matches("[A-Za-z0-9+_.-]+(.+)$",uname)){
JOptionPane.showMessageDialog(login,"Invalid
email");
}
else if(!Pattern.matches("(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.*[@!$#%&*?])(?=.*\\S+$.){8,20}$", pass)){
JOptionPane.showMessageDialog(login,"Set valid
password");
}
else{
Connection con=DatabaseOperation.getConnection();
Statement st=con.createStatement();
String query="insert into doctor_details
values('"+uname+"','"+pass+"')";
st.executeUpdate(query);
con.close();
JOptionPane.showMessageDialog(null,"Success!");
}
```

# SNAPSHOTS-LOGIN PAGE



A stylized illustration of two women in an office setting. One woman stands on the left, wearing an orange top and black skirt, carrying an orange bag. A blue speech bubble is positioned above her head. The other woman sits at a blue desk on the right, wearing a white top and yellow skirt, working on a green laptop. The background is plain white.

**LOGIN FORM**

*To continue, Sign in to your account*

**Username**

**Password**

Show Password

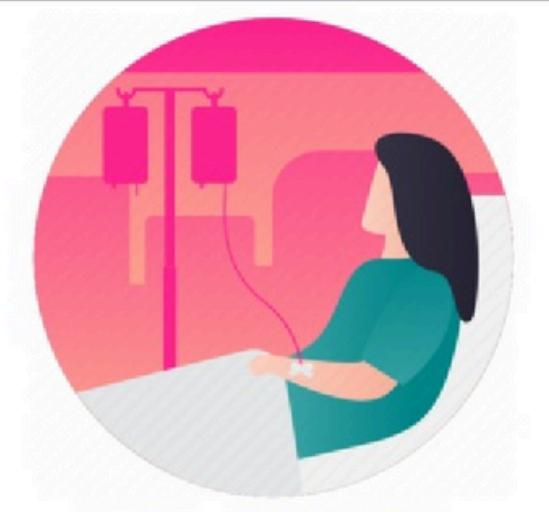
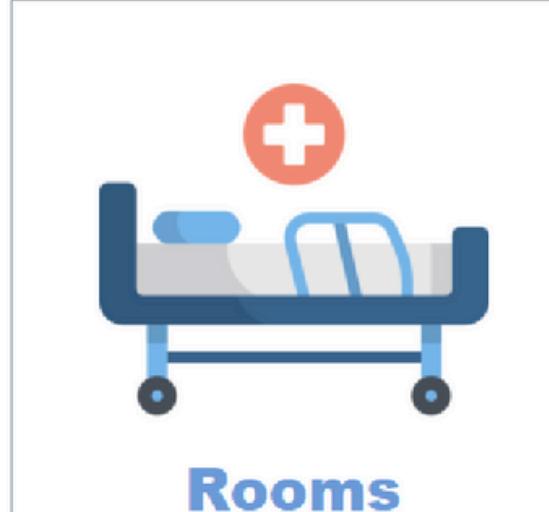


# SNAPSHOTS-ADMIN PAGE



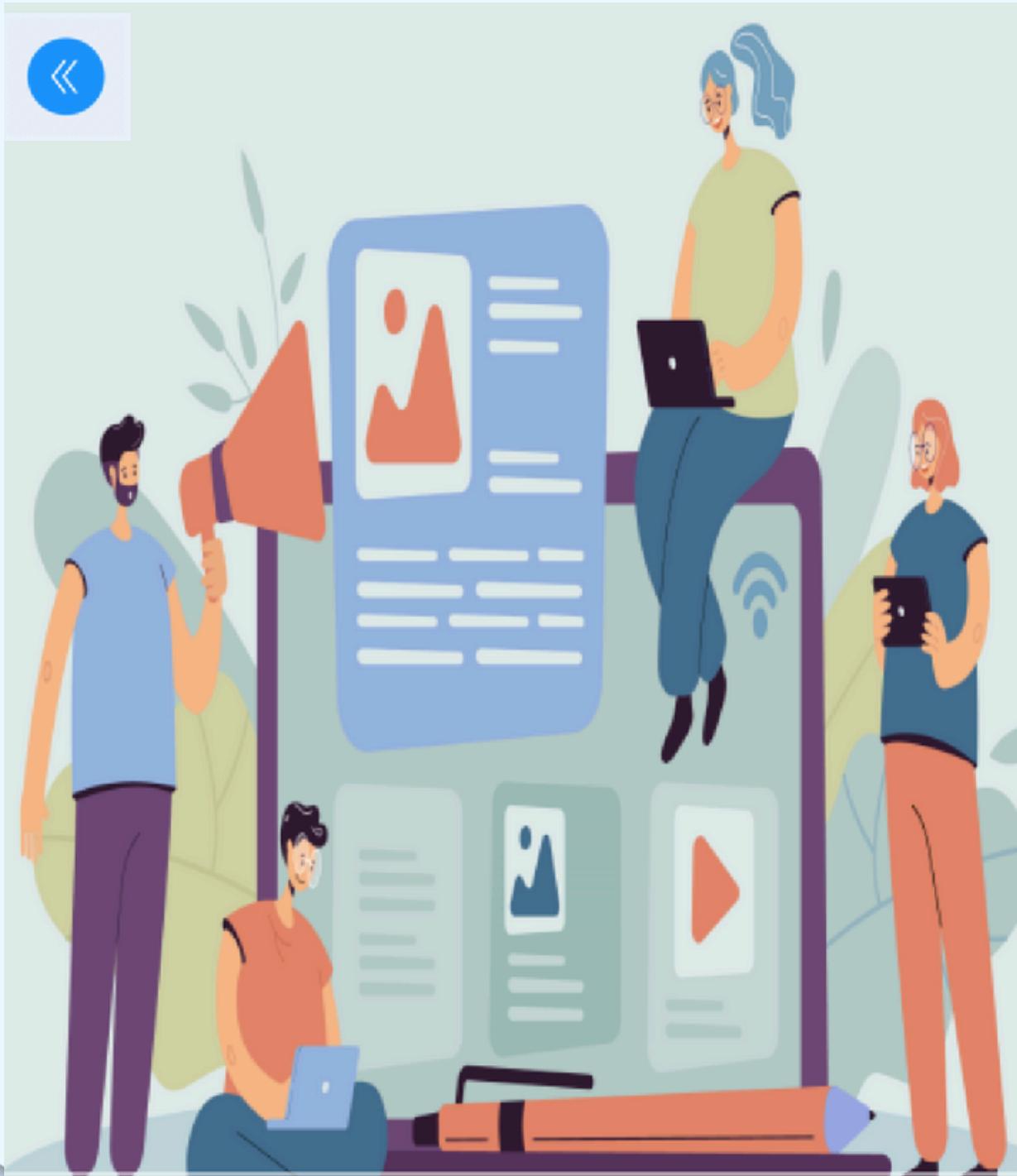
**splendid International Ho...**  
Coimbatore, Tamil Nadu  
+91 9123456780  
skcethospital@gmail....

**Admin Controls**

- **Doctor Registration**
- **Doctor Details**
- **Patient Details**
- **Rooms**



# SNAPSHOTS-DOCTOR DETAILS



**VIEW DETAILS**

*Inorder to view the details, please enter the id*

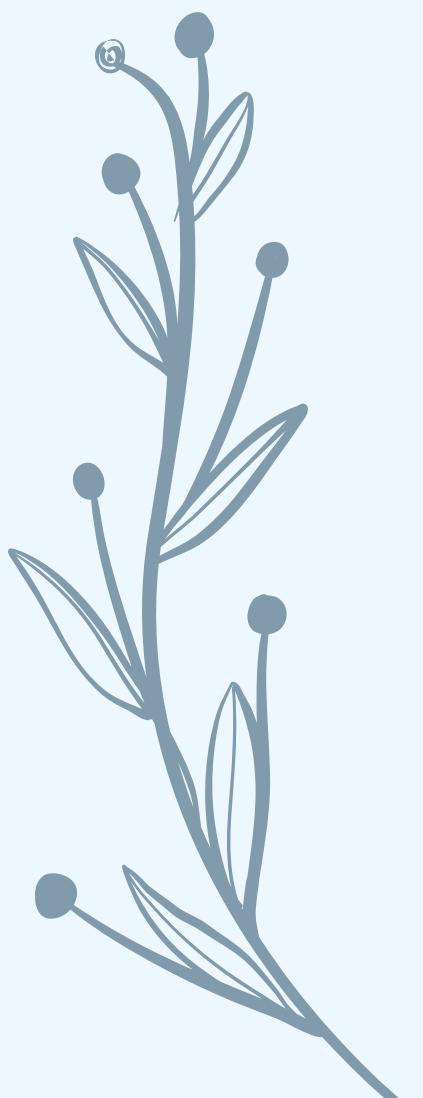
Doctor ID  ✓



# SNAPSHOTS-DETAILS VIEW

**VIEW THE DETAILS**

<b>ID</b>	<input type="text" value="4"/>
<b>Name</b>	<input type="text" value="Dr. David Brown"/>
<b>Email</b>	<input type="text" value="david.brown@example.com"/>
<b>Password</b>	<input type="text" value="password101"/>
<b>Address</b>	<input type="text" value="101 Birch St"/>
<b>Qualification</b>	<input type="text" value="DO, Family Medicine"/>
<b>Gender</b>	<input type="text" value="Male"/>
<b>Blood Group</b>	<input type="text" value="AB+"/>
<b>Date Of Joining</b>	<input type="text" value="2020-06-18"/>



# SNAPSHOTS-ROOM DETAILS

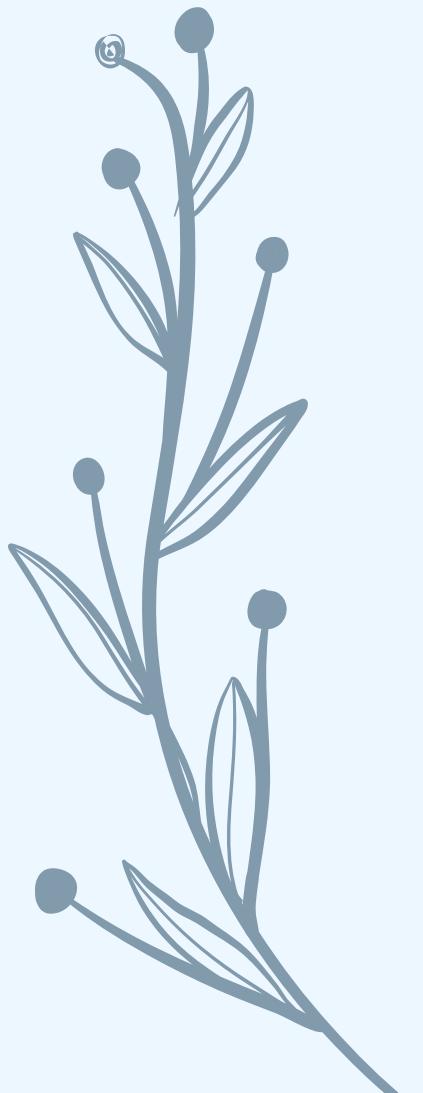
«

## ROOMS

Total Wards	100
Used Wards	5
Available Wards	95

Room ID	Patient Name	Room Type
1	John Doe	Single
2	Jane Smith	Double
3	Mike Johnson	Single
4	Emily Davis	ICU
5	Robert Brown	General





Thank  
You