**LOVELY PROFESSIONAL UNIVERSITY**

**PHAGWARA, PUNJAB**

Java Project on

Hospital Management System



GROUP – 9

**SUBMITTED BY:**

**Name:** Nardeep Singh Shekhawat **Registration Number**: 12016980

**Name:** Prashant Saini **Registration Number:** 12016739

**Name:** Gursujan Madem Reddy **Registration Number:** 12016943

**Section:** E2003

**School:** School of Electronics and Electrical Engineering

**Course Code & Name:** CSE310, Programming in JAVA

**SUBMITTED TO:**

Dr. Aarti

**TABLE OF CONTENTS**

[**LIST OF FIGURES** 3](#_Toc133314807)

[**1.** **Abstract** 4](#_Toc133314808)

[**2.** **Introduction** 5](#_Toc133314809)

[**2.1** **Reading and writing from files-** 6](#_Toc133314810)

[**2.2** **Including Java Swing-** 6](#_Toc133314811)

[**2.3** **Including exception handlers-** 7](#_Toc133314812)

[**3.** **Source Code** 7](#_Toc133314813)

[**3.1** **HospitalHomepage-** 7](#_Toc133314814)

[**3.2** **AdminRegistration** 8](#_Toc133314815)

[**3.3 AdminSignIn –** 10](#_Toc133314816)

[**3.4 DoctorRegistration –** 12](#_Toc133314817)

[**3.5 DoctorSign –** 14](#_Toc133314818)

[**3.6 DoctorPage –** 15](#_Toc133314819)

[**3.7 PatientRegistration –** 16](#_Toc133314820)

[**3.8 PatientLogin –** 18](#_Toc133314821)

[**3.9 PatientPage –** 19](#_Toc133314822)

[**3.10 PayBill –** 21](#_Toc133314823)

[**3.11 SeeAppointment –** 23](#_Toc133314824)

[**3.12 WriteMedicine –** 24](#_Toc133314825)

[**3.13 ChargeFee –** 25](#_Toc133314826)

[**3.14 DeleteDoctor –** 26](#_Toc133314827)

[**3.15 DeletePatient –** 28](#_Toc133314828)

[**3.16 BookAppointment –** 30](#_Toc133314829)

[**3.17 PatientMedicine –** 31](#_Toc133314830)

[**5** **Output** 33](#_Toc133314831)

[**6** **Conclusion** 40](#_Toc133314832)

### **LIST OF FIGURES**

[**Figure 2- Hospital Homepage** 33](#_Toc133314305)

[**Figure 3- Admin Registration** 34](#_Toc133314306)

[**Figure 4-Admin SignIn** 34](#_Toc133314307)

[**Figure 5-Doctor Registration** 35](#_Toc133314308)

[**Figure 6- Doctor SignIn** 35](#_Toc133314309)

[**Figure 7- Doctor Page** 35](#_Toc133314310)

[**Figure 8- See Appointments** 36](#_Toc133314311)

[**Figure 9-Write Medicine** 36](#_Toc133314312)

[**Figure 10-Patient Registration** 37](#_Toc133314313)

[**Figure 11-Patient SignIn** 37](#_Toc133314314)

[**Figure 12-Patient Page** 38](#_Toc133314315)

[**Figure 11-Book Appointment** 38](#_Toc133314316)

[**Figure 11-Patient Medicine** 39](#_Toc133314317)

[**Figure 11-Pay Fee** 39](#_Toc133314318)



### **Abstract**

Continuum robots with redundant degrees of freedom and postactuated devices are suitable for application in

aerospace, nuclear facilities, and other narrow and multiobstacle special environments. The development of a

snake-inspired continuum robot is presented in this study. The morphological skeleton structure of the snake

body is simulated using underactuated continuum joints, which include several rigid-body joints in series. Each

rigid-body joint is driven by the traction of a wire rope. Based on the layered-drive principle, angular synchro-

nous motion can be realized in space with multiple rigid-body joints in a single continuous joint, which can

considerably reduce the complexity of the inverse kinematics solution, terminal drive box, and control system.

The static and dynamic characteristics of the snake-inspired robot are obtained through torque balance and an

equivalent transformation. Finally, we demonstrate trajectory planning and load capacity testing in two robot

prototypes with arm lengths of 1500 and 2300 mm (including two and four continuous joints, respectively). The

rationality of the structure and the correctness of the control of the layered-drive snake-inspired robot are veriﬁed

Continuum robots with redundant degrees of freedom and postactuated devices are suitable for application in

aerospace, nuclear facilities, and other narrow and multiobstacle special environments. The development of a

snake-inspired continuum robot is presented in this study. The morphological skeleton structure of the snake

body is simulated using underactuated continuum joints, which include several rigid-body joints in series. Each

rigid-body joint is driven by the traction of a wire rope. Based on the layered-drive principle, angular synchro-

nous motion can be realized in space with multiple rigid-body joints in a single continuous joint, which can

considerably reduce the complexity of the inverse kinematics solution, terminal drive box, and control system.

The static and dynamic characteristics of the snake-inspired robot are obtained through torque balance and an

equivalent transformation. Finally, we demonstrate trajectory planning and load capacity testing in two robot

prototypes with arm lengths of 1500 and 2300 mm (including two and four continuous joints, respectively). The

rationality of the structure and the correctness of the control of the layered-drive snake-inspired robot are veriﬁed

Continuum robots with redundant degrees of freedom and postactuated devices are suitable for application in

aerospace, nuclear facilities, and other narrow and multiobstacle special environments. The development of a

snake-inspired continuum robot is presented in this study. The morphological skeleton structure of the snake

body is simulated using underactuated continuum joints, which include several rigid-body joints in series. Each

rigid-body joint is driven by the traction of a wire rope. Based on the layered-drive principle, angular synchro-

nous motion can be realized in space with multiple rigid-body joints in a single continuous joint, which can

considerably reduce the complexity of the inverse kinematics solution, terminal drive box, and control system.

The static and dynamic characteristics of the snake-inspired robot are obtained through torque balance and an

equivalent transformation. Finally, we demonstrate trajectory planning and load capacity testing in two robot

prototypes with arm lengths of 1500 and 2300 mm (including two and four continuous joints, respectively). The

rationality of the structure and the correctness of the control of the layered-drive snake-inspired robot are veriﬁed

Hospital Management System is a project based on Java Swing and File Management. The main objective of this project is to create a user-friendly software that can manage the records of patients, doctors, and other hospital staff. This project has been designed to help hospitals to manage their day-to-day activities efficiently.

The system is designed in such a way that it can store and manage the records of all the patients, doctors, and admin in the hospital. The system can keep track of patient appointments, their medical history, and billing information. It also allows doctors to update and access their patients' records from anywhere in the hospital.

The Hospital Management System is equipped with various features such as appointment scheduling, patient management, billing and invoicing, and report generation. It also has a user-friendly interface that makes it easy for hospital staff to navigate through the system.

The File Management feature of this project allows the hospital to store all its important documents and records in a centralized location. This ensures that the hospital's data is secure and easy to access.

In conclusion, the Hospital Management System project is a powerful software that can help hospitals manage their activities effectively. With its advanced features and user-friendly interface, this project is a valuable asset to any hospital.

### **Introduction**

The Hospital Management System is a Java Swing and File Management-based project designed to help hospitals manage their day-to-day activities efficiently. The system provides an easy-to-use interface that can store and manage the records of patients, doctors, and other hospital staff members.

The Hospital Management System project aims to streamline hospital operations by automating many of the manual processes involved in patient care, such as appointment scheduling, patient management, billing and invoicing, and report generation.

The project is built using the Java Swing user interface framework, which provides a rich set of components for creating desktop applications. The File Management feature of the project allows the hospital to store all its important documents and records in a centralized location, ensuring that data is secure and easy to access.

Overall, the Hospital Management System project is a valuable asset for any hospital looking to improve its efficiency and patient care. With its advanced features and user-friendly interface, this project can help hospitals manage their activities effectively, resulting in better patient outcomes and improved operational efficiency.

### **Reading and writing from files-**

import java.io.BufferedReader;

import java.io.BufferedWriter;

These classes provide functionality for input and output operations, such as reading and writing data from and to files or streams.

Specifically, the first import statement imports the BufferedReader class, which provides a way to read text from a character-input stream, such as a file or network connection, in an efficient manner.

The second import statement imports the BufferedWriter class, which provides a way to write text to a character-output stream, such as a file or network connection, in an efficient manner.

### **Including Java Swing-**

import javax.swing.JLabel;

import javax.swing.JOptionPane;

The above lines of code are Java import statements that are used to include classes from the javax.swing package in the current program. This package contains classes that provide graphical user interface (GUI) components for creating desktop applications.

Specifically, the first import statement imports the JLabel class, which provides a display area for a short text string or an image, or both. JLabel is a common component used to display static text or images in a GUI.

Second import statement imports the JOptionPane class, which provides a standardized dialog box for displaying messages, requesting input, or providing feedback to the user. JOptionPane provides an easy way to display different types of messages to the user, such as information, warning, error, or confirmation messages, and prompt the user for input.

### **Including exception handlers-**

import java.util.InputMismatchException;

import java.io.FileNotFoundException;

The first import statement imports the InputMismatchException class from the java.util package. This class is an exception that is thrown by the Scanner class when an input token does not match the expected pattern or data type. It is commonly used to handle input errors when reading data from the console or a file.

The second import statement imports the FileNotFoundException class from the java.io package. This class is an exception that is thrown when a file with the specified pathname does not exist or cannot be opened for reading or writing. It is commonly used to handle file input/output errors when reading or writing data from or to a file.

By including these classes in the program, the developer can catch these exceptions using try-catch blocks and handle them accordingly to avoid program crashes or unexpected behaviour.



### **Source Code**

### **HospitalHomepage-**

1. package saini;  
     
   import javax.swing.\*;  
   import java.awt.\*;  
   import java.awt.event.\*;  
     
   public class HospitalHomepage extends JFrame implements ActionListener {  
     
    JLabel nameLabel, addressLabel, phoneLabel;  
    JButton aboutBtn, doctorRegBtn, patientRegBtn, adminRegBtn, doctorSignBtn, patientSignBtn, adminSignBtn;  
     
    public HospitalHomepage() {  
     
    nameLabel = new JLabel("JAN SEVA KENDRA");  
    addressLabel = new JLabel("NAI DILLI");  
    phoneLabel = new JLabel("1234567890");  
     
    aboutBtn = new JButton("About Page");  
    doctorRegBtn = new JButton("Doctor Registration");  
    patientRegBtn = new JButton("Patient Registration");  
    adminRegBtn = new JButton("Admin Registration");  
    doctorSignBtn = new JButton("Doctor Sign-in");  
    patientSignBtn = new JButton("Patient Sign-in");  
    adminSignBtn = new JButton("Admin Sign-in");  
     
    aboutBtn.addActionListener(this);  
    doctorRegBtn.addActionListener(this);  
    patientRegBtn.addActionListener(this);  
    adminRegBtn.addActionListener(this);  
    doctorSignBtn.addActionListener(this);  
    patientSignBtn.addActionListener(this);  
    adminSignBtn.addActionListener(this);  
     
    JPanel panel = new JPanel();  
    panel.setLayout(new GridLayout(4, 1));  
    panel.add(nameLabel);  
    panel.add(addressLabel);  
    panel.add(phoneLabel);  
    panel.add(aboutBtn);  
    panel.add(doctorRegBtn);  
    panel.add(patientRegBtn);  
    panel.add(adminRegBtn);  
    panel.add(doctorSignBtn);  
    panel.add(patientSignBtn);  
    panel.add(adminSignBtn);  
     
    getContentPane().add(panel, BorderLayout.*CENTER*);  
    setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
    setTitle("Hospital Page");  
    setSize(400, 400);  
    setVisible(true);  
    }  
     
    public void actionPerformed(ActionEvent e) {  
    if (e.getSource() == aboutBtn) {  
    JOptionPane.*showMessageDialog*(this, "i am hospital");  
    } else if (e.getSource() == doctorRegBtn) {  
    new DoctorRegistration();  
    } else if (e.getSource() == patientRegBtn) {  
    new PatientRegistration();  
    } else if (e.getSource() == adminRegBtn) {  
    new AdminRegistration();  
    } else if (e.getSource() == doctorSignBtn) {  
    new DoctorSignIn();  
    } else if (e.getSource() == patientSignBtn) {  
    new PatientSignIn();  
    } else if (e.getSource() == adminSignBtn) {  
    new AdminSignIn();  
    }  
    }  
     
    public static void main(String[] args) {  
    new HospitalHomepage();  
    }  
   }

### **AdminRegistration**

1. package saini;  
     
   import javax.swing.\*;  
   import java.awt.\*;  
   import java.awt.event.\*;  
   import java.io.\*;  
     
   public class AdminRegistration extends JFrame implements ActionListener {  
     
    private JLabel nameLabel, ageLabel, bloodGroupLabel, usernameLabel, passwordLabel;  
    private JTextField nameField, ageField, usernameField;  
    private JPasswordField passwordField;  
    private JComboBox<String> bloodGroupBox;  
    private JButton submitButton, backButton;  
     
    public AdminRegistration() {  
    super("Admin Registration");  
    setSize(400, 300);  
    setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
    setLayout(new GridLayout(6, 2));  
     
    nameLabel = new JLabel("Name:");  
    add(nameLabel);  
    nameField = new JTextField(20);  
    add(nameField);  
     
    ageLabel = new JLabel("Age:");  
    add(ageLabel);  
    ageField = new JTextField(20);  
    add(ageField);  
     
    bloodGroupLabel = new JLabel("Blood Group:");  
    add(bloodGroupLabel);  
    String[] bloodGroups = {"A+", "A-", "B+", "B-", "O+", "O-", "AB+", "AB-"};  
    bloodGroupBox = new JComboBox<>(bloodGroups);  
    add(bloodGroupBox);  
     
    usernameLabel = new JLabel("Username:");  
    add(usernameLabel);  
    usernameField = new JTextField(20);  
    add(usernameField);  
     
    passwordLabel = new JLabel("Password:");  
    add(passwordLabel);  
    passwordField = new JPasswordField(20);  
    add(passwordField);  
     
    submitButton = new JButton("Submit");  
    submitButton.addActionListener(this);  
    add(submitButton);  
     
    backButton = new JButton("Back");  
    backButton.addActionListener(this);  
    add(backButton);  
     
    setVisible(true);  
    }  
     
    public void actionPerformed(ActionEvent e) {  
    if (e.getSource() == submitButton) {  
    String name = nameField.getText();  
    String age = ageField.getText();  
    String bloodGroup = (String) bloodGroupBox.getSelectedItem();  
    String username = usernameField.getText();  
    String password = new String(passwordField.getPassword());  
     
    try {  
    BufferedWriter writer = new BufferedWriter(new FileWriter("Admin.txt", true));  
    writer.write(name + "," + age + "," + bloodGroup + "," + username + "," + password);  
    writer.newLine();  
    writer.close();  
    JOptionPane.*showMessageDialog*(this, "Admin registered successfully!");  
    } catch (IOException ex) {  
    ex.printStackTrace();  
    }  
    } else if (e.getSource() == backButton) {  
    dispose();  
    new HospitalHomepage();  
    }  
    }  
   }

### **3.3 AdminSignIn –**

package saini;  
  
import java.awt.\*;  
import java.awt.event.\*;  
import javax.swing.\*;  
  
import java.io.\*;  
  
public class AdminSignIn implements ActionListener {  
 private JFrame frame;  
 private JLabel userLabel, passwordLabel, messageLabel;  
 private JTextField userField;  
 private JPasswordField passwordField;  
 private JButton loginButton, backButton;  
 private JComboBox<String> optionsBox;  
  
 public AdminSignIn() {  
 // create frame and components  
 frame = new JFrame("Admin Sign In");  
 userLabel = new JLabel("Username:");  
 passwordLabel = new JLabel("Password:");  
 userField = new JTextField();  
 passwordField = new JPasswordField();  
 messageLabel = new JLabel();  
 loginButton = new JButton("Login");  
 backButton = new JButton("Back");  
 optionsBox = new JComboBox<>(new String[]{"Delete Doctor", "Delete Patient"});  
  
 // set layout and add components  
 JPanel panel = new JPanel(new GridLayout(4, 2));  
 panel.add(userLabel);  
 panel.add(userField);  
 panel.add(passwordLabel);  
 panel.add(passwordField);  
 panel.add(messageLabel);  
 panel.add(new JLabel());  
 panel.add(loginButton);  
 panel.add(backButton);  
 frame.add(panel, BorderLayout.*CENTER*);  
 frame.add(optionsBox, BorderLayout.*PAGE\_END*);  
  
 // add action listeners  
 loginButton.addActionListener(this);  
 backButton.addActionListener(this);  
  
 // set frame properties  
 frame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 frame.pack();  
 frame.setLocationRelativeTo(null);  
 frame.setVisible(true);  
 }  
  
 public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == loginButton) {  
 String username = userField.getText().trim();  
 String password = new String(passwordField.getPassword());  
  
 // check credentials  
 boolean success = false;  
 try {  
 BufferedReader reader = new BufferedReader(new FileReader("Admin.txt"));  
 String line;  
 while ((line = reader.readLine()) != null) {  
 String[] fields = line.split(",");  
 String storedUsername = fields[3].trim();  
 String storedPassword = fields[4].trim();  
 if (storedUsername.equals(username) && storedPassword.equals(password)) {  
 success = true;  
 break;  
 }  
 }  
 reader.close();  
 } catch (IOException ex) {  
 ex.printStackTrace();  
 }  
  
 // handle login success  
 if (success) {  
 messageLabel.setText("Login successful!");  
 int option = optionsBox.getSelectedIndex();  
 if (option == 0) {  
 frame.dispose();  
 new DeleteDoctor();  
 } else if (option == 1) {  
 frame.dispose();  
 new DeletePatient();  
 }  
 } else {  
 messageLabel.setText("Invalid username or password.");  
 }  
 } else if (e.getSource() == backButton) {  
 frame.dispose();  
 new HospitalHomepage();  
 }  
 }  
  
  
}

## **3.4 DoctorRegistration –**

package saini;  
  
import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.\*;  
import java.io.\*;  
  
public class DoctorRegistration extends JFrame implements ActionListener {  
 private JLabel nameLabel, ageLabel, qualificationLabel, bloodGroupLabel, usernameLabel, passwordLabel;  
 private JTextField nameTextField, ageTextField, qualificationTextField, usernameTextField;  
 private JComboBox<String> bloodGroupComboBox;  
 private JPasswordField passwordField;  
 private JButton registerButton, backButton;  
  
 public DoctorRegistration() {  
 // Setting the frame properties  
 setTitle("Doctor Registration");  
 setSize(400, 400);  
 setLocationRelativeTo(null);  
 setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 setLayout(new GridLayout(7, 2, 10, 10));  
  
 // Creating labels and fields for input  
 nameLabel = new JLabel("Name:");  
 add(nameLabel);  
 nameTextField = new JTextField();  
 add(nameTextField);  
  
 ageLabel = new JLabel("Age:");  
 add(ageLabel);  
 ageTextField = new JTextField();  
 add(ageTextField);  
  
 qualificationLabel = new JLabel("Qualification:");  
 add(qualificationLabel);  
 qualificationTextField = new JTextField();  
 add(qualificationTextField);  
  
 bloodGroupLabel = new JLabel("Blood Group:");  
 add(bloodGroupLabel);  
 String[] bloodGroups = {"A+", "A-", "B+", "B-", "O+", "O-", "AB+", "AB-"};  
 bloodGroupComboBox = new JComboBox<>(bloodGroups);  
 add(bloodGroupComboBox);  
  
 usernameLabel = new JLabel("Username:");  
 add(usernameLabel);  
 usernameTextField = new JTextField();  
 add(usernameTextField);  
  
 passwordLabel = new JLabel("Password:");  
 add(passwordLabel);  
 passwordField = new JPasswordField();  
 add(passwordField);  
  
 // Creating register and back buttons  
 registerButton = new JButton("Register");  
 registerButton.addActionListener(this);  
 add(registerButton);  
  
 backButton = new JButton("Back");  
 backButton.addActionListener(this);  
 add(backButton);  
  
 // Setting the frame visible  
 setVisible(true);  
 }  
  
 // Action performed when a button is clicked  
 public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == registerButton) {  
 String name = nameTextField.getText();  
 int age = Integer.*parseInt*(ageTextField.getText());  
 String qualification = qualificationTextField.getText();  
 String bloodGroup = (String) bloodGroupComboBox.getSelectedItem();  
 String username = usernameTextField.getText();  
 String password = new String(passwordField.getPassword());  
  
 // Writing doctor information to file  
 try (PrintWriter writer = new PrintWriter(new FileWriter("Doctor.txt", true))) {  
 writer.println(name + "," + age + "," + qualification + "," + bloodGroup + "," + username + "," + password);  
 JOptionPane.*showMessageDialog*(null, "Registration successful!");  
 } catch (IOException ex) {  
 JOptionPane.*showMessageDialog*(null, "Error writing to file: " + ex.getMessage());  
 }  
  
 // Clearing input fields  
 nameTextField.setText("");  
 ageTextField.setText("");  
 qualificationTextField.setText("");  
 bloodGroupComboBox.setSelectedIndex(0);  
 usernameTextField.setText("");  
 passwordField.setText("");  
 } else if (e.getSource() == backButton) {  
 new HospitalHomepage();  
 dispose();  
 }  
 }  
  
  
}

## **3.5 DoctorSign –**

package saini;  
  
import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.\*;  
import java.io.\*;  
  
public class DoctorSignIn extends JFrame implements ActionListener {  
  
 private JLabel usernameLabel, passwordLabel;  
 private JTextField usernameField;  
 private JPasswordField passwordField;  
 private JButton signInButton, backButton;  
  
 public DoctorSignIn() {  
 setTitle("Doctor Sign In");  
 setSize(400, 200);  
 setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 setLayout(new BorderLayout());  
  
 JPanel panel = new JPanel(new GridLayout(3, 2));  
 usernameLabel = new JLabel("Username:");  
 panel.add(usernameLabel);  
 usernameField = new JTextField();  
 panel.add(usernameField);  
 passwordLabel = new JLabel("Password:");  
 panel.add(passwordLabel);  
 passwordField = new JPasswordField();  
 panel.add(passwordField);  
 signInButton = new JButton("Sign In");  
 signInButton.addActionListener(this);  
 panel.add(signInButton);  
 backButton = new JButton("Back");  
 backButton.addActionListener(this);  
 panel.add(backButton);  
 add(panel, BorderLayout.*CENTER*);  
 setVisible(true);  
 }  
  
 public void actionPerformed(ActionEvent ae) {  
 String action = ae.getActionCommand();  
 if (action.equals("Sign In")) {  
 String username = usernameField.getText();  
 char[] passwordChars = passwordField.getPassword();  
 String password = new String(passwordChars);  
 boolean validLogin = false;  
  
 try {  
 BufferedReader br = new BufferedReader(new FileReader("Doctor.txt"));  
 String line;  
 while ((line = br.readLine()) != null) {  
 String[] fields = line.split(",");  
 String uniqueUsername = fields[4];  
 String uniquePassword = fields[5];  
 if (uniqueUsername.equals(username) && uniquePassword.equals(password)) {  
 validLogin = true;  
 break;  
 }  
 }  
 br.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
 if (validLogin) {  
 dispose();  
 String text = usernameField.getText();  
 new DoctorPage(text);  
 } else {  
 JOptionPane.*showMessageDialog*(this, "Invalid login. Please try again.");  
 }  
 } else if (action.equals("Back")) {  
 dispose();  
 new HospitalHomepage();  
 }  
 }  
  
 public static void main(String[] args) {  
 new DoctorSignIn();  
 }  
}

## **3.6 DoctorPage –**

package saini;  
  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import javax.swing.JButton;  
import javax.swing.JFrame;  
import javax.swing.JLabel;  
import javax.swing.JPanel;  
  
public class DoctorPage extends JFrame implements ActionListener {  
  
 private String username;  
  
 public DoctorPage(String username) {  
 this.username = username;  
 setTitle("Doctor Page");  
 setSize(400, 300);  
 setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
  
 JPanel panel = new JPanel();  
 panel.setLayout(null);  
  
 JLabel welcomeLabel = new JLabel("Welcome, Dr. " + username);  
 welcomeLabel.setBounds(20, 20, 200, 30);  
 panel.add(welcomeLabel);  
  
 JButton appointmentButton = new JButton("See Appointments");  
 appointmentButton.setBounds(20, 70, 150, 30);  
 appointmentButton.addActionListener(this);  
 panel.add(appointmentButton);  
  
  
  
 JButton medicineButton = new JButton("Write Medicine");  
 medicineButton.setBounds(20, 170, 150, 30);  
 medicineButton.addActionListener(this);  
 panel.add(medicineButton);  
  
 JButton backButton = new JButton("Back");  
 backButton.setBounds(250, 200, 100, 30);  
 backButton.addActionListener(this);  
 panel.add(backButton);  
  
 add(panel);  
 setVisible(true);  
 }  
  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 if (e.getActionCommand().equals("See Appointments")) {  
  
 new SeeAppointment(username);  
 }  
 else if (e.getActionCommand().equals("Write Medicine")) {  
 new WriteMedicine(username);  
 } else if (e.getActionCommand().equals("Back")) {  
 dispose();  
 new HospitalHomepage();  
 }  
 }  
}

## **3.7 PatientRegistration –**

package saini;  
  
import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.\*;  
import java.io.\*;  
  
public class PatientRegistration extends JFrame implements ActionListener {  
  
 private JLabel nameLabel, ageLabel, bgLabel, userLabel, passLabel;  
 private JTextField nameField, ageField, userField;  
 private JPasswordField passField;  
 private JComboBox<String> bgComboBox;  
 private JButton submitButton, backButton;  
  
 public PatientRegistration() {  
 super("Patient Registration");  
  
 // create labels and fields for input  
 nameLabel = new JLabel("Name:");  
 ageLabel = new JLabel("Age:");  
 bgLabel = new JLabel("Blood Group:");  
 userLabel = new JLabel("Username:");  
 passLabel = new JLabel("Password:");  
  
 nameField = new JTextField();  
 ageField = new JTextField();  
 userField = new JTextField();  
 passField = new JPasswordField();  
  
 String[] bgOptions = {"A+", "A-", "B+", "B-", "O+", "O-", "AB+", "AB-"};  
 bgComboBox = new JComboBox<>(bgOptions);  
  
 // create submit and back buttons  
 submitButton = new JButton("Submit");  
 backButton = new JButton("Back");  
  
 // add action listeners to buttons  
 submitButton.addActionListener(this);  
 backButton.addActionListener(this);  
  
 // set layout and add components to panel  
 JPanel panel = new JPanel(new GridLayout(6, 2));  
 panel.add(nameLabel);  
 panel.add(nameField);  
 panel.add(ageLabel);  
 panel.add(ageField);  
 panel.add(bgLabel);  
 panel.add(bgComboBox);  
 panel.add(userLabel);  
 panel.add(userField);  
 panel.add(passLabel);  
 panel.add(passField);  
 panel.add(submitButton);  
 panel.add(backButton);  
  
 // set panel and window properties  
 add(panel);  
 setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 setSize(300, 250);  
 setLocationRelativeTo(null);  
 setVisible(true);  
 }  
  
 public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == submitButton) {  
 String name = nameField.getText();  
 String age = ageField.getText();  
 String bg = (String) bgComboBox.getSelectedItem();  
 String username = userField.getText();  
 String password = String.*valueOf*(passField.getPassword());  
  
 try {  
 FileWriter fw = new FileWriter("Patient.txt", true);  
 BufferedWriter bw = new BufferedWriter(fw);  
 PrintWriter pw = new PrintWriter(bw);  
  
 pw.println(name + "," + age + "," + bg + "," + username + "," + password);  
  
 pw.close();  
 JOptionPane.*showMessageDialog*(this, "Patient registered successfully!");  
 } catch (IOException ex) {  
 JOptionPane.*showMessageDialog*(this, "Error: " + ex.getMessage());  
 }  
  
 } else if (e.getSource() == backButton) {  
 new HospitalHomepage();  
 dispose();  
 }  
 }  
}

## **3.8 PatientLogin –**

package saini;  
  
import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.\*;  
import java.io.\*;  
  
public class PatientSignIn {  
 private JFrame frame;  
 private JLabel labelUsername, labelPassword;  
 private JTextField txtUsername;  
 private JPasswordField txtPassword;  
 private JButton btnLogin, btnBack;  
  
 public PatientSignIn() {  
 frame = new JFrame("Patient Sign In");  
 frame.setSize(350, 200);  
 frame.setLayout(null);  
 frame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
  
 labelUsername = new JLabel("Username:");  
 labelUsername.setBounds(30, 20, 80, 25);  
 frame.add(labelUsername);  
  
 txtUsername = new JTextField();  
 txtUsername.setBounds(120, 20, 200, 25);  
 frame.add(txtUsername);  
  
 labelPassword = new JLabel("Password:");  
 labelPassword.setBounds(30, 50, 80, 25);  
 frame.add(labelPassword);  
  
 txtPassword = new JPasswordField();  
 txtPassword.setBounds(120, 50, 200, 25);  
 frame.add(txtPassword);  
  
 btnLogin = new JButton("Login");  
 btnLogin.setBounds(100, 90, 80, 25);  
 btnLogin.addActionListener(new ActionListener() {  
 public void actionPerformed(ActionEvent e) {  
 String username = txtUsername.getText();  
 String password = new String(txtPassword.getPassword());  
  
 if (login(username, password)) {  
 frame.setVisible(false);  
 new PatientPage(username).setVisible(true);  
 } else {  
 JOptionPane.*showMessageDialog*(frame, "Invalid username or password.");  
 }  
 }  
 });  
 frame.add(btnLogin);  
  
 btnBack = new JButton("Back");  
 btnBack.setBounds(190, 90, 80, 25);  
 btnBack.addActionListener(new ActionListener() {  
 public void actionPerformed(ActionEvent e) {  
 frame.setVisible(false);  
 new HospitalHomepage().setVisible(true);  
 }  
 });  
 frame.add(btnBack);  
  
 frame.setVisible(true);  
 }  
  
 private boolean login(String username, String password) {  
 try {  
 BufferedReader reader = new BufferedReader(new FileReader("Patient.txt"));  
 String line = reader.readLine();  
 while (line != null) {  
 String[] data = line.split(",");  
 if (data[3].equals(username) && data[4].equals(password)) {  
 reader.close();  
 return true;  
 }  
 line = reader.readLine();  
 }  
 reader.close();  
 return false;  
 } catch (IOException e) {  
 e.printStackTrace();  
 return false;  
 }  
 }  
}

## **3.9 PatientPage –**

package saini;  
  
import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
  
public class PatientPage extends JFrame implements ActionListener {  
  
 private JLabel titleLabel, optionsLabel;  
 private JButton bookAppointmentButton, payBillButton, patientMedicineButton, backButton;  
 private String username;  
  
 public PatientPage(String username) {  
 this.username = username;  
  
 setTitle("Patient Page");  
 setSize(500, 500);  
 setLayout(null);  
  
 titleLabel = new JLabel("Welcome " + username + "!");  
 titleLabel.setFont(new Font("Arial", Font.*BOLD*, 20));  
 titleLabel.setBounds(150, 50, 300, 50);  
 add(titleLabel);  
  
 optionsLabel = new JLabel("Please select an option:");  
 optionsLabel.setFont(new Font("Arial", Font.*PLAIN*, 16));  
 optionsLabel.setBounds(50, 150, 200, 25);  
 add(optionsLabel);  
  
 bookAppointmentButton = new JButton("Book an Appointment");  
 bookAppointmentButton.setBounds(50, 200, 200, 25);  
 bookAppointmentButton.addActionListener(this);  
 add(bookAppointmentButton);  
  
 payBillButton = new JButton("Pay Bill");  
 payBillButton.setBounds(50, 250, 200, 25);  
 payBillButton.addActionListener(this);  
 add(payBillButton);  
  
 patientMedicineButton = new JButton("Patient Medicine");  
 patientMedicineButton.setBounds(50, 300, 200, 25);  
 patientMedicineButton.addActionListener(this);  
 add(patientMedicineButton);  
  
 backButton = new JButton("Back");  
 backButton.setBounds(50, 400, 100, 25);  
 backButton.addActionListener(this);  
 add(backButton);  
  
 setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 setVisible(true);  
 }  
  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == bookAppointmentButton) {  
 // Redirect to BookAppointment class  
 new BookAppointment(username);  
 dispose();  
 } else if (e.getSource() == payBillButton) {  
 // Redirect to PayBill class  
 new PayBill(username);  
 dispose();  
 } else if (e.getSource() == patientMedicineButton) {  
 // Redirect to PatientMedicine class  
 new PatientMedicine(username);  
 dispose();  
 } else if (e.getSource() == backButton) {  
 // Redirect to Home page  
 new HospitalHomepage();  
 dispose();  
 }  
 }  
}

## **3.10 PayBill –**

package saini;  
  
import javax.swing.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.io.File;  
import java.io.FileNotFoundException;  
import java.io.FileWriter;  
import java.util.Scanner;  
  
public class PayBill extends JFrame implements ActionListener {  
 private final String patientUsername;  
 private final JLabel feeLabel;  
 private final JButton payButton;  
 private final JButton backButton;  
  
 public PayBill(String patientUsername) {  
 this.patientUsername = patientUsername;  
  
 // Set up the GUI  
 setTitle("Pay Fee");  
 setSize(400, 200);  
 setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
  
 JPanel panel = new JPanel();  
 getContentPane().add(panel);  
  
 feeLabel = new JLabel("Fee: " + getFee());  
 panel.add(feeLabel);  
  
 payButton = new JButton("PAY");  
 payButton.addActionListener(this);  
 panel.add(payButton);  
  
 backButton = new JButton("Back");  
 backButton.addActionListener(this);  
 panel.add(backButton);  
  
 setVisible(true);  
 }  
  
 // Helper method to retrieve the fee from the file  
 private String getFee() {  
 String fee = "N/A";  
 try {  
 File file = new File("Fee.txt");  
 Scanner scanner = new Scanner(file);  
 while (scanner.hasNextLine()) {  
 String line = scanner.nextLine();  
 String[] tokens = line.split(",");  
 if (tokens[1].equals(patientUsername)) {  
 fee = tokens[2];  
 break;  
 }  
 }  
 scanner.close();  
 } catch (FileNotFoundException e) {  
 e.printStackTrace();  
 }  
 return fee;  
 }  
  
 // Helper method to remove the fee information from the file  
 private void removeFee() {  
 try {  
 File inputFile = new File("Fee.txt");  
 File tempFile = new File("Fee\_temp.txt");  
 Scanner scanner = new Scanner(inputFile);  
 FileWriter writer = new FileWriter(tempFile);  
 while (scanner.hasNextLine()) {  
 String line = scanner.nextLine();  
 String[] tokens = line.split(",");  
 if (!tokens[1].equals(patientUsername)) {  
 writer.write(line + "\n");  
 }  
 }  
 scanner.close();  
 writer.close();  
 inputFile.delete();  
 tempFile.renameTo(inputFile);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == payButton) {  
 // Remove the fee information from the file  
 removeFee();  
  
 // Update the fee label  
 feeLabel.setText("Fee: " + getFee());  
  
 // Show a message dialog to confirm payment  
 JOptionPane.*showMessageDialog*(this, "Payment successful!");  
  
 } else if (e.getSource() == backButton) {  
 // Go back to the patient page  
 new PatientPage(patientUsername);  
 dispose();  
 }  
 }  
}

### **3.11 SeeAppointment –**

package saini;  
  
import java.awt.\*;  
import java.io.BufferedReader;  
import java.io.FileReader;  
import java.io.IOException;  
import javax.swing.\*;  
  
public class SeeAppointment extends JFrame {  
 private String doctorUsername;  
 private JTextArea appointmentList;  
  
 public SeeAppointment(String username) {  
 super("Appointments for " + username);  
 this.doctorUsername = username;  
 setDefaultCloseOperation(JFrame.*DISPOSE\_ON\_CLOSE*);  
 setPreferredSize(new Dimension(400, 300));  
  
 JPanel mainPanel = new JPanel(new BorderLayout());  
  
 appointmentList = new JTextArea(10, 30);  
 appointmentList.setEditable(false);  
 JScrollPane scrollPane = new JScrollPane(appointmentList);  
  
 mainPanel.add(scrollPane, BorderLayout.*CENTER*);  
  
 try {  
 displayAppointments();  
 } catch (IOException e) {  
 JOptionPane.*showMessageDialog*(null, "Error: " + e.getMessage(), "Error", JOptionPane.*ERROR\_MESSAGE*);  
 }  
  
 getContentPane().add(mainPanel);  
 pack();  
 setLocationRelativeTo(null);  
 setVisible(true);  
 }  
  
 private void displayAppointments() throws IOException {  
 String line;  
 BufferedReader reader = new BufferedReader(new FileReader("Appointment.txt"));  
 appointmentList.setText("Appointments for " + doctorUsername + ":\n\n");  
  
 while ((line = reader.readLine()) != null) {  
 String[] appointment = line.split(",");  
 if (appointment[0].equals(doctorUsername)) {  
 appointmentList.append("- " + appointment[1] + "\n");  
 }  
 }  
  
 reader.close();  
 }  
}

### **3.12 WriteMedicine –**

package saini;  
  
import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.\*;  
import java.io.\*;  
  
public class WriteMedicine extends JFrame implements ActionListener {  
 private JLabel patientUsernameLabel, diseaseLabel, medicine1Label, medicine2Label, medicine3Label;  
 private JTextField patientUsernameField, diseaseField, medicine1Field, medicine2Field, medicine3Field;  
 private JButton saveButton, backButton;  
 private String doctorUsername;  
  
 public WriteMedicine(String doctorUsername) {  
 this.doctorUsername = doctorUsername;  
 setTitle("Write Medicine");  
 setSize(400, 300);  
 setLocationRelativeTo(null);  
 setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
  
 Container c = getContentPane();  
 c.setLayout(new GridLayout(6, 2));  
  
 patientUsernameLabel = new JLabel("Patient Username:");  
 c.add(patientUsernameLabel);  
  
 patientUsernameField = new JTextField();  
 c.add(patientUsernameField);  
  
 diseaseLabel = new JLabel("Disease:");  
 c.add(diseaseLabel);  
  
 diseaseField = new JTextField();  
 c.add(diseaseField);  
  
 medicine1Label = new JLabel("Medicine 1:");  
 c.add(medicine1Label);  
  
 medicine1Field = new JTextField();  
 c.add(medicine1Field);  
  
 medicine2Label = new JLabel("Medicine 2:");  
 c.add(medicine2Label);  
  
 medicine2Field = new JTextField();  
 c.add(medicine2Field);  
  
 medicine3Label = new JLabel("Medicine 3:");  
 c.add(medicine3Label);  
  
 medicine3Field = new JTextField();  
 c.add(medicine3Field);  
  
 saveButton = new JButton("Save");  
 saveButton.addActionListener(this);  
 c.add(saveButton);  
  
 backButton = new JButton("Back");  
 backButton.addActionListener(this);  
 c.add(backButton);  
  
 setVisible(true);  
 }  
  
 public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == saveButton) {  
 String patientUsername = patientUsernameField.getText();  
 String disease = diseaseField.getText();  
 String medicine1 = medicine1Field.getText();  
 String medicine2 = medicine2Field.getText();  
 String medicine3 = medicine3Field.getText();  
 String medicineData = doctorUsername + "," + patientUsername + "," + disease + "," + medicine1 + "," + medicine2 + "," + medicine3 + "\n";  
 try {  
 BufferedWriter writer = new BufferedWriter(new FileWriter("Medicine.txt", true));  
 writer.write(medicineData);  
 writer.close();  
 JOptionPane.*showMessageDialog*(null, "Medicine data saved successfully");  
 dispose();  
 new ChargeFee(doctorUsername, patientUsername);  
 } catch (IOException ex) {  
 ex.printStackTrace();  
 }  
 } else if (e.getSource() == backButton) {  
 dispose();  
 new DoctorPage(doctorUsername);  
 }  
 }  
}

## **3.13 ChargeFee –**

package saini;  
  
import javax.swing.\*;  
import java.awt.event.\*;  
import java.io.\*;  
  
public class ChargeFee extends JFrame implements ActionListener {  
 private JLabel chargeLabel;  
 private JTextField chargeField;  
 private JButton submitButton, backButton;  
  
 private String doctorUsername, patientUsername;  
  
 public ChargeFee(String doctorUsername, String patientUsername) {  
 this.doctorUsername = doctorUsername;  
 this.patientUsername = patientUsername;  
  
 // Initialize components  
 chargeLabel = new JLabel("Charge:");  
 chargeField = new JTextField(10);  
 submitButton = new JButton("Submit");  
 backButton = new JButton("Back");  
  
 // Set layout  
 JPanel panel = new JPanel();  
 panel.setLayout(new BoxLayout(panel, BoxLayout.*PAGE\_AXIS*));  
  
 // Add components to the panel  
 panel.add(chargeLabel);  
 panel.add(chargeField);  
 panel.add(submitButton);  
 panel.add(backButton);  
  
 // Add action listeners  
 submitButton.addActionListener(this);  
 backButton.addActionListener(this);  
  
 // Set the frame properties  
 setTitle("Charge Fee");  
 setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 setSize(400, 150);  
 setLocationRelativeTo(null);  
 setResizable(false);  
 add(panel);  
 setVisible(true);  
 }  
  
 public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == submitButton) {  
 String charge = chargeField.getText();  
 if (charge.isEmpty()) {  
 JOptionPane.*showMessageDialog*(this, "Please enter the charge.");  
 } else {  
 // Write the fee information to the file  
 try {  
 FileWriter writer = new FileWriter("Fee.txt", true);  
 writer.write(doctorUsername + "," + patientUsername + "," + charge + "\n");  
 writer.close();  
 JOptionPane.*showMessageDialog*(this, "Fee charged successfully.");  
 } catch (IOException ex) {  
 JOptionPane.*showMessageDialog*(this, "Error occurred while writing to the file.");  
 }  
 }  
 } else if (e.getSource() == backButton) {  
 new HospitalHomepage();  
 setVisible(false);  
 }  
 }  
}

## **3.14 DeleteDoctor –**

package saini;  
  
import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.\*;  
import java.io.\*;  
  
public class DeleteDoctor extends JFrame implements ActionListener {  
 private JLabel label;  
 private JTextField usernameField;  
 private JButton deleteButton, backButton;  
  
 public DeleteDoctor() {  
 // Setting the frame properties  
 setTitle("Delete Doctor");  
 setSize(400, 200);  
 setLocationRelativeTo(null);  
 setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 setLayout(new BorderLayout());  
  
 // Creating a panel for the heading  
 JPanel headingPanel = new JPanel(new FlowLayout());  
 label = new JLabel("Delete Doctor Information");  
 label.setFont(new Font("Serif", Font.*BOLD*, 20));  
 headingPanel.add(label);  
 add(headingPanel, BorderLayout.*NORTH*);  
  
 // Creating a panel for the input fields and buttons  
 JPanel inputPanel = new JPanel(new GridLayout(2, 2, 10, 10));  
 JLabel usernameLabel = new JLabel("Username:");  
 usernameField = new JTextField();  
 deleteButton = new JButton("Delete");  
 deleteButton.addActionListener(this);  
 backButton = new JButton("Back");  
 backButton.addActionListener(this);  
 inputPanel.add(usernameLabel);  
 inputPanel.add(usernameField);  
 inputPanel.add(deleteButton);  
 inputPanel.add(backButton);  
 add(inputPanel, BorderLayout.*CENTER*);  
  
 // Setting the frame visible  
 setVisible(true);  
 }  
  
 // Action performed when a button is clicked  
 public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == deleteButton) {  
 String username = usernameField.getText();  
  
 if (username.isEmpty()) {  
 JOptionPane.*showMessageDialog*(null, "Please enter a valid username.");  
 return;  
 }  
  
 try {  
 // Reading the contents of the Doctor.txt file  
 File file = new File("Doctor.txt");  
 BufferedReader br = new BufferedReader(new FileReader(file));  
  
 // Creating a temporary file  
 File tempFile = new File("temp.txt");  
 BufferedWriter bw = new BufferedWriter(new FileWriter(tempFile));  
  
 String line;  
 boolean found = false;  
  
 // Writing all the lines except the one with the specified username to the temporary file  
 while ((line = br.readLine()) != null) {  
 String[] doctorData = line.split(",");  
 if (!doctorData[4].equals(username)) {  
 bw.write(line);  
 bw.newLine();  
 } else {  
 found = true;  
 }  
 }  
  
 bw.close();  
 br.close();  
  
 // Replacing the Doctor.txt file with the temporary file  
 if (found) {  
 file.delete();  
 tempFile.renameTo(new File("Doctor.txt"));  
 JOptionPane.*showMessageDialog*(null, "Doctor information deleted successfully.");  
 } else {  
 JOptionPane.*showMessageDialog*(null, "Doctor with username '" + username + "' not found.");  
 }  
  
 } catch (IOException ex) {  
 ex.printStackTrace();  
 }  
  
 } else if (e.getSource() == backButton) {  
 new AdminSignIn();  
 dispose();  
 }  
 }  
  
 public static void main(String[] args) {  
 new DeleteDoctor();  
 }  
}

## **3.15 DeletePatient –**

package saini;  
  
import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.\*;  
import java.io.\*;  
  
public class DeletePatient extends JFrame implements ActionListener {  
  
 private JLabel usernameLabel;  
 private JTextField usernameField;  
 private JButton deleteButton, backButton;  
  
 public DeletePatient() {  
 setTitle("Delete Patient");  
 setSize(400, 200);  
 setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 setLocationRelativeTo(null);  
  
 Container container = getContentPane();  
 container.setLayout(new BorderLayout());  
  
 JPanel centerPanel = new JPanel(new GridLayout(1,2));  
 usernameLabel = new JLabel("Enter Patient Username:");  
 usernameField = new JTextField();  
 centerPanel.add(usernameLabel);  
 centerPanel.add(usernameField);  
 container.add(centerPanel, BorderLayout.*CENTER*);  
  
 JPanel bottomPanel = new JPanel(new FlowLayout());  
 deleteButton = new JButton("Delete");  
 deleteButton.addActionListener(this);  
 backButton = new JButton("Back");  
 backButton.addActionListener(this);  
 bottomPanel.add(deleteButton);  
 bottomPanel.add(backButton);  
 container.add(bottomPanel, BorderLayout.*SOUTH*);  
  
 setVisible(true);  
 }  
  
 public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == deleteButton) {  
 String patientUsername = usernameField.getText().trim();  
  
 try {  
 File inputFile = new File("Patient.txt");  
 File tempFile = new File("temp.txt");  
 BufferedReader reader = new BufferedReader(new FileReader(inputFile));  
 BufferedWriter writer = new BufferedWriter(new FileWriter(tempFile));  
 String line;  
 boolean deleted = false;  
 while ((line = reader.readLine()) != null) {  
 String[] data = line.split(",");  
 if (data[3].equals(patientUsername)) {  
 deleted = true;  
 continue;  
 }  
 writer.write(line + System.*getProperty*("line.separator"));  
 }  
 reader.close();  
 writer.close();  
 if (!deleted) {  
 JOptionPane.*showMessageDialog*(this, "Patient not found.");  
 } else {  
 inputFile.delete();  
 tempFile.renameTo(inputFile);  
 JOptionPane.*showMessageDialog*(this, "Patient deleted successfully.");  
 }  
 } catch (IOException ex) {  
 ex.printStackTrace();  
 }  
 } else if (e.getSource() == backButton) {  
 dispose();  
 new AdminSignIn();  
 }  
 }  
}

## **3.16 BookAppointment –**

package saini;  
  
import javax.swing.\*;  
import java.awt.event.\*;  
import java.io.\*;  
  
public class BookAppointment extends JFrame implements ActionListener {  
 private JLabel doctorLabel;  
 private JTextField doctorField;  
 private JButton bookButton, backButton;  
 private String patientUsername;  
  
 public BookAppointment(String patientUsername) {  
 this.patientUsername = patientUsername;  
  
 // create UI elements  
 doctorLabel = new JLabel("Doctor Name:");  
 doctorField = new JTextField();  
 bookButton = new JButton("Book Appointment");  
 backButton = new JButton("Back to Home");  
  
 // set layout  
 setLayout(new BoxLayout(getContentPane(), BoxLayout.*Y\_AXIS*));  
  
 // add UI elements to content pane  
 add(doctorLabel);  
 add(doctorField);  
 add(bookButton);  
 add(backButton);  
  
 // add action listeners  
 bookButton.addActionListener(this);  
 backButton.addActionListener(this);  
  
 // set frame properties  
 setTitle("Book Appointment");  
 setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 setSize(400, 200);  
 setLocationRelativeTo(null); // center the frame  
 setVisible(true);  
 }  
  
 public void actionPerformed(ActionEvent e) {  
 if (e.getSource() == bookButton) {  
 String doctorName = doctorField.getText();  
 String doctorUsername = null;  
  
 try {  
 // search for doctor's username in Doctor.txt  
 BufferedReader reader = new BufferedReader(new FileReader("Doctor.txt"));  
 String line;  
 while ((line = reader.readLine()) != null) {  
 String[] fields = line.split(",");  
 String name = fields[0];  
 String username = fields[4];  
  
 if (name.equalsIgnoreCase(doctorName)) {  
 doctorUsername = username;  
 break;  
 }  
 }  
 reader.close();  
  
 if (doctorUsername == null) {  
 JOptionPane.*showMessageDialog*(this, "Doctor not found");  
 return;  
 }  
  
 // append appointment to Appointment.txt  
 BufferedWriter writer = new BufferedWriter(new FileWriter("Appointment.txt", true));  
 writer.write(doctorUsername + "," + patientUsername);  
 writer.newLine();  
 writer.close();  
  
 JOptionPane.*showMessageDialog*(this, "Appointment booked successfully");  
  
 } catch (IOException ex) {  
 JOptionPane.*showMessageDialog*(this, "Error: " + ex.getMessage());  
 ex.printStackTrace();  
 }  
  
 } else if (e.getSource() == backButton) {  
 // go back to home page  
 new HospitalHomepage();  
 dispose();  
 }  
 }  
}

## **3.17 PatientMedicine –**

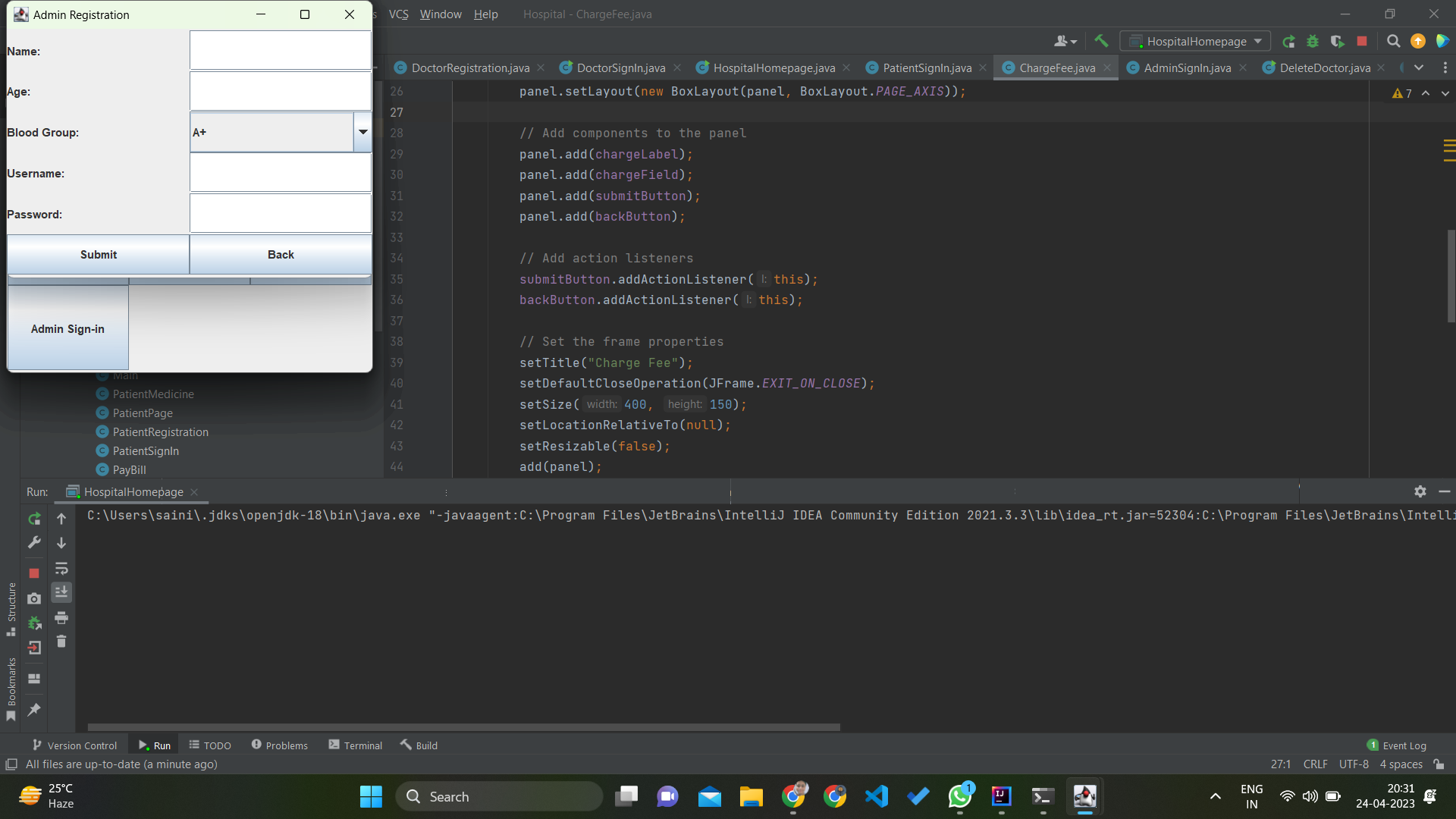
package saini;  
  
import javax.swing.\*;  
import java.awt.\*;  
import java.io.\*;  
  
public class PatientMedicine extends JFrame {  
 private JTextArea medicineArea;  
  
 public PatientMedicine(String username) {  
 setTitle("Patient Medicine");  
 setSize(500, 500);  
 setLocationRelativeTo(null);  
  
 JPanel panel = new JPanel();  
 panel.setLayout(new BorderLayout());  
  
 JLabel heading = new JLabel("Medicines Prescribed for " + username);  
 heading.setFont(new Font("Arial", Font.*BOLD*, 18));  
 heading.setHorizontalAlignment(JLabel.*CENTER*);  
 panel.add(heading, BorderLayout.*NORTH*);  
  
 medicineArea = new JTextArea();  
 medicineArea.setEditable(false);  
 JScrollPane scrollPane = new JScrollPane(medicineArea);  
 panel.add(scrollPane, BorderLayout.*CENTER*);  
  
 JButton backButton = new JButton("Back to Home");  
 backButton.addActionListener(e -> {  
 new HospitalHomepage();  
 dispose();  
 });  
 panel.add(backButton, BorderLayout.*SOUTH*);  
  
 try {  
 File file = new File("Medicine.txt");  
 if (file.exists()) {  
 BufferedReader reader = new BufferedReader(new FileReader(file));  
 String line = reader.readLine();  
 while (line != null) {  
 String[] medicineInfo = line.split(",");  
 if (medicineInfo[1].equals(username)) {  
 medicineArea.append("Doctor: " + medicineInfo[0] + "\n");  
 medicineArea.append("Disease: " + medicineInfo[2] + "\n");  
 medicineArea.append("Medicine 1: " + medicineInfo[3] + "\n");  
 medicineArea.append("Medicine 2: " + medicineInfo[4] + "\n");  
 medicineArea.append("Medicine 3: " + medicineInfo[5] + "\n\n");  
 }  
 line = reader.readLine();  
 }  
 reader.close();  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
 add(panel);  
 setVisible(true);  
 setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 }  
}

### **Output**

Table

Description automatically generated

**Figure 1- Hospital Homepage**



**Figure 2- Admin Registration**

Graphical user interface, application

Description automatically generated

**Figure 3-Admin SignIn**

Graphical user interface, application

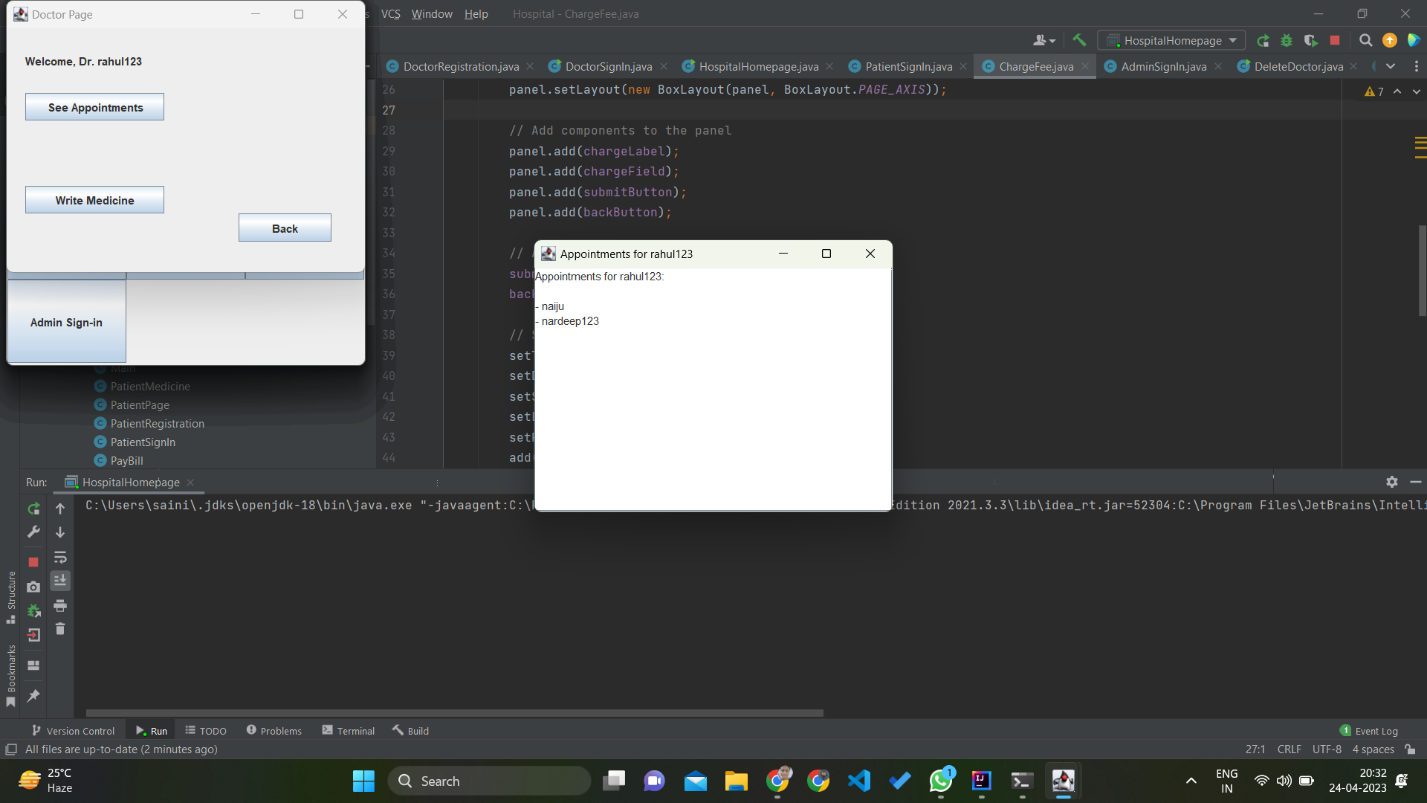
Description automatically generated

**Figure 4-Doctor Registration**

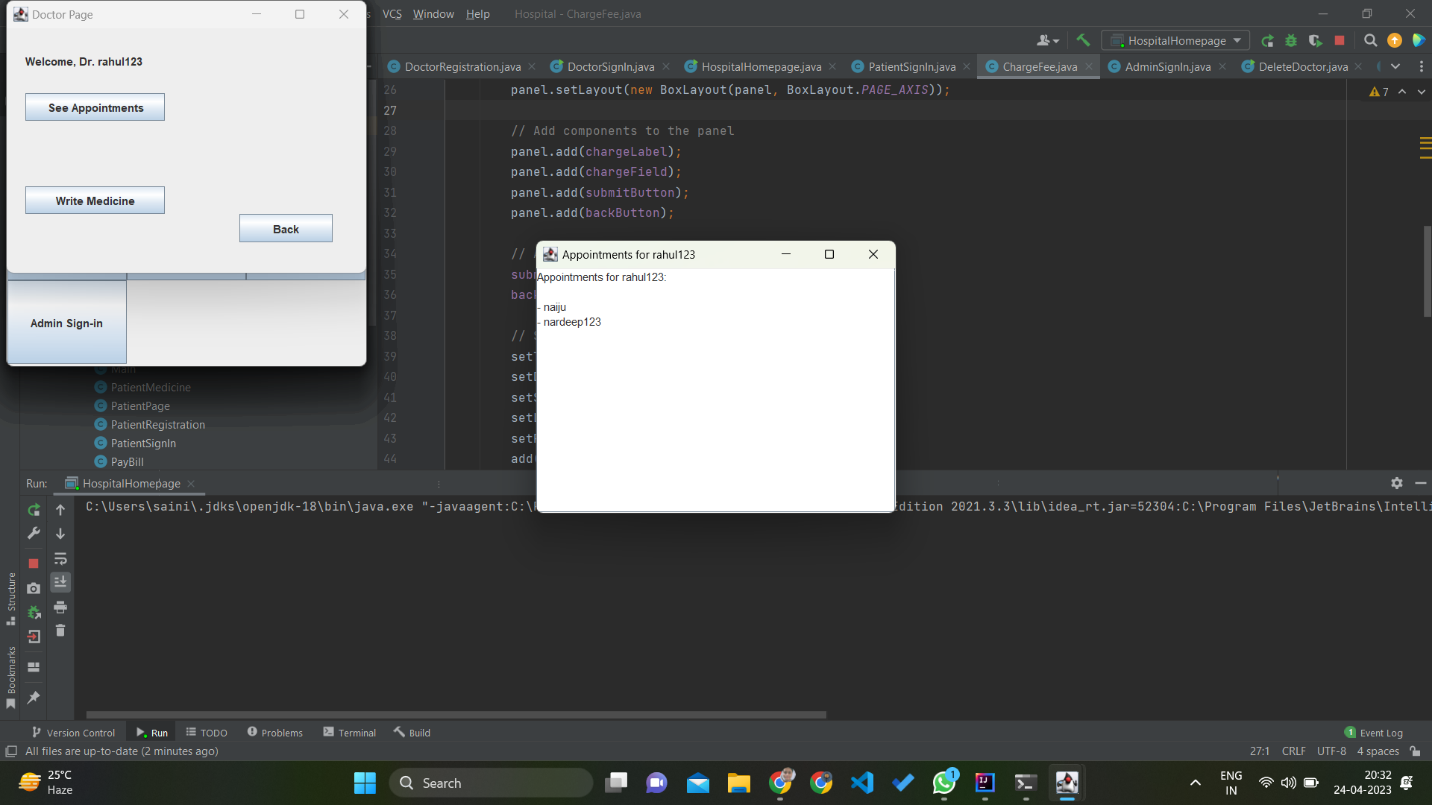
A screenshot of a computer

Description automatically generated

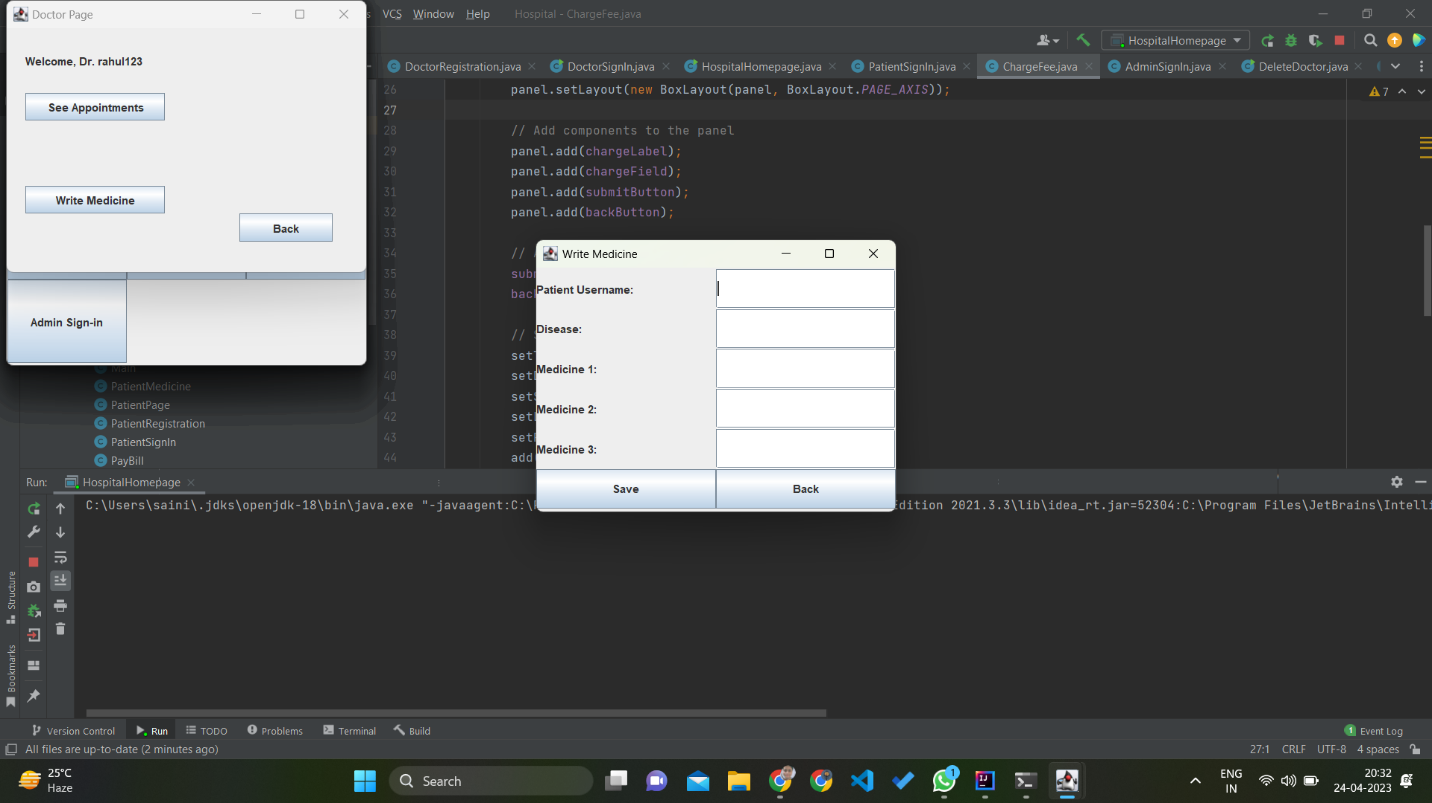
**Figure 5- Doctor SignIn**



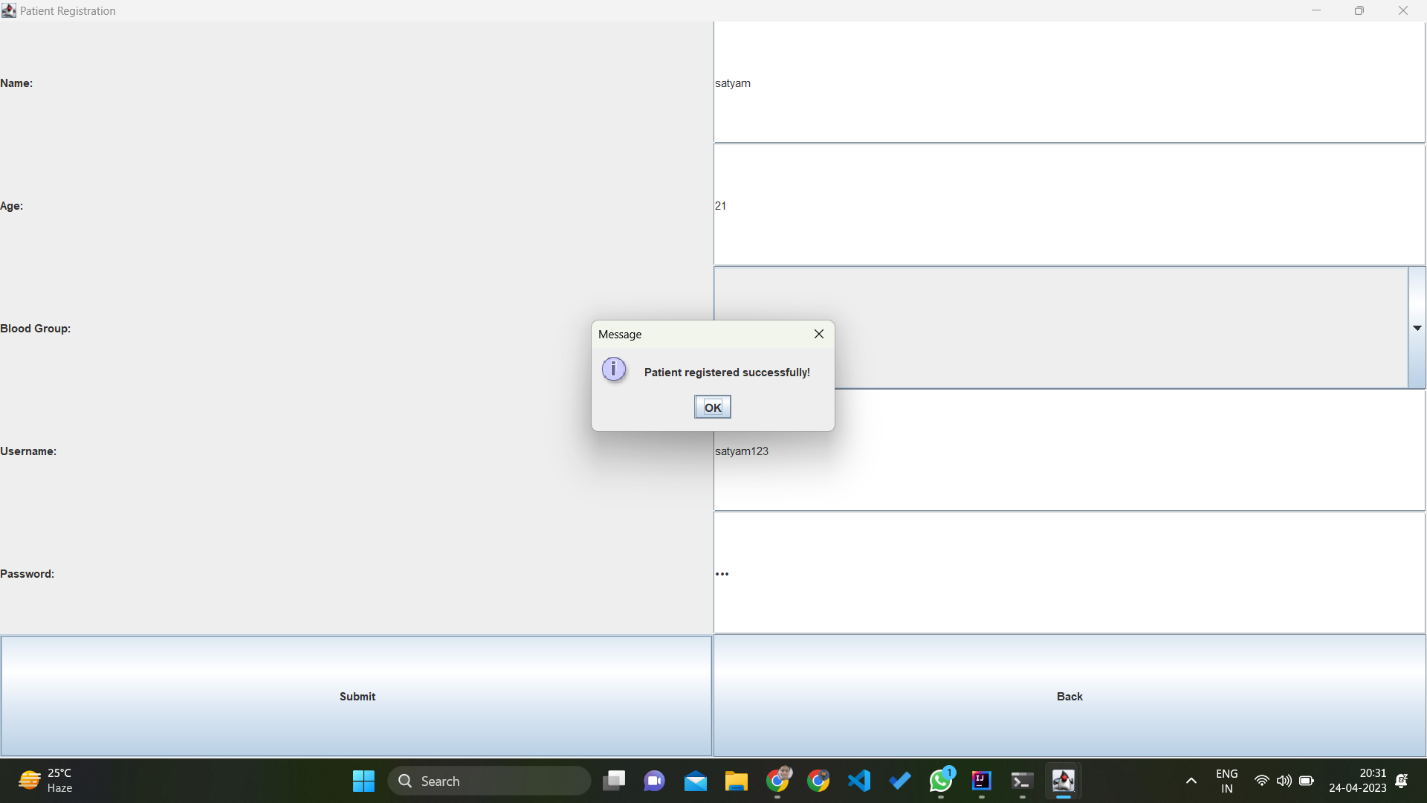
**Figure 6- Doctor Page**



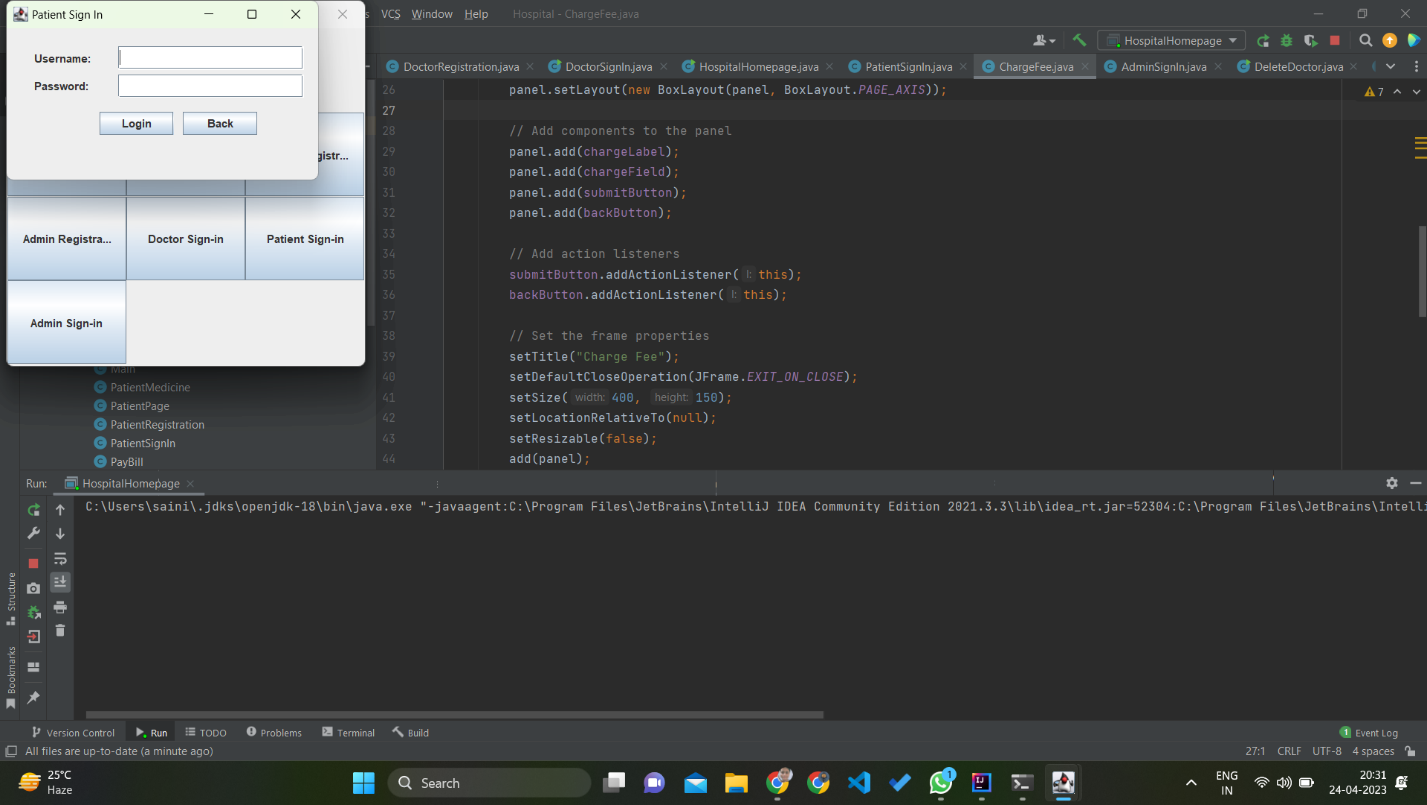
**Figure 7- See Appointments**



**Figure 8-Write Medicine**



**Figure 9-Patient Registration**

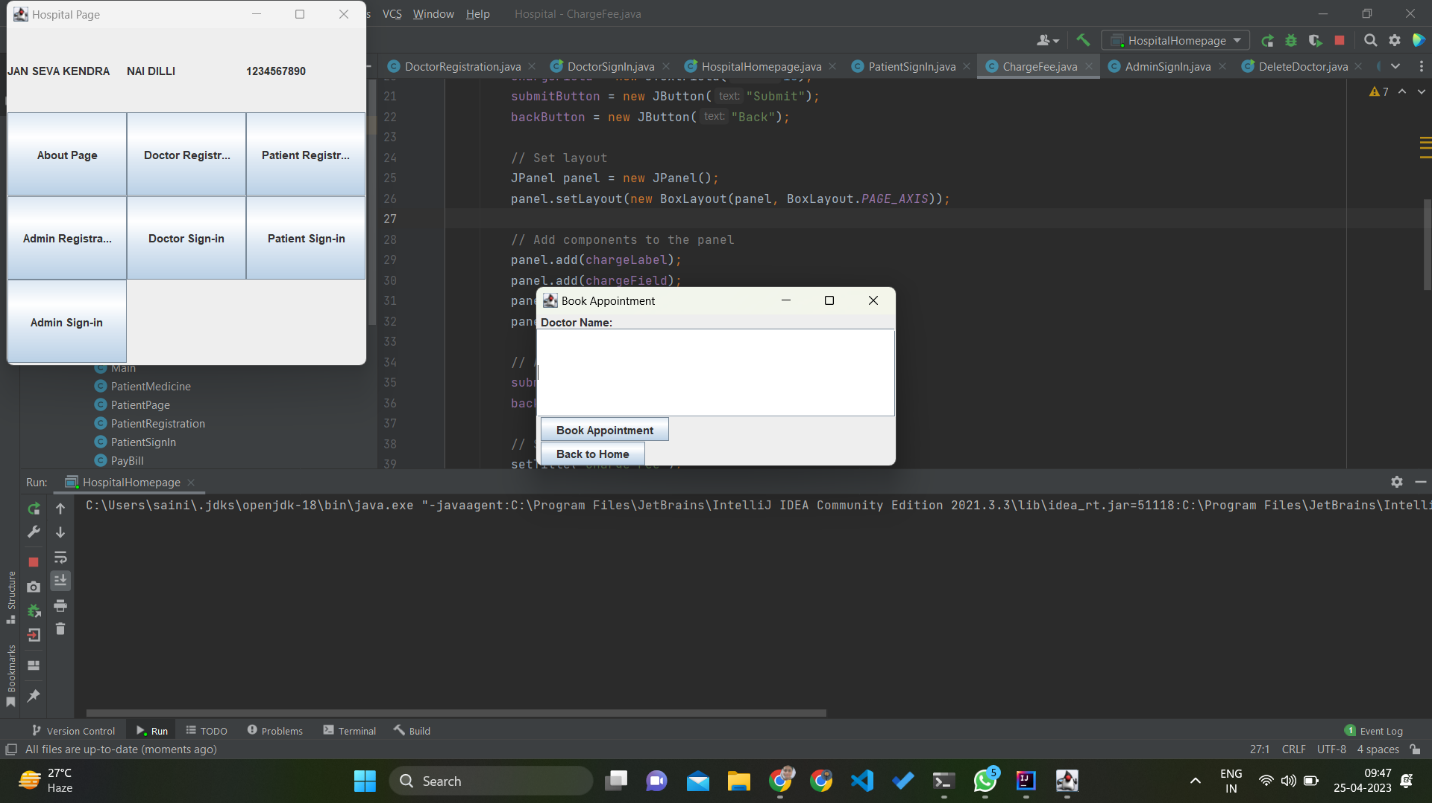


**Figure 10-Patient SignIn**

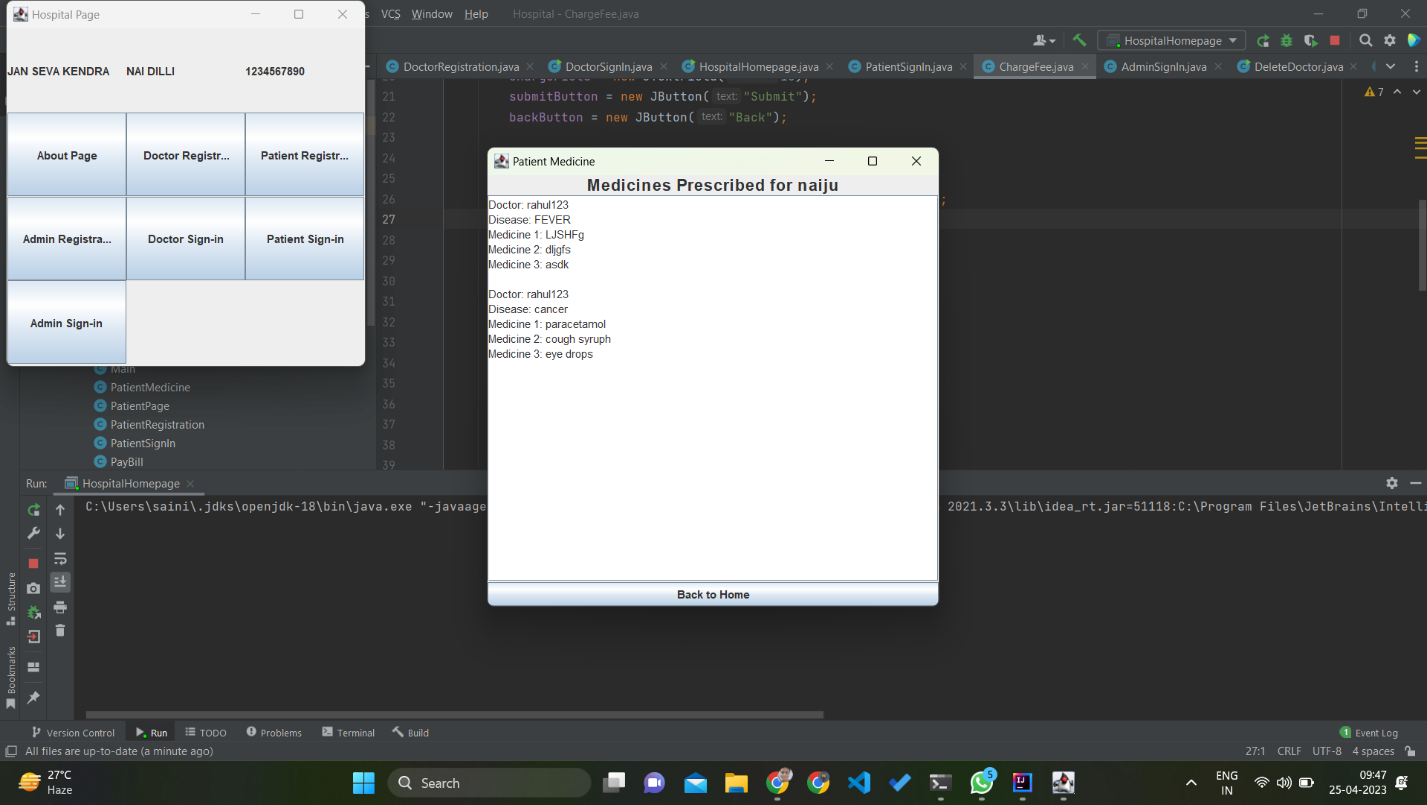
Graphical user interface, text

Description automatically generated

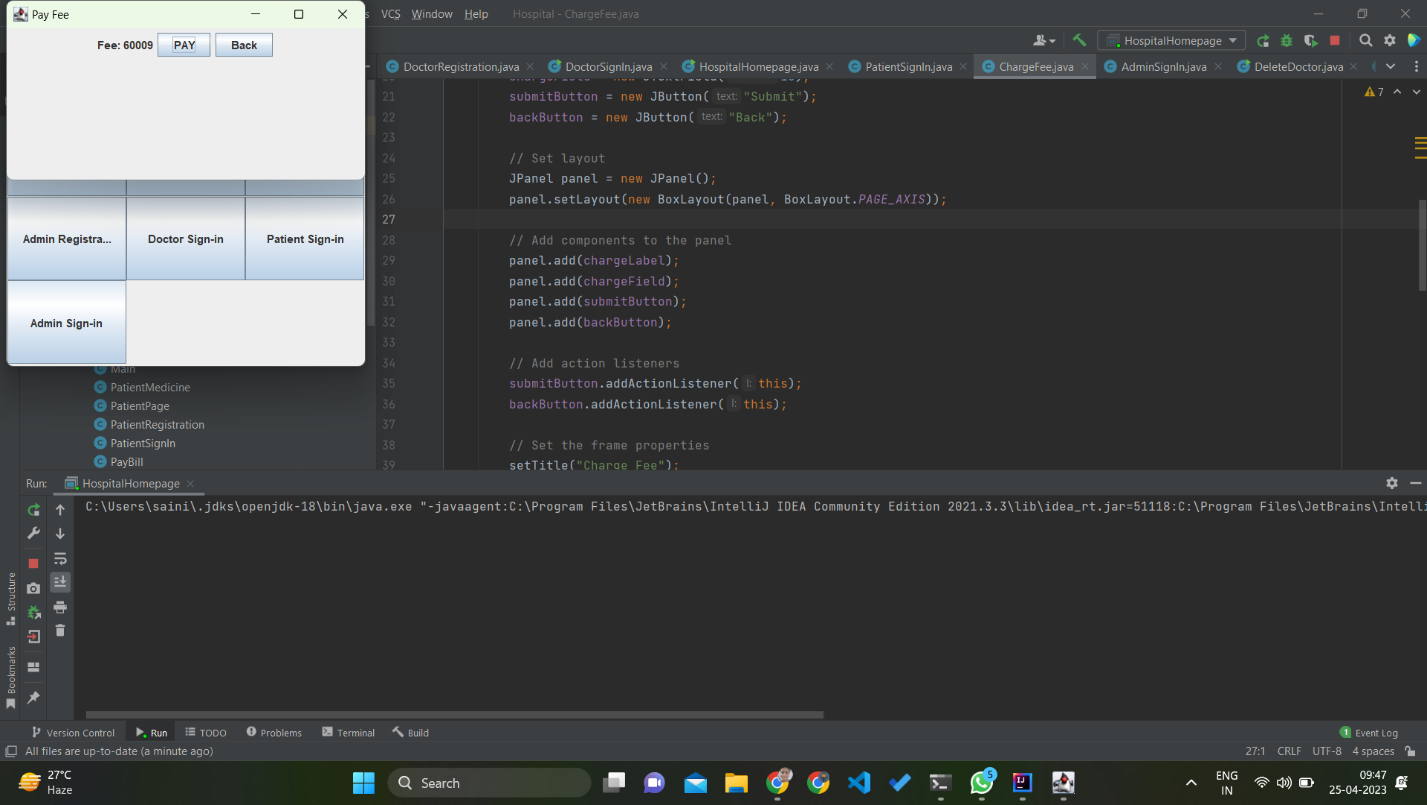
**Figure 11-Patient Page**



**Figure 12-Book Appointment**



**Figure 13-Patient Medicine**



**Figure 14-Pay Fee**

### **Conclusion**

In conclusion, the Hospital Management System project is an excellent example of how Java Swing and File Management can be used to create a powerful and user-friendly software solution for managing hospital operations.

By automating many of the manual processes involved in patient care, the Hospital Management System project can help hospitals improve their efficiency and provide better patient outcomes.

The project's user-friendly interface, combined with its advanced features such as appointment scheduling, patient management, billing and invoicing, and report generation, makes it a valuable asset for any hospital looking to improve its operational efficiency.

Furthermore, the File Management feature of the project ensures that all important documents and records are stored in a centralized location, making it easy for hospital staff to access and manage data securely.

Overall, the Hospital Management System project is a valuable addition to any hospital's technology infrastructure, and its use of Java Swing and File Management demonstrates the power and versatility of these technologies in developing robust software solutions.