Mini-project report

Project Title:

Hospital Management System

Submitted in partial fulfillment of the requirements of the Course on

Object-Oriented Programming using Java (BTCS-T-ES-301)

of

Bachelor of Technology (B-Tech)

Department of

Computer Science Engineering

Gandhi Institute for Technology (GIFT), Bhubaneswar

Name of Student	Registration No	Semester	Branch
Mohanty Hitesh Rabindranath	**2201298111**	**3rd**	**CSE**

Dr. Satya Ranjan Pattanaik

(Name of the Guide & Head of the Department)

CERTIFICATE

This is to certify that the mini-project entitled “Hospital Management System” has been carried out by 1. Mohanty Hitesh Rabindranath (2201298111) and completed under my guidance and the project meets the academic requirement of the subject Object-Oriented Programming using Java for Computer Science (BTCS-T-ES-301).

Signature of the guide

Table of Contents

Abstract

The Hospital Management System (HMS) is a comprehensive software solution developed using Java programming language and SQL (Structured Query Language) for database management. This system aims to streamline and enhance the efficiency of hospital operations, facilitating better patient care, and administrative management.

The HMS is designed to automate various tasks within a hospital environment, including patient registration, appointment scheduling, managing medical records, billing, inventory management, and generating reports. It provides a user-friendly interface for different stakeholders, such as administrators, doctors, nurses, and patients, to interact with the system based on their respective roles and permissions.

Introduction

A Hospital Management System is a software application that automates various processes and tasks involved in managing a hospital's daily operations. The system provides a comprehensive solution for managing patient information, appointments, medical history, billing, and other administrative tasks. This Java-based Hospital Management System project aims to provide a reliable and efficient platform for hospitals to manage their operations. The system will enable healthcare providers to streamline their daily tasks, improve patient care, and increase operational efficiency. The system will feature a user-friendly interface that will be easy to navigate, making it accessible to hospital staff with varying levels of technical expertise. The project will also integrate various modules, such as patient registration, doctor appointment scheduling, electronic medical records, pharmacy management, and billing. Overall, this Hospital Management System project will be a valuable tool for healthcare providers to manage their daily operations efficiently, providing better patient care, and improving the overall quality of healthcare services.

Literature Survey

1.“A hospital resource and patient management system based on real-time data capture and intelligent decision making” Author(s): Musa, A. Lancashire Bus. Sch. Systems and Informatics (ICSAI), 2012 International Conference.

This paper highlights such limitations of existing systems and proposes a RFID (Radio Frequency ID) and wireless sensor based, location and information management framework that facilitates real time tracking of hospital assets, personnel and patients as they move through pre-set procedures as part of daily activities of the hospitals.

2. International Journal of Engineering and Management Research: e-ISSN: 2250-0758, p-ISSN: 2394-6962, Volume-12, Issue-5 (October 2022)

This Article says, the existing system requires a lot of time. Absence of security components Every task needs to be completed by hand. The majority of tasks and activities depend on specialists and human resources. No direct communication with the senior officers. The accuracy level is subjective. High expense is required for manual system management. Difficulty in getting backup data and transfer data.

3. Hospital Management Maturity Models: Literature Review Modelos de Maturidade de Gestão Hospitalar: Revisão da Literatura

This ensure that only relevant records were analyzed, some inclusion and exclusion criteria were established, such as: research objectives, research methods, focus and scope, and characteristics of model design. Decisions on the inclusion and exclusion are subjective.

Problem Statement

The manual system for managing hospital operations is prone to errors and inefficient, resulting in delays in patient care, inaccurate record-keeping, and decreased operational efficiency. To overcome these challenges, a digital solution is required that can streamline hospital management processes and improve the overall quality of healthcare services. This project aims to develop a Java-based Hospital Management System that can automate various administrative tasks, providing healthcare providers with a reliable and efficient platform for managing their daily operations.

- **Manual and Paper-Based Records:**

Many healthcare facilities still rely on manual documentation and paper-based records, leading to inefficiencies, data redundancy, and the risk of errors during record maintenance.

- **Data Discrepancies and Inconsistencies**:

The absence of a centralized, standardized system often leads to data discrepancies, duplication, or inconsistencies across different departments, compromising the accuracy and reliability of information.

- **Limited Accessibility and Data Retrieval Challenges:**

Inefficient data retrieval mechanisms hinder quick access to critical patient records or doctor details, impacting the speed and quality of patient care.

- **Inadequate Integration and Communication**:

Lack of integration between various departments within a hospital, results in communication gaps, hindering the seamless flow of information between administrators, doctors, and other staff members.

Methodology

The methodology for developing the Hospital Management System project will follow a structured approach that includes the following stages: 1. Requirements Gathering: The first stage will involve identifying the system's requirements, including the different modules needed, the user interface, and the system's performance specifications. This stage will involve interacting with stakeholders such as hospital staff, patients, and administrators to identify their needs. 2. Design: Once the requirements are identified, the next stage will involve designing the system's architecture and user interface. This stage will also involve selecting the appropriate programming language and software tools needed to develop the system. 3. Implementation: The implementation stage will involve developing the system's various modules and integrating them into a functional system. The system will be developed using the Java programming language and will leverage various software development tools such as Visual Studio Code, MySQL etc. Page | 3 4. Testing: After the system's implementation, the next stage will involve testing the system's functionality and performance. This stage will include various testing methods such as unit testing, integration testing, and system testing. 5. Deployment: Once the system has been tested and validated, the final stage will involve deploying the system to the hospital's servers. This stage will involve configuring the system to work seamlessly with the hospital's existing IT infrastructure and training hospital staff on how to use the system.

Hardware Interfaces

▪ **Laptop/Desktop PC** - Purpose of this is to give information when Patients ask information about doctors, medicine available lab tests etc. To perform such actions, it needs very efficient computer otherwise due to that reason patients have to wait for a long time to get what they ask for.

▪ **Laser Printer (B/W)** - This device is for printing patients’ info etc.

▪ **Wi-Fi router** - Wi-Fi router is used to for internetwork operations inside of a hospital and simply data transmission from pcs to sever

Software Interfaces

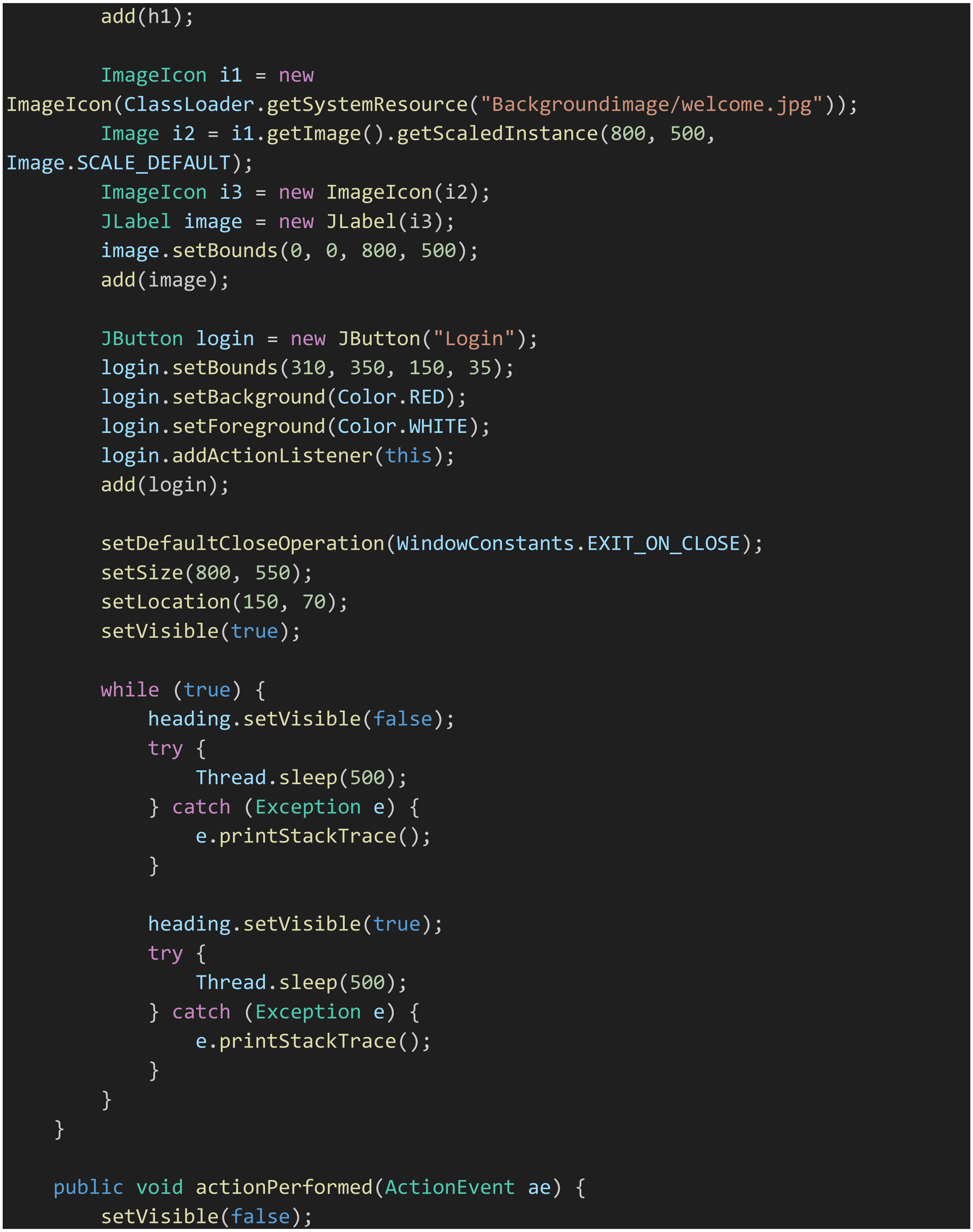
- **JDK 1.8** - Java is fast, secure, and reliable. From laptops to data centers, game consoles to scientific supercomputers, cell phones to the Internet,
- **MySQL server**- Database connectivity and management
- **OS Windows 7/8/8.1-** Very user friendly and common OS
- **JRE 1.8** - JAVA Runtime Environment for run Java Application and System

SOURCE CODE

Following are the screens of the Hospital management system where you can see all the features of this system in use and you can also see the GUI of the system. I. Welcome Screen:

1. Welcome Screen:

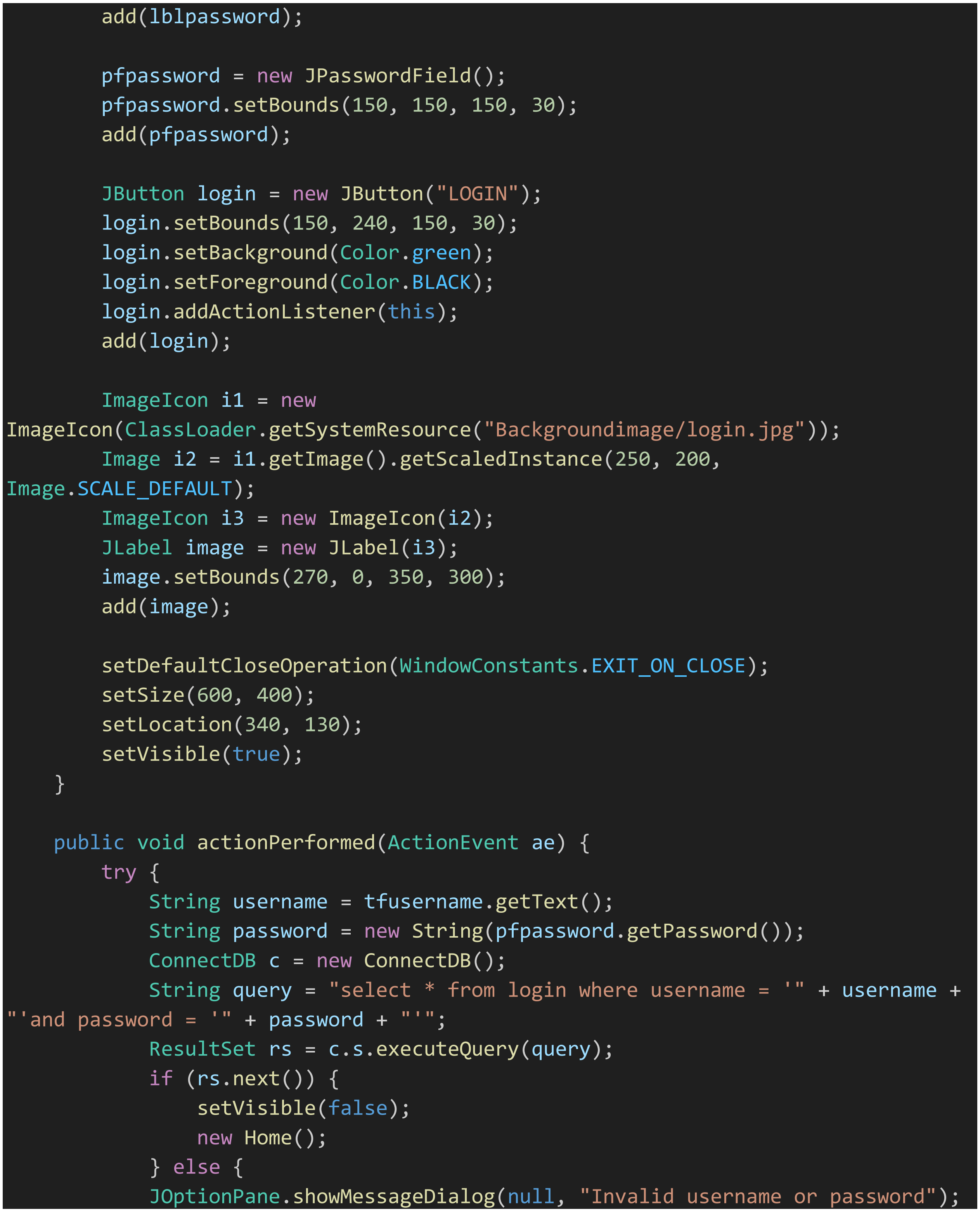
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class welcome extends JFrame implements ActionListener {
    public welcome() {

        getContentPane().setBackground(Color.WHITE);
        setLayout(null);

        JLabel heading = new JLabel("HOSPITAL MANAGEMENT SYSTEM");
        heading.setBounds(55, 40, 1000, 50);
        heading.setFont(new Font("SAN_SERIF", Font.BOLD, 40));
        heading.setForeground(Color.RED);
        add(heading);

        JLabel h1 = new JLabel("HOSPITAL MANAGEMENT SYSTEM");
        h1.setBounds(55, 40, 800, 50);
        h1.setFont(new Font("SAN_SERIF", Font.BOLD, 40));
        h1.setForeground(Color.BLUE);
```



```
        add(h1);

        ImageIcon i1 = new
ImageIcon(ClassLoader.getSystemResource("Backgroundimage/welcome.jpg"));
        Image i2 = i1.getImage().getScaledInstance(800, 500,
Image.SCALE_DEFAULT);
        ImageIcon i3 = new ImageIcon(i2);
        JLabel image = new JLabel(i3);
        image.setBounds(0, 0, 800, 500);
        add(image);

        JButton login = new JButton("Login");
        login.setBounds(310, 350, 150, 35);
        login.setBackground(Color.RED);
        login.setForeground(Color.WHITE);
        login.addActionListener(this);
        add(login);

        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        setSize(800, 550);
        setLocation(150, 70);
        setVisible(true);

        while (true) {
            heading.setVisible(false);
            try {
                Thread.sleep(500);
            } catch (Exception e) {
                e.printStackTrace();
            }

            heading.setVisible(true);
            try {
                Thread.sleep(500);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }

    public void actionPerformed(ActionEvent ae) {
        setVisible(false);
        new Login();
    }

    public static void main(String args[]) {
        new welcome();
    }
}
```

2. Login page:

```
import javax.swing.*;

import java.awt.*;
import java.awt.event.*;
import java.sql.*;

public class Login extends JFrame implements ActionListener {

    JTextField tfusername;
    JPasswordField pfpassword;

    public Login() {

        getContentPane().setBackground(Color.WHITE);
        setLayout(null);
        JLabel heading = new JLabel("Login");
        heading.setBounds(230, 15, 300, 50);
        heading.setFont(new Font("SAN_SERIF", Font.BOLD, 35));
        heading.setForeground(Color.darkGray);
        add(heading);

        JLabel lblusername = new JLabel("Username");
        lblusername.setBounds(40, 100, 100, 30);
        add(lblusername);

        tfusername = new JTextField();
        tfusername.setBounds(150, 100, 150, 30);
        add(tfusername);

        JLabel lblpassword = new JLabel("Password");
        lblpassword.setBounds(40, 150, 100, 30);
```

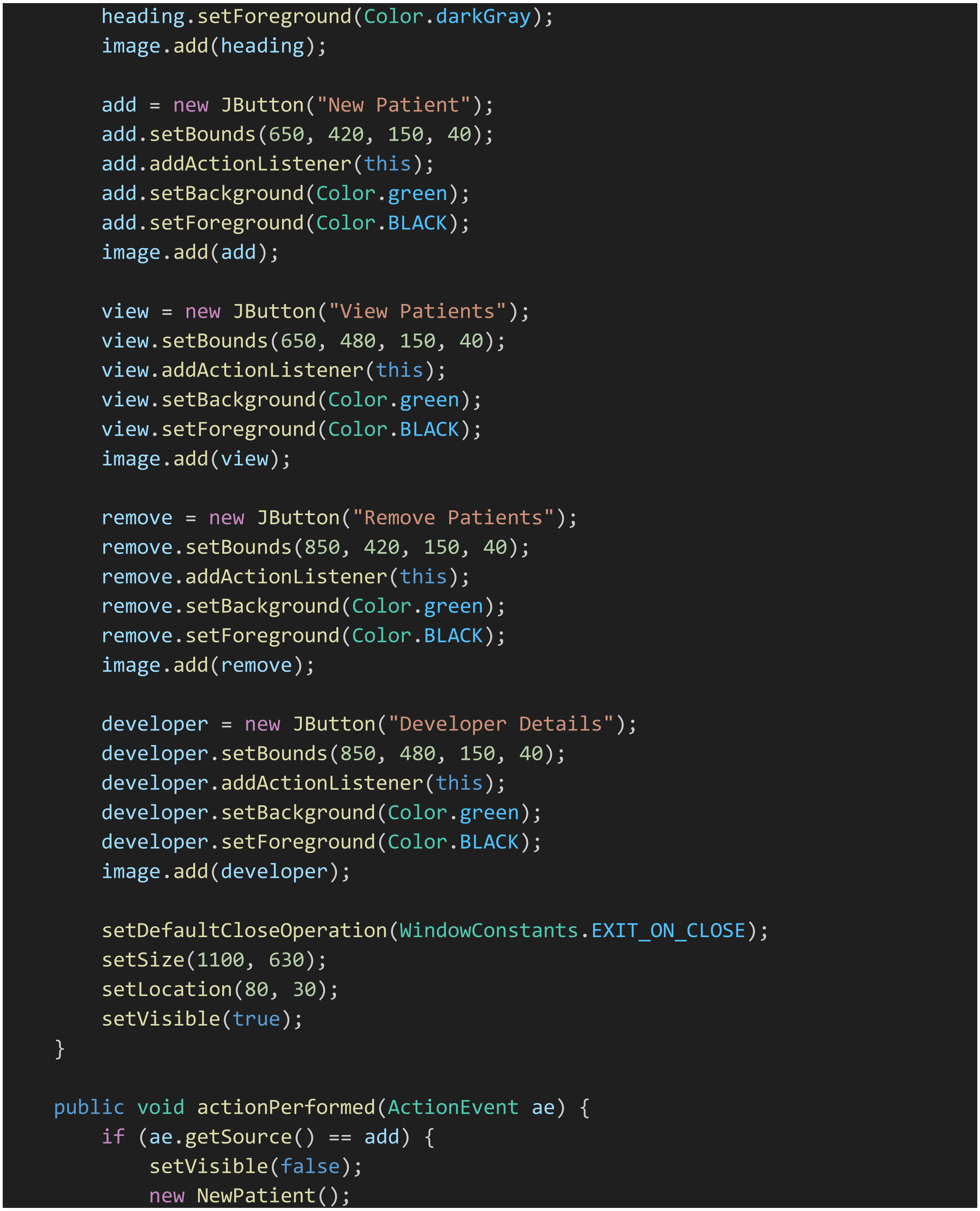


```
        add(lblpassword);

        pfpassword = new JPasswordField();
        pfpassword.setBounds(150, 150, 150, 30);
        add(pfpassword);

        JButton login = new JButton("LOGIN");
        login.setBounds(150, 240, 150, 30);
        login.setBackground(Color.green);
        login.setForeground(Color.BLACK);
        login.addActionListener(this);
        add(login);

        ImageIcon i1 = new
ImageIcon(ClassLoader.getSystemResource("Backgroundimage/login.jpg"));
        Image i2 = i1.getImage().getScaledInstance(250, 200,
Image.SCALE_DEFAULT);
        ImageIcon i3 = new ImageIcon(i2);
        JLabel image = new JLabel(i3);
        image.setBounds(270, 0, 350, 300);
        add(image);

        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        setSize(600, 400);
        setLocation(340, 130);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae) {
        try {
            String username = tfusername.getText();
            String password = new String(pfpassword.getPassword());
            ConnectDB c = new ConnectDB();
            String query = "select * from login where username = '" + username +
"'and password = '" + password + "'";
            ResultSet rs = c.s.executeQuery(query);
            if (rs.next()) {
                setVisible(false);
                new Home();
            } else {
            JOptionPane.showMessageDialog(null, "Invalid username or password");
```

```
                tfusername.setText("");
                pfpassword.setText("");
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        new Login();
    }
}
```

3.Home Page:

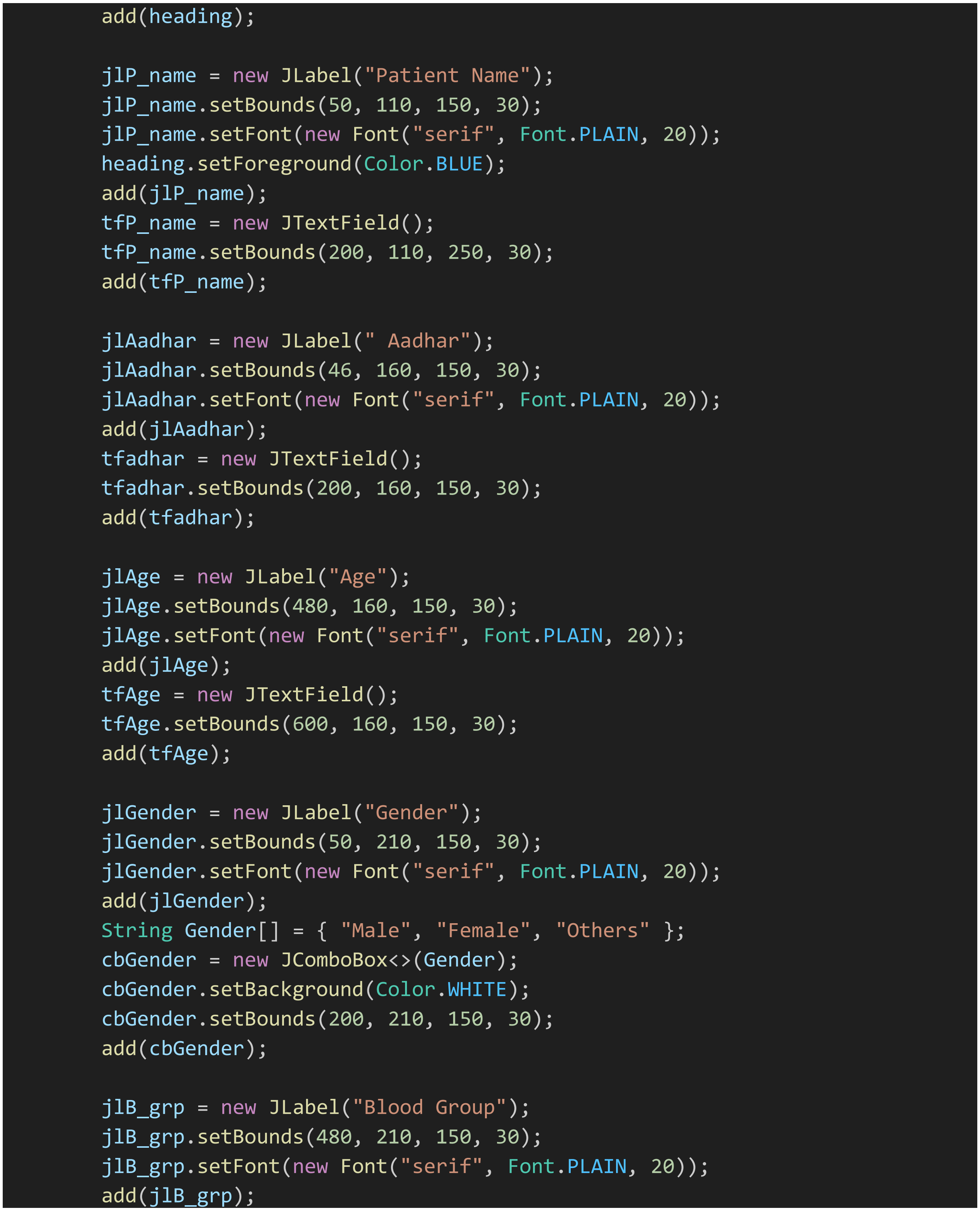
```
import javax.swing.*;

import java.awt.*;
import java.awt.event.*;

public class Home extends JFrame implements ActionListener {

    JButton view, add, developer, remove;

    public Home() {

        getContentPane().setBackground(Color.CYAN);
        setLayout(null);

        ImageIcon i1 = new
ImageIcon(ClassLoader.getSystemResource("Backgroundimage/Aesthetic.jpg"));
        Image i2 = i1.getImage().getScaledInstance(1120, 630,
Image.SCALE_DEFAULT);
        ImageIcon i3 = new ImageIcon(i2);
        JLabel image = new JLabel(i3);
        image.setBounds(0, 0, 1090, 600);
        add(image);

        JLabel heading = new JLabel("Gandhi Hospital, Bhubaneswar");
        heading.setBounds(40, 30, 800, 50);
        heading.setFont(new Font("SAN_SERIF", Font.BOLD, 50));
```



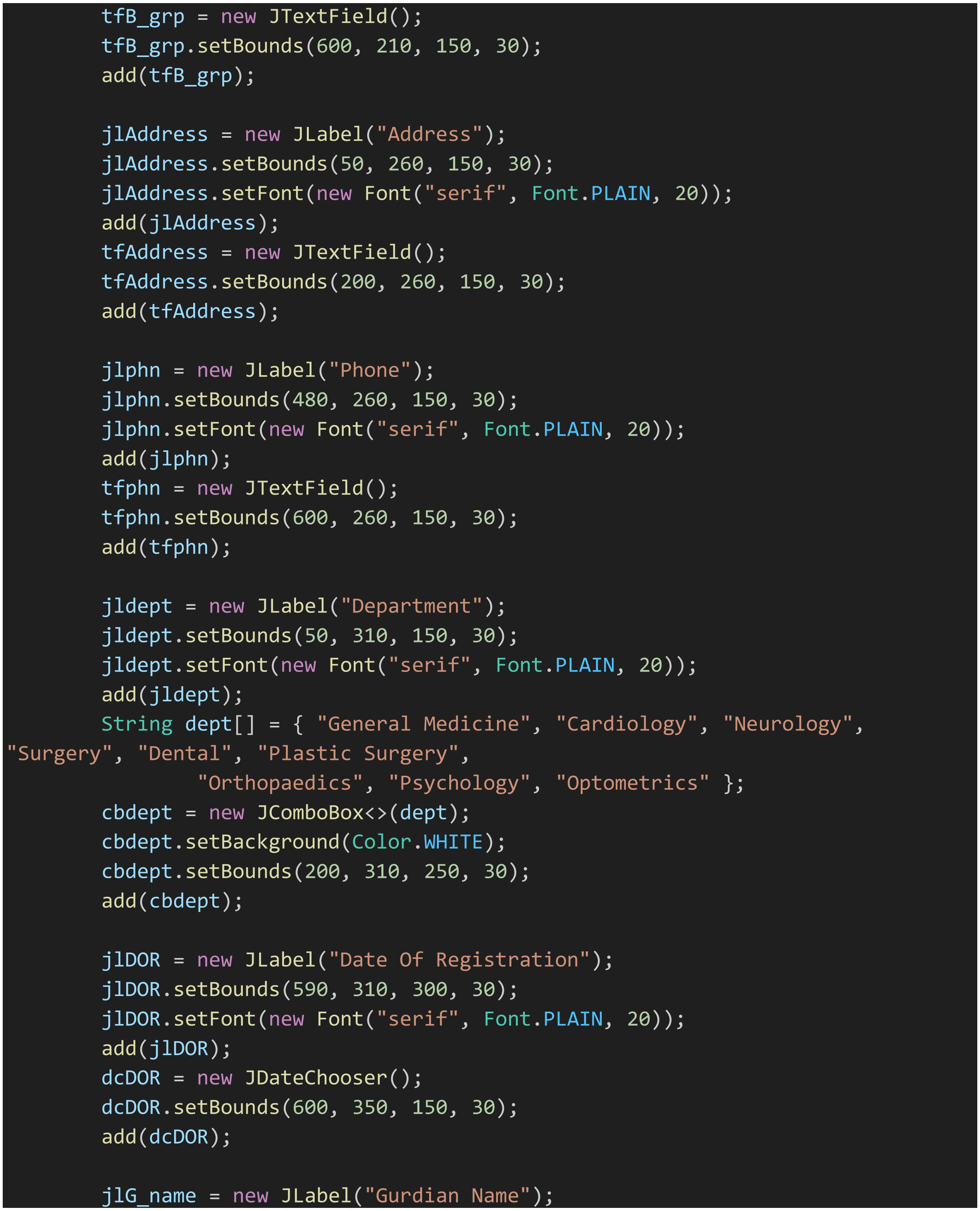
```
        heading.setForeground(Color.darkGray);
        image.add(heading);

        add = new JButton("New Patient");
        add.setBounds(650, 420, 150, 40);
        add.addActionListener(this);
        add.setBackground(Color.green);
        add.setForeground(Color.BLACK);
        image.add(add);

        view = new JButton("View Patients");
        view.setBounds(650, 480, 150, 40);
        view.addActionListener(this);
        view.setBackground(Color.green);
        view.setForeground(Color.BLACK);
        image.add(view);

        remove = new JButton("Remove Patients");
        remove.setBounds(850, 420, 150, 40);
        remove.addActionListener(this);
        remove.setBackground(Color.green);
        remove.setForeground(Color.BLACK);
        image.add(remove);

        developer = new JButton("Developer Details");
        developer.setBounds(850, 480, 150, 40);
        developer.addActionListener(this);
        developer.setBackground(Color.green);
        developer.setForeground(Color.BLACK);
        image.add(developer);

        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        setSize(1100, 630);
        setLocation(80, 30);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == add) {
            setVisible(false);
            new NewPatient();
        } else if (ae.getSource() == view) {
            setVisible(false);
            new ViewPatients();
        } else if (ae.getSource() == developer) {
            setVisible(false);
            new developer();
        } else {
            setVisible(false);
            new RemovePatient();
        }
    }

    public static void main(String[] args) {
        new Home();
    }
}
```

4.New Patient Registration Screen:

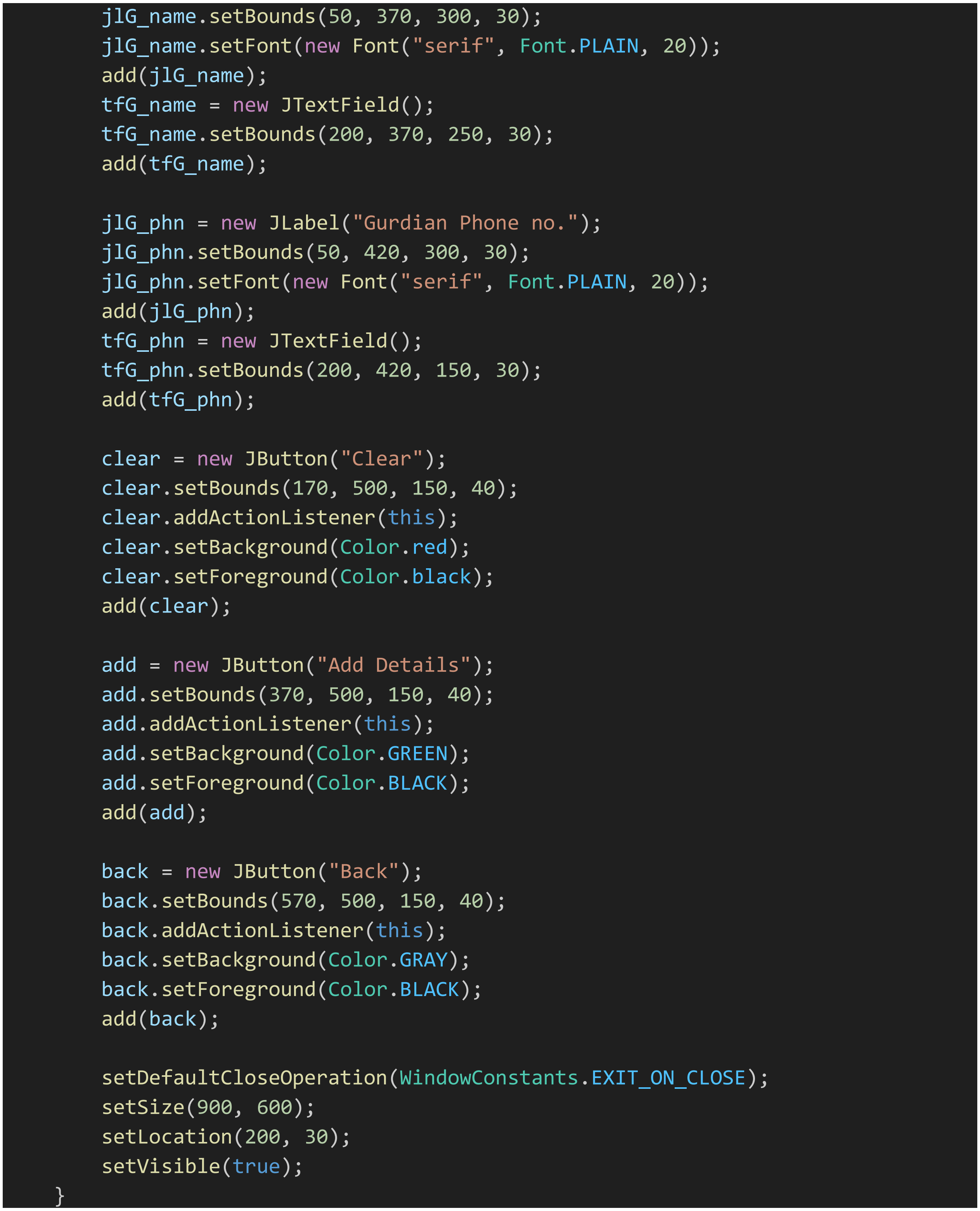
```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import com.toedter.calendar.JDateChooser;

public class NewPatient extends JFrame implements ActionListener {

    JLabel jlP_name, jlAadhar, jlAge, jlB_grp, jlphn, jlGender, jlG_name, jldept,
jlDOR, heading, jlAddress, jlG_phn;
    JTextField tfP_name, tfadhar, tfAge, tfB_grp, tfphn, tfG_name, tfG_phn,
tfAddress;
    JDateChooser dcDOR;
    JComboBox<String> cbdept, cbGender;
    JButton add, back, clear;

    public NewPatient() {
        getContentPane().setBackground(Color.cyan);
        setLayout(null);

        heading = new JLabel("Register New Patient");
        heading.setBounds(320, 10, 500, 50);
        heading.setFont(new Font("SAN_SERIF", Font.BOLD, 25));
```



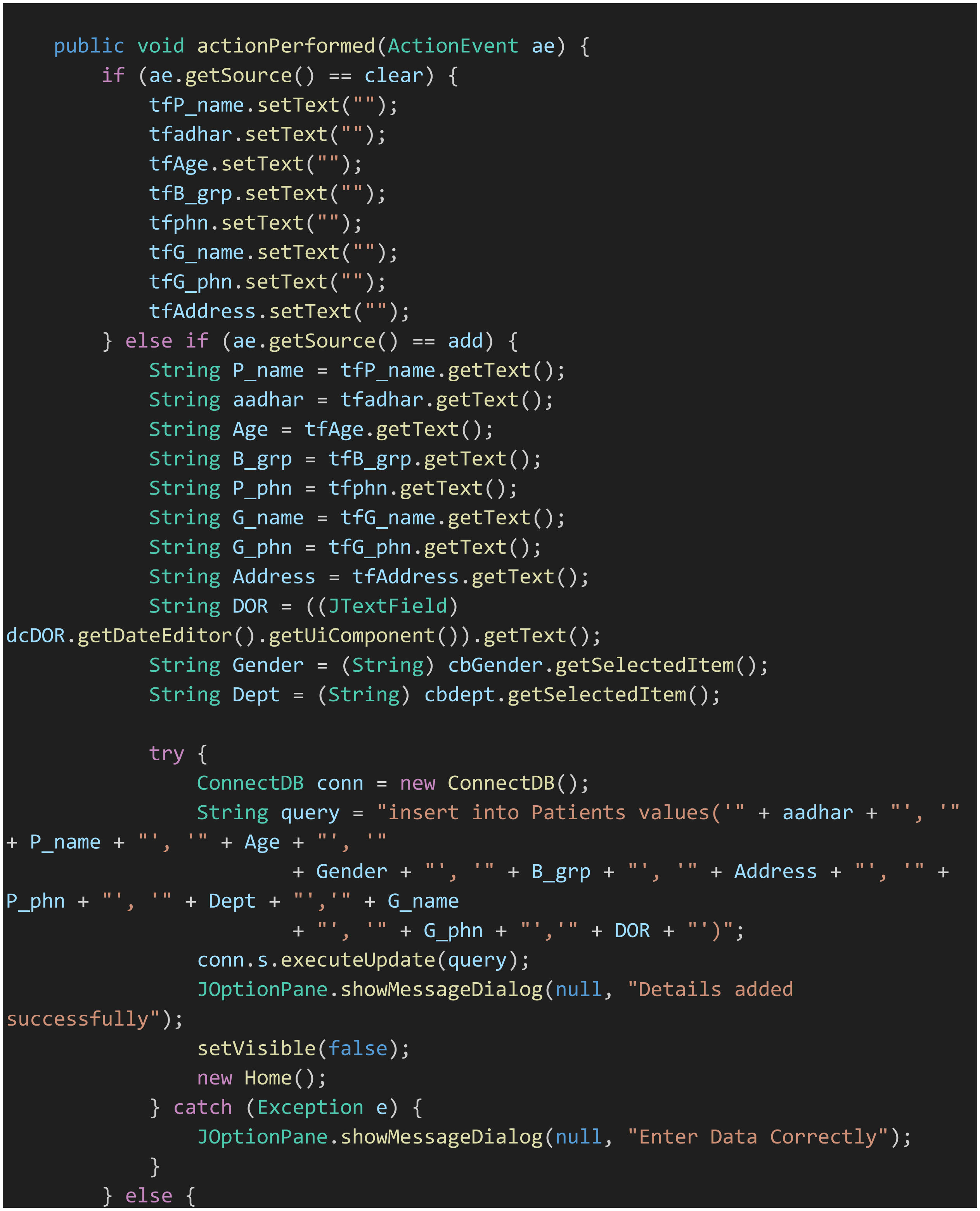
```
        add(heading);

        jlP_name = new JLabel("Patient Name");
        jlP_name.setBounds(50, 110, 150, 30);
        jlP_name.setFont(new Font("serif", Font.PLAIN, 20));
        heading.setForeground(Color.BLUE);
        add(jlP_name);
        tfP_name = new JTextField();
        tfP_name.setBounds(200, 110, 250, 30);
        add(tfP_name);

        jlAadhar = new JLabel(" Aadhar");
        jlAadhar.setBounds(46, 160, 150, 30);
        jlAadhar.setFont(new Font("serif", Font.PLAIN, 20));
        add(jlAadhar);
        tfadhar = new JTextField();
        tfadhar.setBounds(200, 160, 150, 30);
        add(tfadhar);

        jlAge = new JLabel("Age");
        jlAge.setBounds(480, 160, 150, 30);
        jlAge.setFont(new Font("serif", Font.PLAIN, 20));
        add(jlAge);
        tfAge = new JTextField();
        tfAge.setBounds(600, 160, 150, 30);
        add(tfAge);

        jlGender = new JLabel("Gender");
        jlGender.setBounds(50, 210, 150, 30);
        jlGender.setFont(new Font("serif", Font.PLAIN, 20));
        add(jlGender);
        String Gender[] = { "Male", "Female", "Others" };
        cbGender = new JComboBox<>(Gender);
        cbGender.setBackground(Color.WHITE);
        cbGender.setBounds(200, 210, 150, 30);
        add(cbGender);

        jlB_grp = new JLabel("Blood Group");
        jlB_grp.setBounds(480, 210, 150, 30);
        jlB_grp.setFont(new Font("serif", Font.PLAIN, 20));
        add(jlB_grp);
```



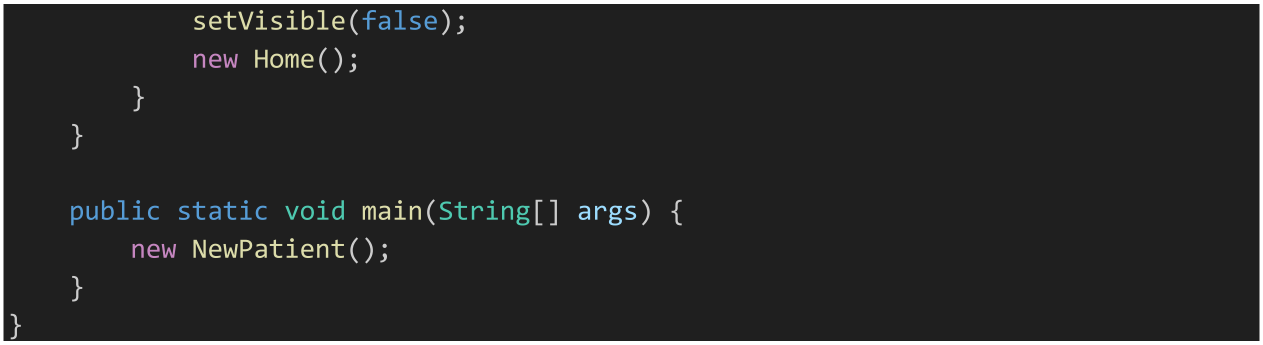
```
        tfB_grp = new JTextField();
        tfB_grp.setBounds(600, 210, 150, 30);
        add(tfB_grp);

        jlAddress = new JLabel("Address");
        jlAddress.setBounds(50, 260, 150, 30);
        jlAddress.setFont(new Font("serif", Font.PLAIN, 20));
        add(jlAddress);
        tfAddress = new JTextField();
        tfAddress.setBounds(200, 260, 150, 30);
        add(tfAddress);

        jlphn = new JLabel("Phone");
        jlphn.setBounds(480, 260, 150, 30);
        jlphn.setFont(new Font("serif", Font.PLAIN, 20));
        add(jlphn);
        tfphn = new JTextField();
        tfphn.setBounds(600, 260, 150, 30);
        add(tfphn);

        jldept = new JLabel("Department");
        jldept.setBounds(50, 310, 150, 30);
        jldept.setFont(new Font("serif", Font.PLAIN, 20));
        add(jldept);
        String dept[] = { "General Medicine", "Cardiology", "Neurology",
"Surgery", "Dental", "Plastic Surgery",
                "Orthopaedics", "Psychology", "Optometrics" };
        cbdept = new JComboBox<>(dept);
        cbdept.setBackground(Color.WHITE);
        cbdept.setBounds(200, 310, 250, 30);
        add(cbdept);

        jlDOR = new JLabel("Date Of Registration");
        jlDOR.setBounds(590, 310, 300, 30);
        jlDOR.setFont(new Font("serif", Font.PLAIN, 20));
        add(jlDOR);
        dcDOR = new JDateChooser();
        dcDOR.setBounds(600, 350, 150, 30);
        add(dcDOR);

        jlG_name = new JLabel("Gurdian Name");
```



```
        jlG_name.setBounds(50, 370, 300, 30);
        jlG_name.setFont(new Font("serif", Font.PLAIN, 20));
        add(jlG_name);
        tfG_name = new JTextField();
        tfG_name.setBounds(200, 370, 250, 30);
        add(tfG_name);

        jlG_phn = new JLabel("Gurdian Phone no.");
        jlG_phn.setBounds(50, 420, 300, 30);
        jlG_phn.setFont(new Font("serif", Font.PLAIN, 20));
        add(jlG_phn);
        tfG_phn = new JTextField();
        tfG_phn.setBounds(200, 420, 150, 30);
        add(tfG_phn);

        clear = new JButton("Clear");
        clear.setBounds(170, 500, 150, 40);
        clear.addActionListener(this);
        clear.setBackground(Color.red);
        clear.setForeground(Color.black);
        add(clear);

        add = new JButton("Add Details");
        add.setBounds(370, 500, 150, 40);
        add.addActionListener(this);
        add.setBackground(Color.GREEN);
        add.setForeground(Color.BLACK);
        add(add);

        back = new JButton("Back");
        back.setBounds(570, 500, 150, 40);
        back.addActionListener(this);
        back.setBackground(Color.GRAY);
        back.setForeground(Color.BLACK);
        add(back);

        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        setSize(900, 600);
        setLocation(200, 30);
        setVisible(true);
    }
```

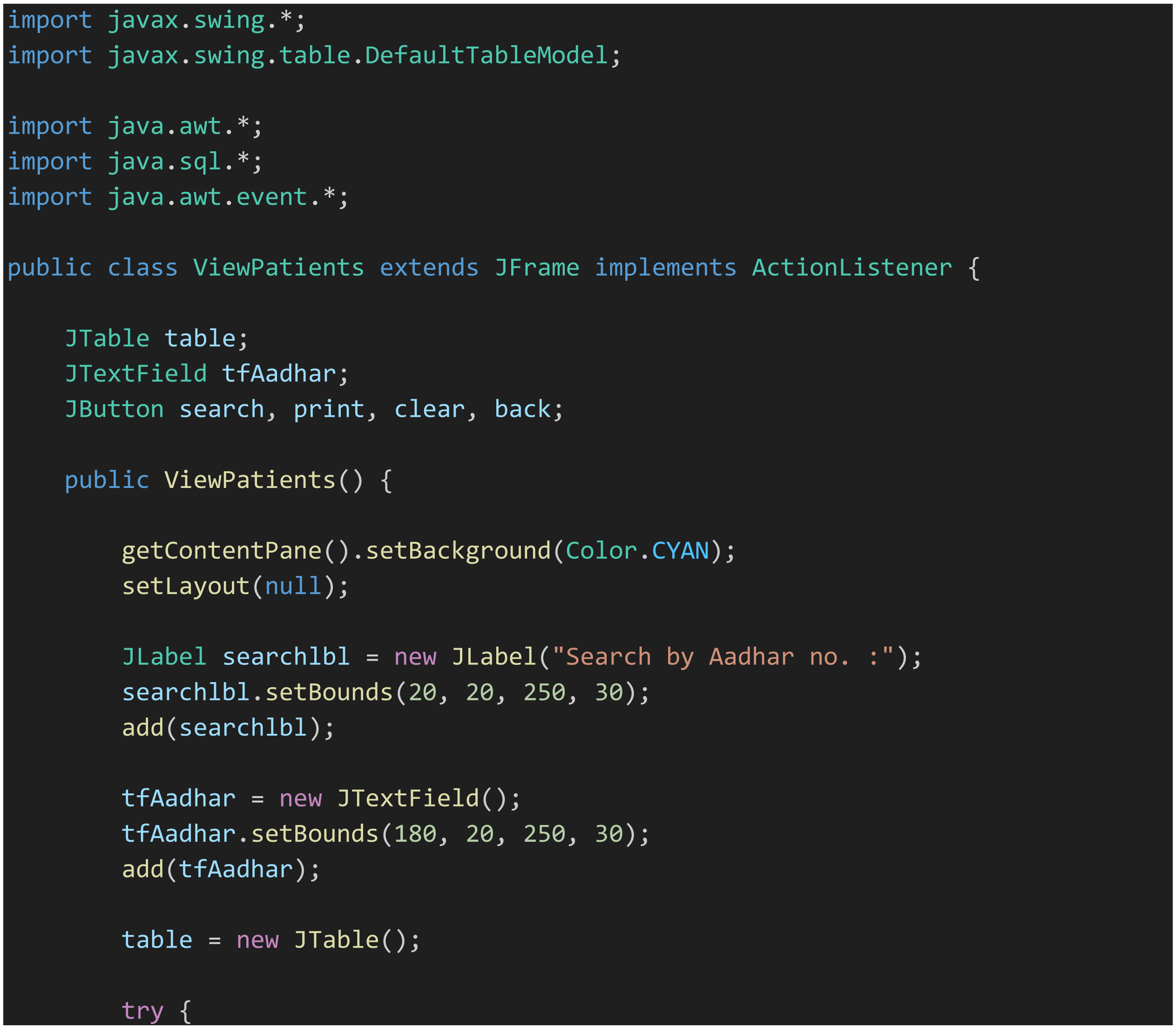


```
    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == clear) {
            tfP_name.setText("");
            tfadhar.setText("");
            tfAge.setText("");
            tfB_grp.setText("");
            tfphn.setText("");
            tfG_name.setText("");
            tfG_phn.setText("");
            tfAddress.setText("");
        } else if (ae.getSource() == add) {
            String P_name = tfP_name.getText();
            String aadhar = tfadhar.getText();
            String Age = tfAge.getText();
            String B_grp = tfB_grp.getText();
            String P_phn = tfphn.getText();
            String G_name = tfG_name.getText();
            String G_phn = tfG_phn.getText();
            String Address = tfAddress.getText();
            String DOR = ((JTextField)
dcDOR.getDateEditor().getUiComponent()).getText();
            String Gender = (String) cbGender.getSelectedItem();
            String Dept = (String) cbdept.getSelectedItem();

            try {
                ConnectDB conn = new ConnectDB();
                String query = "insert into Patients values('" + aadhar + "', '"
+ P_name + "', '" + Age + "', '"
                        + Gender + "', '" + B_grp + "', '" + Address + "', '" +
P_phn + "', '" + Dept + "','" + G_name
                        + "', '" + G_phn + "','" + DOR + "')";
                conn.s.executeUpdate(query);
                JOptionPane.showMessageDialog(null, "Details added
successfully");
                setVisible(false);
                new Home();
            } catch (Exception e) {
                JOptionPane.showMessageDialog(null, "Enter Data Correctly");
            }
        } else {
```

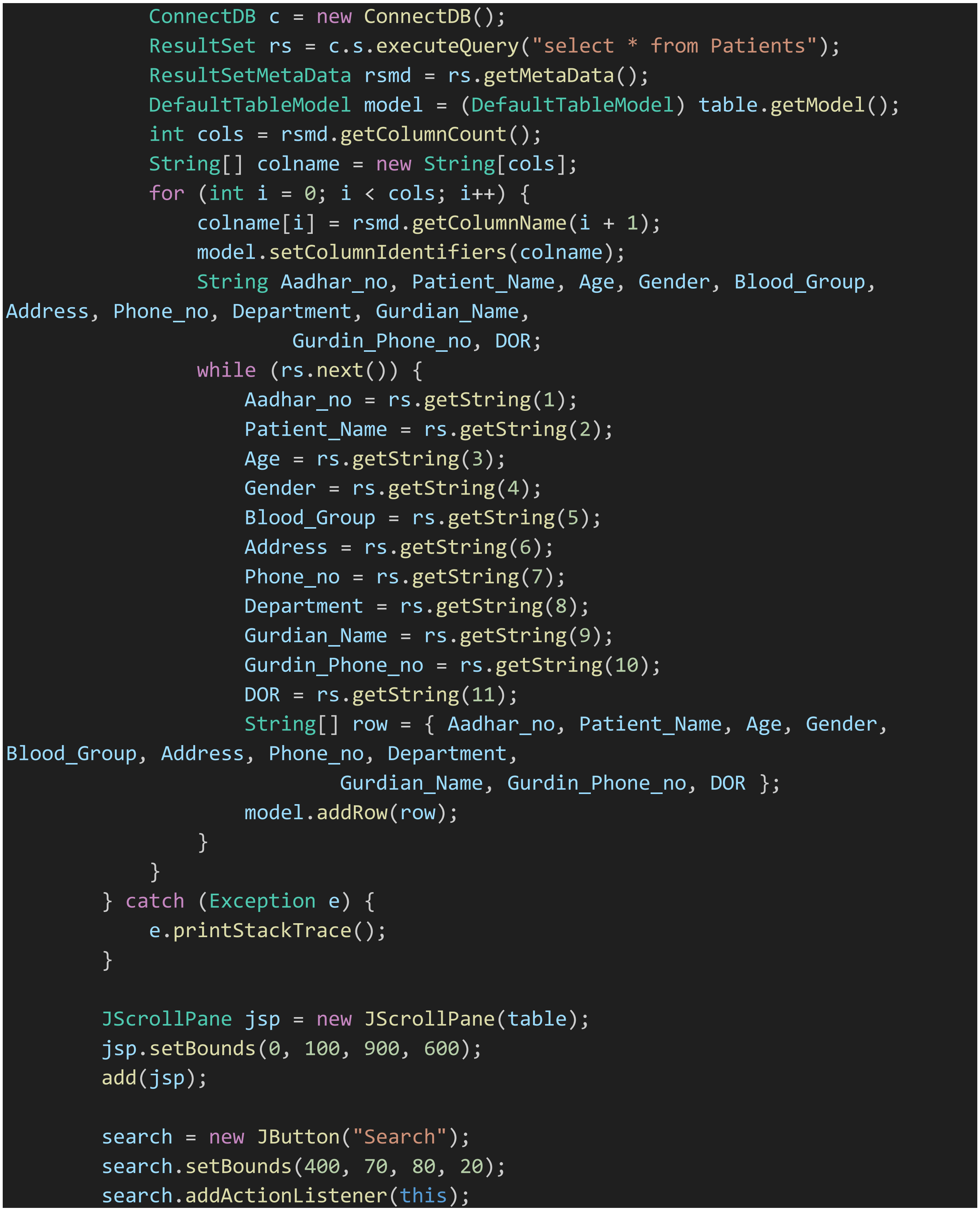


```
            setVisible(false);
            new Home();
        }
    }

    public static void main(String[] args) {
        new NewPatient();
    }
}
```

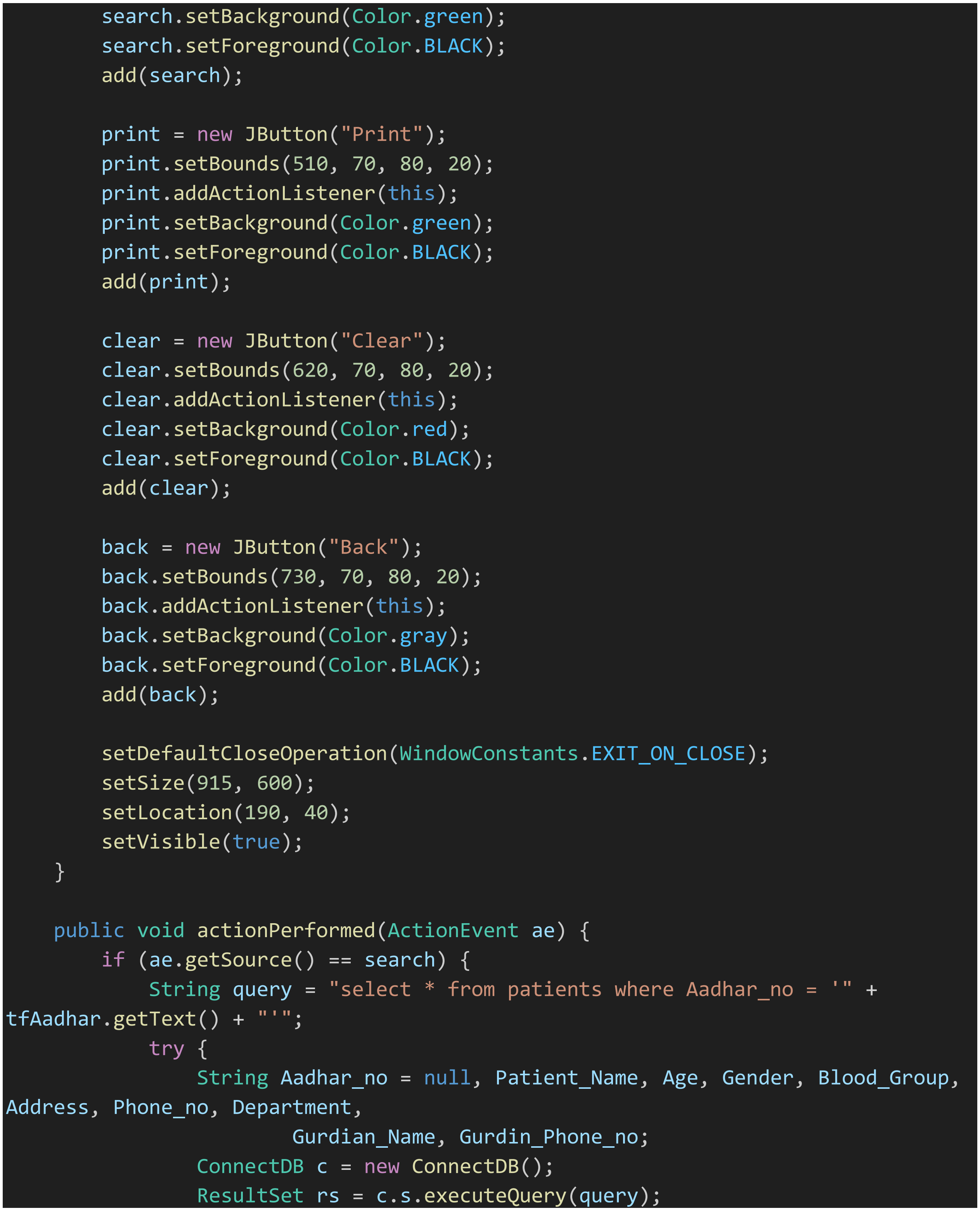
5.View Patient Screen:



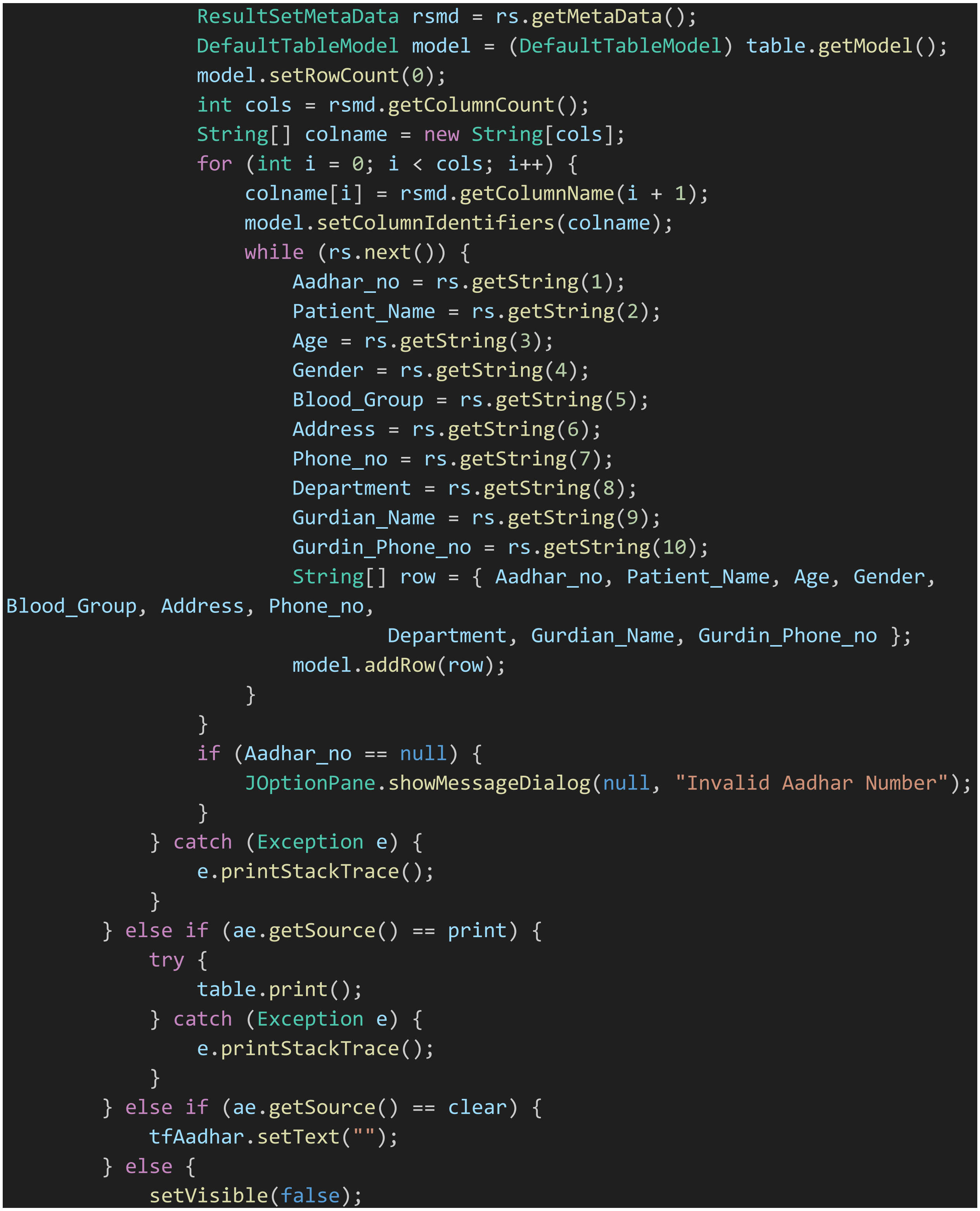
```
import javax.swing.*;
import javax.swing.table.DefaultTableModel;

import java.awt.*;
import java.sql.*;
import java.awt.event.*;

public class ViewPatients extends JFrame implements ActionListener {

    JTable table;
    JTextField tfAadhar;
    JButton search, print, clear, back;

    public ViewPatients() {

        getContentPane().setBackground(Color.CYAN);
        setLayout(null);

        JLabel searchlbl = new JLabel("Search by Aadhar no. :");
        searchlbl.setBounds(20, 20, 250, 30);
        add(searchlbl);

        tfAadhar = new JTextField();
        tfAadhar.setBounds(180, 20, 250, 30);
        add(tfAadhar);

        table = new JTable();

        try {
```



```
            ConnectDB c = new ConnectDB();
            ResultSet rs = c.s.executeQuery("select * from Patients");
            ResultSetMetaData rsmd = rs.getMetaData();
            DefaultTableModel model = (DefaultTableModel) table.getModel();
            int cols = rsmd.getColumnCount();
            String[] colname = new String[cols];
            for (int i = 0; i < cols; i++) {
                colname[i] = rsmd.getColumnName(i + 1);
                model.setColumnIdentifiers(colname);
                String Aadhar_no, Patient_Name, Age, Gender, Blood_Group,
Address, Phone_no, Department, Gurdian_Name,
                        Gurdin_Phone_no, DOR;
                while (rs.next()) {
                    Aadhar_no = rs.getString(1);
                    Patient_Name = rs.getString(2);
                    Age = rs.getString(3);
                    Gender = rs.getString(4);
                    Blood_Group = rs.getString(5);
                    Address = rs.getString(6);
                    Phone_no = rs.getString(7);
                    Department = rs.getString(8);
                    Gurdian_Name = rs.getString(9);
                    Gurdin_Phone_no = rs.getString(10);
                    DOR = rs.getString(11);
                    String[] row = { Aadhar_no, Patient_Name, Age, Gender,
Blood_Group, Address, Phone_no, Department,
                            Gurdian_Name, Gurdin_Phone_no, DOR };
                    model.addRow(row);
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }

        JScrollPane jsp = new JScrollPane(table);
        jsp.setBounds(0, 100, 900, 600);
        add(jsp);

        search = new JButton("Search");
        search.setBounds(400, 70, 80, 20);
        search.addActionListener(this);
```



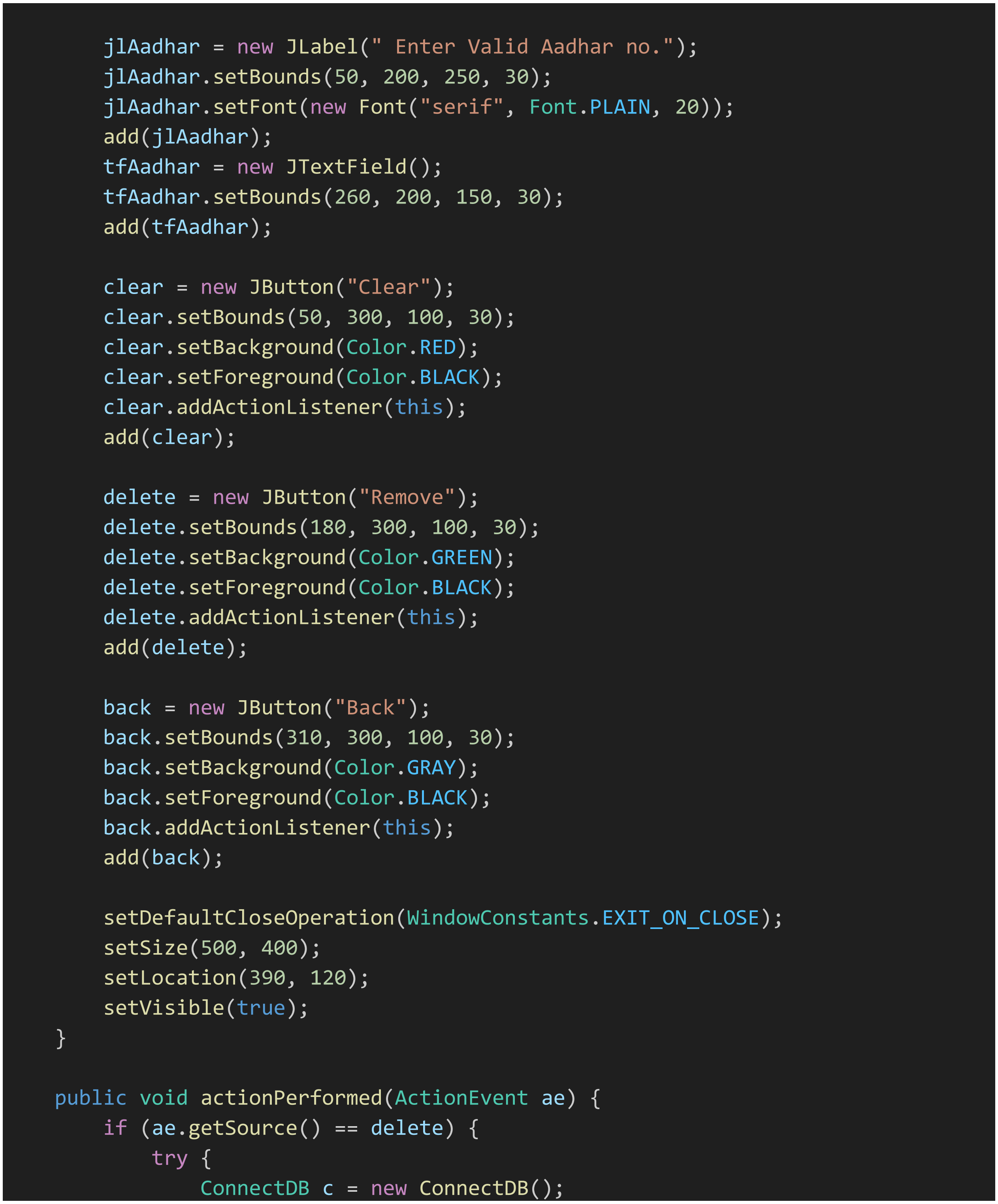
```
        search.setBackground(Color.green);
        search.setForeground(Color.BLACK);
        add(search);

        print = new JButton("Print");
        print.setBounds(510, 70, 80, 20);
        print.addActionListener(this);
        print.setBackground(Color.green);
        print.setForeground(Color.BLACK);
        add(print);

        clear = new JButton("Clear");
        clear.setBounds(620, 70, 80, 20);
        clear.addActionListener(this);
        clear.setBackground(Color.red);
        clear.setForeground(Color.BLACK);
        add(clear);

        back = new JButton("Back");
        back.setBounds(730, 70, 80, 20);
        back.addActionListener(this);
        back.setBackground(Color.gray);
        back.setForeground(Color.BLACK);
        add(back);

        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        setSize(915, 600);
        setLocation(190, 40);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == search) {
            String query = "select * from patients where Aadhar_no = '" +
tfAadhar.getText() + "'";
            try {
                String Aadhar_no = null, Patient_Name, Age, Gender, Blood_Group,
Address, Phone_no, Department,
                        Gurdian_Name, Gurdin_Phone_no;
                ConnectDB c = new ConnectDB();
                ResultSet rs = c.s.executeQuery(query);
```



```
                ResultSetMetaData rsmd = rs.getMetaData();
                DefaultTableModel model = (DefaultTableModel) table.getModel();
                model.setRowCount(0);
                int cols = rsmd.getColumnCount();
                String[] colname = new String[cols];
                for (int i = 0; i < cols; i++) {
                    colname[i] = rsmd.getColumnName(i + 1);
                    model.setColumnIdentifiers(colname);
                    while (rs.next()) {
                        Aadhar_no = rs.getString(1);
                        Patient_Name = rs.getString(2);
                        Age = rs.getString(3);
                        Gender = rs.getString(4);
                        Blood_Group = rs.getString(5);
                        Address = rs.getString(6);
                        Phone_no = rs.getString(7);
                        Department = rs.getString(8);
                        Gurdian_Name = rs.getString(9);
                        Gurdin_Phone_no = rs.getString(10);
                        String[] row = { Aadhar_no, Patient_Name, Age, Gender,
Blood_Group, Address, Phone_no,
                                Department, Gurdian_Name, Gurdin_Phone_no };
                        model.addRow(row);
                    }
                }
                if (Aadhar_no == null) {
                    JOptionPane.showMessageDialog(null, "Invalid Aadhar Number");
                }
            } catch (Exception e) {
                e.printStackTrace();
            }
        } else if (ae.getSource() == print) {
            try {
                table.print();
            } catch (Exception e) {
                e.printStackTrace();
            }
        } else if (ae.getSource() == clear) {
            tfAadhar.setText("");
        } else {
            setVisible(false);
            new Home();
        }
    }

    public static void main(String[] args) {
        new ViewPatients();
    }
}
```

6.Remove Patient Screen:

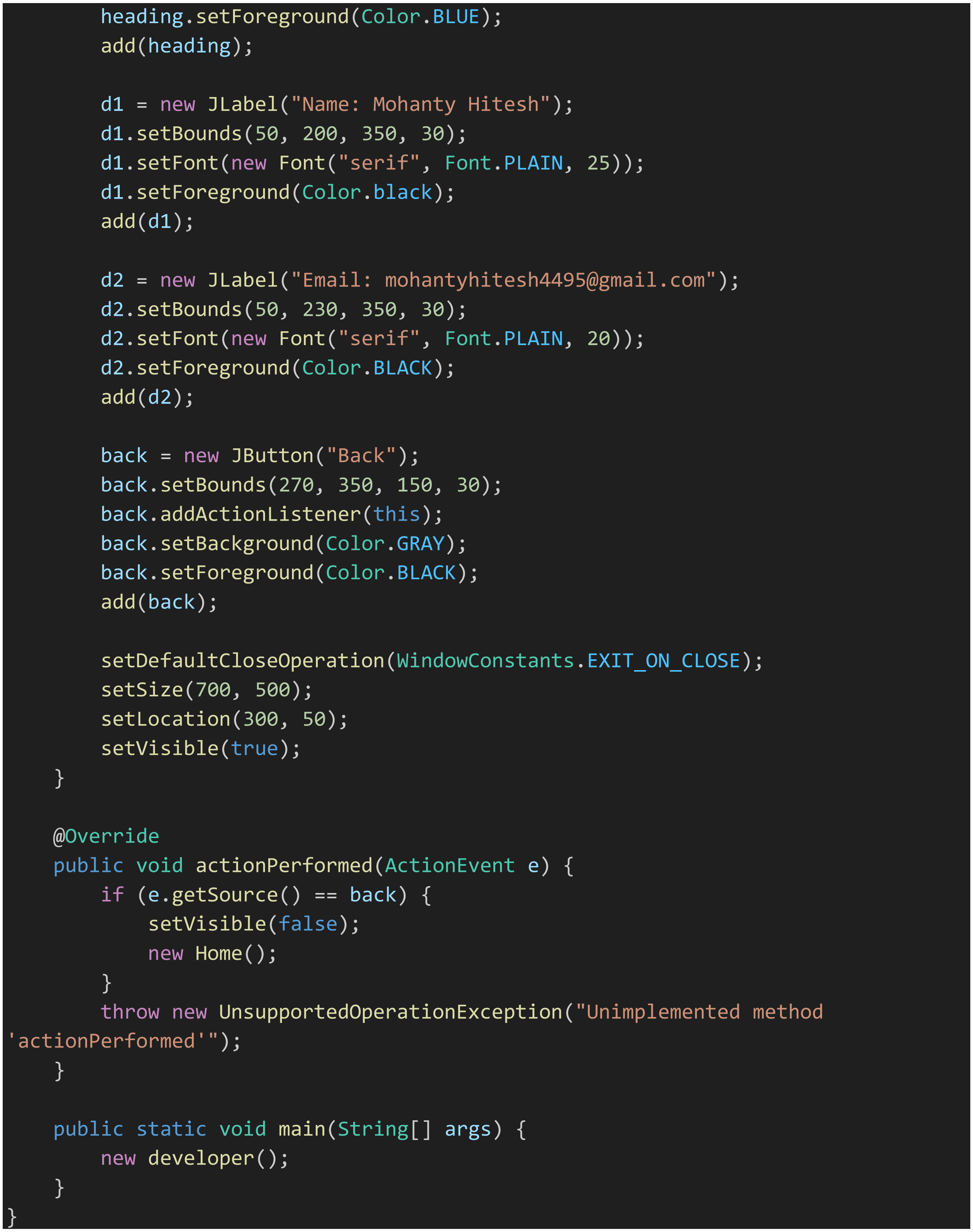
```
import javax.swing.*;

import java.awt.*;
import java.awt.event.*;

public class RemovePatient extends JFrame implements ActionListener {

    Choice cEmpId;
    JLabel jlAadhar, heading;
    JTextField tfAadhar;
    JButton delete, back, clear;

    public RemovePatient() {
        getContentPane().setBackground(Color.cyan);
        setLayout(null);

        heading = new JLabel("Remove Patient Details");
        heading.setBounds(100, 10, 350, 50);
        heading.setFont(new Font("SAN_SERIF", Font.BOLD, 25));
        heading.setForeground(Color.BLUE);
        add(heading);

        ImageIcon i1 = new
ImageIcon(ClassLoader.getSystemResource("Backgroundimage/remove.jpg"));
        Image i2 = i1.getImage().getScaledInstance(250, 150,
Image.SCALE_DEFAULT);
        ImageIcon i3 = new ImageIcon(i2);
        JLabel image = new JLabel(i3);
        image.setBounds(40, 80, 400, 100);
        add(image);
```



```
        jlAadhar = new JLabel(" Enter Valid Aadhar no.");
        jlAadhar.setBounds(50, 200, 250, 30);
        jlAadhar.setFont(new Font("serif", Font.PLAIN, 20));
        add(jlAadhar);
        tfAadhar = new JTextField();
        tfAadhar.setBounds(260, 200, 150, 30);
        add(tfAadhar);

        clear = new JButton("Clear");
        clear.setBounds(50, 300, 100, 30);
        clear.setBackground(Color.RED);
        clear.setForeground(Color.BLACK);
        clear.addActionListener(this);
        add(clear);

        delete = new JButton("Remove");
        delete.setBounds(180, 300, 100, 30);
        delete.setBackground(Color.GREEN);
        delete.setForeground(Color.BLACK);
        delete.addActionListener(this);
        add(delete);

        back = new JButton("Back");
        back.setBounds(310, 300, 100, 30);
        back.setBackground(Color.GRAY);
        back.setForeground(Color.BLACK);
        back.addActionListener(this);
        add(back);

        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        setSize(500, 400);
        setLocation(390, 120);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == delete) {
            try {
                ConnectDB c = new ConnectDB();
                String query = "delete from patients where Aadhar_no = '" +
tfAadhar.getText() + "'";
                c.s.executeUpdate(query);
                JOptionPane.showMessageDialog(null, "Patient Information
Removed");
                setVisible(false);
                new Home();
            } catch (Exception e) {
                JOptionPane.showMessageDialog(null, "Invalid Aadhar");
                tfAadhar.setText("");
            }
        } else if (ae.getSource() == clear) {
            tfAadhar.setText("");
        } else {
            setVisible(false);
            new Home();
        }
    }

    public static void main(String[] args) {
        new RemovePatient();
    }
}
```

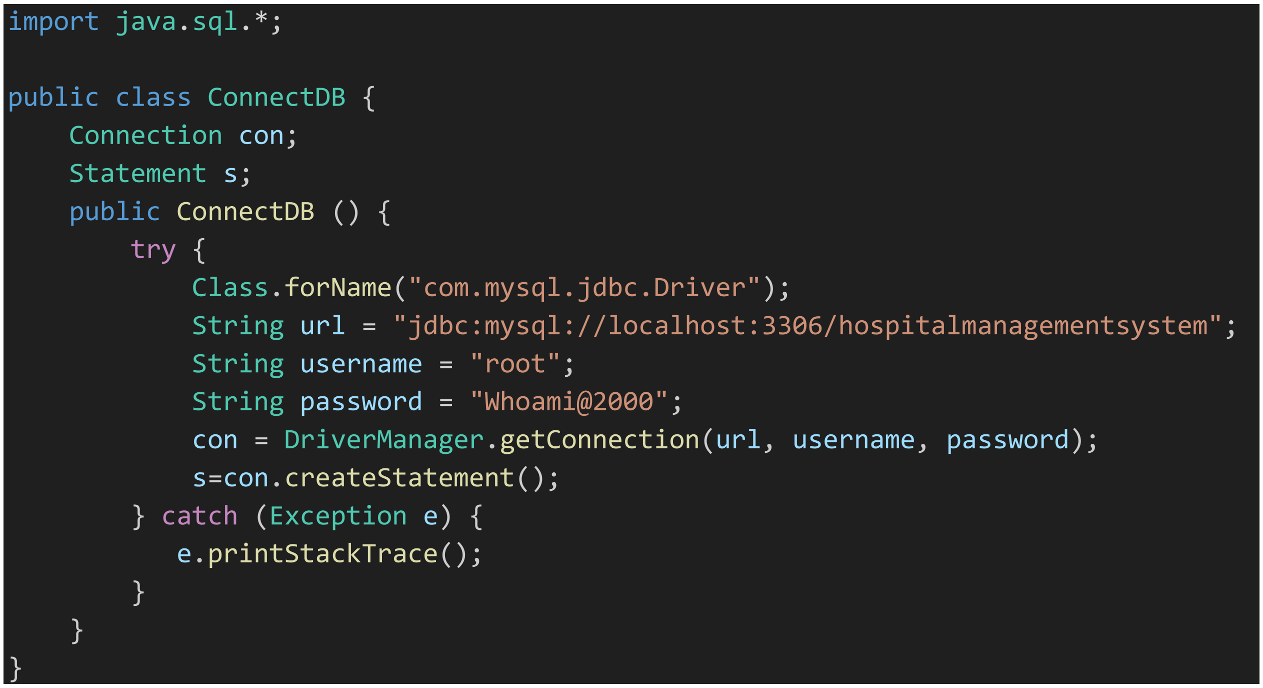
7.Developers Details Screen:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class developer extends JFrame implements ActionListener {
    JLabel heading, d1, d2, d3, d4, d5;
    JButton back;

    developer() {
        getContentPane().setBackground(Color.CYAN);
        setLayout(null);

        heading = new JLabel("Developer Details");
        heading.setBounds(150, 30, 450, 70);
        heading.setFont(new Font("SAN_SERIF", Font.BOLD, 47));
```



```
        heading.setForeground(Color.BLUE);
        add(heading);

        d1 = new JLabel("Name: Mohanty Hitesh");
        d1.setBounds(50, 200, 350, 30);
        d1.setFont(new Font("serif", Font.PLAIN, 25));
        d1.setForeground(Color.black);
        add(d1);

        d2 = new JLabel("Email: mohantyhitesh4495@gmail.com");
        d2.setBounds(50, 230, 350, 30);
        d2.setFont(new Font("serif", Font.PLAIN, 20));
        d2.setForeground(Color.BLACK);
        add(d2);

        back = new JButton("Back");
        back.setBounds(270, 350, 150, 30);
        back.addActionListener(this);
        back.setBackground(Color.GRAY);
        back.setForeground(Color.BLACK);
        add(back);

        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        setSize(700, 500);
        setLocation(300, 50);
        setVisible(true);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == back) {
            setVisible(false);
            new Home();
        }
        throw new UnsupportedOperationException("Unimplemented method
'actionPerformed'");
    }

    public static void main(String[] args) {
        new developer();
    }
}
```

8.DataBase Connection Code:



```
import java.sql.*;

public class ConnectDB {
    Connection con;
    Statement s;
    public ConnectDB () {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            String url = "jdbc:mysql://localhost:3306/hospitalmanagementsystem";
            String username = "root";
            String password = "Whoami@2000";
            con = DriverManager.getConnection(url, username, password);
            s=con.createStatement();
        } catch (Exception e) {
           e.printStackTrace();
        }
    }
}
```

Result Analysis

1. Output Snapshot of Welcome Screen:

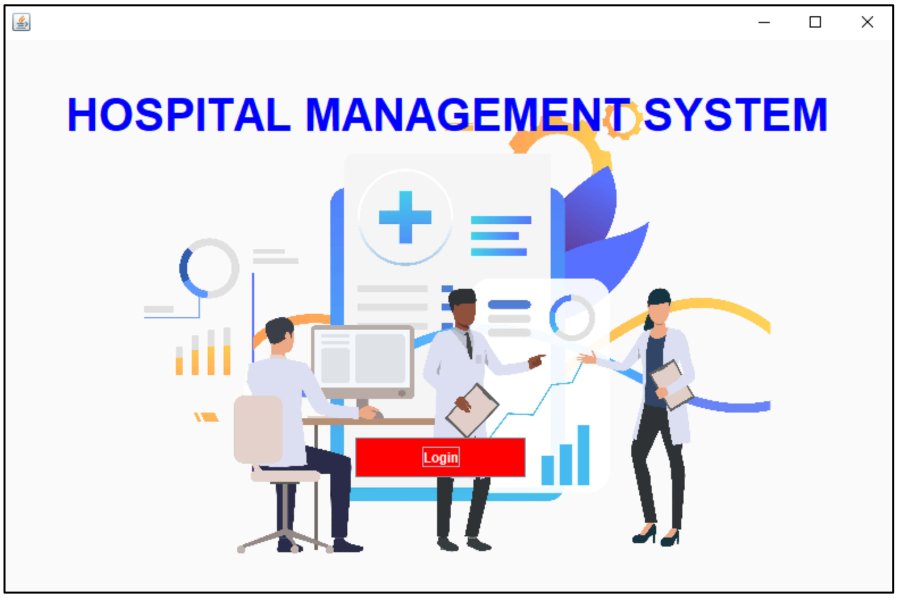
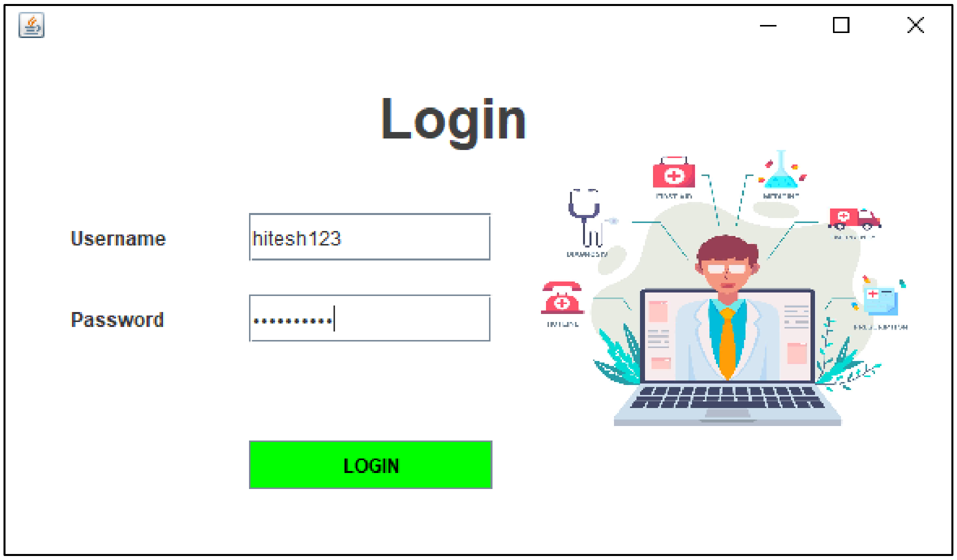


Figure 1 Welcome Page

2. Output Snapshot of Login Screen:



Username hitesh123

Password ••••••••••

Figure 2. Login Page

3. Output Snapshot for Home screen:

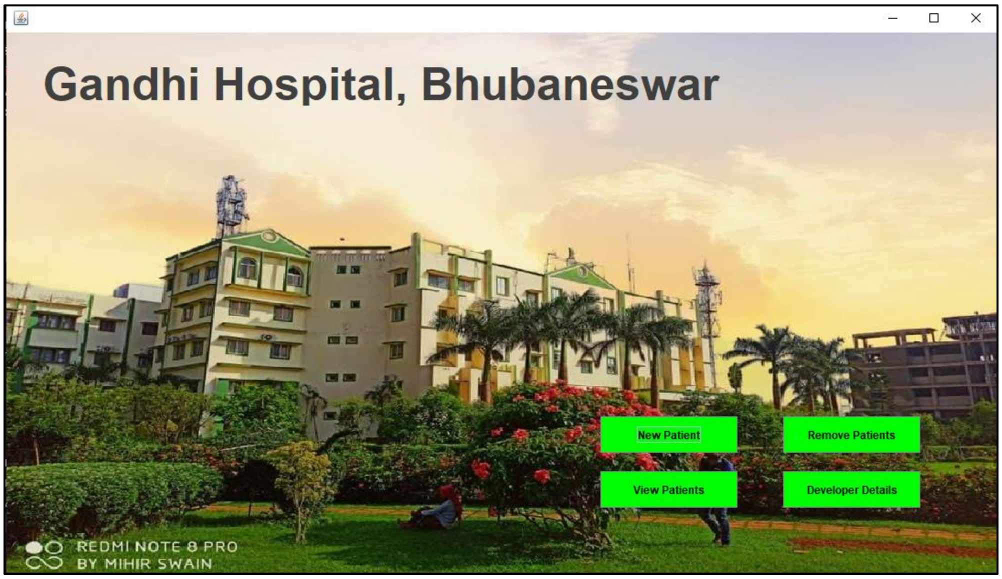
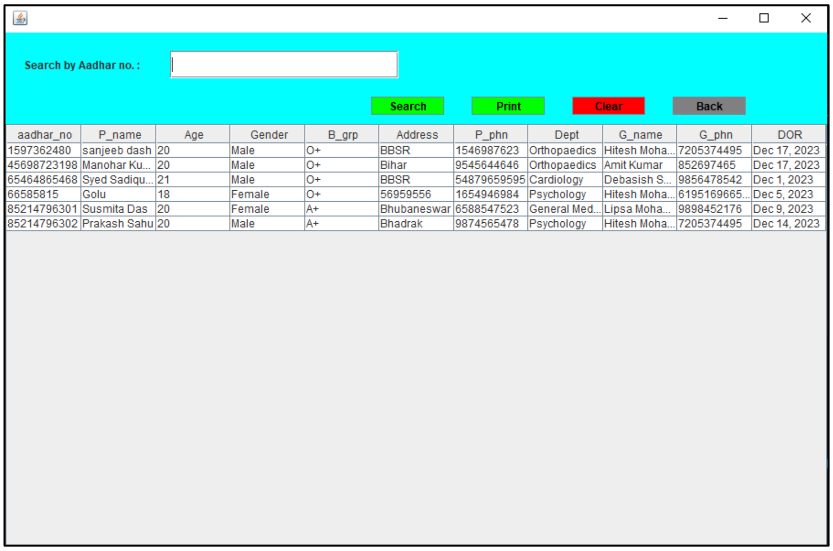


Figure 3. Home Screen

4. Output Snapshot for View Patient details screen:

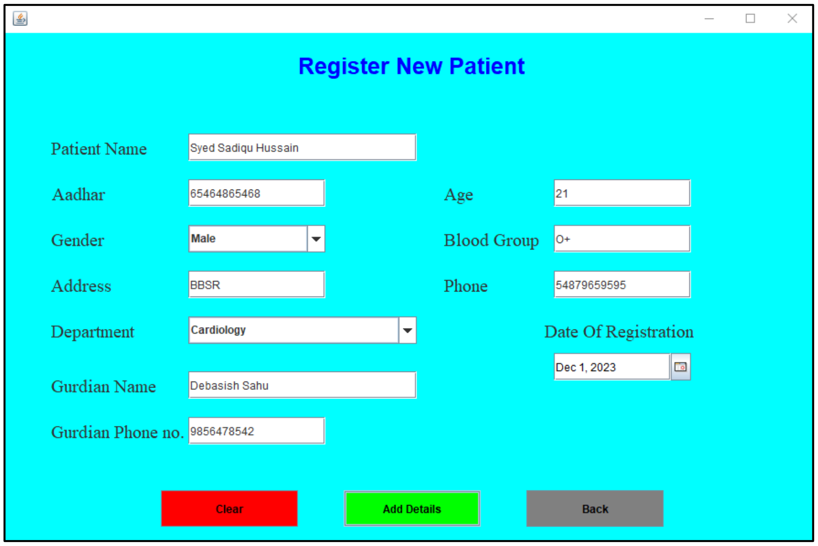


Search by Aadhar no. :

aadhar_no	P_name	Age	Gender	B_grp	Address	P_phn	Dept	G_name	G_phn	DOR
1597362480	sanjeeb dash	20	Male	O+	BBSR	1546987623	Orthopaedics	Hitesh Moha...	7205374495	Dec 17, 2023
45698723198	Manohar Ku...	20	Male	O+	Bihar	9545644646	Orthopaedics	Amit Kumar	852697465	Dec 17, 2023
65464865468	Syed Sadiqu...	21	Male	O+	BBSR	54879659595	Cardiology	Debasish S...	9856478542	Dec 1, 2023
66585815	Golu	18	Female	O+	56959556	1654946984	Psychology	Hitesh Moha...	6195169665...	Dec 5, 2023
85214796301	Susmita Das	20	Female	A+	Bhubaneswar	6588547523	General Med...	Lipsa Moha...	9898452176	Dec 9, 2023
85214796302	Prakash Sahu	20	Male	A+	Bhadrak	9874565478	Psychology	Hitesh Moha...	7205374495	Dec 14, 2023

Figure 4. Patients Details screen

5. Output Snapshot of Adding New Patient's Details:



Register New Patient

Patient Name Syed Sadiqu Hussain

Aadhar 65464865468

Age 21

Gender Male

Blood Group O+

Address BBSR

Phone 54879659595

Department Cardiology

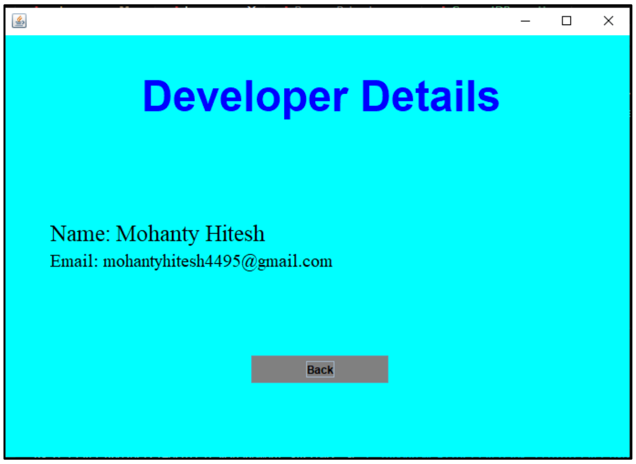
Date Of Registration Dec 1, 2023

Gurdian Name Debasish Sahu

Gurdian Phone no. 9856478542

Figure 7. Adding New Patient

6. Output Snapshot of Developers details screen:



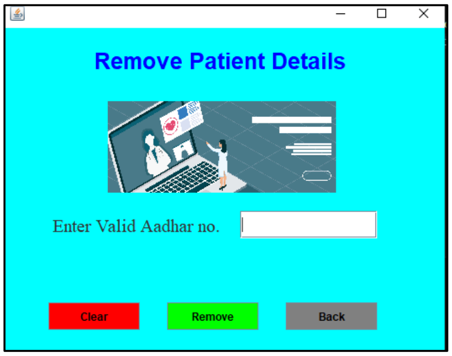
Developer Details

Name: Mohanty Hitesh

Email: mohantyhitesh4495@gmail.com

Figure 5. Developer's details

7. Output Snapshot of Remove Patient details:



Remove Patient Details

Enter Valid Aadhar no.

Figure 6. Delete Patient

Future Scope

The Hospital Management System project has great potential for future enhancements and expansions. Some of the possible future scope for this project includes: 1. Mobile App Integration: Developing a mobile app version of the system will allow hospital staff to access patient records, appointments, and other hospital-related information remotely from their smartphones, improving accessibility and convenience. 2. AI Integration: Integrating artificial intelligence (AI) technologies can enable the system to predict patient outcomes, identify disease patterns, and provide insights for effective decision-making. 3. Telemedicine Integration: Integrating telemedicine technologies will enable healthcare providers to offer remote healthcare services, such as video consultations, remote monitoring, and online prescription. 4. Analytics and Reporting: Adding analytics and reporting functionalities can help hospital administrators monitor the system's performance, identify trends, and make informed decisions. Page | 26 5. Integration with IoT devices: Integration with Internet of Things (IoT) devices, such as wearables, can enable healthcare providers to collect real-time patient data and improve patient monitoring. In conclusion, the Hospital Management System project has the potential for further advancements and expansion, making it an essential tool for healthcare providers in managing hospital operations and providing high-quality patient care.

Conclusion

In conclusion, the Hospital Management System project is a comprehensive software application designed to automate various administrative tasks involved in managing a hospital's daily operations. The project's objective is to improve the quality of healthcare services by streamlining hospital management processes, reducing errors, and increasing operational efficiency. The project will be developed using Java programming language, and it will integrate various modules such as patient registration, doctor appointment scheduling, electronic medical records, pharmacy management, and billing. The project is technically, economically, and operationally feasible, and it aims to provide healthcare providers with a reliable and efficient platform to manage their daily operations. Overall, the Hospital Management System project will be a valuable tool for healthcare providers, helping them provide better patient care, reduce errors, and increase operational efficiency.

Reference

1. Awojide, Simon, I. M. Omogbhemhe, O. S. Awe, and T. S. Babatope, "Towards the digitalization of Hospital Management System of Appolo Hospital Private Hospital: The Design Perspective. A Study of Samuel Medical Collage," Int. J. Sci. Res. Publ., vol. 8, no. 5, pp. 46–54, 2018.

2. O. I. Mike and A. Simon, "Towards the Digitalization of Appolo Hospital : The Design Perspective," vol. 8, no. 2, pp. 1175–1178, 2017.

3. Adithya. R., A. Singh, S. Pathan, and V. Kanade, "Hospital Management System," Int. J. Comput. Appl., vol. 180, no. 6, pp. 22–24, 2017.

4. Varsha Chavan, Priya Jadhav, Snehal Korade, Priyanka Teli, Implementing Customizable Online Food Ordering System Using Web Based Application", International Journal of Innovative Science, Engineering Technology (IJISET) 2015.

5. Patel, Mayurkumar, " of Hospital Management System " (2015). Technical Library. Paper 219