Name: Javeria Rahim BCS-IIB 2023-BCS-046

Submitted to: Sir Adeel Khalid

**Project: E-HOSPITAL MANAMGEMENT SYSTEM**

**Source code:**

**Login form:**

package com.hospital;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

public class LoginForm extends JFrame {

private static final String DB\_URL = "jdbc:oracle:thin:@localhost:1521:XE";

private static final String USERNAME = "ehms";

private static final String PASSWORD = "epass";

private JTextField emailField;

private JPasswordField passwordField;

private JButton loginButton, signupButton;

public LoginForm() {

setTitle("E-Hospital Management System - Login");

setSize(400, 300);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLocationRelativeTo(null);

setLayout(new BorderLayout(10, 10));

// Title

JLabel titleLabel = new JLabel("E-Hospital Management System", JLabel.CENTER);

titleLabel.setFont(new Font("Serif", Font.BOLD, 16));

add(titleLabel, BorderLayout.NORTH);

// Center Panel for form inputs

JPanel centerPanel = new JPanel(new GridBagLayout());

GridBagConstraints gbc = new GridBagConstraints();

gbc.insets = new Insets(10, 10, 10, 10);

gbc.fill = GridBagConstraints.HORIZONTAL;

// Email Label

gbc.gridx = 0;

gbc.gridy = 0;

JLabel emailLabel = new JLabel("Email:");

centerPanel.add(emailLabel, gbc);

// Email Field

gbc.gridx = 1;

emailField = new JTextField(20);

centerPanel.add(emailField, gbc);

// Password Label

gbc.gridx = 0;

gbc.gridy = 1;

JLabel passLabel = new JLabel("Password:");

centerPanel.add(passLabel, gbc);

// Password Field

gbc.gridx = 1;

passwordField = new JPasswordField(20);

centerPanel.add(passwordField, gbc);

// Buttons Panel

JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER, 10, 10));

// Login Button

loginButton = new JButton("Login");

buttonPanel.add(loginButton);

// Signup Button

signupButton = new JButton("Signup");

buttonPanel.add(signupButton);

// Adding Panels to Frame

add(centerPanel, BorderLayout.CENTER);

add(buttonPanel, BorderLayout.SOUTH);

// Action Listeners

loginButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

login();

}

});

signupButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

new SignupForm().setVisible(true);

dispose();

}

});

}

private void login() {

try {

Connection conn = DriverManager.getConnection(DB\_URL, USERNAME, PASSWORD);

String sql = "SELECT \* FROM users WHERE email = ? AND password = ?";

PreparedStatement pst = conn.prepareStatement(sql);

pst.setString(1, emailField.getText());

pst.setString(2, new String(passwordField.getPassword()));

ResultSet rs = pst.executeQuery();

if (rs.next()) {

String role = rs.getString("role");

if ("patient".equals(role)) {

int patientId = rs.getInt("patient\_id");

showPatientDashboard(patientId);

} else if ("doctor".equals(role)) {

showDoctorForm();

} else {

JOptionPane.showMessageDialog(null, "Invalid role assigned to user.");

}

dispose();

} else {

JOptionPane.showMessageDialog(null, "Invalid email or password");

}

conn.close();

} catch (SQLException ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(null, "Error: " + ex.getMessage());

}

}

private void showPatientDashboard(int patientId) {

SwingUtilities.invokeLater(() -> new PatientDashboard(patientId).setVisible(true));

}

private void showDoctorForm() {

SwingUtilities.invokeLater(() -> new DoctorForm().setVisible(true));

}

public static void main(String[] args) {

SwingUtilities.invokeLater(() -> new LoginForm().setVisible(true));

}

}

**Signup Form:**

package com.hospital;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.SQLException;

public class SignupForm extends JFrame {

private static final String DB\_URL = "jdbc:oracle:thin:@localhost:1521:XE"; // Replace with your Oracle DB URL

private static final String USERNAME = "ehms"; // Replace with your Oracle DB username

private static final String PASSWORD = "epass"; // Replace with your Oracle DB password

private JTextField firstNameField;

private JTextField lastNameField;

private JTextField emailField;

private JPasswordField passwordField;

private JTextField contactNumberField;

private JTextField addressField;

public SignupForm() {

super("Signup Form");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setSize(400, 400);

setLocationRelativeTo(null); // Center the window

setLayout(new BorderLayout(10, 10));

// Title

JLabel titleLabel = new JLabel("E-Hospital Management System", JLabel.CENTER);

titleLabel.setFont(new Font("Serif", Font.BOLD, 16));

add(titleLabel, BorderLayout.NORTH);

// Form Panel

JPanel formPanel = new JPanel(new GridBagLayout());

// Add First Name Label and Field

GridBagConstraints gbc = new GridBagConstraints();

gbc.insets = new Insets(10, 10, 10, 10);

gbc.anchor = GridBagConstraints.WEST;

gbc.gridx = 0;

gbc.gridy = 0;

formPanel.add(new JLabel("First Name:"), gbc);

firstNameField = new JTextField(15);

gbc.gridx = 1;

formPanel.add(firstNameField, gbc);

// Add Last Name Label and Field

gbc.gridx = 0;

gbc.gridy = 1;

formPanel.add(new JLabel("Last Name:"), gbc);

lastNameField = new JTextField(15);

gbc.gridx = 1;

formPanel.add(lastNameField, gbc);

// Add Email Label and Field

gbc.gridx = 0;

gbc.gridy = 2;

formPanel.add(new JLabel("Email:"), gbc);

emailField = new JTextField(15);

gbc.gridx = 1;

formPanel.add(emailField, gbc);

// Add Password Label and Field

gbc.gridx = 0;

gbc.gridy = 3;

formPanel.add(new JLabel("Password:"), gbc);

passwordField = new JPasswordField(15);

gbc.gridx = 1;

formPanel.add(passwordField, gbc);

// Add Contact Number Label and Field

gbc.gridx = 0;

gbc.gridy = 4;

formPanel.add(new JLabel("Contact Number:"), gbc);

contactNumberField = new JTextField(15);

gbc.gridx = 1;

formPanel.add(contactNumberField, gbc);

// Add Address Label and Field

gbc.gridx = 0;

gbc.gridy = 5;

formPanel.add(new JLabel("Address:"), gbc);

addressField = new JTextField(15);

gbc.gridx = 1;

formPanel.add(addressField, gbc);

// Button Panel

JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.RIGHT));

JButton btnSignup = new JButton("Signup");

btnSignup.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

signup(); // Call signup method when button is clicked

}

});

buttonPanel.add(btnSignup);

add(formPanel, BorderLayout.CENTER);

add(buttonPanel, BorderLayout.SOUTH);

setVisible(true);

}

private void signup() {

String firstName = firstNameField.getText().trim();

String lastName = lastNameField.getText().trim();

String email = emailField.getText().trim();

String password = new String(passwordField.getPassword()); // Get plain password from field

String contactNumber = contactNumberField.getText().trim();

String address = addressField.getText().trim();

// Database connection and SQL statement

try (Connection conn = DriverManager.getConnection(DB\_URL, USERNAME, PASSWORD)) {

// Use a PreparedStatement to safely insert data

String sql = "INSERT INTO patients (first\_name, last\_name, email, password, contact\_number, address) " +

"VALUES (?, ?, ?, ?, ?, ?)";

try (PreparedStatement pst = conn.prepareStatement(sql)) {

// Set parameters for the PreparedStatement

pst.setString(1, firstName);

pst.setString(2, lastName);

pst.setString(3, email);

pst.setString(4, password); // Insert plain password

pst.setString(5, contactNumber);

pst.setString(6, address);

// Execute the insert statement

int rowsInserted = pst.executeUpdate();

if (rowsInserted > 0) {

JOptionPane.showMessageDialog(null, "Signup successful! Patient added.");

clearFields();

} else {

JOptionPane.showMessageDialog(null, "Signup failed. Please try again.");

}

}

} catch (SQLException ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(null, "Error: " + ex.getMessage());

}

}

private void clearFields() {

firstNameField.setText("");

lastNameField.setText("");

emailField.setText("");

passwordField.setText("");

contactNumberField.setText("");

addressField.setText("");

}

public static void main(String[] args) {

SwingUtilities.invokeLater(new Runnable() {

public void run() {

new SignupForm();

}

});

}

}

**Database Connection:**

package com.hospital;

import java.sql.Connection;

import java.sql.DriverManager;

public class DatabaseConnection {

private static final String URL = "jdbc:oracle:thin:@localhost:1521:xe";

private static final String USERNAME = "ehms";

private static final String PASSWORD = "epass";

public static Connection getConnection() throws Exception {

Class.forName("oracle.jdbc.driver.OracleDriver");

return DriverManager.getConnection(URL, USERNAME, PASSWORD);

}

}

**Patient Dashboard Form:**

package com.hospital;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

public class PatientDashboard extends JFrame {

private static final String DB\_URL = "jdbc:oracle:thin:@localhost:1521:XE";

private static final String USERNAME = "ehms";

private static final String PASSWORD = "epass";

private int patientId;

private JLabel nameLabel;

public PatientDashboard(int patientId) {

setForeground(new Color(128, 0, 0));

setBackground(new Color(245, 245, 245));

this.patientId = patientId;

initializeComponents();

fetchPatientName();

}

private void initializeComponents() {

setTitle("Patient Dashboard");

setSize(400, 400);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLocationRelativeTo(null); // Center the window

getContentPane().setLayout(new BorderLayout(10, 10));

// Title

JLabel titleLabel = new JLabel("E-Hospital Management System", JLabel.CENTER);

titleLabel.setForeground(new Color(47, 79, 79));

titleLabel.setFont(new Font("Serif", Font.BOLD, 16));

getContentPane().add(titleLabel, BorderLayout.NORTH);

// Name Label

nameLabel = new JLabel("Welcome, ");

nameLabel.setForeground(new Color(128, 0, 0));

nameLabel.setBackground(new Color(245, 245, 245));

nameLabel.setFont(new Font("Serif", Font.ITALIC, 14));

getContentPane().add(nameLabel, BorderLayout.CENTER);

// Buttons Panel

JPanel panelButtons = new JPanel(new GridBagLayout());

panelButtons.setBackground(new Color(245, 245, 245));

panelButtons.setForeground(new Color(128, 0, 0));

GridBagConstraints gbc = new GridBagConstraints();

gbc.insets = new Insets(10, 10, 10, 10);

gbc.fill = GridBagConstraints.HORIZONTAL;

gbc.anchor = GridBagConstraints.CENTER;

JButton btnFixAppointment = createButton("Fix Appointment", e -> openAppointmentFixForm(), gbc, 0);

JButton btnBilling = createButton("Manage Billing", e -> openBillingForm(), gbc, 1);

JButton btnUpdateDetails = createButton("Update Details", e -> openUpdateForm(), gbc, 2);

JButton btnDeleteAccount = createButton("Delete Account", e -> deleteAccount(), gbc, 3);

JButton btnViewDetails = createButton("View Details", e -> viewPatientDetails(), gbc, 4);

addButtonToPanel(panelButtons, btnFixAppointment, gbc, 0);

addButtonToPanel(panelButtons, btnBilling, gbc, 1);

addButtonToPanel(panelButtons, btnUpdateDetails, gbc, 2);

addButtonToPanel(panelButtons, btnDeleteAccount, gbc, 3);

addButtonToPanel(panelButtons, btnViewDetails, gbc, 4);

getContentPane().add(panelButtons, BorderLayout.SOUTH);

}

private JButton createButton(String text, ActionListener action, GridBagConstraints gbc, int gridY) {

JButton button = new JButton(text);

button.setFont(new Font("Tahoma", Font.ITALIC, 11));

button.setForeground(new Color(128, 0, 0));

button.setBackground(new Color(245, 245, 245));

button.addActionListener(action);

return button;

}

private void addButtonToPanel(JPanel panel, JButton button, GridBagConstraints gbc, int gridY) {

gbc.gridx = 0;

gbc.gridy = gridY;

panel.add(button, gbc);

}

private void fetchPatientName() {

try (Connection conn = DriverManager.getConnection(DB\_URL, USERNAME, PASSWORD)) {

String sql = "SELECT first\_name || ' ' || last\_name AS patient\_name FROM patients WHERE patient\_id = ?";

PreparedStatement pst = conn.prepareStatement(sql);

pst.setInt(1, patientId);

ResultSet rs = pst.executeQuery();

if (rs.next()) {

String patientName = rs.getString("patient\_name");

nameLabel.setText("Welcome, " + patientName);

} else {

JOptionPane.showMessageDialog(null, "Patient not found");

}

} catch (SQLException ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(null, "Error: " + ex.getMessage());

}

}

private void openAppointmentFixForm() {

new AppointmentForm(patientId).setVisible(true);

}

private void openBillingForm() {

new BillingForm(patientId).setVisible(true);

}

private void openUpdateForm() {

new UpdatePatientForm(patientId).setVisible(true);

}

private void deleteAccount() {

int response = JOptionPane.showConfirmDialog(this, "Are you sure you want to delete your account?", "Confirm", JOptionPane.YES\_NO\_OPTION);

if (response == JOptionPane.YES\_OPTION) {

try (Connection conn = DriverManager.getConnection(DB\_URL, USERNAME, PASSWORD)) {

String sql = "DELETE FROM patients WHERE patient\_id = ?";

PreparedStatement pst = conn.prepareStatement(sql);

pst.setInt(1, patientId);

int rowsDeleted = pst.executeUpdate();

if (rowsDeleted > 0) {

JOptionPane.showMessageDialog(this, "Account deleted successfully.");

dispose();

} else {

JOptionPane.showMessageDialog(this, "Failed to delete account.");

}

} catch (SQLException ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(this, "Error: " + ex.getMessage());

}

}

}

private void viewPatientDetails() {

try (Connection conn = DriverManager.getConnection(DB\_URL, USERNAME, PASSWORD)) {

String sql = "SELECT first\_name, last\_name, date\_of\_birth, gender, address, contact\_number FROM patients WHERE patient\_id = ?";

PreparedStatement pst = conn.prepareStatement(sql);

pst.setInt(1, patientId);

ResultSet rs = pst.executeQuery();

if (rs.next()) {

String firstName = rs.getString("first\_name");

String lastName = rs.getString("last\_name");

String dateOfBirth = rs.getString("date\_of\_birth");

String gender = rs.getString("gender");

String address = rs.getString("address");

String contactNumber = rs.getString("contact\_number");

StringBuilder details = new StringBuilder();

details.append("First Name: ").append(firstName).append("\n");

details.append("Last Name: ").append(lastName).append("\n");

details.append("Date of Birth: ").append(dateOfBirth).append("\n");

details.append("Gender: ").append(gender).append("\n");

details.append("Address: ").append(address).append("\n");

details.append("Contact Number: ").append(contactNumber).append("\n");

JOptionPane.showMessageDialog(this, details.toString(), "Patient Details", JOptionPane.INFORMATION\_MESSAGE);

} else {

JOptionPane.showMessageDialog(null, "Patient details not found.");

}

} catch (SQLException ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(null, "Error: " + ex.getMessage());

}

}

public static void main(String[] args) {

SwingUtilities.invokeLater(() -> new PatientDashboard(1).setVisible(true)); // Replace with actual patient ID

}

}

**Update Patient Dashboard Form:**

package com.hospital;

import javax.swing.\*;

import java.awt.\*;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

public class UpdatePatientForm extends JFrame {

private static final String DB\_URL = "jdbc:oracle:thin:@localhost:1521:XE";

private static final String USERNAME = "ehms";

private static final String PASSWORD = "epass";

private int patientId;

private JTextField firstNameField;

private JTextField lastNameField;

private JTextField contactNumberField;

private JTextField emailField;

public UpdatePatientForm(int patientId) {

setForeground(new Color(128, 0, 0));

setBackground(new Color(245, 245, 245));

this.patientId = patientId;

initializeComponents();

fetchPatientDetails();

}

private void initializeComponents() {

setTitle("Update Patient Details");

setSize(400, 300);

setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

setLocationRelativeTo(null);

getContentPane().setLayout(new BorderLayout());

JLabel titleLabel = new JLabel("E-Hoapital Manamgement System", SwingConstants.CENTER);

titleLabel.setBackground(new Color(245, 245, 245));

titleLabel.setForeground(new Color(47, 79, 79));

titleLabel.setFont(new Font("Serif", Font.BOLD, 18));

getContentPane().add(titleLabel, BorderLayout.NORTH);

JPanel formPanel = new JPanel(new GridLayout(4, 2, 10, 10));

formPanel.setBackground(new Color(245, 245, 245));

formPanel.setForeground(new Color(128, 0, 0));

getContentPane().add(formPanel, BorderLayout.CENTER);

JLabel lblFirstName = new JLabel("First Name:");

lblFirstName.setFont(new Font("Tahoma", Font.ITALIC, 11));

lblFirstName.setBackground(new Color(245, 245, 245));

lblFirstName.setForeground(new Color(128, 0, 0));

formPanel.add(lblFirstName);

firstNameField = new JTextField(15);

firstNameField.setFont(new Font("Tahoma", Font.ITALIC, 11));

firstNameField.setBackground(new Color(245, 245, 245));

firstNameField.setForeground(new Color(128, 0, 0));

formPanel.add(firstNameField);

JLabel lblLastName = new JLabel("Last Name:");

lblLastName.setFont(new Font("Tahoma", Font.ITALIC, 11));

lblLastName.setBackground(new Color(245, 245, 245));

lblLastName.setForeground(new Color(128, 0, 0));

formPanel.add(lblLastName);

lastNameField = new JTextField(15);

lastNameField.setFont(new Font("Tahoma", Font.ITALIC, 11));

lastNameField.setBackground(new Color(245, 245, 245));

lastNameField.setForeground(new Color(128, 0, 0));

formPanel.add(lastNameField);

JLabel lblContactNumber = new JLabel("Contact Number:");

lblContactNumber.setFont(new Font("Tahoma", Font.ITALIC, 11));

lblContactNumber.setForeground(new Color(128, 0, 0));

lblContactNumber.setBackground(new Color(245, 245, 245));

formPanel.add(lblContactNumber);

contactNumberField = new JTextField(15);

contactNumberField.setFont(new Font("Tahoma", Font.ITALIC, 11));

contactNumberField.setBackground(new Color(245, 245, 245));

contactNumberField.setForeground(new Color(128, 0, 0));

formPanel.add(contactNumberField);

JLabel lblEmail = new JLabel("Email:");

lblEmail.setFont(new Font("Tahoma", Font.ITALIC, 11));

lblEmail.setBackground(new Color(245, 245, 245));

lblEmail.setForeground(new Color(128, 0, 0));

formPanel.add(lblEmail);

emailField = new JTextField(15);

emailField.setFont(new Font("Tahoma", Font.ITALIC, 11));

emailField.setBackground(new Color(245, 245, 245));

emailField.setForeground(new Color(128, 0, 0));

formPanel.add(emailField);

JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.RIGHT));

buttonPanel.setBackground(new Color(245, 245, 245));

buttonPanel.setForeground(new Color(32, 178, 170));

JButton btnUpdate = new JButton("Update");

btnUpdate.setFont(new Font("Tahoma", Font.ITALIC, 11));

btnUpdate.setForeground(new Color(128, 0, 0));

btnUpdate.setBackground(new Color(245, 245, 245));

btnUpdate.addActionListener(e -> updateDetails());

buttonPanel.add(btnUpdate);

getContentPane().add(buttonPanel, BorderLayout.SOUTH);

}

private void fetchPatientDetails() {

try (Connection conn = DriverManager.getConnection(DB\_URL, USERNAME, PASSWORD)) {

String sql = "SELECT first\_name, last\_name, contact\_number, email FROM patients WHERE patient\_id = ?";

PreparedStatement pst = conn.prepareStatement(sql);

pst.setInt(1, patientId);

ResultSet rs = pst.executeQuery();

if (rs.next()) {

firstNameField.setText(rs.getString("first\_name"));

lastNameField.setText(rs.getString("last\_name"));

contactNumberField.setText(rs.getString("contact\_number"));

emailField.setText(rs.getString("email"));

} else {

JOptionPane.showMessageDialog(null, "Patient not found");

}

} catch (SQLException ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(null, "Error: " + ex.getMessage());

}

}

private void updateDetails() {

String firstName = firstNameField.getText().trim();

String lastName = lastNameField.getText().trim();

String contactNumber = contactNumberField.getText().trim();

String email = emailField.getText().trim();

try (Connection conn = DriverManager.getConnection(DB\_URL, USERNAME, PASSWORD)) {

String sql = "UPDATE patients SET first\_name = ?, last\_name = ?, contact\_number = ?, email = ? WHERE patient\_id = ?";

PreparedStatement pst = conn.prepareStatement(sql);

pst.setString(1, firstName);

pst.setString(2, lastName);

pst.setString(3, contactNumber);

pst.setString(4, email);

pst.setInt(5, patientId);

int rowsUpdated = pst.executeUpdate();

if (rowsUpdated > 0) {

JOptionPane.showMessageDialog(this, "Details updated successfully.");

dispose();

} else {

JOptionPane.showMessageDialog(this, "Failed to update details.");

}

} catch (SQLException ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(this, "Error: " + ex.getMessage());

}

}

public static void main(String[] args) {

SwingUtilities.invokeLater(() -> {

new UpdatePatientForm(1).setVisible(true);

});

}

}

**Doctor Form:**

package com.hospital;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.\*;

public class DoctorForm extends JFrame {

private JTextField doctorIdField, firstNameField, lastNameField, specializationField, contactNumberField, emailField, hospitalIdField;

private JPasswordField passwordsField; // New password field

private JButton addButton, updateButton, deleteButton, viewButton;

// Database connection details

private static final String DB\_URL = "jdbc:oracle:thin:@localhost:1521:XE";

private static final String USERNAME = "ehms";

private static final String PASSWORD = "epass";

public DoctorForm() {

getContentPane().setBackground(new Color(245, 245, 245));

setTitle("Doctor Management");

setSize(500, 500); // Increased height to accommodate password field

setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

setLocationRelativeTo(null); // Center the window

getContentPane().setLayout(new BorderLayout(10, 10)); // Main layout

JLabel titleLabel = new JLabel("E-Hospital Management System", JLabel.CENTER);

titleLabel.setForeground(new Color(47, 79, 79));

titleLabel.setBackground(new Color(245, 245, 245));

titleLabel.setFont(new Font("Serif", Font.BOLD, 16));

getContentPane().add(titleLabel, BorderLayout.NORTH);

JPanel formPanel = new JPanel(new GridBagLayout());

formPanel.setForeground(new Color(0, 0, 0));

formPanel.setBackground(new Color(245, 245, 245));

JLabel label = new JLabel("Doctor ID:");

label.setBackground(new Color(245, 245, 245));

label.setForeground(new Color(0, 0, 0));

label.setFont(new Font("Tahoma", Font.PLAIN, 11));

addFormComponent(formPanel, label, 0, 0);

doctorIdField = new JTextField(15);

addFormComponent(formPanel, doctorIdField, 1, 0);

addFormComponent(formPanel, new JLabel("First Name:"), 0, 1);

firstNameField = new JTextField(15);

addFormComponent(formPanel, firstNameField, 1, 1);

addFormComponent(formPanel, new JLabel("Last Name:"), 0, 2);

lastNameField = new JTextField(15);

addFormComponent(formPanel, lastNameField, 1, 2);

addFormComponent(formPanel, new JLabel("Specialization:"), 0, 3);

specializationField = new JTextField(15);

addFormComponent(formPanel, specializationField, 1, 3);

addFormComponent(formPanel, new JLabel("Contact Number:"), 0, 4);

contactNumberField = new JTextField(15);

addFormComponent(formPanel, contactNumberField, 1, 4);

addFormComponent(formPanel, new JLabel("Email:"), 0, 5);

emailField = new JTextField(15);

addFormComponent(formPanel, emailField, 1, 5);

addFormComponent(formPanel, new JLabel("Hospital ID:"), 0, 6);

hospitalIdField = new JTextField(15);

addFormComponent(formPanel, hospitalIdField, 1, 6);

addFormComponent(formPanel, new JLabel("Passwords:"), 0, 7);

passwordsField = new JPasswordField(15);

addFormComponent(formPanel, passwordsField, 1, 7);

getContentPane().add(formPanel, BorderLayout.CENTER);

JPanel buttonPanel = new JPanel(new GridLayout(1, 4, 10, 10));

buttonPanel.setBackground(new Color(245, 245, 245));

buttonPanel.setForeground(new Color(128, 0, 0));

addButton = new JButton("Add Doctor");

addButton.setFont(new Font("Tahoma", Font.ITALIC, 11));

addButton.setForeground(new Color(128, 0, 0));

addButton.setBackground(new Color(245, 245, 245));

buttonPanel.add(addButton);

updateButton = new JButton("Update Doctor");

updateButton.setBackground(new Color(245, 245, 245));

updateButton.setFont(new Font("Tahoma", Font.ITALIC, 11));

updateButton.setForeground(new Color(128, 0, 0));

buttonPanel.add(updateButton);

deleteButton = new JButton("Delete Doctor");

deleteButton.setFont(new Font("Tahoma", Font.ITALIC, 11));

deleteButton.setBackground(new Color(245, 245, 245));

deleteButton.setForeground(new Color(128, 0, 0));

buttonPanel.add(deleteButton);

viewButton = new JButton("View Doctor");

viewButton.setFont(new Font("Tahoma", Font.ITALIC, 11));

viewButton.setBackground(new Color(245, 245, 245));

viewButton.setForeground(new Color(128, 0, 0));

buttonPanel.add(viewButton);

getContentPane().add(buttonPanel, BorderLayout.SOUTH);

addButton.addActionListener(e -> addDoctor());

updateButton.addActionListener(e -> updateDoctor());

deleteButton.addActionListener(e -> deleteDoctor());

viewButton.addActionListener(e -> viewDoctor());

doctorIdField.addActionListener(e -> fetchDoctorDetails(Integer.parseInt(doctorIdField.getText())));

setVisible(true);

}

private void addFormComponent(JPanel panel, Component comp, int x, int y) {

GridBagConstraints gbc = new GridBagConstraints();

gbc.insets = new Insets(10, 10, 10, 10);

gbc.anchor = GridBagConstraints.WEST;

gbc.gridx = x;

gbc.gridy = y;

panel.add(comp, gbc);

}

private void addDoctor() {

String firstName = firstNameField.getText();

String lastName = lastNameField.getText();

String specialization = specializationField.getText();

String contactNumber = contactNumberField.getText();

String email = emailField.getText();

int hospitalId = Integer.parseInt(hospitalIdField.getText());

String passwords = new String(passwordsField.getPassword()); // Get password as String

try {

Connection conn = DriverManager.getConnection(DB\_URL, USERNAME, PASSWORD);

String sql = "INSERT INTO EHMS.DOCTORS (DOCTOR\_ID, FIRST\_NAME, LAST\_NAME, SPECIALIZATION, CONTACT\_NUMBER, EMAIL, HOSPITAL\_ID, PASSWORDS) " +

"VALUES (SEQ\_DOCTORS.NEXTVAL, ?, ?, ?, ?, ?, ?, ?)";

try (PreparedStatement pst = conn.prepareStatement(sql)) {

pst.setString(1, firstName);

pst.setString(2, lastName);

pst.setString(3, specialization);

pst.setString(4, contactNumber);

pst.setString(5, email);

pst.setInt(6, hospitalId);

pst.setString(7, passwords);

int rowsInserted = pst.executeUpdate();

if (rowsInserted > 0) {

JOptionPane.showMessageDialog(null, "Doctor added successfully!");

clearFields();

} else {

JOptionPane.showMessageDialog(null, "Failed to add doctor.", "Error", JOptionPane.ERROR\_MESSAGE);

}

}

conn.close();

} catch (SQLException ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(null, "Error adding doctor: " + ex.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);

} catch (NumberFormatException ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(null, "Invalid hospital ID. Please enter a valid number.", "Error", JOptionPane.ERROR\_MESSAGE);

}

}

private void updateDoctor() {

int doctorId = Integer.parseInt(doctorIdField.getText());

String firstName = firstNameField.getText();

String lastName = lastNameField.getText();

String specialization = specializationField.getText();

String contactNumber = contactNumberField.getText();

String email = emailField.getText();

int hospitalId = Integer.parseInt(hospitalIdField.getText());

String passwords = new String(passwordsField.getPassword()); // Get password as String

try {

Connection conn = DriverManager.getConnection(DB\_URL, USERNAME, PASSWORD);

String sql = "UPDATE EHMS.DOCTORS SET FIRST\_NAME = ?, LAST\_NAME = ?, SPECIALIZATION = ?, CONTACT\_NUMBER = ?, EMAIL = ?, HOSPITAL\_ID = ?, PASSWORDS = ? WHERE DOCTOR\_ID = ?";

try (PreparedStatement pst = conn.prepareStatement(sql)) {

pst.setString(1, firstName);

pst.setString(2, lastName);

pst.setString(3, specialization);

pst.setString(4, contactNumber);

pst.setString(5, email);

pst.setInt(6, hospitalId);

pst.setString(7, passwords);

pst.setInt(8, doctorId);

int rowsUpdated = pst.executeUpdate();

if (rowsUpdated > 0) {

JOptionPane.showMessageDialog(null, "Doctor updated successfully!");

clearFields();

} else {

JOptionPane.showMessageDialog(null, "Failed to update doctor.", "Error", JOptionPane.ERROR\_MESSAGE);

}

}

conn.close();

} catch (SQLException ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(null, "Error updating doctor: " + ex.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);

} catch (NumberFormatException ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(null, "Invalid ID. Please enter a valid number.", "Error", JOptionPane.ERROR\_MESSAGE);

}

}

private void deleteDoctor() {

int doctorId = Integer.parseInt(doctorIdField.getText());

int response = JOptionPane.showConfirmDialog(this, "Are you sure you want to delete this doctor?", "Confirm", JOptionPane.YES\_NO\_OPTION);

if (response == JOptionPane.YES\_OPTION) {

try {

Connection conn = DriverManager.getConnection(DB\_URL, USERNAME, PASSWORD);

String sql = "DELETE FROM EHMS.DOCTORS WHERE DOCTOR\_ID = ?";

try (PreparedStatement pst = conn.prepareStatement(sql)) {

pst.setInt(1, doctorId);

int rowsDeleted = pst.executeUpdate();

if (rowsDeleted > 0) {

JOptionPane.showMessageDialog(null, "Doctor deleted successfully!");

clearFields();

} else {

JOptionPane.showMessageDialog(null, "Failed to delete doctor.", "Error", JOptionPane.ERROR\_MESSAGE);

}

}

conn.close();

} catch (SQLException ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(null, "Error deleting doctor: " + ex.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);

}

}

}

private void viewDoctor() {

int doctorId = Integer.parseInt(doctorIdField.getText());

try {

Connection conn = DriverManager.getConnection(DB\_URL, USERNAME, PASSWORD);

String sql = "SELECT \* FROM EHMS.DOCTORS WHERE DOCTOR\_ID = ?";

try (PreparedStatement pst = conn.prepareStatement(sql)) {

pst.setInt(1, doctorId);

ResultSet rs = pst.executeQuery();

if (rs.next()) {

firstNameField.setText(rs.getString("FIRST\_NAME"));

lastNameField.setText(rs.getString("LAST\_NAME"));

specializationField.setText(rs.getString("SPECIALIZATION"));

contactNumberField.setText(rs.getString("CONTACT\_NUMBER"));

emailField.setText(rs.getString("EMAIL"));

hospitalIdField.setText(String.valueOf(rs.getInt("HOSPITAL\_ID")));

passwordsField.setText(rs.getString("PASSWORDS")); // Set password field

} else {

JOptionPane.showMessageDialog(null, "Doctor not found.", "Error", JOptionPane.ERROR\_MESSAGE);

}

}

conn.close();

} catch (SQLException ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(null, "Error viewing doctor: " + ex.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);

} catch (NumberFormatException ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(null, "Invalid ID. Please enter a valid number.", "Error", JOptionPane.ERROR\_MESSAGE);

}

}

private void fetchDoctorDetails(int doctorId) {

try {

Connection conn = DriverManager.getConnection(DB\_URL, USERNAME, PASSWORD);

String sql = "SELECT \* FROM EHMS.DOCTORS WHERE DOCTOR\_ID = ?";

try (PreparedStatement pst = conn.prepareStatement(sql)) {

pst.setInt(1, doctorId);

ResultSet rs = pst.executeQuery();

if (rs.next()) {

firstNameField.setText(rs.getString("FIRST\_NAME"));

lastNameField.setText(rs.getString("LAST\_NAME"));

specializationField.setText(rs.getString("SPECIALIZATION"));

contactNumberField.setText(rs.getString("CONTACT\_NUMBER"));

emailField.setText(rs.getString("EMAIL"));

hospitalIdField.setText(String.valueOf(rs.getInt("HOSPITAL\_ID")));

passwordsField.setText(rs.getString("PASSWORDS")); // Set password field

} else {

clearFields();

}

}

conn.close();

} catch (SQLException ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(null, "Error fetching doctor details: " + ex.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);

}

}

private void clearFields() {

doctorIdField.setText("");

firstNameField.setText("");

lastNameField.setText("");

specializationField.setText("");

contactNumberField.setText("");

emailField.setText("");

hospitalIdField.setText("");

passwordsField.setText(""); // Clear password field

}

public static void main(String[] args) {

SwingUtilities.invokeLater(() -> new DoctorForm());

}

}

**Billing Form:**

package com.hospital;

import javax.swing.\*;

import java.awt.\*;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.SQLException;

import java.sql.Timestamp;

import java.util.Date;

public class BillingForm extends JFrame {

private static final String DB\_URL = "jdbc:oracle:thin:@localhost:1521:XE";

private static final String USERNAME = "ehms";

private static final String PASSWORD = "epass";

private int patientId;

private JTextField amountField;

private JTextField paymentStatusField;

public BillingForm(int patientId) {

setForeground(new Color(128, 0, 0));

setBackground(new Color(245, 245, 245));

this.patientId = patientId;

initializeComponents();

}

private void initializeComponents() {

setTitle("Manage Billing");

setSize(400, 200);

setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

setLocationRelativeTo(null); // Center the window

getContentPane().setLayout(new BorderLayout(10, 10));

JLabel titleLabel = new JLabel("E-Hospital Management System", JLabel.CENTER);

titleLabel.setForeground(new Color(47, 79, 79));

titleLabel.setBackground(new Color(245, 245, 245));

titleLabel.setFont(new Font("Serif", Font.BOLD, 16));

getContentPane().add(titleLabel, BorderLayout.NORTH);

JPanel formPanel = new JPanel(new GridBagLayout());

formPanel.setForeground(new Color(128, 0, 0));

formPanel.setBackground(new Color(245, 245, 245));

// Create a new GridBagConstraints object for each component

GridBagConstraints gbcLblAmount = new GridBagConstraints();

gbcLblAmount.insets = new Insets(10, 10, 10, 10);

gbcLblAmount.gridx = 0;

gbcLblAmount.gridy = 0;

gbcLblAmount.anchor = GridBagConstraints.EAST;

JLabel lblAmount = new JLabel("Amount:");

lblAmount.setForeground(new Color(128, 0, 0));

lblAmount.setFont(new Font("Tahoma", Font.ITALIC, 11));

lblAmount.setBackground(new Color(245, 245, 245));

formPanel.add(lblAmount, gbcLblAmount);

GridBagConstraints gbcAmountField = new GridBagConstraints();

gbcAmountField.insets = new Insets(10, 10, 10, 10);

gbcAmountField.gridx = 1;

gbcAmountField.anchor = GridBagConstraints.WEST;

amountField = new JTextField(15);

amountField.setFont(new Font("Tahoma", Font.ITALIC, 11));

amountField.setBackground(new Color(245, 245, 245));

amountField.setForeground(new Color(128, 0, 0));

formPanel.add(amountField, gbcAmountField);

GridBagConstraints gbcLblPaymentStatus = new GridBagConstraints();

gbcLblPaymentStatus.insets = new Insets(10, 10, 10, 10);

gbcLblPaymentStatus.gridx = 0;

gbcLblPaymentStatus.gridy = 1;

gbcLblPaymentStatus.anchor = GridBagConstraints.EAST;

JLabel lblPaymentStatus = new JLabel("Payment Status:");

lblPaymentStatus.setForeground(new Color(128, 0, 0));

lblPaymentStatus.setFont(new Font("Tahoma", Font.ITALIC, 11));

lblPaymentStatus.setBackground(new Color(245, 245, 245));

formPanel.add(lblPaymentStatus, gbcLblPaymentStatus);

GridBagConstraints gbcPaymentStatusField = new GridBagConstraints();

gbcPaymentStatusField.insets = new Insets(10, 10, 10, 10);

gbcPaymentStatusField.gridx = 1;

gbcPaymentStatusField.anchor = GridBagConstraints.WEST;

paymentStatusField = new JTextField(15);

paymentStatusField.setFont(new Font("Tahoma", Font.ITALIC, 11));

paymentStatusField.setBackground(new Color(245, 245, 245));

paymentStatusField.setForeground(new Color(128, 0, 0));

formPanel.add(paymentStatusField, gbcPaymentStatusField);

getContentPane().add(formPanel, BorderLayout.CENTER);

JButton btnSave = new JButton("Save");

btnSave.setFont(new Font("Tahoma", Font.ITALIC, 11));

btnSave.setForeground(new Color(128, 0, 0));

btnSave.setBackground(new Color(245, 245, 245));

btnSave.addActionListener(e -> saveBillingInfo());

JPanel buttonPanel = new JPanel();

buttonPanel.setForeground(new Color(128, 0, 0));

buttonPanel.setBackground(new Color(245, 245, 245));

buttonPanel.add(btnSave);

getContentPane().add(buttonPanel, BorderLayout.SOUTH);

setVisible(true);

}

private void saveBillingInfo() {

String amountStr = amountField.getText().trim();

String paymentStatus = paymentStatusField.getText().trim();

double amount;

try {

amount = Double.parseDouble(amountStr);

} catch (NumberFormatException ex) {

JOptionPane.showMessageDialog(null, "Invalid amount format. Please enter a valid number.");

return;

}

Timestamp currentTimestamp = new Timestamp(new Date().getTime());

try (Connection conn = DriverManager.getConnection(DB\_URL, USERNAME, PASSWORD)) {

String sql = "INSERT INTO BILLING (BILL\_ID, PATIENT\_ID, CHECKIN\_DATE\_TIME, CHECKOUT\_DATE\_TIME, TOTAL\_AMOUNT, PAYMENT\_STATUS) " +

"VALUES (BILLING\_SEQ.NEXTVAL, ?, ?, ?, ?, ?)";

PreparedStatement pst = conn.prepareStatement(sql);

pst.setInt(1, patientId);

pst.setTimestamp(2, currentTimestamp); // Assuming current time for CHECKIN\_DATE\_TIME

pst.setTimestamp(3, currentTimestamp); // Assuming current time for CHECKOUT\_DATE\_TIME

pst.setDouble(4, amount);

pst.setString(5, paymentStatus);

int rowsInserted = pst.executeUpdate();

if (rowsInserted > 0) {

JOptionPane.showMessageDialog(null, "Billing information saved successfully!");

dispose();

} else {

JOptionPane.showMessageDialog(null, "Failed to save billing information. Please try again.");

}

} catch (SQLException ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(null, "Error: " + ex.getMessage());

}

}

public static void main(String[] args) {

SwingUtilities.invokeLater(() -> new BillingForm(1).setVisible(true)); // Replace with actual patient ID

}

}

**Appointment Form:**

package com.hospital;

import javax.swing.\*;

import java.awt.\*;

import java.sql.\*;

import java.util.ArrayList;

import java.util.List;

public class AppointmentForm extends JFrame {

private static final String DB\_URL = "jdbc:oracle:thin:@localhost:1521:XE";

private static final String USERNAME = "ehms";

private static final String PASSWORD = "epass";

private int patientId;

private JTextField dateField;

private JTextArea reasonArea;

private JComboBox<String> doctorComboBox;

private List<Integer> doctorIds;

public AppointmentForm(int patientId) {

this.patientId = patientId;

initializeComponents();

}

private void initializeComponents() {

setTitle("Fix Appointment");

setSize(400, 350);

setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

setLocationRelativeTo(null); // Center the window

setLayout(new BorderLayout(10, 10));

// Title

JLabel titleLabel = new JLabel("E-Hospital Management System", JLabel.CENTER);

titleLabel.setFont(new Font("Serif", Font.BOLD | Font.ITALIC, 16));

add(titleLabel, BorderLayout.NORTH);

// Form panel using GridBagLayout

JPanel formPanel = new JPanel(new GridBagLayout());

GridBagConstraints gbc = new GridBagConstraints();

gbc.insets = new Insets(20, 20, 20, 20);

gbc.fill = GridBagConstraints.HORIZONTAL;

// Doctor label and combo box

gbc.gridx = 0;

gbc.gridy = 0;

JLabel lblDoctor = new JLabel("Select Doctor:");

formPanel.add(lblDoctor, gbc);

gbc.gridx = 1;

doctorComboBox = new JComboBox<>();

doctorIds = new ArrayList<>();

loadDoctors();

formPanel.add(doctorComboBox, gbc);

// Date label and text field

gbc.gridx = 0;

gbc.gridy = 1;

JLabel lblDate = new JLabel("Appointment Date (YYYY-MM-DD):");

formPanel.add(lblDate, gbc);

gbc.gridx = 1;

dateField = new JTextField(20);

formPanel.add(dateField, gbc);

// Reason label and text area

gbc.gridx = 0;

gbc.gridy = 2;

JLabel lblReason = new JLabel("Reason:");

formPanel.add(lblReason, gbc);

gbc.gridx = 1;

gbc.gridy = 2;

gbc.gridheight = 2;

reasonArea = new JTextArea(5, 20);

formPanel.add(new JScrollPane(reasonArea), gbc);

gbc.gridheight = 1; // Reset gridheight to default

// Fix Appointment button with increased size

gbc.gridx = 1;

gbc.gridy = 4;

gbc.anchor = GridBagConstraints.CENTER;

JButton btnFixAppointment = new JButton("Fix Appointment");

btnFixAppointment.setPreferredSize(new Dimension(150, 40)); // Increase size of button

btnFixAppointment.addActionListener(e -> fixAppointment());

formPanel.add(btnFixAppointment, gbc);

add(formPanel, BorderLayout.CENTER);

setVisible(true);

}

private void loadDoctors() {

try (Connection conn = DriverManager.getConnection(DB\_URL, USERNAME, PASSWORD)) {

String sql = "SELECT DOCTOR\_ID, FIRST\_NAME, LAST\_NAME FROM EHMS.DOCTORS";

try (Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery(sql)) {

while (rs.next()) {

int doctorId = rs.getInt("DOCTOR\_ID");

String doctorName = rs.getString("FIRST\_NAME") + " " + rs.getString("LAST\_NAME");

doctorComboBox.addItem(doctorName);

doctorIds.add(doctorId);

}

}

} catch (SQLException ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(this, "Error loading doctors: " + ex.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);

}

}

private void fixAppointment() {

String appointmentDate = dateField.getText().trim();

String reason = reasonArea.getText().trim();

int selectedDoctorIndex = doctorComboBox.getSelectedIndex();

int doctorId = doctorIds.get(selectedDoctorIndex);

if (appointmentDate.isEmpty() || reason.isEmpty()) {

JOptionPane.showMessageDialog(this, "Appointment date and reason cannot be empty.");

return;

}

try {

java.sql.Date sqlAppointmentDate = java.sql.Date.valueOf(appointmentDate);

try (Connection conn = DriverManager.getConnection(DB\_URL, USERNAME, PASSWORD)) {

String sql = "INSERT INTO appointments (patient\_id, appointment\_date, reason, doctor\_id) VALUES (?, ?, ?, ?)";

PreparedStatement pst = conn.prepareStatement(sql);

pst.setInt(1, patientId);

pst.setDate(2, sqlAppointmentDate);

pst.setString(3, reason);

pst.setInt(4, doctorId);

int rowsInserted = pst.executeUpdate();

if (rowsInserted > 0) {

JOptionPane.showMessageDialog(this, "Appointment fixed successfully!");

dispose();

} else {

JOptionPane.showMessageDialog(this, "Failed to fix appointment. Please try again.");

}

} catch (SQLException ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(this, "Error: " + ex.getMessage());

}

} catch (IllegalArgumentException e) {

JOptionPane.showMessageDialog(this, "Invalid date format. Please use YYYY-MM-DD.");

}

}

public static void main(String[] args) {

SwingUtilities.invokeLater(() -> new AppointmentForm(1).setVisible(true)); // Replace with actual patient ID

}

}