import javax.swing.\*;

import java.awt.\*;

import java.sql.\*;

public class HospitalManagement extends JFrame {

private static final String URL = "jdbc:mysql://localhost:3306/test";

private static final String USER = "root";

private static final String PASSWORD = "Brijith123$$$";

private JTextField patientNameField, ageField, diagnosisField, doctorNameField, specialtyField;

private JTextArea outputArea;

public static void main(String[] args) {

SwingUtilities.invokeLater(() -> {

try {

new HospitalManagement().setVisible(true);

} catch (SQLException e) {

e.printStackTrace();

}

});

}

public HospitalManagement() throws SQLException {

setTitle("Hospital Management System");

setSize(600, 600);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLocationRelativeTo(null);

setLayout(new BorderLayout());

JPanel inputPanel = new JPanel(new GridLayout(5, 2));

// Input fields

patientNameField = new JTextField();

ageField = new JTextField();

diagnosisField = new JTextField();

doctorNameField = new JTextField();

specialtyField = new JTextField();

inputPanel.add(new JLabel("Patient Name:"));

inputPanel.add(patientNameField);

inputPanel.add(new JLabel("Age:"));

inputPanel.add(ageField);

inputPanel.add(new JLabel("Diagnosis:"));

inputPanel.add(diagnosisField);

inputPanel.add(new JLabel("Doctor Name:"));

inputPanel.add(doctorNameField);

inputPanel.add(new JLabel("Specialty:"));

inputPanel.add(specialtyField);

// Buttons for CRUD operations

JPanel buttonPanel = new JPanel();

JButton addButton = new JButton("Add Record");

JButton viewButton = new JButton("View Records");

JButton deleteButton = new JButton("Delete Record");

JButton updateButton = new JButton("Update Record"); // Renamed Update Button

buttonPanel.add(addButton);

buttonPanel.add(viewButton);

buttonPanel.add(deleteButton);

buttonPanel.add(updateButton); // Add renamed Update Button to panel

// Output area for results

outputArea = new JTextArea(10, 40);

outputArea.setEditable(false);

JScrollPane scrollPane = new JScrollPane(outputArea);

add(inputPanel, BorderLayout.NORTH);

add(buttonPanel, BorderLayout.CENTER);

add(scrollPane, BorderLayout.SOUTH);

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

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

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

updateButton.addActionListener(e -> updateRecord()); // Add ActionListener for renamed Update Button

}

private Connection getConnection() throws SQLException {

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

}

private void addRecord() {

String patientName = patientNameField.getText();

int age = Integer.parseInt(ageField.getText());

String diagnosis = diagnosisField.getText();

String doctorName = doctorNameField.getText();

String specialty = specialtyField.getText();

String sql = "INSERT INTO hospital (patient\_name, age, diagnosis, doctor\_name, specialty) VALUES (?, ?, ?, ?, ?)";

try (Connection conn = getConnection(); PreparedStatement stmt = conn.prepareStatement(sql)) {

stmt.setString(1, patientName);

stmt.setInt(2, age);

stmt.setString(3, diagnosis);

stmt.setString(4, doctorName);

stmt.setString(5, specialty);

stmt.executeUpdate();

outputArea.setText("Record added successfully!");

} catch (SQLException e) {

e.printStackTrace();

outputArea.setText("Error adding record.");

}

}

private void viewRecords() {

String sql = "SELECT \* FROM hospital";

StringBuilder output = new StringBuilder();

try (Connection conn = getConnection(); Statement stmt = conn.createStatement(); ResultSet rs = stmt.executeQuery(sql)) {

while (rs.next()) {

output.append("Patient ID: ").append(rs.getInt("patient\_id")).append(", ")

.append("Name: ").append(rs.getString("patient\_name")).append(", ")

.append("Age: ").append(rs.getInt("age")).append(", ")

.append("Diagnosis: ").append(rs.getString("diagnosis")).append(", ")

.append("Doctor Name: ").append(rs.getString("doctor\_name")).append(", ")

.append("Specialty: ").append(rs.getString("specialty")).append("\n");

}

outputArea.setText(output.toString());

} catch (SQLException e) {

e.printStackTrace();

outputArea.setText("Error retrieving records.");

}

}

private void deleteRecord() {

String patientId = JOptionPane.showInputDialog(this, "Enter Patient ID to delete:");

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

try (Connection conn = getConnection(); PreparedStatement stmt = conn.prepareStatement(sql)) {

stmt.setInt(1, Integer.parseInt(patientId));

int rowsAffected = stmt.executeUpdate();

if (rowsAffected > 0) {

outputArea.setText("Record deleted successfully!");

} else {

outputArea.setText("No record found with Patient ID " + patientId);

}

} catch (SQLException e) {

e.printStackTrace();

outputArea.setText("Error deleting record.");

}

}

// New method to update doctor name for a given patient ID

private void updateRecord() {

String patientId = JOptionPane.showInputDialog(this, "Enter Patient ID to update:");

String newDoctorName = JOptionPane.showInputDialog(this, "Enter new Doctor Name:");

String sql = "UPDATE hospital SET doctor\_name = ? WHERE patient\_id = ?";

try (Connection conn = getConnection(); PreparedStatement stmt = conn.prepareStatement(sql)) {

stmt.setString(1, newDoctorName);

stmt.setInt(2, Integer.parseInt(patientId));

int rowsAffected = stmt.executeUpdate();

if (rowsAffected > 0) {

outputArea.setText("Doctor name updated successfully for Patient ID " + patientId);

} else {

outputArea.setText("No record found with Patient ID " + patientId);

}

} catch (SQLException e) {

e.printStackTrace();

outputArea.setText("Error updating record.");

}

}

}