import javafx.application.Application;

import javafx.geometry.Insets;

import javafx.scene.Scene;

import javafx.scene.control.\*;

import javafx.scene.layout.GridPane;

import javafx.stage.Stage;

import java.sql.\*;

public class StudentManagementSystem extends Application {

// SQLite database connection

private static final String DB\_URL = "jdbc:sqlite:students.db";

public static void main(String[] args) {

createDatabase();

launch(args);

}

@Override

public void start(Stage primaryStage) {

primaryStage.setTitle("Student Management System");

// Labels and TextFields for student information

Label lblName = new Label("Name:");

TextField txtName = new TextField();

Label lblGrade = new Label("Grade:");

TextField txtGrade = new TextField();

Label lblAttendance = new Label("Attendance:");

TextField txtAttendance = new TextField();

Label lblAssignment = new Label("Assignment:");

TextField txtAssignment = new TextField();

// Buttons

Button btnAdd = new Button("Add Student");

Button btnUpdate = new Button("Update Student");

Button btnDelete = new Button("Delete Student");

Button btnReport = new Button("Generate Report");

// Layout

GridPane gridPane = new GridPane();

gridPane.setPadding(new Insets(10, 10, 10, 10));

gridPane.setVgap(8);

gridPane.setHgap(10);

// Adding components to the layout

gridPane.add(lblName, 0, 0);

gridPane.add(txtName, 1, 0);

gridPane.add(lblGrade, 0, 1);

gridPane.add(txtGrade, 1, 1);

gridPane.add(lblAttendance, 0, 2);

gridPane.add(txtAttendance, 1, 2);

gridPane.add(lblAssignment, 0, 3);

gridPane.add(txtAssignment, 1, 3);

gridPane.add(btnAdd, 0, 4);

gridPane.add(btnUpdate, 1, 4);

gridPane.add(btnDelete, 2, 4);

gridPane.add(btnReport, 1, 5);

// Button functionality

btnAdd.setOnAction(e -> addStudent(txtName.getText(), txtGrade.getText(), txtAttendance.getText(), txtAssignment.getText()));

btnUpdate.setOnAction(e -> updateStudent(txtName.getText(), txtGrade.getText(), txtAttendance.getText(), txtAssignment.getText()));

btnDelete.setOnAction(e -> deleteStudent(txtName.getText()));

btnReport.setOnAction(e -> generateReport());

// Scene and stage setup

Scene scene = new Scene(gridPane, 400, 300);

primaryStage.setScene(scene);

primaryStage.show();

}

// Database setup

private static void createDatabase() {

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

if (conn != null) {

Statement stmt = conn.createStatement();

String createTable = "CREATE TABLE IF NOT EXISTS students ("

+ "name TEXT PRIMARY KEY,"

+ "grade TEXT,"

+ "attendance TEXT,"

+ "assignment TEXT)";

stmt.execute(createTable);

}

} catch (SQLException e) {

System.out.println(e.getMessage());

}

}

// Add a student to the database

private void addStudent(String name, String grade, String attendance, String assignment) {

String sql = "INSERT INTO students(name, grade, attendance, assignment) VALUES(?, ?, ?, ?)";

try (Connection conn = DriverManager.getConnection(DB\_URL);

PreparedStatement pstmt = conn.prepareStatement(sql)) {

pstmt.setString(1, name);

pstmt.setString(2, grade);

pstmt.setString(3, attendance);

pstmt.setString(4, assignment);

pstmt.executeUpdate();

System.out.println("Student added successfully.");

} catch (SQLException e) {

System.out.println(e.getMessage());

}

}

// Update a student's information in the database

private void updateStudent(String name, String grade, String attendance, String assignment) {

String sql = "UPDATE students SET grade = ?, attendance = ?, assignment = ? WHERE name = ?";

try (Connection conn = DriverManager.getConnection(DB\_URL);

PreparedStatement pstmt = conn.prepareStatement(sql)) {

pstmt.setString(1, grade);

pstmt.setString(2, attendance);

pstmt.setString(3, assignment);

pstmt.setString(4, name);

pstmt.executeUpdate();

System.out.println("Student updated successfully.");

} catch (SQLException e) {

System.out.println(e.getMessage());

}

}

// Delete a student from the database

private void deleteStudent(String name) {

String sql = "DELETE FROM students WHERE name = ?";

try (Connection conn = DriverManager.getConnection(DB\_URL);

PreparedStatement pstmt = conn.prepareStatement(sql)) {

pstmt.setString(1, name);

pstmt.executeUpdate();

System.out.println("Student deleted successfully.");

} catch (SQLException e) {

System.out.println(e.getMessage());

}

}

// Generate a report for all students

private void generateReport() {

String sql = "SELECT \* FROM students";

try (Connection conn = DriverManager.getConnection(DB\_URL);

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery(sql)) {

System.out.println("Student Report:");

while (rs.next()) {

System.out.println("Name: " + rs.getString("name"));

System.out.println("Grade: " + rs.getString("grade"));

System.out.println("Attendance: " + rs.getString("attendance"));

System.out.println("Assignment: " + rs.getString("assignment"));

System.out.println("--------------------");

}

} catch (SQLException e) {

System.out.println(e.getMessage());

}

}

}