1.Creating my sql table

```
CREATE DATABASE student_db;


USE student_db;


CREATE TABLE students (
    student_id INT PRIMARY KEY,
    name VARCHAR(100),
    department VARCHAR(100),
    marks FLOAT
);
```

2.creating modal class

```
package model;


public class Student {
    private int studentId;
    private String name;
    private String department;
    private float marks;


    public Student(int studentId, String name, String department, float marks) {
        this.studentId = studentId;
        this.name = name;
        this.department = department;
        this.marks = marks;
    }


    // Getters and setters
    public int getStudentId() { return studentId; }
    public void setStudentId(int studentId) { this.studentId = studentId; }
```

```
    public String getName() { return name; }
    public void setName(String name) { this.name = name; }

    public String getDepartment() { return department; }
    public void setDepartment(String department) { this.department = department; }

    public float getMarks() { return marks; }
    public void setMarks(float marks) { this.marks = marks; }

    @Override
    public String toString() {
        return "Student [ID=" + studentId + ", Name=" + name + ", Department=" + department + ",
Marks=" + marks + "]";
    }
}
```

3. StudentController class for database operations.

```
package controller;

import model.Student;
import java.sql.*;
import java.util.ArrayList;
import java.util.List;

public class StudentController {

    private Connection conn;
```

```
public StudentController(Connection conn) {
    this.conn = conn;
}

public void addStudent(Student s) throws SQLException {
    String sql = "INSERT INTO students VALUES (?, ?, ?, ?)";
    PreparedStatement ps = conn.prepareStatement(sql);
    ps.setInt(1, s.getStudentId());
    ps.setString(2, s.getName());
    ps.setString(3, s.getDepartment());
    ps.setFloat(4, s.getMarks());
    ps.executeUpdate();
}

public void updateStudent(Student s) throws SQLException {
    String sql = "UPDATE students SET name=?, department=?, marks=? WHERE student_id=?";
    PreparedStatement ps = conn.prepareStatement(sql);
    ps.setString(1, s.getName());
    ps.setString(2, s.getDepartment());
    ps.setFloat(3, s.getMarks());
    ps.setInt(4, s.getStudentId());
    ps.executeUpdate();
}

public void deleteStudent(int id) throws SQLException {
    String sql = "DELETE FROM students WHERE student_id=?";
    PreparedStatement ps = conn.prepareStatement(sql);
    ps.setInt(1, id);
    ps.executeUpdate();
}
```

```
public Student getStudent(int id) throws SQLException {
    String sql = "SELECT * FROM students WHERE student_id=?";
    PreparedStatement ps = conn.prepareStatement(sql);
    ps.setInt(1, id);
    ResultSet rs = ps.executeQuery();
    if (rs.next()) {
        return new Student(
            rs.getInt("student_id"),
            rs.getString("name"),
            rs.getString("department"),
            rs.getFloat("marks")
        );
    }
    return null;
}

public List<Student> getAllStudents() throws SQLException {
    List<Student> list = new ArrayList<>();
    String sql = "SELECT * FROM students";
    Statement stmt = conn.createStatement();
    ResultSet rs = stmt.executeQuery(sql);
    while (rs.next()) {
        list.add(new Student(
            rs.getInt("student_id"),
            rs.getString("name"),
            rs.getString("department"),
            rs.getFloat("marks")
        ));
    }
    return list;
}
```

```
}
```

4. DBConnection

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBConnection {
    public static Connection getConnection() throws SQLException {
        String url = "jdbc:mysql://localhost:3306/student_db";
        String user = "root";
        String password = "your_password"; // Replace with your password
        return DriverManager.getConnection(url, user, password);
    }
}
```

5. StudentView

```
package view;

import controller.StudentController;
import model.Student;
import java.sql.Connection;
import java.sql.SQLException;
import java.util.List;
import java.util.Scanner;

public class StudentView {

    public static void main(String[] args) {
        try (Connection conn = DBConnection.getConnection(); Scanner sc = new Scanner(System.in)) {
            StudentController controller = new StudentController(conn);
            while (true) {
```

```
System.out.println("\n===== Student Management System =====");
System.out.println("1. Add Student");
System.out.println("2. Update Student");
System.out.println("3. Delete Student");
System.out.println("4. View Student");
System.out.println("5. View All Students");
System.out.println("6. Exit");
System.out.print("Choose an option: ");
int choice = sc.nextInt();

switch (choice) {
   case 1:
      System.out.print("Enter ID, Name, Department, Marks: ");
      Student s = new Student(sc.nextInt(), sc.next(), sc.next(), sc.nextFloat());
      controller.addStudent(s);
      System.out.println("Student added.");
      break;
   case 2:
      System.out.print("Enter ID to update: ");
      int idToUpdate = sc.nextInt();
      System.out.print("Enter New Name, Department, Marks: ");
      Student s2 = new Student(idToUpdate, sc.next(), sc.next(), sc.nextFloat());
      controller.updateStudent(s2);
      System.out.println("Student updated.");
      break;
   case 3:
      System.out.print("Enter ID to delete: ");
      controller.deleteStudent(sc.nextInt());
      System.out.println("Student deleted.");
      break;
   case 4:
```

```
                System.out.print("Enter ID to view: ");
                Student fetched = controller.getStudent(sc.nextInt());
                if (fetched != null) System.out.println(fetched);
                else System.out.println("Student not found.");
                break;
            case 5:
                List<Student> all = controller.getAllStudents();
                all.forEach(System.out::println);
                break;
            case 6:
                System.out.println("Exiting...");
                return;
            default:
                System.out.println("Invalid option!");
          }
        }
      } catch (SQLException e) {
        e.printStackTrace();
      }
    }
}
```