Prac 11) Web services

```
student-grade-system/
 ├── src/main/java/com/example/student/
 │   ├── StudentApplication.java
 │   ├── controller/StudentController.java
|   ├── controller/StudentRestController.java
         repository/StudentRepository.java
 │   ├── model/Student.java
 │   ├── service/StudentService.java
 │   └── service/GradeCalculator.java
 ├── src/main/resources/
 │   ├── templates/
 │   │    ├── index.html
 │   │    ├── add.html
 │   │    ├── edit.html
 │   │    └── view.html
 │   └── application.properties
 └── src/test/java/com/example/student/
     └── StudentServiceTests.java
```

StudentApplication.java

```
package com.example.StudentGrade;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class StudentApplication {
    public static void main(String[] args) {
        SpringApplication.run(StudentApplication.class, args);
        System.out.println("Student Grade System running at http://localhost:8080/");
    }
}
```

StudentController.java

```
package com.example.StudentGrade.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;
import com.example.StudentGrade.model.Student;
import com.example.StudentGrade.service.StudentService;

@Controller
public class StudentController {

    @Autowired
    private StudentService service;

    @GetMapping("/")
    public String index(Model model) {
        model.addAttribute("students", service.getAll());
        return "index";
    }
```

```java
    @GetMapping("/add")
    public String addForm(Model model) {
        model.addAttribute("student", new Student());
        return "add";
    }

    @PostMapping("/add")
    public String addStudent(@ModelAttribute Student student) {
        service.add(student);
        return "redirect:/";
    }

    @GetMapping("/edit/{id}")
    public String editForm(@PathVariable int id, Model model) {
        model.addAttribute("student", service.getById(id));
        return "edit";
    }

    @PostMapping("/edit/{id}")
    public String updateStudent(@PathVariable int id, @ModelAttribute Student student) {
        service.update(id, student);
        return "redirect:/";
    }

    @GetMapping("/delete/{id}")
    public String deleteStudent(@PathVariable int id) {
        service.delete(id);
        return "redirect:/";
    }

    @GetMapping("/view/{id}")
    public String viewStudent(@PathVariable int id, Model model) {
        model.addAttribute("student", service.getById(id));
        return "view";
    }
}
```

StudentRestController.java

```java
package com.example.StudentGrade.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import com.example.StudentGrade.model.Student;
import com.example.StudentGrade.service.StudentService;

import java.util.List;

@RestController
@RequestMapping("/api/students")
public class StudentRestController {

    @Autowired
    private StudentService service;

    // Get all students
    @GetMapping
    public List<Student> getAllStudents() {
        return service.getAll();
    }

    // Get single student by ID
    @GetMapping("/{id}")
    public ResponseEntity<Student> getStudent(@PathVariable int id) {
        Student student = service.getById(id);
        if (student == null) return ResponseEntity.notFound().build();
        return ResponseEntity.ok(student);
    }

    // Add new student
    @PostMapping
    public ResponseEntity<Student> addStudent(@RequestBody Student student) {
        service.add(student);
        return ResponseEntity.ok(student);
    }

    // Update existing student
    @PutMapping("/{id}")
    public ResponseEntity<Student> updateStudent(@PathVariable int id, @RequestBody Student student) {
        if (service.getById(id) == null) return ResponseEntity.notFound().build();
        service.update(id, student);
        return ResponseEntity.ok(student);
    }

    // Delete student
    @DeleteMapping("/{id}")
    public ResponseEntity<Void> deleteStudent(@PathVariable int id) {
        if (service.getById(id) == null) return ResponseEntity.notFound().build();
        service.delete(id);
        return ResponseEntity.noContent().build();
    }
}
```

StudentRepository.java

```
 package com.example.StudentGrade.repository;

import org.springframework.data.jpa.repository.JpaRepository;
import com.example.StudentGrade.model.Student;

public interface StudentRepository extends JpaRepository<Student, Integer> {

}
```

Student.java

```
package com.example.StudentGrade.model;

import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;

@Entity
public class Student {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;
    private String name;
    private int marks;
    private String grade;

    public Student() {}

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

    public int getId() { return id; }
    public void setId(int id) { this.id = id; }

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

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

    public String getGrade() { return grade; }
    public void setGrade(String grade) { this.grade = grade; }
}
```

StudentService.java

```
package com.example.StudentGrade.service;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import com.example.StudentGrade.model.Student;
import com.example.StudentGrade.repository.StudentRepository;

import java.util.List;
import java.util.Optional;

@Service
public class StudentService {

    @Autowired
    private StudentRepository repository;

    @Autowired
    private GradeCalculator gradeCalculator;

    public List<Student> getAll() {
        return repository.findAll();
    }

    public Student getById(int id) {
        Optional<Student> student = repository.findById(id);
        return student.orElse(null);
    }
    public Student add(Student s) {
        s.setGrade(gradeCalculator.calculateGrade(s.getMarks()));
        return repository.save(s);
    }

    public Student update(int id, Student s) {
        s.setId(id);
        s.setGrade(gradeCalculator.calculateGrade(s.getMarks()));
        return repository.save(s);
    }

    public void delete(int id) {
        repository.deleteById(id);
    }
}
```

GradeCalculator.java

```
package com.example.StudentGrade.service;

import org.springframework.stereotype.Component;

@Component
public class GradeCalculator {
    public String calculateGrade(int marks) {
        if (marks < 0 || marks > 100) return "Invalid";
        if (marks >= 90) return "A+";
        else if (marks >= 75) return "A";
        else if (marks >= 60) return "B";
        else if (marks >= 40) return "C";
        else return "F";
    }
}
```

index.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <title>Student List</title>
  <style>
    body { font-family: Arial; background: #f8f9fa; padding: 20px; }
    table { border-collapse: collapse; width: 100%; background: white; }
    th, td { border: 1px solid #ddd; padding: 10px; text-align: left; }
    th { background-color: #007bff; color: white; }
    a { text-decoration: none; color: #007bff; }
    .btn { background: #007bff; color: white; padding: 6px 12px; border-radius: 4px; }
  </style>
</head>
<body>
  <h2>Student List</h2>
  <a class="btn" th:href="@{/add}">+ Add Student</a>
  <br><br>
  <table>
    <tr><th>ID</th><th>Name</th><th>Marks</th><th>Grade</th><th>Actions</th></tr>
    <tr th:each="s : ${students}">
      <td th:text="${s.id}"></td>
      <td th:text="${s.name}"></td>
      <td th:text="${s.marks}"></td>
      <td th:text="${s.grade}"></td>
      <td>
        <a th:href="@{'/view/' + ${s.id}}">View</a> |
        <a th:href="@{'/edit/' + ${s.id}}">Edit</a> |
        <a th:href="@{'/delete/' + ${s.id}}">Delete</a>
      </td>
    </tr>
  </table>
</body>
</html>
```

add.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head><title>Add Student</title></head>
<body>
  <h2>Add New Student</h2>
  <form th:action="@{/add}" th:object="${student}" method="post">
    <label>ID:</label><input type="number" th:field="*{id}" required><br>
    <label>Name:</label><input type="text" th:field="*{name}" required><br>
    <label>Marks:</label><input type="number" th:field="*{marks}" required><br>
    <button type="submit">Save</button>
  </form>
  <a th:href="@{/}">Back</a>
</body>
</html>
```

edit.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head><title>Edit Student</title></head>
<body>
  <h2>Edit Student</h2>
  <form th:action="@{'/edit/' + ${student.id}}" th:object="${student}" method="post">
    <label>ID:</label><input type="number" th:field="*{id}" readonly><br>
    <label>Name:</label><input type="text" th:field="*{name}" required><br>
    <label>Marks:</label><input type="number" th:field="*{marks}" required><br>
    <button type="submit">Update</button>
  </form>
  <a th:href="@{/}">Back</a>
</body>
</html>
```

view.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head><title>View Student</title></head>
<body>
  <h2>Student Details</h2>
  <p><strong>ID:</strong> <span th:text="${student.id}"></span></p>
  <p><strong>Name:</strong> <span th:text="${student.name}"></span></p>
  <p><strong>Marks:</strong> <span th:text="${student.marks}"></span></p>
  <p><strong>Grade:</strong> <span th:text="${student.grade}"></span></p>
  <a th:href="@{/}">Back</a>
</body>
</html>
```

application.properties

```
# Server
server.port=8080

# Database Connection
spring.datasource.url=jdbc:mysql://localhost:3306/studentdb
spring.datasource.username=root
spring.datasource.password=root

# JPA (Hibernate)
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect
```