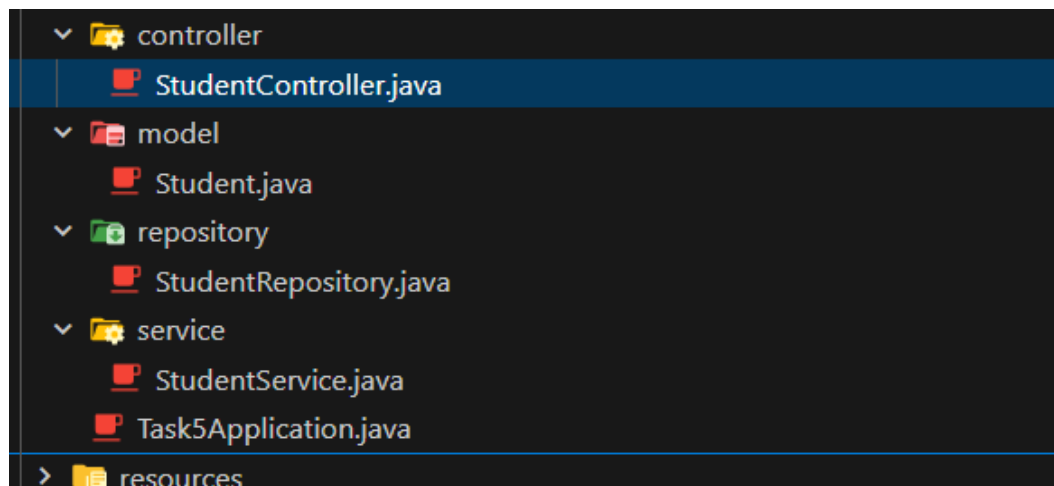
 Marwadi University Marwadi Chandarana Group	Marwadi University Faculty of Engineering and Technology Department of Information and Communication Technology	
Subject: Advance Java (01CT1623)	Aim: Develop REST endpoints to perform the following operations using Spring Boot: Fetch all students, Add new student, Search students by name	
Task : 05	Date: 21/4/2025	Enrolment No: 92200133003

Step 1: Set Up the Spring Boot Project

You can generate the project using [Spring Initializr](#) with the following settings:

- **Project:** Maven
- **Language:** Java
- **Dependencies:**
 - Spring Web
 - Spring Data JPA
 - H2 Database (or MySQL/PostgreSQL if needed)

Step 2: Setup the following File Structure:



Step 3: In the resources folder, add the following configuration to **application.properties**

```
spring.datasource.url=jdbc:h2:mem:testdb
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.username=sa
spring.datasource.password=
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
spring.jpa.hibernate.ddl-auto=update
spring.h2.console.enabled=true
```


Step 4: Add the following code to relevant files:

Student.java (Model)

```
package com.prashant.task5.model;
```

```
import jakarta.persistence.*;
```

```
@Entity
```

 Marwadi University Marwadi Chandarana Group	Marwadi University Faculty of Engineering and Technology Department of Information and Communication Technology	
Subject: Advance Java (01CT1623)	Aim: Develop REST endpoints to perform the following operations using Spring Boot: Fetch all students, Add new student, Search students by name	
Task : 05	Date: 21/4/2025	Enrolment No: 92200133003

```
public class Student {
```

```
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
```

```
    private String name;
```

```
    public Student() {}
```

```
    public Student(String name) {
        this.name = name;
    }
```

```
    // Getters and setters
    public Long getId() {
        return id;
    }
```

```
    public void setId(Long id) {
        this.id = id;
    }
```

```
    public String getName() {
        return name;
    }
```

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

StudentRepository.java


```
package com.prashant.task5.repository;
```

```
import com.prashant.task5.model.Student;
import org.springframework.data.jpa.repository.JpaRepository;
import java.util.List;
```

```
public interface StudentRepository extends JpaRepository<Student, Long> {
    List<Student> findByNameContainingIgnoreCase(String name);
}
```

StudentService.java

```
package com.prashant.task5.service;
```

 Marwadi University Marwadi Chandarana Group	Marwadi University Faculty of Engineering and Technology Department of Information and Communication Technology	
Subject: Advance Java (01CT1623)	Aim: Develop REST endpoints to perform the following operations using Spring Boot: Fetch all students, Add new student, Search students by name	
Task : 05	Date: 21/4/2025	Enrolment No: 92200133003

```
import com.prashant.task5.model.Student;
import com.prashant.task5.repository.StudentRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
```

```
import java.util.List;
```

```
@Service
```

```
public class StudentService {
```

```
    @Autowired
```

```
    private StudentRepository repository;
```

```
    public List<Student> getAllStudents() {
```

```
        return repository.findAll();
```

```
    }
```

```
    public Student addStudent(Student student) {
```

```
        return repository.save(student);
```

```
    }
```

```
    public List<Student> searchStudentsByName(String name) {
```

```
        return repository.findByNameContainingIgnoreCase(name);
```

```
    }
```

```
}
```

StudentController.java

```
package com.prashant.task5.controller;
```

```
import com.prashant.task5.model.Student;
```

```
import com.prashant.task5.service.StudentService;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.web.bind.annotation.*;
```

```
import java.util.List;
```

```
@RestController
```


```
@RequestMapping("/api/students")
```

```
public class StudentController {
```

```
    @Autowired
```

```
    private StudentService service;
```

```
    @GetMapping
```

 Marwadi University Faculty of Engineering and Technology Department of Information and Communication Technology		
Subject: Advance Java (01CT1623)	Aim: Develop REST endpoints to perform the following operations using Spring Boot: Fetch all students, Add new student, Search students by name	
Task : 05	Date: 21/4/2025	Enrolment No: 92200133003

```
public List<Student> getAllStudents() {
    return service.getAllStudents();
}
```

```
@PostMapping
public Student addStudent(@RequestBody Student student) {
    return service.addStudent(student);
}
```

```
@GetMapping("/search")
public List<Student> searchStudents(@RequestParam String name) {
    return service.searchStudentsByName(name);
}
}
```

Output:

POST

http://localhost:8080/api/students

Send

Query

Headers²

Auth

Body¹

Tests

Pre Run

JSON

XML

Text

Form

Form-encode

GraphQL

Binary

JSON Content

Format

```
1 {
2   "name": "Prashant"
3 }
```

Status: 200 OK

Size: 26 Bytes

Time: 298 ms

Response

Headers⁴

Cookies

Results

Docs

```
1 {
2   "id": 1,
3   "name": "Prashant"
4 }
```

GET

http://localhost:8080/api/students

Send

Query

Headers²

Auth

Body

Tests

Pre Run

JSON

XML

Text

Form

Form-encode

GraphQL

Binary

JSON Content

Format

```
1
```

Status: 200 OK

Size: 28 Bytes

Time: 246 ms

Response

Headers⁴

Cookies

Results

Docs

```
1 [
2   {
3     "id": 1,
4     "name": "Prashant"
5   }
6 ]
```

GET

http://localhost:8080/api/students/search?name=Prashant

Send

Query

Headers²

Auth

Body

Tests

Pre Run

JSON

XML

Text

Form

Form-encode

GraphQL

Binary

JSON Content

Format

```
1
```

Status: 200 OK

Size: 28 Bytes

Time: 57 ms

Response

Headers⁴

Cookies

Results

Docs

```
1 [
2   {
3     "id": 1,
4     "name": "Prashant"
5   }
6 ]
```