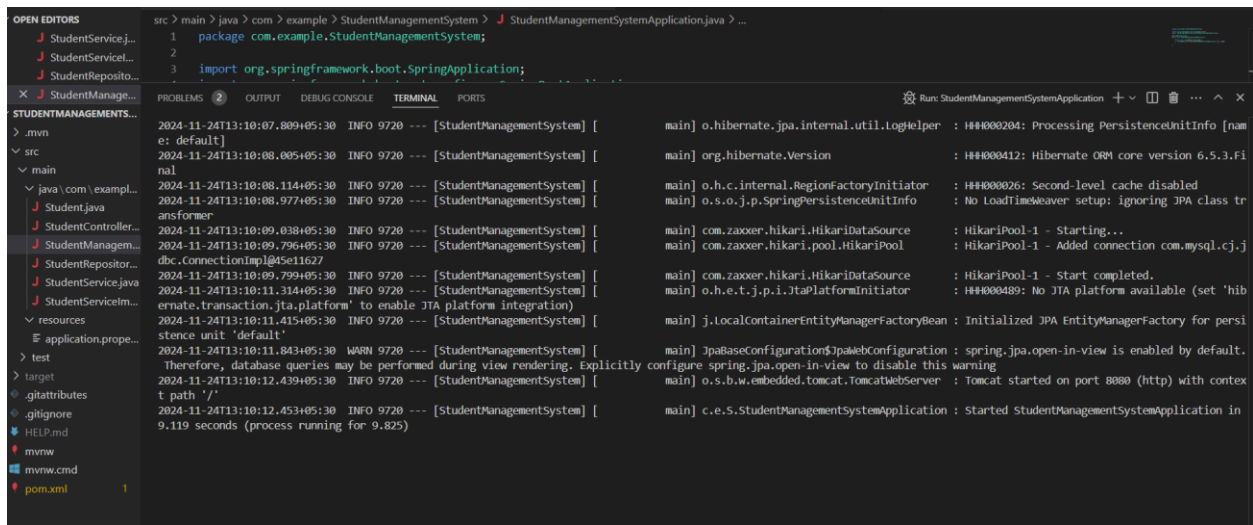


Git hub Link: <https://github.com/SewwandiPeiris/assignment02.git>



Student.java

```
package com.example.StudentManagementSystem;
```

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

```
import lombok.AllArgsConstructor;
```

```
import lombok.Data;
```

```
import lombok.NoArgsConstructor;
```

```
@Entity
```

```
@AllArgsConstructor
```

```
@NoArgsConstructor
```

```
@Data
```

```
@Table(name="")
```

```
public class Student{
```

```
    @Id
```

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

```
    @Column(name = "", nullable=false) private String firstName;
```

```
@Column(name = "", nullable=false) private String lastName;

@Column(name = "", nullable=false) private String email;

@Column(name = "", nullable=false) private String department;

@Column(name = "", nullable=false) private String yearOfEnrollment;

}
```

StudentController.java

```
package com.example.StudentManagementSystem;

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

import java.util.List;

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

    @Autowired
    private StudentService studentService;

    @PostMapping
    public ResponseEntity<Student> saveStudent(@RequestBody Student student){
        return new ResponseEntity<Student>
(studentService.saveStudent(student),HttpStatus.CREATED);
    }

    @GetMapping
    public List<Student> getAllStudents(){
        return studentService.getAllStudents();
    }
}
```

```
}

@GetMapping("/{id}")
public ResponseEntity<Student> getStudentById(@PathVariable("id") long studentID){
    return new ResponseEntity<Student>
(studentService.getStudentById(studentID),HttpStatus.OK);
}

@PutMapping("/{id}")
public ResponseEntity<Student> updateStudent(@PathVariable("id") long id, @RequestBody
Student student){
    return new ResponseEntity<>(studentService.updateStudent(student,id),HttpStatus.OK);
}

@DeleteMapping("/{id}")
public ResponseEntity<String> deleteStudent(@PathVariable("id") long id){
    studentService.deleteStudent(id);
    return new ResponseEntity<String>("Student deleted Successfully.",HttpStatus.OK);
}

}
```

StudentManagementSystemApplication.java

```
package com.example.StudentManagementSystem;

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

@SpringBootApplication
```

```
public class StudentManagementSystemApplication {  
  
    public static void main(String[] args) {  
        SpringApplication.run(StudentManagementSystemApplication.class, args);  
    }  
  
}
```

StudentRepository.java

```
package com.example.StudentManagementSystem;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
public interface StudentRepository extends JpaRepository<Student, Long>{  
    //crud operations  
}
```

StudentService.java

```
package com.example.StudentManagementSystem;  
  
import java.util.List;  
  
import org.springframework.stereotype.Service;  
  
@Service  
public interface StudentService {  
    Student saveStudent(Student student);  
}
```

```
List<Student> getAllStudents();  
  
Student getStudentById(long id);  
  
Student updateStudent(Student student,long id);  
  
void deleteStudent(long id);  
  
}
```

StudentServiceImpl.java

```
package com.example.StudentManagementSystem;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
import java.util.List;  
import java.util.Optional;  
  
@Service  
public class StudentServiceImpl implements StudentService{  
    @Autowired  
    private StudentRepository studentRepository;  
  
    @Override  
    public Student saveStudent(Student student){  
        return studentRepository.save(student);  
    }  
  
    @Override  
    public List<Student> getAllStudents() {  
        return studentRepository.findAll();  
    }  
}
```

@Override

```
public Student getStudentById(long id) {  
    Optional<Student> student = studentRepository.findById(id);  
    if(student.isPresent()){  
        return student.get();  
    }else {  
        throw new RuntimeException();  
    }  
}
```

@Override

```
public Student updateStudent(Student student, long id) {  
    Student existingStudent =studentRepository.findById(id).orElseThrow(()-> new  
RuntimeException());  
    existingStudent.setFirstName(student.getFirstName());  
    existingStudent.setLastName(student.getLastName());  
    existingStudent.setEmail(student.getEmail());  
  
    studentRepository.save(existingStudent);  
    return existingStudent;  
}
```

@Override

```
public void deleteStudent(long id){  
    studentRepository.findById(id).orElseThrow(()-> new RuntimeException());  
    studentRepository.deleteById(id);  
}
```

```
}
```

application.properties

```
spring.application.name=StudentManagementSystem
```

```
spring.jpa.hibernate.ddl-auto=update
```

```
spring.datasource.url=
```

```
jdbc:mysql://localhost:3306/student_management?createDatabaseIfNotExist=true
```

```
spring.datasource.username=root
```

```
spring.datasource.password=1234
```

```
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
```

```
spring.jpa.show-sql=true
```