

```
=====
SPRING BOOT MICROSERVICES PROGRAMS
=====
```

```
----- STUDENT MICROSERVICE -----
```

```
StudentEntity.java
```

```
-----
@Entity
@Table(name="students")
public class StudentEntity {
    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private Long regId;
    private String rollNo;
    private String name;
    private String dept;
    private String section;
    private String aadharId;
    private String skills;
    private Double cgpa;
}
```

```
StudentRepository.java
```

```
-----
public interface StudentRepository extends JpaRepository<StudentEntity, Long> {}
```

```
StudentService.java
```

```
-----
public interface StudentService {
    StudentEntity createStudent(StudentEntity s);
    List<StudentEntity> getAllStudents();
    StudentEntity getStudentById(Long id);
    StudentEntity updateStudent(Long id, StudentEntity s);
    void deleteStudent(Long id);
}
```

```
StudentServiceImpl.java
```

```
-----
@Service
public class StudentServiceImpl implements StudentService {
    @Autowired private StudentRepository repo;
    public StudentEntity createStudent(StudentEntity s){ return repo.save(s); }
    public List<StudentEntity> getAllStudents(){ return repo.findAll(); }
    public StudentEntity getStudentById(Long id){ return repo.findById(id).orElseThrow(); }
    public StudentEntity updateStudent(Long id, StudentEntity s){
        StudentEntity e = getStudentById(id);
        e.setName(s.getName()); e.setDept(s.getDept()); return repo.save(e);
    }
    public void deleteStudent(Long id){ repo.deleteById(id); }
}
```

```
StudentController.java
```

```
-----
@RestController
@RequestMapping("/api/students")
public class StudentController {
    @Autowired private StudentService service;
    @PostMapping public StudentEntity create(@RequestBody StudentEntity s){ return service.createStudent(s); }
    @GetMapping public List<StudentEntity> all(){ return service.getAllStudents(); }
    @GetMapping("/{id}") public StudentEntity get(@PathVariable Long id){ return service.getStudentById(id); }
    @PutMapping("/{id}") public StudentEntity update(@PathVariable Long id,@RequestBody StudentEntity s){ return service.updateStudent(id,s); }
    @DeleteMapping("/{id}") public void delete(@PathVariable Long id){ service.deleteStudent(id); }
}
```

```
application.properties
```

```
-----
spring.datasource.url=jdbc:mysql://localhost:3306/college_db
spring.datasource.username=root
spring.datasource.password=your_password
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
```

----- FACULTY MICROSERVICE -----

FacultyEntity.java

```
-----
@Entity
@Table(name="faculties")
public class FacultyEntity {
    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private Long facultyId;
    private String name;
    private String dept;
    private String designation;
    private String email;
    private String phoneNo;
}
```

FacultyRepository.java

```
-----
public interface FacultyRepository extends JpaRepository<FacultyEntity, Long> {}
```

FacultyService.java / FacultyController.java

```
-----
(Same structure as Student Microservice)
```

----- ROOM MICROSERVICE -----

RoomEntity.java

```
-----
@Entity
@Table(name="rooms")
public class RoomEntity {
    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private Long roomId;
    private String dept;
    private Integer capacity;
    private String buildingName;
    private String block;
}
```

RoomRepository.java

```
-----
public interface RoomRepository extends JpaRepository<RoomEntity, Long> {}
```

RoomService.java / RoomController.java

```
-----
(Same CRUD structure as Student Microservice)
```

MAVEN DEPENDENCIES (pom.xml)

```
-----
<dependencies>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-j</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-validation</artifactId>
    </dependency>
</dependencies>
```

