■ Question 1: Student Microservice

■ Objective

To build a Spring Boot microservice named college-mgmt-student-service that performs CRUD operations on StudentEntity and persists data in MySQL.

■■ Entity Class

```
@Entity
@Table(name = "students")
public class StudentEntity {

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

    @Column(nullable = false, unique = true)
    private String rollNo;

    @NotBlank
    private String name;

    private String dept;
    private String section;

    @NotBlank
    @Size(min = 12, max = 12)
    private String aadharId;

    private String skills;

    @DecimalMin("0.0")
    @DecimalMax("10.0")
    private Double cgpa;

    // Getters and Setters
}
```

■ Repository Layer

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

■■ Service Layer

```
@Service
public class StudentService {

    @Autowired
    private StudentRepository repository;

    public StudentEntity createStudent(StudentEntity student) {
        return repository.save(student);
    }

    public List<StudentEntity> getAllStudents() {
        return repository.findAll();
    }

    public StudentEntity getStudentById(Long id) {
        return repository.findById(id)
                .orElseThrow(() -> new RuntimeException("Student not found with id " + id));
    }

    public StudentEntity updateStudent(Long id, StudentEntity student) {
        StudentEntity existing = getStudentById(id);
        existing.setName(student.getName());
        existing.setDept(student.getDept());
        existing.setSection(student.getSection());
        existing.setCgpa(student.getCgpa());
        existing.setSkills(student.getSkills());
        return repository.save(existing);
    }

    public void deleteStudent(Long id) {
        repository.deleteById(id);
    }
}
```

■ Controller Layer

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

    @Autowired
    private StudentService service;

    @PostMapping
    public ResponseEntity<StudentEntity> createStudent(@Valid @RequestBody StudentEntity student) {
        return new ResponseEntity<>(service.createStudent(student), HttpStatus.CREATED);
    }

    @GetMapping
    public List<StudentEntity> getAllStudents() {
        return service.getAllStudents();
    }

    @GetMapping("/{id}")
    public ResponseEntity<StudentEntity> getStudentById(@PathVariable Long id) {
        return ResponseEntity.ok(service.getStudentById(id));
    }

    @PutMapping("/{id}")
    public ResponseEntity<StudentEntity> updateStudent(@PathVariable Long id, @RequestBody StudentEnti
        return ResponseEntity.ok(service.updateStudent(id, student));
    }

    @DeleteMapping("/{id}")
    public ResponseEntity<Void> deleteStudent(@PathVariable Long id) {
        service.deleteStudent(id);
        return ResponseEntity.noContent().build();
    }
}
```

■■ MySQL Configuration (application.properties)

```
spring.datasource.url=jdbc:mysql://localhost:3306/college_db
spring.datasource.username=root
spring.datasource.password=admin123
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect
```

■ Output Example

```
POST /api/students        → Creates new student
GET  /api/students        → Fetch all students
GET  /api/students/{id}   → Fetch one student
PUT  /api/students/{id}   → Update student
DELETE /api/students/{id} → Delete student
```