

## ■■■ Question 2: Faculty Microservice

### ■ Objective

To build a Spring Boot microservice named college-mgmt-faculty-service for managing faculty details.

### ■■ Entity Class

```
@Entity
@Table(name = "faculties")
public class FacultyEntity {

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

    @NotBlank
    private String name;

    private String dept;
    private String designation;

    @Email
    private String email;

    @Pattern(regexp = "[0-9]{10}$", message = "Phone number must be 10 digits")
    private String phoneNo;

    // Getters and Setters
}
```

### ■ Repository

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

### ■■ Service

```
@Service
public class FacultyService {

    @Autowired
    private FacultyRepository repo;

    public FacultyEntity createFaculty(FacultyEntity faculty) {
        return repo.save(faculty);
    }

    public List<FacultyEntity> getAllFaculties() {
        return repo.findAll();
    }

    public FacultyEntity getFacultyById(Long id) {
        return repo.findById(id)
    }
}
```

```

        .orElseThrow(() -> new RuntimeException("Faculty not found"));
    }

    public FacultyEntity updateFaculty(Long id, FacultyEntity faculty) {
        FacultyEntity existing = getFacultyById(id);
        existing.setName(faculty.getName());
        existing.setDept(faculty.getDept());
        existing.setDesignation(faculty.getDesignation());
        existing.setEmail(faculty.getEmail());
        existing.setPhoneNo(faculty.getPhoneNo());
        return repo.save(existing);
    }

    public void deleteFaculty(Long id) {
        repo.deleteById(id);
    }
}

```

## ■ Controller

```

@RestController
@RequestMapping("/api/faculties")
public class FacultyController {

    @Autowired
    private FacultyService service;

    @PostMapping
    public ResponseEntity<FacultyEntity> createFaculty(@Valid @RequestBody FacultyEntity faculty) {
        return new ResponseEntity<>(service.createFaculty(faculty), HttpStatus.CREATED);
    }

    @GetMapping
    public List<FacultyEntity> getAllFaculties() {
        return service.getAllFaculties();
    }

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

    @PutMapping("/{id}")
    public ResponseEntity<FacultyEntity> updateFaculty(@PathVariable Long id, @RequestBody FacultyEntity faculty) {
        return ResponseEntity.ok(service.updateFaculty(id, faculty));
    }

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

```