

## ■ Question 3: Rooms Microservice

### ■ Objective

Develop a microservice named college-mgmt-room-service to manage room details.

### ■■ Entity

```
@Entity
@Table(name = "rooms")
public class RoomEntity {

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

    @NotBlank
    private String dept;

    @NotNull
    @Min(10)
    private Integer capacity;

    private String buildingName;
    private String block;

    // Getters and Setters
}
```

### ■ Repository

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

### ■■ Service

```
@Service
public class RoomService {

    @Autowired
    private RoomRepository repo;

    public RoomEntity createRoom(RoomEntity room) {
        return repo.save(room);
    }

    public List<RoomEntity> getAllRooms() {
        return repo.findAll();
    }

    public RoomEntity getRoomById(Long id) {
        return repo.findById(id)
            .orElseThrow(() -> new RuntimeException("Room not found"));
    }
}
```

```

    public RoomEntity updateRoom(Long id, RoomEntity room) {
        RoomEntity existing = getRoomById(id);
        existing.setDept(room.getDept());
        existing.setCapacity(room.getCapacity());
        existing.setBuildingName(room.getBuildingName());
        existing.setBlock(room.getBlock());
        return repo.save(existing);
    }

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

```

## ■ Controller

```

@RestController
@RequestMapping("/api/rooms")
public class RoomController {

    @Autowired
    private RoomService service;

    @PostMapping
    public ResponseEntity<RoomEntity> createRoom(@Valid @RequestBody RoomEntity room) {
        return new ResponseEntity<>(service.createRoom(room), HttpStatus.CREATED);
    }

    @GetMapping
    public List<RoomEntity> getAllRooms() {
        return service.getAllRooms();
    }

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

    @PutMapping("/{id}")
    public ResponseEntity<RoomEntity> updateRoom(@PathVariable Long id, @RequestBody RoomEntity room) {
        return ResponseEntity.ok(service.updateRoom(id, room));
    }

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

```