Backend Task: Employee Management Module

Objective

Develop a RESTful backend service to manage employees, departments, and reporting managers. The system must enforce referential integrity, support relational queries (joins), and include middleware for request–response logging.

Entities (Relational Model)

Employee

- id (PK)
- first_name
- last_name
- email (unique)
- department_id (FK → Department.id)
- manager_id (FK → Employee.id, nullable)
- created_at
- updated_at

Department (Master Table)

- id (PK)
- name (unique)

Business Rules & Constraints

1. Fetch employee details along with department name, and manager name using joins.
2. Deleting an employee who is a manager should set manager_id = NULL for subordinates.
3. Deleting a department is not allowed if employees are mapped to it.
4. Enforce foreign keys and prevent orphan records.

Essential API Endpoints

Employee APIs

POST /api/employees
GET /api/employees
GET /api/employees/{id}
PUT /api/employees/{id}
DELETE /api/employees/{id}

Department APIs

POST /api/departments
GET /api/departments
DELETE /api/departments/{id} (blocked if employees exist)

Middleware Requirement

Implement global request–response logging middleware that logs:

- HTTP method
- Endpoint
- Response status code
- Response time

Technical Expectations

- Use any backend stack
- Relational database (PostgreSQL / MySQL/ SQLServer)
- Proper error handling and HTTP status codes
- Comments for every endpoint
- Clean project structure
- Use of an ORM for database access is mandatory