

Employee Management System

Technical Requirements

1. Develop the API in **.NET Core** following a **multi-tiered architecture**. ✓
 2. Use the **Entity Framework Code-First** approach (optional but recommended). ✓
 3. Provide a **compilable and functioning** project.
 4. Ensure proper **security and authentication** for employees and admins. ✓
 5. If any functionality is **not implemented**, clearly document the exclusions.
-

Functional Requirements

1. User Roles: ✓

The system will have two main user roles:

- **Employee**
- **Admin**

2. Employee Details: ✓

Each employee should have the following details:

- **Employee ID (Unique Identifier)**
- **First Name**
- **Last Name**
- **Email**
- **Phone Number**
- **Department**
- **Tech Stack** (Skills & Technologies known)
- **Date of Birth (Optional)**
- **Address (Optional)**

3. Timesheet Management: ✓

- Employees should be able to **log their working hours** daily in a timesheet.
- Each timesheet entry should include:

- **Date**
- **Start Time**
- **End Time**
- **Total Hours Worked**
- **Description of Work Done (Optional)**
- Employees can **view and edit** their timesheet entries.

4. Admin Functionalities: ✓

- Admins should be able to **log in using email and password**.
- Admins should have access to **view all employees** and their timesheets.
- Admins should be able to **export timesheets** to **Excel** for reporting.
- Admins can manage employee profiles (edit details, activate/deactivate accounts).

5. Profile Page: ✓

- Employees should have a profile page displaying their **basic details**.
- Employees should be able to **update their phone number, tech stack, and address**.
- Password reset functionality should be available for employees.

6. Leave Management (Additional Feature) ✓

- Employees should be able to **apply for leave** by selecting:
 - **Start Date & End Date**
 - **Type of Leave** (Sick Leave, Casual Leave, Vacation, etc.)
 - **Reason (Optional)**

7. Reports & Analytics (Additional Feature) ✓

- Admins should be able to generate **reports** for:
 - **Employee Work Hours** (weekly, monthly)

8. Authentication & Security: ✓

- Employees and Admins should log in using **email & password**.
 - Passwords should be stored **securely** using **hashing**.
 - Users should have the ability to **reset their passwords** via email verification.
-

Database Relationships

1. **One-to-Many:** Department (1) → (M) Employee ✓
 - One department can have multiple employees.
2. **One-to-Many:** Employee (1) → (M) Timesheet ✓
 - One employee can log multiple timesheet entries.
3. **Independent Entity:** Admin has no direct relationship with other tables. ✓

Additional Considerations:

- A **dashboard** should be available where: ✓
 - Employees can see their **total logged hours**, leave balance, and latest timesheet entries.
 - Admins can see an **overview of employee activities**, pending leave requests, and analytics.
- Use **JWT authentication** for secure API access. ✓
- Follow **RESTful API principles** and use proper **HTTP status codes**. ✓

