

# DpWorld Employee Management App

## Tech Stack:

- **Frontend:** Angular 17, Angular Material
  - **Backend:** ASP.NET Core 8, Entity Framework Core, ASP.NET Identity
  - **Authentication:** JWT (JSON Web Token)
  - **Database:** (e.g., SQL Server, SQLite – specify your actual DB)
- 

## 1. Overview

This web application allows company admins to manage employees and track attendance, while employees can check in daily and view their profiles and attendance history. The system enforces role-based access and secure authentication.

## 2. Features

### Admin

- Add, edit, and delete employees (with validation)
- View paginated, sortable, and filterable employee list
- Track daily and weekly attendance for all employees
- View employee signatures

### Employee

- View personal profile and weekly attendance summary
  - Daily check-in (only once per day, between 7:30 AM and 9:00 AM)
  - Upload or draw a signature (if not already set)
  - View check-in history
- 

## 3. Architecture

### Frontend (Angular)

- **Modules:** Auth, Admin, Employee (lazy-loaded)
- **Routing:** Role-based guards, JWT-protected routes
- **UI:** Angular Material for responsive design
- **State:** Services for API communication, localStorage for JWT
- **Forms:** Reactive Forms for validation
- **HTTP:** Interceptors attach JWT to all API requests

## Backend (ASP.NET Core)

- **Controllers:** Separate for Admin and Employee
  - **Services:** UserService, AttendanceService (DI via interfaces)
  - **Authentication:** ASP.NET Identity, JWT tokens
  - **Database:** Stores users, attendance, and signatures (as base64)
  - **Validation:** DTOs and model validation
  - **API Docs:** Swagger/Swashbuckle enabled
- 

## 4. Database Schema (Simplified)

- **Users**
  - Id (PK)
  - FirstName, LastName, Email, UserName, PhoneNumber, NationalId, Age, PasswordHash, Role, SignatureUrl
- **Attendance**
  - Id (PK)
  - UserId (FK)
  - CheckInTime (DateTime)

## 5. Setup & Usage

### Backend

#### 1. Install dependencies:

`dotnet restore`

#### 2. Update DB (if needed):

`dotnet ef database update`

#### 3. Run the server:

`dotnet run`

#### 4. API available at:

`https://localhost:7029/api/`

### Frontend

#### 1. Install dependencies:

`npm install`

#### 2. Run Angular dev server (with proxy):

`ng serve --proxy-config proxy.conf.json`

#### 3. App available at:

`http://localhost:53142/`

## 6. Security

- JWT-based authentication for all API calls
  - Role-based route guards (Admin/Employee)
  - Passwords hashed via ASP.NET Identity
  - CORS and HTTPS enforced
- 

## 7. Usage Guide

- **Admin Login:** Use credentials provided by the system or seeded in the DB.
- **Add Employee:** Fill out all required fields, including email and password.
- **Employee Login:** Use credentials provided by admin.
- **Check-In:** Employees can check in once per day, only between 7:30 AM and 9:00 AM.
- **Signature:** Employees can upload a signature image if not already set.