.NET Core Developer – ERP Task (Mid-Level)

Thank you for applying to our .NET Core / MVC Developer position.

To proceed with your application, we'd like to invite you to complete the following technical task.

Task Objective:

Build a backend module for an ERP-style Employee Management System using ASP.NET Core Web API (Executed in .NET Core only).The task should demonstrate clean architecture, strong database design, use of design patterns, and structured business logic.

Task Requirements:

System Scope:

-Design and implement a backend API that supports the following:

-Employee Management:

-Add, edit, delete, and retrieve employees

-Each employee should include: ID, Name, Email, Department, Hire Date, Status (Active/Suspended)

Department Management:

-Add and retrieve departments

-Every employee must belong to a department

Log History Tracking:

-Record every action performed on employees (create/update/delete) in a LogHistory table with timestamp

-Show this log through a separate endpoint

Validation and Error Handling:

-Add server-side validation (Data Annotations or FluentValidation)

-Return clear error messages for invalid inputs

Data Filtering + Sorting (Advanced):

-Endpoint to get employees with the following optional filters:

-By name, department, status

-By hire date range

Sorting by name or hire date (ascending/descending)

Architecture Requirements:

-Use Clean Architecture or Layered Structure (Controllers, Services, Repositories, Models)

-Use Entity Framework Core (Code First) with migrations Include Entity configurations using Fluent API or Data Annotations

-Implement Repository Pattern using Interfaces to abstract data access

-Map with AutoMapper (optional but preferred)

-Apply Dependency Injection properly

Optional Bonus (for higher assessment):

Implement Pagination for the employee list

Use Unit of Work pattern

Include Swagger for API testing

Deliverables:

-The candidate is expected to submit:

-A GitHub repo or ZIP file of the complete solution

-SQL Migration file or SQL Script to generate the database schema (Code First approach)

A README.md file that includes:

-Setup instructions (how to run the project)

-Explanation of folder structure and layers

-Description of filters, sorting, and log implementation

-Any assumptions made during the implementation

-Sample API requests or Postman collection (if available)

Deadline:

Please complete and submit the task within 48 hours of receiving this email..