Exercise 1: Minimal APIs and Dependency Injection

Task:

Create a minimal API using ASP.NET Core and implement dependency injection to manage services.

Requirements:

- Create a minimal API with at least one endpoint.
- Implement a service that performs a simple operation (e.g., addition, subtraction).
- Use dependency injection to inject the service into the API controller.
- The endpoint should accept input, utilize the service to perform the operation, and return the result.

Exercise 2: Relational Databases (SQL Server) and Design Patterns

Task:

Develop a .NET application that interacts with a SQL Server database and implement a design pattern.

Requirements:

- Set up a SQL Server database with a table named Employees having columns: EmployeeId, FirstName, LastName, Salary.
- Create a .NET application that connects to the database using Entity Framework (or any other ORM).
- Implement a repository pattern to perform CRUD operations on the Employees table.
- Create an API endpoint to demonstrate the functionality of your repository (e.g., add, update, retrieve employees).

General Guidelines:

- Documentation: Include clear comments in your code.
- Code Quality: Follow best coding practices and maintain a clean code structure.
- Error Handling: Implement proper error handling for critical sections of your code.
- Submission: Share your solution via a compressed folder containing the code files.
- Readme File: Include a readme file explaining how to run your application, any assumptions made, and any additional information you think is relevant.