

Task Management System Assessment

Target: Fresh .NET Developers / Interns

1. Assessment Overview

You are required to build a simple **Task Management System** where users can **Create, Read, Update, and Delete (CRUD)** tasks.

The system must follow:

- **N-tier architecture**
- **Repository Pattern**
- **database operations** using EF Core

Objective: Evaluate your understanding of EF Core, project layering, repository pattern, and clean code principles.

2. Requirements

2.1 Task Entity

Create a class `TaskItem` with these properties:

Property	Type	Notes
Id	int	Primary Key
Title	string	Max length 100
Description	string	Max length 500
IsCompleted	bool	Status of the task
CreatedAt	DateTime	Default: current date

DueDate DateTime Optional

2.2 Database Configuration

- Use **Entity Framework Core**
 - Configure entity using for:
 - Max length of `Title(100)` and `Description(500)`
 - Default value for `CreatedAt`
-

2.3 CRUD Operations

Your system should allow:

1. **Create:** Add a new task
 2. **Read:** List all tasks & get task by ID
 3. **Update:** Modify task details
 4. **Delete:** Remove a task
-

2.4 Repository Pattern

1. Define interface `ITaskRepository` with methods for all CRUD operations.
 2. Implement the interface in `TaskRepository`.
 3. Ensure database operations
-

2.5 N-tier Architecture

Structure your project in layers:

1. **Core/Domain Layer:** Entities and repository interfaces
2. **Infrastructure Layer:** DbContext, EF configurations, repository implementations
3. **Web/UI Layer:** Controllers (API or MVC)

Optional: Add a service/application layer for business logic.

2.6 Dependency Injection

- Register your repository interface and implementation in DI container.
 - Inject repository into controllers via constructor.
-

3. Bonus Tasks (Optional)

- Filter tasks by `IsCompleted` status
 - Sort tasks by `DueDate`
 - Implement simple unit tests for repository methods
-

4. Submission Requirements

- Source code of the project
- EF Core migrations applied
- Brief documentation explaining:
 - Each layer in your project

- How the repository is implemented
- How CRUD operations work

