

Senior .NET Developer Assessment

Project: Task & Team Management System

Objective: Build a mini Task & Team Management System with user roles and task workflows. The system will allow managers to assign tasks to users and track status.

Time Limit: 4 hours (strict)

Core Requirements

1. Entities

- **User:** Id, FullName, Email, Role (Admin / Manager / Employee)
- **Team:** Id, Name, Description
- **Task:** Id, Title, Description, Status (Todo / InProgress / Done), AssignedToUserId, CreatedByUserId, TeamId, DueDate

2. API Features

- CRUD operations for Users, Teams, and Tasks
- Search and filter tasks by Status, AssignedTo, Team, and DueDate
- Support pagination and sorting
- Admins can manage users and teams (create, update, delete)
- Managers can create tasks and update any task details
- Employees can view and update status of their assigned tasks

3. Authentication & Authorization

- Implement JWT-based authentication
- Seed default users:
 - o Admin: admin@demo.com / Admin123!
 - o Manager: manager@demo.com / Manager123!
 - o Employee: employee@demo.com / Employee123!
- Enforce role-based access control

4. Technology Stack (must use)

- .NET Core Web API (latest LTS)
- Entity Framework Core
- Relational Database (any)
- Swagger / OpenAPI
- Centralized Logging in file system (any logging library)
- Unit testing (any framework)
- FluentValidation (or similar) for request validation
- Implement CQRS (Command Query Responsibility Segregation)
- Global exception handling via middleware

Bonus Requirements

1. **Docker** (containerization)
2. **Event-Driven Messaging with Message Broker**
 - Publish events to a queue when tasks are created or updated
 - Implement a basic consumer that listens the queue
 - Write logs of these events as soon as message is consumed
1. **Real-time Notifications (SignalR or any Web-socket Implementation)**
 - Simulate notifications when tasks are assigned or completed
 - Save notifications into a **Notifications** table
 - Provide an endpoint to retrieve user-specific notifications (optional)

Submission Guidelines

1. Host solution in a **public GitHub** repository
2. Include a **README.md** with:
 - Setup and run instructions