Assessment Overview: Users & Tasks API

Welcome! This assessment is designed to evaluate your ability to build a structured, well-documented, and secure API. The task is straightforward but offers room to demonstrate your coding style, best practices, and problem-solving skills.

We're not just looking for a working solution—we want to see how you make it great! 🎉

📌 Part 1: API Development

1. Create a new **ASP.NET Core Web API** project.
2. Define the following models:
   - **User**: `ID`, `Username`, `Email`, `Password`
   - **Task**: `ID`, `Title`, `Description`, `Assignee (UserID)`, `DueDate`
3. Implement **CRUD** operations for both users and tasks using RESTful principles.

🔐 Part 2: Authentication

Choose **either** API Key authentication **or** Bearer Token authentication:

- **API Key Authentication**

  - Generate a **unique API key** for each user.
  - Require the API key for accessing endpoints.

- **Bearer Token Authentication (JWT)**

  - Implement **JWT (JSON Web Tokens)** authentication.
  - Generate and validate JWT tokens for users.

🛢️ Part 3: Database Interaction

1. Choose **Entity Framework Core** or **Dapper** for database access.
2. Set up a **local database** (SQL Server, SQLite, etc.) to store users and tasks.

🔍 Part 4: Filtering

Add filtering options to the **Tasks** endpoint:

- Get all **expired** tasks.
- Get all **active** tasks.
- Get all tasks from a **specific date**.

🧪 Part 5: Testing

1. Write **unit tests** for the REST endpoints and repository methods.
2. Use a testing framework like **MSTest, NUnit, or xUnit**.

📖 Part 6: Swagger Documentation

1. Integrate **Swagger** to generate API documentation.
2. Ensure all API endpoints are properly documented and easy to understand.

📤 Part 7: Submission

Submit your completed project as a **public GitHub repository**.
⛔ **No ZIP files or Google Drive links will be accepted.**

✅ What the reviewer will do:

They will clone your repository, build the solution, with the following commands:

```
git clone https://github.com/[your-username]/[your-repository-name].git
cd [your-repository-name]
dotnet build
dotnet test
dotnet run
```

⛔ **If any of these steps fail, the assignment won't be reviewed.**

💡 Final Thoughts

This is a basic assignment, but we encourage you to go beyond the requirements!

Write clean, maintainable code. Follow best practices in API design and security. Show off your skills—add extra features if you like! We're excited to see your approach. Good luck! 🚀