Test task

We are inviting you to assess your knowledge and solve the task below.

Please **pay attention to the requirements** for the technologies to be used and code quality. Try to stick to the plan, however you have room for creativity

The completed test task must be uploaded to the GitHub.

We wish you every success!

### MANDATORY REQUIREMENTS

The code of the completed **task should be**:

* Well structured
* Easy to read
* Contain the necessary comments

The program you have written **must be a complete program product**, i.e. should be easy to install, provide for the handling of non-standard situations, be resistant to incorrect user actions, etc.

### TECHNOLOGY REQUIREMENTS

Tasks should be completed:

- On .Net or .Net Core (EntityFramework, MVC / Web API)

- With any DBMS (MS SQL Server, PostgreSQL)

- With Swagger (<https://swagger.io>)

- Uploaded to the GitHub

- Using English to write comments and descriptions of classes, fields, etc

Non-functional requirements:

- Three-level project architecture (data access level, logic level, representation)

- When using third-party frameworks and packages - they must be publicly available

- It is recommended to use unit tests for debugging the logic level

### REQUIREMENTS FOR PRESENTATION FORMAT

The completed task should be uploaded to GitHub and include:

1. **Mandatory:** provide all source files along with the project files

2. If necessary, provide a description of the system configuration, startup process in the explanatory note

### WILL BE A PLUS

- README.md added

- Applying patterns

- Deploy and provide public access to the application

- Using Docker

- Textual description of the product, technologies and templates used, instructions for use

TIME TO PERFORM THE TEST TASK

Approximate completion time for a developer of the corresponding level can take up to **8-16 hours**.

# Task: Implement Web-API for entering project data into the database (task tracker)

You need to implement task storage by projects. “Task” – is an instance which contains at least 3 fields listed below:

1. Id

2. Task name

3. Task description

Solution should provide an ability to easily add new fields in Task entity.

Each task should be a part of only one project. Project – is an instance which contains name, id and code (and also keep Tasks entities).

The program must be a Web-API.

Functional requirements:

- Ability to create / view / edit / delete information about projects

- Ability to create / view / edit / delete task information

- Ability to add and remove tasks from a project (one project can contain several tasks)

- Ability to view all tasks in the project

- To view projects, have to provide various filtering methods (by start date range, by priority, etc.) and sorting (by main fields)

Project information required for storage:

- the name of the project

- project start date

- project completion date

- the current status of the project (enum: NotStarted, Active, Completed)

- priority (int)

Task information required for storage:

- task name

- task status (enum: ToDo / InProgress / Done)

- description

- priority (int)