Interview Home Work Assignment

Summary

Implement a system that can store or retrieve a list of power plants.

Use best practices and coding standards as you think best.

Functional Requirements

1. A power plant should be defined by the following fields

Name	Type	Is Required	Example
Owner	text	true	Vardenis Pavardenis
Power	number (decimal)	true	9.3
Valid From	date	true	2020-01-01
Valid To	date	false	2025-01-01

2. Create a GET API endpoint that retrieves all stored power plants as a JSON response. Example of an expected response:

   Response Example

```
{
    "powerPlants": [
        {
            "owner": "Vardenis Pavardenis",
            "power": 9.3,
            "validFrom": "2020-01-01",
            "validTo": "2025-01-01"
        },
        {
            "owner": "Jonas Jonaitis",
            "power": 5.7,
            "validFrom": "2021-06-15",
            "validTo": "2026-06-15"
        },
        {
            "owner": "Ona Petraitė",
            "power": 12.5,
            "validFrom": "2019-09-10",
            "validTo": null
        }
    ]
}
```

3. Create a POST API endpoint that adds a new power plant to the stored list. Expected JSON request:
    1. **Given** a required field is missing
       When a POST request is submitted
       Then a Bad Request (HTTP status code 400) response should be returned by the API
    2. **Given** power value is less than 0 or greater than 200
       When a POST request is submitted
       Then a Bad Request (HTTP status code 400) response should be returned by the API
    3. **Given** "owner" value does not consist of two words (text-only characters) separated by a whitespace
       When a POST request is submitted
       Then a Bad Request (HTTP status code 400) response should be returned by the API
    4. **Given** all power plant data is valid
       When a POST request is submitted
       Then a Created (HTTP status code 201) response should be returned by the API, expected response:

       Request Example

       ```
       {
           "owner": "Ona Petraitė",
           "power": 12.5,
           "validFrom": "2019-09-10",
           "validTo": null
       }
       ```
4. Filtering support for GET endpoint
    1. **Given** a query parameter "owner" is provided (e.g. "?owner=ona")
       When a GET request is submitted
       Then only power plants whose "owner" field contains the specified parameter value should be returned

Non-Functional Requirements

1. Use the latest stable version of .NET Download .NET (Linux, macOS, and Windows) | .NET
2. Code should be stored on a git repository reachable by a shared URL, e.g. https://github.com/
3. Use a relational database for persistence

Bonus Requirements (nice to have)

1. Use EFCore
2. Add a unit test for the validation logic of the POST endpoint

3. Add paging to GET endpoint
4. Improve filtering to work with accented characters, e.g. filter "petraite" should match "Ona Petraitė"
5. Responses for Bad Request responses (POST endpoint) should include error descriptions of what went wrong

During the interview we review the completed homework assignment and ask questions / give feedback about the implementation.

An additional on-the-spot requirement can be added to the assignment during the interview so the interviewee implements on-the-spot changes.