BI Norwegian Business School – .NET Developer

Your response to this assignment is meant to be discussed with us in the interview. The interesting part for us is what you do, and that you are able to reason about why you do the things you do. Please **bring your laptop** and we'll walk through the code together.

Instructions

- Attached you will find the coding assignment that you are expected to complete before the next interview.
- We expect the backend solution to be written in C#/.NET and the front-end solution to be React/TypeScript
- We appreciate tests, and we expect you to **submit at least one (unit or integration) test** that we can discuss during the interview.

The assignment will be the basis for our next meeting, and it does not need to be perfect. We expect you to spend 2-3 hours on the assignment. We ask you to share the code with us at least 24 hours before our next interview. Send the details on how to access the code to andreas.norstein@bi.no. A link to a Github repository is preferred. Please include a description of how to run the code.

Assignment:

The assignment is divided into two parts, one for setting up a new API. The second part is to create a front-end application and utilize the API you just created. We have also included an *optional* part 3 for you to solve.

Part 1

At BI, we provide an open API for fetching calendar events hosted at our campuses or online. These events cater to a variety of audiences. Your task is to create an **ASP.NET Core Minimal API** that integrates with this existing endpoint:

Endpoint: https://bi.no/api/calendar-events
Query Parameters:
• **Take** (int) – The number of events to fetch (default: 5).
• **Language** (string) – The language of the events (default: "all"). Accepted values: "no", "en", "all".
• **Campus** (string) – Filters events by campus.
• **Audience** (string) – Filters events by audience.

Requirements:

• The API should expose an endpoint that allows fetching calendar events with the same filtering options as the original API.

• Use **Swagger/OpenAPI** to document and present the endpoint.

• Consider how the API can handle data efficiently, as the event data **rarely changes**.

• Structure the code to accommodate for future expansion and to be easily maintained.

Part 2

Your task is to create a **React application using TypeScript** that fetches and displays calendar events from the **API you created in part 1**.

Requirements:

• Display each event with the following details:
• **Title**
• **Image**
• **Location**
• **Start time** and **End time**
• **Audience**
• Implement a filtering feature that enhances usability. You can decide on the most relevant filtering options based on the available data.

The design and styling of the application are not a priority. The focus should be on functionality and code quality.

Part 3 (Optional):

Set up the ASP.NET Core Minimal API with the Following Features:

1. Use **Azure App Configuration** and **Key Vault** for configuration.
2. Utilize **Azure Managed Identity** for authentication to **Azure App Configuration** and **Key Vault**.
3. Expose a **protected API endpoint** using **Roles and Scopes** in **Entra ID**.
4. Use **Swagger/OpenAPI** to document and present the endpoint.

Provide a brief description (with **screenshots and comments**) of the configuration in **Azure** and **Entra ID** needed to establish this setup in **Azure**.