Technical Assessment

Demonstrate capability to build web applications

Description

Simple ‘news portal’ web application built using Angular (Frontend) and .NET (Backend) where the user can view and manage news Articles and Categories.

Categories should have the following fields:

- Id
- Name
- CreatedDateTime

Articles should have the following fields:

- Id
- Title
- Description
- CategoryId
- CreatedDateTime

Pages

Home page

Should list existing Articles (newest on top). Each article should show the Title, Description, Category Name and CreatedDateTime (in a user friendly format).

Only 5 articles should appear. On the bottom of the page there should be a pagination element to load the next/previous 5 Articles. Pagination should be done on the backend (the Frontend should only retrieve and display data for the current page).

On the top of the page there should be a search box and once the user types something in it, the list should filter the articles that are matching the search criteria. It should match the Title and Description fields. Searching should be done on the backend.

Next to each Article there should be an Edit button. When actioned, it should show the Article page where its details can be updated.

Next to each Article there should also be a Delete button. The delete action should be done using AJAX, without a page reload.

Article page

Page where the user can add new Articles or edit existing ones as described on the Home page section.

Categories page

Page where the user can add new Categories, edit and delete existing ones.

Project structure

The project should consist of two separate layers

- Frontend (Angular 2+)
- Backend (Web API built with .NET Core 2+)

The communication between the two should be through HTTP calls with JSON as the data format.

Use Visual Studio (the free Community version can be downloaded from here https://www.visualstudio.com/vs/community/) or Visual Studio Code (https://code.visualstudio.com/) as IDE.

Data

- Data should be stored in a Microsoft SQL Server database (this doesn't have to be a full SQL Server installation, for example you can use the LocalDb version that is part of Visual Studio)
- Data access should be implemented using Entity Framework Core.

Backend

- API Controllers should not access Entity Framework entities directly. Database operations should be done through Repositories (Repository Pattern) or similar approach
- Controllers should gain access to Repositories or other dependencies through Dependency Injection
- CreatedDateTime should be automatically set to the current datetime when a record is saved in the database and it should not be editable

Frontend

- The pages described above should be implemented as Angular components.
- HTML/CSS - this is not very important, but you can use any Angular component libraries like PrimeNg, CSS frameworks like Bootstrap or Foundation or to build the layout of the pages and to style form elements, buttons, etc.

Quickstart Tutorials

- https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-web-api
- https://docs.microsoft.com/en-us/learn/modules/build-web-api-net-core/
- https://www.youtube.com/playlist?list=PLC3y8-rFHvwhBRAgFinJR8KHIrCdTkZcZ
- https://www.udemy.com/course/angular-quick-start/