Zeux full stack Developer

The purpose of this assignment is for the candidate to demonstrate the ability to

- create a web component and manage its state
- create a web application and integrate it with a secure RESTful api.

Do not spend more than 5 hours on this assessment.

Objective:

Create a web page containing the following design:
https://invis.io/97NNP97B6FK#/282382917_My_Assets_-_Scroll_Down
Please use a mobile first approach. It's a bonus if you can simply demonstrate fluid layout and responsive design when the screen size increases.

Please replace the following with text:

- The logo in on the top left => "logo"
- The cog in on the top right => "settings"
- Remove all logos in bottom navigation.

The navigation bar should direct you to different routes. Routes other than "Assets" will be empty. The top bar with the title and the bottom navigation bar should always be visible.

The "Opportunities" tab can be empty but should be integrated with routing.

The "Savings, P2P, Funds" tabs can be empty but should be integrated with routing.

The data to populate the list of investment products should be retrieved from a RESTful api. This api should be **secured** so that only authenticated users can retrieve it.

You only need to retrieve 10 products each containing the following information

- Name
- Product Type
- Price
- Returns (e.g. 2.34% Yearly)

Requirement:

The web components can be written using the language/framework of your choice (pure Javascript/Angular/React Redux).

The web api must by written in C#, but you can use any .net framework.

You have the freedom to choose how you secure the api. An example would be using Http verb attributes, and authenticating a token for this API call. We don't need full authentication but we do expect you to demonstrate your understanding as to how an API endpoint can be restricted to authenticated users.

To Save time you do not need to implement the following:

Do not set up account registration or login process. I will assume they have already successfully taken place. You may hard code any key/token on the client and server side.

As an alternative to implementing a database you can choose to simply provide a fake in memory data layer.

You do not need to implement pagination.

Things to consider

Good, comprehensive implementation of an API (limit the method with action verbs, return correct status etc...)

Architecture, separation of concerns

Unit tests

Submission

Include a readme.md file with clear instructions on how to build and run the project along with any tests you've written. Versions of dependencies like node, npm, any global packages, and .net framework version should also be stated to avoid problems running the project.

Zip up your solution and send to nathan.liu@zeux.com, or provide a public github repository. The solution must run on a windows machine.