**Siccar Assessment - accompanying document**

I have implemented the wireframes I was given into a very simple html/css solution and is fully responsive.

This is a basic HTML implementation and not using any javascript libraries such as React or Vue.

The CSS is built using SCSS with some includes for colour and font sizes, meaning that colour can easily be switched simply by changing the values in the include values.

Limited use of javascript as to not get bogged down with making the solution fully working.  I left a couple of things out such as a calendar control as there are many of that can be used.  I have always found it better to use an already created calendar control because of the level of complexity that is involved with date controls.

The following will show what could be done if this was a fully fledged solution.

**Javascript Library/Frameworks**

The solution would ideally be converted into a Single Page Application (SPA) using any of the 2 main javascript libraries, either React or Vue.  Or, if more complexity were required then perhaps a framework such as Angular.

The good thing about using any of these is the ability to create a centralised store (single source of truth) of data on the front end, meaning the data can be

accessed from anywhere in the application and also updated to the same central location.

**Styling**

Ideally a common reusable theme would be used, perhaps something like Bootstrap, Bulma, Tailwind or Materialize, or given time, create a custom css library just for Siccar. Creating a custom library allows you to create a unique look that doesn't just look like everything else out there. When you see a bootstrapped site you tend to know that it is.

**Database integration**

To send and receive data we would ideally have some sort of Rest API either built in a language such as Python or even Javascript (or whatever the business currently uses). This should allow for GET, PUT/PATCH, POST, AND DELETE.

**Page Implementation**

- **Confirm your details**

  This is a simple form that asks a user for their information, ideally the data should be validated, especially the Postcode and National Insurance Number. In the design the gender was shown as checkboxes, I changed this to radio buttons as only one gender can be selected.

- **Loading your details**

  A simple javascript loading bar has been implemented here and once 100% of the loader has been filled the number of records retrieved is shown along with a button that can be clicked to show the current pension schemes the user has.

- **Pension List**

  A simple table based layout showing the user their current pension schemes.