✨✨ Simplesurance Frontend Engineer Coding Challenge (v6) ✨✨

Welcome to the Simplesurance Frontend Engineer Coding Challenge. We use this challenge to access our Frontend Engineer candidates - in this case you!

This coding challenge is meant to measure your skills regarding:

- React best practices
- Application state handling
- Component based development
- Array manipulation
- Data visualisation
- CSS
- Testing

These are a few of the technical skills that you will need as a Frontend Engineer at Simplesurance.

We hope that you will enjoy this challenge and also learn something in the process. The coding challenge should take anywhere from 4 - 8 hours depending on your speed. Please read the requirements carefully and let us know if something is not clear or if you have other questions - we are here to assist.

The Problem

Imagine that your company sells insurance for devices (e.g. Simplesurance). You want to offer an easy way to record a sale of an insurance for a device. The stakeholder also wants to understand, for which device, the most insurances are being sold. Additionally, sometimes fake or incorrect sales are recorded, and these need to be deleted or updated in the system. It's up to you, to make this all possible. The company is trusting in you and relying on your creativity and skills as a developer to create something that will grow into something great! See below for all the details.

Functional Requirements

Our product team has broken down the requirements into the following points (usually these would be tickets):

1. *Capture product insurance sales:*

- Create a form which accepts as input the following data with the associated validation rules:
  - First Name (required)
  - Last Name (required)
  - Email Address (required | valid email address format)
  - Age (required | age should be >= 18)
  - Product to ensure (required | should come from the list of available products - see 2. below)
- All fields should be validated before the sale can be submitted
- Store the sale entry on submission in the application state and reset the form
- Tests should be written for this form to ensure valid and invalid input is accepted and not accepted, respectively.

2. *Load the list of products from the backend*

- Use the following API endpoint to retrieve a list of the latest 20 devices
  - Retrieve the latest devices
  - Ensure these products are selectable from the *Product* field in the form created in 1. (Show only the *DeviceName*)
  - Note: this is a 3rd party api so if you experience any issues, simply create your own mock data locally in the application (i.e. don't let this block you from proceeding with the task).

2. *List the sale entries*

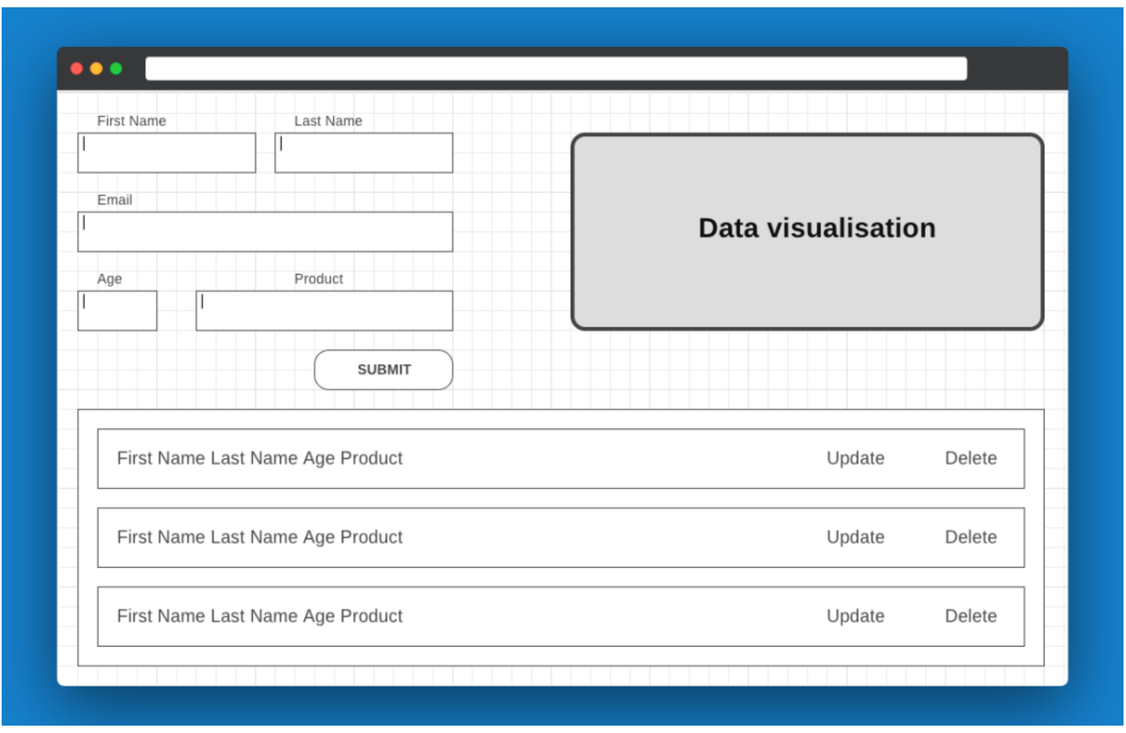
- Implement a list which shows the following details of each sale entry:
  - First Name
  - Last Name
  - Age
  - Product
- Find a way for a sale entry to be update - this could repopulate the form fields and allow the user to edit the values and update the sale entry or allow for inline editing in the list
- Allow for a sale entry to be deleted - this permanently removes that entry from the application state and therefore the list

3. *Visualise the number of insurance sales per product*

- Find a way to visualise the number of insurance sales per product. For instance, you could use a pie chart or graph. Feel free to use a third-party library if needed.

Design Requirements

We have put together the following low-fidelity design to illustrate the features that need to be available in the solution:



It's not required that you follow this design/layout but rather it is there to help clarify the requirements. Be creative but keep in mind that your solution should ideally be responsive.

General Requirements

- How you setup the app is up to you but please use React to build the UI and avoid any *large* boilerplate projects which add a lot of unused additional code. The solution should include a README documentation that explains how to run the solution.
- If you cannot complete something within the given time, please include in the README file, what still needs to be done and how you would approach it if you had more time.
- Please send us a link to the solution's Github (or any other similar source code control system) repository.

Thanks for taking the time to do this challenge, we really appreciate it and we are looking forward to your submission. Best of luck! 🎉