Frontend Coding Assignment

At Layer Health we want to build clean interfaces so that users can understand potentially messy and complex patient data. For this assignment we'd like you to build a web based application for browsing patient notes. Given a collection of patients and notes associated with those patients, we want to be able to select patients and notes that are potentially interesting. For example, we may want to find patients over the age of 65 that have notes that mention cancer.

We've provided a mock dataset for the assignment consisting of two json files, `mock_patients.json` and `mock_notes.json`. Each file contains an array of objects representing patients and notes.

JavaScript

```
interface patient {
    /** The id of the patient. */
    id: number;
    /** The name of the patient. */
    name: string;
    /** The gender of the patient. */
    gender: string;
    /** An ISO 8601 date string. */
    date_of_birth: string;
}

interface note {
    /** The id of the note. */
    id: number;
    /** The name of the provider (doctor) who wrote the note. */
    provider_name: string;
    /** the name of the hospital where the note was written. */
    hospital_name: string;
    /** The date the note was written formatted as an ISO 8601 date string. */
    created_at: string;
    /** The id of the patient associated with the note. */
    patient_id: number;
    /** The text contents of the note. */
    text: string;
}
```

To simplify prototyping we've also provided two smaller json files `mock_patients_small.json` and `mock_notes_small.json`. These contain the same data but with fewer patients and notes.

> **Note**: All the data in the mock dataset was synthetically generated. The notes were randomly sampled from https://huggingface.co/datasets/starmpcc/Asclepius-Synthetic-Clinical-Notes so data in the note such as patient names, ages, etc… most likely will not match the mock patient metadata or mock note metadata.

Requirements

1. The interface should be user friendly and intuitive.
2. Implement functionality to search for notes based on keywords or phrases within the note `text`.
3. Implement filters to allow users to filter notes based on the note metadata.
4. Implement filters to allow users to filter patients based on patient metadata.
5. Write the application using React and TypeScript.
    a. You may use any third party packages/libraries you want.
    b. If you have familiarity with tailwindcss please use it for styling, otherwise you may style the application however you would like.
6. Write clean, well structured, and maintainable code.

Non-Requirements

Anything listed here is considered out of scope given the time constraints.

1. The application does not have to meet accessibility standards.
2. The application does not have to be responsive. It should look reasonable on a laptop or desktop computer but does not have to work on mobile.
3. Data can be stored directly in memory; you do not need to implement a backend.

Deliverables

Source code (preferably a GitHub repo) and instructions for running the application.