**CareFinder Client**

Name: Poojan G Patel

**Overview**:  
For the CareFinder Client project, I implemented seven endpoints that allow users to query hospitals based on various criteria. The endpoints include:

1. hospitals: Retrieves all hospitals.
2. hospitals/id: Fetches a specific hospital by its ID.
3. hospitals/city: Lists hospitals located in a particular city.
4. hospitals/state: Lists hospitals within a specified state.
5. hospitals/county: Lists hospitals within a particular county.
6. hospitals/citystate: Lists hospitals using a combination of city and state.
7. hospitals/name: Finds hospitals by their name.

The client interface allows users to select their query type and input the relevant details. The data is then retrieved and displayed in a table format.

**Trials & Tribulations**:

1. **CORS Policy Restrictions**:  
   A significant challenge was managing CORS (Cross-Origin Resource Sharing) restrictions. The backend server blocked direct API calls from the frontend, resulting in frequent errors during testing. Finding a solution without having to set up a custom server was tough. This required deep dives into CORS configurations and alternative approaches like using third-party proxies.
2. **Data Display Issues**:  
   Initially, data retrieved from the API wasn't displaying correctly in the table format. This was due to mismatched data structures between the backend responses and the frontend code. Adjusting the structure and ensuring proper parsing on the frontend side resolved this.
3. **UI Improvements**:  
   Making the webpage look appealing while keeping it functional was challenging. I refined the design of the table and ensured that it could dynamically display different sets of data based on user queries. Simplifying the XML content display also helped in improving the user experience.

**Remaining Bugs**:

1. **Intermittent API Timeouts**:  
   There are occasional delays or timeouts when querying the backend, especially for large datasets. These appear to be server-side latency issues, which can't be addressed directly from the client-side.
2. **UI Responsiveness**:  
   While the current version of the client is functional on desktop, the layout struggles on smaller screens. Some tables overflow, and the dropdown menus aren't fully responsive on mobile devices.

**Conclusion**:  
The implementation of the CareFinder Client involved overcoming backend connectivity hurdles and refining the user interface for better data presentation. While a few issues persist, the project successfully meets the core functionality requirements.