JavaScript Test Description: Comprehensive User Management Dashboard

Instructions

You have 48 hours to return your code to this email, doing as much as you can.

- Please deploy your code to https://vercel.com/, and then send the page URL to us.
- Please also send your code to us.
- If needed add instructions of how to run/build the project
- If you are busy and don't have enough time please contact me and we can postpone the test to another moment.
- If you have any questions feel free to ask.

Objective

Develop a fully functional web application capable of fetching and presenting intricate user data from a designated external API. This task is designed to rigorously assess your proficiency in manipulating the DOM, handling intricate JavaScript objects, and effectively interacting with an external API in real-time.

Task Overview

You will construct a web application that retrieves a comprehensive list of users from an external API and presents their detailed information within a structured table. This includes implementing real-time search and filter capabilities based on user input. The application must dynamically adjust the DOM based on user interactions and API responses, ensuring a smooth and responsive user experience.

Detailed Requirements

HTML Structure

Your application should include the following: Input Field: An input field for filtering users by their name, equipped with real-time search capabilities. Action Button: A button labeled "Fetch Users" to initiate the API request. Table Element: A

element to display user data, with columns for "Name", "Username", "Email", "Phone", "City", and "Company".

JavaScript Objects

Fetch data from the external API and convert it into an array of JavaScript objects. Each object should adhere to the following structure:

```
const user = {
  id: 1,
  name: "Leanne Graham",
  username: "Bret",
  email: "Sincere@april.biz",
  address: {
    street: "Kulas Light",
    suite: "Apt. 556",
    city: "Gwenborough",
    zipcode: "92998-3874",
    geo: {
      lat: "-37.3159",
      lng: "81.1496",
    },
  },
  phone: "1-770-736-8031 x56442",
  website: "hildegard.org",
  company: {
    name: "Romaguera-Crona",
    catchPhrase: "Multi-layered client-server neural-net",
    bs: "harness real-time e-markets",
  },
};
```

External API Request

Utilize the public API JSONPlaceholder https://jsonplaceholder.typicode.com/users to retrieve the user list when the "Fetch Users" button is activated. The data fetched should be stored in an array of user objects.

DOM Manipulation

Dynamic Table Generation: Automatically generate and populate the table with rows for each user retrieved from the API. Each row should display the user's name, username, email, phone, city (from address.city), and company name. Filter Function: Implement a function to filter the displayed users as the user types into the input field. This should update the table rows in real-time, displaying only users whose names match the input.

Error Handling

Implement robust error handling for the API request: Display a clear error message in the UI if the API request fails or if any issues occur during the data fetching process. Advanced Features (Optional but Recommended) Implement sorting capabilities on the table headers (e.g., sort users by name, email, etc.). Allow users to select a specific user and display more detailed information in

a modal or separate section. Add pagination to manage the display of a large number of users efficiently.

Styling (Optional)

While basic styling is encouraged, the primary focus should remain on functionality and user experience. Feel free to use CSS frameworks like Bootstrap or Tailwind CSS to enhance the visual appeal of your application.

Advice

Ensure that your project runs smoothly and without errors; otherwise, it will not be evaluated. You are free to use any framework, typescript or plain JavaScript, according to your preference. You are free to use other tools like AI assistants coding like copilot