Assessment for Frontend Developer

Your task is to create a React component called `UserList` that fetches data from a given API endpoint (`https://dummyjson.com/users`) and displays a list of users. You should use the fetch API or any preferred library to retrieve the data and handle the loading state while fetching. The component should render each user's name, email, and additional details. Additionally, implement advanced features such as pagination, filtering, sorting, and fetching individual user posts on user click.

Expectations:

- Create a functional or class component named `UserList`.
- Implement the component using TypeScript.
- Set up a well-organised folder structure for your React project.
- Follow best practices and write clean, maintainable code.
- Optimise the code for performance and efficiency.
- Handle advanced features, including pagination, filtering, sorting, and fetching individual user posts on user click.
- Demonstrate knowledge of advanced React concepts and techniques.

Instructions:

- Set up a new React project with TypeScript using your preferred build tool (e.g., Create React App with TypeScript).
- Create a new folder structure for your project that separates components, styles, and utility functions.
- Inside the src folder, create a components folder and a utils folder.
- Inside the components folder, create a new file called UserList.tsx.
- Implement the UserList component as a functional or class component using TypeScript.
- Inside the UserList component, fetch the user data from the API endpoint (https://dummyjson.com/users) using the fetch API or any preferred library (e.g., axios).
- While the data is being fetched, display a loading message or spinner to indicate that the data is being loaded.
- Once the data is fetched, update the component's state with the received user data.
- Implement pagination functionality to allow users to navigate through the list of users.
- Implement filtering functionality to allow users to search and filter the list of users by name, email, or other relevant criteria.
- Implement sorting functionality to allow users to sort the list of users by different fields (e.g., name, email, date created).
- Render the list of users, displaying their names, emails, and any additional details provided by the API.
- Add a button or link for each user to fetch their individual posts when clicked.
- On user click, fetch the posts for the selected user using the API endpoint (https://dummyjson.com/posts) where the id of the user is userId in posts and display them.
- If there is an error while fetching the data, display an appropriate error message.

- Ensure that the code follows TypeScript best practices, including using proper type annotations for variables, props, and state.
- Optimize the code for performance and efficiency, considering factors like minimizing re-renders and handling edge cases.

Submission Instructions:

- Host your project on a suitable platform (e.g., Netlify, Vercel, etc.)
- Create a public GitHub repository to store your project code.
- Commit your code to the repository, including all necessary files and folders.
- Provide the hosted URL of your project, where the UserList component is accessible.
- Provide the URL to your GitHub repository.