MERN Stack Interview Task

Backend:

1. Initialize a Node.js (Express.js) Application:

Begin by creating a new Node.js application using Express.js. This application will serve as the backend for our project, handling data requests and responses.

2. Create an Endpoint - `/v1/users`:

Define an API endpoint `/v1/users` that the candidate needs to implement. This endpoint will integrate two external APIs:

- https://jsonplaceholder.typicode.com/users: Fetches user data.
- https://jsonplaceholder.typicode.com/posts: Fetches post data.

The candidate's task is to call both APIs in a single request, retrieve the data, and then combine the post data with user data. The combination should be based on the `userId` field, and the result should be sent as a response.

3. Implement User Search Functionality:

Extend the API to support search functionality by user name. If a search query is provided in the request parameters (e.g., `/v1/users?searchText=John`), the API should filter the results based on the user's name and return matching users.

Frontend (React.js):

1. Set Up React App:

Create a new React.js application using Create React App. This will be the frontend part of our MERN stack.

2. Display Users Data in a Table with Search:

The candidate is required to create a visually appealing table to display the combined user and post data obtained from the backend. Additionally, implement a search input within the table that allows users to search for a specific user by name.

3. Integrate Search Functionality:

The search input in the table should send a request to the backend API with the search text as a query parameter (e.g., `/v1/users?searchText=John`). The backend should then filter the results based on the provided search text and return the matching users.

4. Plain CSS Styling:

Emphasize the use of plain CSS for styling. The goal is to make the table look good and eye-catching without relying on external frameworks or libraries.