Frontend Engineer Take-Home Challenge

The objective of this assignment is to create a **frontend application** using **React.js** that retrieves and displays data from a public API. The goal is to demonstrate your proficiency in React, your eye for design and user experience, and your ability to write clean, maintainable code.

The Task

1. **Build a React Application**:
   Create a **Single Page Application (SPA)** using **React.js** to display data from a public API of your choice (or you can use one of the suggested APIs below). You are encouraged to choose an API with interesting data that you can present in a creative way.
2. **Fetch and Display Data**:
   - **Retrieve data from an open-source API**: Make API calls to fetch data and display it on your React application. Feel free to use any API, but here are a few suggestions:
     - **The Cat API**: Retrieve pictures and information about cats.
     - **The Dog API**: Fetch different breeds of dogs and their images.
   - **Design a clean and intuitive UI**: Present the data in a visually appealing manner, using HTML, CSS, and any React component library if you wish (e.g., **Material-UI**, **Bootstrap**).
3. **Basic Functionality**:
   - **List View**: Display a list of items fetched from the API (e.g., a list of cat breeds, Dogs, etc).
   - **Detail View**: Implement a feature to show more details for a specific item when clicked (e.g., clicking on a breed shows its characteristics).
   - **Search/Filter** (Optional but Encouraged): Allow users to search or filter the displayed data based on some criteria.
4. **Styling and Responsiveness**:
   - Use **CSS** to style your application. It should look good on both desktop and mobile devices.
   - Make sure the design is clean, with a focus on user experience and responsiveness.
5. **Documentation**:
   - Provide a **well-documented** `README.md` file with:
     - **Overview**: A brief description of your project.
     - **Setup Instructions**: Step-by-step guide on how to install and run your application.
     - **Dependencies**: A list of all the dependencies used in your project.

        - **Features**: A summary of the features and functionality of your app.
        - **Code Structure**: An explanation of the code structure and key files.

Submission Guidelines

- Please submit the assignment by **[Submission Deadline Date]**.
- Upload your code to a **GitHub repository** and share the link with us. Please make sure the repository is public.
- The repository should contain:
    - **Source code** for the React application.
    - **README.md** file with clear setup instructions.
    - Any additional assets (e.g., images, stylesheets).

Important Rules

- **No Use of AI Tools**: All code, UI, and documentation must be your original work. **Any use of AI-based tools like ChatGPT or code generators to create code or content will result in disqualification.**

Evaluation Criteria

- **Correctness**: Does the application function as expected?
- **UI/UX Design**: Is the UI clean, intuitive, and responsive? Is there a good balance of functionality and visual design?
- **Code Quality**: Is the code well-organized, readable, and maintainable?
- **React Best Practices**: Use of React hooks, component organization, and state management.
- **Styling**: Use of CSS to create an attractive and responsive design.
- **Documentation**: Is the `README.md` well-written with clear instructions and an overview of the project?

Bonus Points

- Adding **unit or component tests** (using frameworks like Jest, React Testing Library).
- Using **TypeScript** for type safety in your code.
- Implementing additional features like **pagination**, **sorting**, or **filtering**.
- Creative UI components and animations to enhance user experience.

Good to Know

This is an opportunity for you to show your skills, creativity, and ability to think critically. We are excited to see how you approach this challenge and look forward to your submission!