Frontend Development with React.js

Project Documentation format

1. **Introduction**
    - **Project Title**: Navigate the news landscape
    - **Team leader** : N Kutdraleshwaran - 12091cs22@princescience.in

    - **Team Members** : **D Gokul** - 11960cs22@princescience.in

      : S Jaisuriya - 12537cs22@princescience.in

      : D Velmurugan - 12463cs22@princescience.in

      : R Dinesh - 12107cs22@princescience.in

2. **Project Overview**
    - **Purpose**: Briefly describe the purpose and goals of the project.
    - **Features**: Highlight the key features and functionalities of the frontend.
3. **Architecture**
    - **Component Structure**: Outline the structure of major React components and how they interact.
    - **State Management**: Describe the state management approach used (e.g., Context API, Redux).
    - **Routing**: Explain the routing structure if using react-router or another routing library.
4. **Setup Instructions**
    - **Prerequisites**: List software dependencies (e.g., Node.js).
    - **Installation**: Provide a step-by-step guide to clone the repository, install dependencies, and configure environment variables.
5. **Folder Structure**
    - **Client**: Describe the organization of the React application, including folders like components, pages, assets, etc.
    - **Utilities**: Explain any helper functions, utility classes, or custom hooks used in the project.
6. **Running the Application**
    - Provide commands to start the frontend server locally.
        - **Frontend**: npm start in the client directory.
7. **Component Documentation**
    - **Key Components**: Document major components, their purpose, and any props they receive.
    - **Reusable Components**: Detail any reusable components and their configurations.
8. **State Management**
    - **Global State**: Describe global state management and how state flows across the application.
    - **Local State**: Explain the handling of local states within components.
9. **User Interface**
    - Provide screenshots or GIFs showcasing different UI features, such as pages, forms, or interactions.
10. **Styling**

- **CSS Frameworks/Libraries**: Describe any CSS frameworks, libraries, or pre-processors (e.g., Sass, Styled-Components) used.
- **Theming**: Explain if theming or custom design systems are implemented.

11. **Testing**

- **Testing Strategy**: Describe the testing approach for components, including unit, integration, and end-to-end testing (e.g., using Jest, React Testing Library).
- **Code Coverage**: Explain any tools or techniques used for ensuring adequate test coverage.

12. **Screenshots or Demo**

- Provide screenshots or a link to a demo showcasing the application's features and design.

13. **Known Issues**

- Document any known bugs or issues that users or developers should be aware of.

14. **Future Enhancements**

- Outline potential future features or improvements, such as new components, animations, or enhanced styling.