Frontend Development with React.js

Project Documentation format

It looks like you want to structure a project documentation based on the outlined template, and the project seems to relate to navigating or streaming news in a modern way. Here's how you might go about filling in this template with a specific focus on a news streaming application.

---

1. Introduction

- **Project Title:** InsightStream: Navigating the News Landscape
- **Team leader**

  Vishali.N

- **Team Members:**

  Nithya.D
  Priya Dharshini.K
  sowmiyaja.A.V

---

2. Project Overview

- **Purpose:**
  InsightStream is a news streaming platform designed to provide users with real-time, curated news feeds. The app uses intelligent algorithms to prioritize and personalize news content based on user interests, trending topics, and user behavior. Our goal is to make it easier for users to stay informed with a simple, intuitive interface.
- **Features:**
  - Real-time news feed based on categories (e.g., World, Technology, Sports).
  - Personalization of news according to user preferences.
  - Search functionality for articles, sources, and keywords.
  - Bookmark and save articles for later reading.
  - Dark mode and light mode support.
  - Push notifications for breaking news.

---

3. Architecture

- **Component Structure:**
  The main React components include:
  - `App`: The root component that houses the global layout and routing.
  - `NewsFeed`: Displays the list of news articles.
  - `ArticleCard`: A reusable component for displaying each news article.
  - `SearchBar`: Provides users with the ability to search for specific articles.

- `ArticleDetails`: Displays full details of a selected article.
- `Settings`: Manages user preferences for theme and notification settings.

- **State Management:**
  We use **React Context API** for global state management. The global state includes user preferences (theme, language), the current news feed data, and search results. For local component-specific states, we rely on React's `useState` hook.
- **Routing:**
  We use **react-router-dom** to handle navigation within the app:
  - `/home`: Displays the main news feed.
  - `/article/:id`: Displays a detailed view of an article.
  - `/search`: Displays search results based on user input.

---

4. Setup Instructions

- **Prerequisites:**
  - Node.js (version 14 or higher)
  - npm or yarn (for dependency management)
  - A News API key (for fetching news data)
- **Installation:**

1. Clone the repository:
   ```
   2. git clone https://github.com/yourusername/insightstream.git
   ```
3. Navigate to the `client` directory:
   ```
   4. cd insightstream/client
   ```
5. Install dependencies:
   ```
   6. npm install
   ```
7. Configure environment variables:
   - Create a `.env` file in the root directory and add the following:
   - `REACT_APP_NEWS_API_KEY=your_api_key_here`
8. Start the development server:
   ```
   9. npm start
   ```

---

5. Folder Structure

- **Client:**
  - `/components`: Reusable React components (e.g., `ArticleCard`, `SearchBar`).
  - `/pages`: React components corresponding to different pages (`Home`, `ArticleDetails`, `Search`).
  - `/assets`: Images, icons, and other static resources.
  - `/utils`: Helper functions, custom hooks, API calls.
- **Utilities:**
  - **API Utility:** A custom hook `useFetchNews` fetches the latest news data from the external API.
  - **Theme Utility:** A helper to toggle between dark and light modes.

---

6. Running the Application

- **Frontend:**

- Run the following command in the `client` directory to start the frontend server locally:
- `npm start`

This will launch the app at `http://localhost:3000`.

---

7. Component Documentation

- **Key Components:**
  - **`NewsFeed`:** Displays the list of articles. It accepts `category` and `articles` props.
  - **`ArticleCard`:** Displays a preview of an article with the title, image, and summary. Receives props like `title`, `image`, and `summary`.
  - **`ArticleDetails`:** Shows the detailed view of a single article. It receives the `articleId` as a prop to fetch specific article details.
- **Reusable Components:**
  - **`Button`:** A simple button component used across the application, which accepts `label`, `onClick`, and `style` as props.

---

8. State Management

- **Global State:**
  - User preferences (theme, language) are stored using the Context API and shared across components.
  - The current news feed data is also stored globally, ensuring that all components accessing the feed are updated when new data is fetched.
- **Local State:**
  - Components like `ArticleDetails` use local state to manage dynamic changes, such as loading state or saving articles.

---

9. User Interface

- Screenshots or GIFs showcasing the news feed, article details, and search functionality will be added here. These would show the app's interface in both light and dark modes, and how the user interacts with articles.

---

10. Styling

- **CSS Frameworks/Libraries:**
  - We are using **Styled-Components** for scoped and themeable CSS, allowing for a modular design.
- **Theming:**
  - The app supports both **light** and **dark** themes, with styles dynamically applied based on the user's preferences, stored in global state.

---

11. Testing

- **Testing Strategy:**
  - **Unit Tests**: We use Jest for testing individual components (e.g., `Button`, `ArticleCard`).
  - **Integration Tests**: Testing interactions between components, like submitting a search query and displaying results.
  - **End-to-End Tests**: Using **Cypress** to simulate user actions and ensure that the entire flow works as expected (e.g., searching for articles, navigating between pages).
- **Code Coverage:**
  - We use Jest's built-in code coverage tool to ensure that at least 80% of the code is tested.

---

12. Screenshots or Demo

- Screenshots of the app's interface and a demo link will be added here once the app is deployed.

https://drive.google.com/file/d/1-6dAGmfmWqdaSdUvzfyYR2yi9TmaQ6tT/view?usp=drivesdk

---

13. Known Issues

- **Bug with Search:** Sometimes the search bar doesn't immediately update the results when typing. We're currently investigating this issue.
- **Theme Switching Lag:** There may be a slight delay in switching between light and dark mode, which will be optimized in future releases.

---

14. Future Enhancements

- **Push Notifications:** To alert users of breaking news even when the app is not in focus.
- **More Personalization:** Allow users to create personalized news topics.
- **Social Sharing:** Add the ability to share articles directly to social media platforms.
