**1. Introduction**

* **Project Title**: News App
* **Team Members**:
  + SathiyaPriya M
  + Swathi B
  + Soundariya S
  + Sujathidevi V
  + Sumithra V
* **Team Leader Email**: sathiya842005@gmail.com

**2. Project Overview**

* **Purpose**:  
  The News App is designed to provide users with the latest news and updates from various sources. The main objective is to create an intuitive interface where users can browse news articles, filter news based on categories, and read detailed content. The app aims to bring the world’s news right to the fingertips of users.
* **Features**:
  + **News Feed**: A dynamic feed displaying the latest articles from different sources.
  + **Category Filters**: Users can filter news based on categories (e.g., Technology, Sports, Health).
  + **Search Functionality**: Users can search for specific news articles.
  + **Article Details Page**: Clicking on a news item takes the user to the detailed article page.
  + **Bookmarking**: Users can bookmark their favorite articles for quick access later.
  + **Dark Mode**: An optional dark theme for better readability at night.

**3. Architecture**

* **Component Structure**:  
  The app is divided into the following key components:
  + **App**: The root component, responsible for rendering other components.
  + **NewsFeed**: Displays the list of news articles with brief details.
  + **ArticleDetail**: Displays detailed information for a selected article.
  + **SearchBar**: Provides the search functionality for finding news.
  + **CategoryFilter**: Filters news by category (e.g., Technology, Health).
  + **BookmarkList**: Displays saved/bookmarked articles.

The components interact through a shared state (using Context API or Redux for state management).

* **State Management**:  
  We use **Context API** for global state management to maintain the current news data, user preferences (like theme or bookmarks), and any other shared state across components. Local state management within individual components is handled using useState hooks.
* **Routing**:  
  We use **React Router** for handling navigation between different pages, such as the home page, article details, and bookmarks. The routing structure is as follows:
  + /: Home page displaying the news feed.
  + /article/:id: Article detail page for viewing a full article.
  + /bookmarks: List of user-bookmarked articles.

**4. Setup Instructions**

* **Prerequisites**:
  + Node.js (v14 or higher)
  + npm (v6 or higher)
* **Installation**:
  + Clone the repository:

bash

Copy

git clone https://github.com/your-repository/news-app.git

* + Navigate to the project folder:

bash

Copy

cd news-app

* + Install dependencies:

bash

Copy

npm install

* + Set up environment variables for API keys (if required) in a .env file:

ini

Copy

REACT\_APP\_API\_KEY=your\_api\_key\_here

**5. Folder Structure**

* **Client**: The main folder containing all React-related code.
  + **/src**
    - **/components**: Contains React components like NewsFeed, ArticleDetail, SearchBar, etc.
    - **/pages**: Contains the React pages (e.g., Home, Bookmarks).
    - **/assets**: Stores images, icons, and other static resources.
    - **/utils**: Contains utility functions, helper classes, or custom hooks like useFetch.
    - **/styles**: Contains global CSS files or styled-components.
* **Utilities**:
  + Custom hooks such as useFetch to fetch API data.
  + Helper functions for managing local storage (e.g., for bookmarks).

**6. Running the Application**

To run the app locally:

* Navigate to the client directory and start the development server:

bash

Copy

npm start

This will launch the app in the default browser at http://localhost:3000.

**7. Component Documentation**

* **Key Components**:
  + **NewsFeed**:
    - **Purpose**: Displays a list of news articles.
    - **Props**: newsData (Array of news articles).
  + **ArticleDetail**:
    - **Purpose**: Shows a full article based on the selected ID.
    - **Props**: articleId (ID of the selected article).
  + **SearchBar**:
    - **Purpose**: Allows users to search for articles.
    - **Props**: onSearch (Function to handle the search query).
* **Reusable Components**:
  + **Button**: A generic button component used throughout the app for actions like "Read More" or "Bookmark".
  + **Card**: A reusable card component to display individual news articles in the feed.

**8. State Management**

* **Global State**:  
  The global state is managed using the Context API. We manage:
  + **News Data**: Stores the list of news articles fetched from the API.
  + **Bookmarks**: Stores the articles that the user has saved.
  + **Theme**: Manages whether the app is in light or dark mode.
* **Local State**:  
  Local state is used within components to handle UI-related states, such as toggling dark mode, managing form inputs, or controlling loading states during data fetching.

**9. User Interface**

Provide screenshots or GIFs showcasing the UI for various features. Examples might include:

* **Home Page**: Showing the news feed and category filters.
* **Article Page**: Displaying the full content of a selected article.
* **Search Functionality**: Displaying results based on search queries.

(Insert relevant UI screenshots or GIFs here)

**10. Styling**

* **CSS Frameworks/Libraries**:  
  We use **Styled-Components** for component-level styling and **CSS Modules** for page-level styling.
* **Theming**:  
  A light/dark mode toggle is implemented. When the user switches themes, the UI adapts accordingly by applying different sets of styles.

**11. Testing**

* **Testing Strategy**:
  + **Unit Testing**: We use **Jest** for unit testing React components.
  + **Integration Testing**: We use **React Testing Library** to simulate user interactions and test component interactions.
  + **End-to-End Testing**: We are planning to integrate **Cypress** for end-to-end testing in future updates.
* **Code Coverage**:  
  We use Jest’s built-in code coverage tool to ensure high test coverage across the codebase.

**12. Screenshots or Demo**

Provide a link to a live demo or include images showcasing the application.

* [Link to live demo (if available)](http://your-app-url.com)

**13. Known Issues**

* **Bug**: Sometimes, the news feed takes a bit longer to load due to slow API responses.
* **Limitation**: The app does not handle offline mode properly yet.
* **UI**: The dark mode switch may not work correctly on mobile in some cases.

**14. Future Enhancements**

* **User Authentication**: Add login functionality so users can save their preferences across devices.
* **News Categories**: Expand the news categories to allow users to follow specific sources or topics.
* **Offline Mode**: Implement offline caching of news for viewing when the internet is unavailable.