**Frontend Development with React.js**

**Project Documentation**

**Introduction**

**Project Title:** news-app (v0.1.0)  
**Team Members:** *Adishwaran-Team leader, Kiran-Team member,Deepak Kumar Ram-Team member*

**Project Overview**

**Purpose:**  
This project is a React-based news application designed to display top stories, category-wise articles, and newsletters. It delivers a modern, responsive UI and allows users to browse news content efficiently.

**Features:**

* Home, Category, and News pages
* Top stories and latest articles section
* Newsletter subscription component
* Responsive Navbar and Footer
* Context API for global state management
* Routing with react-router-dom
* Styled using CSS and Bootstrap

**Architecture**

**Component Structure:**

* **Pages**: Home, NewsPage, CategoryPage
* **Components**: NavbarComponent, NewsLetter, HomeArticles, TopStories, Hero, Footer
* **Context**: GeneralContext for shared state
* **Entry**: App.js integrates routing and global layout

**State Management:**

* Global state handled using **Context API**.
* Local state handled within components using **React hooks** (useState, useEffect).

**Routing:**

* Implemented with **react-router-dom**, providing navigation between Home, News, and Category pages.

**Setup Instructions**

**Prerequisites:**

* Node.js (>= 14)
* npm

**Installation:**

git clone <repository\_url>

cd code

npm install

**Environment Variables:**

* Configure API keys (e.g., for fetching news using axios) if required in .env.

**Folder Structure**

code/

package.json

README.md

src/

index.js

App.js

components/

NavbarComponent.jsx

NewsLetter.jsx

HomeArticles.jsx

TopStories.jsx

Hero.jsx

Footer.jsx

pages/

Home.jsx

NewsPage.jsx

CategoryPage.jsx

context/

GeneralContext.jsx

styles/

\*.css

public/

index.html

favicon.ico

manifest.json

**Client:** Organized into **components**, **pages**, **context**, and **styles**.  
**Utilities:** Axios is used for API requests. Custom hooks may be added as needed.

**Running the Application**

To start the app locally:

npm start

**Component Documentation**

**Key Components:**

* **NavbarComponent:** Navigation across pages
* **Hero:** Highlights top content
* **HomeArticles:** Displays main articles on the home page
* **TopStories:** Lists trending stories
* **NewsLetter:** Newsletter subscription form
* **Footer:** Application footer

**Reusable Components:**

* Navbar and Footer are reused across pages.
* NewsArticle-style components are used in Home and Category pages.

**State Management**

**Global State:** Managed via GeneralContext for shared state like selected categories or news data.  
**Local State:** Handled in individual components for UI-specific states (like form input, toggles, etc.).

**User Interface**

* Responsive Navbar and Footer
* Hero section on Home page
* Category-based browsing
* Newsletter subscription form

**Styling**

* **CSS Frameworks/Libraries:** Bootstrap & React-Bootstrap
* **Custom Styles:** Defined in /src/styles/ as CSS files
* **Icons:** Implemented via react-icons

**Testing**

**Testing Strategy:**

* Unit and integration tests with **Jest** and **React Testing Library**.
* Example test file: App.test.js.

**Code Coverage:**

* Testing configured via setupTests.js for Jest environment setup.

**Screenshots or Demo**

(*Add screenshots of Home, Category, and News pages here.*)

**Known Issues**

* API key must be configured for live data fetching.
* Some styles may need optimization for smaller screens.

**Future Enhancements**

* Add animations and transitions for smoother UI
* Dark mode theming
* Advanced filtering and search functionality
* Progressive Web App (PWA) support

Top of Form

Bottom of Form