News Application Documentation

1. Introduction

The News Application is a web/mobile application that provides users with real-time news updates from various sources. It allows users to browse different categories, search for specific topics, and read full articles. The app is designed to be fast, responsive, and user-friendly.

Project Members:-

1) Karthik.S

2) Tharun Vikash.S

3) Prasanth.G

4) Siva Thanga Balaji.S

2. Project Overview

Features:-

Fetch and display news articles from an API (e.g., NewsAPI, custom backend).

Categorized news sections (e.g., Politics, Sports, Technology, Entertainment).

Responsive design for web and mobile compatibility.

3. Architecture

The application follows a modular component-based architecture with clear separation between concerns.

Architecture Diagram

```
📂 news-app
 ├📂 src
 │  ├📂 components
 │  │  ├📂 NewsCard
 │  │  ├📂 Header
 │  │  ├📂 Footer
 │  │  ├📂 SearchBar
 │  │  └📂 CategoryFilter
 │  ├📂 pages
 │  │  ├📜 Home.js
 │  │  ├📜 Article.js
 │  │  └📜 Bookmarks.js
 │  ├📂 redux
 │  │  ├📜 store.js
 │  │  └📜 newsSlice.js
 │  ├📂 styles
 │  ├📂 utils
 │  └📂 services
```

```
│   └ 📜 newsService.js
├ 📜 package.json
├ 📜 README.md
└ 📜 .gitignore
```

4. Setup Instructions

Prerequisites:- Node.js

(v16 or later)

npm or yarn

A valid API key for fetching news

Installation Steps:-

1. Clone the repository:

git clone https://github.com/yourusername/news-app.git cd news-app

2. Install dependencies:

npm install # or yarn install

3. Create a .env file in the root directory and add:

REACT_APP_NEWS_API_KEY=your_api_key_here

5. Folder Structure

📂 news-app

├ 📂 src

│ ├ 📂 components

│ ├ 📂 pages

│ ├ 📂 redux

│ ├ 📂 styles

│ ├ 📂 utils

│ └ 📂 services

├ 📜 package.json

├ 📜 README.md

└ 📜 .gitignore

6. Running the Application

To start the development server, run:

 npm start

Project Link : https://github.com/sivathangabalaji2005/news-app.git
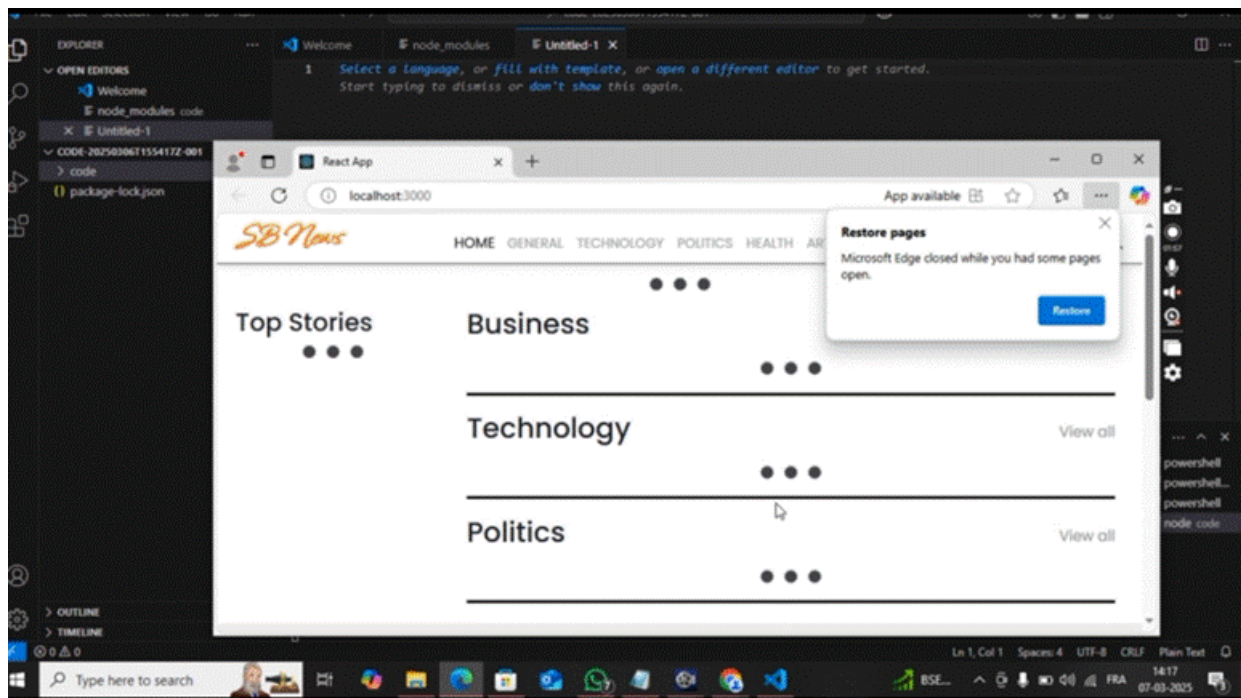
7.Screenshots Or Demo

　　1) Project Demo:

🎬 News Application. Audio

2) Screenshots:

8. Documentation

The `NewsCard` component displays individual news articles, showing the title, description, an image, and a link to the full article. It accepts four props: `title`, `description`, `imageUrl`, and `url`. The `SearchBar` component allows users to search for articles by entering a query and clicking a search button. It takes an `onSearch` function prop to handle the search action. In a parent component, the `handleSearch` function fetches articles based on the search query and updates the state. The `NewsCard` components then display the fetched articles. These components work together to provide a search feature and display news content interactively. The `SearchBar` triggers the search, and `NewsCard` showcases each article's details. The combination enables users to search for and read articles seamlessly.

9. State Management

1.   **Redux Slice with Async Actions**: The `newsSlice.js` uses `createSlice` from Redux Toolkit to manage the state for news articles, including a `loading` flag and `articles` array, along with reducers (`setArticles`, `setLoading`) to update the state.

2.   **Integrating Async Thunks with `createAsyncThunk`**: To fetch data asynchronously, `createAsyncThunk` is used to make an API call to get news articles, while the `extraReducers` handle different states: loading, success, and failure (error handling).

3.   **React Component with Redux**: The React component dispatches the `fetchArticles` thunk on mount using `useDispatch`, and uses `useSelector` to get the state (articles, loading, error) from Redux. The UI conditionally renders loading, error, or the list of articles.

10. Styling

CSS Modules (for component-specific styles).

Tailwind CSS (for utility-first styling).

Styled Components (for dynamic styling).

11. Testing

Using Jest and React Testing Library for unit tests.

Example test for NewsCard:

```
import { render, screen } from '@testing-library/react'; import
NewsCard from '../components/NewsCard';

test('renders news title', () => {   render(<NewsCard title="Test News"
description="Test Description" imageUrl="test.jpg" url="#" />);   const
titleElement = screen.getByText(/Test News/i);
expect(titleElement).toBeInTheDocument();

});
```

12. Future Enhancements

User Authentication: Allow users to save bookmarks across devices.

Push Notifications: Notify users about breaking news.

Offline Mode: Cache articles for offline reading.

Multiple Language Support: Provide news in different languages.