

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 and Initial Setup

Features:

Your news application will have the following core features:

1. Fetch and Display News Articles from an API**
 - The app will use a news API (such as [NewsAPI] or a custom backend) to fetch the latest articles in real-time.
 - You can filter news by various topics, such as politics, sports, and technology.

2. Categorized News Sections

- To enhance user experience, news will be organized into categories:
 - **Politics**
 - **Sports**
 - **Technology**
 - **Entertainment**
 - **Business**
- And more based on your choice of API.

3. Responsive Design for Web and Mobile Compatibility

- The app will be built to be mobile-friendly using CSS frameworks (e.g., Bootstrap, Tailwind) or custom CSS.
- The layout will adjust seamlessly on desktop and mobile screens for a better user experience.

Prerequisites

Before starting the installation, ensure that you have the following:

- **Node.js** (v16 or later)
Download it from [here](https://nodejs.org/).
- **npm** or **yarn**
Package managers for installing project dependencies. `npm` comes bundled with Node.js, but you can also install [yarn](https://yarnpkg.com/) if you prefer.
- A **valid API key** for fetching news data from a service like [NewsAPI](https://newsapi.org/) or a similar provider.

Installing the Application and Configuration

Installation Steps

1. **Clone the Repository**

Open your terminal and clone the project repository:

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

After cloning, navigate into the project directory:

```
```bash
cd news-app
```
```

2. ****Install Dependencies****

Next, you need to install all the necessary dependencies for the app.

- With ****npm****:

```
```bash
npm install
```
```

- With ****yarn****:

```
```bash
yarn install
```
```

This will download and set up all the required libraries and packages that the app depends on.

Set Up the Environment File

Now, you need to configure your API key for the app.

- In the project root, create a ``.env`` file.
- Add your API key in the following format:

```
```env
REACT_APP_NEWS_API_KEY=your_api_key_here
```
```

Replace ``your_api_key_here`` with the actual API key from your chosen news provider.

****Additional Features to Add****

- ****Search Functionality:**** Allow users to search for specific articles or topics.

- **Pagination/Infinite Scrolling:** If the API supports it, implement pagination or infinite scrolling to load more news articles dynamically.
- **Bookmarks:** Let users save articles they want to read later.
- **Dark Mode:** Implement a dark mode toggle for better user experience, especially at night.

More Features to Consider Adding

1. User Authentication (Optional)

- If you want to make your app more interactive, consider adding user authentication where users can sign up, log in, and save their preferences (e.g., favorite categories or articles).

2. News Article Sharing

- Allow users to share articles on social media platforms like Facebook, Twitter, or email.

3. Push Notifications

- For real-time updates, integrate push notifications to inform users of breaking news or specific category updates.

4. News Article Comments

- Allow users to comment on or discuss articles within the app. This can help increase user engagement.

5. Article Details Page

- When users click on a news article, show more detailed information, including images, full article text, and additional metadata (e.g., source, author, publication date).

6. Trending News Section

- Display a "Trending" section that highlights the most popular articles based on current user activity or analytics.

3. Architecture

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

Architecture Diagram



| └─ 📄 newsService.js

| └─ 📄 package.json

| └─ 📄 README.md

| └─ 📄 .gitignore

4. Setup Instructions

Page 1: Prerequisites and Initial Setup

Prerequisites

Before starting the installation, ensure that you have the following:

- **Node.js** (v16 or later)
If you don't have Node.js installed, you can download it from [here](#).
- **npm** or **yarn**
These are package managers used to install the dependencies. **npm** comes bundled with Node.js, while **yarn** can be installed from [here](#).
- A **valid API key** for fetching news data
You will need an API key from a news service like [NewsAPI](#) or any other provider to fetch the news. You can sign up for an API key from their platform.

Page 2: Cloning the Repository and Installing Dependencies

Installation Steps

Clone the repository

First, open your terminal or command prompt. Then, clone the project repository to your local machine using the following command:

After cloning, navigate into the project directory with:

```
cd news-app
```

1. Install dependencies

Once you're inside the project directory, you need to install all the necessary dependencies. You can do this using either **npm** or **yarn**.

With npm:

```
npm install
```

○

With yarn:

```
yarn install
```

○

2. This command will install all the required packages for the project to function.

Page 3: Configuration and Running the App

Set up the environment file

Now that the dependencies are installed, you need to configure the application to use your API key.

1. In the root directory of the project, create a new file called **.env**.

Add your API key to the **.env** file in the following format:

```
REACT_APP_NEWS_API_KEY=your_api_key_here
```

2. Replace **your_api_key_here** with the actual API key you got from the news provider.

Start the application

After setting up the `.env` file with your API key, you're ready to start the application.

With npm:

```
npm start
```

-

With yarn:

```
yarn start
```

-

This will launch the application, and you should be able to access it in your browser at <http://localhost:3000>.

5. Folder Structure

📁 news-app

├─ 📁 src

| └─ 📁 components

| └─ 📁 pages

| └─ 📁 redux

| └─ 📁 styles

| └─ 📁 utils

| └─ 📁 services

└─ 📄 package.json

└─ 📄 README.md

L 📄 .gitignore

6. Running the Application

To start the development server, run:

npm start

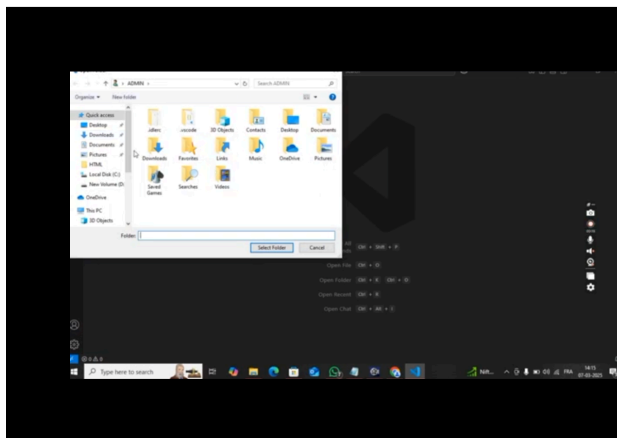
Project Link : <https://github.com/sivathangabalaji2005/news-app.git>

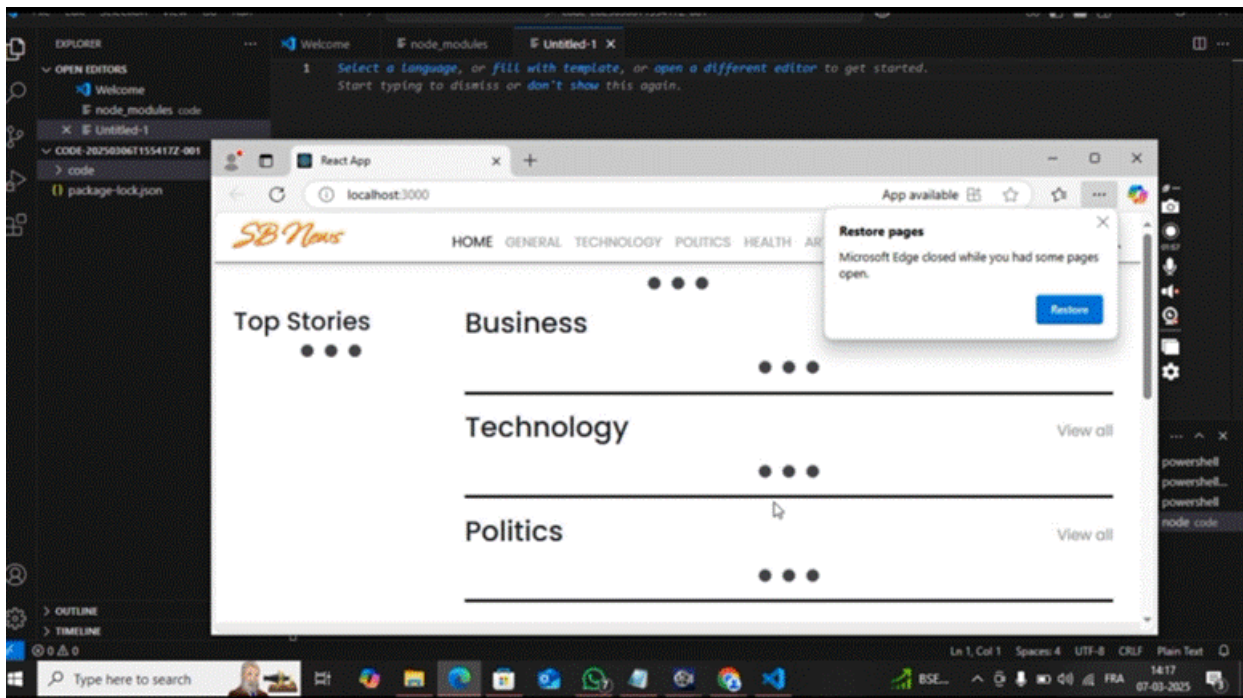
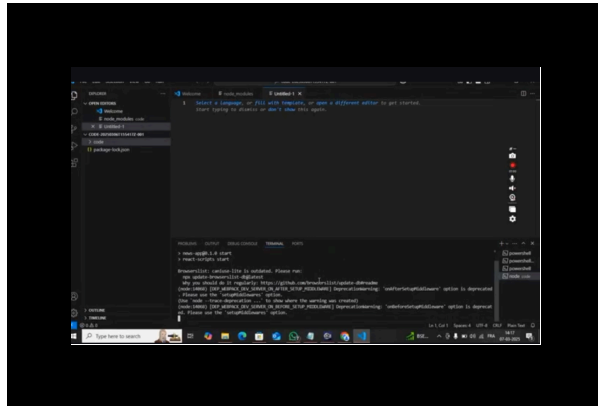
7.Screenshots Or Demo

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.