Frontend Development with React.js

PROJECT DOCUMENTATION

**INTRODUCTION**

**PROJECT TITLE:** **INSIGHTSREAM : Navigate the News Landscape(React Application)**

**TEAM MEMBERS:**

**Team leader: AARTHY.G - gopalraji1973@gmail.com**

**Team member: AMALORPAVAMARY.J - amalorpavam525@gmail.com**

**Team member: VAISHALI.S - vaishalisundaram1705@gmail.com**

**Team member: HARINI.S - harinihari583@gmail.com**

**Team member: BHAVANA.C - bhavanac1905@gmail.com**

**PROJECT OVERVIEW**

**PURPOSE:** **InsightStream** is a modern, interactive news platform built to deliver real-time, categorized news updates in a structured and user-friendly manner. The platform aims to simplify news consumption by offering seamless navigation, personalized content, and a visually appealing interface.

With the vast amount of information available online, **InsightStream** helps users stay informed without being overwhelmed. By leveraging modern web technologies, the platform ensures a fast, responsive, and accessible news experience for users across different devices. It focuses on providing reliable news, easy topic-based filtering, and an engaging reading experience, making it an essential tool for staying updated on current events.

### FEATURES:

### 1.User Interface (UI)

* **Responsive Design:** Ensures the app works smoothly on different screen sizes (mobile, tablet, desktop).
* **Navigation Bar:** Includes categories like *Home, World, Business, Technology, Sports, Health, and Entertainment*.
* **Dark Mode Toggle:** Allows users to switch between light and dark themes for better readability.
* **Search Bar:** Enables users to search for specific news articles by keywords.

**2.News Display**

* **Dynamic Content Rendering:** Fetches and displays news articles from an API or a local database.
* **Category-Based News:** Displays news based on selected categories from the navigation bar.
* **Featured News Section:** Highlights trending or top news articles at the top of the page.
* **Image & Headline Preview:** Each news article includes an image, headline, short description, and a "Read More" link.

**3.Interactivity & User Experience**

* **Live News Updates:** Uses JavaScript to fetch and display real-time news updates dynamically.
* **Infinite Scroll / Pagination:** Allows users to load more articles without refreshing the page.
* **Bookmark Feature:** Users can save articles to read later (using localStorage).
* **Share Functionality:** Buttons to share news articles on social media platforms.

**4.Accessibility & Performance**

* **Keyboard Navigation:** Users can navigate the app using keyboard shortcuts.
* **Lazy Loading Images:** Optimizes page speed by loading images only when needed.
* **Optimized CSS & JavaScript:** Ensures fast loading times with minimal CSS/JS files.

**ARCHITECTURE**

### COMPONENT:

### 1.COMPONENT STRUCTURE:

### 📂 src

┣ 📂 components

┃ ┣ 📜 NavbarComponent.js

┃ ┣ 📜 Footer.js

┣ 📂 pages

┃ ┣ 📜 Home.js

┃ ┣ 📜 CategoryPage.js

┃ ┣ 📜 NewsPage.js

┣ 📜 App.js

┗ 📜 index.js

**2.COMPONENT INTERACTIONS:**

### ****App.js (Main Component)****

* **Manages Routing:** Handles navigation between pages using react-router-dom.
* **Includes Global Components:** Renders <NavbarComponent /> and <Footer /> across all pages.
* **Contains Routes:**
  + / → Loads <Home />
  + /category/:id → Loads <CategoryPage /> dynamically based on id.
  + /news/:id → Loads <NewsPage /> dynamically based on id.

### ****NavbarComponent.js****

* **Displays Navigation Links**: Home, Categories, Search, etc.
* **Handles Routing**: Allows navigation to different categories and home page.
* **Interacts with App.js** by updating the route when a category is selected.

### ****Home.js****

* **Main landing page** displaying general news.
* **Fetches & Displays News Data** in a list format.
* **Calls CategoryPage.js when a category is selected.**

### ****CategoryPage.js****

* **Displays news articles of a specific category.**
* **Uses useParams()** to extract id from URL (/category/:id).
* **Fetches relevant news articles based on the category.**

### ****NewsPage.js****

* **Displays full details of a news article.**
* **Uses useParams()** to get id from URL (/news/:id).
* **Fetches & displays complete article content.**
* **Provides Share and Bookmark options.**

### ****Footer.js****

* **Displayed on all pages** (included in App.js).
* **Contains additional links, social media, and copyrights.**

**3.INTERACTION FLOW:**

1️.User clicks on **a category** → NavbarComponent.js updates the URL → CategoryPage.js loads the news list.  
2️.User clicks on **a news article** → NewsPage.js loads and displays full content using the id from URL.  
3️.User navigates back **to home** → Home.js fetches and displays general news.  
4️. NavbarComponent.js and Footer.js remain visible across all pages.

**STATE MANAGEMENT:** The news app uses **React's Context API** for state management, ensuring a centralized and efficient way to handle global data like news articles, user preferences, and theme settings.

#### **1. Global State Management (Context API)**

* A **NewsContext** is created using React.createContext() to store and manage state globally.
* A **NewsProvider** component wraps the app, making the state accessible to all child components.
* The useContext(NewsContext) hook is used to consume the global state within components.

#### **2. State Handling with** useState **and** useReducer

* useState manages UI-related states like search queries, dark mode toggle, and category selection.
* useReducer is used for handling complex state transitions, such as managing fetched news data.

#### **3. Asynchronous Data Handling (Fetching News Articles)**

* The useEffect hook is used to fetch news articles from an API when the category or search term changes.
* The fetched data is stored in the global state and updated using the setNewsData function.

**SET INSTRUCTIONS:**

● Create a new React app:

npx create-react-app my-react-app Replace my-react-app with your preferred project name.   
● Navigate to the project directory: cd my-react-app

● Running the React App: With the React app created, you can now start the development server and see your React application in action.

● Start the development server: npm start This command launches the development server, and you can access your React app at http://localhost:3000 in your web browser

**PREREQUISITES:**

Here are the key prerequisites for developing a frontend application using React.js:

✔ **Node.js and npm:** Node.js is a powerful JavaScript runtime environment that allows you to run JavaScript code on the local environment. It provides a scalable and efficient platform for building network applications. Install Node.js and npm on your development machine, as they are required to run JavaScript on the server-side.

● Download: <https://nodejs.org/en/download/>

● Installation instructions: <https://nodejs.org/en/download/package-manager/>

✔ **React.js**: React.js is a popular JavaScript library for building user interfaces. It enables developers to create interactive and reusable UI components, making it easier to build dynamic and responsive web applications. Install React.js, a JavaScript library for building user interfaces.

**INSTALLATION**:

Folllow the below instrcutions,

Install Dependencies:

• Navigate into the cloned repository directory and install libraries:

**cd news-app-react npm install**

Start the Development Server:

• To start the development server, execute the following command:

**npm start**

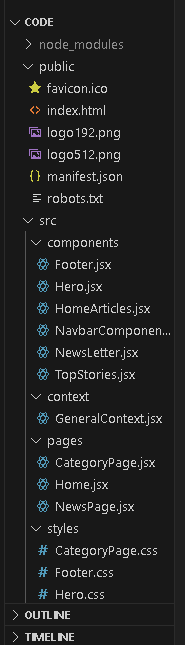
Access the App:

• Open your web browser and navigate to **http://localhost:3000.**

You should see the applications homepage, indicating that the installation and setup were successful. You have successfully installed and set up the application on your local machine. You can now proceed with further customization, development, and testing as needed.

**FOLDER STRUCTURE:**

The project follows a well-organized folder structure to maintain **scalability, readability, and reusability**.Below is an overview of each folder as follows,

**RUNNING THE APPLICATION:**

After completing the code, run the react application by using the command **“npm start” or “npm run dev” if you are using vite.js**

**Hero components** In the hero component, the trending news articles are displayed. It is to highlight them. Apart from that, the search bar is also available to search for various articles and categories.

**Popular categories** In the hero component, the trending news articles are displayed. It is to highlight them. Apart from that, the search bar is also available to search for various articles and categories.

**Newsletter** Staying informed is key! This section would act as a magnet for users who want to stay up-to-date on the latest news. A brief signup form with an email field would be presented, along with a clear call to action button like "Subscribe Now" or "Get Daily News Updates." With a simple click, users can join the InsightStream community and receive curated news delivered straight to their inbox.

**Category/Search result page** Finding the news you crave is effortless with InsightStream. This page displays a neatly organized list of articles matching your chosen category or specific search query. Each entry would provide a clear headline, a concise summary, and if available, an image to give you a quick glimpse into the story. To further refine your exploration, filters or sorting options might be available. Imagine narrowing down results by date, source, keyword, or other relevant criteria to pinpoint exactly what you're looking for.

**Redirected Article page** This is where you dive deep! The article page proudly displays the complete news story, retrieved directly from the original source. To keep you engaged and exploring related topics, the page might also suggest additional articles based on the current story. These suggestions can open doors to a world of interconnected information, allowing you to become a well-rounded news connoisseur.

**COMPONENT DOCUMENTATION:**

### ****Navbar Component**** (Navigation Bar)

**Purpose:**

* The Navbar component provides a structured navigation menu for users to switch between different news categories and the home page.
* It ensures easy access to various sections of the app by utilizing client-side routing.

**Props:**

* This component does not receive any props but utilizes React Router to navigate between different categories dynamically.

### ****News List Component**** (Displays a List of News Articles)

**Purpose:**

* The NewsList component is responsible for displaying multiple news articles on a page.
* It receives an array of articles and dynamically renders each article as a separate news card.

**Props:**

* It accepts an array of news articles, where each article contains details like the title, image, description, and URL.

### ****News Card Component(Reusable news article card)****

**Purpose:**

* The NewsCard is a reusable UI component that represents an individual news article in a visually structured format.
* It includes an image, title, short description, and a link to the full article.

**Props:**

* It receives details such as the **news title, image URL, description, and article URL**, making it adaptable for various sections of the app.

### ****Category Component**** (Displays News by Category)

**Purpose:**

* The Category component dynamically fetches and displays news articles based on the selected category.
* It extracts the category name from the URL and updates the news feed accordingly.

**Props:**

* This component does not take explicit props but uses **React Router’s URL parameters** to determine the selected category.

### ****Article Component**** (Displays Full News Article)

**Purpose:**

* The Article component is responsible for rendering the detailed view of a selected news article.
* It extracts the article ID from the URL and fetches the complete news content for display.

**Props:**

* No explicit props are passed to this component. Instead, it dynamically retrieves the required news details based on the selected article.

## **REUSABLE COMPONENTS:**

### ****Button Component****

**Purpose:**

* A customizable button that can be used for different actions across the app, such as **"Read More," "Load More," or "Filter by Category."**
* Ensures consistency in button styling and behavior.

**Props:**

* Accepts properties like **label (text on the button), onClick (click handler), and style configurations.**

### ****Loader Component**** (Displays Loading Animation)

**Purpose:**

* Displays a loading indicator when fetching news articles from an API.
* Enhances user experience by showing visual feedback while data is being retrieved.

**Props:**

* This component does not take specific props but is conditionally rendered when data is loading.

### ****Search Bar Component**** (Allows Users to Search for News)

**Purpose:**

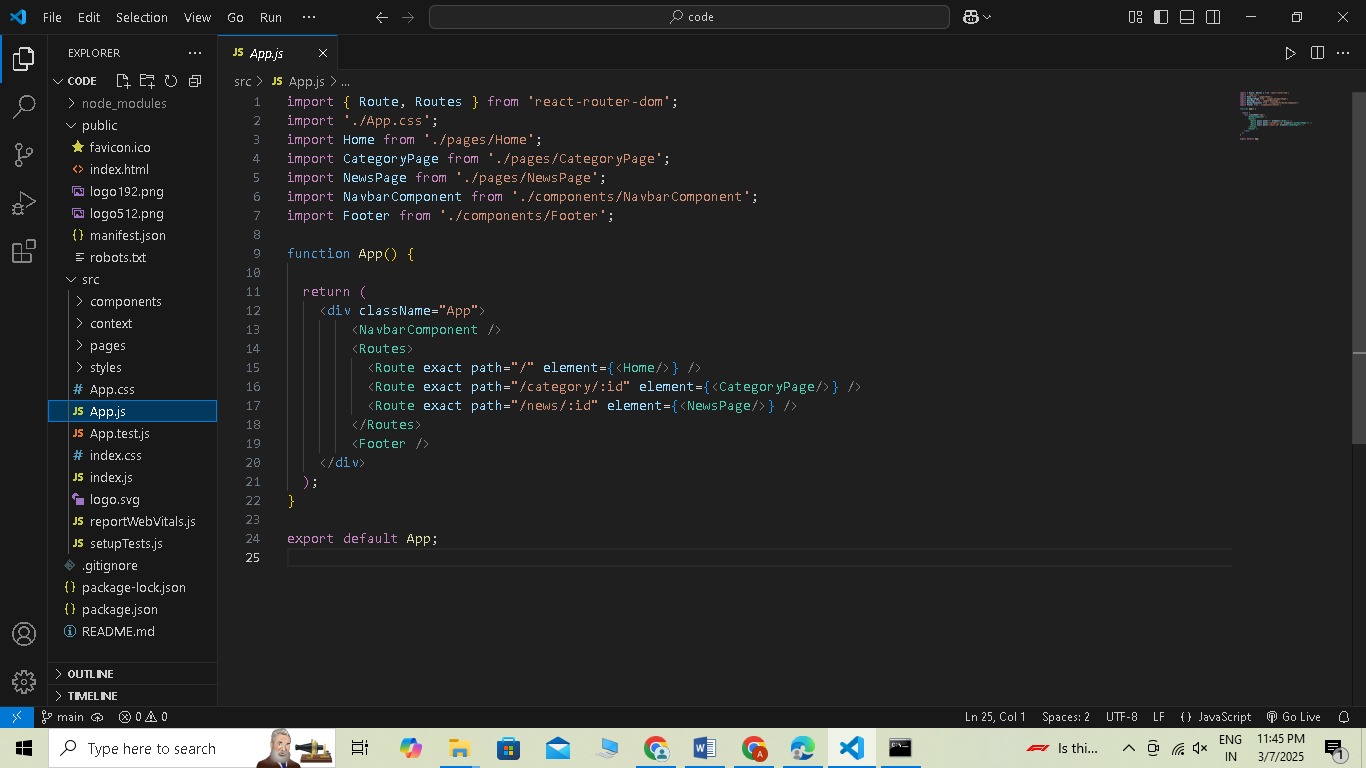
* Enables users to **search for news articles** based on keywords.
* Triggers a search query and filters the news articles accordingly.

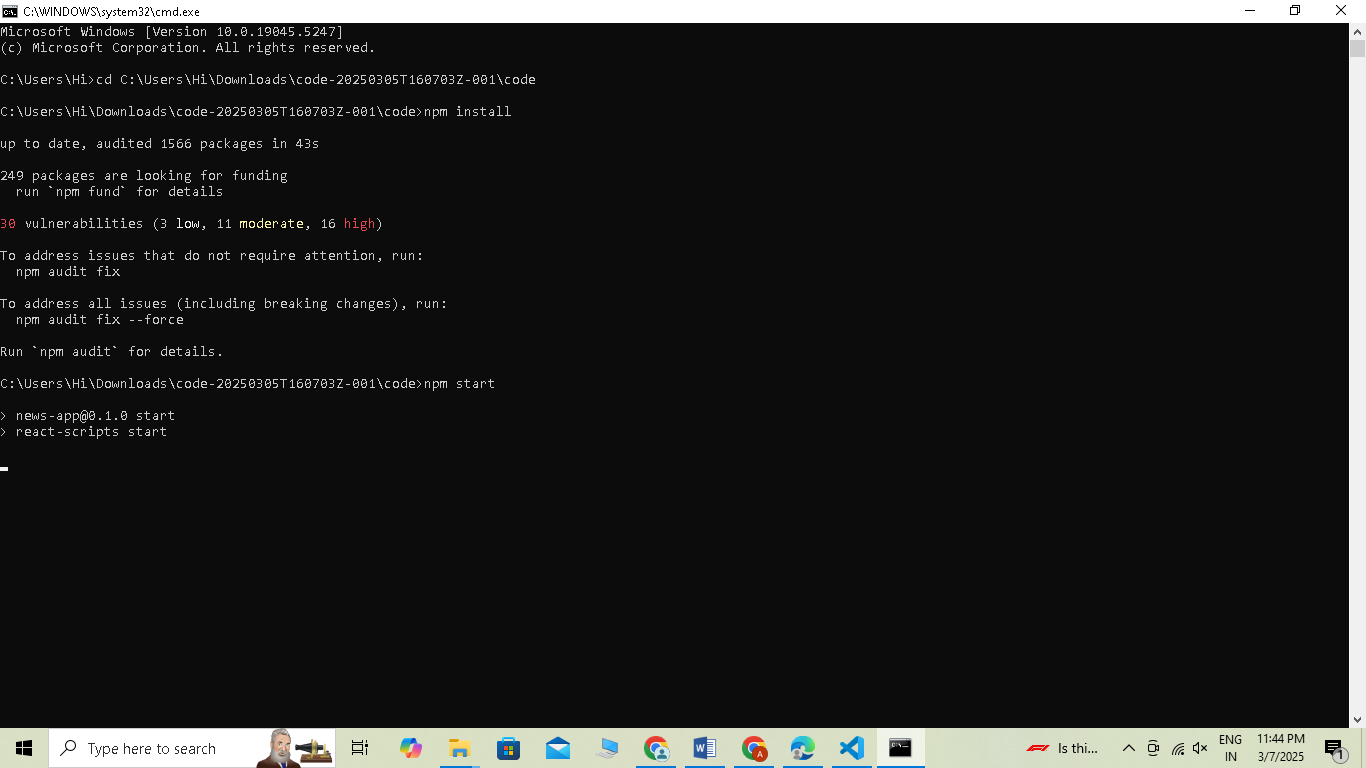
**Props:**

* Accepts a **search input value and a function to handle search queries dynamically.**

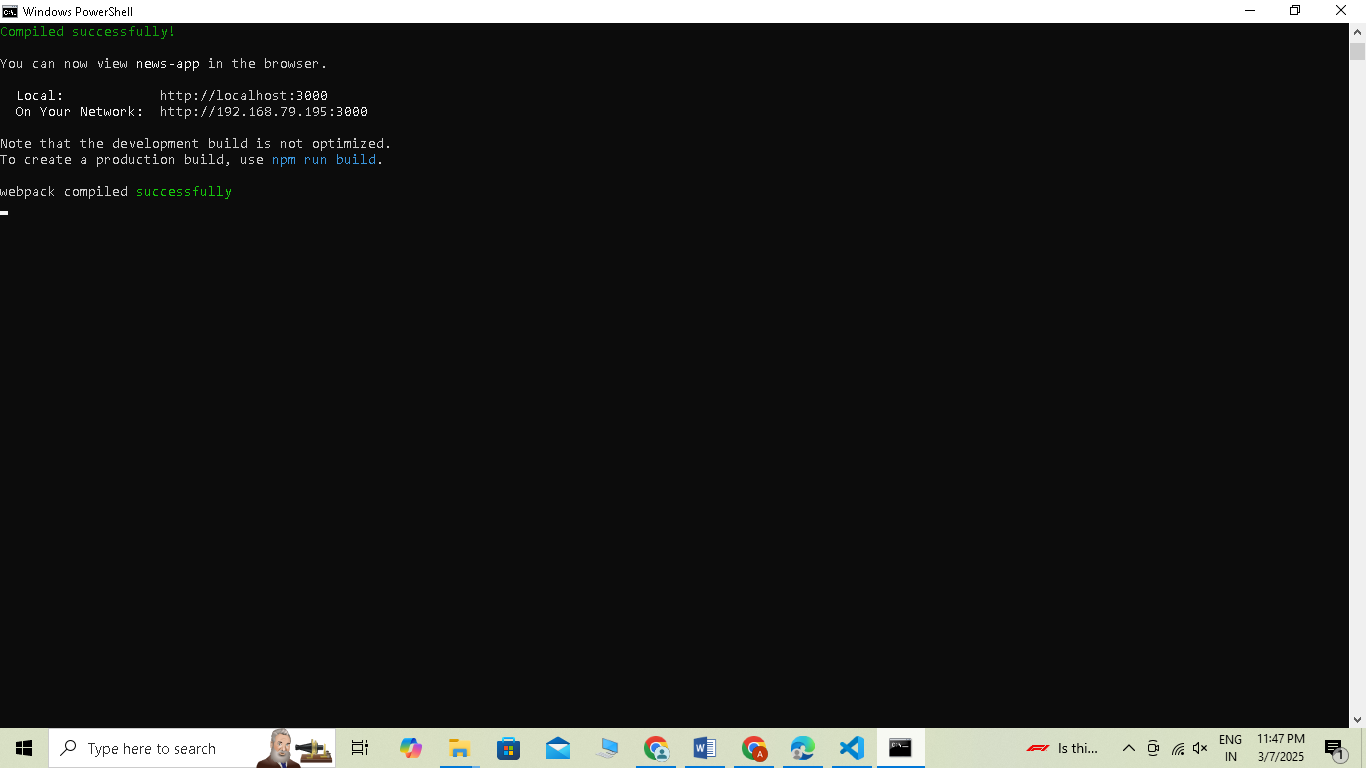
SCREENSHOTS:

CODINGS:

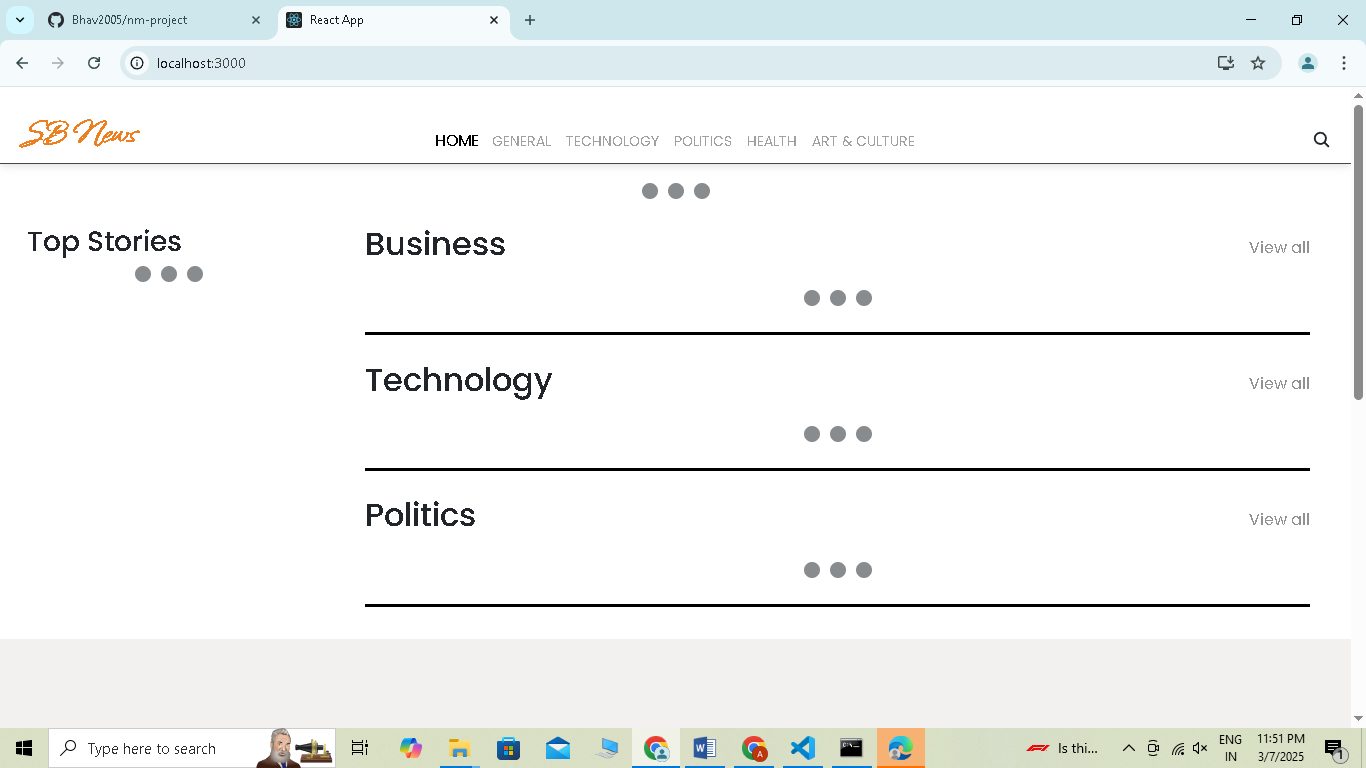


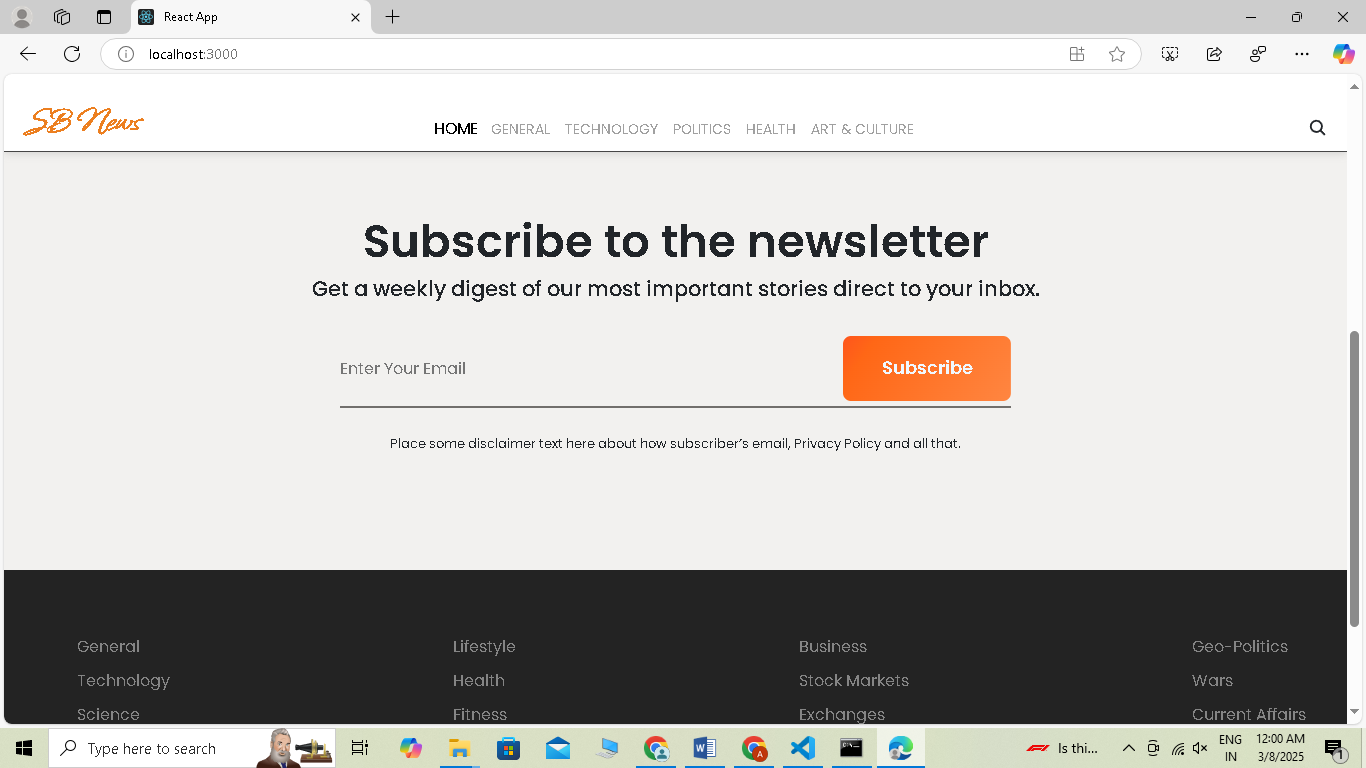
RUNNING npm COMMANDS:

COMPLIED SUCCESSFULLY:



HOME PAGE:





**STYLING**:

**HTML, CSS, and JavaScript:** Basic knowledge of HTML for creating the structure of your app, CSS for styling, and JavaScript for client-side interactivity is essential.

Here are some necessary tools used:

React Js

React Router Dom

React Icons

Bootstrap/tailwind css

Axios

**KNOWN ISSUES:**

#### **Issue: Home Page Stuck on "Loading"**

#### The homepage continuously displays a loading state and does not render the news articles properly. This issue occurs when the API call to fetch news data does not resolve or fails due to network issues, incorrect API configurations, or missing state updates.

**FUTURE ENHANCEMENTS:**

To ensure continuous improvement and provide a seamless user experience, several enhancements are planned for future updates of the News App. These enhancements will focus on **performance optimization, improved state management, enhanced user experience, and accessibility features.**

One of the key areas of improvement is **performance optimization.** Currently, the app may experience delays in loading news articles due to repeated API calls. To address this, we plan to implement **lazy loading** for images and components, ensuring faster page load times. Additionally, **caching mechanisms** will be introduced to minimize redundant API requests and improve overall efficiency. Features like **pagination or infinite scrolling** will also be considered to handle large volumes of news articles more effectively.

To enhance **state management,** we may transition from the Context API to a more robust solution like **Redux,** allowing for better scalability when handling large datasets. Proper **error handling mechanisms** will be integrated to prevent issues like infinite loading states and improve debugging.

From a **user experience** perspective, several features will be introduced to make the app more engaging and user-friendly. A **search functionality** will allow users to quickly find specific news articles, while **category-based filtering** will help them explore topics of interest effortlessly. Additionally, a **bookmarking feature** will enable users to save their favorite news articles for later reading.

The app’s **theming and UI** will also be improved with a **dark mode/light mode toggle,** providing a customizable experience based on user preference. A more consistent **design system** will be implemented to enhance the overall aesthetics and maintain a uniform look across different devices.

Furthermore, **accessibility improvements** will be a major focus. Features such as **keyboard navigation support, screen reader compatibility, and voice search** will be integrated to make the app more inclusive for visually impaired users. This aligns with the goal of making the platform accessible to a wider audience.

Lastly, for **deployment and scalability**, we aim to host the app using **serverless solutions** such as Firebase or Vercel, ensuring smooth performance even under high traffic. SEO enhancements will also be considered to improve the discoverability of news articles on search engines.