**InsightStream: Navigate the News Landscape**

**1. Introduction**

**1.1 Project Title:**

**InsightStream: Navigate the News Landscape2**

**1.2 Team Details:**

**Team ID: SWTID1741529681155067**

|  |  |  |
| --- | --- | --- |
| **Team Leader** | Abinaya sri A | [aabiazhagesan@gmail.com](mailto:aabiazhagesan@gmail.com) |

|  |  |  |
| --- | --- | --- |
| **Team Member 1** | Abinaya sri A | [aabiazhagesan@gmail.com](mailto:aabiazhagesan@gmail.com) |
| **Team Member 2** | Dhanashree N | [ds06022005@gmail.com](mailto:ds06022005@gmail.com) |
| **Team Member 3** | Yazhini R | [yazhinir11b@gmail.com](mailto:yazhinir11b@gmail.com) |
| **Team Member 4** | Geethanjali S | [geethuma.2307@gmail.com](mailto:geethuma.2307@gmail.com) |
| **Team Member 5** | Krithika R G | [krithikaguna2210@gmail.com](mailto:krithikaguna2210@gmail.com) |

**2. Project Overview**

InsightStream is a web application designed to redefine how people discover and consume news. It aims to provide an intuitive interface, dynamic search, and a vast range of news categories for all types of users, from casual readers to seasoned information professionals. The app seeks to solve the problem of information overload by offering a streamlined and personalized news experience

**Use Case Scenario:** Imagine rushing home after work and wanting a quick update on world events. InsightStream offers breaking headlines, in-depth articles, and diverse categories in an easily navigable format.

**Key Frontend Features:**

* **News from API Sources:** Accesses a vast library of global news spanning various categories and interests, ensuring well-rounded coverage of current affairs.
* **Visual News Exploration**: Discovers breaking stories and explores different news categories through curated image galleries, enhancing the visual appeal of news discovery.
* **Intuitive Design:** Navigates the application seamlessly with a clean, modern interface designed for optimal user experience and clarity in information presentation.
* **Advanced Search Feature:** Easily accesses news articles on specific topics through a powerful search feature, providing users with tailored news content based on their interests.
* **Dynamic UI Elements:** Interactive elements like the search bar in the hero component, newsletter subscription, and categorized display of articles enhance user experience.
* **Category/Search Result Page:** Displays a neatly organized list of articles matching the chosen category or specific search query, complete with headlines, summaries, and images for quick story insight.

**3. Architecture**

* **Frontend:** React.js UI with components for navigation (Navbar), trending news (Hero), category listings, and a newsletter signup.
* **API Client:** Handles communication with news APIs via Rapid API.
* **Data Source:** External news APIs providing articles, headlines, and images.

**Component Structure:**

* **App.js:** Main application wrapper.
* **Components:** Reusable UI elements.
* **Pages:** Views like Home, Category, and Search Results.
* **Context:** (Potentially) Manages global state.
* **Styles:** CSS styling.

**Project Flow:**

1. User interacts with the React UI.
2. The UI requests data via the API client.
3. The API client uses Rapid API to retrieve data from news APIs.
4. Data is processed and displayed in the UI.

**4.Setup Instructions:**

✓ **Node.js and npm:**

Run JavaScript code.\* Download: <https://nodejs.org/en/download/>\* Install: <https://nodejs.org/en/download/package-manager/>

✓ **React.js:** Build the user interface.\* Create a new React app: npx create-react-app my-react-app (replace my-react-app)\* Go to the project directory: cd my-react-app\* Start the development server: npm start (access at <http://localhost:3000>)

✓ **HTML, CSS, and JavaScript:** Basic knowledge is essential.

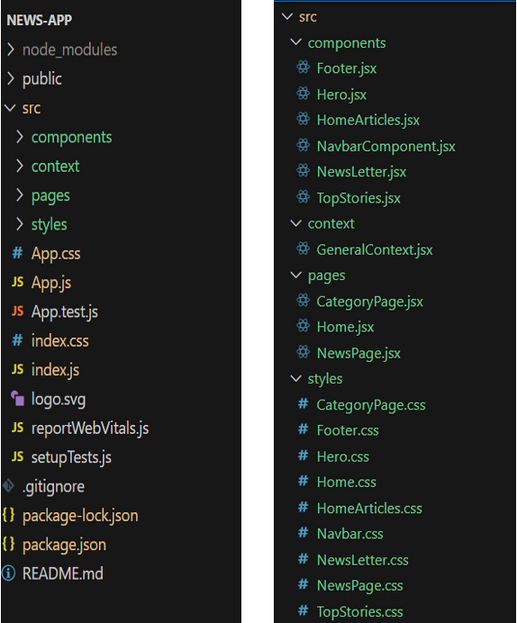
✓ **Version Control (Git):** Track code changes.\* Download: <https://git-scm.com/downloads>

✓ **Code Editor:** Choose your editor (VS Code, Sublime Text, WebStorm).\* Visual Studio Code: <https://code.visualstudio.com/download>\* Sublime Text: <https://www.sublimetext.com/download>\* WebStorm: <https://www.jetbrains.com/webstorm/download>

✓ **Get the Project Code:**\* Download from Google Drive: <https://drive.google.com/drive/folders/1tDoSwd-1I3HsPJ9_92MnZTUtteeda-hL?usp=sharing>

✓ **Install and Run:**\* Go to the project folder: cd news-app-react\* Install dependencies: npm install\* Start the app: npm start\* Open browser: http://localhost:3000

**5. Folder Structure**



In this project, we’ve split the files into 4 major folders, Components, Context, Pages and Styles. In the pages folder, we store the files that acts as pages at different URLs in the application. The components folder stores all the files, that returns the small components in the application. The context Api will be coded in the context folder. All the styling css files will be stored in the styles folder.

**6.Running the Application**.

1. **✔ Get the code:**
2. • Download the code from the drive link given below: <https://drive.google.com/drive/folders/1tDoSwd-1I3HsPJ9_92MnZTUtteeda-hL?usp=sharing>
3. **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.

**7. Component Documentation**

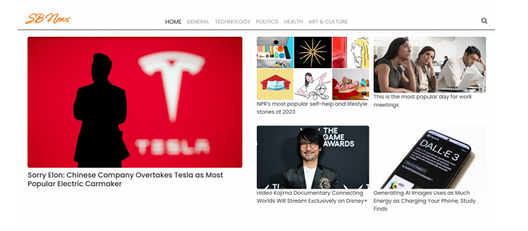
* **Navbar:** Site navigation, links to sections, may include search.
* **Hero:** Highlights trending news, includes search bar.
* **CategoryList:** Displays popular news categories (fetched from API).
* **Newsletter:** Allows users to subscribe with their email.
* **Home:** Main landing page (trending news, categories, newsletter).
* **Category Page:** Displays news articles within a specific category.
* **Search Results Page:** Displays news articles matching search terms.
* Explain commonly used UI elements:
* **Hero:** Spotlights trending news and includes a search bar.
* **Newsletter:** Allows users to subscribe with their email for news updates.
* **Category/Search Result Page:** Displays an organized list of articles with headlines, summaries, and images.

**8. State Management**

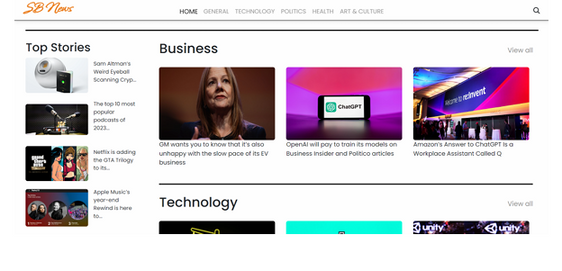
* **Context API:** Uses NewsContext.js (inferred) for global news data (trending, search results).
  + Provides data & fetch methods.
  + Accessed via useContext in components.
* **Local State:** Components use useState and useEffect.
  + useState: Manages component-specific data (search input, email).
  + useEffect: Used for fetching data on component mount/update.
* **No Redux:** Project does *not* appear to use Redux.
* **Main Pattern:** Combines Context (shared data) + Local State (UI data).

**9. User Interface**

* 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.

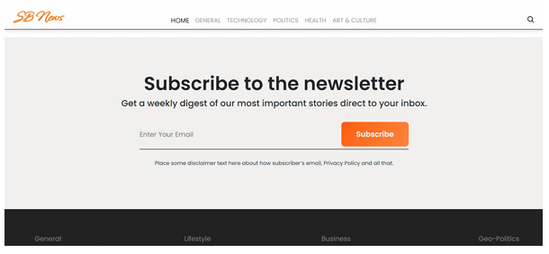


* Popular categories:
* Popular categories In the hero component, the trending news articles are displayed. It is to highlight them. Apart from that, the search bar is also available to search for various articles and categories.



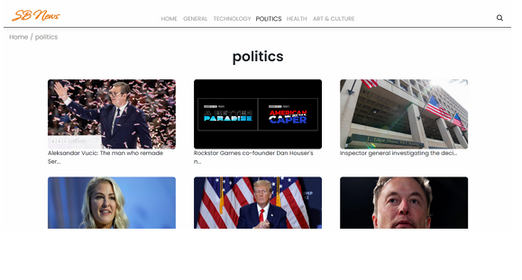
**Newsletter**

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**

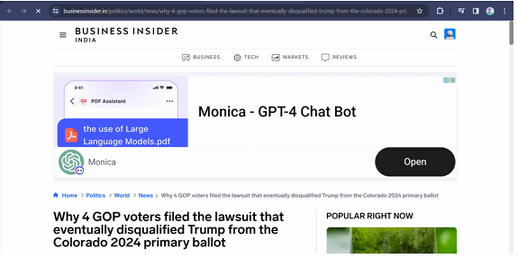
Finding the news you crave is effortless with InsightStream. This page displays a neatly organized list of articles matching your chosen category or specific search query. Each entry would provide a clear headline, a concise summary, and if available, an image to give you a quick glimpse into the story. 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.







**10. Styling**

**CSS Fr**Mention styling choices such as:

* **CSS:** Styling is done using CSS files.
* **Files:** All styling CSS files are stored in the styles folder.
* **Libraries (Choice):** Project is set up to use either **Bootstrap** or **Tailwind CSS** for pre-built styles and icons.

**11.Testing**

Given the API integrations for fetching news, testing likely involves verifying API calls return data correctly, ensuring accurate display of news content, and proper functionality of search and category filters. Here's a breakdown:

1. **Functionality Testing:** The code snippets suggest a focus on validating the functionality of fetching news articles. This could involve:

Confirming that API requests are successfully made to the news API endpoints using Axios.

Verifying that the data returned from the API is correctly parsed and displayed in the UI.

Testing search functionality to ensure relevant results are returned based on user queries.

Checking that news articles are correctly categorized and displayed on category pages.



1. **API Testing:** Based on the importance of retrieving data correctly from a third party API, testing to ensure data can be collected from rapid API needs to be tested and verified.
3. **Manual Testing** - The use of manual testing of the app functionality to ensure all components are tested and working.

**12. Demo Video**

* A link to a **live demo** :- https://drive.google.com/file/d/1i8y09FiMk7QM0akH3my10OBWqXH-8dNh/view?usp=sharing

**13. Known Issues**

Document any **current bugs or limitations**, such as:

* **API Problems:** Relies on external APIs, which *could* fail or change. Code includes error handling.
* **Responsiveness:** *Possible* responsiveness issues if Bootstrap/Tailwind not used correctly. (No explicit evidence).

**14. Future Enhancements**

Based on the project description, several enhancements could improve InsightStream:

* **Optimized Accessibility:** Enhance support for screen readers and keyboard navigation.
* **Performance Improvements:** Utilize memoization, lazy loading, and efficient rendering practices.
* **Real-time Updates:** Implementing real-time updates using WebSockets.
* **Internationalization Support:** Adding internationalization support.