1. INTRODUCTION

PROJECT TITLE : INSIGHTSTREAM - NAVIGATE THE NEWS LANDSCAPE

Team Members :

NAME : R.MERLIN JOYCE (TEAM LEAD)

EMAIL ID : 212201865@newprincearts.edu.in

NAME : M.DEEPA

EMAIL ID : 212201847@newprincearts.edu.in

NAME : V.PARKAVI

EMAIL ID : 212201869@newprincearts.edu.in

NAME : M.MEENA

EMAIL ID : 212201864@newprincearts.edu.in

2. PROJECT OVERVIEW

Purpose:

InsightStream is designed to redefine news discovery and consumption by offering an intuitive, user-friendly platform with comprehensive news coverage tailored for diverse users.

Key Goals:

✔ Provide a seamless and intuitive user experience.

✔ Offer dynamic search options for personalized news access.

✔ Leverage modern web technologies for performance and scalability.

✔ Deliver real-time news updates from trusted API sources.

Features:

✔ News from API Sources: Access a vast library of news from different categories.

✔ Visual News Exploration: Engaging UI with image-based news previews.

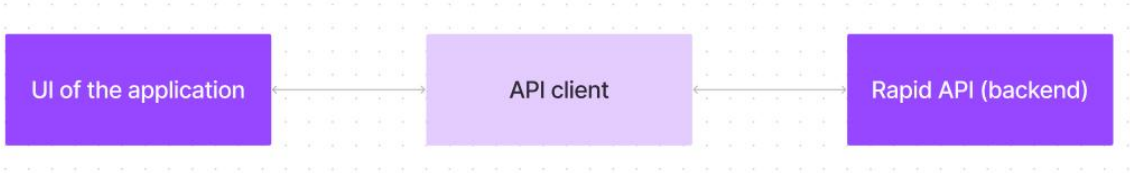
✔ Advanced Search: Quickly find relevant news articles based on keywords.

✔ Personalized Experience: Save and access preferred news articles.

✔ Responsive Design: Optimized for all devices.

3. ARCHITECTURE

Technical Architecture:



The user experience starts with the InsightStream web application's UI, likely built with a

framework like React or Vue.js for a smooth, single-page experience. This UI interacts with

an API client specifically designed for InsightStream. This client handles communication with

the backend, but with a twist: it leverages Rapid API, a platform providing access to various

external APIs. This suggests InsightStream might integrate external data feeds or

functionalities through Rapid API, enriching the user experience without building everything from scratch.

Component Structure

✔ App.js - Root component handling layout and routing.

✔ Navbar.js - Manages navigation between pages.

✔ Home.js - Displays trending news and featured categories.

✔ Search.js - Implements dynamic search for articles.

✔ NewsList.js - Fetches and displays news articles.

✔ NewsCard.js - Individual news article preview component.

✔ NewsDetails.js - Shows full article details with related news suggestions.

✔ Footer.js - Provides site links and social media navigation.

State Management in InsightStream

InsightStream uses the Context API for global state management, ensuring efficient data sharing across components.

Routing in InsightStream

✔ Home (`/`) - Displays top news and trending stories.

✔ Category (`/category/:name`) - Shows articles under a specific category.

✔ Search (`/search/:query`) - Displays articles matching the search query.

✔ Article (`/article/:id`) - Shows detailed news content.

4. SETUP INSTRUCTIONS

Ensure you have the following prerequisites installed before setting up the project.

PRE-REQUISITES:

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.

● 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.

✔ **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.

✔**Version Control**: Use Git for version control, enabling collaboration and tracking

changes throughout the development process. Platforms like GitHub or Bitbucket can

host your repository.

• Git: Download and installation instructions can be found at:

https://git-scm.com/downloads

✔ **Development Environment**: Choose a code editor or Integrated Development

Environment (IDE) that suits your preferences, such as Visual Studio Code, Sublime

Text, or WebStorm.

• Visual Studio Code: Download from https://code.visualstudio.com/download

• Sublime Text: Download from https://www.sublimetext.com/download

• WebStorm: Download from https://www.jetbrains.com/webstorm/download

5. RUNNING THE APPLICATION

To install and run the Application project from google drive:

Follow below steps:

✔ **Get the code:**

• Download the code from the drive link given below:

https://drive.google.com/drive/folders/1tDoSwd-1I3HsPJ9_92MnZTUtteeda-hL?usp=sharing

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.

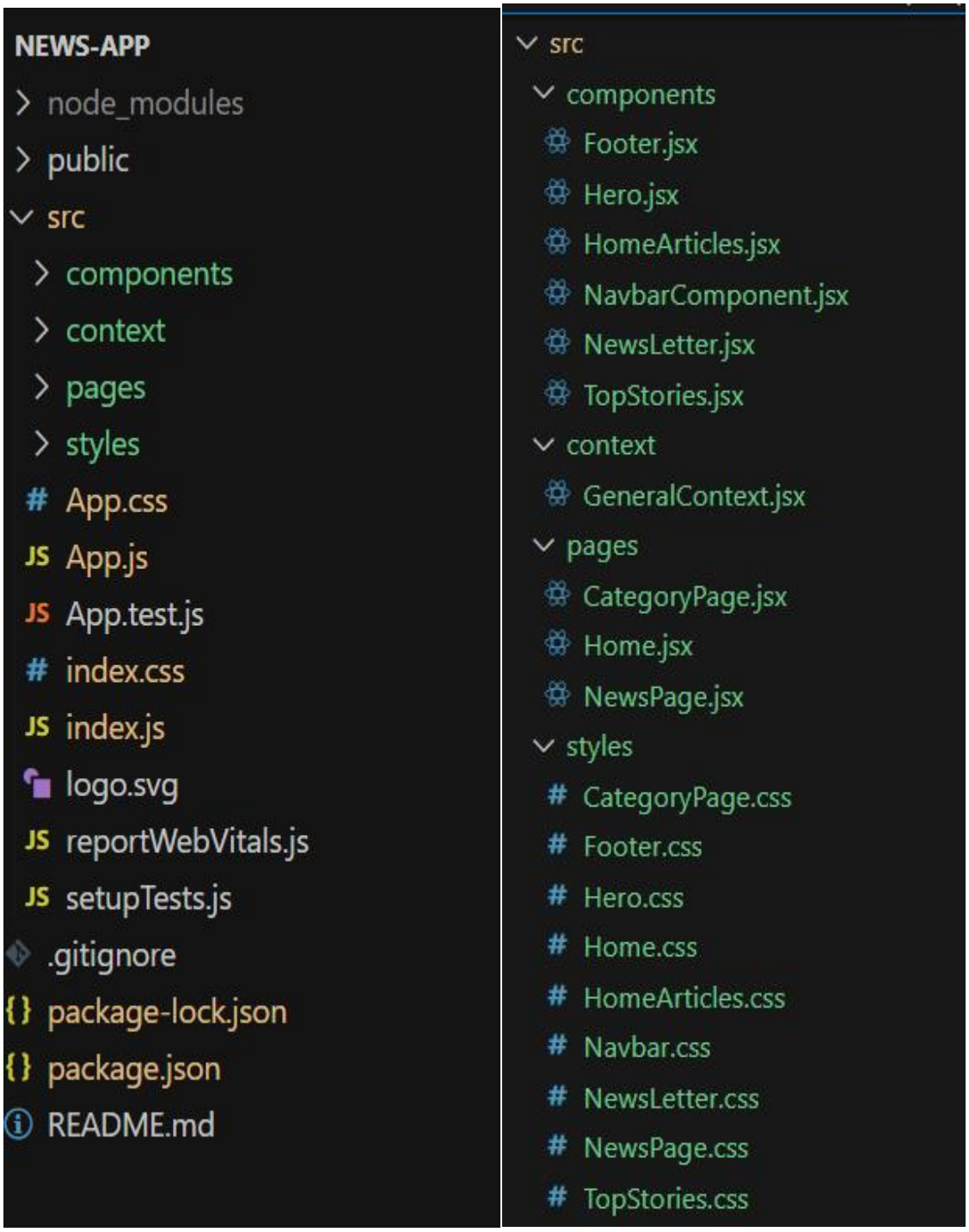
• You should see the applications homepage, indicating that the installation

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

.**Project 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.

Project Flow:

Project demo:

Before starting to work on this project, let’s see the demo.

Demo link:

https://drive.google.com/file/d/1i8y09FiMk7QM0akH3my10OBWqXH-8dNh/view?usp=sharing

Use the code in:

https://drive.google.com/drive/folders/1tDoSwd-1I3HsPJ9_92MnZTUtteeda-hL?usp=sharing

Milestone 1: Project setup and configuration.

● **Installation of required tools**:

To build InsightStream, we'll need a developer's toolkit. We'll use React.js for

the interactive interface, React Router Dom for seamless navigation, and Axios to

fetch news data. For visual design, we'll choose either Bootstrap or Tailwind CSS for

pre-built styles and icons.

Open the project folder to install necessary tools. In this project, we use:

o React Js

o React Router Dom

o React Icons

o Bootstrap/tailwind css

o Axios

● For further reference, use the following resources

o https://react.dev/learn/installation

o https://react-bootstrap-v4.netlify.app/getting-started/introduction/

o https://axios-http.com/docs/intro

o https://reactrouter.com/en/main/start/tutorial

Milestone 2: Project Development

❖ Setup the Routing paths

Setup the clear routing paths to access various files in the application.

```
<div className="App">
    <NavbarComponent />
    <Routes>
      <Route exact path="/" element={<Home/>} />
      <Route exact path="/category/:id" element={<CategoryPage/>} />
      <Route exact path="/news/:id" element={<NewsPage/>} />
    </Routes>
    <Footer />
</div>
```

❖ Develop the Navbar and Hero components

❖ Code the popular categories components and fetch the categories from ***newsapi***.

❖ Also, add the trending news in the home page.

❖ Additionally, we can add the component to subscribe for the newsletter and the

footer.

❖ Now, develop the category page to display various news articles under the different

categories.

Important Code snips:

With the API request, we fetch the trending news articles.

```
const fetchTopNews = async () => {
  try {
    const response = await axios.get("https://newsapi.org/v2/everything?q=popular&apiKey=37306aca596542f0a8402978de3d4224");
    setTopNews(response.data.articles);
  } catch (error) {
    console.error(error);
  }
}
```

The code snippet shows a function written in Python called fetchTopNews that

fetches news articles from an API. Here's a breakdown of the code:

Async function fetchTopNews:

The code defines an asynchronous function named fetchTopNews. An

asynchronous function is used to handle asynchronous operations, such as making

API requests that take time to complete.

try...catch block:

● The try...catch block is used to handle the API request.

● The try block contains the code that attempts to fetch data from the API using

axios.get.

● axios is an external Python library for making HTTP requests. If you don't

already use Axios in your project, you'll need to install it using a package

manager like pip.

● The .get method makes a GET request to the specified URL.

API URL:

The URL used in the API request is

'https://newsapi.org/v2/everything?q=popular&apiKey=37306aca596542f0a840297

8de3d4224'.

This is likely a specific API endpoint that returns popular news articles. You might

need to replace this URL with the actual endpoint you want to use depending on the

API you're using. Replace '37306aca596542f0a8402978de3d4224' with a placeholder

instructing users to replace it with their own API key.

Error Handling (catch block):

The catch block handles any errors that might occur during the API request. If there's

an error, it's logged to the console using console.error(error).

Setting State (then block not shown):

The .then method (not shown in the code snippet) is likely used to process the

fetched data after a successful API request.

In this case, it likely updates a state variable named topNews (based on the function

name fetchTopNews) with the fetched news articles. This state variable might be

used to display the news articles in a user interface.

Fetching news by search/category

With the specific category or search keyword, we use API request to fetch all the

news articles related to that.

The code snippet shows a function called get_news_articles that fetches news

articles from a news API. Here's a breakdown of the code:

Imports:

The code starts by importing the requests library. The requests library is a popular

Python library for making HTTP requests. If you don't already have it installed in your

project, you can install it using pip install requests.

API Key:

The line API_KEY = 'YOUR_API_KEY' defines a variable named API_KEY and assigns it

a placeholder value 'YOUR_API_KEY'. You should replace this with a placeholder

instructing users to replace it with their own API key obtained from the news API

provider they want to use.

Function Definition (get_news_articles):

The code defines a function named get_news_articles that takes two parameters:

● query: This parameter is likely a string representing the search query for news

articles.

● source: This parameter is likely a string representing the news source (e.g.,

'bbc-news', 'cnn').

Building the API Request URL:

The line url = f'https://newsapi.org/v2/everything?q={query}&apiKey={API_KEY}'

constructs the URL for the API request using a formatted string literal (f-string).

The URL includes the following parts:

● Base URL: https://newsapi.org/v2/everything● Query parameters:

● q: This parameter is set to the query argument passed to the function.

● apiKey: This parameter is set to the API_KEY variable, which should

contain the user's API key.

Making the API Request (requests.get):

The line response = requests.get(url) sends a GET request to the API URL constructed

earlier. The requests.get function from the requests library is used to make the HTTP

request. The response from the API is stored in the response variable.

Error Handling (try...except block):

● The try...except block is used to handle potential errors during the API

request.

● The try block contains the code that attempts to fetch data from the API using

requests.get(url).

● The except block handles any exceptions that might occur during the request,

such as network errors or invalid API responses. In this case, it prints an error

message to the console using print(f'Error fetching news articles: {e}').

6. COMPONENT DOCUMENTATION

Key Components

✔ Navbar: Handles site-wide navigation.

✔ Hero Section: Displays trending news with a search bar.

✔ NewsCard: Represents an individual news item.

✔ NewsList: Fetches and displays multiple articles.

✔ NewsDetails: Shows full article content.

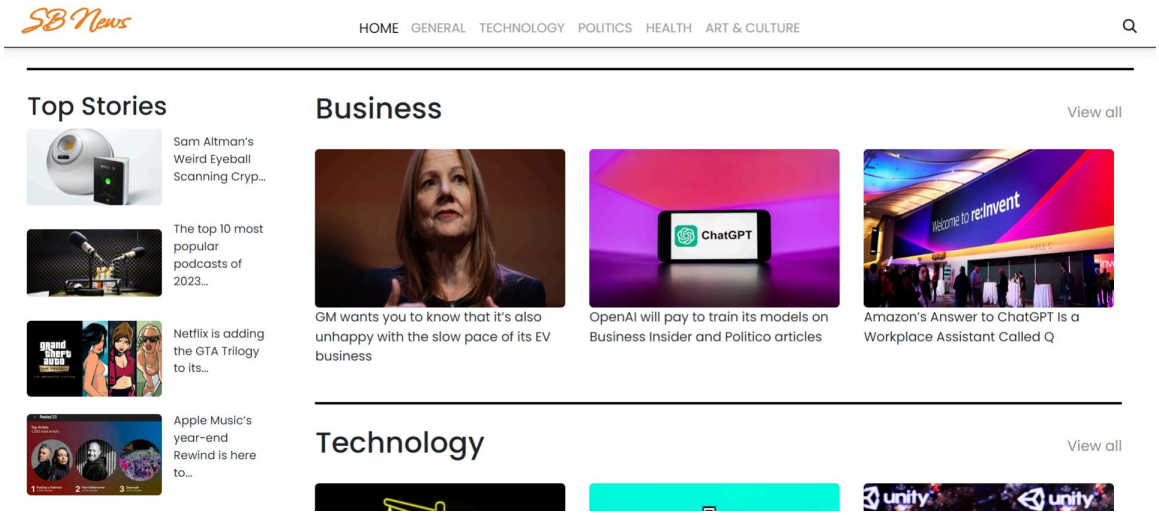
✔ Footer: Provides social media and contact links.

7. USER INTERFACE

Screenshots and descriptions of UI components such as homepage, categories, search results, and article pages.

Scenario 1: Browsing Categories

A user is interested in sports news. They navigate to the 'Sports' category and see the latest news related to various sports events.



8. STYLING

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

9. TESTING

Testing Strategy
✔ Unit Testing: Jest & React Testing Library

✔ Integration Testing: Ensuring API integration works correctly

Code Coverage
Ensuring high test coverage with Jest.

Testing - Sample Test Cases

Test Case 1: Search Functionality
Scenario: The user enters a search query, and related news articles should be displayed.

Expected Result: The app fetches and shows relevant news articles based on the query.

Test Case 2: Navigation Bar
Scenario: The user clicks on 'Technology' in the navigation bar.

Expected Result: The app navigates to the Technology category and displays relevant articles.

10. SCREENSHORTS OR DEMO

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

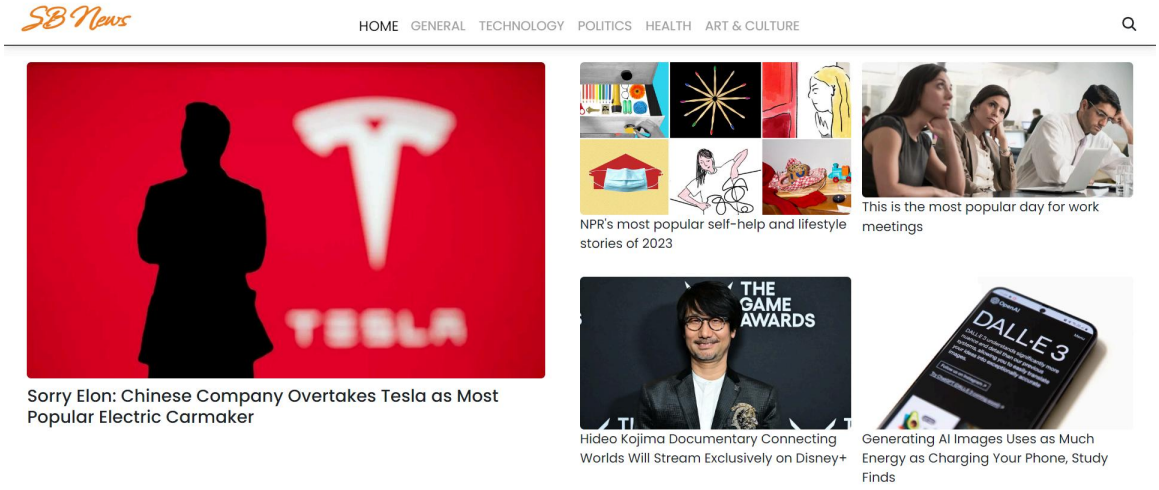
Here are some of the screenshots of the application.

Hero components

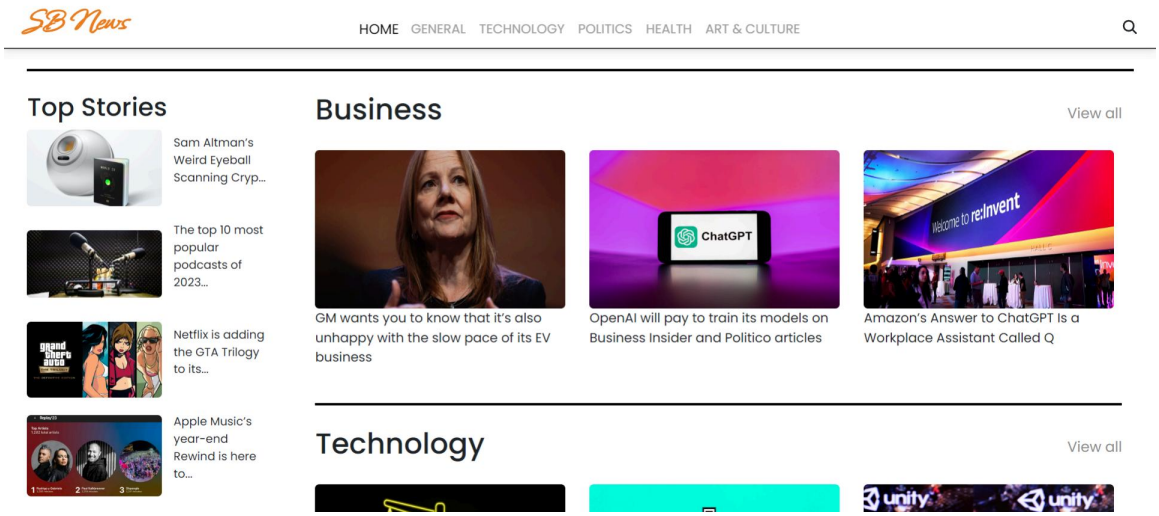
In the hero component, the trending news articles are displayed. It is to highlight

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

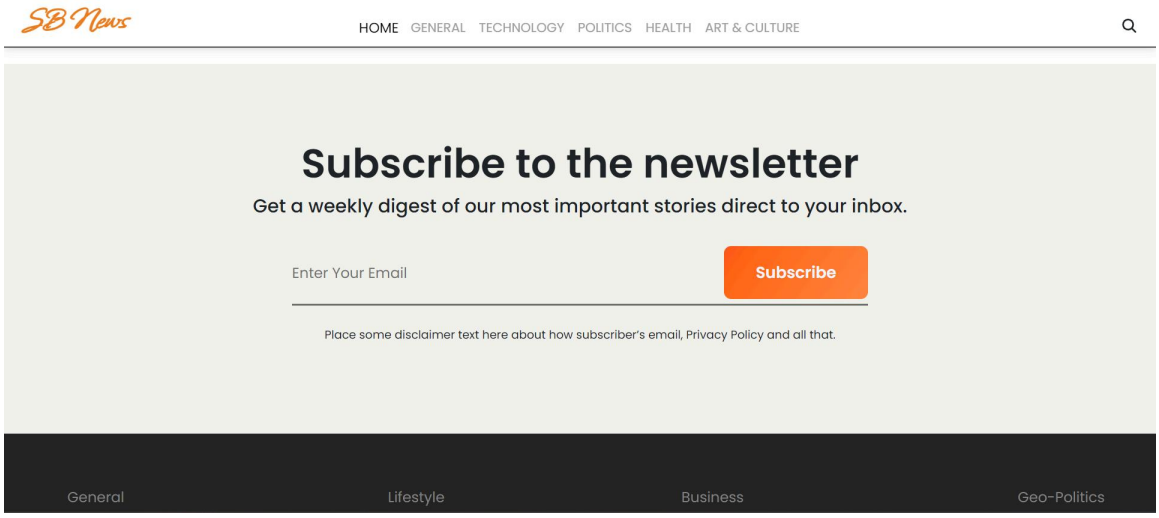
to stay up-to-date on the latest news. A brief signup form with an email field

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

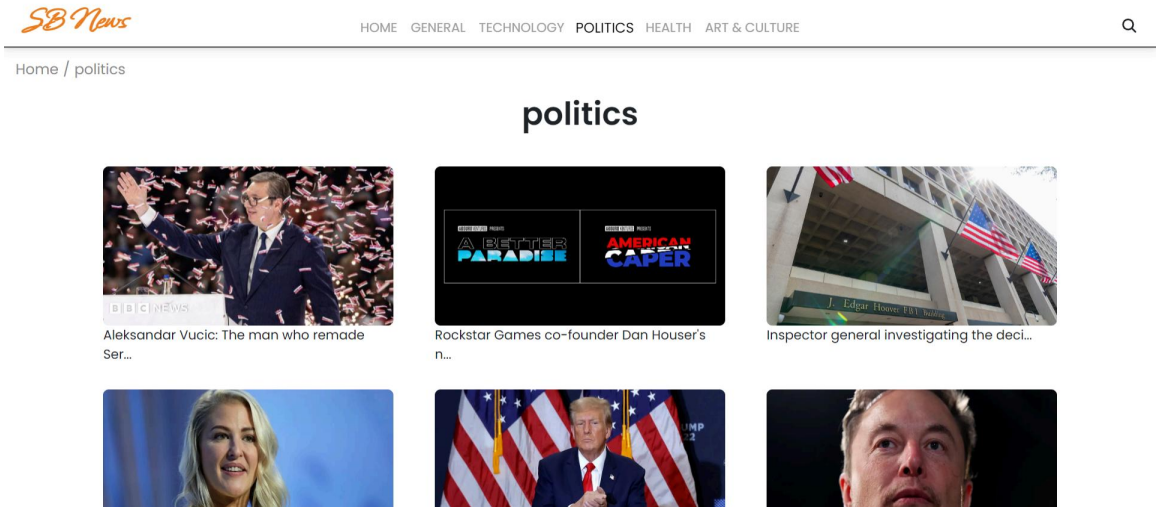
neatly organized list of articles matching your chosen category or specific search

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 narrowingdown results by date, source, keyword, or other relevant criteria to pinpoint

exactly what you're looking for.



Redirected Article page

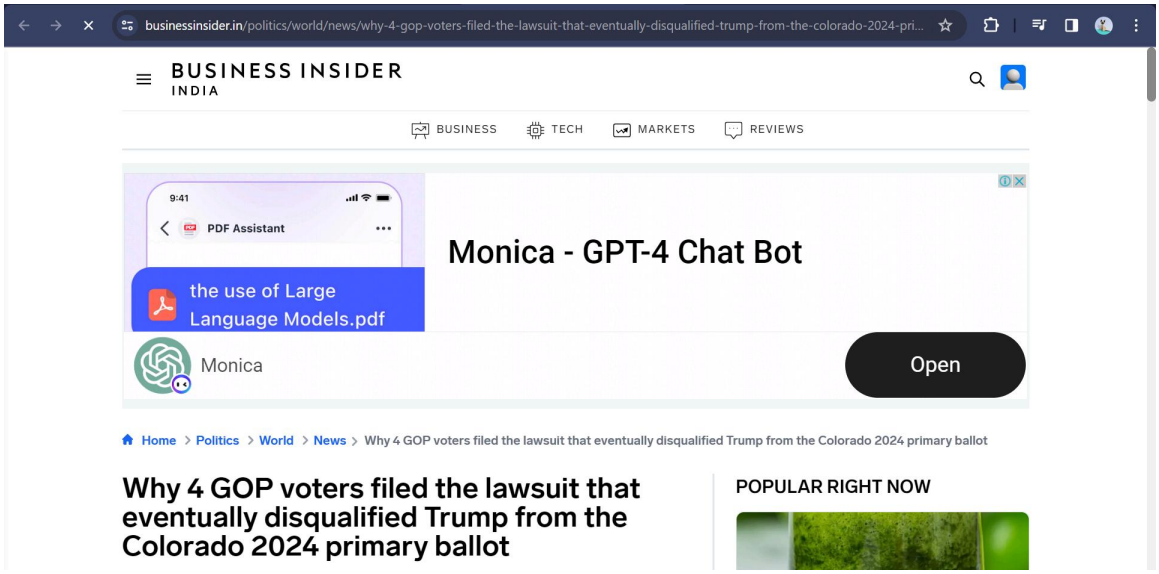
This is where you dive deep! The article page proudly displays the complete

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.



11. KNOWN ISSUES

✔ API rate limits might affect news fetch performance.

✔ Some articles may not have images, leading to inconsistent UI.

✔ Page refresh on dynamic routes might cause navigation issues.

12. FUTURE ENHANCEMENTS

✔ User Authentication: Allow users to save and bookmark articles.

✔ Customizable News Feed: Users can personalize their news categories.

✔ Push Notifications: Alerts for breaking news updates.

✔ Dark Mode & UI Enhancements for better readability.

✔ Progressive Web App (PWA) support for offline access.