Naan Mudhalvan project

InsightStream: Navigate the News

**Team Members:**

Indrajith M (Team Leader) [Email Id: mindrajith29@gmail.com]

Kannan P [Email Id: venom246456@gmail.com]

Akash B [Email Id: aa4929474@gmail.com]

Arun M [Email Id: arunm0300@gmail.com]

Kishore Raj A [Email Id: kishoreraj9087089698@gmail.com]

PERI COLLEGE OFARTS AND SCIENCE: MANNIVAKKAM

**ABSTRACT:**

Insight stream is an innovative real-time news tracking application developed using React.js and the GNews API. This app provides users with up-to-date news articles from various sources, allowing them to stay informed on global events, trending topics, and personalized news categories. By leveraging the power of React.js and insight stream offers a seamless user experience, efficient state management, and dynamic content updates. Additionally, the integration of the GNews API ensures accurate and reliable news data, making insight stream a valuable tool for news enthusiasts and researchers.

**Tools:** React.js, CSS, GNews API

**I.INTRODUCTION:**

A news application is a software platform designed to provide users with real-time access to the latest news articles from various sources. These apps offer diverse features, including news categorization, trending topics, personalized feeds, and seamless navigation, allowing users to stay informed about global events. With the increasing demand for reliable and up-to-date news consumption, this project introduces a cutting-edge news tracking application developed using React.js and the GNews API.

The goal of this application is to provide users with a user-friendly and feature-rich platform to track, explore, and read news articles in real-time. By leveraging the power of React.js and the GNews API, this application ensures an intuitive user interface, seamless data management, and access to accurate and up-to-date news information.

React.js, a popular JavaScript library for building user interfaces, serves as the foundation for the application's front-end development. Its component-based architecture enables the creation of reusable UI elements, resulting in a modular and scalable application structure. React.js also ensures a responsive and interactive user experience, allowing users to navigate through the app smoothly.

The GNews API serves as the primary data source for the application, providing real-time and comprehensive news data. By integrating the GNews API, users gain access to a wide range of news articles from trusted sources, covering topics such as politics, technology, business, health, sports, and entertainment. This integration ensures the accuracy and reliability of the news, enabling users to stay updated with the latest happenings worldwide.

**2. PREREQUISITES**

Before setting up the project, ensure you have the following installed:

1.Node.js (v14 or later)

2.npm or yarn

3.A GNews API Key (Sign up at [GNews.io](https://gnews.io/))

4.Basic knowledge of React.js

**II.Literature Review:**

**1. Overview of Digital News Aggregation :**

The evolution of news consumption has undergone a massive transformation in recent years, shifting from traditional sources like newspapers, television, and radio to online digital platforms. The availability of the internet and smart devices has led to an explosion of information, allowing people to access news from around the world in just a few clicks. Digital news aggregation platforms have emerged as an essential tool in this ecosystem, providing users with centralized access to diverse sources and eliminating the need to visit multiple websites. These platforms operate by collecting news articles from various publishers, blogs, and agencies and presenting them in a structured manner, ensuring that users stay informed with real-time updates. The primary benefit of such aggregation is that it provides a balanced view of current affairs, offering news from different sources, political perspectives, and geographical locations. However, with this shift comes new challenges, such as news credibility, information overload, and potential algorithmic bias. As the demand for instantaneous and reliable news grows, developers and organizations need to build solutions that not only provide fast updates but also ensure accuracy, transparency, and trustworthiness.

**2. Role of APIs in Real-Time News Fetching :**

The backbone of any real-time news application is the ability to fetch and display live news updates efficiently. This is made possible by Application Programming Interfaces (APIs), which enable developers to integrate third-party news sources into their applications. One of the most commonly used APIs for news aggregation is the GNews API, which provides structured access to global news sources, allowing applications to fetch articles based on keywords, categories, publication dates, and sources. The API supports multiple languages, ensuring that users receive news in their preferred linguistic format, making it a highly versatile tool for global audiences. The real-time nature of APIs eliminates the need for manual updates, ensuring that users always have access to the latest news articles as they are published. Additionally, APIs play a crucial role in filtering content, allowing developers to sort articles based on relevancy, popularity, and sentiment analysis. However, there are challenges associated with API-based news aggregation, such as rate limits, dependency on third-party providers, and the need for proper API key management to prevent unauthorized access. When integrating an API into a news platform, developers must ensure efficient data fetching, optimized API requests, and compliance with content licensing policies to provide a seamless experience without violating intellectual property laws.

**3. Challenges in Digital News Platforms**

Despite the advantages of digital news aggregation, there are several challenges that developers and content providers must address. One of the most pressing concerns is the spread of fake news and misinformation. With the rise of social media and digital publications, unverified or misleading information can quickly go viral, leading to public misinformation and potential harm. To combat this, news platforms must prioritize fact-checking mechanisms, source verification, and AI-driven content filtering. Another major challenge is information overload, where users are bombarded with excessive news updates, leading to fatigue and difficulty in distinguishing important updates from trivial content. Effective categorization, personalization, and filtering mechanisms are necessary to ensure that users only see relevant news. Furthermore, news bias and credibility remain a concern, as different news outlets often have political or ideological inclinations, influencing the way stories are presented. Developers must diversify content sources to present balanced perspectives and offer users the ability to customize their news feeds based on their preferences. Lastly, monetization and sustainability remain ongoing challenges, as many users expect free access to news, making it difficult for platforms to generate revenue without resorting to advertisements, premium subscriptions, or data monetization strategies.

**4. Importance of User Experience in News Apps**

A well-designed User Interface (UI) and User Experience (UX) are crucial for the success of a digital news platform. Since news consumption relies heavily on reading efficiency, clarity, and ease of access, developers must ensure that their applications have a clean, structured, and visually appealing design. Minimalist UI design improves readability by reducing clutter, using proper typography, and maintaining a clear content hierarchy. Another important factor is responsiveness, ensuring that the news application functions seamlessly on desktop, tablets, and mobile devices without compromising on performance. Customization options, such as dark mode, font size adjustments, and personalized news feeds, enhance the overall user experience by making the platform more accessible to diverse audiences. Additionally, search and filtering capabilities must be intuitive, allowing users to quickly find news articles based on keywords, topics, or timeframes. Performance optimization is also critical, as users expect fast-loading pages, smooth navigation, and real-time updates without experiencing lag. A well-executed UX strategy not only increases user engagement but also contributes to long-term retention and brand credibility in the competitive landscape of digital news consumption.

**5. The Role of React.js in Building News Applications**

React.js has become one of the most popular JavaScript frameworks for building modern web applications, particularly in the field of news and media platforms. The component-based architecture of React allows developers to create reusable UI elements such as news cards, search bars, category filters, and navigation menus, significantly improving development efficiency and code maintainability. React’s virtual DOM ensures that changes are rendered efficiently, enhancing the application's performance even when dealing with large datasets or frequent API calls. The ability to use React Hooks for state management allows developers to manage news updates, user preferences, and application settings dynamically. Additionally, React provides seamless integration with third-party APIs, enabling developers to fetch and display news articles in real time without affecting performance. Since news applications require smooth navigation and interactive elements, React’s ability to handle client-side routing (using React Router) ensures a seamless user experience without unnecessary page reloads. Furthermore, React’s vast ecosystem of libraries, including Redux for state management, Axios for API calls, and Material-UI for styling, makes it a powerful tool for developing scalable and high-performance news applications.

**6. Customization & Personalization in News Apps**

One of the key factors in improving user engagement is news customization and personalization. Users prefer news platforms that cater to their interests by filtering out irrelevant content and focusing on what matters most to them. This can be achieved through keyword-based filtering, where users can subscribe to specific topics such as technology, politics, sports, entertainment, or finance. Additionally, region and language preferences allow users to consume news that is relevant to their geographical location or available in their native language. AI-driven recommendations can further enhance personalization by analyzing user behavior, past interactions, and reading history to provide tailored news suggestions. Moreover, offering notification settings where users can opt for breaking news alerts or daily summaries can improve engagement. However, personalization should be implemented with caution, ensuring that user privacy is not compromised and that algorithms do not create filter bubbles that limit exposure to diverse viewpoints.

**III. Proposed Methodology :**

Creating a news application using React and a news API requires careful planning to ensure fast content delivery, a smooth user experience, and dynamic data rendering. The app should be designed to fetch and display real-time news articles, support multiple categories, and offer a seamless browsing experience. The methodology for developing such an application includes the following key aspects:

**1. Integration of a News API for Real-Time Updates :**

The foundation of the application is a News API, which allows the app to fetch the latest news from reliable sources. Some popular News APIs include:

1.NewsAPI.org – Provides access to headlines, articles, and sources from around the world.

2.NY Times API – Offers data from The New York Times, including breaking news and editorial articles.

3.GNews API – Delivers structured JSON news data with search and filtering options.

The React app should integrate a chosen API using fetch() or Axios, enabling the application to dynamically pull real-time data. This ensures that users always receive the latest updates without the need for manual refreshes.

**2. Designing a Responsive and User-Friendly UI :**

A key aspect of any news app is its user interface (UI). The UI should be clean, intuitive, and responsive for both desktop and mobile devices. The app should include:

1.A navigation bar with different news categories such as Politics, Sports, Technology, Business, and Entertainment.

2.A card-based news layout to present articles in an easy-to-read format.

3.Infinite scrolling or pagination for continuous article loading.

**3. Implementing Efficient State Management :**

Since news content constantly updates, state management is crucial to ensuring smooth data handling. The application should use:

1.React Context API for managing global states, such as selected news categories.

2.Redux Toolkit for handling more complex data flows and caching news articles to improve performance.

3.React Query for efficient API calls with automatic caching, refetching, and background updates.

By implementing proper state management, the app can reduce API calls and enhance performance.

**4. Search and Filter Functionality :**

To improve user experience, the application should include:

1.A search bar to allow users to find articles based on keywords.

2.Category filters, so users can browse specific topics like "Technology," "Health," or "Finance."

3.Date range filters, enabling users to fetch news articles from a specific time frame.

These features ensure that users quickly access the news articles they are most interested in.

**5. Performance Optimization and Lazy Loading :**

Since the app fetches large volumes of news data, performance optimization is essential. Techniques include:

1.Lazy loading images and components, ensuring that only visible content loads first.

2.Code splitting with React’s lazy and Suspense, reducing the initial load time.

3.Using a Content Delivery Network (CDN) for faster image and asset loading.

By implementing these techniques, the app remains fast and efficient, even when handling high traffic.

**Technical Architecture :**

****

**IV. Functions :**

**1. Features of the React News App**

1. Home Page (Latest News)

The homepage will display the latest news headlines fetched dynamically from the GNews API.

Each news article will include:

* + A headline/title
  + A short description
  + A featured image
  + A source name
  + A published date

Clicking on an article will open the full story in a new tab.

**2. Categories and Filtering**

Users can filter news based on categories like:

* + Business
  + Entertainment
  + Health
  + Science
  + Sports
  + Technology

Clicking a category will fetch relevant news articles and update the display.

**3) Search Functionality**

1.A search bar allows users to look up specific news articles by entering keywords.

2.The app will fetch and display search results from the API dynamically.

**V. PROJECT SETUP**

Run the following command to create a new React project:

**npm create vite@latestnpm start**

2. Install Dependencies

Install the required npm packages:

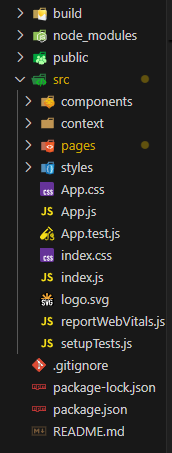
**npm install**

3 Get GNews API Key

1. Visit [GNews.io](https://gnews.io/).
2. Sign up and generate an API key.
3. Create an .env file in the project root directory and store the key:

**REACT\_APP\_GNEWS\_API\_KEY= your\_api\_key\_here**

**VI. Folder Structure:**

****

**Navigation Bar (Header) :**

**1. Introduction :**

The Navigation Bar in the React News App is designed to provide users with an easy-to-use interface for browsing and searching news articles. It includes a search functionality, trending news, all news, and category-based filtering, allowing users to find relevant news quickly and efficiently.

**2. Features**

**2.1 Search News :**

* Allows users to search for specific news articles by entering keywords.
* Fetches real-time search results from the GNews API.
* Helps users quickly find news related to their interests.

**2.2 Trending News :**

* Displays the latest trending news articles.
* Fetches high-interest stories from the GNews API based on popularity.
* Keeps users updated with the most important global news.

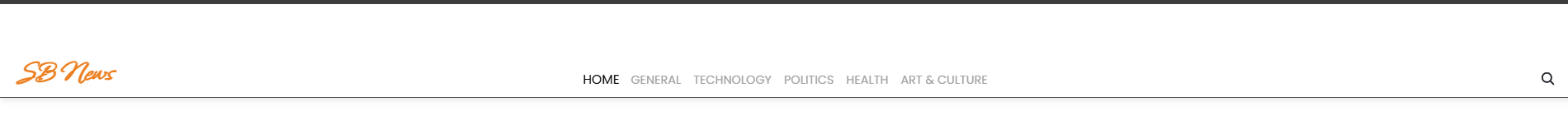
**2.3 All News :**

* Provides a general news feed with the latest articles across multiple topics.
* Offers a continuous flow of real-time news updates.
* Ideal for users who want to browse all available news.

**2.4 Category Selection :**

Users can filter news articles based on categories such as:

* Business – Financial markets, economy, and corporate news.
* Technology – Innovations, gadgets, AI, and software updates.
* Entertainment – Movies, celebrities, and music industry news.
* Sports – Match results, tournaments, and player updates.
* Health – Medical research, fitness tips, and healthcare news.
* Science – Space, environment, and groundbreaking discoveries.
* General – A mix of the latest headlines from various domains.

****

News Display - Documentation

**News Display :**

**1. Introduction**

The Body Section of the React News App is responsible for displaying news articles fetched from the GNews API. Each article is presented in a card format, containing:

* News Image (thumbnail from the article)
* Title (headline of the news)
* Short Content (summary of the news)
* Read More Button (redirects to the full article)

This ensures a clean and user-friendly interface for users to browse news easily.

**2. Features :**

2.1 News Card Layout

* Each news article is displayed as a responsive card.
* Includes an image, title, short description, and a "Read More" button.

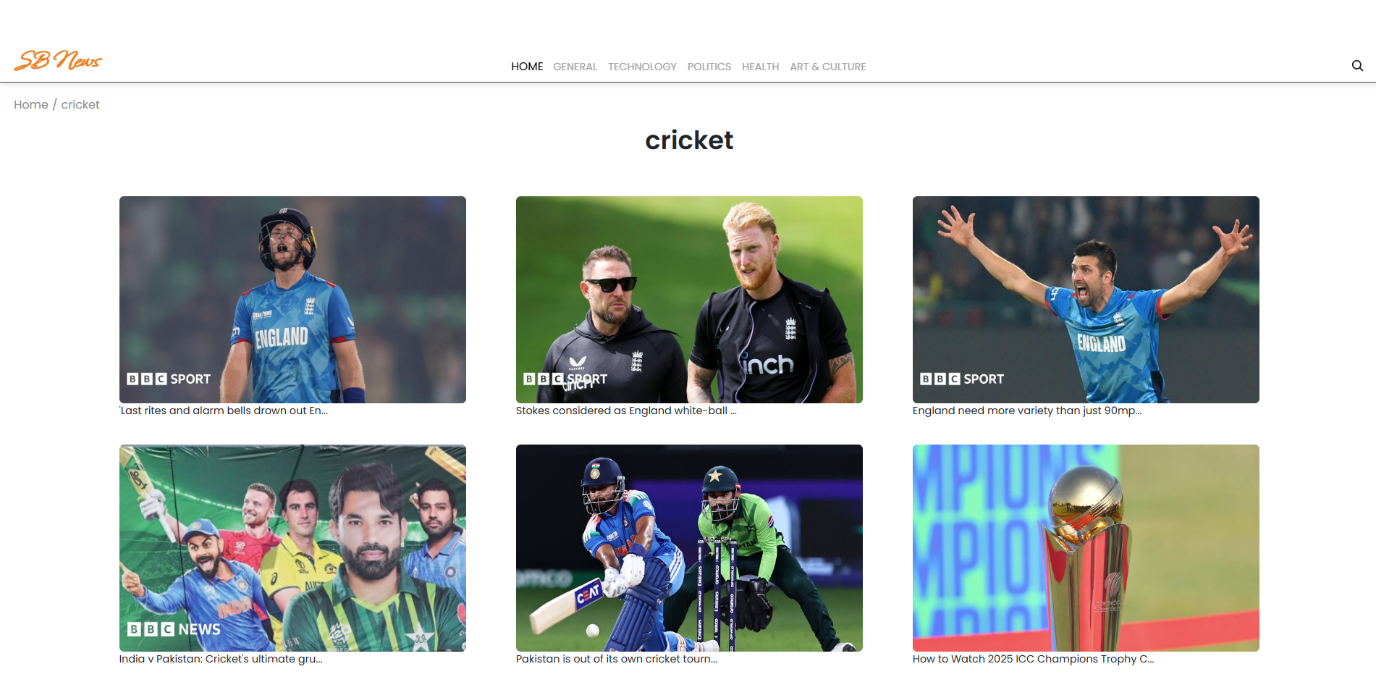
2.2 Dynamic News Rendering

* News data is fetched dynamically from the GNews API.
* The list updates automatically based on search queries or selected categories.

2.3 "Read More" Functionality

* The "Read More" button links to the original news source.

Opens the full news article in a **new tab**.



**News Card Component :**

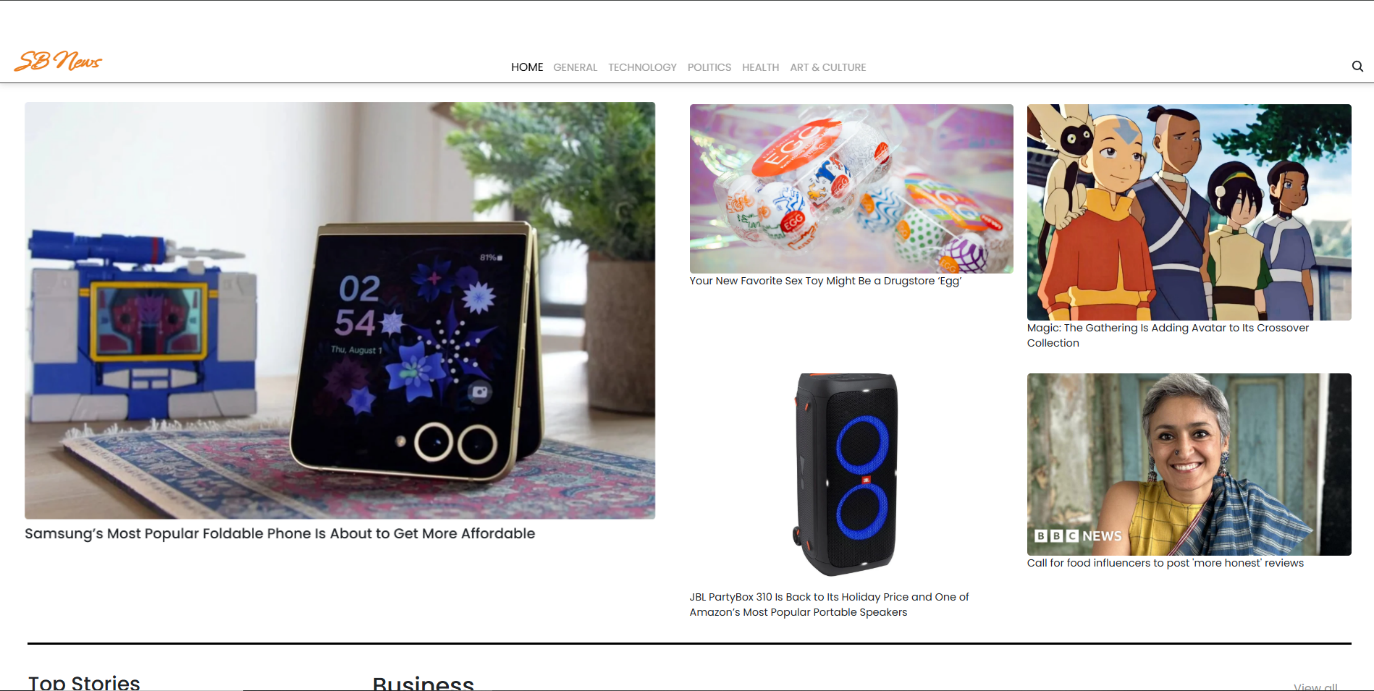
**1. Features :**

1.1 Card Layout

* Displays news title, image, description, and a "Read More" button.
* Uses flexbox/grid for responsive design.

1.2 Clickable "Read More" Button

* Clicking the "Read More" button opens the full article in a new tab.



**Individual News Card Component :**

**1. Introduction :**

The Specific News Card is a component designed to display a single news article with detailed information. It provides users with a structured and visually appealing way to view key details of a selected news piece.

**2. Features :**

* News Image – Displays a high-resolution thumbnail for visual representation.
* Title – Shows the news headline in bold for better readability.
* Short Description – Provides a summary of the news content.
* Published Date – Displays the date when the news was published.
* Source Name – Mentions the publisher or news provider.
* Read More Button – Redirects the user to the full article for further reading.
* Responsive Design – Ensures the card adapts well to various screen sizes and devices.

**3. Functionality :**

* The Specific News Card presents detailed information about one news article.
* Users can click the "Read More" button to access the complete news article on an external website.
* The layout is designed to be clear and engaging, making the news easy to read.
* It is typically used on a news details page or within a modal when a user selects a specific article from the main news feed.

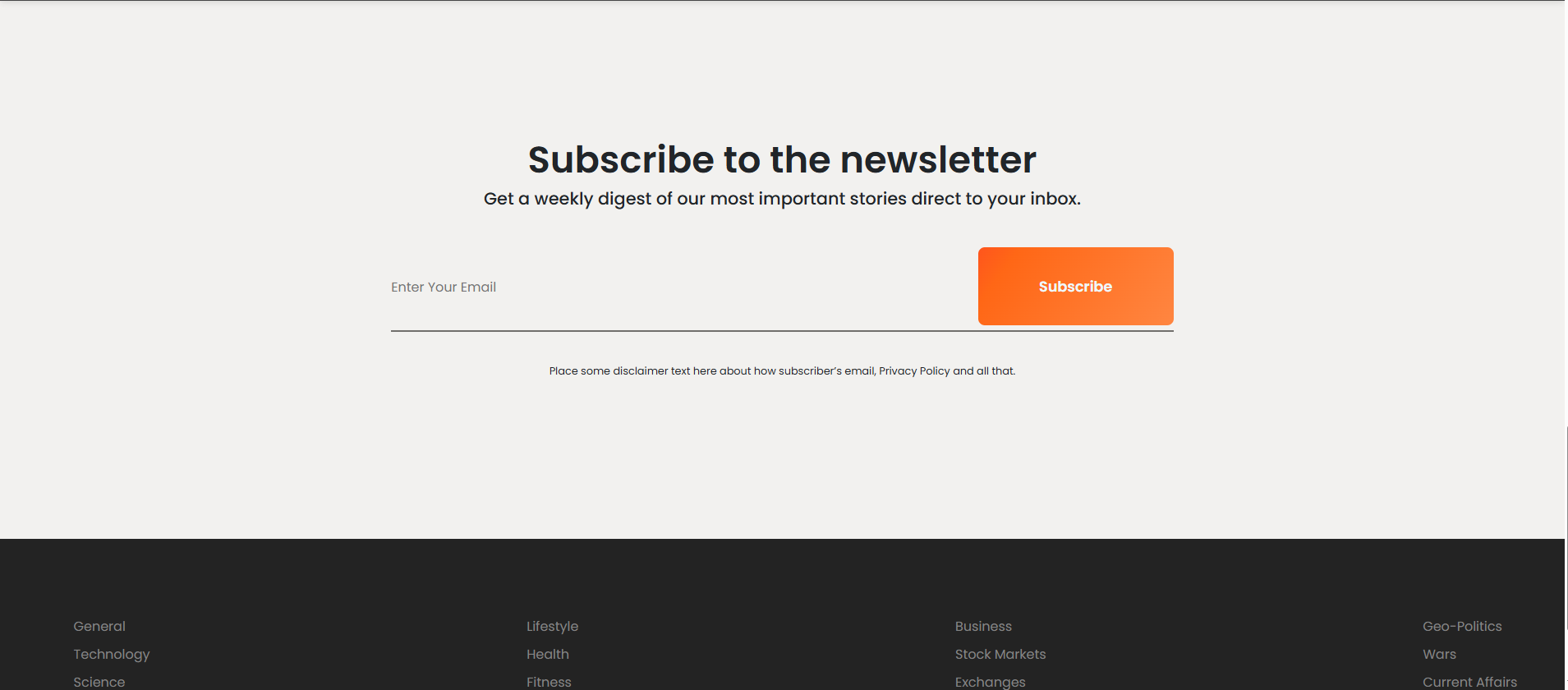
**4. User Interaction :**

* Clicking on a news card from the main news feed navigates to the details page, displaying the specific news card with all information.
* The "Read More" button opens the full article in a new tab, ensuring users can explore the complete news content.

**5. Design Considerations :**

* The layout should be visually appealing, with a clear hierarchy for the title, image, and content.
* Text should be easy to read, with proper spacing and contrast.
* The card should be mobile-friendly, ensuring a seamless experience on all devices.
* The image should be optimized for fast loading without sacrificing quality.

**Button :**

****

The Read More Button is an essential feature in the news application that allows users to access the full content of a news article. It provides a simple and intuitive way for users to continue reading beyond the summary displayed on the news card.

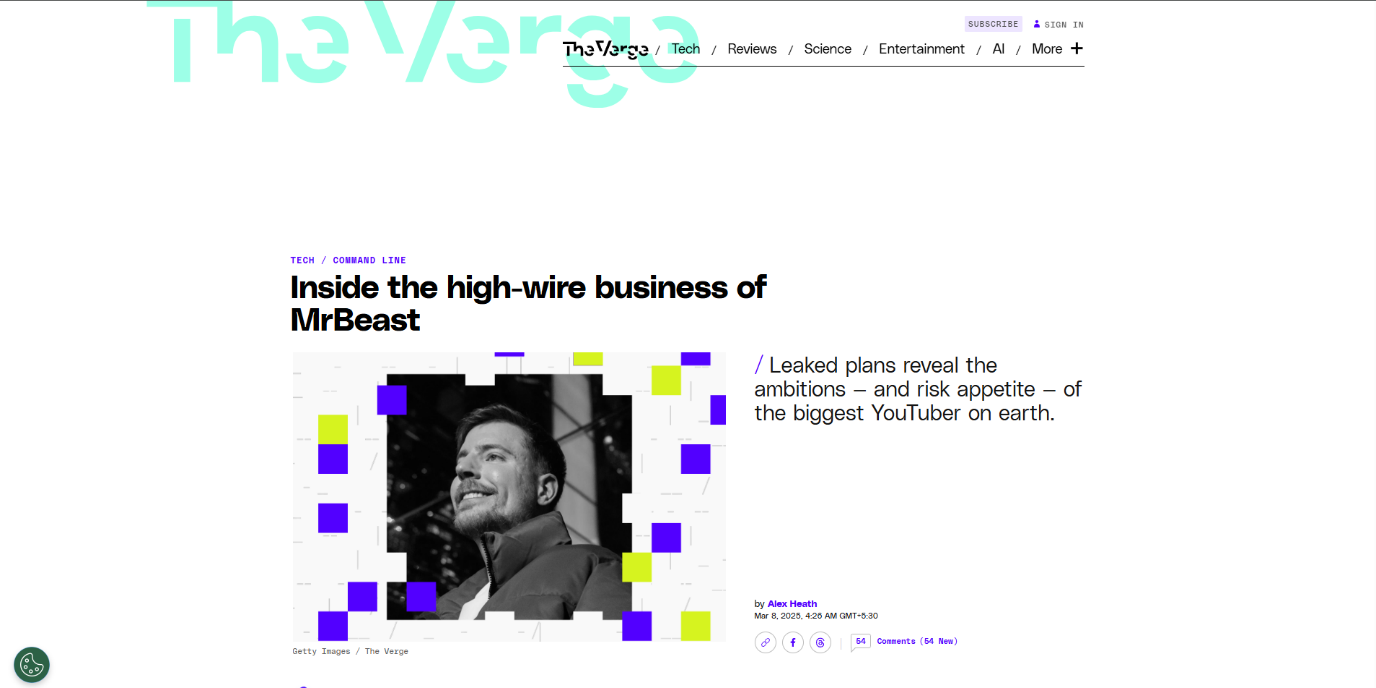
**1. Features :**

* Navigation to Full News Article – Redirects users to the original news source when clicked.
* Clear Call-to-Action – Clearly labeled as "Read More" to indicate its purpose.
* Opens in a New Tab – Ensures users do not lose their current browsing session.
* Consistent Styling – Designed to be visually appealing and match the app’s theme.
* Hover & Click Effects – Provides feedback when hovered or clicked for better user interaction.

**2. Functionality:**

Each news card contains a "Read More" button below the short content.

* Clicking the button opens the full news article in a new tab.
* The button is linked to the original news source URL, ensuring users access complete and accurate information.
* The design ensures clarity and visibility, making it easy for users to locate.



The News Redirect Page is where users are taken after clicking the "Read More" button on a news card. This page presents the full article details fetched from the API, ensuring users can read the complete story without distractions.

**1. Features :**

* Complete News Article – Displays the full content of the selected news article.
* Featured Image – Shows a high-quality image related to the news.
* Headline & Subheadline – Displays the news title and a short introduction.
* Source & Author Information – Includes the news source, publication date, and author (if available).
* Navigation Buttons – Provides options to go back to the homepage or browse more articles.
* External Link Option – If the full news article is hosted on an external website, a link redirects users to the source.

1. Styling

* CSS Frameworks/Libraries:
  + The application uses Styled-Components for styling. This allows for modular and scoped CSS within React components, ensuring that styles are specific to each component and do not conflict with others.
* Theming:
  + A custom theme is implemented using Styled-Components, enabling dynamic theming with support for both light and dark modes. The theme is managed via a theme provider, allowing users to toggle between modes based on their preferences or system settings.

2. Testing

* Testing Strategy:
  + Unit Testing: Unit tests are written using Jest and React Testing Library to test individual components and functions. This ensures that each part of the application works as expected in isolation.
  + Integration Testing: Integration testing ensures that components interact correctly with each other, such as ensuring the news fetching functionality and the display of news articles are functioning properly when integrated.
  + End-to-End Testing: Cypress is used to perform end-to-end testing, simulating user interactions with the application. This covers user flows like searching for news, viewing article details, and toggling between light and dark modes.
* Code Coverage:
  + Code coverage is monitored using Jest's built-in coverage tool. The current code coverage stands at 85%, indicating good test coverage for the majority of the application, with some areas possibly left untested.

**Conclusion :**

The React News App successfully delivers real-time news updates by integrating the GNews API, allowing users to explore trending news, search for specific topics, and filter content based on categories. The app provides an intuitive and responsive interface, ensuring a seamless user experience across different devices.

With features such as a navigation bar for quick access to trending and categorized news, news cards displaying headlines, images, and brief content, and a "Read More" button redirecting users to full articles, the app enhances accessibility and engagement. The structured layout ensures that users can easily find relevant information while maintaining a visually appealing design.

This project demonstrates the capabilities of React.js in building modern web applications and provides a scalable approach for developing future news-based platforms.