**INSIGHTSTREAM:NAVIGATE THE NEWS LANDSCAPE**

**TEAM MEMBERS:**

ARAVINDAN.S - PROJECT MANAGER

VITHYASURIYA.P -DEVELOPER

SUGANTHAN.M -DESIGNER

ARJUN.R -TESTER

**PROJECT OVERVIEW:**

Insight Stream is a dynamic news web application that allows users to fetch and display the latest news from multiple sources. The app features an intuitive interface with a responsive navbar, news category selection, and a real-time search function.

**KEY FEATURES:**

✅ **Fetch News from Various Sources** – Users can select news providers like BBC, CNN, TechCrunch, etc.  
✅ **Category-Based Filtering** – News can be filtered by categories such as Sports, Politics, and Entertainment.  
✅ **Live Search Functionality** – A search bar allows users to find relevant news articles instantly.  
✅ **Bootstrap-Powered UI** – A modern, responsive design for a seamless user experience.  
✅ **Expandable News Articles** – News items are displayed in an accordion format for easy reading.

**ARCHITECTURE:**

**🔹 1. Frontend (UI Layer) - HTML, CSS, Bootstrap**

* **Navbar: News source selection, search bar.**
* **News Display: Bootstrap Accordion for dynamic updates.**
* **Search & Filters: Category-based filtering.**
* **Footer: Basic branding info.**

**🔹 2. Logic Layer (JavaScript - index.js)**

* **Fetch News API: Retrieves JSON data from NewsAPI.**
* **Render News: Updates UI dynamically.**
* **User Interaction: Handles dropdown selection & search filtering.**

**🔹 3. Data Layer (External API - NewsAPI.org)**

* **Fetches real-time news based on user selection.**
* **API Example:URL**
* **https://newsapi.org/v2/top-headlines?sources=bbc-news&apiKey=YOUR\_API\_KEY**

**🔹 4. Component Flow**

**User → UI (Navbar & Search) → JavaScript (index.js) → API Fetch ↓ ↓ News Display (Accordion) ← API Response (JSON) ← NewsAPI.org**

**STATE MANAGEMENT:**

**The state management in *Insight Stream* revolves around three key variables: selectedSource for tracking the chosen news source, newsData for storing fetched articles, and searchQuery for filtering news. When a user selects a source, selectedSource updates and triggers an API call, storing the response in newsData, which is then rendered in the UI. The search functionality updates searchQuery, dynamically filtering displayed articles. State persistence is managed using localStorage to retain the selected source and search input. Future improvements could include Redux for scalable state management and IndexedDB for offline storage.**

**SETUP INSTRUCTIONS:**

**1. Prerequisites**

**Before installing, make sure you have the following installed on your system:**

* **Node.js (for package management)**
* **Python (if using a simple local server)**
* **Git (to clone the repository)**
* **A web browser (Chrome, Firefox, Edge, etc.)**

**2. Clone the Repository**

**Open a terminal and run:**

*git clone https://github.com/your-username/insight-stream.git*

*cd insight-stream*

**3. Install Dependencies (If Using Node.js)**

**If there is a package.json file, install required dependencies:**

*npm install*

**4. Start the Local Server**

**Option 1: Using Python (for a simple server)**

*python -m http.server 8000*

**Then, open** *http://localhost:8000/* **in your browser.**

**Option 2: Using Live Server in Node.js  
If you have live-server installed globally:**

*live-server*

**Or run directly using:**

*npx live-server*

**5. Configure API Key**

* **Go to** [**NewsAPI.org**](https://newsapi.org/) **and get an API key.**
* **Open index.js and replace API\_KEY with your own.**

**6. Run the Application**

* **Open** *http://localhost:8000/* **in your browser.**
* **Browse news, search for articles, and filter by source.**

**FOLDER STRUCTURE:**

**1. Client (Frontend/UI):**

* **index.html → The main HTML file that structures the UI.**
* **styles.css → Contains the styling for the website.**
* **index.js → Handles user interactions, fetching, and displaying news.**
* **assets/ → Stores images, icons, and other UI assets.**

**2. Utilities (Logic & API Handling):**

* **scripts/ → Contains additional JavaScript files for modular functionality.**
* **api/ → Manages API requests and responses, keeping API logic separate.**
* **config.js → (If needed) Stores API keys and configurations securely.**

**RUNNING THE APPLICATION:**

**To start the frontend server locally, follow these steps:**

1. **Navigate to the project directory:**

*cd client*

1. **Install dependencies:**

*npm install*

1. **Start the frontend server:**

*npm start*

**COMPONENT DOCUMENTATION:**

**1. Key Components**

**Below are the major components in the project, along with their purpose and the props they receive:**

* **NewsList**
  + **Purpose: Fetches and displays a list of news articles.**
  + **Props:** 
    - **articles (Array): The list of news articles to display.**
* **NewsItem**
  + **Purpose: Represents a single news article with a title, description, and a "Read More" link.**
  + **Props:** 
    - **title (String): The headline of the news article.**
    - **content (String): The short description of the news article.**
    - **url (String): The URL for the full article.**
* **SearchBar**
  + **Purpose: Allows users to search for specific news articles.**
  + **Props:** 
    - **onSearch (Function): Handles the search input and updates the displayed news articles.**
* **CategoryFilter**
  + **Purpose: Dropdown to filter news by category (e.g., Sports, Politics, Entertainment).**
  + **Props:** 
    - **onSelectCategory (Function): Handles category selection and updates the news feed.**
* **SourceFilter**
  + **Purpose: Dropdown to filter news by source (e.g., BBC, CNN, TechCrunch).**
  + **Props:** 
    - **onSelectSource (Function): Handles source selection and fetches relevant news.**

**2. Reusable Components**

* **Button**
  + **Purpose: A styled button used throughout the application.**
  + **Props:** 
    - **text (String): The label on the button.**
    - **onClick (Function): Handles button click events.**
* **Dropdown**
  + **Purpose: A reusable dropdown component for category/source selection.**
  + **Props:** 
    - **options (Array): List of options to display.**
    - **onSelect (Function): Handles selection changes.**

**STATE MANAGEMENT:**

**1. Global State**

**The application uses global state management to handle data that needs to be accessed across multiple components. This includes:**

* **News Articles: Stored at the top level and shared with NewsList, SearchBar, and CategoryFilter components.**
* **Selected Category & Source: Managed globally to update the news feed dynamically when a user selects a category or source.**
* **Search Query: Stored globally to ensure all components receive the filtered news results.**

**State Flow:**

1. **The App component fetches news data and maintains it in a global state.**
2. **SearchBar, CategoryFilter, and SourceFilter update the global state when user input changes.**
3. **NewsList listens to these state changes and re-renders accordingly.**

**2. Local State**

* **Search Input (SearchBar): Stores the current search text before updating the global state.**
* **Dropdown Selection (CategoryFilter and SourceFilter): Handles temporary selection before updating the global state.**
* **Accordion Toggle (NewsItem): Stores the state of whether a news item is expanded or collapsed.**

**STYLING:**

**CSS Frameworks/Libraries**

**This project primarily uses Bootstrap 5 for styling, providing a responsive and modern design with minimal custom CSS. The Bootstrap CDN is included in the project for easy integration.**

* **Bootstrap Version: 5.1.1**
* **Usage: Used for grid layout, buttons, accordions, navbar, and forms.**

**Custom Styling**

* **Additional custom styles are applied using internal CSS within the <style> tag in index.html.**
* **The background, footer, and accordion components have custom colors and designs.**

**Theming & Custom Design**

* **The project does not use a full-fledged theming system but applies consistent colors and styles manually.**
* **If needed, Bootstrap's built-in dark mode and utility classes can be leveraged for additional styling.**

**TESTING:**

**Testing Strategy**

**The testing approach for this project ensures that key components and functionalities work as expected. The testing includes:**

* **Unit Testing:**
  + **Tests individual functions, utility modules, and components.**
  + **Framework: Jest (for JavaScript testing).**
  + **Example: Testing the getdata() function to ensure it correctly fetches news articles.**
* **Integration Testing:**
  + **Tests interactions between multiple components (e.g., how the search bar filters news).**
  + **Framework: React Testing Library (if using React) or Jasmine/Karma (for vanilla JavaScript).**
* **End-to-End (E2E) Testing:**
  + **Simulates user interactions (e.g., clicking the "Get News" button, searching for articles).**
  + **Tool: Cypress (for browser-based testing).**

**SCREENSHOTS LINK:**

[***https://drive.google.com/drive/folders/18788tAwor3Wy4KcU2bEsPaLiSGuT0QMF***](https://drive.google.com/drive/folders/18788tAwor3Wy4KcU2bEsPaLiSGuT0QMF)

**DEMO VIDEO LINK:**

[***https://drive.google.com/file/d/1M\_-wloyyr7J6j7Ft29v6iGXHnaxxF3A4/view?usp=drivesdk***](https://drive.google.com/file/d/1M_-wloyyr7J6j7Ft29v6iGXHnaxxF3A4/view?usp=drivesdk)

**KNOWN ISSUES:**

**1. News API Limitations: The free tier of the News API may restrict the number of requests per day, which can cause the news feed to stop updating.**

**2. UI Responsiveness: Some UI elements may not display correctly on smaller screens due to CSS inconsistencies.**

**3. Search Functionality Delay: Searching for news articles may take longer depending on the number of articles loaded.**

**4. Pagination Missing: Currently, the application fetches and displays all news at once without pagination, which may affect performance.**

**5. CORS Issues: Some API requests may fail due to cross-origin restrictions, requiring server-side handling.**

**FUTURE ENHANCEMENTS:**

**Future enhancements include pagination support for better navigation, category-based filtering to refine news selection, and a dark mode toggle for improved UX. Adding user authentication will enable personalized feeds, while infinite scrolling will enhance browsing efficiency. Improved search functionality, smooth UI animations, and offline mode for cached news access are also planned.**