1. INTRODUCTION

- **PROJECT TITLE:**

InsightStream: Navigate the News Landscape

(React Application)

- **TEAM MEMBERS:**

TEAM MEMBERS	E-MAIL ID	ROLE
GOPIKA P	apsparthasarathy79@@gmail.com	Team Leader
HARIPRIYA M	haripriyamanikumar2005@gmail.com	Team Member
DEEPIKA T	deepikadeepika2641@gmail.com	Team Member
RETHIKASRI S	rethikasenthilkumar76@gmail.com	Team Member

2. PROJECT OVERVIEW

○ PURPOSE:

The primary purpose of the news app is to provide users with quick and convenient access to current news stories and information from various sources, allowing them to stay updated on events happening around the world through their mobile devices, while its key goals including delivering personalized news feeds, fostering user engagement and maintaining a reliable sources of credible information.

key goals include:

- ✓ User-Friendly Experience: Develop an interface that is intuitive and easy to navigate, ensuring users can effortlessly access, save, and share their preferred news articles.
- ✓ Comprehensive News Management: Provide robust features for organizing and managing news content, incorporating advanced search options for a personalized news experience.
- ✓ Technology Stack: Employ cutting-edge web development technologies, such as React.js, to ensure an efficient and enjoyable user interface.

○ FEATURES:

Core Features

1. News Feed: A scrollable list of news articles, with images, headlines, and summaries.
2. Real-time Updates: News articles are updated in real-time, with new articles added to the feed as they become available.
3. Search Bar: A search bar that allows users to search for specific news articles or topics.
4. Categories: News articles are categorized by topic (e.g. politics, sports, entertainment), with users able to select specific categories to view.

React-Specific Features

1. Components: The app is built using React components, which are reusable and modular.
2. JSX: The app uses JSX to render components and manage state.
3. Hooks: The app uses React Hooks (e.g. useState, useEffect) to manage state and side effects.

3. ARCHITECTURE

- **COMPONENT STRUCTURE:**

I. Components

1. App.js (Top-Level Component)

- Responsible for rendering the entire app
- Contains the main navigation and routing

2. NewsFeed.js (Main Component)

- Displays a list of news articles
- Fetches data from the API
- Uses infinite scrolling to load more articles

3. Article.js (Component)

- Displays a single news article
- Contains the article's title, image, and text
- Allows users to save and share articles

4. SearchBar.js (Component)

- Allows users to search for news articles
- Sends search queries to the API
- Displays search results

5. Categories.js (Component)

- Displays a list of news categories
- Allows users to select categories to view

6. UserProfile.js (Component)

- Displays a user's profile information
- Allows users to save and view saved articles

II. Component Interactions

1. App.js → NewsFeed.js

- App.js renders NewsFeed.js as its main component
- NewsFeed.js fetches data from the API and displays the news feed

2. NewsFeed.js → Article.js

- NewsFeed.js renders Article.js components for each news article
- Article.js displays the article's details and allows users to interact with it

3. SearchBar.js → NewsFeed.js

- SearchBar.js sends search queries to the API
- NewsFeed.js receives the search results and displays them

4. Categories.js → NewsFeed.js

- Categories.js allows users to select categories to view
- NewsFeed.js receives the selected category and displays the corresponding news articles

5. UserProfile.js → NewsFeed.js

- UserProfile.js allows users to save and view saved articles
- NewsFeed.js receives the saved articles and displays them in the user's profile

III. API Interactions

1. NewsFeed.js → API

- NewsFeed.js sends requests to the API to fetch news articles
- API returns the news articles, which are then displayed in the news feed

2. SearchBar.js → API

- SearchBar.js sends search queries to the API
- API returns the search results, which are then displayed in the news feed

3. Categories.js → API

- Categories.js sends requests to the API to fetch news articles for a specific category
- API returns the news articles, which are then displayed in the news feed

○ STATE MANAGEMENT:

The state management approach used in this React application is a combination of React Context API and Redux.

React Context API:

The React Context API is used to manage state that needs to be shared between components, but doesn't require a full-fledged state management solution like Redux.

- Contexts: Multiple contexts are created to manage different types of state, such as user authentication, news articles, and categories.
- Providers: Context providers are used to wrap the entire application, providing the state to all components that need it.
- Consumers: Context consumers are used to access the state in individual components.

Redux

Redux is used to manage global state that requires more complex management, such as news article fetching and caching.

- Store: A single Redux store is created to manage the global state.
- Actions: Actions are dispatched to trigger state changes, such as fetching news articles or saving a user's preferences.
- Reducers: Reducers are used to handle actions and update the state accordingly.
- Selectors: Selectors are used to access the state in individual components.

- **ROUTING:**

Routing Structure

The routing structure is designed to provide a logical and intuitive navigation experience for users.

Route Configuration

The route configuration is defined in a separate file (routes.js) and is imported into the main app component (App.js).

4. SETUP INSTRUCTIONS

- **PRE-REQUISITES**:

Here are the key prerequisites for developing a frontend application:

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

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.

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.

Development Environment:
Choose a code editor or Integrated Development Environment (IDE) that suits your preferences, such as Visual Studio Code, Sublime Text, or WebStorm.

○ INSTALLATION:

Clone the repository:

Step 1: Create a GitHub Account
Step 2: Create a New Repository
Step 3: Initialize the Repository
Step 4: Add Files to the Repository
Step 5: Push Changes to GitHub
Step 6: Configure Repository Settings
Step 7: Create a README File
Step 8: Create a License File
Step 9: Create a .gitignore File

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.

Configure Environment Variables:

Environment variables are values that are set outside of a program and are used to configure the program's behavior. In the context of a News App, environment variables can be used to store sensitive information such as API keys, database credentials, and other configuration settings.

Configuring Environment Variables:

1. Create a .env File
2. Add Environment Variables

5. FOLDER STRUCTURE

- **CLIENT:**

The React application is organized into several folders, each with its own specific purpose. The main folders are:

- components/: Reusable UI components

- pages/: Top-level routes for the application

- styles/: Global CSS styles

-context/: General Context

- **UTILITIES:**

Helper Functions

- api.js: A file containing helper functions for making API requests, such as fetchData, postData, and putData.
- storage.js: A file containing helper functions for storing and retrieving data in local storage, such as storeData and retrieveData.
- string.js: A file containing helper functions for string manipulation, such as capitalize and truncate.

Utilities

- constants.js: A file containing constants used throughout the application, such as API endpoints and color schemes.
- enums.js: A file containing enumerations used throughout the application, such as status codes and error messages.

Classes

- NewsArticle.js: A class representing a news article, with properties such as title, author, and content.
- User.js: A class representing a user, with properties such as name, email, and password.

Custom Hooks

- useFetchData.js: A custom hook for fetching data from an API, with options for caching and error handling.
- useStorage.js: A custom hook for storing and retrieving data in local storage, with options for encryption and expiration.
- useForm.js: A custom hook for managing form state and validation, with options for custom validation rules and error messages.

6. RUNNING THE APPLICATION

The commands to start the frontend using npm start in the client directory:

Navigate to the Client Directory

1. Open a terminal or command prompt.
2. Navigate to the client directory using the cd command:

```
cd client
```

Install Dependencies (Optional)

1. If you haven't installed dependencies before, run the following command:

```
npm install
```

Start the Frontend

1. Start the frontend using the following command:

```
npm start
```

This will start the development server, and you can access your application at http://localhost:3000 (or the port number specified in your package.json file).

Verify the Application

1. Open a web browser and navigate to http://localhost:3000.
2. Verify that your application is running correctly.

7. COMPONENT DOCUMENTATION

- **KEY COMPONENTS:**

1. App Component

- Purpose: The top-level component that renders the entire application.
- Props:
  -children: The child components to render.
- Description: The App component is the main entry point of the application. It renders the Header, Main, and Footer components.

2. Header Component

- Purpose: Renders the application header.
- Props:
- title: The title of the application.

- logo: The logo of the application. - Description: The Header component renders the application title and logo.

3. Main Component

- Purpose: Renders the main content of the application.
- Props:
- children: The child components to render.
- Description: The Main component renders the main content of the application, which includes the NewsList component.

4. NewsList Component

- Purpose: Renders a list of news articles.
- Props:
- newsArticles: An array of news article objects.
- onArticleClick: A callback function to handle article clicks.
- Description: The NewsList component renders a list of news articles and handles article clicks.

5. NewsArticle Component

- Purpose: Renders a single news article.
- Props:
- article: A news article object.
- onArticleClick: A callback function to handle article clicks.
- Description: The NewsArticle component renders a single news article and handles article clicks.

6. Footer Component

- Purpose: Renders the application footer.
- Props:
- copyrightText: The copyright text to display. - Description: The Footer component renders the application footer with copyright text.

7. SearchBar Component

- Purpose: Renders a search bar to search news articles.
- Props:
- onSearch: A callback function to handle search queries. - Description: The SearchBar component renders a search bar and handles search queries.

8. LoadingIndicator Component

- Purpose: Renders a loading indicator to display while data is being fetched.

- Props:
- loading: A boolean indicating whether data is being fetched.
- Description: The LoadingIndicator component renders a loading indicator while data is being fetched.

○ REUSABLE COMPONENTS:

some reusable components and their configurations:

1. Button Component

- Purpose: A reusable button component.
- Configurations:
- variant: Can be one of primary, secondary, success, danger, or warning.
- size: Can be one of small, medium, or large.
- onClick: A callback function to handle button clicks.   - disabled: A boolean indicating whether the button is disabled.

2. Input Component

- Purpose: A reusable input component.
- Configurations:
- type: Can be one of text, email, password, or number.
- placeholder: A string to display as a placeholder.
- onChange: A callback function to handle input changes.   - value: The initial value of the input.

3. Select Component

- Purpose: A reusable select component.
- Configurations:
- options: An array of options to display.
- onChange: A callback function to handle select changes.   - value: The initial value of the select.

4. Table Component

- Purpose: A reusable table component.
- Configurations:
- columns: An array of column definitions.
- data: An array of data to display.   - onRowClick: A callback function to handle row clicks.

5. Modal Component

- Purpose: A reusable modal component.

- Configurations:
- isOpen: A boolean indicating whether the modal is open.
- onClose: A callback function to handle modal closure.
- title: The title of the modal.   - children: The content of the modal.

8. STATE MANAGEMENT

GLOBAL STATE:
Global state management refers to the process of managing state that is shared across multiple components and features of an application. In the context of the NewsApp, global state management involves managing the following:

- News articles: A list of news articles that are fetched from an API.
- Selected article: The currently selected news article.
- Search query: The current search query.
- Loading state: A boolean indicating whether the application is currently loading data. - Error state: An error message or object indicating whether an error has occurred.

State Flow

The state flows across the application as follows:

1. Action Dispatch
2. Reducer
3. Store
4. Component Subscription
5. Component Rendering

- **LOCAL STATE:**
Local state refers to the state that is specific to a single component and is not shared with other components. Local state is used to store data that is only relevant to the component itself.

Handling Local State
Local state is handled using the useState hook in React**.**

Updating Local State Local state can be updated using the setState function returned by useState.

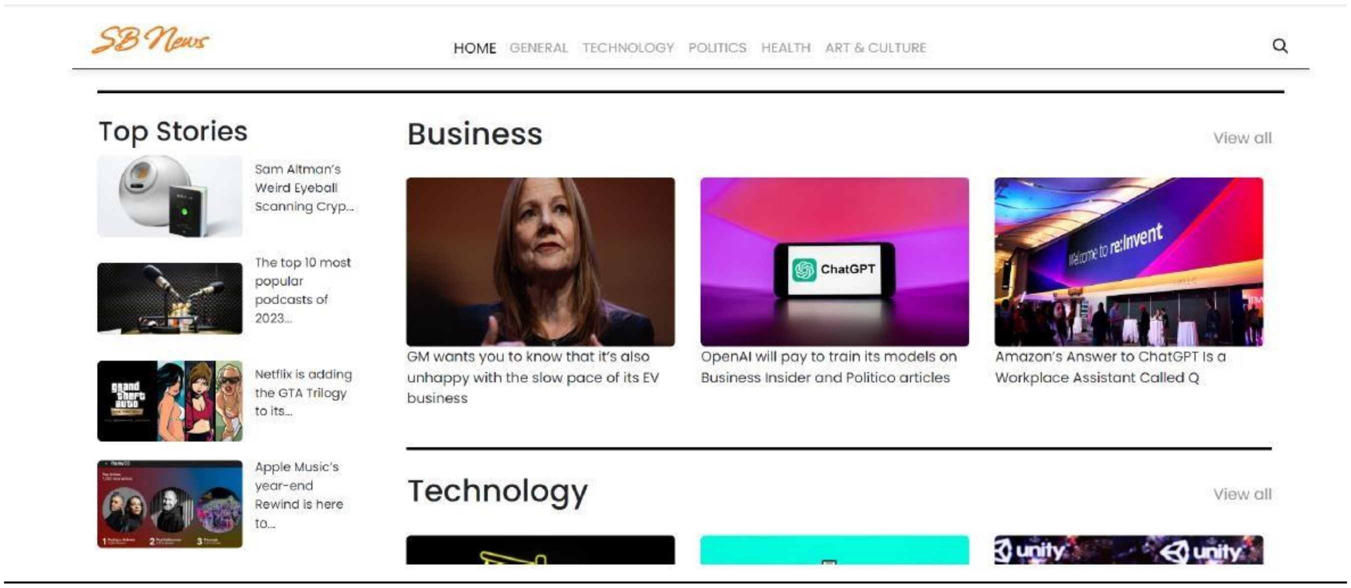
Using Local State in Components Local state can be used in components to store data that is specific to the component.

Benefits of Local State Local state provides several benefits, including:

- Encapsulation
- Reusability
- Simplified Code

9. USER INTERFACE

A user interface (UI) is the visual representation of a software application, website, or device that allows users to interact with it. The UI provides a way for users to input data, navigate through the application, and view the output.



Sam Altman's Weird Eyeball Scanning Cryp...

The top 10 most popular podcasts of 2023...

Netflix is adding the GTA Trilogy to its...

Apple Music's year-end Rewind is here to...

GM wants you to know that it's also unhappy with the slow pace of its EV business

OpenAI will pay to train its models on Business Insider and Politico articles

Amazon's Answer to ChatGPT Is a Workplace Assistant Called Q

Screens

1. **Home Screen**: Displays a list of top news stories, with images, headlines, and summaries.
2. **General Screen**: Displays a general news, with image, headline, and content.
3. **Teachnology Screen**: Displays a technology news, with image, headline, and content.
4. **Politics Screen:** Displays a political news, with image, headline, and content.
5. **Health Screen:** Displays a health news, with image, headline, and content. 6.**Art and Culture screen:** Displays a art and culture news, with image, headline, and content.

Buttons and Icons

1. **Home Button**: Returns the user to the Home Screen.
2. **Search Button**: Activates the search function.

3. **Settings Button**: Opens the Settings Screen.
4. **Subscribe Button**: To subscribe the news channel. 5. **Share Button**: Allows users to share a news article on social media.

Animations and Transitions

1. **Page Transitions**: Smooth transitions between pages.
2. **Button Animations**: Buttons animate on hover and click.
3. **Loading Animations**: Loading animations display while content is loading.

10. STYLING

CSS FRAMEWORKS/LIBRARIES:

CSS Frameworks

1. Bootstrap: A popular front-end framework that provides a comprehensive set of CSS classes and components for building responsive and mobile-first UI components.
2. Tailwind CSS: A utility-first CSS framework that provides a set of pre-defined classes for styling HTML elements.
3. Material-UI: A popular CSS framework developed by Google that provides a set of pre-built UI components based on the Material Design specification.
4. Bulma: A modern CSS framework that provides a set of pre-built UI components and a simple grid system.
5. Foundation: A popular front-end framework that provides a comprehensive set of CSS classes and components for building responsive and mobile-first UI components.

CSS Libraries

1. Normalize.css: A small CSS library that provides a set of CSS rules to normalize the styling of HTML elements across different browsers.
2. Reset CSS: A small CSS library that provides a set of CSS rules to reset the styling of HTML elements to a consistent baseline.
3. Animate.css: A CSS library that provides a set of pre-defined animation classes for adding animations to HTML elements.

CSS Preprocessors

1. Sass: A popular CSS preprocessor that provides a set of features such as variables, nesting, and mixing for writing more efficient and modular CSS code.

○

2. Less: A CSS preprocessor that provides a set of features such as variables, nesting, and mixing for writing more efficient and modular CSS code.
3. Stylus: A CSS preprocessor that provides a set of features such as variables, nesting, and mixing for writing more efficient and modular CSS code.

CSS-in-JS Libraries

1. Styled Components: A popular CSS-in-JS library that provides a set of features such as dynamic styling and theming for writing more efficient and modular CSS code.
2. Emotion: A CSS-in-JS library that provides a set of features such as dynamic styling and theming for writing more efficient and modular CSS code.

○ THEMING:

Theming allows you to change the visual appearance of your application, such as colors, typography, and layout, without modifying the underlying code. In NewsApp, theming can be implemented using various techniques:

1. **CSS Variables**: Define CSS variables for colors, typography, and other visual elements. Then, use these variables in your CSS code. This way, you can easily switch between different themes by changing the values of these variables.
2. **Theme Provider:** Create a theme provider component that wraps your entire application. This component can provide a theme object that contains the visual styles for your application. Then, use this theme object in your components to apply the styles.
3. **Styled Components:** Use a library like Styled Components, which allows you to define styles as JavaScript functions. This way, you can easily switch between different themes by changing the styles.

Custom Design System

A custom design system is a collection of reusable UI components, guidelines, and assets that define the visual design of your application. In NewsApp, a custom design system can be implemented using various techniques:

1. **Create a Design Language System (DLS)**: Define a set of principles, guidelines, and assets that describe the visual design of your application. This includes typography, colors, spacing, and other visual elements.
2. **Build a UI Component Library:** Create a library of reusable UI components that adhere to the design language system. This includes components such as buttons, inputs, cards, and other UI elements.

3. **Use a Design System Tool:** Use a tool like Storybook, Bit, or Lona to create, manage, and document your design system.

11.TESTING

TESTING STRATEGY:

1. Unit Testing
Unit testing involves testing individual components or functions in isolation. For React components, unit testing typically involves rendering the component and verifying that it renders correctly.

Tools
- Jest: A popular testing framework for React applications. - React Testing Library: A testing library that provides a set of APIs for testing React components.

2. Integration Testing
Integration testing involves testing how multiple components interact with each other. For React applications, integration testing typically involves rendering a component tree and verifying that the components interact correctly.

Tools
- Jest: A popular testing framework for React applications. - React Testing Library: A testing library that provides a set of APIs for testing React components.

○

3. End-to-End Testing
End-to-end testing involves testing the entire application from start to finish, simulating user interactions and verifying that the application behaves correctly.

Tools
- Cypress: A popular end-to-end testing framework for web applications. - Playwright: A browser automation framework that can be used for end-to-end testing**.**

○ **CODE COVERAGE:**

Testing Tools

1. **Jest:** A popular testing framework for React applications.
2. **React Testing Library:** A testing library that provides a set of APIs for testing React components.
3. **Cypress:** A popular end-to-end testing framework for web applications. **4. Enzyme:** A testing library that provides a set of APIs for testing React components.

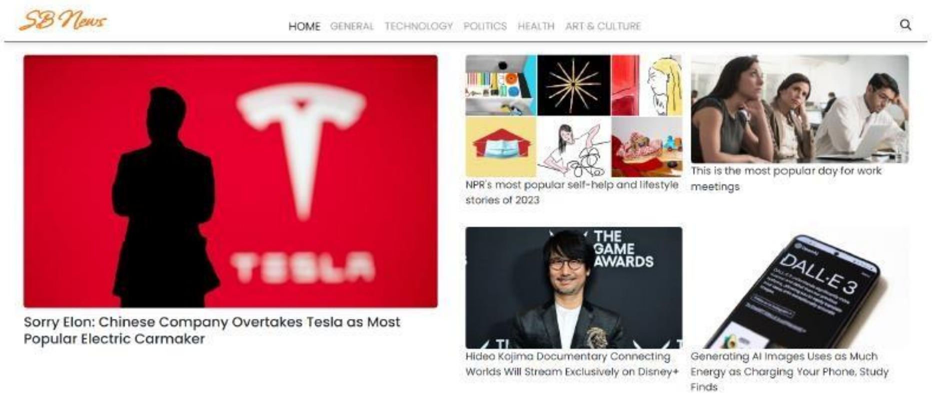
Testing Techniques

1. **Unit Testing:** Test individual React components in isolation.
2. **Integration Testing:** Test how multiple React components interact with each other.
3. **End-to-End Testing**: Test the entire application from start to finish.
4. **Snapshot Testing:** Test that the component's UI matches a previously saved snapshot.

12. SCREENSHOTS OR DEMO

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.



Sorry Elon: Chinese Company Overtakes Tesla as Most Popular Electric Carmaker

NPR's most popular self-help and lifestyle stories of 2023

This is the most popular day for work meetings

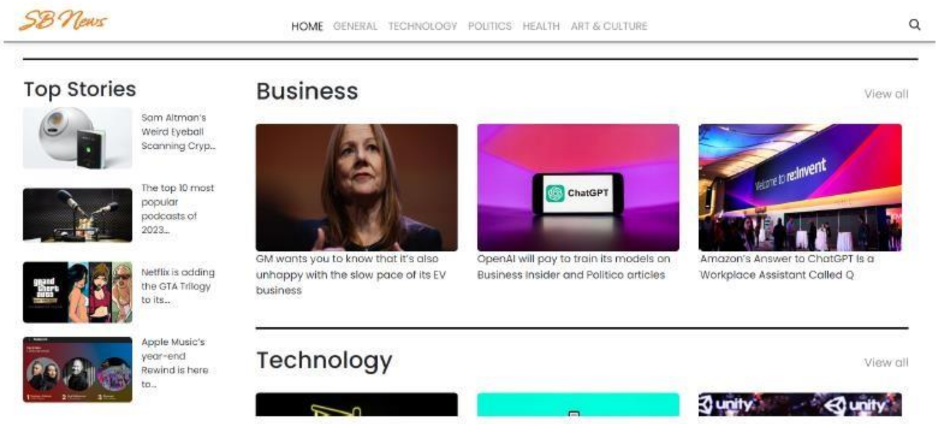
Hideo Kojima Documentary Connecting Worlds Will Stream Exclusively on Disney+

Generating AI Images Uses as Much Energy as Charging Your Phone, Study Finds

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.



Sam Altman's Weird Eyeball Scanning Cryp...

The top 10 most popular podcasts of 2023...

Netflix is adding the GTA Trilogy to its...

Apple Music's year-end Rewind is here to...

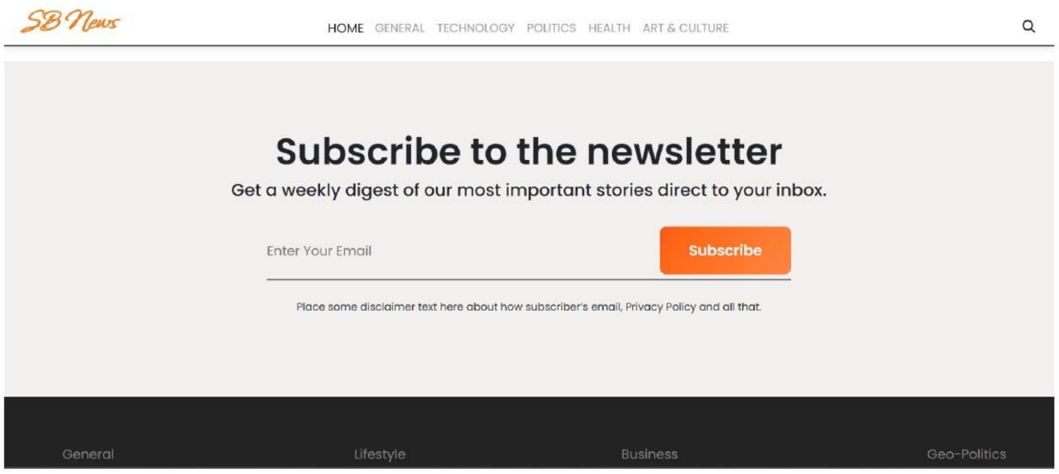
GM wants you to know that it's also unhappy with the slow pace of its EV business

OpenAI will pay to train its models on Business Insider and Politico articles

Amazon's Answer to ChatGPT Is a Workplace Assistant Called Q

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.



Subscribe to the newsletter

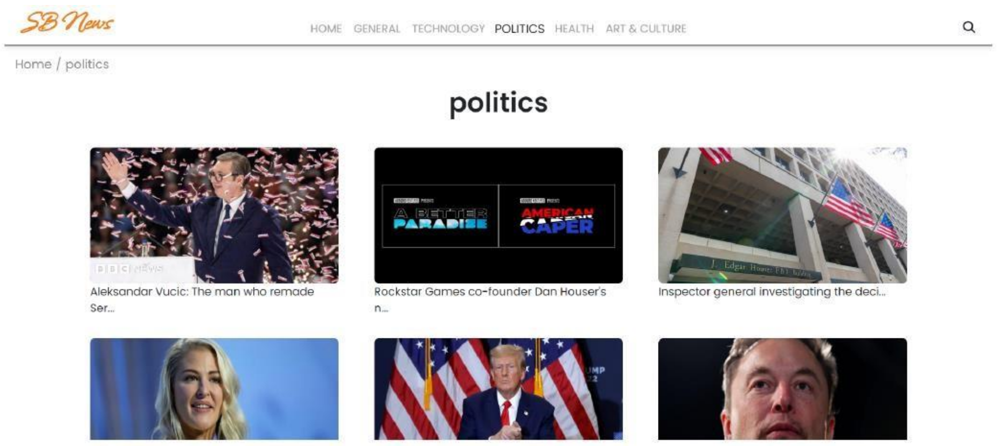
Get a weekly digest of our most important stories direct to your inbox.

Enter Your Email

Subscribe

Place some disclaimer text here about how subscriber's email, Privacy Policy and all that.

Category/Search result page
Finding the news you crave is effortless with InsightStream. This page displays a neatly organized ssclear headline, a concise summary, and if available, an image to give you a quick glimpse into the story.



Aleksandar Vucic: The man who remade Ser...

Rockstar Games co-founder Dan Houser's n...

Inspector general investigating the deci...

PROJECT DEMO LINK:

https://drive.google.com/file/d/16pfCD9zSm79tfRI2S7aidLzsfxkRu3ka/view?usp=sharing

13. KNOWN ISSUES

Some known bugs and issues are:

- Article image loading issue
- Performance issue Accessibility issue

14.FUTURE ENHANCEMENT

some potential future features or improvements for News App:

New Components

1. Video Player Component
2. Podcast Player Component
3. Quiz Component
4. Poll Component

Animations

1. Article Loading Animation
2. Component Transition Animations
3. Scrolling Animations

Enhanced Styling

1. Customizable Themes
2. Improved Typography
3. Consistent Spacing
4. Enhanced Image Display