INSIGHT STREAM NAVIGATE THE NEWS LANDSCAPE

INTRODUCTION:

Project Title: Insight Stream Navigate News Landscape

Team ID: NM2025TMID46900

Team Leader: Kavi Presilla . A & presilladoss1127@gmail.com

S.No	Name	Mail ID
1	Priyadharshini .S	priyadharshini.110512@gmail.com
2	Punitha .V	punithavelu2007@gmail.com
3	Ramya.M	ramyamoorthi8555@gmail.com
4	Selvamathi.V	vselvamathi007@gmail.com

PROJECT OVERVIEW :

PURPOSE:

The purpose of such a project could be to:

- Help users stay informed on current events
- Curate relevant news based on user interests
- Provide a streamlined way to navigate through vast news content
- Possibly offer insights or analysis on news topics

FEAUTERS:

- Personalized news feed: Tailoring news content based on user interests or preferences
- News aggregation: Collecting news from multiple sources for a comprehensive view
- Topic tracking: Allowing users to follow specific topics or keywords
- Insights or analysis: Providing context or expert opinions on news stories

- ❖ Customizable filters: Enabling users to filter news by category, source, or other criteria
- ❖ Breaking news alerts: Sending notifications for major news events

ARCHITECTURE:

COMPONENT STATEMENT :

1.Problem Statement :

- ➢ Issues in the current news ecosystem
- ➢ Misinformation & fake news.
- ➢ Biased reporting & echo chambers.
- ➢ Overload of information.
- ➢ Lack of critical evaluation by readers.

2. Objectives

- ➢ To study the digital news ecosystem.
- ➢ To identify factors influencing credibility.
- ➢ To develop methods/tools for evaluating news.
- ➢ To promote critical thinking and responsible news consumption.

3. Literature Review / Background Study

- ➢ Overview of how algorithms, AI, and social media impact news.
- ➢ Case studies of misinformation (e.g., elections, health crises).
- ➢ Existing tools or strategies (fact-checking websites, awareness campaigns).

4. Methodology

- ➢ Data Collection: Analyze digital platforms, news apps, and user behavior.
- ➢ Analysis: Identify patterns of misinformation, bias, or manipulation.
- ➢ Solution Design:News credibility checklist.
- ➢ Prototype (dashboard, app, or awareness module).
- ➢ Educational strategies for media literacy.
- ➢ Testing & Feedback: Pilot run with users, gather feedback.

5. Proposed Solution / System Design

- System architecture or workflow (if app/dashboard).
- Features (e.g., source verification, bias detection, fact-check integration).
- Flowchart or diagram showing how users interact with the solution.

6. Implementation

- Tools/technologies used (if any app/software is developed).
- Steps for creating awareness (if campaign-based solution).
- Deployment strategy (small scale → large scale).

7. Results / Findings

- Key insights from research & testing.
- Examples of how the solution/tool improves news understanding.
- Comparison of user behavior before & after intervention.

8. Expected Outcomes

- Increased media literacy.
- Better ability to identify credible vs fake news.
- Awareness of digital news manipulation.
- A working solution (tool, guide, or campaign).

9. Challenges & Limitations:Difficulties in changing user behavior.

- Limitations in detecting all forms of misinformation.
- Technical/ethical challenges in designing tools

10.News Aggregator: Collects news from various sources (APIs, websites, etc.)

11.Content Processor: Analyzes and categorizes news content

12.User Interface: Displays news feed, allows filtering, and provides insights

13.Personalization Engine: Tailors news feed based on user preferences or behavior

14.Alert System: Sends notifications for breaking news or specific topics

15.Data Storage: Manages storage of news data, user preferences, and interaction data

STATE MANAGEMENT:

- ➔ User preferences: Managing what topics or sources users are interested in
- ➔ News feed state: Keeping track of what's been shown to the user, what's new, etc.
- ➔ Filter settings: Storing user's filter choices (e.g., by category, source)
- ➔ Alert status: Managing whether alerts are enabled for certain topics or breaking news
- ➔ Interaction history: Tracking user interactions (e.g., clicks, dismissals) for personalization
- ➔ Client-side state: Using React's state management (e.g., useState, Context API) or other frameworks' equivalents
- ➔ Server-side state: Managing state in a backend database for persistence across sessions

ROUTING:

- ➔ Home/Feed route: Displaying the main news feed based on user preferences
- ➔ Topic/category routes: Showing news filtered by specific topics or categories
- ➔ Source routes: Displaying news from specific sources
- ➔ Search route: Handling user searches for news topics or keywords
- ➔ Settings route: Managing user preferences, alert settings, etc.
- ➔ Client-side routing: Using libraries like React Router for single-page app (SPA) navigation
- ➔ Server-side routing: Handling routes on the server for multi-page apps

SETUP INSTRUCTION :

PREREQUISITES:

- ➔ Node.js (for JavaScript-based projects)
- ➔ Package manager like npm or yarn
- ➔ News API key from a provider like NewsAPI or GNews
- ➔ Code editor/IDE of choice

INSTALLATION:

Create project: Run npm create vite@latest (for Vite/React) or use your preferred method.

- ➔ INSTALL DEPENDENCIES:Run npm install or yarn install for necessary package like react,react router.
- ➔ SETUP NEWS API : Get an API key from a news provider and configure it in your project.
- ➔ RUN PROJECT:Use npm run dev or npm start to start the development server

FOLDER STRUCTURE :

CLIENT :

- /src:
- /components: Reusable UI components (NewsCard, Header)
- /pages: Components for different routes (Home, TopicPage)
- /services: API calls or utility functions for news fetching
- /utils: Helper functions
- App.js: Main app component
- index.js: Entry point

UTILITIES :

- CredibilityCheck.js: Functions for checking news credibility (e.g., source verification, fact-check integration)
- DateFormatter.js: Functions for formatting dates or timestamps in news items
- StringHelpers.js: Functions for manipulating text (e.g., truncating headlines)

RUNNING THE APPLICATON :

FRONTEND: npm start in the client directory

- o Install dependencies: Run npm install or yarn install in the project root.

- Start development server: Run npm run dev or npm start to launch the app.
- View in browser: Open http://localhost:3000 (or another specified port) in a browser.
- Build the app: Run npm run build to create a production-ready build.
- Deploy: Deploy the build to a hosting platform (Vercel, Netlify, etc.).

COMPONENT DOCUMENTATION:

KEY COMPONENT:

- NewsCard: Displays individual news items with details like headline, source, date.
- NewsFeed: Renders a list of news items using NewsCard.
- SearchBar: Allows users to search news by keywords or topics.
- FilterOptions: Enables filtering news by categories, sources, etc.

REUSABLE COMPONENT:

- NewsCard: A component for displaying a single news item, reusable across different views.
- Button: A customizable button component for actions like "Load More" or "Filter".
- Loader: A component for showing loading states while fetching news.

STATE MANAGEMENT :

GLOBAL STATE:

- ☆ React Context API: For managing state like user preferences, news filters across components.
- ☆ Redux: For more complex state management needs.

Global state management helps with:

- ☆ Sharing state between components without prop drilling
- ☆ Updating state from anywhere in the app

LOCAL STATE:

- ✰ UseState hook: For simple state like input values, toggle states.
- ✰ UseReducer hook: For more complex state logic within a component.

Local state is useful for:

- ✰ Managing component-specific state without affecting global state.
- ✰ Keeping logic self-contained within a component.

USER INTERFACE:

- ✦ News feed: A list or grid of news items with headlines, sources, dates.
- ✦ Search and filter options: For users to find news by keywords, categories, or sources.
- ✦ News detail view: Showing full news content or summary with credibility indicators.

UI considerations:

- ✦ Responsiveness: UI should work well on desktop and mobile.
- ✦ -Accessibility: UI should be usable for people with different abilities.
- ✦ Credibility indicators: Visual cues for news credibility or source reliability.

STYLING :

CSS FRAMEWORKS/LIBRARIES:

- Tailwind CSS: Utility-first framework for custom designs.
- Bootstrap: Component-based framework for rapid UI development.

These help with:

- Consistent styling: Across components and pages.
- Responsive design: Adapting to different screen sizes.
- Customizability: Tailoring look and feel.

THEMING :

- ❖ Theme variables: Define in CSS (e.g., :root in CSS or theme object in JS for CSS-in-JS setups).
- ❖ Consistent application: Use theme variables across components for unified styling.

Example theming in "Insight stream":

- ❖ Define primary color for headlines, buttons.
- ❖ Use typography settings for consistent text styling across news items.

TESTING :

TESTING STRATEGY :

- ➢ Unit tests: Testing individual functions or components (e.g., NewsCard rendering).
- ➢ Integration tests: Testing how components work together (e.g., news fetching and display)
- ➢ End-to-end tests: Testing user flows like searching for news or filtering.

Tools for testing:

- ➢ Jest: For unit tests.
- ➢ React Testing Library: For component tests.
- ➢ Cypress: For end-to-end tests.

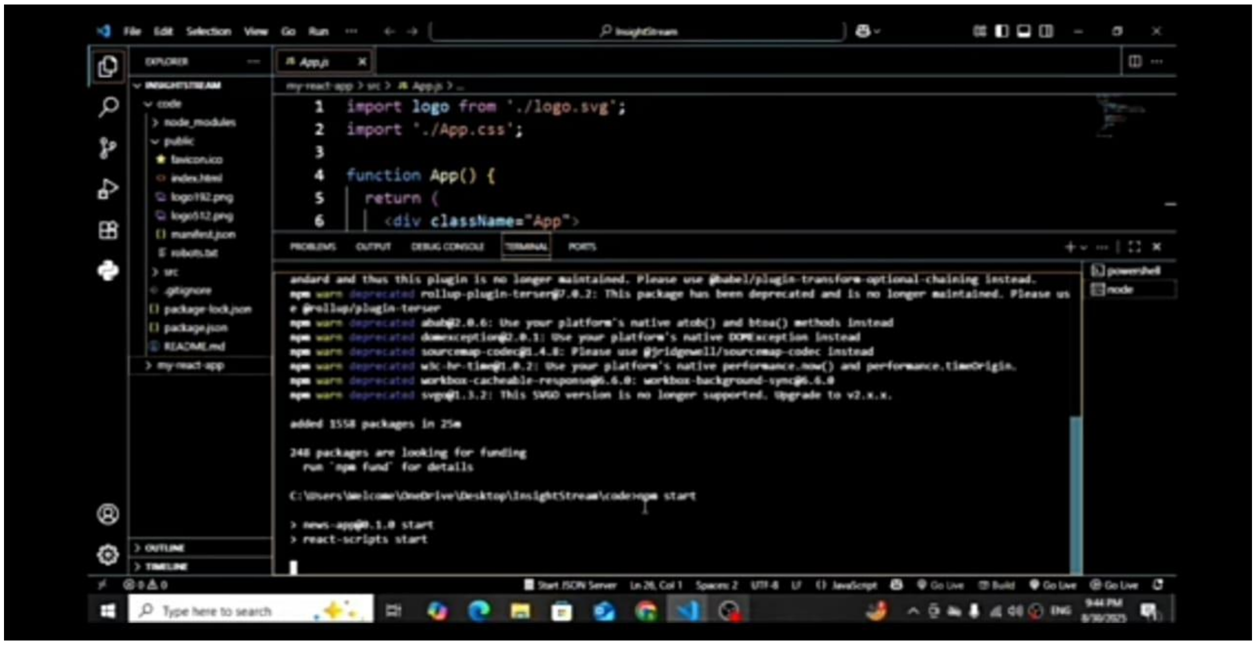
CODE COVERAGE :

- ➢ Coverage metrics: Tools like Jest report coverage of statements, branches, functions, lines.
- ➢ Goal: Aim for high coverage in critical parts like news fetching, filtering logic.

Checking coverage:

- ➢ Run tests with coverage flags (e.g., jest --coverage).
- ➢ Review reports to identify untested code paths.

SCREENSHOTS OR DEMO:



```
import logo from './logo.svg';
import './App.css';

function App() {
  return (
    <div className="App">
```

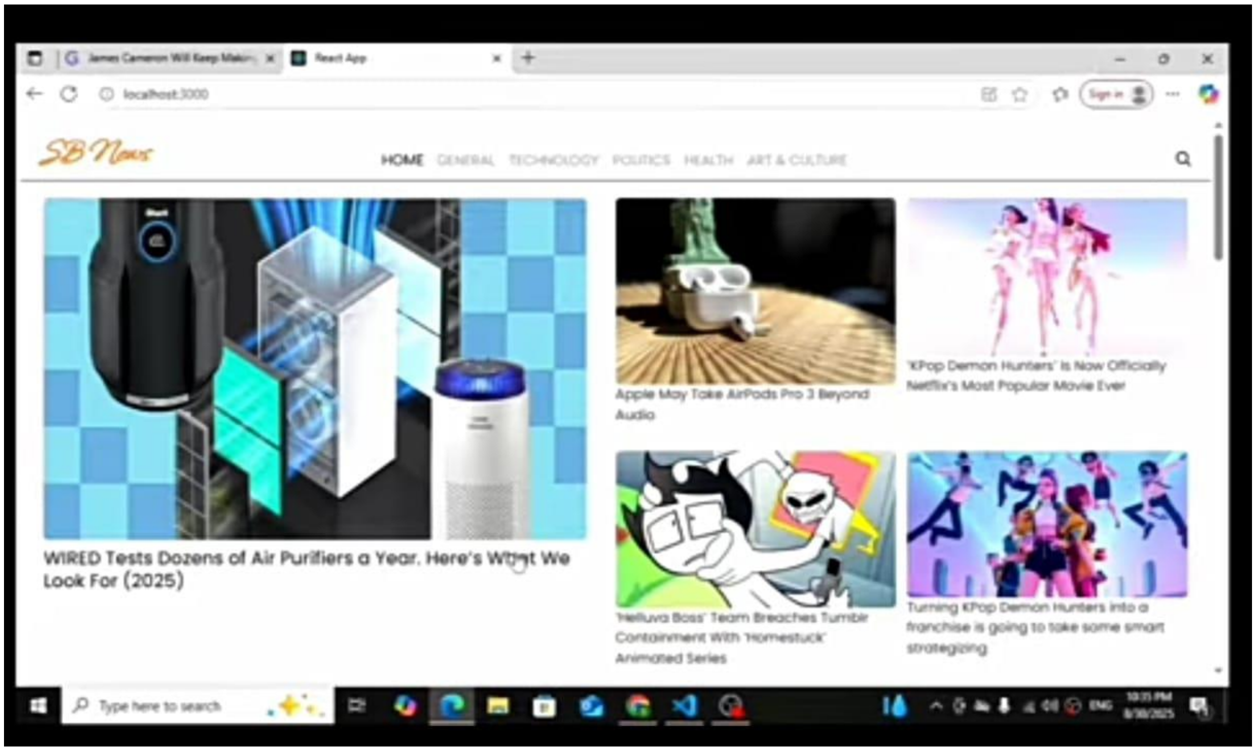
```
andard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-optional-chaining instead.
npm warn deprecated rollup-plugin-terser@7.0.2: This package has been deprecated and is no longer maintained. Please us
e @rollup/plugin-terser
npm warn deprecated abab@2.0.6: Use your platform's native atob() and btoa() methods instead
npm warn deprecated domexception@2.0.1: Use your platform's native DOMException instead
npm warn deprecated sourcemap-codec@1.4.8: Please use @jridgewell/sourcemap-codec instead
npm warn deprecated w3c-hr-time@1.0.2: Use your platform's native performance.now() and performance.timeOrigin.
npm warn deprecated workbox-cacheable-response@6.6.0: workbox-background-sync@6.6.0
npm warn deprecated svgo@1.3.2: This SVGO version is no longer supported. Upgrade to v2.x.x.

added 1558 packages in 25m

248 packages are looking for funding
  run `npm fund` for details

C:\Users\Welcome\OneDrive\Desktop\InsightStream\code>npm start

> news-app@0.1.0 start
> react-scripts start
```



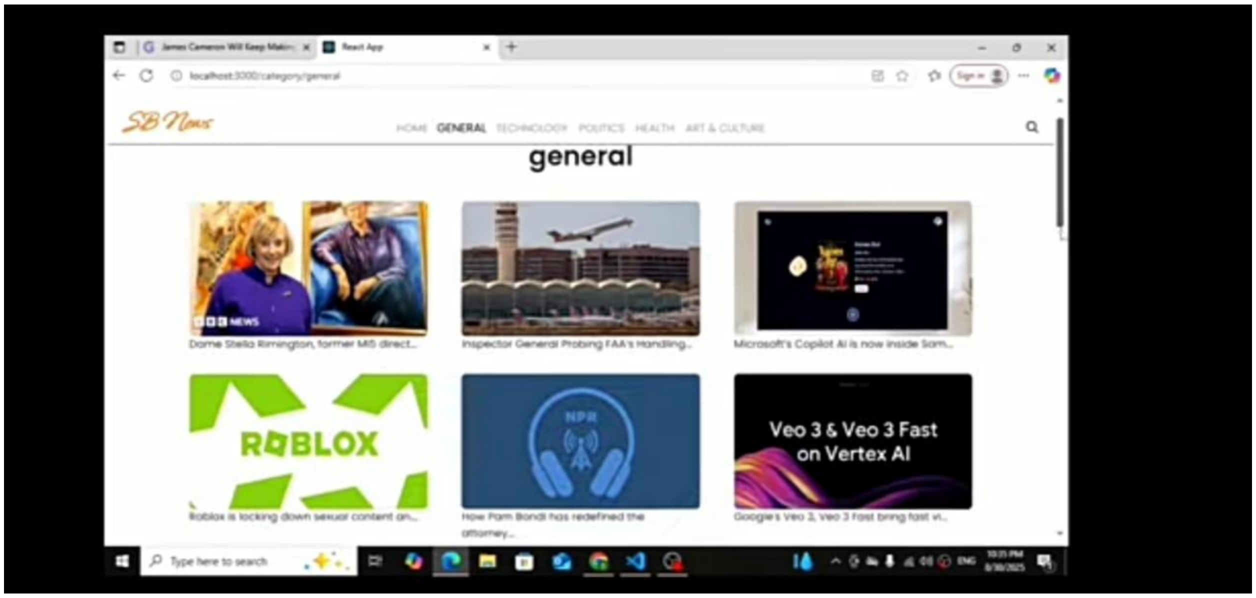
WIRED Tests Dozens of Air Purifiers a Year. Here's What We Look For (2025)

Apple May Take AirPods Pro 3 Beyond Audio

'KPop Demon Hunters' Is Now Officially Netflix's Most Popular Movie Ever

'Helluva Boss' Team Breaches Tumblr Containment With 'Homestuck' Animated Series

Turning KPop Demon Hunters into a franchise is going to take some smart strategizing



general

Dame Stella Rimington, former MI5 direct...

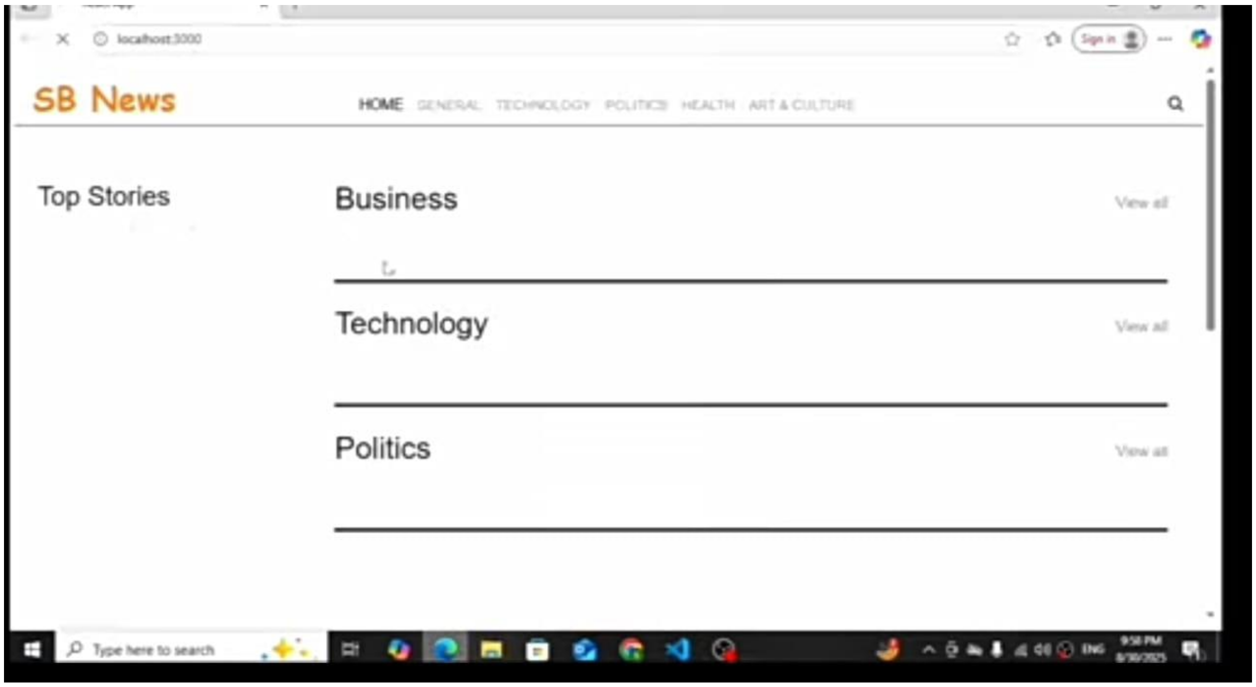
Inspector General Probing FAA's Handling...

Microsoft's Copilot AI is now inside Sam...

Roblox is locking down sexual content an...

How Pam Bondi has redefined the attorney...

Google's Veo 3, Veo 3 Fast bring fast vi...

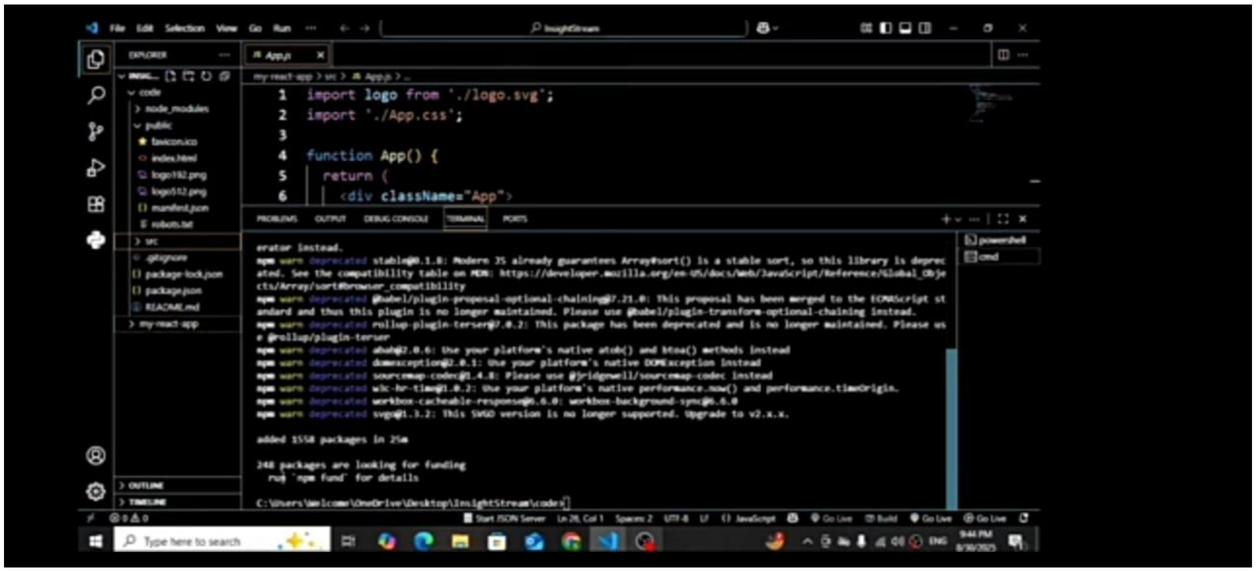


Top Stories

Business
View all

Technology
View all

Politics
View all



```
import logo from './logo.svg';
import './App.css';

function App() {
  return (
    <div className="App">
```

```
erator instead.
npm warn deprecated stable@0.1.8: Modern JS already guarantees Array#sort() is a stable sort, so this library is deprec
ated. See the compatibility table on MDN: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Obje
cts/Array/sort#browser_compatibility
npm warn deprecated @babel/plugin-proposal-optional-chaining@7.21.0: This proposal has been merged to the ECMAScript st
andard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-optional-chaining instead.
npm warn deprecated rollup-plugin-terser@7.0.2: This package has been deprecated and is no longer maintained. Please us
e @rollup/plugin-terser
npm warn deprecated abab@2.0.6: Use your platform's native atob() and btoa() methods instead
npm warn deprecated domexception@2.0.1: Use your platform's native DOMException instead
npm warn deprecated sourcemap-codec@1.4.8: Please use @jridgewell/sourcemap-codec instead
npm warn deprecated w3c-hr-time@1.0.2: Use your platform's native performance.now() and performance.timeOrigin.
npm warn deprecated workbox-cacheable-response@6.6.0: workbox-background-sync@6.6.0
npm warn deprecated svgo@1.3.2: This SVGO version is no longer supported. Upgrade to v2.x.x.

added 1558 packages in 25m

248 packages are looking for funding
  run `npm fund` for details

C:\Users\Welcome\OneDrive\Desktop\InsightStream\code>
```

KNOWN ISSUES:

- News fetching errors: Handling API errors or slow responses.
- Performance issues: Optimizing for large news datasets or slow networks.
- UI glitches: Ensuring consistent layout on different devices.

Addressing issues:

- Logging and monitoring: Track errors in production.
- User feedback: Gather feedback to identify issues users encounter.

FUTURE ENHANCEMENTS :

- Personalized news feeds: Let users customize their news feed based on interests.
- News credibility scoring: Show credibility scores for news sources or articles.
- Push notifications: Notify users of breaking news in their areas of interest.

Benefits of enhancements:

- Better user experience: Tailoring news to user preferences.
- Increased transparency: Showing news credibility.