InsightStream — Project Documentation

Project Title: InsightStream —- Navigate The News Landscape.

Team ID: NM2025TMID39083

Team Leader:

- KeerthanaCK— 24bsccs15@cttewc.edu.in

Team Members:

- Reshma J — 24bsccs44@cttewc.edu.in
- Dharsini S — 24bsccs54@cttewc.edu.in
- NithyaD — 24bsccs06@cttewc.edu.in

---

1. Project Overview

Purpose

InsightStream is a React-based news web application designed to help users navigate the news landscape. It organizes articles into categories, provides a clean reading interface, and demonstrates how to integrate APIs into a frontend app.

Goals

- Build a structured, responsive news application.
- Use modern React features (components, hooks, context).
- Provide a maintainable project for learning and extending.

Key Features

- Home page with top stories and featured sections.
- Category-based browsing (Technology, Business, Health, etc.).

- Individual news page for detailed reading.
- Newsletter subscription section.
- Responsive layout with reusable UI components.

---

2. Architecture

Component Structure

- App.js — main entry with routing.
- Components/
  - NavbarComponent.jsx — navigation bar with links.
  - Hero.jsx — hero banner section.
  - HomeArticles.jsx — featured articles grid.
  - TopStories.jsx — top news section.
  - NewsLetter.jsx — subscription form.
  - Footer.jsx — footer section.
- Pages/
  - Home.jsx — home page combining Hero, Articles, Top Stories, etc.
  - CategoryPage.jsx — category-specific news.
  - NewsPage.jsx — detailed article view.
- Context/
  - GeneralContext.jsx — provides shared state.

Routing

- Implemented via React Router.
- Paths:
    - `/` → Home
    - `/category/:name` → Category page
    - `/news/:id` → News detail page

Styling

- Centralized CSS files in `src/styles/` for each component and page.
- Custom CSS, no Tailwind.

---

3. Setup Instructions

Prerequisites

- Node.js and npm installed.

Installation

Clone the repository:

```
 git clone <repo-url>
cd code
```

1.

Install dependencies:

```
 npm install
```

2.

Start the development server:

```
npm start
```

3.
4. The app will run at http://localhost:3000.

---

4. Folder Structure

```
code/
├── public/              # static files
│   ├── index.html
│   ├── favicon.ico
│   ├── logo192.png
│   ├── logo512.png
│   └── manifest.json
├── src/
│   ├── components/        # UI components
│   │   ├── Footer.jsx
│   │   ├── Hero.jsx
│   │   ├── HomeArticles.jsx
│   │   ├── NavbarComponent.jsx
│   │   ├── NewsLetter.jsx
│   │   └── TopStories.jsx
│   ├── context/
│   │   └── GeneralContext.jsx
│   ├── pages/             # route-level pages
│   │   ├── CategoryPage.jsx
│   │   ├── Home.jsx
│   │   └── NewsPage.jsx
│   ├── styles/            # CSS files
│   │   ├── CategoryPage.css
│   │   ├── Footer.css
│   │   ├── Hero.css
│   │   ├── Home.css
│   │   ├── HomeArticles.css
│   │   ├── Navbar.css
│   │   ├── NewsLetter.css
│   │   ├── NewsPage.css
│   │   └── TopStories.css
│   ├── App.css
│   ├── App.js
│   ├── App.test.js
│   ├── index.css
│   ├── index.js
│   ├── logo.svg
│   ├── reportWebVitals.js
│   └── setupTests.js
├── package.json
├── package-lock.json
└── README.md
```

---

5. Component Documentation

NavbarComponent

- **Purpose:** Top navigation bar with category links.
- **Notes:** Should integrate with routing.

Hero

- **Purpose:** Introductory banner on the home page.
- **Props:** Texts or image sources.

HomeArticles

- **Purpose:** Displays a collection of featured articles.
- **Props:** Articles array.

TopStories

- **Purpose:** Shows top trending stories.

NewsLetter

- **Purpose:** Subscription form component.
- **Props:** Handles email input.

Footer

- **Purpose:** Footer with basic info/links.

GeneralContext

- **Purpose:** Global context for managing app-wide state (like selected category).

Pages

- Home — assembles Hero, Articles, Top Stories, Newsletter.
- CategoryPage — displays articles for a chosen category.
- NewsPage — renders a detailed view for a selected news item.

---

6. Running the Application

PrerequisitesRunning

- Node.js and npm installed.

Steps

Clone the repo:
git clone <repo-url>

1. cd code

2. Install dependencies:
   npm install
3. Start the development server:
   npm start
4. Open http://localhost:3000 in your browser.

---

7. Component Documentation

Components in ``:

- NavbarComponent.jsx → Provides navigation across categories and pages.
- Hero.jsx → Displays a hero banner for the home page.
- HomeArticles.jsx → Renders a grid/list of featured articles.
- TopStories.jsx → Highlights trending stories.
- NewsLetter.jsx → Newsletter subscription form (static for now).
- Footer.jsx → Displays footer with links/info.

Pages in ``:

- Home.jsx → Combines Hero, Articles, Top Stories, Newsletter.
- CategoryPage.jsx → Displays category-specific articles.
- NewsPage.jsx → Renders the detailed view of a selected article.

Context:

- GeneralContext.jsx → Provides global state (like selected category) to components.

---

8. State Management

- React Hooks: Uses useState and useEffect for local component state.
- Context API: GeneralContext.jsx supplies shared state across multiple components.
- No Redux or external state libraries — lightweight setup suitable for small-scale apps.

---

9. User Interface

- Home Page: Hero banner + featured articles + top stories + newsletter signup.
- Category Page: Articles filtered by category (via route /category/:name).
- News Page: Detailed article rendering (currently static/mocked).
- Navigation: Navbar provides easy access to categories.
- Footer: Wraps up the page with basic info and links.

---

10. Styling

- All styles in `` folder.
- Each component/page has its own CSS file (Hero.css, Navbar.css, etc.).
- Global styles in App.css and index.css.
- Approach: Semantic class names, modular CSS, responsive design under progress.
- No Tailwind or external CSS frameworks — pure CSS fundamentals.

---

11. Testing

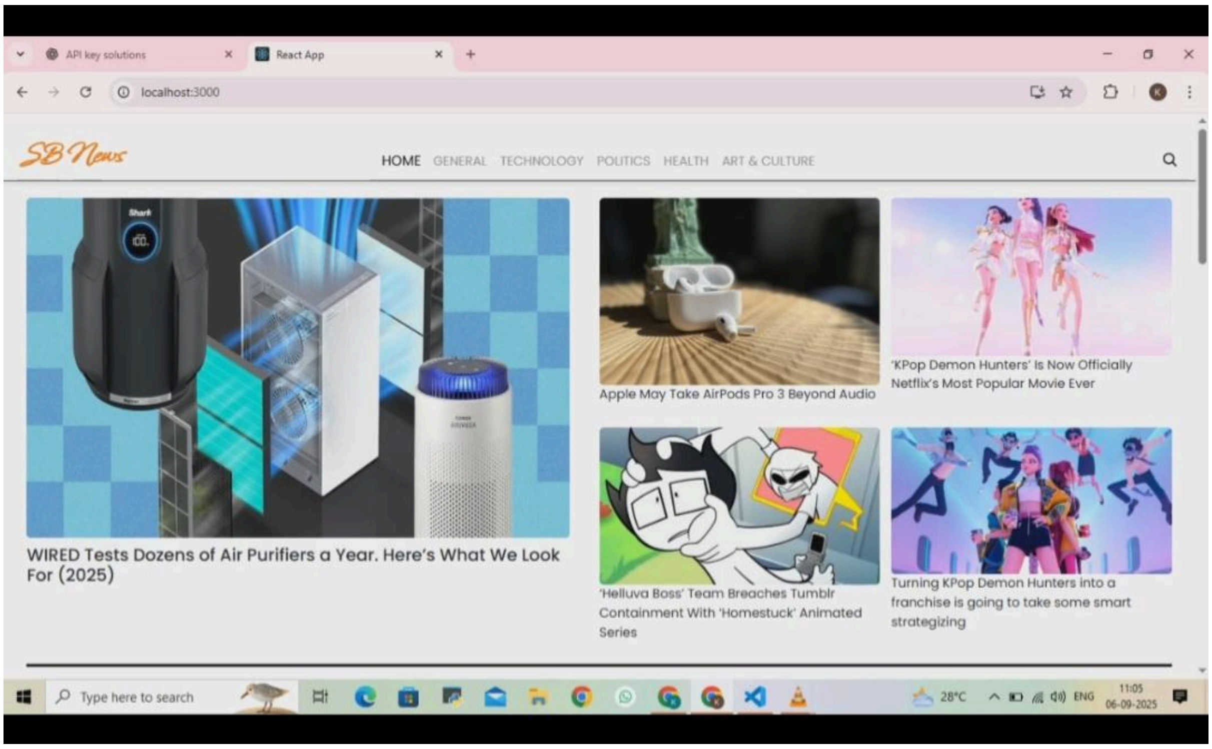
- Frameworks included: Jest + React Testing Library (via App.test.js and setupTests.js).
- What to test:
  - Snapshot tests for Hero, Navbar, and Article components.
  - Context values rendering properly.
  - Routing works for Home, Category, and News pages.
- Run tests:
  ```
  npm test
  ```

---

12. Screenshot:

- Screenshot 1: Home Page view.

- Screenshot 2: Category Page.



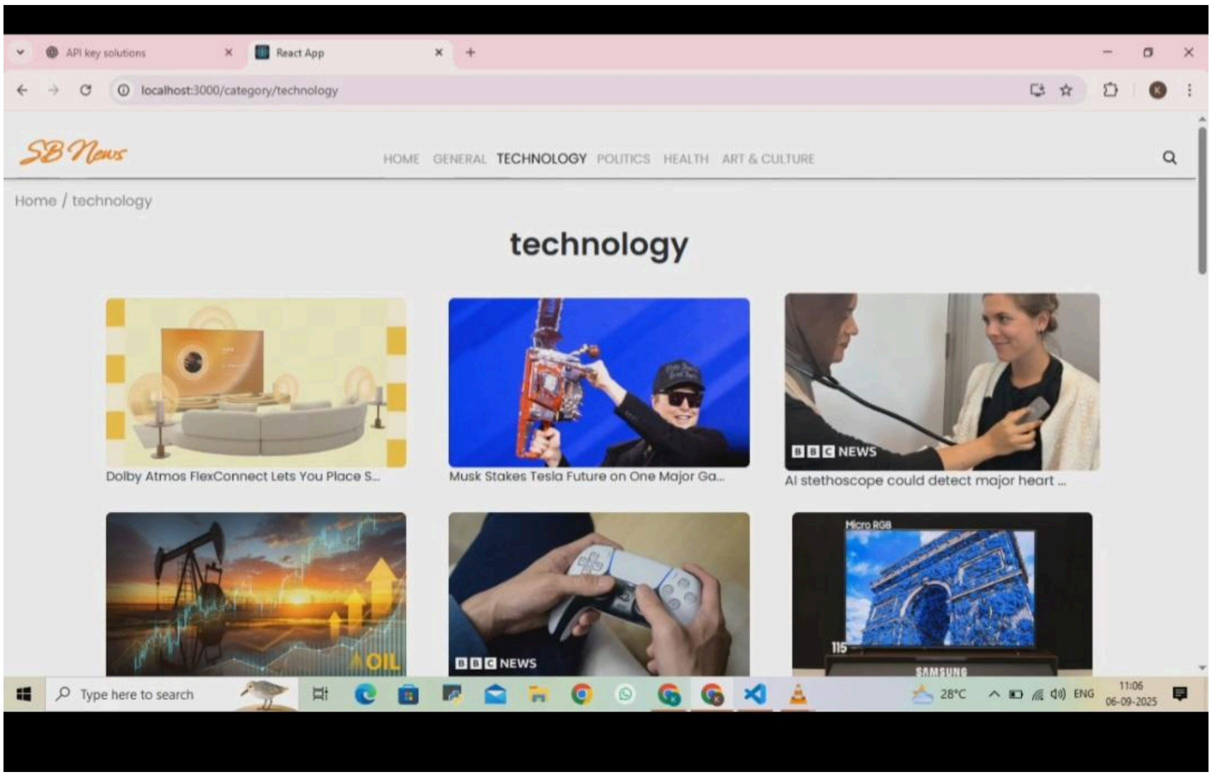
WIRED Tests Dozens of Air Purifiers a Year. Here's What We Look For (2025)

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

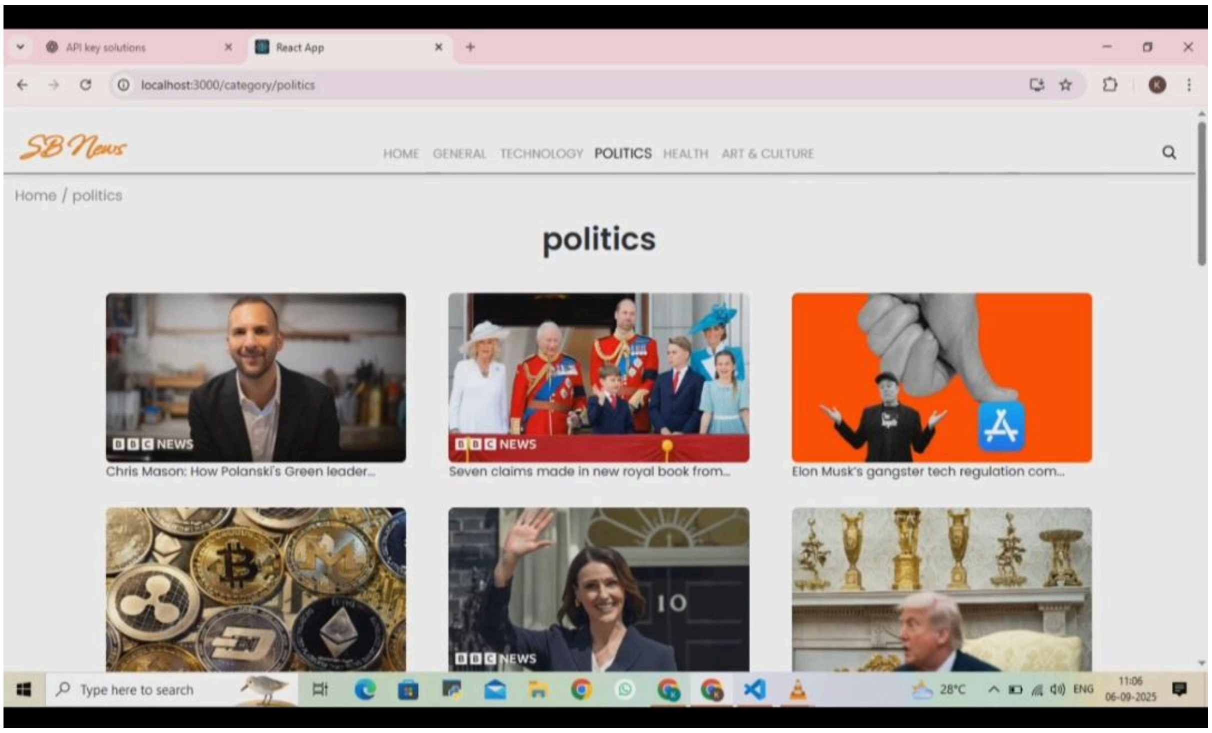


technology

Dolby Atmos FlexConnect Lets You Place S...

Musk Stakes Tesla Future on One Major Ga...

AI stethoscope could detect major heart ...



API key solutions | React App

localhost:3000/category/politics

SB News

HOME GENERAL TECHNOLOGY POLITICS HEALTH ART & CULTURE

Home / politics

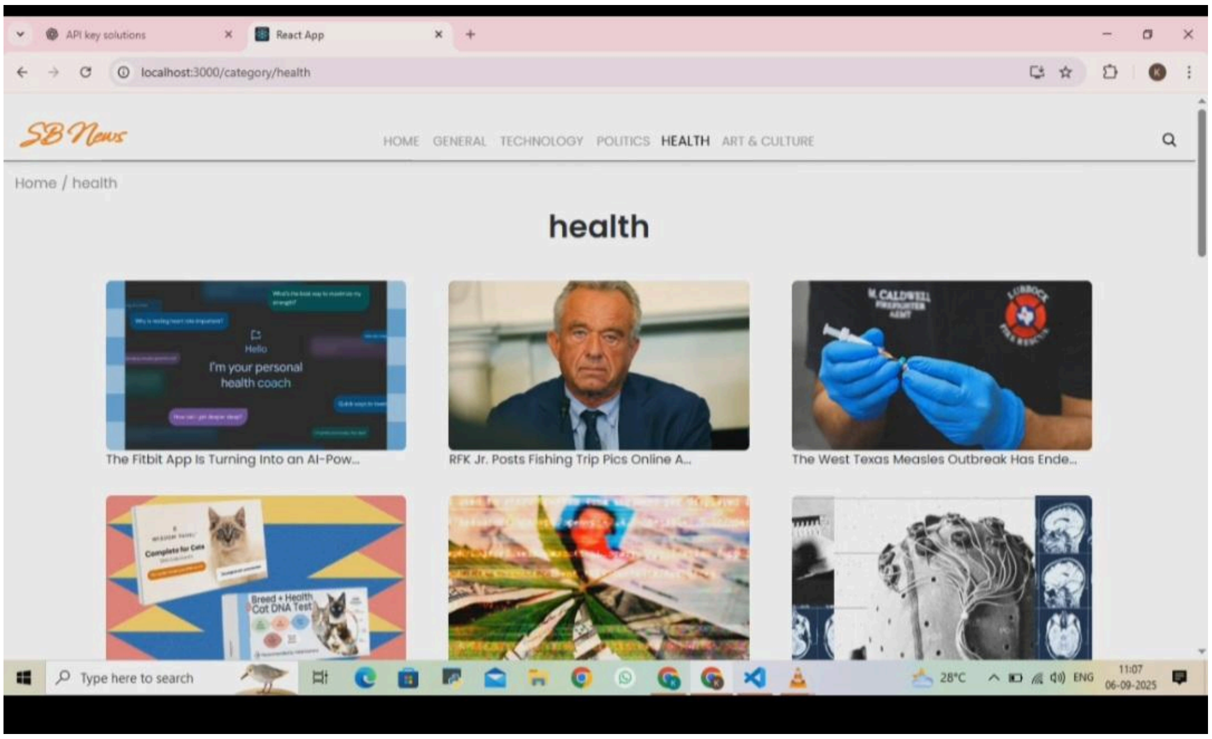
politics

Chris Mason: How Polanski's Green leader...

Seven claims made in new royal book from...

Elon Musk's gangster tech regulation com...

Type here to search 28°C ENG 11:06 06-09-2025



API key solutions | React App

localhost:3000/category/health

SB News

HOME GENERAL TECHNOLOGY POLITICS HEALTH ART & CULTURE

Home / health

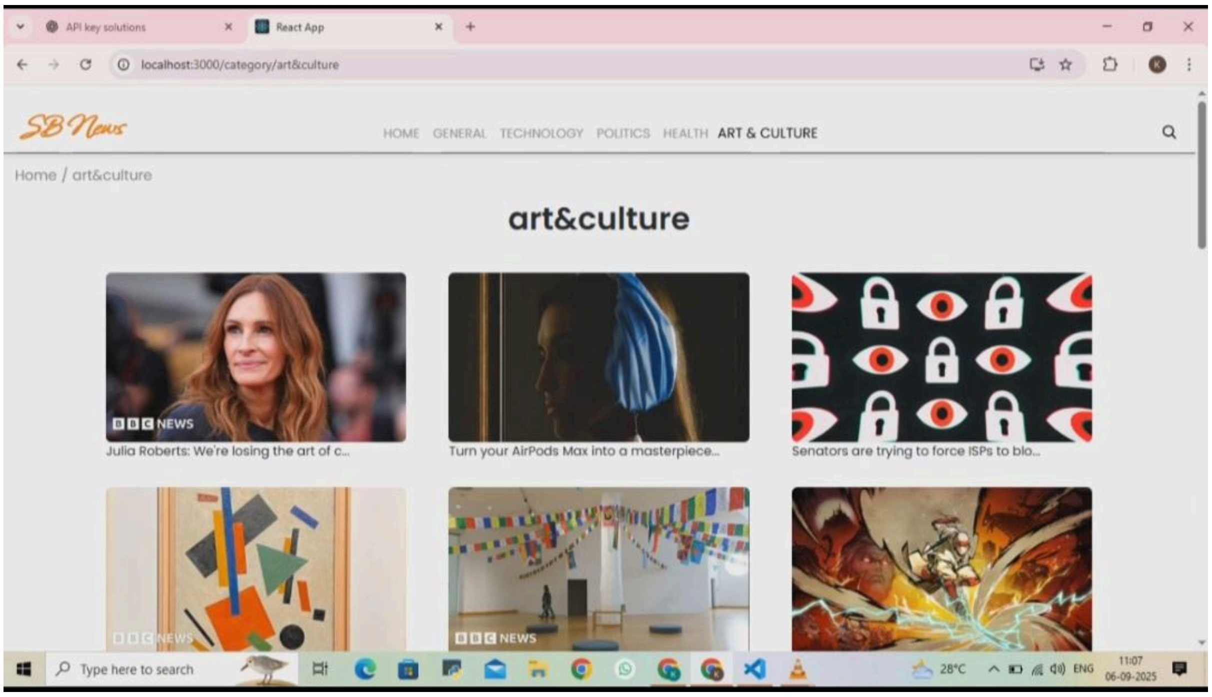
health

The Fitbit App Is Turning Into an AI-Pow...

RFK Jr. Posts Fishing Trip Pics Online A...

The West Texas Measles Outbreak Has Ende...

Type here to search 28°C ENG 11:07 06-09-2025


API key solutions
React App
localhost:3000/category/art&culture
SB News
HOME GENERAL TECHNOLOGY POLITICS HEALTH ART & CULTURE
Home / art&culture
art&culture
Julia Roberts: We're losing the art of c...
Turn your AirPods Max into a masterpiece...
Senators are trying to force ISPs to blo...
Type here to search
28°C
ENG
11:07
06-09-2025

13. Known Issues

- API integration is not included in this code — must be added manually.
- Styles may need refinement for full responsiveness.
- No authentication/authorization implemented.

---

14. Future Enhancements

- Integrate a live News API (GNews, NewsAPI, etc.).
- Add search functionality.
- Improve responsive design for mobile devices.
- Add bookmarking feature for saved articles.
- Add dark/light mode .