

# **INSIGHTSTREAM:NAVIGATE THE NEWS**

## **1. Introduction**

Project Title: INSIGHTSTREAM:NAVIGATE THE NEWS

Team Members:

Team leader: Hema Malini G([hemamalini3104@gmail.com](mailto:hemamalini3104@gmail.com))

Team member: Keerthana S([sanjaykrishna0327@gmail.com](mailto:sanjaykrishna0327@gmail.com))

Team member: Hemasri B ([hema872005@gmail.com](mailto:hema872005@gmail.com))

Team member: Aishwarya R ([aishwaryarn12@gmail.com](mailto:aishwaryarn12@gmail.com))

## **2. Project Overview**

### **Purpose:**

InsightStream is a revolutionary web application designed to redefine how users discover and consume news. It offers an intuitive interface, dynamic search, and a vast range of news categories to enhance the news exploration experience. The platform aggregates news from multiple sources and provides a seamless browsing experience for all types of users.

### **Features:**

- News from API Sources: Fetches real-time news updates from external APIs.
- Dynamic Search: Allows users to find relevant articles quickly.
- Categorized News: Users can explore news across multiple categories like Politics, Technology, Sports, Entertainment, and more.
- Trending News: Highlights the most discussed topics globally.
- User-Friendly Interface: Simple navigation and visually appealing UI.

## **3. Architecture**

### **Component Structure:**

1. The React-based frontend is divided into the following major components:
2. Navbar: Contains navigation links for different sections.
3. Hero Section: Displays featured news articles.
4. Categories Component: Fetches and displays news by category.
5. Search Bar: Allows users to search for specific topics.
6. Trending News Component: Highlights breaking and popular news.
7. News List: Displays a list of articles with brief descriptions and images.
8. Footer: Contains additional links and credits.
9. State Management
10. Local State: Used for UI interactions and temporary states (e.g., toggles, form inputs).
11. Global State: Managed using React Context API to handle API data, user preferences, and search results.

## Routing

Uses React Router for smooth navigation:

/ → Home page

/category/:categoryName → News filtered by category

/search/:query → News based on search queries

## 4. Setup Instructions

### Prerequisites

Ensure the following dependencies are installed:

1. Node.js & npm (Required to run the application)

2. Download Node.js
3. Git (For version control)
4. Download Git
5. React.js (For building the frontend)

## Installation

1. Clone the repository

`cd InsightStream`

2. Install dependencies

`npm install`

3. Run the development server

`npm start`

4. Open the application in a browser

Go to `http://localhost:3000`

## 5. Folder Structure

InsightStream/

```
| — src/  
|   | — components/ # Reusable UI components  
|   | — pages/      # Main page components (Home, Categories, Search)  
|   | — assets/     # Images, icons, and other assets
```

```
| |— utils/      # Helper functions and API calls
| |— context/   # Context API for state management
| |— App.js     # Main application component
| |— index.js   # Entry point
|— public/      # Static files
|— package.json # Project dependencies
|— README.md    # Documentation
```

## 6. Running the Application

- To start the frontend server locally:

```
npm start
```

- The application will be available at <http://localhost:3000>.

## 7. Component Documentation

### Key Components

1. `Navbar.jsx`: Contains navigation links for categories and search.
2. `HeroSection.jsx`: Displays the most trending news articles.
3. `CategoryNews.jsx`: Fetches and displays articles by category.
4. `SearchResults.jsx`: Displays search results based on user input.
5. `NewsCard.jsx`: Reusable component for displaying news articles.

6. Footer.jsx: Contains credits and links.

## **8. State Management**

1. Global State
  - Managed using React Context API
2. Stores:
  - API-fetched news articles
3. User search queries
  - Favorite/saved articles
4. Local State
  - Used for handling UI interactions such as toggles and modal visibility.

## **9. Styling**

- CSS Frameworks/Libraries
- Tailwind CSS (For utility-based styling)
- Bootstrap (For responsive UI components)

## **Theming**

- Dark Mode toggle
- Custom styles for different news categories

## **10. Testing**

Testing Strategy:

- Unit Testing: Tests individual components using Jest and React Testing Library.
- Integration Testing: Ensures API requests work correctly.
- Code Coverage
- Uses Jest for test coverage reports.

## **11. Demo video**

Live Demo: Demo Link

[https://drive.google.com/file/u/0/d/1QeOfj6pHS6-ncc\\_z6](https://drive.google.com/file/u/0/d/1QeOfj6pHS6-ncc_z6)

## **12. Known Issues**

- Search results delay: Sometimes, API requests take longer than expected.
- UI inconsistencies in some browsers: Some styling issues in Safari.

## **13. Future Enhancements**

- User Authentication: Allow users to create accounts and save articles.

- **Personalized News Feed:** Customize news recommendations based on user preferences.
- **Notifications:** Send push notifications for breaking news.
- **Dark Mode Improvements:** Enhance theme switching experience.