

## Insight Stream – News Website Documentation

### 1. Introduction

Project Title: Insight Stream –navigate the new landscape

Team Members:

Ganga K– Frontend Developer

[Janani P] – DBA Designer

[Gobika M] – Backend Integration

[Dharani M] \_ Backend Integration

---

### 2. Project Overview

Purpose:

The goal of Insight Stream is to provide users with real-time, categorized, and reliable news articles in a clean and interactive interface.

Features:

Live news updates fetched via API.

Category-wise filtering (Politics, Technology, Sports, Entertainment, etc.).

Search functionality for specific news topics.

Responsive design for mobile and desktop users.

Bookmarking favorite articles.

---

### 3. Architecture

Component Structure:

App.js → Root component.

Navbar.js → Navigation bar with categories.

NewsList.js → Displays news articles.

NewsCard.js → Individual news article card.

SearchBar.js → For searching news topics.

Bookmark.js → Saved articles section.

State Management:

Implemented using Context API for global state.

Handles news data, bookmarks, and category filtering.

Routing:

Implemented with React Router DOM.

Routes: /, /category/:name, /search, /bookmarks.

---

#### 4. Setup Instructions

Prerequisites:

Node.js (v16+ recommended)

npm or yarn

Installation:

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

```
cd insight-stream
```

```
npm install
```

```
npm start
```

---

## 5. Folder Structure

/insight-stream

├ /public

├ /src

│ └ /components

│ │ └ Navbar.js

│ │ └ NewsCard.js

│ │ └ SearchBar.js

│ │ └ Bookmark.js

│ └ /pages

│ │ └ Home.js

│ │ └ Category.js

```
| | └─ SearchResults.js
| | └─ Bookmarks.js
| └─ /context
| | └─ NewsContext.js
| └─ App.js
| └─ index.js
└─ package.json
```

---

## 6. Running the Application

Frontend:

npm start

Runs the app in development mode at <http://localhost:3000/>.

---

## 7. Component Documentation

Key Components:

Navbar → Navigation across categories.

NewsList → Renders fetched articles.

NewsCard → Displays article title, description, image, and source.

Reusable Components:

Button, Loader, Modal.

---

## 8. State Management

Global State:

News API data, selected category, search results.

Local State:

Pagination handling, modal visibility, input fields.

---

## 9. User Interface

Responsive Material UI + Tailwind CSS design.

Mobile-friendly layout.

Screenshots/GIFs showing:

Home Page with latest news.

Category selection.

Bookmarked articles view.

---

## 10. Styling

CSS Frameworks/Libraries:

Tailwind CSS

Material UI Components

Theming:

Light/Dark mode toggle.

---

## 11. Testing

Testing Strategy:

Unit tests for components with Jest.

UI testing with React Testing Library.

Code Coverage:

Ensuring >80% coverage.



---

## 12. Screenshots or Demo

Screenshots of homepage, categories, and bookmarks.

Demo link :

[https://drive.google.com/file/d/1aLsvU\\_rUGbS5cm3A5oEr3w284YFTPqyK/view?usp=drive\\_link](https://drive.google.com/file/d/1aLsvU_rUGbS5cm3A5oEr3w284YFTPqyK/view?usp=drive_link)

---

## 13. Known Issues

Sometimes API rate limits may cause delays.

Search performance can be optimized for large datasets.

---

## 14. Future Enhancements

Add AI-powered news summarization.

Implement push notifications for breaking news.

Support for multiple languages.

Progressive Web App (PWA) support.