

# InsightStream - News Web Application

## Phase 1: Project Proposal

### 1. Project Title: InsightStream

### 2. Team Details:

Name	Email
<b>Sonashree V – Team Leader</b>	<b>sonaroshan4@gmail.com</b>
<b>Dheekshitha K</b>	dheekshithakalidass@gmail.com
<b>Lakshitha B</b>	lakshithabasker2005@gmail.com
<b>Sweatha K M</b>	sweathamurugan02@gmail.com
<b>Jennifer A L</b>	aruleojenni@gmail.com

### 3. Project Objectives:

- To develop a **real-time news web application**.
- To provide users with categorized news updates.
- To allow users to filter and access trending news easily.
- To implement a **newsletter subscription feature** for updates.

- To ensure a **responsive and interactive user experience**.

#### 4. Expected Outcome:

- A **fully functional** news application built using **React.js**.
- Real-time news fetched from an API.
- Optimized and well-structured frontend architecture.

#### 5. Tools & Technologies:

- **Frontend:** React.js, Bootstrap, Axios
  - **State Management:** React Hooks (useState, useEffect)
  - **API:** News API for fetching news articles
  - **Version Control:** GitHub for source code management
- 

## Phase 2: System Design

### 1. Architecture Overview

The application follows a **component-based architecture** with modular design principles. The key aspects include:

#### Component Structure

- **NavbarComponent.jsx** – Handles navigation between different pages.
- **HeroSection.jsx** – Displays featured news.
- **CategoryPage.jsx** – Renders news based on categories.
- **Newsletter.jsx** – Manages the newsletter subscription.
- **NewsList.jsx** – Dynamically renders news articles.

## State Management

- **Local State Management:** The `useState` and `useEffect` hooks manage API responses and user interactions.
- **Data Fetching:** News articles are retrieved using **Axios** from an external API.

## Routing Structure

- **React Router DOM** is used for seamless navigation between pages.
  - Paths:
    - `/` → Home Page
    - `/category/:id` → Category Page
    - `/subscribe` → Newsletter Subscription
- 

## Phase 3: Implementation

### 1. Folder Structure

The project is structured as follows:

- **public/:** Contains static files such as `index.html` and the `_redirects` file for routing.
- **src/:** Main source directory containing React components, pages, and logic.
  - **components/:** Includes reusable components like `NavbarComponent.jsx`, `HeroSection.jsx`, `Newsletter.jsx`, and `NewsList.jsx`.
  - **pages/:** Contains individual pages like `CategoryPage.jsx` for rendering category-based news.

- **App.jsx**: The root component that integrates all pages and components.
- **index.js**: Entry point of the application, responsible for rendering the App.jsx component.
- **package.json**: Defines project dependencies and scripts for running and deploying the application.

## 2. Setup Instructions

### Prerequisites

- Install **Node.js** and **Git**
- Recommended: **Visual Studio Code**

### Installation Steps

1. Clone the repository:
2. git clone  
`https://github.com/Sona4suriya/InsightStream_NewsApp.git`
3. Navigate to the project directory:
4. `cd InsightStream_NewsApp`
5. Install dependencies:
6. `npm install`
7. Start the application:
8. `npm start`
9. Access the application at <http://localhost:3000>.

## Phase 4: Testing & Deployment

### 1. Testing Strategy

- **Manual Testing**: UI tested across different screen sizes.

- **Console Debugging:** Console logs used for API response validation.

## 2. Deployment Process

### Running the Application Locally

This project has not been deployed online. To run the application, follow these steps:

1. Clone the repository:
2. git clone  
`https://github.com/Sona4suriya/InsightStream_NewsApp.git`
3. Navigate to the project folder:
4. `cd InsightStream_NewsApp`
5. Install dependencies:
6. `npm install`
7. Start the application:
8. `npm start`
9. Open <http://localhost:3000> in a web browser.

The application will work if run using **VS Code** and the above steps.

## 3. Known Issues

- **API Limitations:** News API has request limits.
- **CORS Restrictions:** Certain news sources may not display images.

#### **4. Future Enhancements**

- Implement **User Authentication**.
- Add **Dark Mode** for better UI experience.
- Introduce **Multi-Language Support**.

#### **Submitted By:**

- Sonahree V (Team Leader)
- Dheekshitha K
- Lakshitha B
- Sweatha K M
- Jennifer A L

Date: 06.03.2025