**Project Documentation: Navigate the News Landscape**

**Introduction**

**Project Title:** Navigate the News Landscape (InsightStream)

**Team Members:**

* [Your Name] – Neelaveni D

[Teammate’s Name] – Nathiya C

* [Teammate’s Name] – Praveena X
* [Teammate’s Name] –Priyadharshini D

**Project Overview**

**Purpose:**  
The project helps users explore, analyze, and interact with current news content in a structured and engaging way. It aims to improve critical thinking by presenting multiple perspectives, highlighting media bias, and encouraging informed decision-making.

**Features:**

* Interactive news feed with categorized articles (politics, technology, sports, etc.)
* Bias indicators to show potential leanings of sources
* Search and filter functionality
* Article comparison feature for viewing the same news story from different outlets
* User-friendly and responsive UI

**Architecture**

**Component Structure:**

* **App.js** – Root component handling global layout and routing
* **Navbar** – Navigation menu for categories and search
* **NewsFeed** – Displays list of articles dynamically
* **ArticleCard** – Reusable component for each article preview
* **ArticleDetail** – Detailed article page with related sources
* **BiasIndicator** – Component showing media bias rating
* **Footer** – Application footer with credits/links

**State Management:**

* **Context API** used for managing global state (e.g., user preferences, article filters).

**Routing:**

* Implemented with **React Router**:
  + / – Home (all articles)
  + /category/:name – Category-specific articles
  + /article/:id – Detailed article view

**Setup Instructions**

**Prerequisites:**

* Node.js (>= 16)
* npm or yarn

**Installation:**

bash

Copy code

git clone https://github.com/your-repo/insightstream.git

cd insightstream

npm install

**Environment Variables:**

* API\_KEY for news API integration

**Folder Structure**

bash

Copy code

/client

/src

/components

Navbar.js

ArticleCard.js

BiasIndicator.js

Footer.js

/pages

Home.js

Category.js

ArticleDetail.js

/assets

/images

/context

AppContext.js

/utils

api.js

helpers.js

**Running the Application**

bash

Copy code

npm start

Runs the development server at http://localhost:3000.

**Component Documentation**

**Key Components:**

* **ArticleCard:** Props – title, image, summary, biasLevel
* **BiasIndicator:** Props – biasScore (number), sourceName
* **NewsFeed:** Props – articles (array of article objects)

**Reusable Components:**

* Button, Loader, Modal

**State Management**

**Global State:**

* User preferences (theme, filters) stored in Context API
* Articles fetched once and distributed across components

**Local State:**

* Search query in Navbar
* Article expansion toggle in ArticleDetail

**User Interface**

* Clean and modern layout with emphasis on readability
* Dark and light theme support
* Responsive grid for mobile and desktop

**Styling**

* **Tailwind CSS** for fast and consistent styling
* Custom color palette to differentiate categories
* Bias indicator uses dynamic color coding (green = balanced, red = biased)

**Testing**

* **Unit Testing:** Jest + React Testing Library for components
* **Integration Testing:** Ensuring article fetch and display logic works correctly
* **End-to-End Testing:** Cypress for navigation and user flow tests

**Screenshots or Demo**

* Screenshot of homepage with news feed
* Screenshot of article detail view
* [Insert demo link if available]

**Known Issues**

* API rate limits can sometimes cause articles not to load
* Bias scoring is experimental and may not always be accurate

**Future Enhancements**

* User login system with saved article preferences
* AI-powered article summaries
* More advanced bias detection models
* Offline reading mode

Top of Form

Bottom of Form