**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

| 7 march 2025 | 9 march 2025 |
| --- | --- |
| Team ID | SWT1D1741164806148446 |
| Project Name | InsightStream |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

The technical architecture of InsightStream is a client-side, single-page application (SPA) built with React.js. It follows a component-based structure where the **NewsAPI** serves as the external data source, delivering real-time news data. The architecture consists of a **User Interface** layer for rendering news articles and navigation, an **Application Logic** layer for fetching and processing data, and a **State Management** system using React Context API to handle data flow across components. Routing is managed via react-router-dom to enable seamless navigation between the homepage, category pages, and search results. Styling is achieved with a combination of Bootstrap for responsive layouts and custom CSS in App.css for a cozy, warm aesthetic. The app runs locally on http://localhost:3000 during development and can be built into static files for deployment.

**Architectural Diagram (Textual Description):**

* **External Entity:** NewsAPI → Provides news data (trending, category-specific, search-based).
* **Client-Side Application:**
  + **UI Layer:** Browser renders Navbar, Hero, NewsCard, Footer (HTML, CSS, React.js).
  + **Logic Layer:** NewsContext.js fetches data via API calls (axios), processed in React components.
  + **State Store:** Context API (NewsContext.js) holds topNews, searchResults, etc.
  + **Routing:** App.js manages routes (/, /category/:cat, /search).
* **Output:** User sees news articles in a cozy, responsive interface.

**Table-1 : Components & Technologies:**

| **S.No** | **Component** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | User Interface | Web-based interface for News | HTML, CSS, JavaScript / React Js etc. |
|  | Application Logic-1 | News fetching from News api | React js |
|  | Application Logic-2 | News fetching based on genre | React js |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | Frontend frameworks | React.js, BootStrap, Tailwind CSS |

**References:**

**React.js Documentation:** [**https://reactjs.org/**](https://reactjs.org/)

**Bootstrap Documentation:** [**https://getbootstrap.com/docs/5.3/getting-started/introduction/**](https://getbootstrap.com/docs/5.3/getting-started/introduction/)

**React Router DOM Documentation:** [**https://reactrouter.com/**](https://reactrouter.com/)

**Axios Documentation:** [**https://axios-http.com/docs/intro**](https://axios-http.com/docs/intro)

**NewsAPI Documentation:** [**https://newsapi.org/docs**](https://newsapi.org/docs)

**Create React App Documentation:** [**https://facebook.github.io/create-react-app/docs/getting-started**](https://facebook.github.io/create-react-app/docs/getting-started)

### **Explanation and Alignment with InsightStream**

#### **Technical Architecture**

* **The SPA architecture leverages React’s component model, with App.js as the root tying together UI (Navbar, Footer) and page components (Home, Category, SearchResults).**
* **NewsContext.js centralizes data fetching and state, using axios to connect to NewsAPI endpoints (e.g., top-headlines, everything).**
* **The cozy UI is a mix of Bootstrap’s grid system and custom CSS, ensuring responsiveness and a unique aesthetic.**

#### **Table-1: Components & Technologies**

* **User Interface: Built with React.js components (NewsCard, etc.), styled with Bootstrap and CSS.**
* **Application Logic-1 & 2: NewsContext.js handles trending and category fetches, executed in Home.js and Category.js.**
* **Application Logic-3: Search logic in Navbar.js and SearchResults.js.**
* **State Management: Context API avoids prop drilling, keeping code clean.**

#### **Table-2: Application Characteristics**

* **Open-Source Frameworks: React.js for the frontend, Bootstrap for responsive design—no Tailwind here, but it’s an option for future iterations.**
* **Routing: react-router-dom powers navigation, critical for a multi-page feel in an SPA.**
* **API Integration: axios ensures reliable NewsAPI calls.**
* **Styling: Custom CSS (App.css) adds the cozy touch atop Bootstrap.**
* **Development Environment: Node.js and CRA streamline setup and builds.**