**Project Design Phase-II**

**Data Flow Diagram & User Stories**

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

**Data Flow Diagrams:**

A Data Flow Diagram (DFD) visually represents how data moves through the InsightStream application—from external sources to processes, storage, and the user interface. The core of InsightStream’s data flow revolves around the **NewsAPI**, which acts as the primary external entity providing real-time news data. This data flows through our app’s processes (fetching, processing, rendering), gets stored temporarily in state (via Context API), and is ultimately displayed to the user. The DFD helps us see how user interactions—like browsing categories or searching—trigger these flows.

#### **Key Components in the DFD**

* **External Entity:** NewsAPI (source of news data).
* **Processes:**
  + Fetching data (API calls).
  + Processing data (filtering, formatting in Context).
  + Rendering data (UI display via components).
* **Data Store:** React Context (NewsContext.js) for temporary storage of news data.
* **Data Flows:** User inputs (e.g., search queries, category clicks) and API responses (news articles).
* **User:** The end-user interacting with the app.

#### **Example DFD Description**

Here’s a textual representation of InsightStream’s Level-0 (Context) DFD:

* **External Entity: NewsAPI**
  + Sends: Real-time news data (trending articles, category-specific articles, search results).
  + Flow: Data flows to the "Fetch News" process.
* **Process 1: Fetch News (NewsContext.js)**
  + Receives: User inputs (e.g., category selection from Navbar, search query).
  + Action: Uses axios to call NewsAPI endpoints (e.g., top-headlines for trending news, everything for searches).
  + Sends: Fetched news data to the "Store News" data store.
* **Data Store: News Context (NewsContext.js)**
  + Stores: topNews, searchResults, and category-specific data (temporary state).
  + Flow: Provides data to the "Render UI" process.
* **Process 2: Render UI (App.js, Pages, Components)**
  + Receives: News data from Context.
  + Action: Renders data into UI components (Hero, NewsCard) based on the route (Home, Category, SearchResults).
  + Sends: Visual output to the user.
* **User**
  + Sends: Inputs like category clicks or search terms via Navbar.
  + Receives: Displayed news articles in the browser.

**Flow Example:**

1. User clicks "Technology" in Navbar.
2. Category.js triggers fetchNewsByCategory("technology") in NewsContext.js.
3. NewsAPI returns tech news articles.
4. Data is stored in Context as categoryNews.
5. Category.js renders the articles as NewsCards for the user to see.

This DFD shows how the NewsAPI is the heart of our data flow, driving everything from trending news to user searches.

**Example:**



**User Stories**

**User Story Table – Music Streaming App**

| **User Type** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Acceptance Criteria** | **Priority** | **Release** |
| --- | --- | --- | --- | --- | --- | --- |
| **News Reader** | Trending News | USN-1 | As a user, I can see the trendy news . | I can see the News of all topics. | High | Sprint-1 |
| **News Reader** | Category Browsing | USN-2 | As a user, I can watch Political,Business, and Technology News. | I can also go the source and read the full news. | High | Sprint-1 |
| **News Reader** | Global News Access | USN-3 | As a user, I can able to view the complete news across the world. | I can also watch the news for multiple pages . | Medium | Sprint-2 |
| **News Reader** | Subscription | USN-4 | As a user, I can even subscribe. | I can Subscribe to the newsletter for the news . | High | Sprint-2 |

#### **Explanation of User Stories**

* **USN-1 (Trending News):** Covers the homepage experience—users expect fresh, trending content right away. It’s high-priority since it’s the app’s first impression.
* **USN-2 (Category Browsing):** Lets users dive into their interests, like Tech or Business. Linking to sources adds value—another core feature, so it’s high-priority.
* **USN-3 (Global News Access):** Expands the scope to worldwide news, with a nod to future pagination (not yet implemented). Medium priority as it builds on the basics.
* **USN-4 (Subscription):** The newsletter form is a placeholder now, but it’s a key engagement feature, so it’s high-priority for Sprint-2.