**Project Design Phase**

**Solution Architecture**

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

#### **Solution Architecture**

* **Overview:**

The solution architecture for **InsightStream**, a web-based news application, ensures a scalable, high-performance, and engaging platform for delivering real-time news from credible sources. The architecture prioritizes seamless data retrieval, intuitive navigation, and a cozy, responsive user experience to enhance engagement. It addresses existing problems in news consumption—fragmentation, lack of personalization, and poor accessibility—by leveraging modern web technologies. This design outlines the structure, behavior, and characteristics of the software, defining its features, development phases, and requirements for stakeholders, while providing specifications for its implementation, management, and delivery.

**Purpose:**

* Deliver a robust technical solution to consolidate news from multiple categories into a single platform.
* Describe the system’s components, interactions, and behavior to project stakeholders.
* Define key features (trending news, category browsing, search), development phases (sprints), and solution requirements (functional and non-functional).
* Ensure the app is scalable, maintainable, and user-centric.
* **Architecture Description:**

InsightStream is a client-side, single-page application (SPA) built with React.js, utilizing a component-based architecture. It integrates with the **NewsAPI** as the external data source, fetching real-time news via RESTful API calls. The app’s structure is divided into three main layers:

1. **Presentation Layer:** Handles the user interface, rendering news articles and navigation elements with a warm, cozy design.
2. **Application Logic Layer:** Manages data fetching, state management, and routing to deliver dynamic content.
3. **Data Layer:** Interfaces with the NewsAPI to retrieve and process news data, stored temporarily in application state.

**Key Characteristics:**

* **Scalability:** Modular components and stateless design allow for future features (e.g., pagination, user accounts).
* **Performance:** Optimized API calls and flicker-free UI transitions ensure quick load times (target <2 seconds).
* **Behavior:** Real-time data updates on page load, with user-driven interactions (category clicks, searches) triggering new fetches.
* **Responsiveness:** Bootstrap ensures cross-device compatibility (desktop, mobile).

**Development Phases:**

* **Phase 1 (Sprint-1):** Core features—category browsing and search functionality.
* **Phase 2 (Sprint-2):** Homepage with trending news.
* **Phase 3 (Sprint-3):** Full article linking and UI polish.
* **Phase 4 (Sprint-4):** Newsletter subscription and scalability enhancements.

**Solution Requirements Addressed:**

* **Functional:** Trending news (FR-1), category browsing (FR-2), search (FR-4), newsletter (FR-3).
* **Non-functional:** Usability (NFR-1), performance (NFR-4), reliability (NFR-3), scalability (NFR-6).

#### **Example - Solution Architecture Diagram (Textual Representation)**

Below is a conceptual description of InsightStream’s architecture diagram, as a visual isn’t directly possible here. You can sketch this for submission:

* **External Entity:**
  + **NewsAPI:** Box labeled "NewsAPI" with an arrow pointing right (data output).
  + Description: Provides RESTful endpoints (e.g., /top-headlines, /everything).
* **Client Application (InsightStream):**
  + **Box: Presentation Layer**
    - Components: Navbar, Hero, NewsCard, Footer.
    - Tech: HTML, CSS (App.css), React.js, Bootstrap.
    - Arrow: Output to "User" (rendered UI).
  + **Box: Application Logic Layer**
    - Components: App.js (routing), Home.js, Category.js, SearchResults.js.
    - Tech: React.js, React Router DOM.
    - Arrow: Bidirectional with "State Management" (data flow).
  + **Box: Data Layer (State Management)**
    - Component: NewsContext.js.
    - Tech: React Context API, Axios.
    - Arrow: Input from "NewsAPI" (fetched data), output to "Application Logic" (processed data).
* **User:**
  + Box labeled "User" with an arrow from "Presentation Layer" (receives UI) and an arrow to "Presentation Layer" (inputs like clicks, searches).

**Solution Architecture Diagram:**

