■ Frontend Development with React.js – Project Documentation

Introduction

Project Title: InsightStream – Navigate the News Landscape

Team Leader: SANJAY.N

Team Members: SANTHOSH.K,SATHISH.M,SANJAI KUMAR.K

Project Overview

Purpose:

InsightStream is a revolutionary web application designed to redefine how people discover and consume news. It offers an intuitive interface, dynamic search, and a wide range of news categories for all types of users.

Features:

■ User Authentication (Login/Register)

■ Dashboard with real-time news updates

■ Responsive design for mobile and desktop

■ Dark/Light theme toggle

■ Dynamic search across multiple news categories

■ Personalized feeds based on user preferences

---

3. Architecture

Component Structure:

App.js → Navbar, Sidebar, MainContent, Footer

MainContent → NewsFeed, SearchBar, CategoryFilter NewsFeed → NewsCard, NewsDetails

State Management:

Redux Toolkit for handling user preferences, news feed data, and theme

Context API for authentication state

Routing (React Router DOM v6):

/ → Landing Page

/news → Latest headlines and feeds

/categories/:id → Category-specific news

/search → Search results page

/profile → User profile & saved articles

---

4. Setup Instructions

Prerequisites: Node.js (v18+) npm or yarn Installation Steps: # Clone repository git clone https://github.com/your-repo/insightstream.git

Navigate to client folder cd client

Install dependencies npm install

Setup environment variables cp .env.example .env

---

5. Folder Structure

client/

■■■ public/ # Static assets

■■■ src/

■ ■■■ assets/ # Images, icons, fonts

■ ■■■ components/ # Reusable UI components

■ ■■■ pages/ # Page-level components

■ ■■■ hooks/ # Custom React hooks

■ ■■■ utils/ # Helper functions

■ ■■■ context/ # Context API (auth, theme)

■ ■■■ store/ # Redux slices (news, user, preferences)

■ ■■■ styles/ # CSS/SCSS files

■ ■■■ App.js

■ ■■■

index.js

Utilities:

Axios service for API calls (via RapidAPI)

Custom hooks: useNewsFetch(), useTheme()

Helper: formatDate() for consistent date formatting

Running the Application

Start Frontend Server:

```
npm start
```

Runs the app in development mode → http://localhost:3000

Component Documentation

Key Components:

Navbar: Navigation with links to Home, Categories, Search, Profile

NewsCard: Displays headline, image, source, snippet SearchBar: Provides keyword-based search functionality Reusable Components:

Button: Primary, secondary, outline variants

Modal: For article details, props: { isOpen, onClose, children }

ThemeToggle: Light/Dark mode switch

State Management

Global State:

Redux Toolkit slices: newsSlice, userSlice, preferencesSlice

Context API for theme and authentication

Local State: useState for search input, modals, and form handling useReducer for advanced filtering logic

---

User Interface

Screenshots/GIFs (to be added):

Home Page

Category News Page

Search Results

Responsive Mobile Layout

Styling

CSS Frameworks/Libraries:

Tailwind CSS for fast styling StyledComponents for dynamic theming

Theming:

Dark/Light mode toggle with persistence using localStorage

Testing

Testing Strategy:

Unit tests → Jest

Component tests → React Testing Library E2E tests → Cypress

Code Coverage:

Istanbul (nyc) integrated with CI/CD pipeline

Screenshots or Demo

Screenshots: [Insert images here]

Live Demo Link: [Provide deployed link if available]

Known Issues

Search results load slowly under weak networks

Dark mode toggle not fully optimized on mobile Safari

Future Enhancements

AI-powered news recommendations

Internationalization (i18n)

Offline reading mode (service workers) Social sharing integration

---

■ Scenario-Based Intro

Suppose you're rushing home after work, phone in hand, completely unaware of the world's headlines. You open InsightStream, and instantly, you're welcomed with a vibrant news dashboard.

Breaking headlines slide across your screen. Categories range from politics to sports, science, and culture. You quickly search "climate change" and receive curated, real-time articles from multiple global sources.

Unlike traditional news feeds, InsightStream feels different — fast, interactive, and personalized. It doesn't just deliver the news; it connects you with the pulse of the world.

■ Technical Architecture

Frontend: React.js (SPA with React Router)

State Management: Redux Toolkit + Context API

API Integration: RapidAPI (fetching global news feeds & data sources)

Backend (Optional): Node.js/Express for user profiles & authentication

Database: MongoDB (via Mongoose) for storing users, preferences, and saved articles

Deployment: Vercel (frontend) + MongoDB Atlas (database)