Frontend Development with React.js

Project Documentation

# Introduction

Project Title: InsightStream – Navigate the News Landscape

Team Members:  
- [Name] – Frontend Developer  
- [Name] – Backend Developer  
- [Name] – UI/UX Designer  
- [Name] – QA Engineer

# Project Overview

Purpose:  
InsightStream is a React-based web application designed to help users explore, filter, and analyze news across multiple sources. Its goal is to provide clarity in the modern information landscape by organizing news into topics, trends, and credibility layers.

Features:  
- Aggregated news feed from multiple APIs/sources.  
- Smart filtering (by topic, region, sentiment, source credibility).  
- Interactive visualizations of trending stories.  
- Personalized news dashboard with saved topics.  
- Dark/light theme support.

# Architecture

Component Structure:  
- NewsFeed – displays aggregated articles.  
- FiltersPanel – allows topic/source/date/sentiment filtering.  
- TrendsVisualization – charts and graphs for trending news.  
- ArticleCard – reusable card for individual stories.  
- Dashboard – personalized user view.

State Management:  
- Global state managed using Redux Toolkit.  
- Local state for UI components handled with React hooks.

Routing:  
- Implemented using React Router v6:  
 - / – Home (NewsFeed)  
 - /trends – TrendsVisualization  
 - /dashboard – Personalized Dashboard  
 - /article/:id – Full Article View

# Setup Instructions

Prerequisites:  
- Node.js (v18+)  
- npm or yarn

Installation:  
```bash  
git clone https://github.com/your-repo/insightstream.git  
cd insightstream/client  
npm install  
```  
- Configure .env with API keys for news data providers.

# Folder Structure

- /src/components – Reusable UI components (ArticleCard, Filters, Navbar).  
- /src/pages – Page-level components (Home, Dashboard, Trends).  
- /src/assets – Static assets (images, icons).  
- /src/store – Redux slices and configuration.  
- /src/hooks – Custom hooks for data fetching & filtering.  
- /src/utils – Helper functions (formatting dates, parsing sources).

# Running the Application

```bash  
npm start  
```  
Runs the development server at http://localhost:3000/.

# Component Documentation

Key Components:  
- NewsFeed – fetches and displays article lists.  
- FiltersPanel – accepts props for available filters and updates Redux state.  
- TrendsVisualization – uses Recharts for data visualization.  
- Dashboard – fetches user preferences and displays customized feed.

Reusable Components:  
- ArticleCard – standardized display of news info.  
- Button, Dropdown, Modal – styled UI elements with props for customization.

# State Management

Global State:  
- Articles, filters, user preferences stored in Redux slices.

Local State:  
- UI toggles (modal open, dropdown selection) managed with React hooks.

# User Interface

- Clean, modern design with emphasis on readability.  
- Responsive layout optimized for desktop and mobile.  
- Screenshots: [to be added].

# Styling

- Styled with TailwindCSS.  
- Supports light/dark mode via theme provider.

# Testing

Strategy:  
- Unit tests with Jest.  
- Integration tests with React Testing Library.  
- End-to-end tests with Cypress.

Code Coverage:  
- Enforced via Jest coverage thresholds.

# Screenshots or Demo

- [Link to demo site]  
- [Add sample screenshots of homepage, trends visualization, and dashboard].

# Known Issues

- Limited API rate limits on free tiers.  
- Some sources may provide inconsistent metadata.

# Future Enhancements

- AI-powered news summarization.  
- Sentiment/time-series visualizations.  
- Social sharing and community discussions.  
- Multilingual support.