Frontend Development with React.js

# Project Documentation

## Introduction

Project Title: Insightstream: Navigate the News Landscape

Team Members

**Team ID :** NM2025TMID32578

**Team Leader :** SHEREYA SRI V

**Team member :** ANBUKARASI S

**Team member :** SOUNDARYA K A

**Team member :** SHARMILA R

**Team member :** JAYAPRIYA V

Project Overview

Purpose: Insightstream is a React-based web application designed to provide users with a smart, interactive platform to explore, analyze, and consume news. The goal is to help readers easily navigate the news landscape, filter content by categories, and stay updated with personalized feeds.

Features:

- Real-time news updates from multiple sources  
- Category-wise browsing (Politics, Sports, Technology, Business, Entertainment, etc.)  
- Smart search and filtering options  
- Responsive UI for mobile and desktop users  
- Bookmark and save articles for later reading  
- Dark/Light theme toggle for better readability

## Architecture

Component Structure:  
- Header/Navbar – Navigation links, search bar, and theme toggle  
- News Feed Component – Displays latest articles with images and headlines  
- Category Component – Filters news by category  
- Article Details Component – Shows detailed view of a selected article  
- Bookmark Component – Stores saved articles  
- Footer – App credits, contact info, and external links

State Management: Context API is used for global state handling such as theme preference, bookmarks, and user settings.

Routing: Implemented using React Router v6.  
- `/` → Home/Top Stories  
- `/category/:name` → Category-wise articles  
- `/article/:id` → Detailed article view  
- `/bookmarks` → Saved articles

## Setup Instructions

Prerequisites: Node.js (>= 16.x), npm or yarn

Installation:

1. Clone the repository:  
 git clone https://github.com/your-repo/insightstream.git  
2. Navigate to project folder:  
 cd insightstream  
3. Install dependencies:  
 npm install  
4. Create .env file and add API keys (e.g., News API).

## Folder Structure

/client  
 ├── /src  
 │ ├── /components  
 │ │ ├── Navbar.js  
 │ │ ├── NewsCard.js  
 │ │ ├── Footer.js  
 │ ├── /pages  
 │ │ ├── Home.js  
 │ │ ├── Category.js  
 │ │ ├── Article.js  
 │ │ ├── Bookmarks.js  
 │ ├── /assets  
 │ ├── /utils  
 │ │ ├── fetchNews.js  
 │ │ ├── useLocalStorage.js  
 │ ├── App.js  
 │ ├── index.js

## Running the Application

Start the frontend development server:  
 npm start  
This will run the app at http://localhost:3000

## Component Documentation

- Navbar: Handles navigation, search, and theme toggle  
- NewsCard: Reusable card component displaying article thumbnail, title, and summary  
- Category Page: Fetches and displays articles by category  
- Bookmarks Page: Manages saved articles using local storage

## State Management

Global State (Context API):  
- Theme (light/dark)  
- Bookmarks list  
- User preferences

Local State (useState):  
- Search input  
- API responses within components

## User Interface

• Clean, modern interface designed with React + Tailwind CSS  
• Mobile-first design with responsive layouts  
• Example UI features:  
 - Home Page – Trending articles in grid layout  
 - Article Page – Large image, full content, related articles  
 - Bookmark Page – List of saved news with remove option

## Styling

• CSS Framework: Tailwind CSS  
• Custom Hooks: For theme toggling and API fetching  
• Design System: Consistent typography and color scheme for readability

## Testing

Strategy:  
- Unit tests with Jest + React Testing Library  
- Integration testing for API calls  
Coverage:  
- Ensures critical components (NewsCard, CategoryPage, Navbar) are tested

## Screenshots or Demo

[Insert screenshots of UI pages here once implemented]  
Live demo link can be hosted on Vercel/Netlify

## Known Issues

• API rate limits may restrict the number of articles fetched  
• Some sources may not provide full content

## Future Enhancements

• AI-powered article summarization  
• Sentiment analysis of news content  
• User authentication and profile-based recommendations  
• Multi-language support