Frontend Development with React.js Project Documentation Format

Insight Stream – Navigate the News Landscape

Project Documentation

1. Introduction

Project Title: Insight Stream – Navigate the News Landscape

Team Members:

Team Leader Name: Manjari T & manjarithanigaimalai63@gmail.com

Team Member Name: Megala M & megalaselvi123@gmail.com

Team Member Name: Menaka K& menakabalaji143@gmail.com

Team Member Name: Prathisha S & prathishasanmugam@gmail.com

2. Project Overview

Purpose: To provide users with a streamlined platform for exploring, filtering, and engaging with the latest news across multiple categories.

Features:

Real-time news updates

Category-wise filtering (e.g., politics, sports, technology, entertainment)

Personalized recommendations

Search and bookmarking functionality

Responsive UI for desktop and mobile

3. Architecture

Component Structure:

Navbar (navigation between categories)

NewsFeed (main article list)

ArticleCard (individual news preview)

SearchBar (for keyword search)

BookmarkList (saved articles)

State Management: Using Context API (or Redux if project scales).

Routing: React Router for navigation between home, category pages, search results, and bookmarks.

4. Setup Instructions

Prerequisites: Node.js, npm/yarn

Installation:

git clone https://github.com/Manjari147/Insighthttps://github.com/Manjari147/Insight

cd insight-stream

npm install

npm start

5. Folder Structure

Client:

/src/components → Navbar, NewsFeed, ArticleCard, SearchBar, BookmarkList

/src/pages → Home, Category, Search, Bookmarks

/src/assets → Images, icons, logos

Utilities:

API helper for fetching news (e.g., from NewsAPI or custom backend)

Custom hooks (e.g., useFetchNews)

6. Running the Application

To start the app locally:

cd client

npm start

7. Component Documentation

Key Components:

Navbar: Manages navigation

NewsFeed: Displays list of articles

ArticleCard: Renders each article with title, snippet, and image

Reusable Components:

Button, Card, Loader, Modal

8. State Management

Global State: Handles user preferences, category selection, bookmarks.

Local State: Component-specific states like dropdown open/close, search input, etc.

9. User Interface

Screenshots / GIFs (e.g., home page with live headlines, search results page, bookmarks view).

10. Styling

CSS Frameworks/Libraries: Tailwind CSS for styling, shadcn/ui for UI components.

Theming: Light/dark mode support with custom theme toggle.

11. Testing

Unit Testing: React Testing Library for components

Integration Testing: Jest for API + state flows

UI Testing: Cypress for end-to-end testing