# Frontend Development with React.js
# Project Documentation Format

## 1. Introduction

- Project Title: Insight Stream: Navigate the News Landscape
- Team Members: Mastani I – Project Lead & API Integration; Selvi G – Developer & Error Handling; Gopika R – Documentation & Voice Over; Deepika G – PPT Presentation; Nasrin Fathima M – Documentation Supporting

## 2. Project Overview

- Purpose: The purpose of this project is to provide users with an efficient way to navigate the news landscape by filtering and organizing news articles.
- Features: Filter news content based on various categories. Organize news articles using content-based features. Simplified and interactive user interface for accessing relevant news.

## 3. Architecture

- Component Structure: The frontend uses a simple architecture where HTML code (initially generated using SmartInt) was integrated into Visual Studio.
- State Management: Basic error handling and API-driven data fetching were implemented.
- Routing: Not applicable (single-page news feed application).

## 4. Setup Instructions

- Prerequisites: Node.js, Visual Studio Code, Rapid API account
- Installation: 1. Clone the repository. 2. Open the project in Visual Studio Code. 3. Configure Rapid API key for fetching news data. 4. Run the project using development server.

## 5. Folder Structure

- Client: index.html (main entry point), assets/ (images, CSS files), scripts/ (JavaScript code for API integration and UI updates).
- Utilities: Helper functions for API requests and error handling.

## 6. Running the Application

- Frontend: Run 'npm start' inside the client directory after setup.

## 7. Component Documentation

- Key Components: News API Integration (fetches articles from Rapid API), News Filter (organizes content based on categories).
- Reusable Components: API call functions, Error handling utilities.

## 8. State Management

- Global State: Managed via API calls and response handling.
- Local State: Maintains filtered categories and displayed articles in the UI.

## 9. User Interface

- UI Features: Simple and clean layout to display categorized news. Search and filter options for better navigation.

## 10. Styling

- CSS Frameworks/Libraries: Custom CSS used for styling.
- Theming: Minimalist design to enhance readability of news content.

## 11. Testing

- Testing Strategy: Manual testing of API integration, error handling, and category filtering.
- Code Coverage: Verified API responses and ensured news content loads dynamically without breaking UI.

## 12. Screenshots or Demo

- Provide screenshots or a demo link showcasing the application’s features and design.

## 13. Known Issues

- Occasional delays in API response depending on network speed.
- Some categories may return fewer results due to API limitations.

## 14. Future Enhancements

- Add user authentication for personalized news feeds.
- Integrate advanced filters (e.g., trending, location-based).
- Improve UI with animations and enhanced styling.
- Implement voice-based news search and reading.