Text-Based-Explanation

🔹 System Design

The project follows a **modular architecture** with a **FastAPI backend** and **Next.js frontend**.

📌 System Flow

1. **User interacts with the Next.js dashboard.**
2. **Data Pre-processing is performed.**
3. **Frontend sends API requests to the FastAPI backend.**
4. **FastAPI processes the request**:

- **Time-series analysis** from `processed_data.json`
- **Network visualization** from `data.jsonl`
- **AI Insights** using **Groq Llama3-8B**
- **Semantic search** using **TF-IDF & Cosine Similarity**
- **Next.js fetches data and displays** insights via **interactive charts & UI**

📂 Codebase Overview

Backend (`backend/`)

File	Description
main.py	Main FastAPI application, handles API routes.
chatbot.py	Implements AI chatbot using **Groq API**.
network_visualization.py	Uses **NetworkX & PyVis** to create network graphs.
requirements.txt	Lists all dependencies for FastAPI.

Core FastAPI API

Key Features:
☑ **Time-Series Analysis** – `processed_data.json` used for tracking trends.
☑ **Subreddit & Score Distribution** – Aggregates subreddit data.
☑ **AI Analyzer** – Uses **Groq API (Llama3-8B)** for structured insights.
☑ **CORS Middleware** – Ensures cross-origin API requests work with Next.js.
☑ **Network Graph** – Generates **author-post relations** using **PyVis**.

Conversational AI Chatbot

☑ Uses **Llama AI (Groq API)** for **context-aware chat responses**
☑ **Memory-enabled chat** – Stores conversation history for better responses

How it Works:

1. **User asks a question** → Next.js sends request to `/chatbot` API
2. **FastAPI finds relevant posts** using **TF-IDF & Cosine Similarity**
3. It **generates a structured AI response** using **Groq.**
4. Response **sent back to Next.js** for display

Network Graph Analysis

☑ **Uses `data.jsonl` to build network relations** between authors & subreddits
☑ Generates **interactive graphs using PyVis & NetworkX**
☑ Returns graph as **HTML file**, displayed in Next.js via `iframe`

Frontend (`frontend/`)

File/Folder	Description
components/	Stores reusable UI components (Charts, Chatbot, Sidebar, AI Analyzer).
app/	Next.js **pages & routes**.
public/screenshots/	**Project images** for README.md.
.env.local	Stores **backend API URLs**.