**AI-Powered Comment Moderation System**

**Objective:**

Build an AI-driven, multilingual comment moderation system capable of identifying and managing toxic content across social platforms like YouTube and social media, particularly for the Times Network.

**Goals:**

* Detect toxic, hateful, and offensive comments in multiple languages (English, Hindi, Hinglish, and others).
* Apply different filtering and scoring techniques (regex, LLM, multilingual models).
* Enable human review of borderline cases.
* Provide explainability for flagged content.

**Key Features:**

* Language detection
* Regex + KB-based profanity filter
* Perspective API & XLM-RoBERTa based toxicity analysis
* LLM-based explainability (Groq with LLaMA-3)
* Dashboard-ready logging system
* Modular pipeline-based architecture

**Folder/File Structure:**

├── KB/

│ ├── english.txt # English bad word list

│ ├── hindi.txt # Hindi bad word list

│ └── hinglish.txt # Hinglish bad word list

├── logs/

│ └── moderation.log # All moderation decisions and logs

├── .env # API keys and environment variables

├── api.py # FastAPI backend for moderation API

├── bert\_based\_model\_predict.py # XLM-RoBERTa or similar model for multilingual detection

├── db\_connection.py # Placeholder for DB support (future)

├── explainable\_ai.py # Groq LLM explainability for decisions

├── lang\_detect.py # Language detection module

├── llm\_based\_analysis.py # LLM-based judgment for multilingual content

├── logger.py # Custom logger module

├── main.py # Streamlit frontend or CLI

├── model.py # Wrapper for model utilities

├── perspective\_ai.py # Perspective API integration

├── pipeline.py # Full moderation pipeline

├── profanity\_filter.py # Regex-based KB profanity matcher

├── requirements.txt # Python dependencies

├── README.md # Project intro and instructions

**External APIs/Tools:**

* **Groq (LLaMA-3.3)**: Explainability
* **Perspective API**: Toxicity scoring
* **XLM-RoBERTa-textdetox--**: Multilingual classification

**Dependencies:**

List in requirements.txt will include:

* transformers
* groq
* fastapi
* streamlit
* langdetect
* requests
* python-dotenv