cognizant

Nurture Partner Network Program

AI-Powered Metadata Tagging from Transcripts

Team Name : CineGraph AI 🎬

Hackathon Idea Template

Ideation Submission - Team Information Template

Team Name	College Name	Team Member Name	Mail ID	Department	Year of passing
CineGraph AI	Siksha ‘O’ Anusandhan (ITER)	Anurag Panda	apstudies2249@gmail.com	CSE-AIML	2026
CineGraph AI	Siksha ‘O’ Anusandhan (ITER)	Shreyasi Dey	shreyasidey8555@gmail.com	CSE-AIML	2026
CineGraph AI	Siksha ‘O’ Anusandhan (ITER)	Asish Sahoo	asishsahoo832@gmail.com	CSE-AIML	2026
CineGraph AI	Siksha ‘O’ Anusandhan (ITER)	Ayush Majumder	ayush.majumder04@gmail.com	CSE-AIML	2026
CineGraph AI	Siksha ‘O’ Anusandhan (ITER)	Chandrika Das	chandrika.18.2000@gmail.com	CSE-AIML	2026
CineGraph AI	Siksha ‘O’ Anusandhan (ITER)	Sainee Panda	saineepanda24@gmail.com	CSE-AIML	2026
CineGraph AI	Siksha ‘O’ Anusandhan (ITER)	Abha Mahato	abhamahato2004@gmail.com	CSE-DS	2026
CineGraph AI	Siksha ‘O’ Anusandhan (ITER)	Ashit kumar Panigrahi	ashit.ku02@gmail.com	CSE-AIML	2026

<idea description> | Business Plan (1/3)

WHY

Movie and script datasets are highly unstructured, spread across diverse sources such as raw text files, reviews, character dialogues, and CSV **metadata**. This makes it difficult to integrate and query data efficiently, leading to several challenges:

- **Character-level insights** like emotional arcs, relationships, and dialogue-driven sentiment shifts are hidden in long, unprocessed text.
- **Movie-level analytics** such as awards, producers, genres, and critic reviews are scattered and not directly connected to characters or scripts.
- **Cross-data insights**, such as how a character's sentiment arc influences critical reception or how production choices relate to awards, require combining structured and unstructured data, which is not straightforward.
- **Scalability challenges** as datasets grow to thousands of scripts and reviews, making manual analysis infeasible **without automation**.

Currently, analysts and researchers must manually search, read, and parse through massive corpora of dialogues, scripts, and reviews. This process is **time-consuming, error-prone, and unscalable**, especially as the size of movie and transcript datasets continues to grow.

<idea description> | Business Plan (1/3)

- **Data Ingestion & Processing :**
  - Parsed movie metadata, characters, reviews, and awards
  - Cleaned & structured using Pandas, NumPy
  - Applied semantic chunking for dialogues
- **Graph & Vector Databases :**
  - **Neo4j** for relationships (Movies ↔ People ↔ Awards ↔ Companies)
  - **ChromaDB** for embeddings & semantic search
- **ML & AI Modelling :**
  - Metacritic Score Prediction → XGBoost, Random Forest (Regression)
  - Awards Prediction → Logistic Regression (OvR), XGBoost (Multi-label Classification)
  - Binary Classification (Award vs Not) → Logistic Regression, Random Forest
  - NLP features**: TF-IDF + BERT embeddings**
- **Conversational Agent :**
  - **LLM** (ChatGroq + LangChain) for natural language queries
  - Integrated with Neo4j tool + Vector DB tool
  - Deployed as a Flask backend + React (Vite) frontend
- **User Experience :**
  - Interactive chat interface with thumbs up/down feedback
  - Hero dashboard summarizing capabilities
  - Supports **real-time Q&A over movies, characters**, and **reviews**

How

<idea description> | Business Plan (1/3)

Our solution tackles the challenge of **unstructured movie and script data** by combining graph databases, vector search, and machine learning into one unified system.

We first process raw data such as metadata, reviews, dialogues, and awards, and enrich it with semantic chunking, embeddings, and NLP features. This data is stored in Neo4j to capture relationships **(movies ↔ characters ↔ people ↔ awards)** and in ChromaDB to enable semantic similarity search across large dialogue corpora.

This dual storage approach ensures that users can explore structured graph relationships while also retrieving unstructured insights from character dialogues and reviews.On top of this, we built predictive machine learning models for key tasks: **Metacritic score prediction** (regression), **award prediction** (multi-label classification), and **binary classification** (award-winning vs not) using XGBoost, Random Forest, and Logistic Regression. These models use both structured metadata and unstructured embeddings (**TF-IDF, BERT**).

Finally, we integrated a **conversational agent** powered by **LangChain** and **Groq LLMs**. This agent dynamically uses Neo4j and ChromaDB tools to answer natural language questions, deployed via a **Flask backend** and a **React/Vite frontend** with a chat interface. Users can now interact with the system conversationally, query insights, explore character sentiment arcs, and even provide feedback to refine responses.

WHAT

<Solution benefits and challenges> | Business Plan (2/3)

Benefits & Architecture

Benefits:

Our project delivers multiple benefits by transforming unstructured dialogues, scripts, reviews, and metadata into **structured insights** that enable precise querying. By combining graph and vector search, it powers smarter recommendations across movies, characters, and genres. **Sentiment arcs and character analysis** provide a deeper understanding of storytelling and emotional progression, while Neo4j enables rich exploration through **cross-linking** of entities such as movies, genres, awards, people, and companies. Additionally, the **integration** of **LLMs** allows conversational interaction with the dataset, enabling subject matter experts and analysts to query naturally and gain actionable insights.

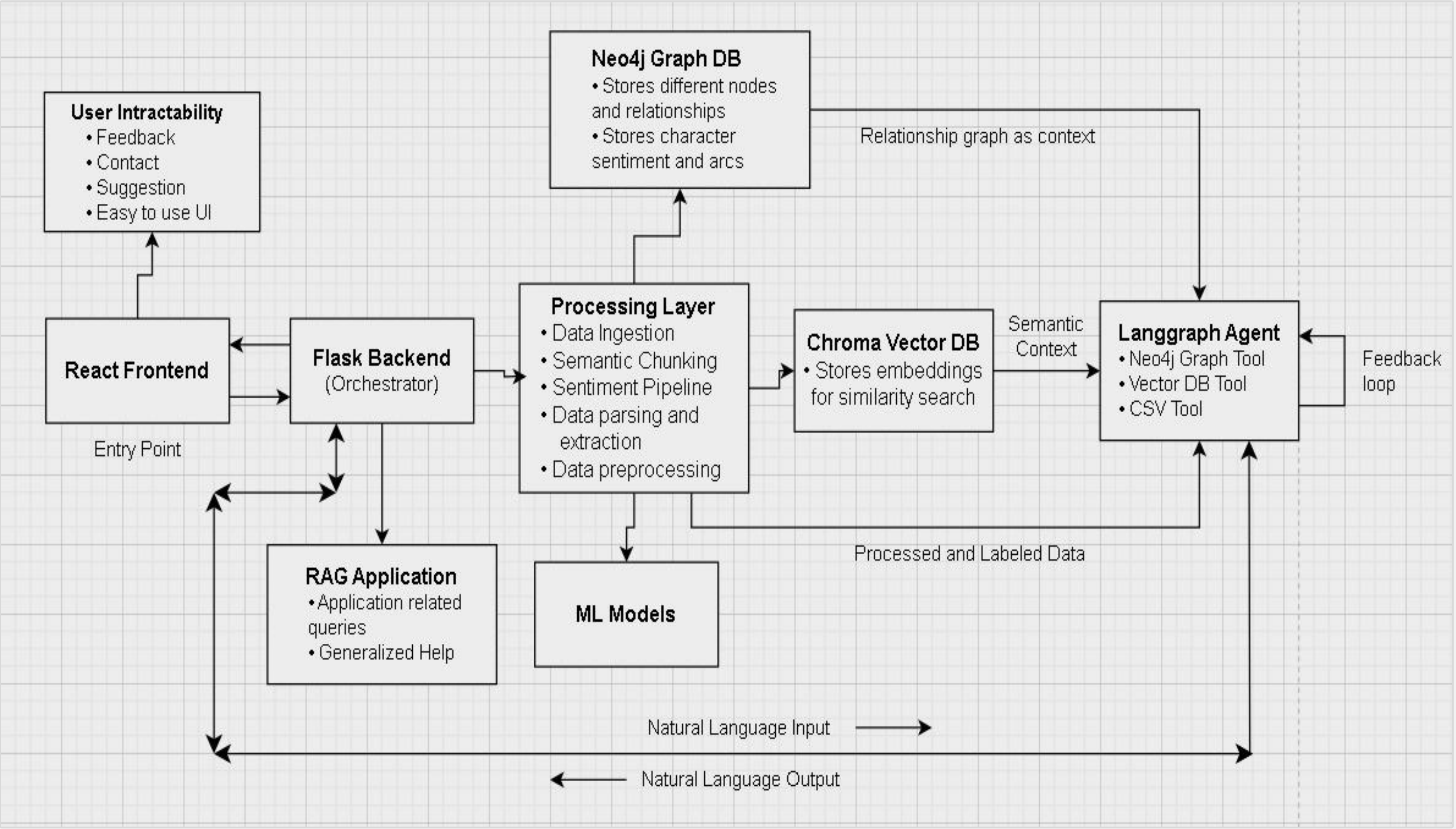
Architecture:

Movie scripts, reviews and character dialogues are large, heterogeneous and mostly unstructured, hiding useful signals like **character emotional arcs**, **relationships** and **cross-source correlations** (e.g., script sentiment vs critic scores). We ingest CSV metadata and raw texts, clean and semantically chunk dialogues, generate **dense embeddings** (Sentence-Transformers) and store them in a **vector store** (Chroma) for fast semantic retrieval. At the same time we model structured entities and relationships (movies, people, awards, genres, character arcs) in Neo4j so complex graph queries are possible and ingestion is idempotent.

We trained **predictive models** (Metacritic regression with **XGBoost/RF**; awards multi-label classification with **XGBoost/LogReg** + **One-vs-Rest;**binary award classifiers) and a transformer-based sentiment pipeline whose outputs (scores, award probabilities, sentiment arcs) are written back into **Neo4j** for use by the system. A **LangGraph**-powered agent (ChatGroq LLM) orchestrates tools at query time—Neo4jGraphQA for structured answers and Chroma for grounding—exposed via a **Flask API** and **React chat UI** with a **feedback** loop. **Scalability** is addressed with semantic chunking, batched GPU embeddings (CUDA) and parallel processing; data validation and MERGE-based ingestion maintain graph consistency while feedback drives iterative model improvements.

<Solution benefits and challenges> | Business Plan (2/3)

Architecture Diagram



<Solution benefits and challenges> | Business Plan (2/3)

Handling huge text datasets – Our corpus consisted of thousands of scripts and character dialogues. To manage this scale, we applied **semantic chunking** to split long dialogues into meaningful segments and processed them using **parallel pipelines**. This ensured **balanced workloads** across CPU cores and prevented bottlenecks.

Slow embedding generation & queries – Embedding large volumes of text with transformer models is computationally expensive. We solved this by leveraging **GPU acceleration** (CUDA, RTX 3060), which drastically **reduced embedding time** and enabled **real-time querying** in ChromaDB for conversational use cases.

Maintaining graph consistency – With data coming from multiple sources (metadata CSVs, reviews, characters, awards), there was a high chance of duplicates and broken links. We tackled this by using **MERGE clauses** in Neo4j Cypher queries, which automatically created or linked nodes without overwriting existing properties, ensuring a **clean** and **connected knowledge graph**.

Ensuring data quality – Raw text and metadata often contained inconsistent formats (countries, taglines, age restrictions, awards). We designed **custom parsers** to standardize these fields and stored them as **structured JSON objects** within nodes, making the graph queries more reliable and **preserving** original data context.

Challenges and innovations

<Solution benefits and challenges> | Business Plan (2/3)

Our solution addresses the challenge of working with **unstructured movie data**—scripts, reviews, and dialogues—by transforming it into a **structured, queryable knowledge graph** enriched with **semantic embeddings**. This allows us to capture complex insights such as character sentiment arcs, evolving relationships, award predictions, and thematic patterns across thousands of films.

At the core, we integrate **Neo4j** (graph DB) for structured relationships, **ChromaDB** (vector store) for semantic similarity, and **LLMs** (via LangChain + Groq) for conversational reasoning. This **hybrid architecture** bridges structured and unstructured data seamlessly, enabling users to ask natural-language questions and receive grounded, explainable, and context-rich answers. Unlike standalone AI tools, our system merges **NLP pipelines, ML models, and graph analytics** into one cohesive ecosystem, powered by **GPU acceleration** and **parallel processing** for speed and scalability.

The uniqueness lies in our ability to connect multiple layers of data: character dialogues inform sentiment arcs, reviews provide **external validation** and **metadata links** to genres, producers, and awards. This multi-angle analysis enables studios, analysts, and researchers to unlock actionable insights, from predicting success and audience reception to understanding creative trends. By automating **metadata extraction** and enabling conversational exploration, we save time, improve decision-making, and open new **opportunities** for entertainment analytics and storytelling innovation.

Value proposition

Financials & Timelines | Business Plan (3/3)

Investments

Building this solution requires **GPU-enabled** compute for embedding generation and sentiment modelling, plus storage layers for Neo4j (**graph DB**) and Chroma (**vector DB**). Development involves Python backend, React frontend, and LLM integration via Groq API.
Human effort is needed for data cleaning, schema design, and pipeline engineering, but much is automated with parallel processing. **Infrastructure** can be **cloud-hosted** or **hybrid** for scalability.
Estimated cost: moderate cloud spend *(~$2k–$5k/month* for GPUs + DB hosting*)* plus initial engineering effort from a **small ML/Dev team**.

Returns

Delivers structured insights from messy text: character arcs, sentiment flows, movie–award linkages, and recommendations — insights impossible to derive manually at scale.
Boosts analyst productivity 5–10x by replacing manual parsing with natural-language queries over graph + vector stores. **SMEs** get instant evidence-backed answers.
If unsolved, orgs face costly manual analysis, missed insights in terabytes of data, and inability to leverage this data for creative or business intelligence.

Timelines

Initial ingestion + embedding pipelines: 2–3 weeks. Graph + vector store integration: 3–4 weeks.
Conversational agent & UI: 2–3 weeks.
Models for score and award prediction can be trained in parallel within 2 weeks. **End-to-end MVP** achievable in ~8 weeks with a small team.
Benefits (querying insights, sentiment arcs, recommendations) begin immediately **post-MVP**, with accuracy and performance improving over 2–3 months as feedback is incorporated.

THANK YOU