**Project Documentation – Movie Q&A Assistant**

**1. Abstract**

This project implements an **AI-powered Question Answering (Q&A) Assistant** for movie data.  
It combines **LangChain**, **FAISS**, and **HuggingFace Embeddings** to create a vector-based search engine, integrated with an interactive **Streamlit UI**.  
Users can ask natural language questions about movies, and the system retrieves relevant metadata from a CSV dataset or embedded documents.

**2. Objectives**

* To enable natural language search across movie metadata.
* To provide both **structured queries** (year, runtime, director, cast) and **unstructured queries** (plots, reviews).
* To ensure accuracy by validating against embeddings and raw metadata.
* To create an intuitive user interface for end users.

**3. System Architecture**

**Components:**

1. **Dataset** – Movies stored in CSV format (movies.csv, embedded\_movies.csv, etc.).
2. **Preprocessing** – Documents converted into embeddings using HuggingFace models.
3. **Vector Store** – FAISS used for similarity-based retrieval.
4. **Retriever** – LangChain retriever (similarity search with score threshold).
5. **LLM/QnA Layer** – Uses embeddings + logic rules to return structured answers.
6. **Streamlit UI** – Interactive interface for user queries.

*(You can insert a block diagram here if needed)*

**4. Features**

* Natural language Q&A for movie data.
* Supports structured queries:
  + Movies released in a given year.
  + Movies with runtime less than X minutes.
  + Group movies by year and rating.
  + Count how many movies are embedded.
* Supports unstructured queries:
  + Director, cast, writers, plot, full details.
  + Awards, ratings, reviews.
* Handles “Not Found” gracefully (fallback to metadata check).

**5. Tools & Technologies**

* **Programming Language:** Python 3.9+
* **Libraries:**
  + LangChain
  + FAISS
  + HuggingFace Embeddings
  + Streamlit
  + Pandas
* **Environment:** Local / Cloud (Streamlit Cloud, HuggingFace Spaces, etc.)

**6. Dataset**

**Example Record**

movie\_id,document

573a1390f29313caabcd42e8,"EmbeddedStatus: Embedded Movie

Title: The Great Train Robbery

Year: 1903

Genres: ['Short', 'Western']

Languages: ['English']

Countries: ['USA']

Rated: TV-G

Runtime: 11.0

Cast: ['A.C. Abadie', 'Gilbert M. Anderson', 'George Barnes', 'Justus D. Barnes']

Directors: ['Edwin S. Porter']

Writers: None

Plot: A group of bandits stage a brazen train hold-up, only to find a determined posse hot on their heels.

IMDb Rating: 7.4

Metacritic: 0.0

Awards: 1 win"

**7. Installation & Setup**

1. **Clone Repository**
2. git clone https://github.com/Sivabarani/mflix-movie-chatbot.git
3. cd mflix-movie-chatbot
4. **Create Virtual Environment**
5. python -m venv env
6. env\Scripts\activate # Windows
7. source env/bin/activate # Linux/Mac
8. **Install Dependencies**
9. pip install -r requirements.txt
10. **Run Application**
11. streamlit run app.py

**8. Usage**

Users can ask questions such as:

* *“Which year was Titanic released?”*
* *“List movies directed by Christopher Nolan”*
* *“How many movies are embedded?”*
* *“List movies released before 2000 with runtime less than 120 minutes”*
* *“Explain full details of Inception”*

**9. Sample Queries & Answers**

| **Query** | **Response** |
| --- | --- |
| “Which year was *American Gun* released?” | 2005 |
| “List movies released in 1995” | [Movie1, Movie2, …] |

**10. Conclusion**

This project successfully demonstrates the integration of **vector-based retrieval** and **structured logic handling** for movie Q&A. It provides a robust foundation for extending to more complex datasets or domains beyond movies.