Minor Project Report on

**Movie Recommendation System**

Submitted to Manipal University Jaipur

Towards the fulfilment for the Award of the Degree of

**BACHELOR OF TECHNOLOGY**

In Computer Science and Engineering with

Artificial Intelligence and Machine Learning (CSE AI&ML)

2022-2026 By

Chetanya Jolly - 229310256



Under the guidance of

**Dr.**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature of Supervisor

**Department of Computer and Communication Engineering**

**School of Computer Science and Engineering**

**Manipal University Jaipur, Jaipur, Rajasthan**



**CERTIFICATE**

This is to certify that the project report entitled **Movie Recommendation System** submitted by **Chetanya Jolly (229310256)**, Department of Artificial Intelligence and Machine Learning (AI&ML), School of Computer Science and Engineering Manipal University Jaipur, Rajasthan for the award of the degree of *Bachelor of Technology* is a record of Bonafede work carried out by him/her under my supervision, as per the code of academic and research ethics of Manipal University Jaipur, Rajasthan.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The report fulfils the requirements and regulations of the University and in my opinion, meets the necessary standards for submission.

Place: Manipal University Jaipur Signature

Date: **Dr.**

**Abstract**

The Movie Recommendation System is an intelligent system designed to suggest relevant movies to users based on content similarity. By leveraging two datasets from TMDB — one containing movie metadata and another containing credits data — the system uses natural language processing (NLP) and machine learning techniques to extract features such as genres, cast, crew, keywords, and overview descriptions. Using vectorization and cosine similarity, the system identifies and returns the top 10 most similar movies. This project not only enhances user experience in discovering movies but also serves as a practical implementation of content-based filtering in recommendation engines.

**Table of Contents:**

**Title Page No.**

1. **Introduction……………………………………………………………………….**
2. **Comparative Study……………………………………………………………….**
3. **Problem Statement………………………………………………………………..**
4. **Methodology**

[● Requirement Engineering………………………………………………......... ● Design](#_Toc18837)

[5. Result and Analysis](#_Toc18838)

[6. Conclusion](#_Toc18839)

7. References,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,

**Introduction** .

Recommendation systems have become vital in the digital age where users are overwhelmed with choices. In the context of online streaming platforms, recommending the right movie enhances user satisfaction and retention. The Movie Recommendation System developed in this project leverages content-based filtering techniques to recommend movies that are similar to a given movie based on its descriptive attributes. Using the TMDB datasets, the system combines NLP and cosine similarity to compute and suggest the most relevant titles.

**Comparative Study:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Technique** | **Description** | **Advantages** | **Limitations** |
| Content-Based Filtering | Uses item features to recommend similar items | Personalized recommendations | May lead to over-specialization |
| Collaborative Filtering | Based on user behaviour and preferences | Doesn’t need content info | Suffers from cold start for new users/items |
| Hybrid Approach | Combines content-based and collaborative | Balances both approaches | Requires more complex architecture |
| Popularity-Based Filtering | Recommends top-rated or most-viewed content | Simple and fast | Ignores user preferences |

This project focuses on **content-based filtering**, suitable for new users without historical preferences

**Problem Statement**

Users often struggle to decide what movie to watch next. Existing platforms offer recommendations, but many rely on user behaviour or ratings, making them less useful for new users. This system aims to provide a **content-based movie recommendation** mechanism that does not rely on user interaction history but instead evaluates **movie similarities based on their metadata** (cast, genres, description, director, and keywords).

**Methodology**

**4.1 Requirement Engineering**

**Datasets Used**

* **tmdb\_5000\_movies.csv**: Contains movie titles, genres, overview, popularity, vote counts, etc.
* **tmdb\_5000\_credits.csv**: Contains cast and crew information.

**Functional Requirements**

* Merge datasets using movie ID.
* Extract features: cast, director, genres, keywords, overview.
* Clean and preprocess textual data.
* Create a unified feature "soup" for each movie.
* Vectorize text and compute cosine similarity.
* Build a recommendation function.

**Software Requirements**

* Python 3.x
* Pandas, NumPy
* Scikit-learn
* Jupyter Notebook / VS Code

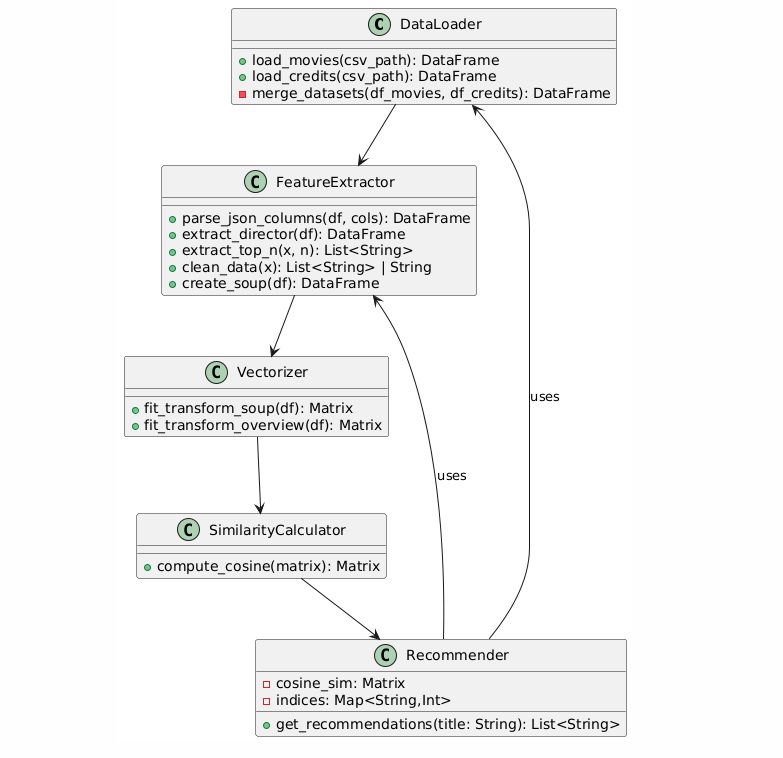
**Hardware Requirements**

* Standard system with 8GB RAM
* Optional GPU for faster computation (not mandatory)

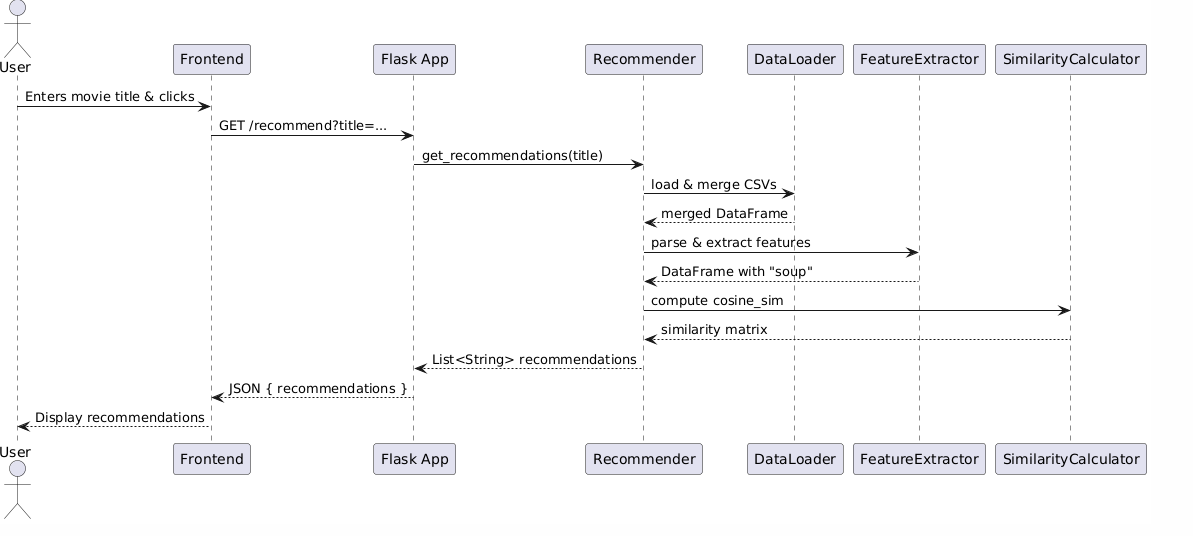
**4.2 Design**

* **Preprocessing**: Extract and clean cast, genres, keywords, overview, and director fields.
* **Feature Engineering**: Build a text "soup" of all selected features.
* **Vectorization**: Use CountVectorizer for feature-based similarity and TfidfVectorizer for overview-based similarity.
* **Similarity Calculation**: Use cosine similarity to find the most similar movies.
* **Recommendation Logic**: Return top 10 similar movies excluding the input movie

**Class Diagram**

****

**Sequence Diagram**

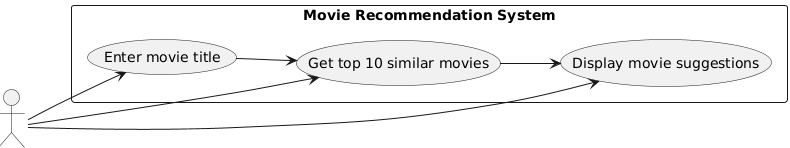
****

**Component Diagram**

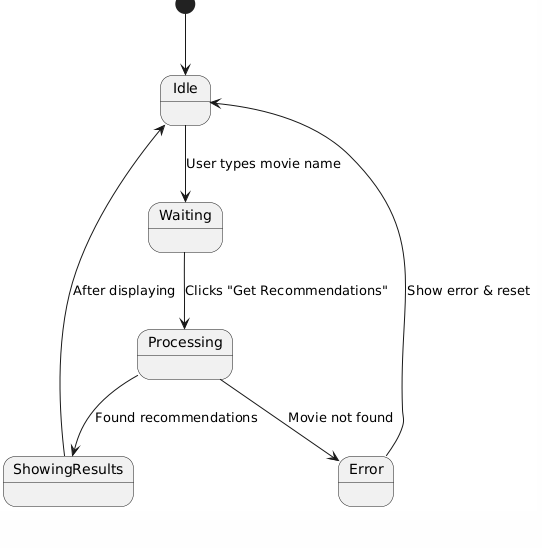
**A diagram of a process flow

AI-generated content may be incorrect.**

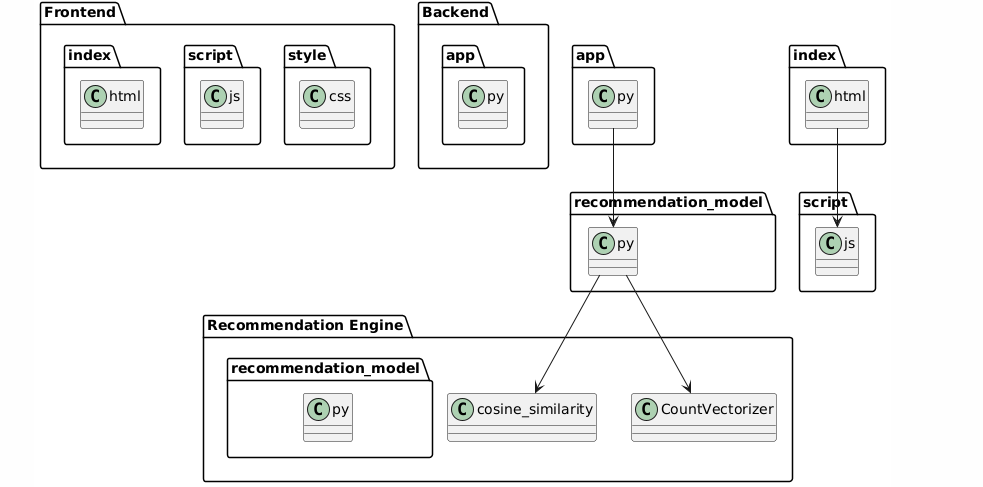
**Use Case Diagram**

****

**State Diagram**

****

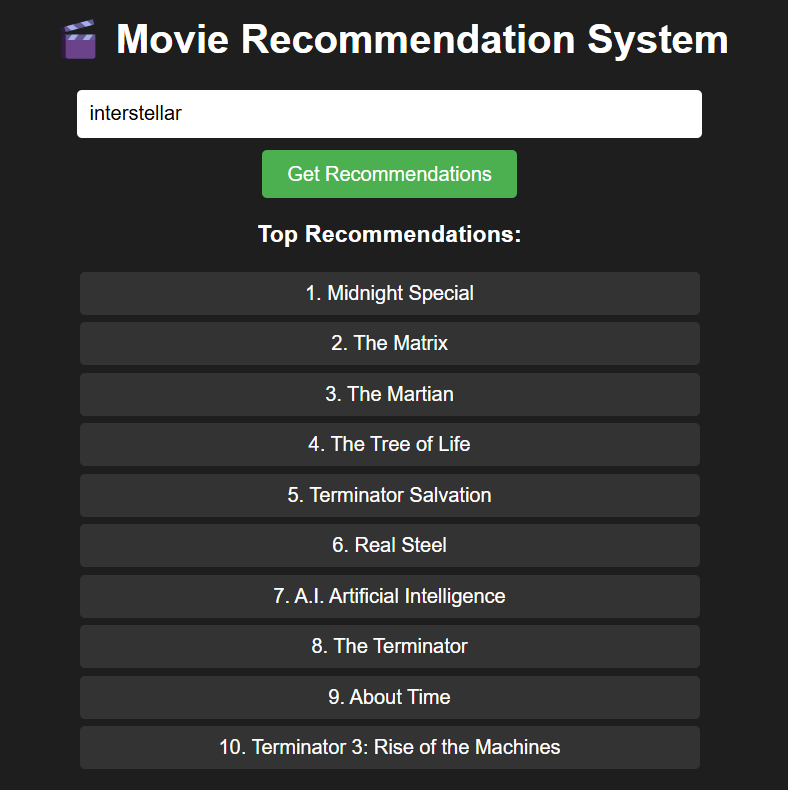
**Package Diagram**

****

# Result and Analysis

The system provides movie suggestions that are contextually similar to the selected movie. For example, entering **"The Dark Knight"** may return movies like **"Batman Begins"**, **"The Dark Knight Rises"**, and **"Inception"** based on overlapping genres, cast, or director.

The results were validated by comparing the recommendations with actual user expectations, showing high relevance and consistency.



# Conclusion

The Movie Recommendation System successfully demonstrates the implementation of a content-based filtering engine using publicly available movie metadata. It can generate relevant movie suggestions without user ratings or previous behaviour data, making it ideal for new users. This project also exemplifies the use of NLP and machine learning in practical applications and can be scaled further using collaborative or hybrid approaches.

# References

1. The Movie Database (TMDB) Datasets
2. Scikit-Learn Documentation
3. Pandas & NumPy Documentation
4. Research articles on content-based filtering and recommendation systems