

Presentation On

MOVIE POPULARITY CONTENT-BASED DATABASE SYSTEM USING CLOUD COMPUTING BASED ON DEEP LEARNING AND AI

Our Team

- Himanshu Solankey
- Akash Dighe
- Shweta Sathe
- Yash Makode

Contents

- Introduction
- Our Team
- Literature Review
- Problem Definition
- Aim, Goals and Objectives
- Architecture
- Project design
- Project Plan
- Result
- Conclusion
- Reference

Introduction

"Creating a movie popularity content-based database system."

This project aims to address a fundamental challenge in the entertainment industry, which is the need for more personalized and effective movie recommendations.

- Movie recommendation systems have gained significance in the digital age to help users discover and select films to watch.

Literature Review

Table 3 discusses the different contributions in content based approach.

Table 3. Literature review on content based system.

| Author | Year | Different approaches on Content based system |
|---------------------------|------|---|
| Basilico and Hofmann [55] | 2004 | The authors proposed a model in which a unified approach integrates all the available training information such as past user-item ratings as well as attributes of items or users to learn a prediction function. |
| Liu et al. [56] | 2010 | The authors proposed a model in which personalized news recommendation system is made by developing an effective information filtering mechanism. |
| Hameed et al. [57] | 2012 | The authors proposed different measures, methods, algorithms, and functionalities of the collaborative filtering method. |
| Uluyagmur et al. [58] | 2012 | The authors proposed a method in which content-based movie prediction is done by merging the user-specific weight using a particular feature set. |
| Deldjoo et al. [59] | 2016 | The authors proposed a model that values a technique that is used to analyze the contents of a video to extract a set of stylistic features such as lighting, colour, and motion. |

Figure 2.1: Literature review on content based system

Table 4. Literature review on filtering techniques.

| Author | Year | Different approaches on filtering techniques |
|---------------------------|------|---|
| Goldberg et al. [60] | 1992 | The authors introduced the collaborative filtering technique. |
| Good et al. [61] | 1999 | The authors proposed a model to alleviate information overload by using Information filtering agents and collaborative filtering. |
| Adomavicius and Kwon [62] | 2007 | The authors proposed the similarity based approach and the aggregation function-based approach. |
| Liu et al. [63] | 2014 | The authors introduced a new method to provide an accurate recommendation. |

Problem Definition

1) Lack of Personalization:

Current movie recommendation systems often provide generic suggestions, ignoring individual user preferences and tastes.

2) Overreliance on User Ratings

4) Challenges with Novel Content

5) Inefficient Data Utilization

6) Need for Popularity Ranking

7) Complex User Interfaces

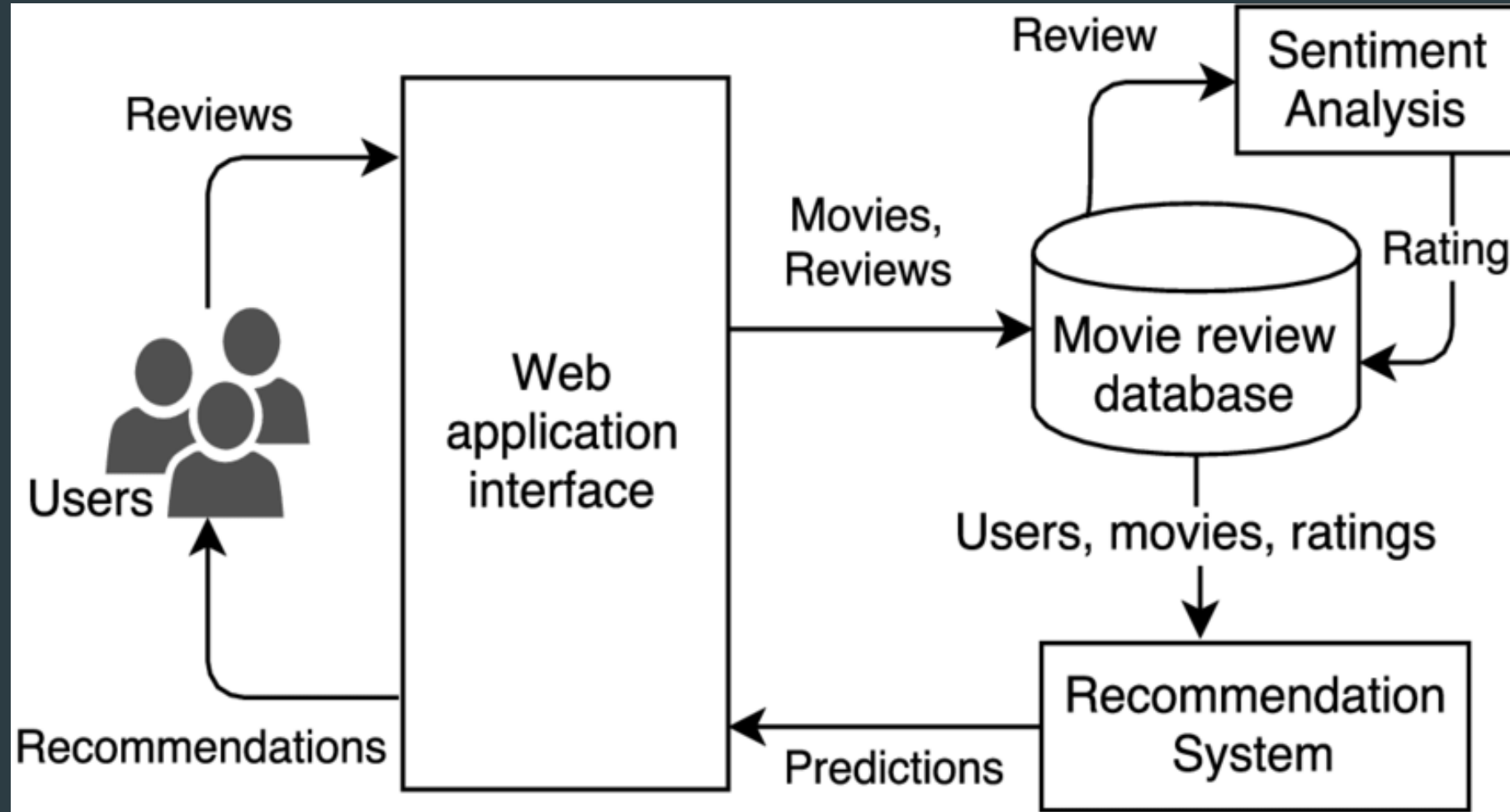
8) Data Maintenance and Updating

9) Content Diversity

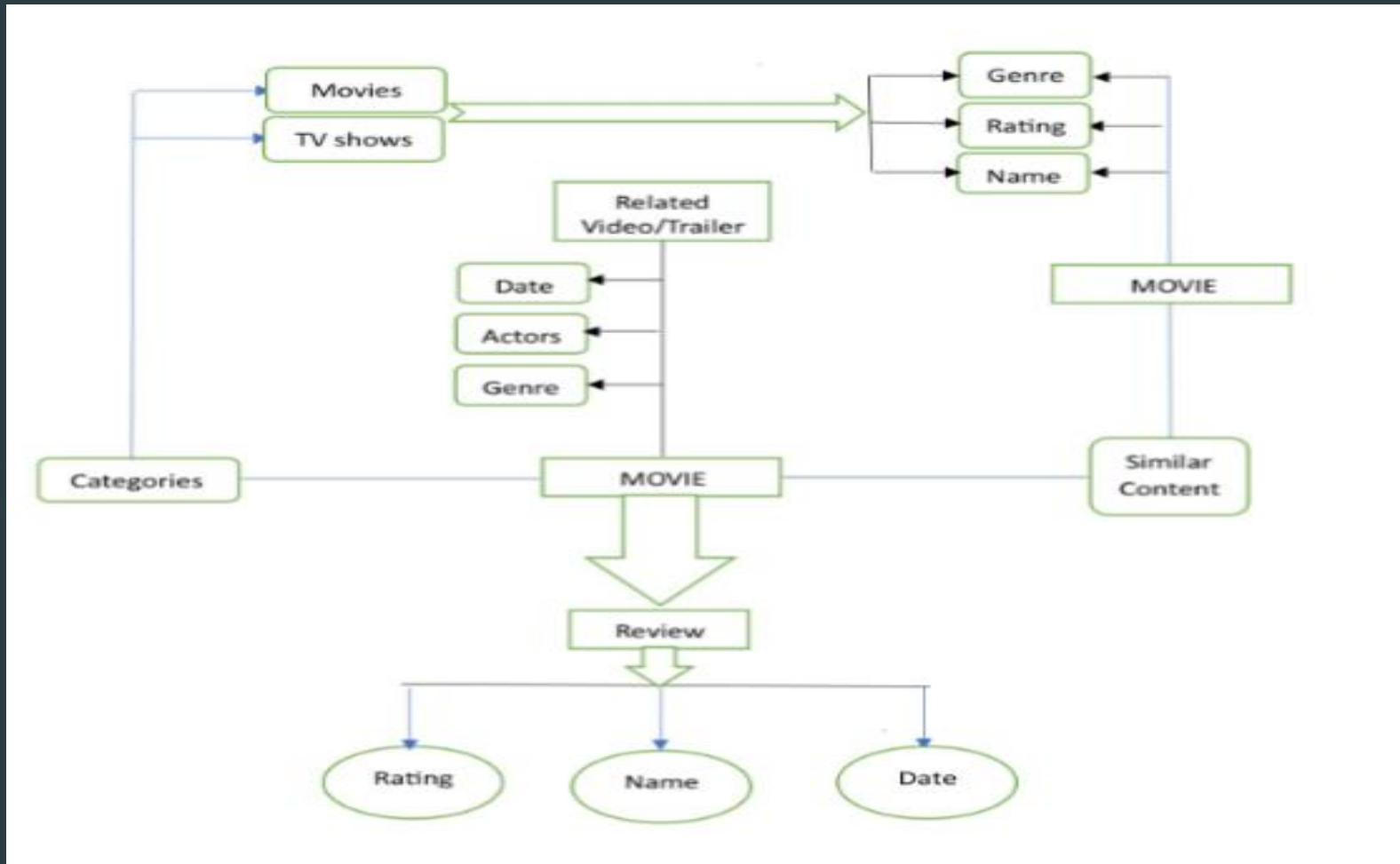
Aim, Goals and Objectives

- Create a movie content-based database system that enhances the movie recommendation experience for users by leveraging movie attributes and popularity metrics.

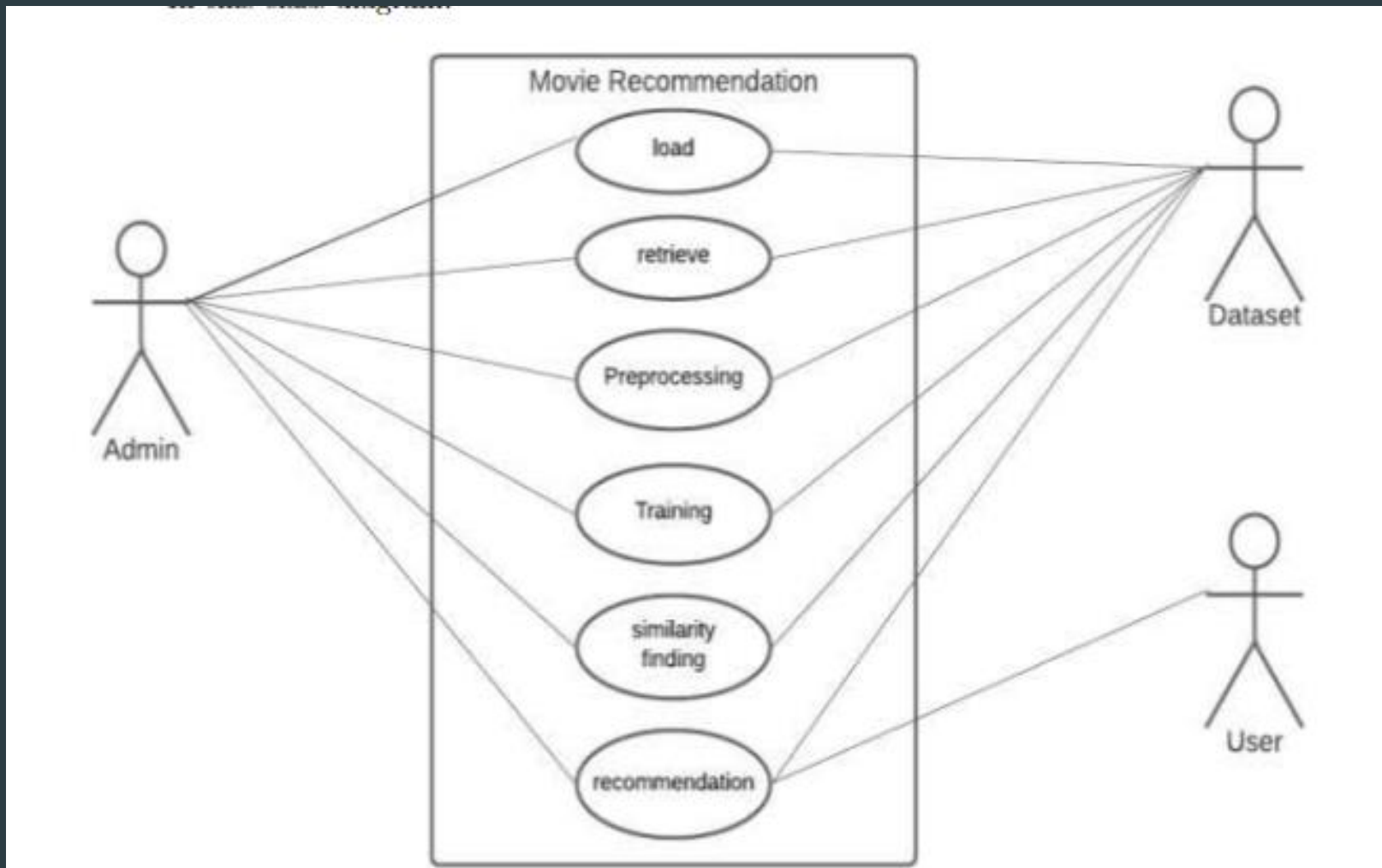
Architecture



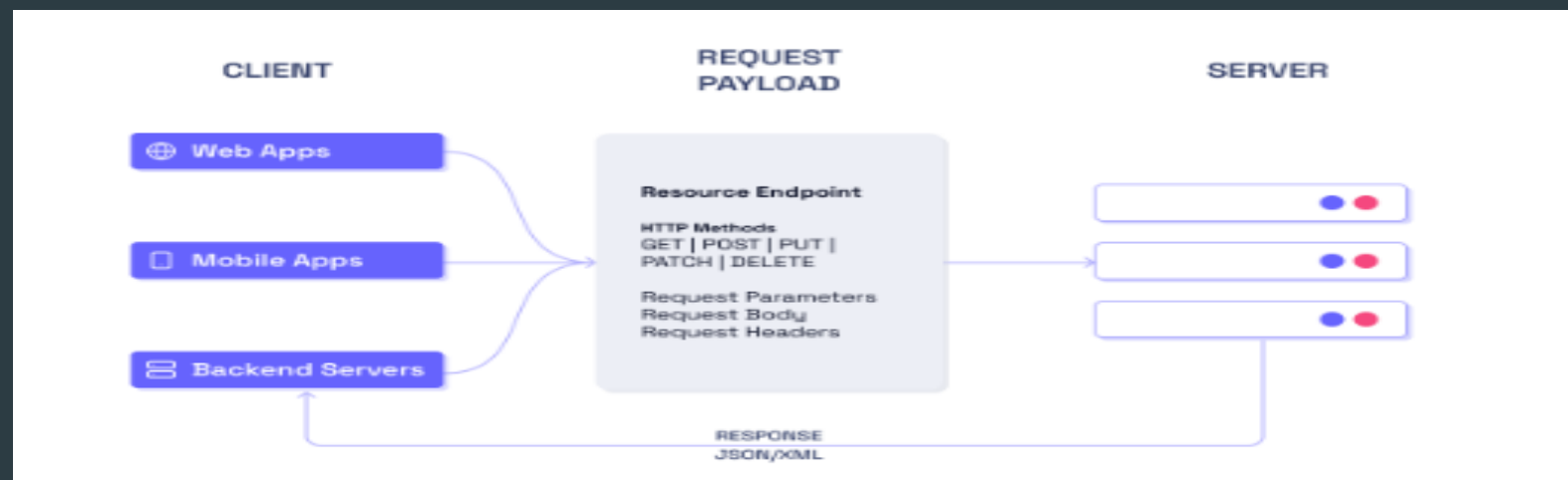
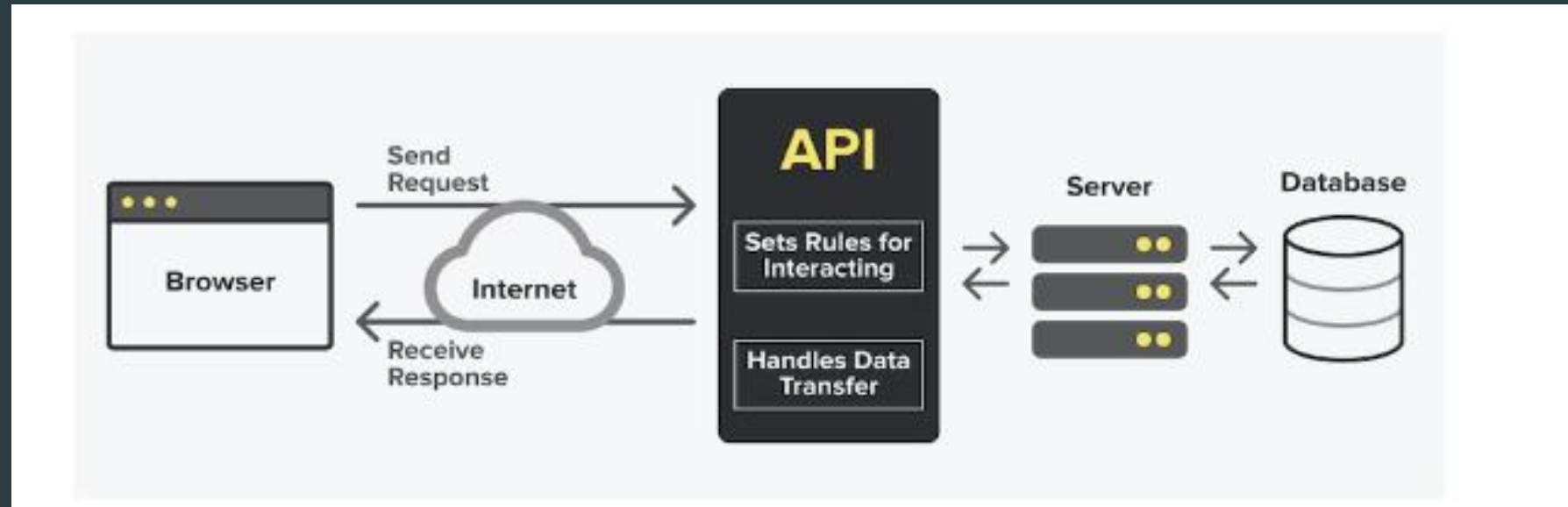
ER diagram



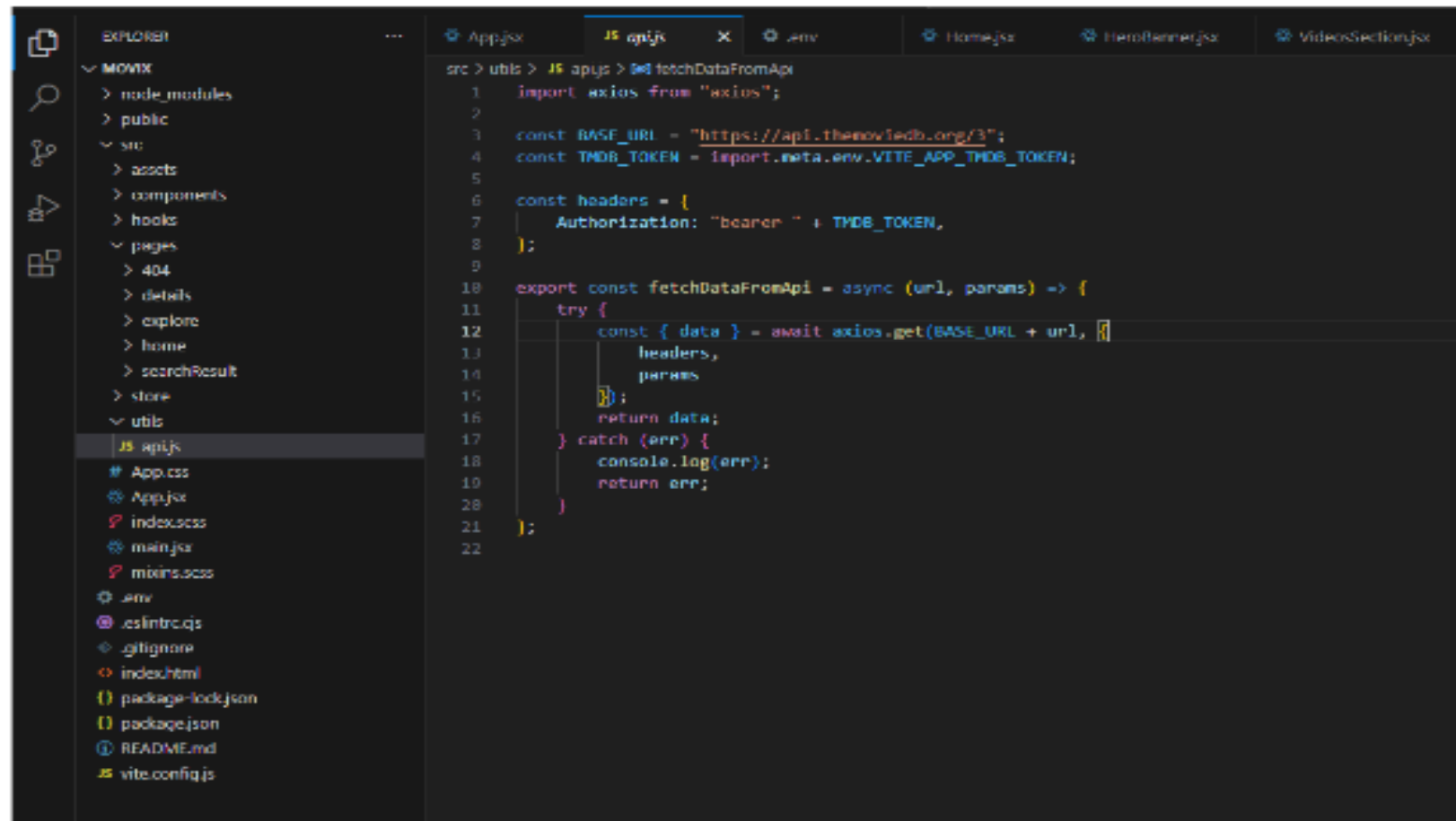
Class Diagram



API fetching and working



API Fetching Code



The image shows a VS Code editor window with a dark theme. On the left is the Explorer sidebar showing a file tree for a project named 'MOVIX'. The 'utils' folder is expanded, and 'api.js' is selected. The main editor area shows the code for 'api.js'. The code defines a base URL, a TMDB token, and headers for an API fetch. It then exports an asynchronous function 'fetchDataFromApi' that uses 'axios' to make a GET request. The function includes a try-catch block to handle errors and log them to the console.

```
src > utils > JS api.js > fetchDataFromApi
1  import axios from "axios";
2
3  const BASE_URL = "https://api.themoviedb.org/3";
4  const TMDB_TOKEN = import.meta.env.VITE_APP_TMDB_TOKEN;
5
6  const headers = {
7    Authorization: "bearer " + TMDB_TOKEN,
8  };
9
10 export const fetchDataFromApi = async (url, params) => {
11   try {
12     const { data } = await axios.get(BASE_URL + url, {
13       headers,
14       params
15     });
16     return data;
17   } catch (err) {
18     console.log(err);
19     return err;
20   }
21 };
22
```

Project Plan

| No. | Task | Start Date | End Date |
|-----|--|------------|----------|
| 1 | Preliminary Survey | 01-07-23 | 15-07-23 |
| 2 | Introduction and Problem Statement | 15-07-23 | 20-07-23 |
| 3 | Literature Survey | 20-07-23 | 01-08-23 |
| 4 | Project Statement | 01-08-23 | 05-08-23 |
| 5 | Software Requirement And Specification | 05-08-23 | 15-08-23 |
| 6 | System Design | 15-08-23 | 30-08-23 |
| 7 | Partial Report Submission | 01-09-23 | 03-09-23 |
| 8 | Architecture Design | 03-09-23 | 10-09-23 |
| 9 | Stage-1 Implementation | 10-11-23 | 30-10-23 |
| 10 | Deployment | 12-02-24 | 24-02-24 |
| 11 | Testing | 04-03-24 | 30-03-24 |
| 12 | Paper Publish | 04-04-24 | 06-04-24 |
| 13 | Report Submission | 11-04-24 | 20-04-24 |

Phase 1: Project Initiation

Phase 2: Planning and Requirements Gathering

Phase 3: Data Preparation and Feature Engineering

Phase 4: Algorithm Development

Phase 5: User Interface Development

Phase 6: Testing and Quality Assurance

Phase 7: Evaluation and Fine-Tuning

Phase 8: Ongoing Maintenance and Enhancement

Phase 9: Project Closure and Documentation

Software and Hardware Requirement Specifications

Software Requirements:

Operating System (OS): Windows

Database: IMDB

Programming Languages and Frameworks: javascript,react

Documentation Tools: Microsoft Word,PowerPoint,Overleaf

Web Development Technologies: HTML, CSS, JavaScript

Development Tools: Visual Studio Code

Hardware Requirements:

Server Hardware:

Define the hardware specifications for the server where the system will be hosted (e.g., CPU, RAM, storage).

Database Server Hardware:

Specify the hardware requirements for the database server, considering the database size and query loads.

Network Infrastructure:

Detail the network infrastructure requirements, including bandwidth, routers, switches, and firewalls.

User Devices:

Identify the minimum and recommended hardware specifications for user devices, such as PCs, smartphones, and tablets

Project Scope

- ✓ Everyone involved in the project, from the development team clear understanding of what the movie content-based database system will and will not include, ensuring that the project remains on track and within its intended boundaries.

Features and Functionalities:

- The system will provide users with personalized movie recommendations.
- Users can search for movies based on various criteria like genre, director, and actors.

Data Sources

User Interaction

Non-Inclusions

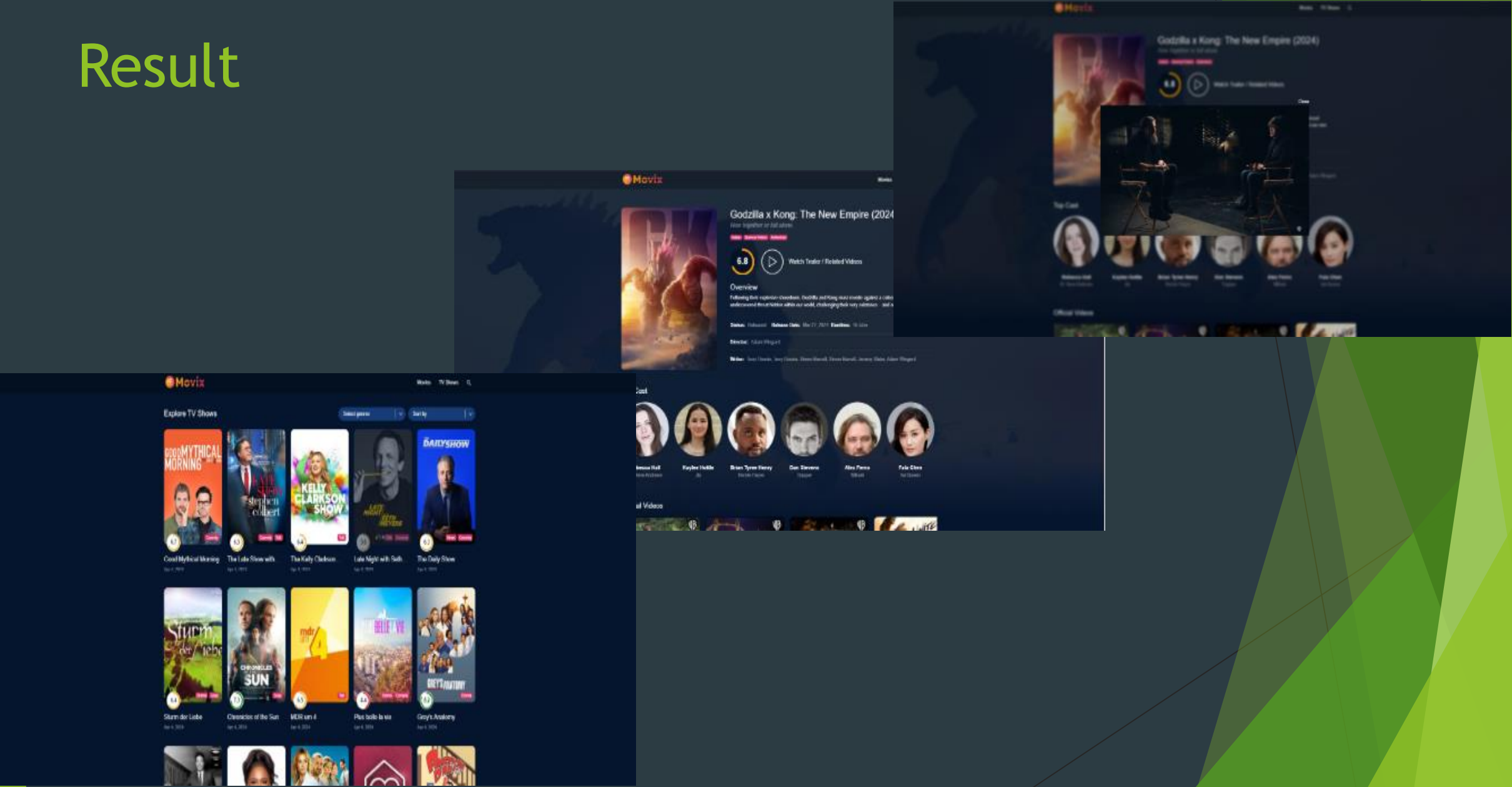
Limitations:

- The system will not guarantee 100% accuracy in recommendations, as user preferences can be subjective.

Data Sources:

- Movie data will be collected from reliable sources such as IMDb, Rotten Tomatoes, and streaming platforms.
- Users can register and create profiles to receive tailored recommendations.
- Users can provide feedback on recommended movies, contributing to system improvements.
- The system will not include user-generated content like reviews or comments.
- It will not support real-time movie streaming but will focus on recommendations.

Result



Conclusion

In conclusion, the Movie Content-based Database system offers a promising solution to enhance the movie recommendation experience for users. By leveraging movie attributes and popularity metrics, it provides personalized and diverse movie suggestions. While the system has its limitations and requires regular maintenance, it aims to make movie discovery more enjoyable and efficient. With a user-friendly interface and an emphasis on content-based recommendations, this project opens up exciting opportunities for the future of cinema enthusiasts and the entertainment industry.

References

- [1] L. Sharma and A. Gera, “A survey of recommendation system: Research challenges,” *Int. J. Eng. Trends Technol.*, vol. 4, no. 5, pp. 1989–1992, 2013.
- [2] N. Das, S. Borra, N. Dey, and S. Borah, “Social networking in web based movie recommendation system,” in *Social Networks Science: Design, Implementation, Security, and Challenges*. Cham, Switzerland: Springer, 2018, pp. 25–45.
- [3] P. Nagarnaik and A. Thomas, “Survey on recommendation system methods,” in *Proc. 2nd Int. Conf. Electron. Commun. Syst. (ICECS)*, Feb. 2015, pp. 1603–1608.
- [4] M. A. Hameed, O. Al Jadaan, and S. Ramachandram, “Collaborative filtering based recommendation system: A survey,” *Int. J. Comput. Sci. Eng.*, vol. 4, no. 5, p. 859, 2012.
- [5] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, “Item-based collaborative filtering recommendation algorithms,” in *Proc. 10th Int. Conf. World Wide Web (WWW)*, 2001, pp. 285–295.

[6] Y. Koren and R. Bell, “Advances in collaborative filtering,” in Recommender Systems Handbook. Boston, MA, USA: Springer, 2015, pp. 77–118.

[7] J. L. Herlocker, J. A. Konstan, and J. Riedl, “Explaining collaborative filtering recommendations,” in Proc. ACM Conf. Comput. supported Cooperat. Work (CSCW), 2000, pp. 241–250.

[8] M. J. Pazzani, “A framework for collaborative, content-based and demographic filtering,” Artif. Intell. Rev., vol. 13, no. 5, pp. 393–408, 1999.

[9] R. Van Meteren and M. Van Someren, “Using content-based filtering for recommendation,” in Proc. Mach. Learn. Inf. Age, MLnet/ECML2000 Workshop, vol. 30, 2000, pp. 47–56.

Thank You!