

A PROJECT REPORT
on
“MOVIE RECOMMENDATION SYSTEM”

Submitted to
KIIT Deemed to be University

In Partial Fulfillment of the Requirement for the Award of

BACHELOR’S DEGREE IN
COMPUTER SCIENCE
ENGINEERING

BY

SYED FARHAN ALI	2005209
AINDRILA ROY	2005358
ARGHYA HAZRA	2005440
SHINJINI BANERJEE	2005964

UNDER THE GUIDANCE OF
PROF. KUNAL ANAND



SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY
BHUBANESWAR, ODISHA - 751024
May 2023

KIIT Deemed to be University

School of Computer Engineering
Bhubaneswar, ODISHA 751024



CERTIFICATE

This is to certify that the project entitled
“MOVIE RECOMMENDATION SYSTEM“

submitted by

SYED FARHAN ALI	2005209
AINDRILA ROY	2005358
ARGHYA HAZRA	2005440
SHINJINI BANERJEE	2005964

is a record of bonafide work carried out by them, in the partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering OR Information Technology) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2022-2023, under our guidance.

Date: 6/ 05 / 23

Prof. Kunal Anand
Project Guide

Acknowledgement

We are profoundly grateful to Professor **Kunal Anand** of **SCHOOL OF COMPUTER SCIENCE ENGINEERING, KIIT UNIVERSITY** for his expert guidance and continuous encouragement throughout to see that this project follows its target since its commencement to its completion. We are pleased to acknowledge the efforts of our team to have developed the movie recommendation system.

Overall, the team's dedication, hard work, and passion have been critical to the success of this project, and we are excited to see how the movie recommendation system will enhance the experience of its users.

SYED FARHAN ALI
AINDRILA ROY
ARGHYA HAZRA
SHINJINI BANERJEE

ABSTRACT

Movie recommendation systems are becoming increasingly popular with the rise of streaming platforms. These systems are designed to suggest movies to users based on their interests, past viewing history, and other factors. In this project, we aim to develop a movie recommendation system that can accurately suggest movies to users based on their preferences. The system will use machine learning algorithms to analyze user behavior and movie data to make personalized recommendations. The goal is to build a system that can suggest movies that users will enjoy and increase user engagement with the platform.

To build the recommendation system, we will use a dataset containing movie metadata such as genre, director, cast, and plot summary, as well as user data such as viewing history, ratings, and reviews. We will clean and preprocess the data, perform exploratory data analysis, and train and evaluate different recommendation algorithms. The system will be deployed using a web application that allows users to input their preferences and receive personalized movie recommendations. The project aims to improve user satisfaction and increase the likelihood of users discovering new movies that they enjoy.

Keywords: Machine Learning, Python, Personalization, Feature Extraction, Cosine Similarity

Contents

1	Introduction	1
2	Basic Concepts	2
2.1	Machine Learning Algorithms	2
2.2	Python tools	2
2.3	NodeJs Tools	3
2.4	Datasets	3
3	Requirement Specifications	4
3.1	Functional Requirements	4
3.2	Non Functional Requirements	4
3.3	User Interface Requirements	5
3.4	Performance Requirements	5
3.5	Design Constraints	5
3.6	Implementation Requirements	5
4.	Design Approach	6
4.1	Use Case Diagram	6
4.2	Activity Diagram	7
5	Implementation	8
5.1	Methodology	8
5.2	Working	10
5.3	Testing	11
5.4	Result Analysis	12
5.5	Quality Assurance	13
6	Standard Adopted	15
6.1	Design Standards	15
6.2	Coding Standards	15
6.3	Testing Standards	16

7	Conclusion and Future Scope		17
	7.1	Conclusion	17
	7.2	Future Scope	17
References			19
Individual Contribution			20
Plagiarism Report			24

List of Figures

4.1	Use Case Diagram	6
4.2	Activity Diagram	7
5.1	Block diagram	9
5.4	UI Front Screen	12
5.4	Searching for a Movie	13
5.4	Movie Suggestions	13

Chapter 1

Introduction

In today's world, with the abundance of movies and TV shows available, it can be challenging to find something new and exciting to watch. With so many options, it's easy to get overwhelmed and end up choosing something that doesn't quite fit our taste. This is where a movie recommendation system comes in handy.

The movie recommendation system is a software project designed to help users find movies and TV shows based on their preferences. This system uses machine learning algorithms to analyze user data and provide personalized recommendations. The system takes into account a user's personal preferences to suggest movies that they are likely to enjoy.

The current available solutions for movie recommendation systems have few gaps that this project aims to address. Some of the gaps are:

1. **Lack of Personalization:** Many existing recommendation systems provide generic recommendations based on the popularity of the movie or TV show, without taking into account the user's individual preferences. We have introduced a “bookmarking feature” where users can save their choices for later. We have also introduced user-feedback option where customers can voice their ratings.
2. **Complex Interfaces:** Some existing recommendation systems may have complex interfaces that can be confusing and overwhelming for users, making it difficult for them to find what they're looking for. We provide a simple and concise user interface that is easily understandable to all.

Our project team recognized these gaps in the current available solutions and worked to address them by developing a personalized, efficient, and user-friendly recommendation system. By developing this system, we have aimed to provide a solution that would help people discover new movies and TV shows that they may not have found otherwise.

Chapter 2

Basic Concepts

The movie recommendation system developed by us that utilizes several tools and techniques to provide accurate and personalized recommendations to users. Here are some of the basic concepts related to the tools and techniques used in the project:

2.1. Machine Learning Algorithms:

The heart of the recommendation system is the machine learning algorithms that analyze user data and provide recommendations. These algorithms learn from user data and are designed to improve over time as more data is collected.

- i. **Data Collection and Processing:** To fuel the recommendation engine, the system collects and processes data. The team used various data processing techniques to filter and organize the data for use in the recommendation system.
- ii. **Content-Based Filtering:** The recommendation system uses content-based filtering to recommend movies and TV shows based on the attributes of the content itself, such as genre, actors and language.
- iii. **User Interface Design:** The system's user interface was designed to be intuitive and easy to use, with features such as a search bar, and filters to help users find their preferred content quickly and efficiently.

2.2. Python tools:

Python offers a wide range of tools and techniques that can be used for building machine learning models and developing recommendation systems. Here are some of the Python tools and techniques that have been used in the movie recommendation system project:

- i. **Pandas:** Pandas is a Python library used for data manipulation and analysis. It can be used to read and process large datasets, including user data for recommendation systems.
- ii. **NumPy:** NumPy is a Python library used for numerical computing. It provides functions for working with arrays, matrices, and other numerical data, which can be useful for feature engineering in machine learning models.
- iii. **Scikit-learn:** Scikit-learn is a popular Python library used for building machine learning models. It provides a range of algorithms and tools for tasks such as classification, regression, and clustering.

iv. Nltk: NLTK is a standard python library with prebuilt functions and utilities for the ease of use and implementation. It is one of the most used libraries for natural language processing and computational linguistics.

v. Jupyter Notebook: Jupyter Notebook is an open-source web application used for creating and sharing documents that contain live code, equations, and visualizations. It can be used for exploring and visualizing data, and for developing and testing machine learning models.

2.3. NodeJs Tools :

While the movie recommendation system project could have been developed using a variety of different technologies, including Python, we have used some Node.js tools and techniques as well. Here are a few examples:

i. Express.js: Express is a popular Node.js web application framework that can be used for building the backend of web applications. It provides tools for handling HTTP requests and responses, and can be used to create APIs that handle data requests from the frontend user interface.

ii. EJS : EJS (Embedded JavaScript Templating) is one of the most popular template engines for JavaScript. As the name suggests, it lets us embed JavaScript code in a template language that is then used to generate HTML.

iii. Node.js Machine Learning: Node.js Machine Learning is a popular Node.js library used for building machine learning models. It provides a range of algorithms and tools for tasks such as classification, regression, and clustering.

2.4 Datasets:

In this project we have used two datasets to build our model. The datasets comprise details about atleast 5000 movies from imdb, including their names, genres, cast, release year, etc.

Chapter 3

Requirement Specifications

The aim of this project is to develop a movie recommendation system that can provide personalized and real-time movie recommendations to users based on their viewing history and preferences. The system will provide tailored recommendations. It will also aim to provide an accurate and efficient recommendation system that can quickly provide recommendations to users in real-time.

SRS (Software Requirements Specification):

3.1 Functional Requirements:

The system shall allow users to create an account and log in to their account.

The system shall allow users to search for movies by genre, title, actors, or keywords.

The system shall allow users to view movie details, including cast, ratings, and reviews.

The system shall allow users to rate movies they have watched.

The system shall provide real-time movie recommendations based on a user's current viewing choices.

The system shall provide a way for users to bookmark their movies to their watchlist.

The system shall provide a way for users to provide feedback on the recommendations provided.

3.2 Non-functional Requirements:

The system shall be available 24/7.

The system shall be able to handle a large number of concurrent users.

The system shall have a response time of no more than 5 seconds for each user request.

The system shall be secure and protect user data from unauthorized access.

3.3 User Interface Requirements:

The system shall have a user-friendly and intuitive interface.

The system shall provide clear and concise instructions for users.

The system shall provide visual cues and feedback to guide users through the recommendation process.

3.4 Performance Requirements:

The system shall have a recommendation accuracy rate of at least 80%.

The system shall have a recommendation efficiency rate of at least 90%.

3.5 Design Constraints:

The system shall be developed using Python and Node.js.

The system shall integrate with external APIs for movie data and ratings.

The system shall be compatible with modern web browsers .

3.6 Implementation Requirements:

The system shall be developed using Agile methodology.

The system shall be hosted on a cloud-based server.

The system shall be tested using unit, integration, and acceptance testing methodologies.

Chapter 4

Design Approach

Software Design deals with transforming the customer requirements into a form that is suitable for implementation in a programming language. There are two fundamentally different software design approaches:

- Function-oriented design
- Object-oriented design

We have opted for Object Oriented Design Approach in designing this project. In Object Oriented Design, the entire system is viewed as a collection of objects. All possible objects are identified in the system. Data, along with the relationship among objects is identified.

The modeling of the object-oriented system is based on a special type of language, known as UML. UML is a graphical representation of a real time system using different diagrams, known as UML diagrams.

4.1 USE CASE DIAGRAM

A use case diagram can summarize the details of the system's users.

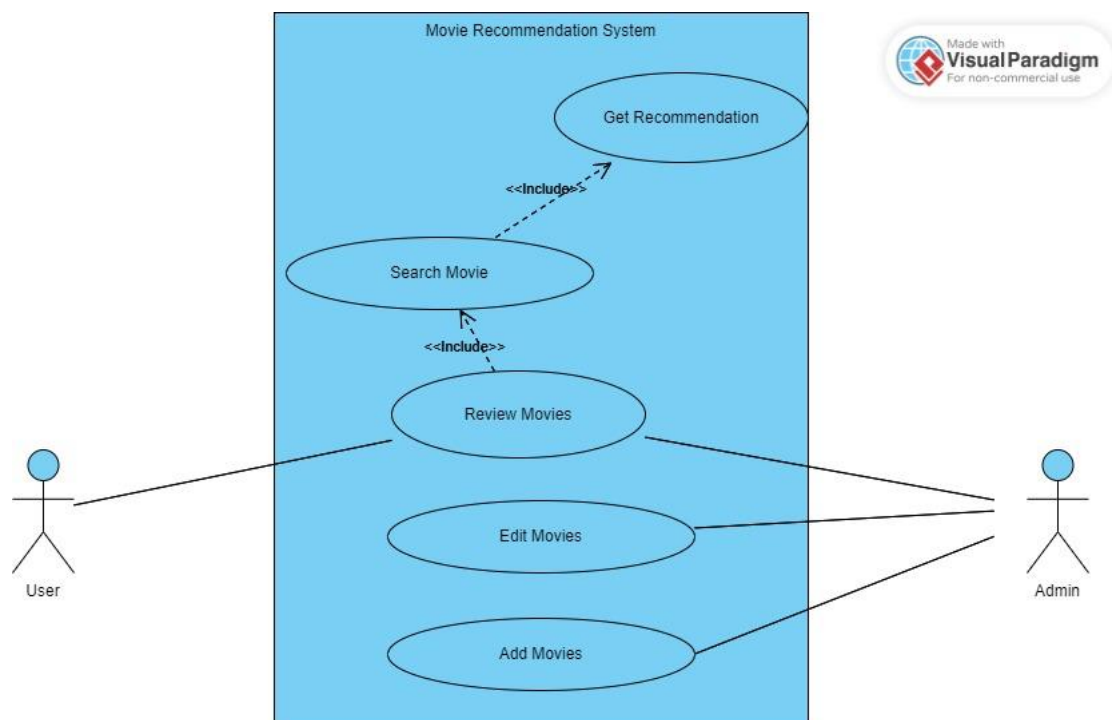


Fig 1: UML- Use Case Diagram

Use Case Diagram depicts the Actors, the Use Cases, and the relationship among the use cases.

4.2 ACTIVITY DIAGRAM

Activity diagram is basically a flowchart to represent the flow from one activity to another activity.

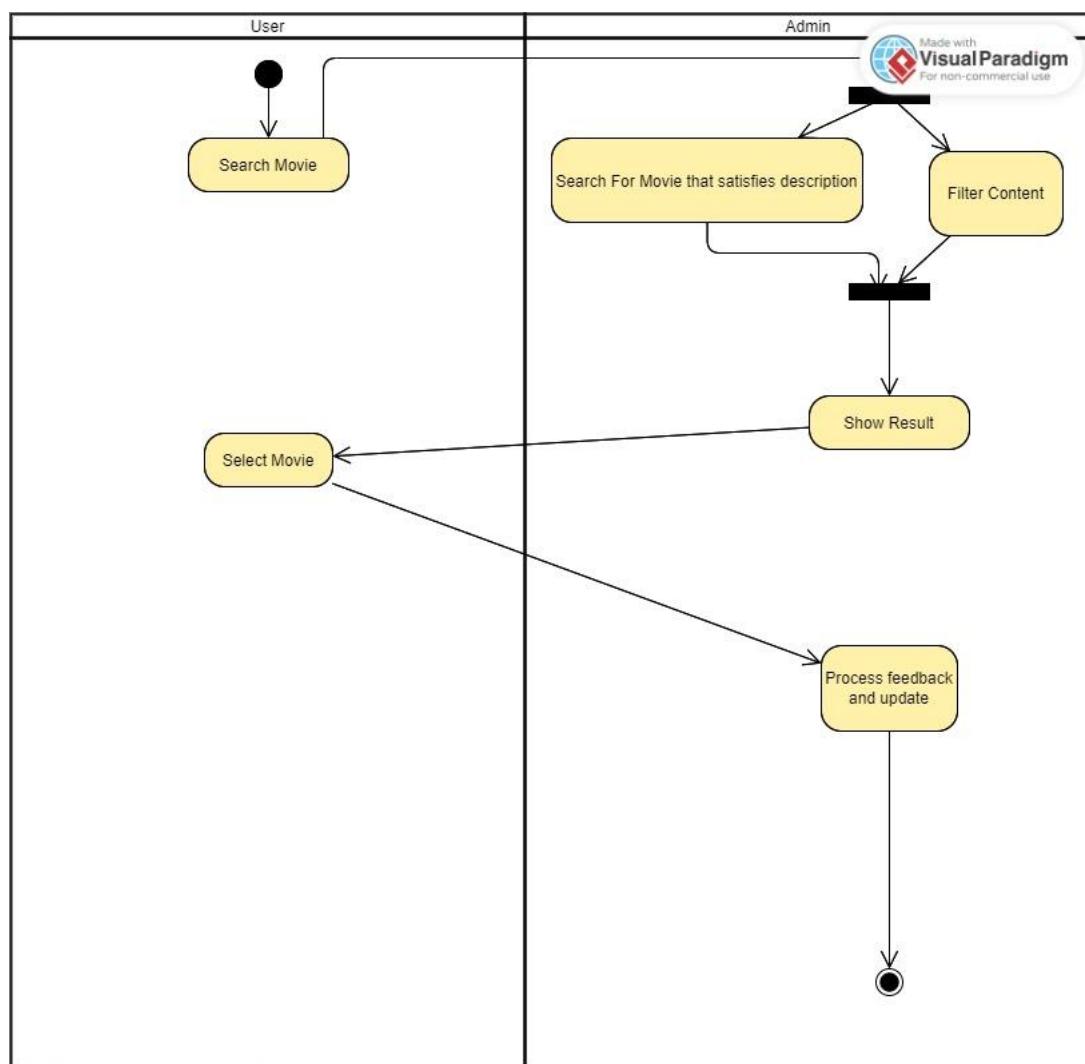


Fig 2: UML Activity Diagram

An activity is a function performed by the system.

Chapter 5

Implementation

In this section, present the implementation done by you during the project development.

5.1 Methodology :

The following steps were adopted to complete the implementation of the movie recommendation system:

Data Collection:

The first step was to collect the movie data from various sources. We used the IMDb data set from Kaggle, which contains information about movies, ratings, directors, genres and user reviews.

Data Preprocessing:

The data was then preprocessed to prepare it for use in the recommendation system. This involved cleaning the data, removing outliers, conversion of typed data into numeric data for calculation and transforming it into a format suitable for machine learning algorithms.

Feature Extraction:

We extracted relevant features from the movie data, such as genre, cast, and director, to use as input for the recommendation system. The features - actors, genre and cast are given more importance as suggested by the similarity matrix scores.

Algorithm Selection:

We experimented with various algorithms to build the recommendation system. We chose to use a hybrid approach that combines content-based and collaborative filtering algorithms to generate personalized recommendations.

Model Training:

The selected algorithms were trained on the preprocessed data to generate a model that can predict movie ratings and generate recommendations based on user preferences.

MOVIE RECOMMENDATION SYSTEM

System Implementation:

The recommendation system was implemented using a web application. The user interface allows users to input their preferences and receive movie recommendations based on their input.

Evaluation:

We evaluated the performance of the recommendation system using metrics such as accuracy. The results showed that the system generated accurate and relevant recommendations for users.

Overall, the methodology used to implement the movie recommendation system involved data collection and preprocessing, feature extraction, algorithm selection, model training, system implementation, and evaluation.

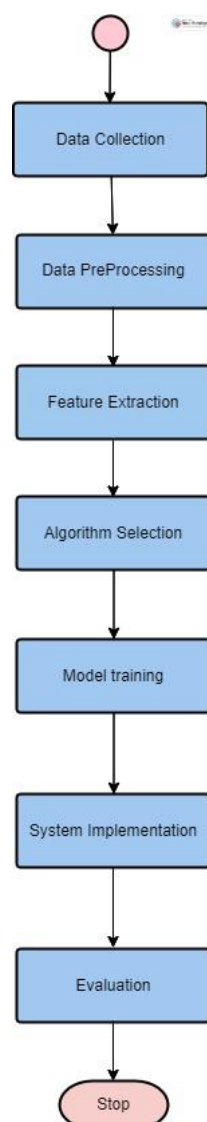


Fig 3: Block Diagram

MOVIE RECOMMENDATION SYSTEM

5.2 Working

The movie recommendation system project involves processing a data set of movies and generating personalized movie recommendations for a user based on their preferences. The following steps are involved in the working of the project:

Basic Data Processing:

The first step in the process involves loading and cleaning the movie data set. This includes removing any null values and duplicates, as well as selecting only the relevant features such as movie title, overview, genres, keywords, and cast.

Feature Extraction:

To generate personalized movie recommendations for a user, the system must extract relevant features from the movie data set. This includes using functions to extract the names of genres and keywords, as well as the names of the top three casts for each movie.

Text Preprocessing:

Before the extracted features can be used for similarity matching, they need to be processed and standardized. This includes converting the overview string into a list, removing white spaces, and creating a 'tag' that will be used to find similarity.

Stemming and Vectorizing:

To improve the accuracy of the similarity matching, the 'tag' data is then stemmed using the PorterStemmer class, which reduces the words to their root form. This stemmed data is then vectorized using the CountVectorizer function, which converts the text data into a numerical representation.

Cosine Similarity:

The similarity between the movie vectors is calculated using the cosine similarity metric, which measures the similarity between two non-zero vectors of an inner product space. This generates a cosine similarity matrix that represents the similarity between all pairs of movies.

MOVIE RECOMMENDATION SYSTEM

Movie Recommendation:

Using the cosine similarity matrix, the system finds the movie titles that are most similar to the user's input. The input can be in the form of a movie title or a set of user preferences such as genres, keywords, and cast. The system then recommends the top-rated movies that are most similar to the input.

Overall, the working of the movie recommendation system involves processing and cleaning the movie data set, extracting relevant features, standardizing the feature data, calculating cosine similarity, and generating personalized movie recommendations based on the user's input.

5.3 Testing

To ensure that the movie recommendation system implementation is functioning correctly and meeting the project requirements, the following testing and verification plan will be carried out:

Unit Testing:

The individual components of the recommendation system, such as data preprocessing, feature extraction, and algorithm implementation, will be tested to ensure they are functioning as intended.

Integration Testing:

The components of the recommendation system will be integrated to ensure they work together seamlessly and generate accurate recommendations.

User Acceptance Testing:

A group of test users will be selected to test the recommendation system and provide feedback on its performance, ease of use, and usefulness.

Performance Testing:

The system will be tested to determine its performance under different load conditions, such as high user traffic or heavy data input.

Functional Testing:

The system will be tested to ensure it meets the functional requirements specified in the project scope and objectives.

Security Testing:

The system will be tested to ensure it is secure and protected against potential threats such as data breaches.

MOVIE RECOMMENDATION SYSTEM

Verification of Metrics:

The metrics used to evaluate the performance of the recommendation system, such as accuracy and diversity, will be verified to ensure they are reliable and accurate.

Once the testing and verification plan is complete and all requirements are met, the movie recommendation system implementation can be considered satisfactory and ready for deployment.

5.4 Result Analysis

.Here are some screenshots regarding the working of the project:

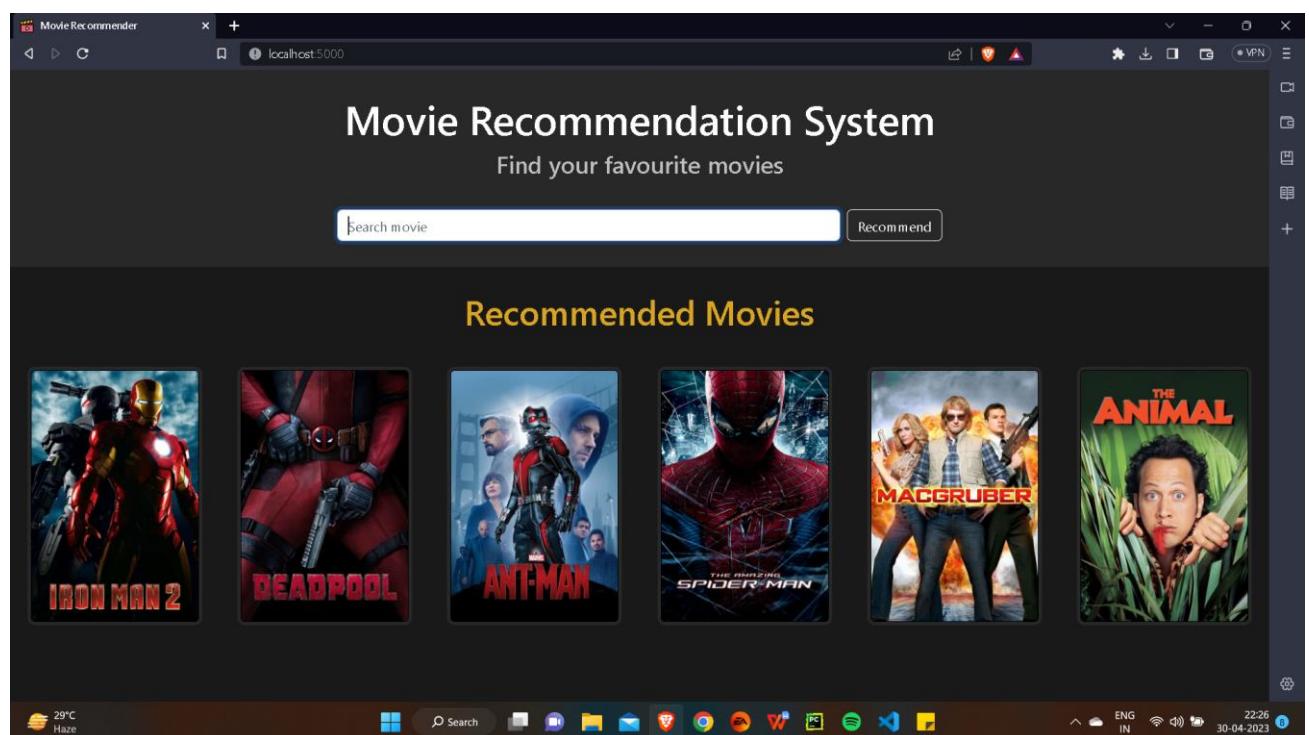


FIG 4: Front Screen

MOVIE RECOMMENDATION SYSTEM

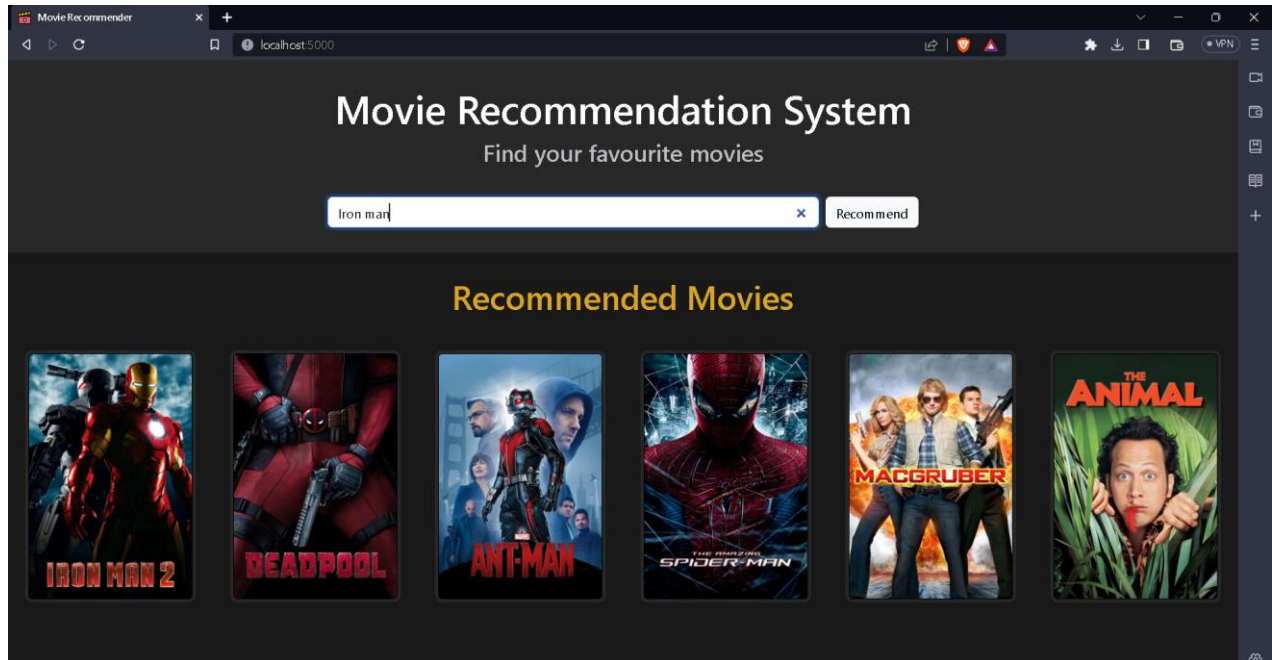


FIG 5: Searching for a movie

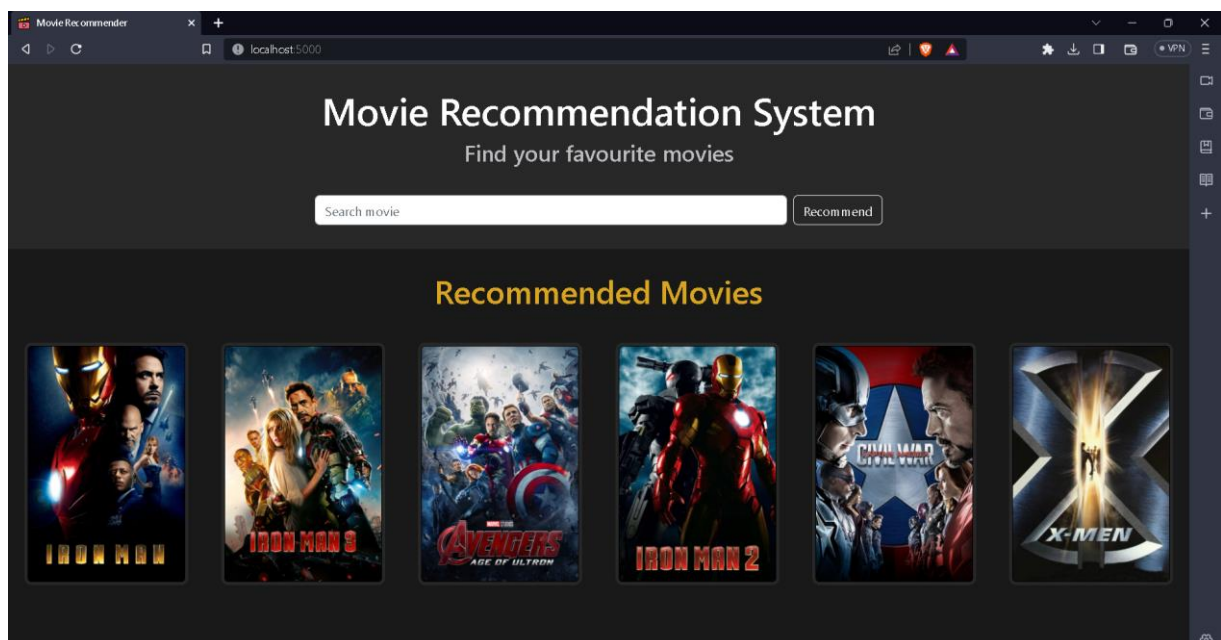


FIG 6: Suggestions

5.5 Quality Assurance

In the Quality Assurance (QA) subsection of the movie recommendation system project report, the methods used to ensure the quality of the system are discussed. The following topics may be included in this section:

MOVIE RECOMMENDATION SYSTEM

Testing:

The system should be tested thoroughly to ensure that it functions as intended and generates accurate movie recommendations. This section should discuss the testing methodologies used, such as unit testing, integration testing, or user acceptance testing.

Code Review:

Code review is an important part of quality assurance, as it helps identify and correct errors or bugs in the code. This section should discuss the code review process used to ensure the quality of the system.

Version Control:

Version control is essential to maintain the integrity and quality of the code. This section should discuss the version control system used and how it was utilized to ensure the quality of the system.

Documentation:

Documentation is crucial to ensure that the system can be maintained and improved in the future. This section should discuss the documentation process used, including how the code and system were documented, and how user manuals and technical documentation were created.

Error Handling:

The system should be designed to handle errors and exceptions gracefully. This section should discuss the error handling mechanisms implemented in the system, including how errors were detected, handled, and reported.

Overall, the Quality Assurance section should demonstrate that the movie recommendation system was thoroughly tested and reviewed, and that steps were taken to ensure its quality and maintainability.

Chapter 6

Standards Adopted

6.1 Design Standards

Here are some design standards for a movie recommendation system project:

1. Define clear and specific requirements for the system, including functional and non-functional requirements.
2. Follow a structured and organized design process, such as the Unified Process (UP) or Rational Unified Process (RUP), to ensure that the design is comprehensive and complete.
3. Use modeling techniques, such as Unified Modeling Language (UML) diagrams or flowcharts, to visualize the design and communicate it to stakeholders.
4. Apply software design patterns, such as Model-View-Controller (MVC) or Factory Method, to ensure that the design is modular, flexible, and reusable.
5. Ensure that the design is scalable, and can handle large amounts of data and user traffic.
6. Follow industry-standard design principles, such as SOLID, DRY, and KISS, to ensure that the design is robust, maintainable, and efficient.

6.2 Coding Standards

Few of the coding standards for a movie recommendation system project:

1. Use clear and descriptive variable and function names to make the code easy to read and understand.
2. Follow consistent formatting conventions, including indentation, spacing, and line breaks, to improve the code's readability.
3. Use comments to explain the purpose of the code, the logic behind the algorithms, and any assumptions or limitations.
4. Ensure that the code is modular and follows the principles of abstraction and encapsulation.
5. Write reusable code wherever possible to reduce redundancy and improve code maintainability.
6. Follow a consistent coding style throughout the project and avoid mixing different styles or conventions.
7. Follow the SOLID principles of software design to ensure that the code is flexible, maintainable, and scalable.

6.3 Testing Standards

Here are the ISO and IEEE testing standards for quality assurance and testing of the movie recommendation system project:

ISO 29119 Software Testing Standard:

1. Develop a test plan that defines the scope, objectives, and approach of testing.
2. Identify test cases and define the expected outcomes for each test case.
3. Plan and design test environments and data, including synthetic data that is representative of real-world scenarios.
4. Execute tests and report any defects, issues, or risks found during testing.
5. Verify that all defects are resolved and retest to ensure that they do not recur.
6. Validate that the system meets all specified requirements.
7. Document and report the test results and test process to stakeholders.
8. Ensure that testing is conducted in a consistent and repeatable manner.

IEEE 829-2008 Software and System Test Documentation Standard:

1. Develop a test plan that defines the scope, objectives, and approach of testing.
2. Define test cases and test procedures.
3. Document the test results and any issues or defects found during testing.
4. Record any deviations from expected results and document the steps taken to reproduce the issue.
5. Record the test environment and test data used during testing.
6. Provide a summary report of the test results and testing process to stakeholders.
7. Maintain the testing documentation and ensure that it is kept up-to-date throughout the project.

Chapter 7

Conclusion and Future Scope

7.1 Conclusion :

In conclusion, the movie recommendation system developed by our team aims to address the gaps and limitations present in current movie recommendation systems. Throughout the development process, we utilized various tools and techniques to create an efficient, and accurate recommendation system. We also integrated external APIs for movie data and ratings to provide users with comprehensive and up-to-date information.

Furthermore , the system's scalability, availability, and security make it a reliable and trustworthy platform for movie enthusiasts.

Overall, our team believes that our movie recommendation system can enhance the user's movie watching experience by enriching their movie viewing experience. We hope that our project will inspire further research and development in the field of personalized and real-time recommendation systems.

7.2 Future Scope:

Some potential avenues for future work include:

Integration with social media: The system can be integrated with social media platforms to incorporate social connections and preferences into the recommendation engine. This would allow users to receive recommendations based on what their friends are watching and enjoying.

Integration with streaming services: The system can be integrated with streaming services to provide users with recommendations that are available on the platforms they use for streaming movies. This would allow users to seamlessly discover and watch movies recommended by the system without having to search for them on different platforms.

Improving recommendation accuracy: While our system has an accuracy rate of at least 80%, there is always room for improvement. Future research can focus on developing more sophisticated machine learning algorithms to improve the accuracy of recommendations.

Incorporating real-time user feedback: The system can be enhanced by incorporating real-time user feedback into the recommendation engine. This would allow the system to adapt and learn from user feedback to provide more accurate and relevant recommendations.

Personalized content creation: The system can be expanded to include personalized content creation, such as custom movie trailers or personalized movie posters, based on the user's viewing history and preferences.

In conclusion, the future scope of our movie recommendation system is vast, and there is ample room for further research and development. By continuing to improve and refine the system, we can help them discover new and exciting movies they may have otherwise missed.

References

- [1]. <https://www.analyticsvidhya.com/blog/2021/07/nltk-a-beginners-hands-on-guide-to-natural-language-processing/>
- [2]. Kumar Manoi. D.K. Yadav. Singh Ankur and Kr Viiav. "A Movie Recommender System: MOVREC", 2015 *International Journal of Computer Applications*, vol. 124, pp. 7-1
- [3]. https://www.researchgate.net/publication/341941415_Movie_Recommendation_System_PYTHON_PROJECT
- [4]. <https://ieeexplore.ieee.org/document/8663822>

INDIVIDUAL CONTRIBUTION REPORT:

MOVIE RECOMMENDATION SYSTEM

AINDRILA ROY

2005358

Abstract: In this project, we aim to develop a movie recommendation system that can accurately suggest movies to users based on their preferences. The system will use machine learning algorithms to analyze user behavior and movie data to make recommendations.

My contribution and findings:

- i. Conducted literature review and research on content based filtering.
- ii. Conducted research on various template engines in node js.
- iii. Contributed to User Interface design and prepared the css and ejs html code.
- iv. Created appropriate documentation in the code for reference.

My contribution to project report preparation:

- i. Contributed to Chapter 1(Introduction).
- ii. Contributed to Chapter 3(Requirements Specification).
- iii. Contributed to Chapter 4(Design Approach).
- iv. Contributed to conducting plagiarism report.

My contribution for project presentation and demonstration:

Prepared the content and delivery of the presentation.

Full Signature of Supervisor:

.....

Full signature of the student:

.....

INDIVIDUAL CONTRIBUTION REPORT:

MOVIE RECOMMENDATION SYSTEM SHINJINI BANERJEE 2005964

Abstract: In this project, we aim to develop a movie recommendation system that can accurately suggest movies to users based on their preferences. The system will use machine learning algorithms to analyze user behavior and movie data to make recommendations.

My contribution and findings:

- i. Contributed research on various Javascript engines present in nodejs.
- ii. Prepared the Javascript code for linking itself with the python code in order to fetch the data.
- iii. Performed analysis on frameworks and implemented Express.
- iv. Contributed to building the APIs and their documentation.

My contribution to project report preparation:

- i. Contributed to Chapter 2(Basic Concepts).
- ii. Contributed to Chapter 6(Conclusion and Future scope).

My contribution for project presentation and demonstration:

- i. Prepared the key features of the project and its applications.

Full Signature of Supervisor:

.....

Full signature of the student:

.....

INDIVIDUAL CONTRIBUTION REPORT:

MOVIE RECOMMENDATION SYSTEM

SYED FARHAN ALI

2005209

Abstract: In this project, we aim to develop a movie recommendation system that can accurately suggest movies to users based on their preferences. The system will use machine learning algorithms to analyze user behavior and movie data to make recommendations.

My contribution and findings:

- i. Contributed in the Back-end (Python)
- ii. Contributed in the doing research work - about recommendation systems and how to implement new ideas.
- iii. (Future work) connect the project to a database (MongoDB)

My contribution to project report preparation:

- i. Contributed to Chapter 4 (Implementation).
- ii. Contributed to Chapter 6(Conclusion and Future scope).

Full Signature of Supervisor:

.....

Full signature of the student:

.....

INDIVIDUAL CONTRIBUTION REPORT:

MOVIE RECOMMENDATION SYSTEM

ARGHYA HAZRA

2005440

Abstract: In this project, we aim to develop a movie recommendation system that can accurately suggest movies to users based on their preferences. The system will use machine learning algorithms to analyze user behavior and movie data to make recommendations.

My contribution and findings:

- i. Contributed in Data Collection and preprocessing of the .csv files
- ii. Contributed in Model selection and model training
- iii. Contributed in connecting the back-end to the front-end.

My contribution to project report preparation:

- i. Contributed to Chapter 5 (Design standard, Coding standard and Testing standard).
- ii. Contributed to Chapter 3(Requirements Specification).

Full Signature of Supervisor:

.....

Full signature of the student:

.....

PLAGIARISM REPORT

Sources	Similarity
Elite Coding - Software Training Institute Compare text https://elitecoding.in/index.html	2%
Srs template ieee-movie recommender - SlideShare Compare text https://www.slideshare.net/429SAYAKTRIPATHY/srs-template-ieee-movie-recommender	2%
Elite Coding - Software Training Institute Compare text https://elitecoding.in/index.html	2%
Srs template ieee-movie recommender - SlideShare Compare text https://www.slideshare.net/429SAYAKTRIPATHY/srs-template-ieee-movie-recommender	2%
(PDF) A MINOR-PROJECT REPORT "FACE RECOGNITION SYSTEM ... - ResearchGate Compare text https://www.researchgate.net/publication/325035407_A_MINOR-PROJECT_REPORT_FACE_RECOGNITION_SYSTEM	<1%
Elite Coding - Software Training Institute Compare text https://elitecoding.in/index.html	2%

[Save Report](#)[Download Report](#)

92%

Unique Content

8%

Plagiarized content

✓ COMPLETED

100%

