INTERNSHIP PROJECT REPORT

ON

MOVIE RECOMMENDATION SYSTEM

SUBMITTED BY:

PRANEETH

(4DM20IS034)

AT

Zephyr Technologies and Solutions Pvt Ltd.

5th FLOOR, OBERLE TOWERS, BALMATTA RD,

BENDOOR, MANGALORE, KARNATAKA 575002,

INDIA

ACKNOWLEDGEMT

We dedicate this page to acknowledge and express our gratitude to those who contributed to the success of this project. Without their guidance and support, the process of constructing the dissertation would not have been as smooth and efficient.

We extend our sincere thanks to the employees of Zephyr Technologies, under whose guidance we executed this project. Their continuous support and willingness to share their extensive knowledge allowed us to gain a deep understanding of the project and its implications, enabling us to complete our assigned tasks effectively.

Last but not least, we would like to express our appreciation to our parents for their blessings, as well as to our friends and classmates for their assistance and well-wishes throughout the successful completion of this project.

ABOUT THE COMPANY

ZEPHYR TECHNOLOGIES is a software company delivering high quality, cost effective, reliable result-oriented web and e-commerce solutions on time for a global clientele. professionalism, skill and expertise are the tools we use to make the web work for your business bringing in maximum return on your investment in the shortest possible time. we have delivered on IT projects of varying complexities for their very demanding and internet clients spread across the globe. they develop unique web solutions which ensure increased efficiency and competitive advantage for your business and thus to your end-users.

Their tools are professionalism, skills and expertise that translate into delivering quality work at every step for any project we undertake. they work towards getting better than the best out of every team member at ZEPHYR TECHNOLOGIES, which means when you hire them all-around quality is assured off as you want it. their advantage quality includes protection of intellectual for the source codes developed specifically for your business. they do not sell the source codes to third parties and all elements that they create for your web solution belongs to you. ZEPHYR TECHNOLOGIES project managers and business analysts place great value on building a clean communication link with you as they consider it the key ingredient for the success of any project at hand.

ABSTRACT

Recommender System is a tool helping users find content and overcome information overload. It predicts interests of users and makes recommendation according to the interest model of users. The original content-based recommender system is the continuation and development of collaborative filtering, which doesn’t need the user’s evaluation for items. Instead, the similarity is calculated based on the information of items that are chose by users, and then make the recommendation accordingly. With the improvement of machine learning, current content-based recommender system can build profile for users and products respectively. Building or updating the profile according to the analysis of items that are bought or visited by users. The system can compare the user and the profile of items and then recommend the most similar products. So this recommender method that compare user and product directly cannot be brought into collaborative filtering model. The foundation of content-based algorithm is acquisition and quantitative analysis of the content. As the research of acquisition and filtering of text information are mature, many current content-based recommender systems make recommendation according to the analysis of text information. This project introduces content-based recommender system for the movie website . There are a lot of features extracted from the movie, they are diversity and unique, which is also the difference from other recommender systems. We use these features to construct movie model and calculate similarity. We introduce a new approach for setting weight of features, which improves the representative of movies. Finally we evaluate the approach to illustrate the improvement.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 Relevance of the Project

A recommendation system or recommendation engine is a model used for information filtering where it tries to predict the preferences of a user and provide suggests based on these preferences. These systems have become increasingly popular nowadays and are widely used today in areas such as movies, music, books, videos, clothing, restaurants, food, places and other utilities. These systems collect information about a user's preferences and behaviour, and then use this information to improve their suggestions in the future. Movies are a part and parcel of life. There are different types of movies like some for entertainment, some for educational purposes, some are animated movies for children, and some are horror movies or action films. Movies can be easily differentiated through their genres like comedy, thriller, animation, action etc. Other way to distinguish among movies can be either by releasing year, language, director etc. Watching movies online, there are a number of movies to search in our most liked movies . Movie Recommendation Systems helps us to search our preferred movies among all of these different types of movies and hence reduce the trouble of spending a lot of time searching our favourable movies. So, it requires that the movie recommendation system should be very reliable and should provide us with the recommendation of movies which are exactly same or most matched with our preferences. A large number of companies are making use of recommendation systems to increase user interaction and enrich a user's shopping experience. Recommendation systems have several benefits, the most important being customer satisfaction and revenue. Movie Recommendation system is very powerful and important system. But, due to the problems associated with pure collaborative approach, movie recommendation systems also suffers with poor recommendation .

1. 2 Problem Statement:

For building a recommender system from scratch, we face several different problems. Currently there are a lot of recommender systems based on the user information, so what should we do if the website has not gotten enough users. After that, we will solve the representation of a movie, which is how a system can understand a movie. That is the precondition for comparing similarity between two movies. Movie features such as genre, actor and director is a way that can categorize movies. But for each feature of the movie, there should be different weight for them and each of them plays a different role for recommendation. So we get these questions:

• How to recommend movies when there are no user information.

• What kind of movie features can be used for the recommender system.

• How to calculate the similarity between two movies.

• Is it possible to set weight for each feature.

1.3 Objective of the Projects

❖ **Literature Review:**

✓ Provide an overview of existing recommendation systems, both content-based and collaborative filtering.

✓ Discuss the advantages and limitations of content-based approaches.

✓ Include any notable studies or implementations related to content-based movie recommendation systems.

❖ **Data Collection and Preprocessing:**

✓ Describe the data sources used for the recommendation system.

✓ Explain the process of data acquisition, cleaning, and transformation.

✓ Highlight any challenges faced during this phase.

❖ **Feature Engineering:**

✓ Identify the relevant features (attributes) for movie representation in the system.

✓ Discuss how features like genre, director, actors, release year, etc., are selected and processed.

❖ **Machine Learning Model Selection:**

✓ Explain the choice of machine learning algorithms for building the recommendation system.

✓ Justify why the chosen algorithms are suitable for the content-based approach.

❖ **Model Training and Evaluation:**

✓ Detail the process of training the machine learning model on the prepared dataset.

✓ Define appropriate evaluation metrics (e.g., accuracy, precision, recall, etc.) and explain why they are chosen for this task.

✓ Provide performance results on both training and validation sets.

❖ **Recommendation Process:**

✓ Describe how the system generates movie recommendations based on user preferences and movie attributes.

✓ Explain the algorithm's decision-making process in recommending movies.

1.4 Scope of the Project

The scope of this project encompasses the development and implementation of a content-based movie recommendation system using machine learning techniques. The primary objective is to create a personalized movie suggestion platform that leverages user preferences and movie attributes to generate tailored recommendations. The project will involve data collection from diverse movie databases, encompassing attributes such as genre, director, actors, release year, and more. Data preprocessing techniques will be employed to clean, transform, and engineer features for effective model training. The machine learning model selection will be a critical aspect, involving algorithms capable of understanding and processing the complex relationships between user preferences and movie characteristics. The project's scope extends to model training, validation, and evaluation, with performance metrics carefully selected to assess the recommendation system's accuracy and effectiveness. Additionally, there is an option to design a user interface for seamless interaction, though this aspect is considered optional. The project's focus will be on content-based approaches, differentiating it from collaborative filtering methods. It's important to acknowledge potential challenges that may arise during the development and deployment phases, and suggestions for future improvements will be considered to enhance the recommendation system's capabilities. Overall, this project aims to deliver a robust and user-friendly content-based movie recommendation system that can significantly enhance the movie-watching experience for users.

CHAPTER 2

BACKGROUND RESEARCH

2.1. Background Study

With the various types of datasets that have been utilized for detection jobs in today's world, the detection sector has been able to accomplish significant growth. However,because real- life objects are often variable in nature due to environmental differences such as size, background, size, and many other aspects, annotation jobs can be a significant difficulty in the field of categorization and detection.

2.1.1 Supervised Learning

There is little doubt machine learning has become one of the most powerful technologies in the last decade. The emphasis on “learning” in machine learning allows computers to make better and better decisions, based on previous experiences. Classical machine learning is often categorized into supervised, unsupervised, semi-supervised or reinforced learning depending on how the algorithm learns to become more accurate in its predictions.Among them, supervised learning is one of the heavily explored and important form of ML. In Supervised Learning, the learning process is done under the seen label of observation variables. Datasets are trained with the training sets to build a model which is used later on to label new observations or data points from the testing set. As for the training set, the input variables are the features which will influence the accuracy of predicted variable. It contains both quantitative and qualitative variables; the output variable is the label class that Supervised Learning will label the new observations.

2.2. Literature Review

There are three techniques of recommendation system: Collaborative Filtering, Content-Based Filtering and Hybrid Filtering. In Content Based recommended system, user provides data either explicitly (rating) or implicitly (by clicking on a link). The system captures this data and generates user profile for every user. By making use of user profile,recommendation is generated. In content based filtering, recommendation is given by only watching single user's profile. System tries to recommend item similar to that item based on users past activity. Unlike content based, collaborative filtering finds those users whose liking are similar to a given user. It then recommends item or any product, by considering that the given user will also like the item which other users like because their taste are similar. Both these technique have their own strength and weakness so to overcome this,hybrid technique came into picture, which is a combination of both these techniques.Hybrid filtering can be used in various types. We can use content based filtering first and then pass those results to collaborative recommender (and vice-versa) or by integrating both the filter into one model to generate the result. These kinds of modifications are also uses to cope up with cold start, data sparsity and scalability problem

2.2.1 Content-Based Filtering

Content-Based Filtering are also known as cognitive filtering. This filtering recommends item to the user based on his past experience. For example, if a user likes only action movies then the system predicts him only action movies similar to it which he has highly rated. The broader explanation could be suppose the user likes only politics related content so the system suggests the websites, blogs or the news similar to that content.Unlike collaborative filtering, content based filtering do not face new user problem. It does not have other user interaction in it. It only deals with particular user's interest.Content based filtering first checks the user preference and then suggest him with the movies or any other product to him. It only focus on single user's ideas, thoughts and gives prediction based on his interest. So if we talk about movies, then the content based filtering technique checks the rating given by the user.

2.2.2 Collaborative Filtering

The concept of collaborative filtering was first introduced in 1991 by Goldberg et al. [20].The Tapestry system applies only to smaller user groups (e.g. a single unit), and has too many demands on the user. As a proto-type of collaborative filtering recommendation system, Tapestry presents a new recommendation, but there are many technical deficiencies. Since then, there has been a scoring based collaborative filtering recommendation system, such as Group lens, which recommends news and films. At present many e commerce sites have been using the recommendation system such as Amazon, CDNow, Drugstore and Movie finder etc. There is massive amount of data available. As we all know that today in this busy life no one has time to search hundreds of thousands of item and select the one which is similar to their taste. So collaborative filtering is one of the ways to filter the data and provide the relevant information in which the user is interested in. Collaborative Filtering is one of the most well known techniques for recommending items. This technique suggests relevant item to the user based neighbouring choice. It first finds out the similarity between the user and his neighbour and then predicts the items. There can be n number of users. This technique finds the similar user from the list of user's. But the similarity between users is found out based the ratings which the users have given to the particular item. This way the approach continues and the desired result is generated. This strategy takes ratings given by user for any item from the large catalog of item catalog of ratings given by the user. This large catalog is referred as user-item matrix

2.3. Current System

The reason behind this improvement is the popularity gained by organizations like Netflix whose primary objective is customer satisfaction. Before existence the recommendation system, individuals would physically choose movies to watch from movie libraries. They either had to read the user's reviews or based on the review they would select a movie or had to randomly select a movie. This procedure isn't feasible, as there is an enormous number of spectators with a unique preference for movies. Hence many recommendation systems have been developed over the past decade. These systems use different approaches like a collaborative approach, a content-based approach, a hybrid approach,etc. Taking a look at the behavior and history of different clients, based on their ratings,the system suggests to us what to watch without having to put effort into deciding what to watch. These recommendation system follows content based recommendation where content-based approach is limited to a single user, where the user's past history and ratings are used for providing recommendations. There are a number of methodologies introduced to implement this recommendation system which includes various fields of Data Mining, Clustering and Bayesian Network methodology

2.4. The Problem with Current System

System Perhaps the biggest issue facing recommender systems is that they need a lot of data to effectively make recommendations. It's no coincidence that the companies most identified with having excellent recommendations are those with a lot of consumer user data:Google, Amazon and Netflix. A good recommender system firstly needs item data (from a catalog or other form), then it must capture and analyze user data (behavioral events), and then the magic algorithm does its work. The more item and user data a recommender system has to work with, the stronger the chances of getting good recommendations. But it can be a chicken and egg problem – to get good recommendations, you need a lot of users, so you can get a lot of data for the recommendations. There are many other issues that can happen with recommender systems – some offer up too many 'lowest common denominator' recommendations, some don't support The Long Tail enough and just recommend obvious items, outliers can be a problem, and so on. Some other problems are unpredictable items, changing user preferences, changing data etc.

CHAPTER 3

SYSTEM ARCHITECTURE

3.1 Working

A content-based movie recommendation system recommends movies to users based on the content of the movies they have previously watched or rated. The system first creates a profile for each user, which contains information about the movies they have interacted with. This information can include the genres of the movies, the actors and directors, and the user's ratings of the movies.

To recommend movies to a user, the system compares their profile to the profiles of other users. The system identifies movies that are similar to the movies that the user has liked in the past. The system then recommends these movies to the user. The accuracy of a content-based movie recommendation system depends on the quality of the data used to create the user profiles. The more movies that a user has watched or rated, the more accurate the recommendations will be.

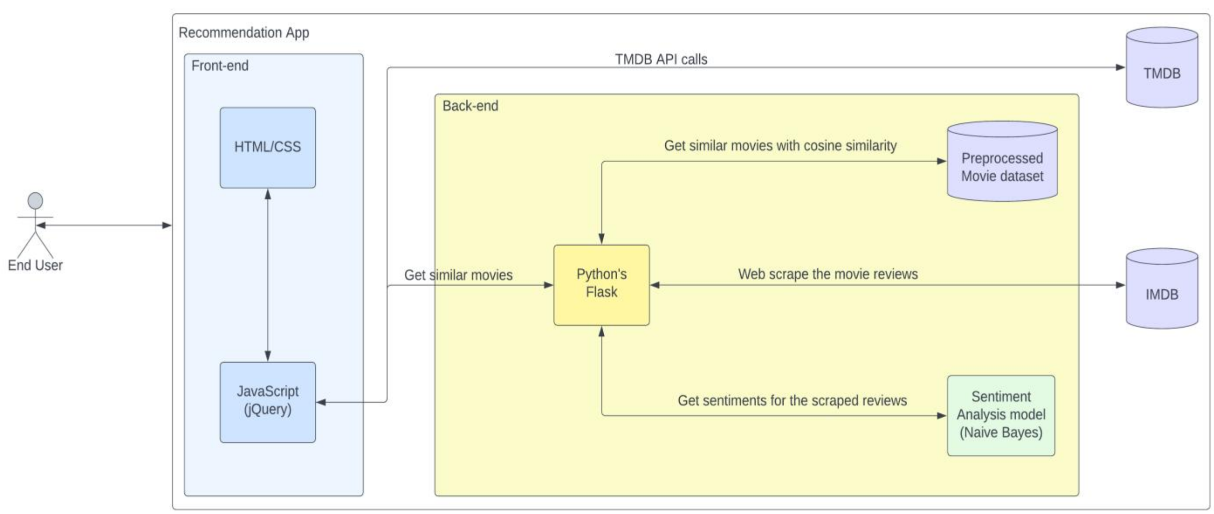


Fig 3.1.1 Architecture

3.2 Multinomial Naive Bayes Algorithm for Sentiment Analysis

The Multinomial Naive Bayes (MNB) algorithm is a probabilistic machine learning method widely used for text classification tasks, such as sentiment analysis. It is based on Bayes' theorem and makes an assumption of conditional independence of features given the class label. In the context of sentiment analysis, MNB models the likelihood of observing words given a particular sentiment class and uses this information to predict the sentiment of new text data.

- **TF-IDF Vectorization**

✓ Before applying the MNB algorithm, the text data is transformed using TF-IDF (Term Frequency-Inverse Document Frequency) vectorization.

✓ TF-IDF assigns a weight to each word based on its frequency in a specific document and its rarity in the entire corpus. This helps capture the importance of words in a document relative to the entire dataset.

- **Training Phase:**

✓ The dataset is divided into training and testing sets. The training set is used to train the classifier.

✓ During training, MNB calculates the probabilities of observing each word in both positive and negative sentiments. It also computes the prior probabilities of a review being positive or negative.

- **Bayes' Theorem for Classification:**

✓ MNB applies Bayes' theorem to calculate the posterior probability of a class (positive or negative sentiment) given an observation (TF-IDF vector).

✓ It combines the prior probabilities with the likelihoods of observing words in each class to make predictions.

- **Cosine Similarity:**

Cosine similarity is a metric used to measure how similar the documents are irrespective of their size. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space.

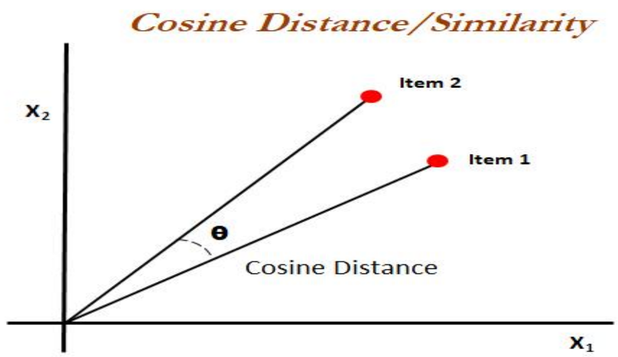


Fig 3.2.1 Cosine Distance

The cosine similarity is advantageous because even if the two similar documents are far apart by the Euclidean distance (due to the size of the document), chances are they may still be oriented closer together. The smaller the angle, higher the cosine similarity.

- **Prediction Phase:**

✓ After training, the model can be used to predict sentiments on new, unseen data.

✓ For each new review, the TF-IDF vector is computed. The model then uses the learned probabilities to predict whether the review is positive or negative.

3.3 Experimental Results

- **Accuracy on Test Set (80-20 Split):**

✓ Approximately 97.47%

✓ This means that when the model was evaluated on a set of data it had never seen before (the test set), it correctly predicted the sentiment of about 97.47% of the reviews.

- **Accuracy on Full Dataset:**

✓ Approximately 98.77%

✓ After training on the entire dataset and then testing on the same dataset, the model achieved an even higher accuracy of about 98.77%.

3.4 Experimental Analysis

The experimental analysis involved the implementation of a sentiment analysis model using the Multinomial Naive Bayes algorithm in conjunction with TF-IDF vectorization. The dataset consisted of a collection of reviews with corresponding sentiment labels. The model was trained and evaluated using two different approaches. In the first approach, the dataset was split into training and testing sets with an 80-20 ratio. The model demonstrated commendable performance, achieving an accuracy of approximately 97.47% on the test set. Subsequently, the model was retrained using the entire dataset, and it exhibited an even higher accuracy of about 98.77% on the same data, showcasing its capability to effectively discern between positive and negative sentiments. These results underscore the robustness of the Multinomial Naive Bayes algorithm in text classification tasks, particularly in the domain of sentiment analysis. However, it is imperative to note that while accuracy provides valuable insights, additional metrics such as precision, recall, and F1-score should be considered for a more comprehensive evaluation, especially in scenarios involving class imbalance. Furthermore, future work could involve conducting cross-validation and exploring alternative algorithms .

3.5 TMDB API

The TMDb API, provided by The Movie Database, is a powerful tool that allows developers to access a vast amount of information related to movies and TV shows. By integrating the TMDb API into a movie recommendation system, you can enhance the system's capabilities by providing detailed metadata, images, ratings, and other information about movies. Here's how the API concept could be applied:

- **Accessing Movie Information:**

The TMDb API allows you to retrieve detailed information about movies, including titles, release dates, genres, cast and crew, plot summaries, and more. This information can enrich the dataset used by the recommendation system.

- **Fetching Images and Posters:**

The API provides access to a wide range of images, including posters, backdrops, and stills. These images can be used to enhance the user interface of the recommendation system, providing visual context for each movie.

- **Ratings and Reviews:**

TMDb offers user ratings and reviews for movies. By utilizing the API, you can integrate these ratings into your recommendation system to provide users with additional insights into the popularity and quality of recommended movies.

- **Discovering Similar Movies:**

TMDb's API provides endpoints for discovering similar movies based on a given movie. This functionality can be used to implement content-based recommendations, where movies with similar attributes are suggested to users.

- **Real-Time Updates:**

The API offers real-time updates, ensuring that your recommendation system stays up-to-date with the latest information about movies. This includes new releases, trending movies, and updates to existing titles.

- **Search and Filtering:**

The API allows for advanced search and filtering options, enabling users to find movies based on criteria such as genre, release year, language, and more. This can be useful for implementing user-specific filters in the recommendation system.

- **Authentication and Security:**

The TMDb API may require authentication using an API key, which helps control access and usage. This ensures that only authorized users or applications can access the data.

- **Compliance with Terms of Use:**

When using the TMDb API, it's important to comply with their terms of use and licensing agreements. This may include providing proper attribution and adhering to any usage limits or restrictions imposed by TMDb.

CHAPTER 4

SYSTEM ANALYSIS

4.1 Requirement Elicitation and Analysis

Movie recommendation system is a web-based app, which provides all the details of the requested movie. Details include recommended movies, as well as top cast, ratings,reviews and so on.

The requirement of this project is given below:

4.1.1. Functional Requirement

- Validate each user input to database.
- Auto suggest user for smooth experience.
- Simple loading screen to inform user that work is in progress.
- Notify user if result is not found.
- Simply UI to show more details about the casts.

4.1.2. Non-Functional Requirement

- The processing of each request should be done around 10 seconds.
- Display default data if some data are missing.
- Search result, although auto suggest unable to suggest.

4.1.3. Hardware Requirements

- Processor: Pentium IV or higher
- RAM: 256 MB
- Network: Bandwidth greater than 50 KBps (400 kbps)

4.1.4 Software Requirements

- OS: Windows, iOS, Linux
- IDE:Jupyter Notebook,Visual Studio Code
- Python version
- Browser: Chrome, Brave, Mozilla Firefox, Microsoft Edge, Apple Safari,Opera.
- Anaconda distribution:

Anaconda is a free and open-source distribution of the Python programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management system and deployment. Package versions are managed by the package management system conda. The anaconda distribution includes data-science packages suitable for Windows, Linux and MacOS.3

- 3 Python libraries:

✓ SKlearn:

It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN.

✓ NumPy:

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python.

✓ Pandas:

Pandas is one of the most widely used python libraries in data science. It provides high-performance, easy to use structures and data analysis tools.

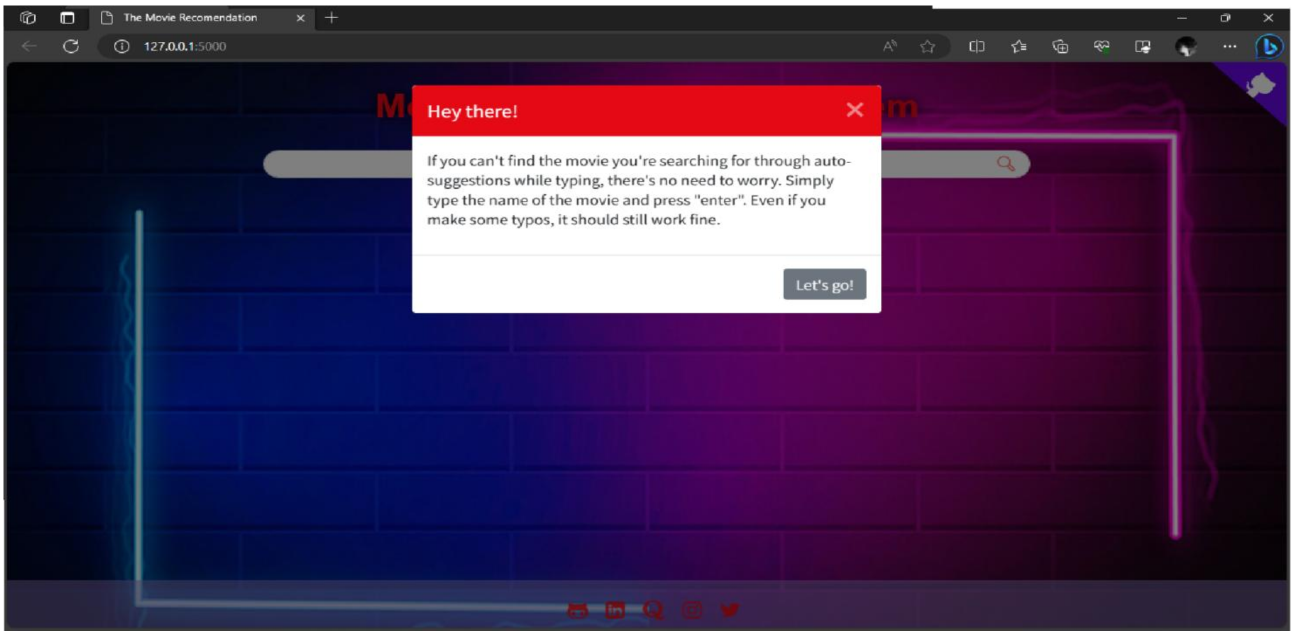
✓ Flask:

It is a lightweight WSGI web application framework.

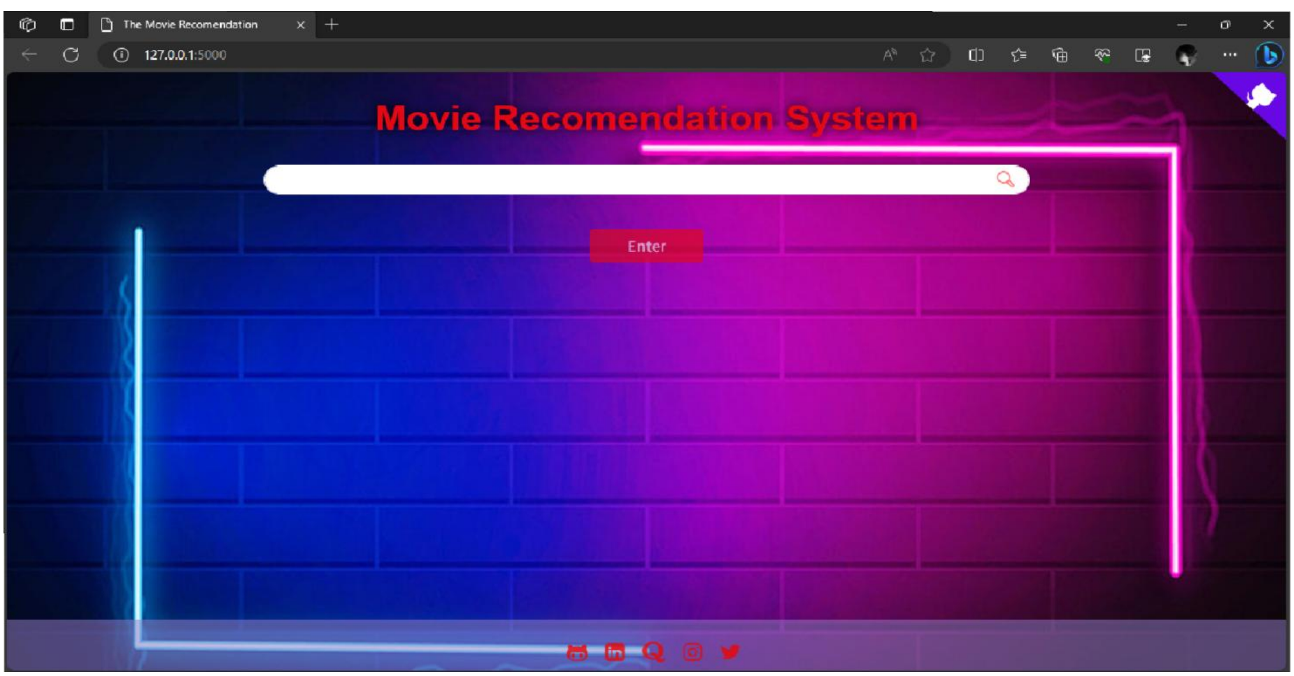
CHAPTER 5

SNAPSHOT AND RESULTS

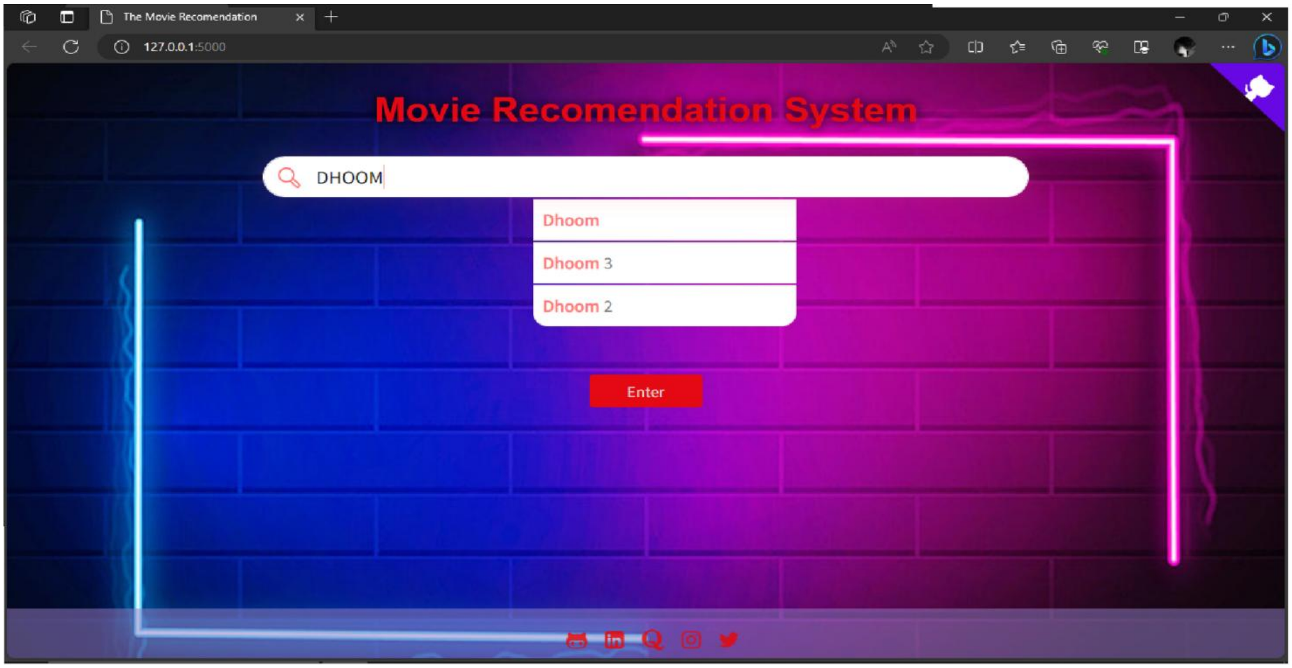
5.1 Instruction Box



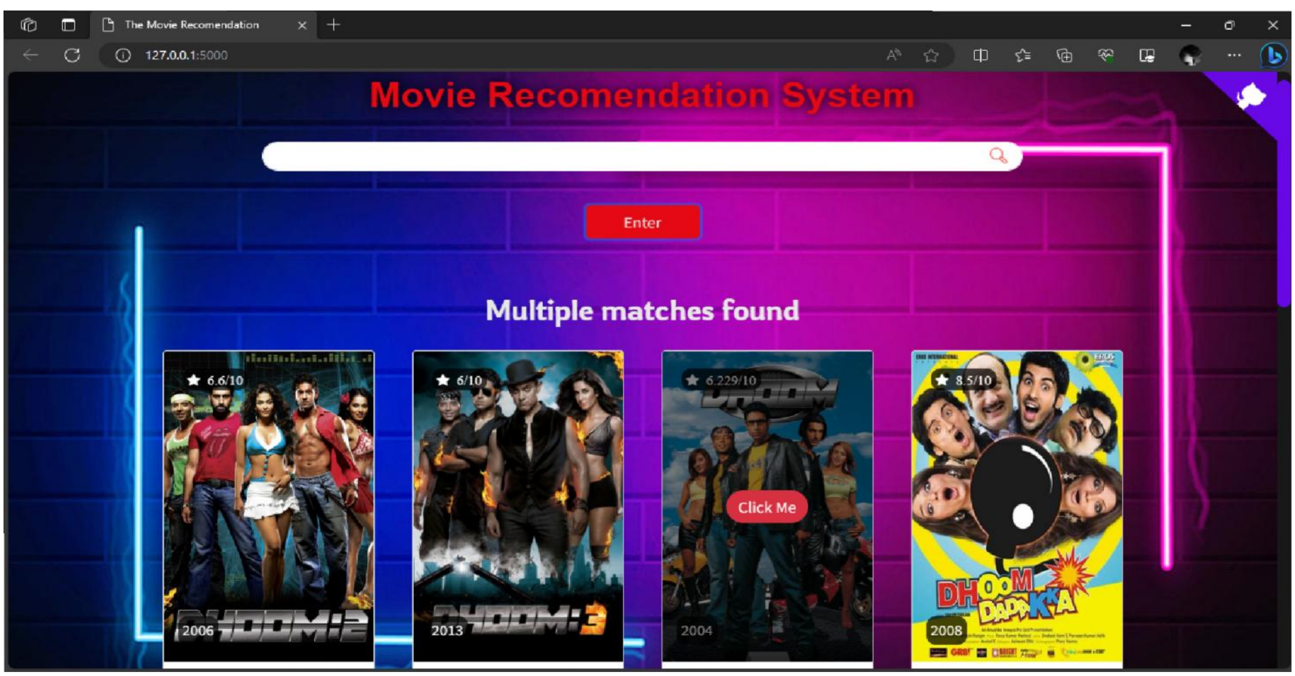
5.2 Home Page



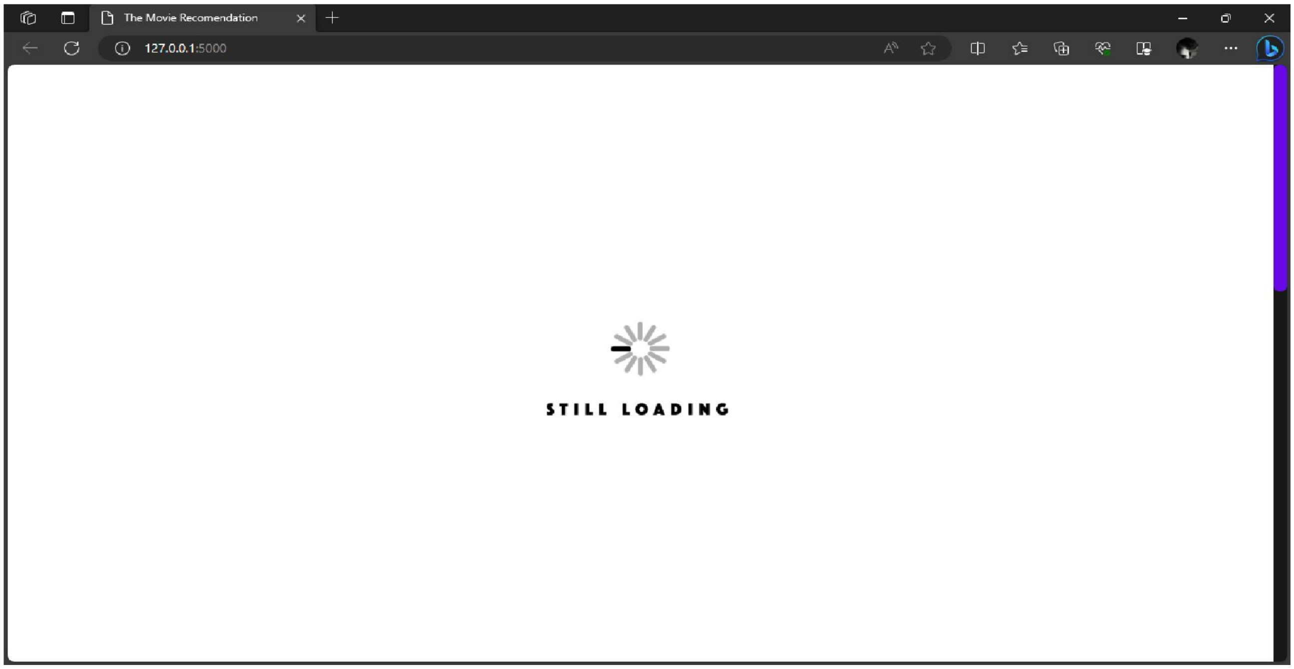
5.3 Movie Search



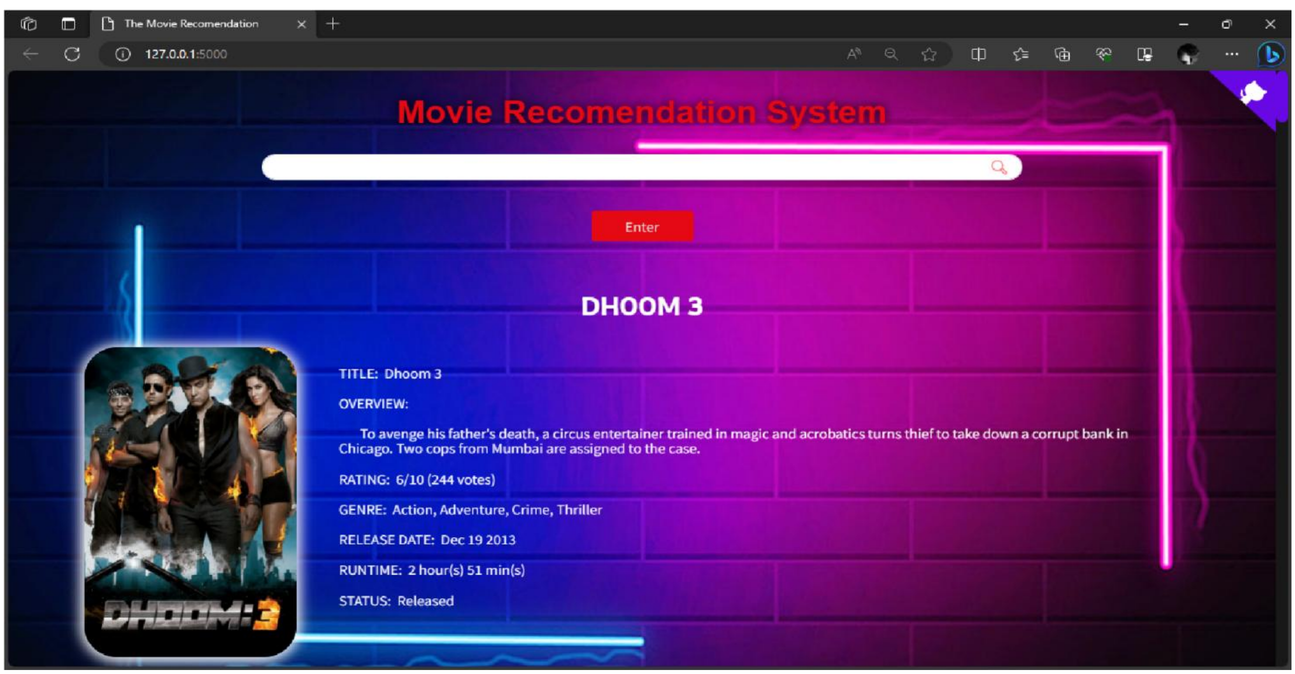
5.3.1 Matching Movie List



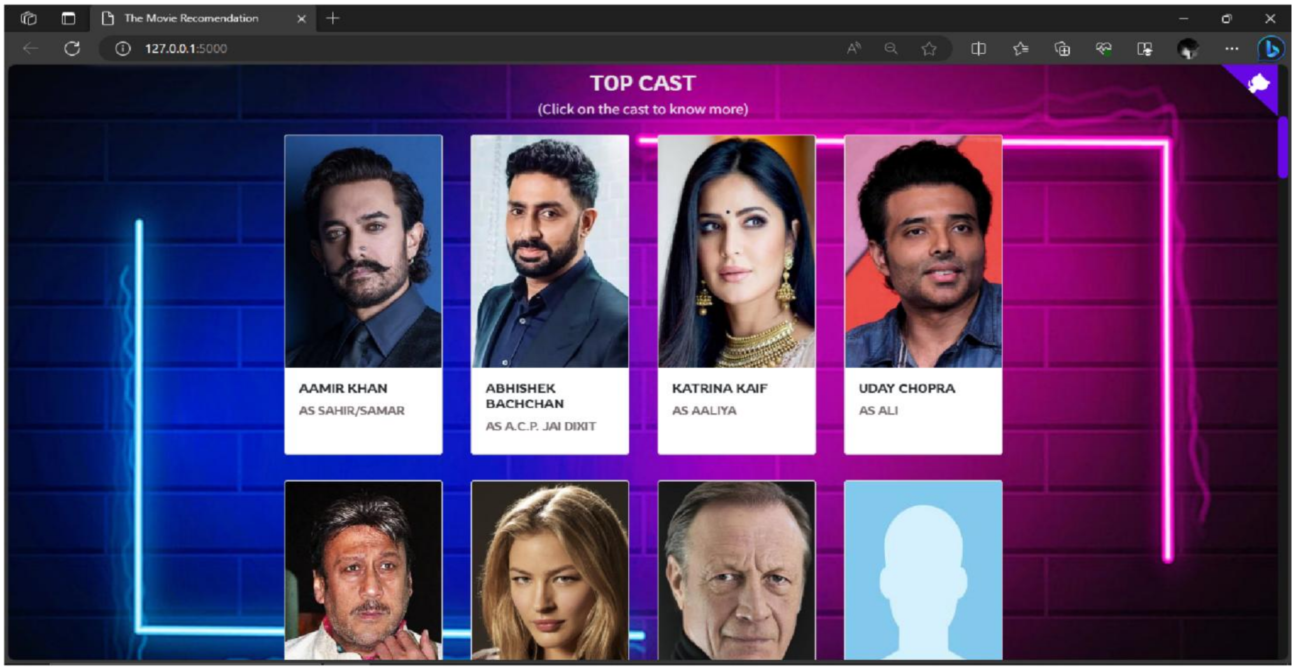
5.3.1.1 Loading to Display



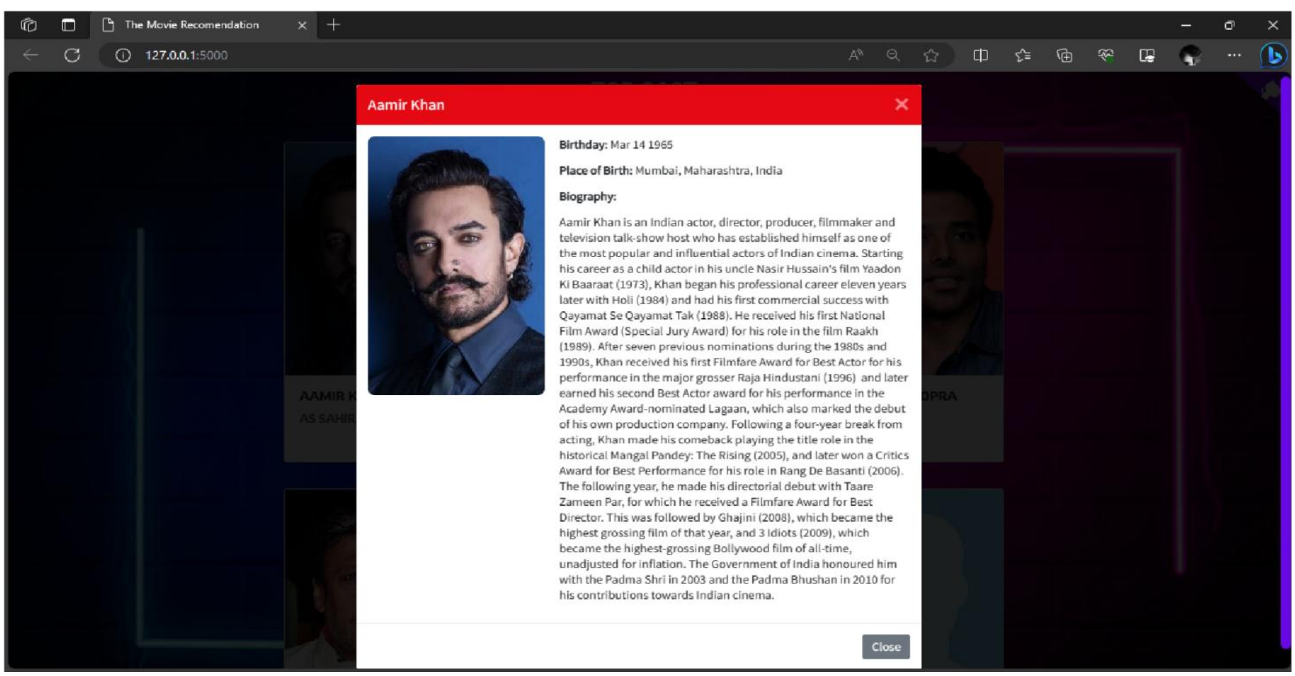
5.3.1.2 Movie Description



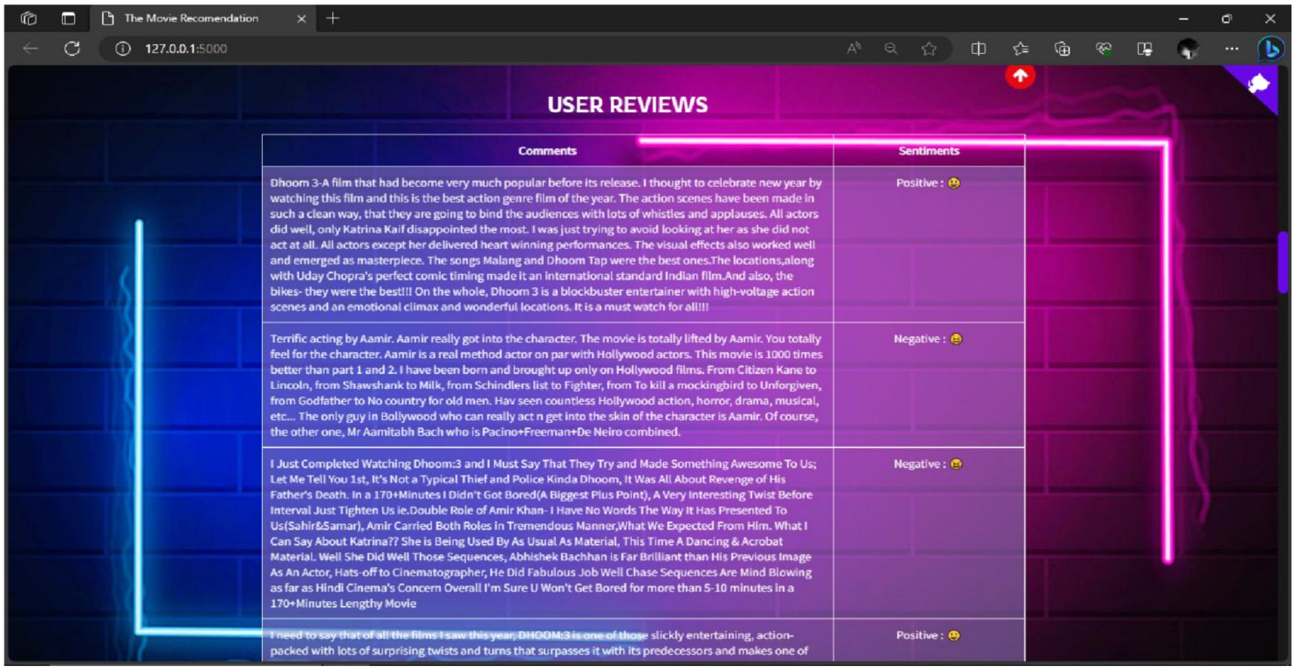
5.3.1.3 Casting of the Movie



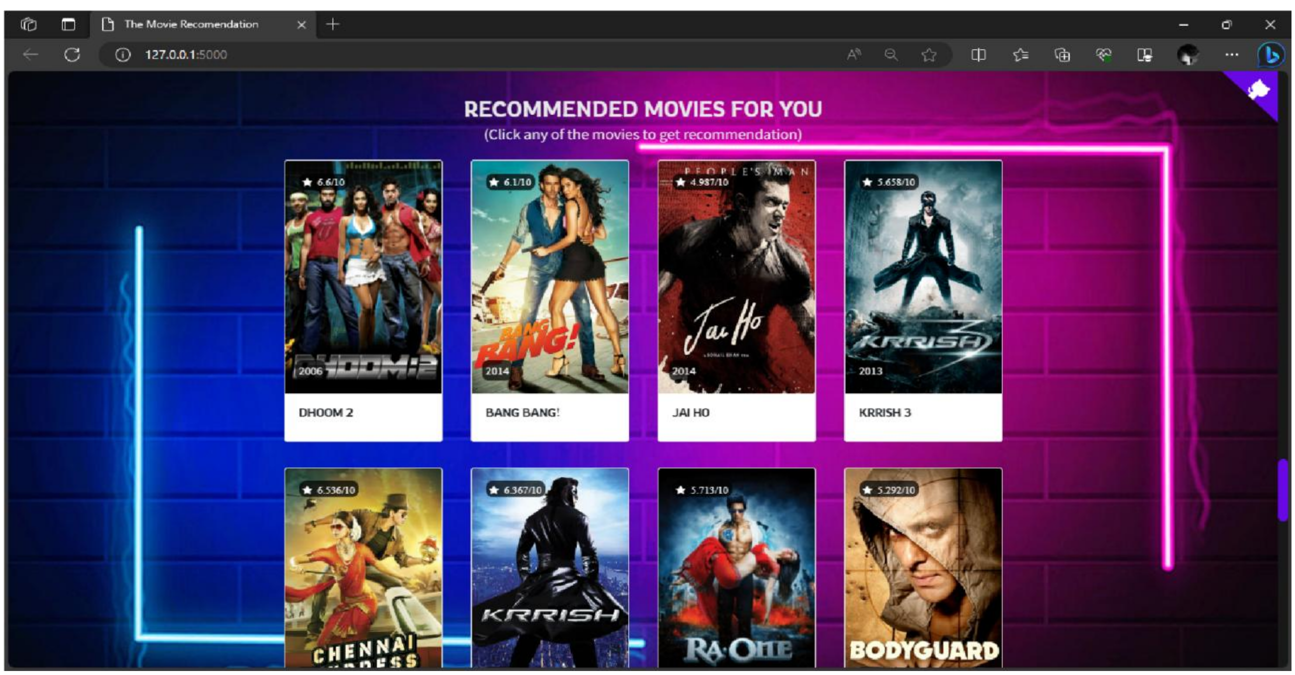
5.3.1.3.1 Actor Details



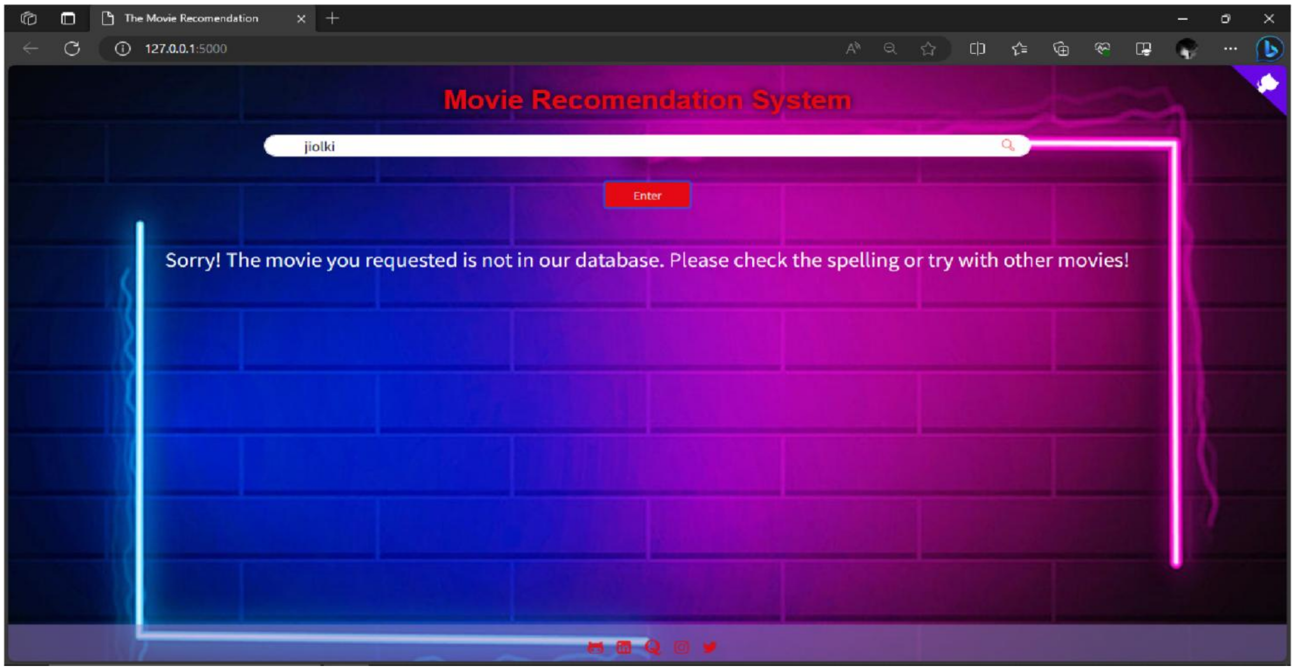
5.3.1.4 User Reviews



5.3.1.5 Recommendations



5.3.1.6 Movie not Found



CHAPTER 6

CONCLUSION

- **Effectiveness of Collaborative Filtering:**

The collaborative filtering approach proved to be effective in generating personalized movie recommendations.

- **Improved User Engagement:**

The recommendation system has the potential to enhance user engagement with the platform.

- **Handling Cold Start Problem:**

The system exhibited robustness in mitigating the cold start problem for new users.

- **Scalability and Efficiency:**

The implementation demonstrated scalability, allowing for real-time recommendations even in large-scale datasets.

- **Evaluation Metrics and Model Tuning:**

The choice of evaluation metrics (e.g., RMSE, MAE) played a crucial role in assessing the model's performance. Fine-tuning hyper parameters and exploring alternative algorithms could further enhance recommendation accuracy.

- **Future Directions:**

Future work could focus on incorporating advanced techniques such as matrix factorization, deep learning models, or hybrid approaches to improve recommendation quality.

CHAPTER 7

REFERENCE

- Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix Factorization Techniques for Recommender Systems. Computer, 42(8), 30-37.

- Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to Recommender Systems Handbook. In Recommender Systems Handbook (pp. 1-35). Springer.

- Resnick, P., & Varian, H. R. (1997). Recommender systems. Communications of the ACM, 40(3), 56-58.

- Netflix Prize. (n.d.). Retrieved from https://www.netflixprize.com/

- Breese, J. S., Heckerman, D., & Kadie, C. (1998). Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence (pp. 43-52).

- Scikit-learn: Machine Learning in Python. (n.d.). Retrieved from https://scikit-learn.org/stable/

- Netflix Technology Blog. (n.d.). Retrieved from https://medium.com/netflix-techblog

- Python Software Foundation. (n.d.). Python Programming Language. Retrieved from https://www.python.org/