Movie Recommendation System Using Machine Learning

Project Team

Sl. No.	Reg. No.	Student Name
1	21MPBS408003	Bindu Shree P
2	21MPBS408004	Kalash Nag

B. Sc (Hons). in Computer Science

June - 2024

FACULTY OF MATHEMATICS AND PHYSICAL SCIENCES

M. S. RAMAIAH UNIVERSITY OF APPLIED SCIENCES

Bengaluru -560 054

FACULTY OF MATHEMATICS AND PHYSICAL SCIENCES

RAMAIAH UNIVERSITY OF APPLIED SCIENCES

Certificate

This is to certify that the Project titled "Movie Recommendation system Using Machine Learning" is a bonafide work carried out in the Department of Computer Science and Engineering by Ms.Kalash Nag bearing Reg. No. 21MPBS408004, 2021-2024 batch in partial fulfilment of requirements for the award of B.Sc.(HONS) Degree in Computer Science and Engineering of M.S. Ramaiah University of Applied Sciences.

JUNE - 2024

Dr.DEEPAK AS
Professor, Dept. of CSE

Dr.SOMSHEKHARA.G
Associate professor
HOD-Mathematics & Statistics

Dr.J V DESAI
Professor and Dean-FET

FACULTY OF MATHEMATICS AND PHYSICAL SCIENCES

RAMAIAH
UNIVERSITY
OF APPLIED SCIENCES

Certificate

This is to certify that the Project titled "Movie Recommendation system Using Machine Learning " is a bonafide work carried out in the Department of Computer Science and Engineering by Ms. Bindu Shree P bearing Reg. No. 21MPBS408003, 2021-2024 batch in partial fulfilment of requirements for the award of B.Sc.(HONS) Degree in Computer Science and Engineering of M.S. Ramaiah University of Applied Sciences.

JUNE - 2024

Dr.DEEPAK AS
Professor, Dept. of CSE

Dr.SOMASHEKHARA. G
Associate professor
HOD-Mathematics and Statistics

Dr.J V DESAI
Professor and Dean-FET

Declaration

Movie Recommendation System Using Machine Learning

The project work is submitted in partial fulfilment of academic requirements for the award of B.Sc. (Hons.) Degree in the Department of Computer Science of the Faculty of Mathematics and Physical Sciences of Ramaiah University of Applied Sciences. The project report submitted herewith is a result of our work and in conformance with the guidelines on plagiarism as laid out in the University Student Handbook. All sections of the text and results which have been obtained from other sources are fully referenced. We understand that cheating and plagiarism constitute a breach of University regulations, hence this project report has been passed through plagiarism check and the report has been submitted to the supervisor.

Sl. No.	Reg. No.	Student Name	Signature
1	21MPBS408003	Bindu Shree P	
2	21MPBS408004	Kalash Nag	

Date:

Acknowledgements

We are pleased to present the **Movie Recommendation System** project and take this opportunity to express our profound gratitude to all those people who helped us in the completion of this project.

We thank our college for providing us with excellent facilities that helped us to complete and present this project. We would also like to thank Dean of CSE, Dr. J.V.Desai, Head of Department- Mathematics and Statistics Dr. Somashekara G., the staff members and lab assistants for permitting us to use computers in the lab as and when required.

We express our deepest gratitude to our project guide Dr. Deepak A S Professor of CSE for his support and valuable and timely advice during the various phases of our project. We would also like to thank him/her for providing us with all the proper facilities and support as the project coordinator. We would like to thank him/her for support, patience and faith in our capabilities and for giving us flexibility in terms of working and reporting schedules.

We would like to thank all our friends for their smiles and friendship making college life enjoyable and memorable and for family members who always stood beside us and provided the utmost important moral support. Finally, we would like to thank everyone who has helped us directly or indirectly with our project.

Abstract

With the rise of digital and social media in recent years, there are increasingly more movies and shows available to consumers. However, this much content can also cause issues for users when trying to figure out what to watch. To address this issue, ratings systems have emerged as a way to find unique content. To address this issue we are proposing a fully digital movie recommendation system based on your preferences and the features of movies.

This system employs machine learning models to learn about user behavior and movie attributes like genres, keywords, and directors names. Through the use of collaborations technologies and content filtering we aim to accurately recommend to users based on their history and to similar users. We also try to optimize by combining predictive techniques like matrix factorization with deep learning and various recommendation methods.

This system can also be used to study the users and the performance of the algorithms and different techniques used in this context, as part of the research in distribution systems. Through extensive testing and experiments we show that our solution provides not only a better user experience in terms of uniqueness and engagement when watching movies, but also contributes to the study of the users.

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Introduction

A Movie recommendation system is one type of filtering system used to predict and suggest you to watch movies according to your interest and viewing history. These systems are developed by using a wide range of datasets and machine learning algorithms to study the users' behaviors and interests, in any virtual community. In simple terms, a movie recommendation system is used to find and recommend items (e.g. :- movie) you might like.

Step 1: Data Collection: A movie recommendation system is nothing but a program that recommends similar items, in this case, movies. This is composed of details about films such as titles, genres, release years, directors, actors, user ratings, and user reviews. It can also gather user data including viewing history, movie ratings and demographic information.

2. Data Pre-Processing: After that data has been collected, it needs to be pre-processed. Clean the data, missing value operations and conversion of data to a data format that we do the job with it. This includes techniques such as changing categorical data into numerical values (such as genres) or normalization and standardization.

3. Feature Engineering: Feature engineering is the process of choosing, creating, and transforming variables to create new features from the raw data. For instance, in the context of a movie recommendation system, these steps can consist in generating user profiles from the movies the person has already liked, extracting features from movie descriptions or reviews or merging different data sources in order to improve the recommendation process.

4. Choosing an algorithm could be the last and most important thing to do when building a movie recommendation system. You can choose between content-based filtering, collaborative filtering, or any hybrid way. Collaborative filtering techniques go through customers' actions and consequently propose products to other customers (users) who resemble one another by means of such behavior as related items. On the contrary, content-based filtering recommends things because of their attributes as well as how similar they are to what the users like. These hybrid recommenders provide higher accuracy with the best features of both worlds.

1.2 Literature Survey

Table 1 Literature survey

sl. no.	Author	journal name and year of publicati on	research focus	finding in research	conclusion derived via authors	limitations in the study	conclusi on of published work
1.	Yehuda Koren, Robert Bell, and Chris Volinsky	"Matrix Factorization Techniques for Recommender Systems"(2009)	The paper introduces matrix factorization techniques for collabrative filtering which are mostly used for recommendation system	the research tells about the usefulness of factorization on models collabrative filtering commonly used for predicting user prefernces. it showcase the imact of model complexity and includes factors like biases,implicit feedback	the authors conclude that factorization methods along with factors like biases, implicit feedbacks and temporal dynamics modelling are very useful in collaborative filtering	the foremost significnt confinement within the think about is the relaince and dependibility on a single dataset which may need differing qualities client inclinations and beahviours on diverse stages or spaces	he conclusion we get from the think about is that like Netflix prize exhibits the exceptional accuracy and productivity of these methods.

				s, temporal dynamics and varying confidence leves to improve prediction accuracy			
2.	Yehuda Koren, Robert Bell, and Chris Volinsky	"Factorization Meets the Neighborhood: a Multifaceted Collaborative Filtering Model"	The **investigate centers** on **upgrading** recommender **frameworks**, **especially** Collaborative **Sifting** (CF) approaches, **to supply** more **exact** and personalized **proposals** to **clients**. It **presents advancements** to both **inactive figure** models and neighborhood models, **pointing**	the study comes up with suggestion to enhance collabrative filtering by optimizing neighborhood base models and including implicit feedbacks to increase the accuracy and prediction	the conclusion that the authors get is that the importance of taking in consideration of all the aspects that can help to improve the quality by the mentioned factors	the limitation we find is that the relaiance on the factors are high which can be sometime misleading and the accuracy can be damaged	the study garnered a lot of enhancement in the collabrative filtering field by including both optimizing neighborhood based and latent factor approaches

			to **combine** them **consistently** for **progressed execution**.				
3.	Hansaim Lim Lei Xie	Target Gene Prediction of Transcription Factor Using a New Neighborhood-regularized Tri-factorization One-class Collaborative Filtering Algorithm	Your research focus appears to be on computational methods for predicting target genes of transcription factors (TFs) and understanding gene regulatory networks.	The accuracy rate of prediction reached 37.8% thanks to this algorith	The research has proved that this new algorithm has improved the accuracy of prediction significantly	Ofcourse with new algorithm there has to be limitations like this is based on input data and factors like experimental metrices.	The conclusion from the published work has to be that this algorith helps very much in accuracy of prediction

1.3 Conclusion

In this chapter, we discussed about three papers we had chosen to research and survey for our project. We read through their finding and the issues they add while working on the recommendation systems. We also looked at other similar applications in the same field and see how and what we offer would work better. Keeping in mind these findings and issues, we went on to make our implementation of the recommendation system.

2. Background Theory

2.1 Pycharm

PyCharm is a specialized integrated development environment (IDE) developed specifically for Python programming language. It comes with several features and tools that make development process easier and smooth working with python codes. PyCharm in this sense is a useful tool to develop, test and deploy machine learning models for a movie recommendation system using datasets.

1. Editing Codes: PyCharm has a strong code editor that highlights syntax errors, autocompletes codes and also refactors variables. Writing python code becomes more uncomplicated when it covers data preprocessing, feature engineering, algorithm definition and model evaluation.

2. Project Management: Code organization is made easier by project management tools which are provided by PyCharma. To create and manage Python projects, virtual environments and package dependencies using such tools as Project Explorer, integrated package manager would be a wise step.

3. Debugging: In building machine learning models, one cannot do without debugging. This is where PyCharm comes in handy since it has an integrated debugger that allows stepping through code, inspecting variables which are used to identify and fix errors in algorithms.

4. Version Control: Git version control system is part of the PyCharm’s capabilities to help you keep track of code changes, work with teammates and trace back revisions in ML codes and models

5. Data Visualization: Though PyCharm doesn’t have its own powerful data visualization libraries, it can be easily integrated with such commonly used ones as Matplotlib or Seaborn for presenting dataset(s), model performance scores as well as recommendations’ outcomes.

6. Model Deployment: When a machine learning model for movie recommendations has been trained and tested, PyCharm can aid in deploying this model to production environments (which may also involve integration of the model into web applications, APIs or any other systems that should provide real-time recommendations for users).

2.2 Jupyter Notebook

On the other hand Jupyter Notebook refers to an interactive computing environment that is designed to accommodate live code snippets, visualizations plus text based multimedia content being used by computer programmers all over the world.

Different programming language support: it supports a variety of programming languages like Python making it well known for data analysis, machine learning, and plotting graphs.

In the case of datasets, this is very effective when used in the real-world movie recommendation system in Jupyter notebook where you can able to analyze data, build model and plot your data to make a clear understanding about your model.

1. Interactive Data Analysis (Jupyter Notebook): this is an environment where you can write python code in cells and execute it to see the result. Its quite a cool thing because all this interactive access allows us to dive into the data in an adhoc way, like cleaner the movie csv or the preprocessing of the data or even to get some exploratory data analysis (EDA) so we can get a better idea of what is actually going on.

2. Machine Learning Model Development: It also supports the inclusion of machine learning libraries such as scikit-learn, TensorFlow, pyTorch, in building and training of machine learning models to predict a movie to be recommended to the user. Numerous recommendations like collaborative filtering, content-based filtering, and hybrid systems can be coded right from the recommendation library in Jupyter Notebook..

3. Visualization: However, as a web application, Jupyter Notebook can easily interface with data visualization tools like Matplotlib, Seaborn, and Plotly. This helps you to generate basic and graphical undertaking like histogram, scatter plot, Heat map, Interactive charts for the visualization of populating movie data, measurement of the model outcomes and results of recommendation systems.

4. Documentation and Collaboration: Jupyter Notebook enables you to create a report, notebook in which the script, the commentary, and visuals are combined making it possible to explain all the actions performed, enhance the legibility of the code and share the results with the team members. It also fosters collaboration, since there is an ability for a team of users to simultaneously work on the same notebook, not only with the help of Jupyter Hub, but with Jupyter Lab too.

5. Experimentation and Prototyping: Jupyter Notebook stands out as it allows for dynamism as in a short amount of time one can experiment with different machine learning

algorithms, tune hyper parameters, and even try different data pre-processing methods without having to switch between numerous applications.

6. Another advantage of each code cell is that it can be rerun, which will allow improving the recommendation system online as long as the model can be improved and modified. Integration with External Tools: In terms of its ability to interact with other tools and services, they include Google Colab, Microsoft Azure Notebooks, and other programming tools like Git to access other enhanced features for data science and collective development.

2.3 Python Programming Language

PYTHON A language employed for web development and is an interpreted and object-oriented language which actually cannot be termed as complex and even a beginner can understand how to use it in a program. Originally created by Guido van Rossum in 1991 during his work at National Research Institute for Mathematics and Computer Science in the Netherlands, Python is one of the most famous and, probably, one of the best programming languages of the present days because of its efficiency and easy-to-use nature. Moreover, Python allows procedural, object-oriented and functional, and other types of programming paradigms and thus it is suitable for different purposes.

Python is also highly suitable for building movie recommendation systems because it is equipped with numerous libraries and frameworks, which make it easy to incorporate data analysis, machine learning, and web development into the application.

2. 4 Streamlit

As for Streamlit, it is the open-source web app framework primarily for creating the data applications in Python, easily deployment. The next is the knowledge about how to create interfaces for constructing conversations with web applications while using techniques of machine learning and data science. Here's an overview of Streamlit and its role in a movie recommendation system: Below, I give an idea on what Streamlit is and how the movie recommendation system could use it:

1. Ease of Use: Streamlit is used with the purpose of creating real-time machine learning web applications, where one can transform their data script into a web application in under a few minutes. Thus, it only requires a little leg work which essentially is part of the front end web development skill sets that would make it acceptable to both the data scientist and the machine learning engineers.

2. Pythonic: As far as I understand, Streamlit applications run in Python, but do not require any other programming languages. It is possible to integrate this program without any prior involvement of HTML, CSS, or JavaScript knowledge or experience.

3. Interactive Widgets: It is another module for constructing engaging and feature-rich apps, encoded with attributes such as a slider, buttons, and text fields.

4. Real-time Updates: This of particular when it comes to changes as the changes made in the code are reflected in the running app immediately and this eliminates the need to refresh the browser hence helpful in prototyping and testing the app to single out bugs.

5. Deployment: The streamlit apps can be run on any cloud services or even within an organization context within a few steps.

A movie recommendation system using Streamlit typically involves the following steps

1. Data Preparation: Gather movie data, clean and prepare it for later use and analysis. This could include features and details such as the movie itself, the ratings provided by users and all other related factors.

2. Model Building: Develop methods for making recommendations based on collaborative filtering, content filtering, or both. To test the model, use the dataset that you have prepared beforehand.

3. Streamlit Application: Develop a Streamlit application that will serve as a front-end for the recommendation model. The app normally consist of user input areas to collect data of users and show them movies of their kinds.

4. Interactivity and Customization: As the additional suggestions add more functions try adding sliders to enable the user to filter recommendation by genre, rating and much more.

5. Deployment: Host the app using Streamlit’s cloud or find other hosting services such as Heroku or AWS.

2. 5 Pandas

It is designed to handle large amounts of data quickly and efficiently and can be used in data analysis, data pre-processing, data wrangling, etc. It is widely used in machine learning and data analysis and processing various kinds of data science projects. With regards to the data, handling and preparation for the creation of a movie recommendation system, Pandas fulfils a vital role in data scrubbing, engineering and munging.

1. Data collection and loading

One of its uses is to read movie and rating data from files, such as CSV or databases.

2. Data Cleaning and Pre-processing

Some of the important operations performed by Pancas pertaining to cleaning processes them are that they help users deal with missing values and duplicates, and also in changing data types.

3. Data Exploration and Analysis

Pandas was employed to analyze this data to compare movies and overall ratings, as well as review the distribution of each.

4. Feature Engineering

Pandas is also used to create new features that can improve the recommendation system, like when there are genres in the data, when there is an average rating and other features.

5. Merging Data

Activity 1 and 2 & Union between movie and ratings data: The various datasets involved are converted into one dataset for subsequent analysis and modeling.

6. Model Evaluation

Assessment of the recommendation system on different dimensions about the test data set and the user data set, RMSE, precision, recall, etc should be calculated.

7. Visualization

Libraries such as Pandas combined with Matplotlib or Seaborn do data analysis of the trend, pattern, and results of a statistical model which can be visualized.

In the movie recommendation system, pandas is indispensable that you can manipulate and transform these datasets almost as significantly as you do with variables.

The Scorpio was a body on frame constructed SUV which meant it had heavy ladder frame under its carriage giving it the ability to bear significant loads and antiquated although rugged solid back axle with leaf spring suspension setup at rear end.

2.6 Merits and Demerits

Merits:

Just like the AI-generated languages, this technique has its own pros and cons. It is easy to point at merits of human written texts since the text is created by a human brain.

1. Personalization:

- Tailored Recommendations: After an offer has been made it can be specific to the users of online systems based on their observed behaviour. This allows, for example, a simple way to directly improve the user experience.

- User Engagement: They also help in improving user engagement and satisfaction by presenting the content that is most probable to be of interest to these users.

- Genre specific: This guidance provides users with the films that fit the bill which will save time for decision making instead of looking at many other propositions.

- Scalability: Those systems which are scalable can handle huge data and serve a large number of people at once.

2. Data Utilization:

- Insight Generation: These systems are capable of analysing huge volume of data to discover regularities and trends which are not directly visible.

- Only does that but also indicates how the machine is better than human beings is better - Discovery of Relevant Content: They improve the discovery of relevant content for users who would otherwise have not found it.

- Boosting sales and rentals can be attributed to personalized recommendations for platforms that sell or rent movies. - Employee retention rates which are the result of keeping your workforce engaged with relevant learning opportunity, can be improved by these systems.

Demerits:

1. Data Dependency:

- Data Quality: Whether the results are precise or not mainly it is based on the richness and accuracy of the data available. Bad data or incomplete data can result to less accurate recommendations.

- Privacy Obstacles: The utilization of personal data in recommendation may pose privacy concerns and hence necessitate implementation of strict data protection regimes.

2. Complexity:

- Algorithmic Bias: It is possible that the algorithms, instead, may suffer from bias and reflect this discrimination in their recommendations, what causes these to be less diverse. - Over fitting: If the system is too closely tailored to specific user behaviours, it may fail to generalize well to new or less common user preferences.

3. Cold Start Problem:

New Users/Items When system comes across a new user who does not have any history or a new movie which does not have any ratings, it will be challenging to get proper recommendations.

4. Implementation and Maintenance:

- On the Resource side: Building and keeping in good shape an advanced recommendation system could drain resources, it could be computationally as well as human resource demanding task.

- In terms of machine learning algorithm, it is hard to trade-off between the accuracy with the complexity of devised model which almost impossible in case of the traditional approach.

A story called Stones by Timothy Findley combines multiple themes and aspects that interlink with each other throughout the story.

2.7 The Conclusion

The background proposition that supports this operation has been discussed in the previous section. The main points of the operation with respect to other operations are made. The mounds and dependencies as far as this operation concerned were enumerated. / The advantages, and disadvantages position you will find.

3. Aim and Objectives

In this section, the actual dissection of project is done and each module is built piece by piece in order to complete the project. In design section, the application has been built in accordance with functional requirements based on which diagrams like Use Case and Low Level Sequence Diagram is drawn. In implementation section, snips of important code is displayed with their explanation given below. In testing section, all functional requirements are tested and the result is analysed resulting in status of test condition.

This chapter serves as the compass for our project, delineating the overarching goal and specific objectives that chart our course. It begins with a clear articulation of the project's title and aim, providing a concise yet comprehensive overview. Delving deeper, we outline the objectives essential for the project's fulfilment, each playing a crucial role in achieving our desired outcome. The section meticulously documents functional requirements, offering a detailed perspective through the lens of various diagrams. Methodologies are presented to articulate the procedural steps guiding the fulfilment of these objectives. As the chapter concludes, a succinct summary ties together the aim, objectives, and methodologies, setting the stage for a structured and purposeful project execution.

3.1 Title

Movie Recommendation System Using Machine Learning

3.2 Aim

To provide personalized movie suggestions to users based on their preferences and past behaviour.

3.3 Objectives

The following are the goals of the project:

1. An overall survey will be conducted on the existing Movie recommendation systems which are based on datasets and their methods and methodologies.

2. The requirements for the implementation will be formulated by conducting the survey.

3. To design the application which is based on the requirements identified. Genre: instructional

4. Implement the design with the help of pycharm, streamlit and python.

5. Test and validate the developed application.

6. To draft project report of the project using the format provided by the university.

3.4 Methods and Methodology

Table 2 Methods and Methodology

Objective No.	Statement of the Objective	Method/Methodology	Resources Utilised
1	To conduct a literature survey on the existing movie recommendation systems from datasets and its methods and methodologies.	1.1 Literature review on website development and referring to related published papers with genuine citations and similar existing applications. 1.2 Using the literature research, the necessary requirements would be collected and documented.	Pycharm documentation and YouTube videos.
2	To formulate the requirements for the implementation based on the survey.	2.1 Functional and non-functional requirements are derived along with their dependencies.	Software development fundamentals textbooks and our guide.
3	To design the application based on the identified requirements.	3.1 The functional requirements will be used to create high level UML diagrams such as use case, class, activity and sequence diagrams. 3.2 To detail the design, a low level diagram (Flowchart) will be used.	DIA
4	To actualize the arrange utilizing pycharm, streamlit and python.	4.1 To arrange the application utilizing pycharm, streamlit and python...	Pycharm documentation and YouTube recordings. To test and endorse

			the made application.
5	To test and validate the developed application.	5.1 Test the application in understanding with prerequisites.	
6	To document the project report based on university template.	6.1 Create a specialized report based on the determinations.. 6.2 Illustrate the application some time recently the board.	Word for archives, PowerPoint introductions and web assets..

3.5 Conclusion

This particular chapter focused on defining the title and Aim of the project correctly. Later this chapter included the required objectives that were required to be fulfilled to complete this project. Method and methodologies are documented successfully in this section and is successfully tabulated in order to know the steps used in completing the objectives including resources used.

4. Problem Solving

4.1 Project Concept

- The project is based on ideas of movies in terms of directors, cast and production team.

•You can search for any specific movie you want to.

Features:

- Personalized Recommendations: This is what the system offers to the users when it comes to personalized movie and it is based on their individual preferences style of music.

- Usage of TMDB 5000 Movies Dataset: The dataset comprising 5000 movies on TMDB will flock you with a plethora of details about movies e.g. titles, overviews, genres, keywords, the cast and the crew.

- Context-Based Filtering: For recommendation system the content of an item is analysed by our recommendation system which is then used to find the most similar items. Our recommendation system implements this approach by analysing movie descriptors, genres and keywords and makes suggestions of related items.

- Data Pre-processing: Data pre-processing covers purging the dataset by discarding all rows with missing values, feature engineering that harvests the important features, and text processing that makes the text data into a numeric form so that it can be used by machine learning algorithms.

- Calculation of Similarity: Cosine Similarity helps computing similarity of two movies by using their feature representations to know how similar they are.

- Interface of Website User Friendly: The recommendation system is fused into an interactive web interface which employs Streamlit ‘Streamlit’, consequently, users can interact with the system easily by selecting a movie from a dropdown menu.

- Model Evaluation: When it comes to model evaluation, the techniques used are precision, recall and user satisfaction surveys which help us to understand effectiveness of the recommendation systems.

- Server Deployment: When the development and testing are over, the website with its recommendation system will be put on a public server. This deployment will later be done on a server that the overall server infrastructure is then optimized for performance, scalability, and reliability.

Improved User Experience: System by implicating personalized movie recommendations on the basis of a user's previous watch history will help to raise the overall movie-watching satisfaction.

4.2 Design

- Functional and Non-Functional requirements
- Diagrams
  - ➢ Flowchart
  - ➢ Class Diagram
  - ➢ Sequence Diagram
  - ➢ Use Case Diagram

4.2.1 Functional Requirements

FR 1. Clients can enlist for an account. They have to be have secure get to to urge their claim motion picture offer.

FR 2. Clients can seek for motion pictures by title. Other strategies or parameters ought to too be considered. Choose a motion picture they like.

FR 3. The framework ought to give individualized film proposals. Suggestions ought to be based on client inclinations and seeing history. Perhaps indeed depending on the rating.

FR 4. The framework ought to utilize content-based sifting strategies to create film proposals. There are a few things to consider. These incorporate motion picture portrayals, catchphrases, and analogies.

FR 5. Clients ought to be able to grant criticism on prescribed motion pictures. Like it or not by their evaluations. This input will assist refine the proposal prepare.

FR 6. Clients ought to be able to oversee their possess profiles. Inclinations and seeing history ought to be upgraded. A bookkeeping framework must be kept up.

FR 7. The proposal framework ought to be consistently coordinates in a user-friendly site. This permits clients to connect with it rapidly.

FR 8. The site interface ought to be responsive. This guarantees the leading conceivable client involvement on all gadgets and screen sizes.

FR 9. The framework must run on a server optimized for execution. It should be flexible. It must be reliable and responsive to the user's demands.

FR 10. Frameworks should be completely tried to guarantee that they are working. Make beyond any doubt the proposals are precise and secure.

4.2.2 Non-functional prerequisites:

NFR1. Framework must react to client demands for proposals and suggestions in an opportune way. More often than not inside seconds.

NFR2. Web interface ought to stack rapidly and the littlest.

NFR3. The proposal calculation ought to scale effectively as the number of clients and motion pictures within the database develops.

NFR4. The site interface ought to be natural and simple to explore, with clear enlightening for utilizing the suggestion framework.

NFR5. Clients ought to be able to get it and translate the prescribed motion pictures effortlessly, counting the reasons behind each proposal.

NFR6. The framework ought to give input to clients amid the proposal prepare, such as advance markers or stacking activities.

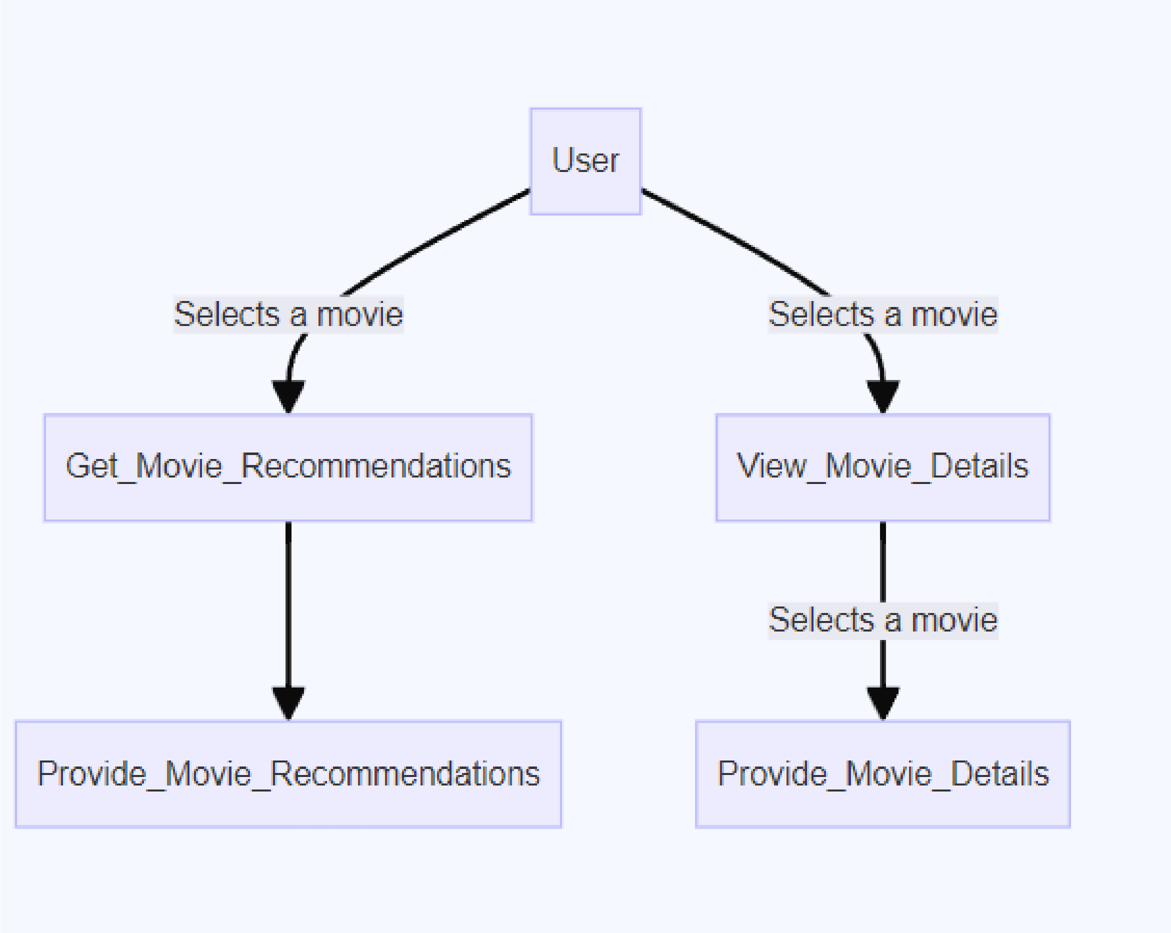
NFR7. Client information, counting individual data and seeing history, ought to be put away safely and secured from unauthorized get to or information breaches.

NFR8. The framework ought to actualize suitable confirmation and authorization instruments to guarantee that as it were authorized clients can get to their accounts and suggestions.

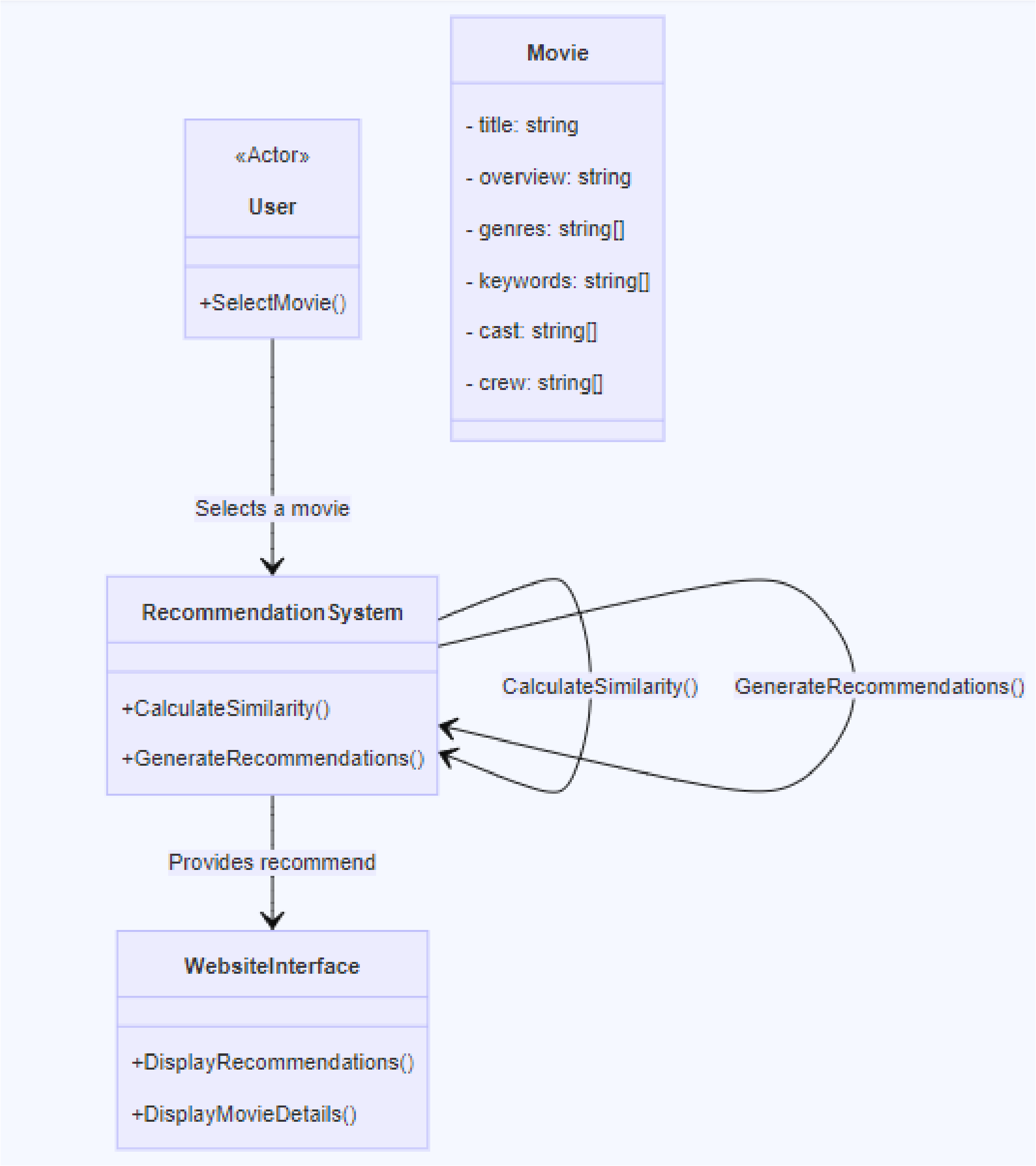
NFR9. Any external APIs or data sources used by the system should be authenticated and encrypted to prevent data interception or tampering.

NFR10. The system should be designed with modular components and well-documented code to facilitate future updates, enhancements, and maintenance tasks.

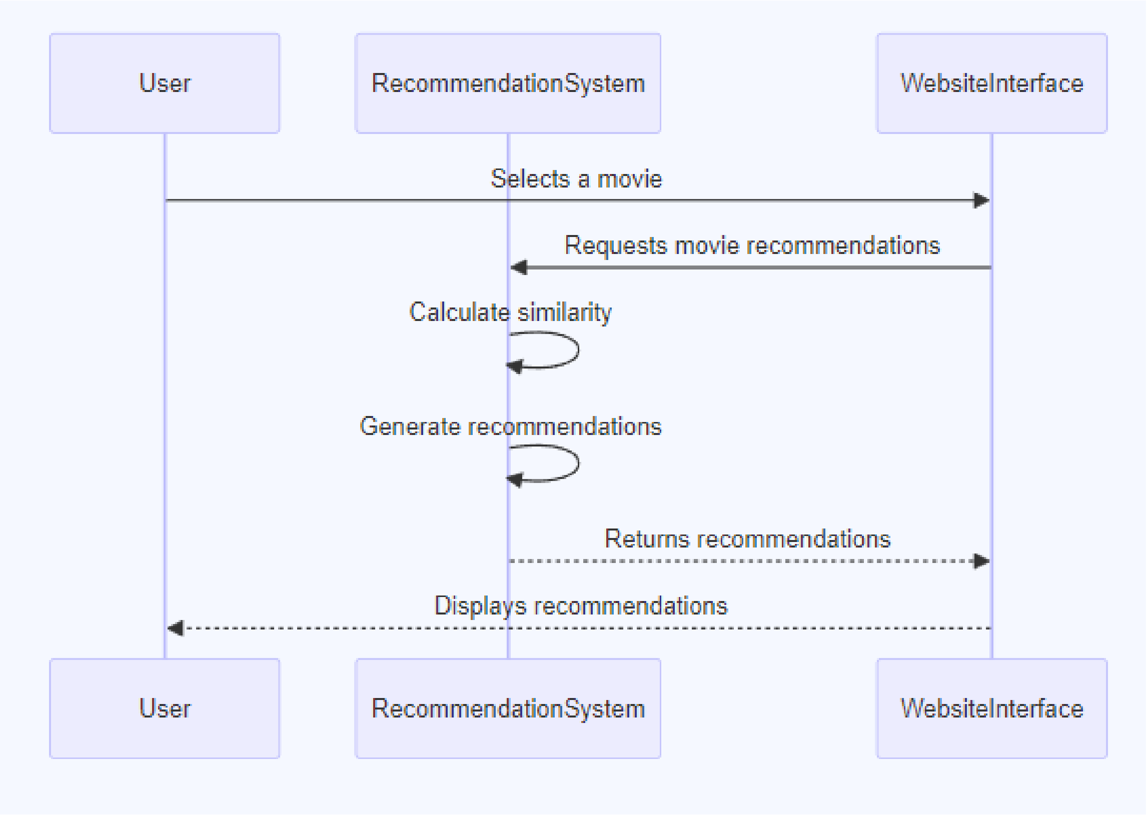
- **Diagram**



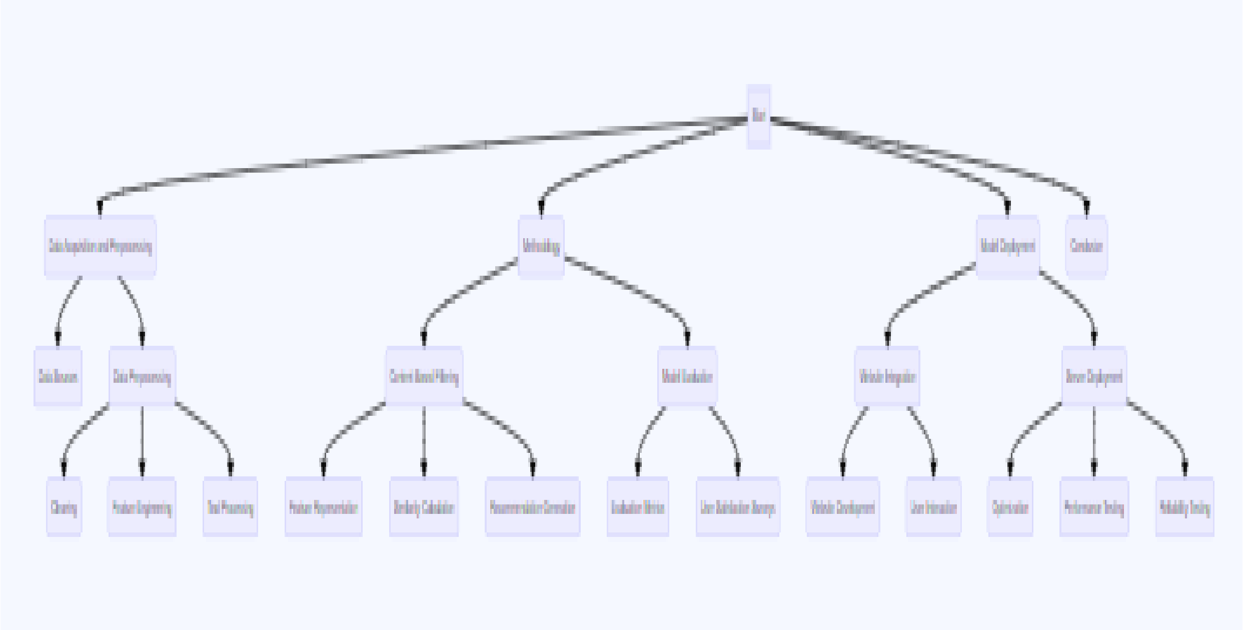
Flowchart diagram



Class diagram



Sequence diagram



Use case diagram

4.3 Implementation

4.3.1 Project Implementation Technology

The Extend application is actualized in PyCharm. We utilized PyCharm for the Plan and coding of the extend. Made and kept up all databases in jupyter note pad, in that we did pre-processing of information, manipulation and vectorisation.

I. Equipment Necessity

- Portable workstation or PC
- MacOS Sierra and over
- Windows 7 or higher
- I3 processor framework or higher
- 8 GB Slam or higher
- 100 GB ROM or higher

II. Computer program Prerequisite
i. Tablet or PC

- PyCharm with streamlit introduced
- Jupyter scratch pad

4.4 Project Structure

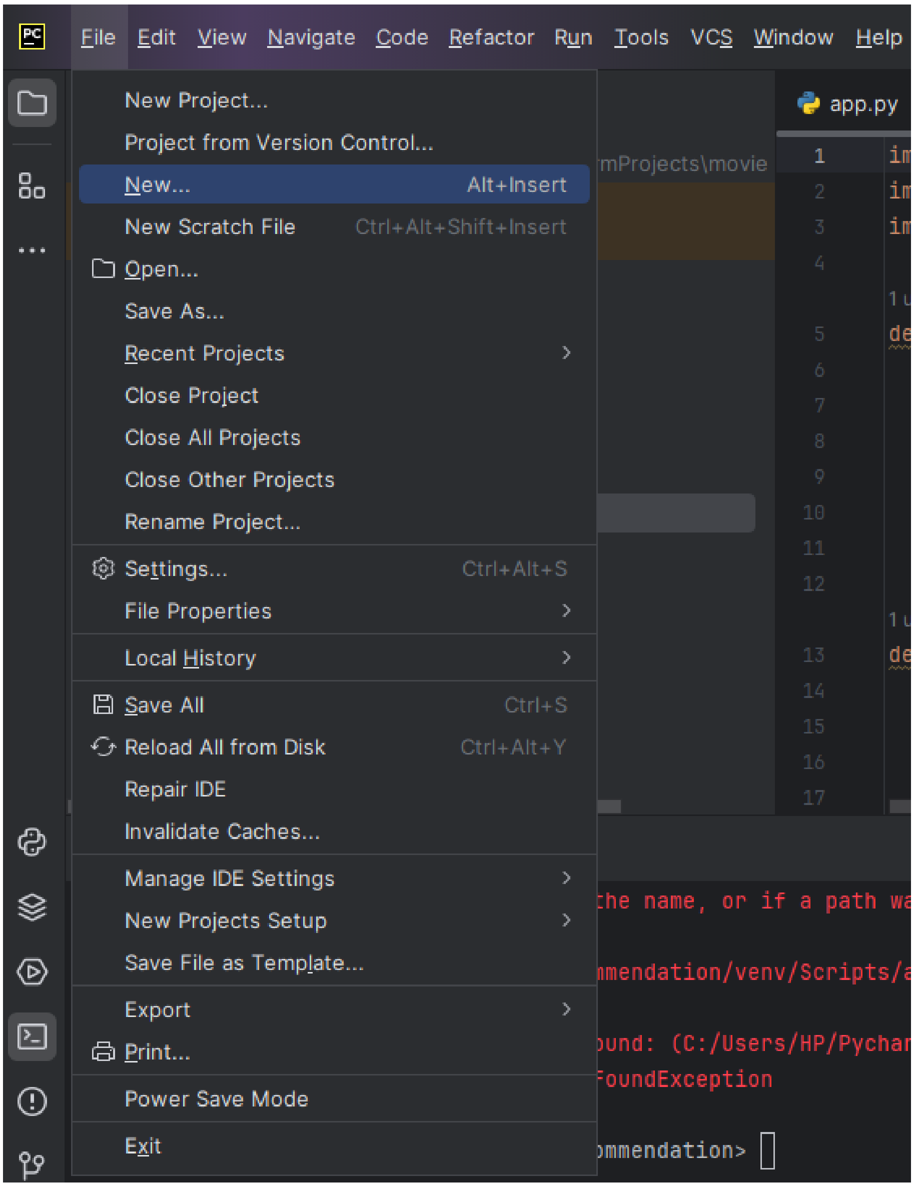
- The **expand** in PyCharm comprises of modules such as Android app modules, Library modules, and Google App **Engine** modules.

Each module contains source code records (.java), **resource** records (e.g., XML **designs**, UI strings, bitmap pictures), and **develop** records (**underneath** Gradle Scripts).
Project View:

- PyCharm appears wander records inside the Wander gadget window, organizing modules and records for fast get to.
- Clients can switch between unmistakable sees, such as Android wander see or Expand see, to see the genuine record structure of the wander.

.

User Interface Components:

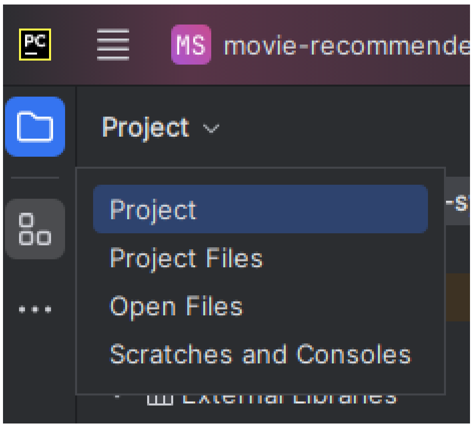


```
the name, or if a path wa

mmendation/venv/Scripts/a

ound: (C:/Users/HP/Pychar
FoundException

ommendation>
```

- Toolbar: Offers **exercises** like running the application, **exploring**, and getting to PyCharm **devices**
- Navigation Bar: **Makes a difference** in **investigating** through the **expand** and opening records for **changing**.
- Editor Window Where engineers make and **alter** code, with context-aware highlights based on record sort.
- Tool Window Bar: Surrounds the IDE window and contains buttons to expand or collapse individual tool windows.
- Apparatus Window Bar:
  Encompasses the IDE window and contains buttons to extend or collapse person apparatus windows.

- Device Windows:
  Give get to to particular errands like extend administration, look, form control, and more.

- Status Bar:
  Shows extend and IDE status, as well as any notices or messages.



Apparatus Windows:

• PyCharm moreover permits you to customize the format of apparatus windows to optimize screen space.

• You'll be able appear or cover up apparatus windows utilizing the See menu or alternate routes.

• To reestablish the default format or customize it, you'll explore through the Window menu choices. Navigation:

• PyCharm offers different route easy routes and activities to assist you move around the IDE productively.

• Utilize alternate routes like Ctrl + E (Command + E on Mac) for Later Records, Ctrl + F12 (Command + F12 on Mac) for Record Structure, and Ctrl + Shift + N (Command + Move + N on Mac) for Explore to Record.

• You'll explore to classes, strategies, areas, and symbols utilizing committed route activities and alternate routes.

Gradle Construct Framework:

• Whereas PyCharm fundamentally centers on Python advancement, you'll still oversee construct arrangements for ventures.

• For Python ventures, you'll regularly bargain with Python bundle administration devices like pip and virtual situations rather than Gradle.

• PyCharm bolsters integration with different construct frameworks and devices commonly utilized in Python advancement, such as setuptools and pip.

Execution Screens:

• PyCharm gives apparatuses for profiling and analyzing Python code execution.

• You'll screen CPU and memory utilization, identify memory spills, and optimize code execution utilizing built-in apparatuses.

• Investigate the Profiler and Memory Profiler apparatuses in PyCharm to analyze your application's execution.

Assignment Tracker:

• PyCharm offers profiling devices and memory utilization examination that can offer assistance distinguish memory assignment issues in Python code.
• You'll utilize the built-in profiler to screen memory allotment and utilization amid program execution.

• By analysing memory assignment designs, you'll optimize your code for superior execution and memory utilization

Code Inspections:

- PyCharm performs code inspections automatically as you write and compile your Python code.
- The inspections help identify structural issues, potential bugs, and optimization opportunities.
- PyCharm integrates with various static analysis tools and linters, including Pylint, Flake8, and mypy, to provide comprehensive code analysis.
- Assessments cover ranges such as code rightness, fashion adherence, execution, and code smells.

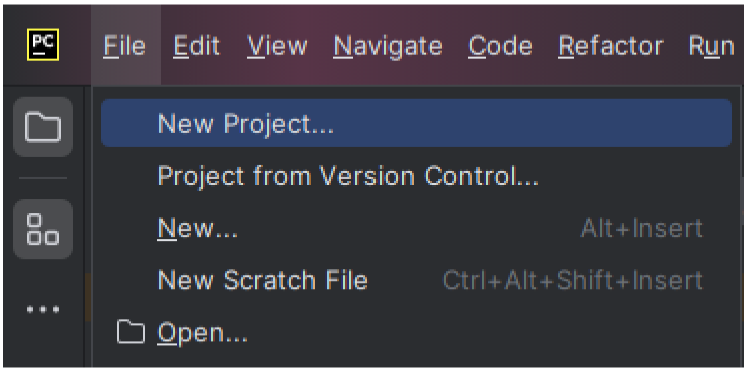
Introducing Streamlit:

Start PyCharm:
Open PyCharm on your framework.
Open Plugin Inclinations:
Go to Settings/Preferences. You'll usually discover this within the "Record" menu.
Explore to Plugins:
Within the Settings/Preferences window, discover the "Plugins" area.
Hunt for Streamlit:
Within the Plugins Commercial center, look for "Streamlit".
Introduce Streamlit Plugin:
Once you discover the Streamlit plugin, press on it, at that point press "Introduce".
Introduce Conditions:
In case incited, introduce any essential conditions or extra plugins that Streamlit requires.
Restart PyCharm:
After establishment is total, restart PyCharm to actuate the Streamlit plugin.

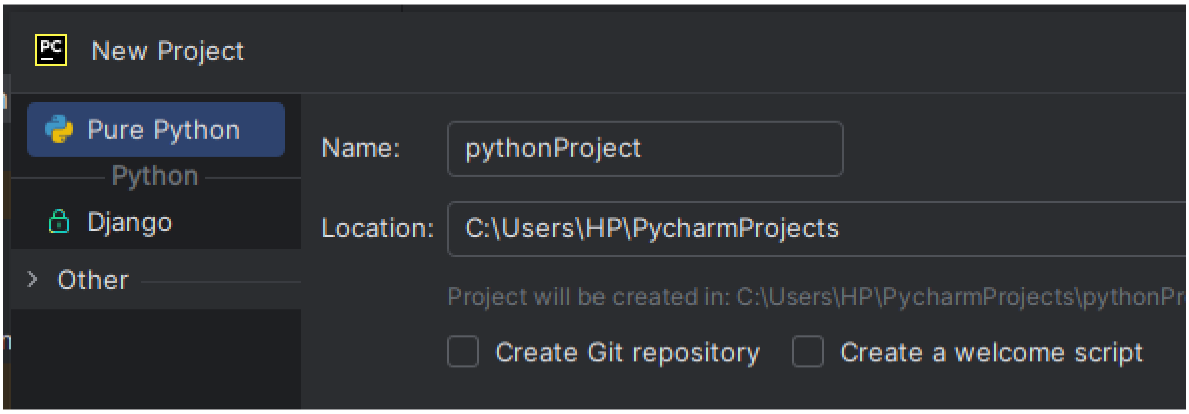
4.4.1 Making the application

Step 1:

Open the Coordinates Improvement Environment and select Begin a modern venture



Step 2: Select the python project as your project type. Then click next.



Name: pythonProject

Location: C:\Users\HP\PycharmProjects

Project will be created in: C:\Users\HP\PycharmProjects\pythonPr

□ Create Git repository □ Create a welcome script

Step 3: Enter project name, then chose the location for it to be saved

Step4: choose virtual environment as the environment

Step5: Click finish

Running the project

Step1: under project make files like app.py.

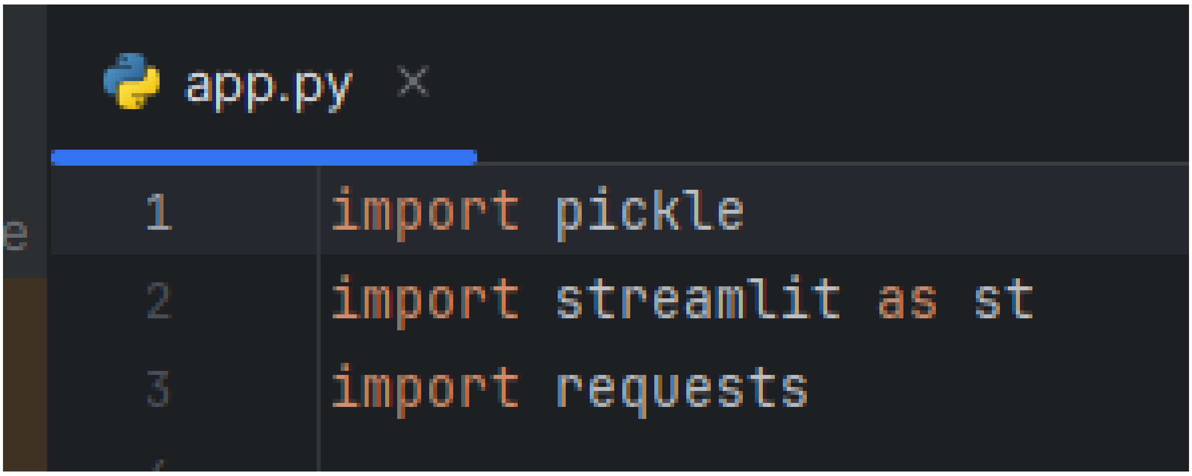
Step2: then upload other important files like movie_list.pkl and similarity_list.pkl

Step4: Start writing the code

5. CODING

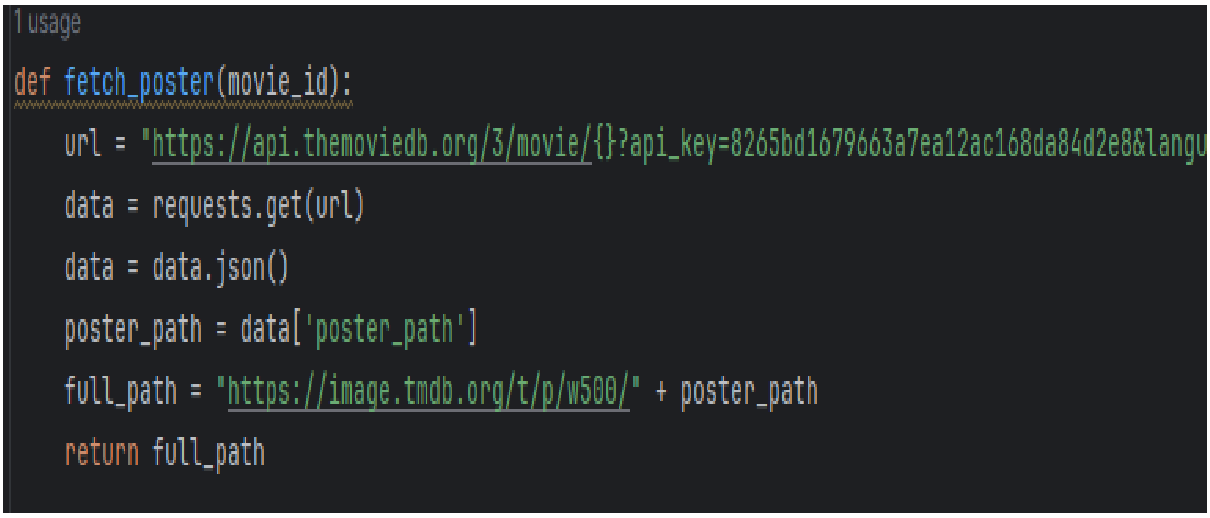
5. CODING

5.1 Code for importing



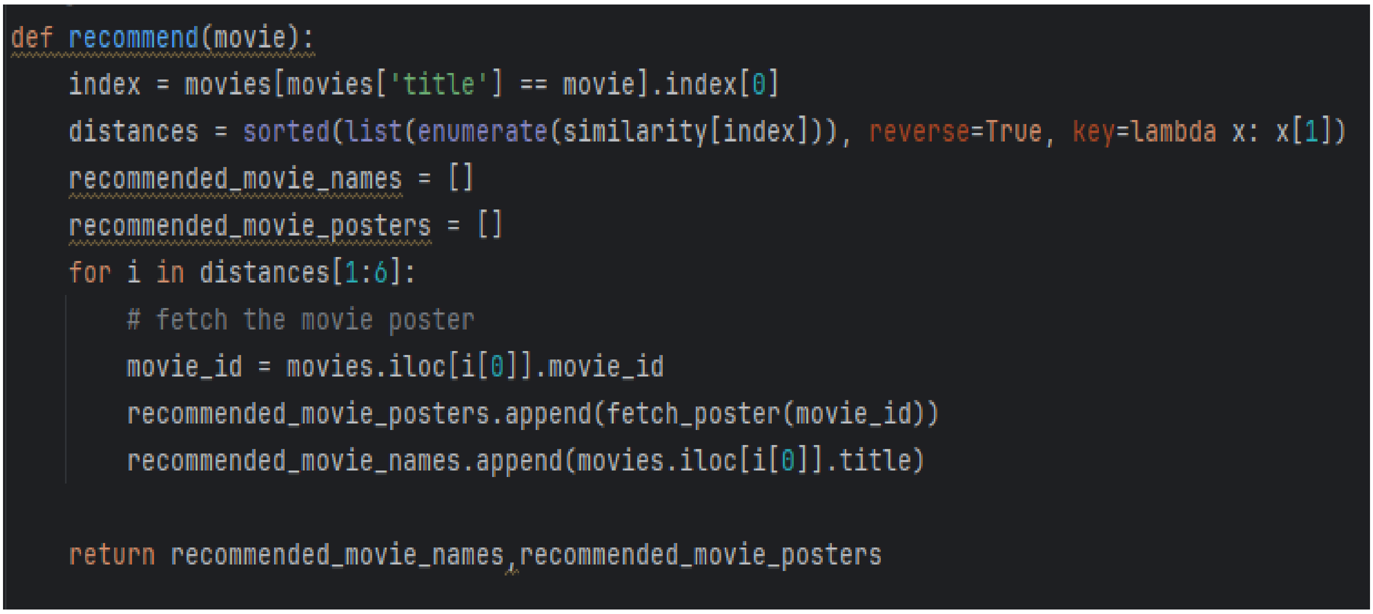
```python
import pickle
import streamlit as st
import requests
```

5. 2 Code for Fetch Poster Function



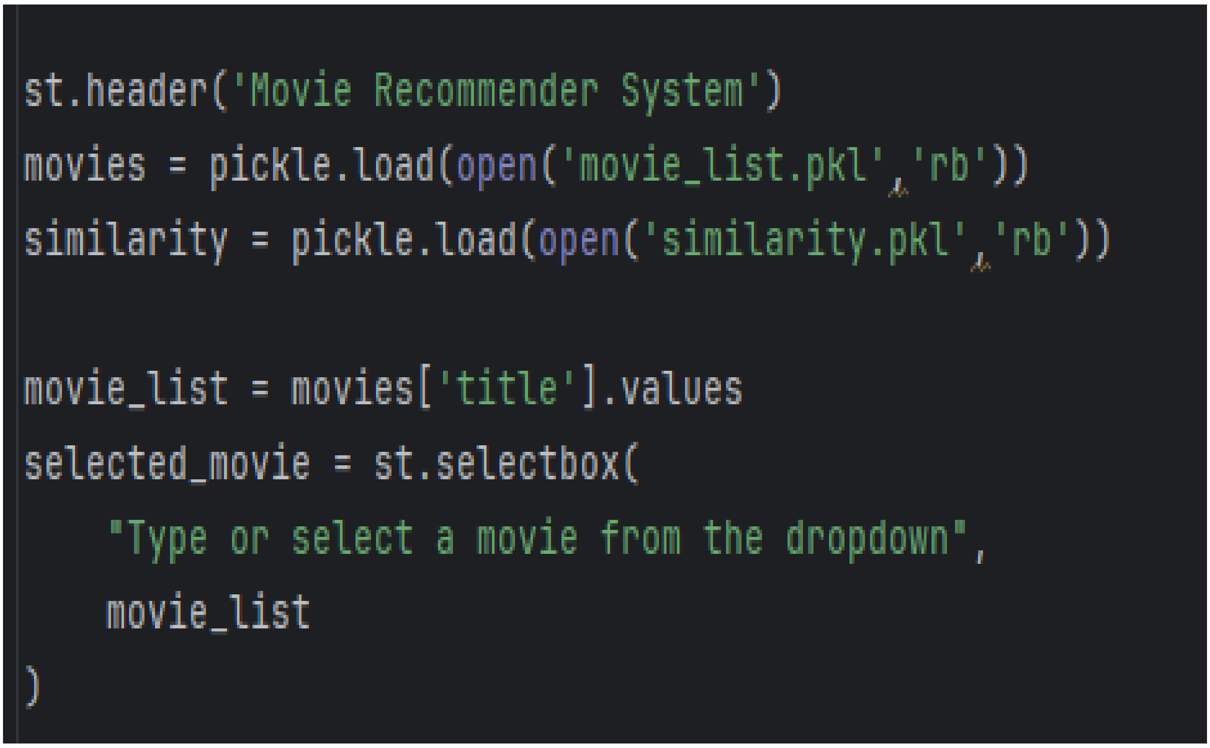
```python
def fetch_poster(movie_id):
    url = "https://api.themoviedb.org/3/movie/{}?api_key=8265bd1679663a7ea12ac168da84d2e8&langu
    data = requests.get(url)
    data = data.json()
    poster_path = data['poster_path']
    full_path = "https://image.tmdb.org/t/p/w500/" + poster_path
    return full_path
```

5. 3 Code for Recommend Function



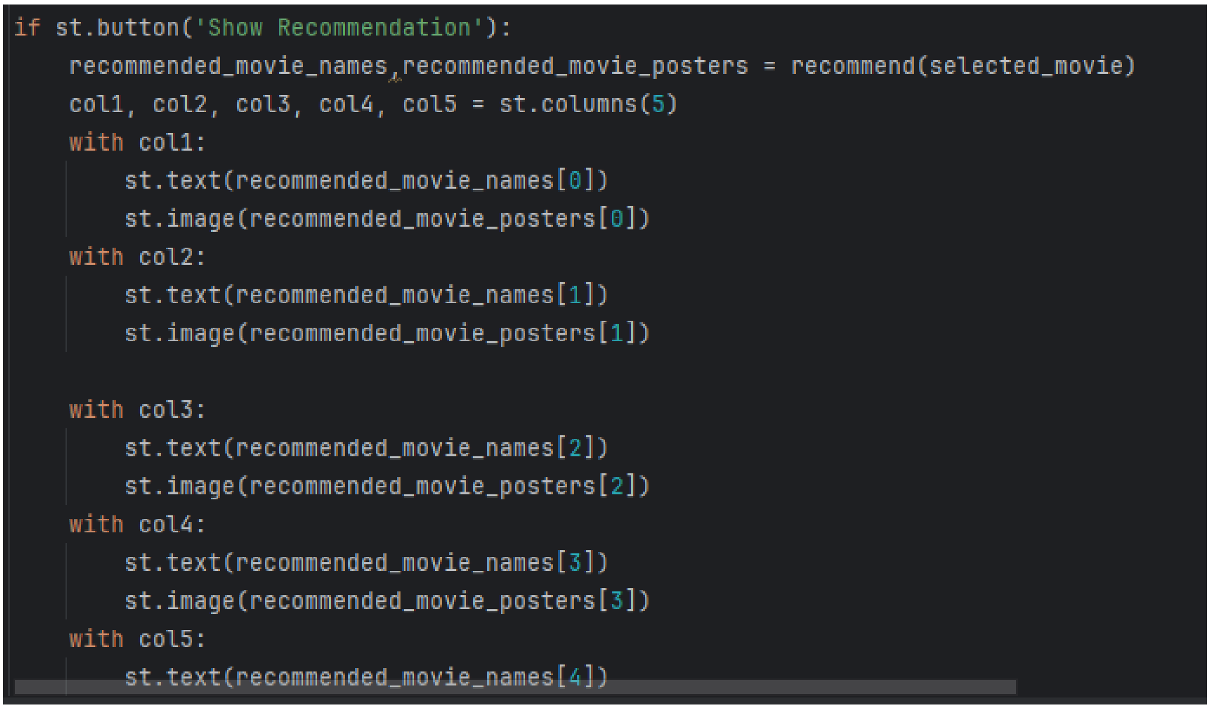
```
def recommend(movie):
    index = movies[movies['title'] == movie].index[0]
    distances = sorted(list(enumerate(similarity[index])), reverse=True, key=lambda x: x[1])
    recommended_movie_names = []
    recommended_movie_posters = []
    for i in distances[1:6]:
        # fetch the movie poster
        movie_id = movies.iloc[i[0]].movie_id
        recommended_movie_posters.append(fetch_poster(movie_id))
        recommended_movie_names.append(movies.iloc[i[0]].title)

    return recommended_movie_names,recommended_movie_posters
```

5. 4 Code for Streamlit Interface



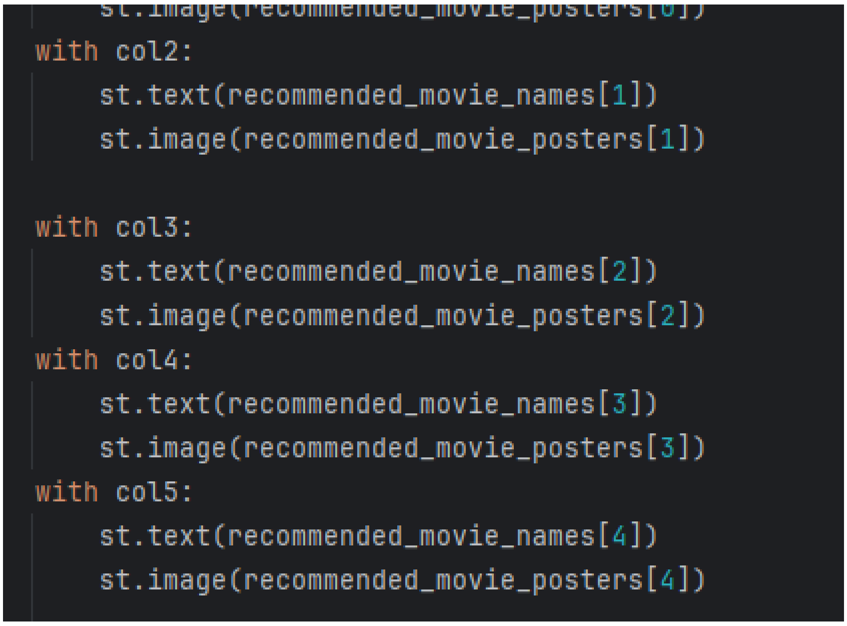
```
st.header('Movie Recommender System')
movies = pickle.load(open('movie_list.pkl','rb'))
similarity = pickle.load(open('similarity.pkl','rb'))

movie_list = movies['title'].values
selected_movie = st.selectbox(
    "Type or select a movie from the dropdown",
    movie_list
)
```

5. 5 Code for User Input and Button



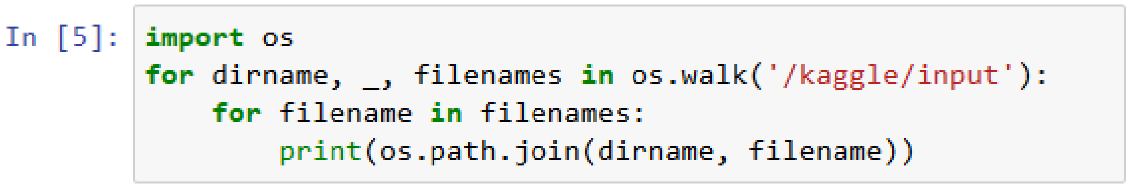
```
if st.button('Show Recommendation'):
    recommended_movie_names,recommended_movie_posters = recommend(selected_movie)
    col1, col2, col3, col4, col5 = st.columns(5)
    with col1:
        st.text(recommended_movie_names[0])
        st.image(recommended_movie_posters[0])
    with col2:
        st.text(recommended_movie_names[1])
        st.image(recommended_movie_posters[1])

    with col3:
        st.text(recommended_movie_names[2])
        st.image(recommended_movie_posters[2])
    with col4:
        st.text(recommended_movie_names[3])
        st.image(recommended_movie_posters[3])
    with col5:
        st.text(recommended_movie_names[4])
```

5. 6 Code for Display Recommended Movies



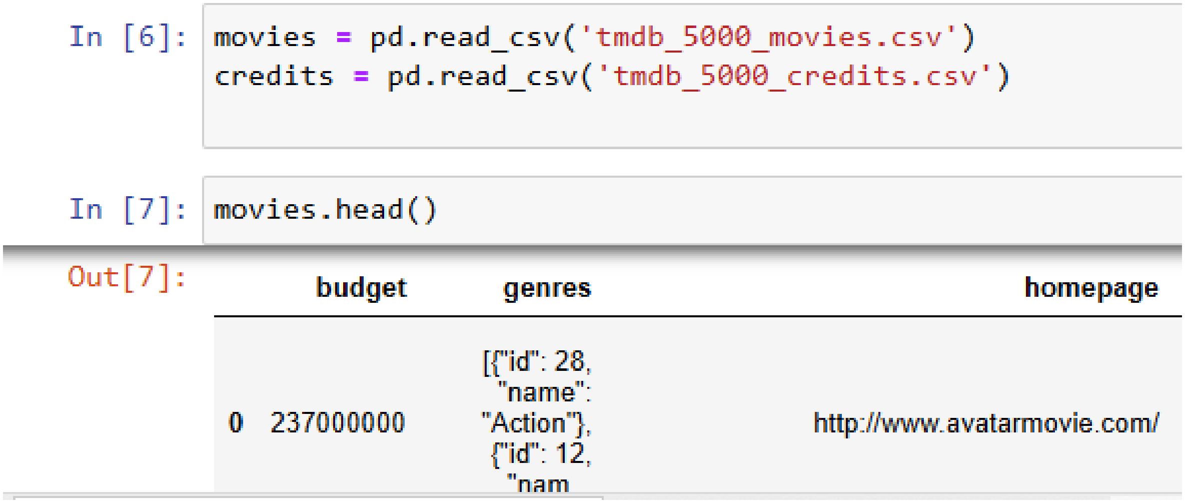
```
        st.image(recommended_movie_posters[0])
    with col2:
        st.text(recommended_movie_names[1])
        st.image(recommended_movie_posters[1])

    with col3:
        st.text(recommended_movie_names[2])
        st.image(recommended_movie_posters[2])
    with col4:
        st.text(recommended_movie_names[3])
        st.image(recommended_movie_posters[3])
    with col5:
        st.text(recommended_movie_names[4])
        st.image(recommended_movie_posters[4])
```

5.7 Code for Exploring Dataset files



In [5]:
```
import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

5.8 Code for Loading Dataset files into Pandas DataFrame



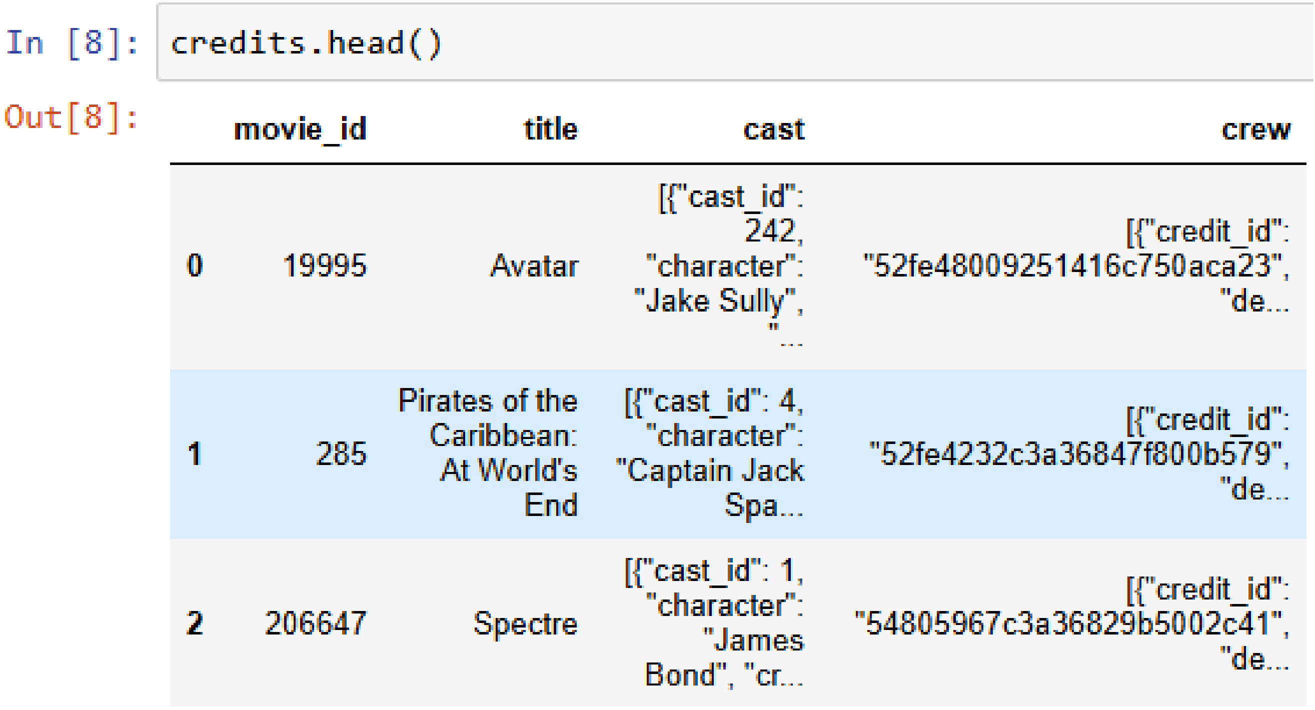
In [6]:
```
movies = pd.read_csv('tmdb_5000_movies.csv')
credits = pd.read_csv('tmdb_5000_credits.csv')
```

In [7]:
```
movies.head()
```

Out[7]:

	budget	genres	homepage
0	237000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam	http://www.avatarmovie.com/

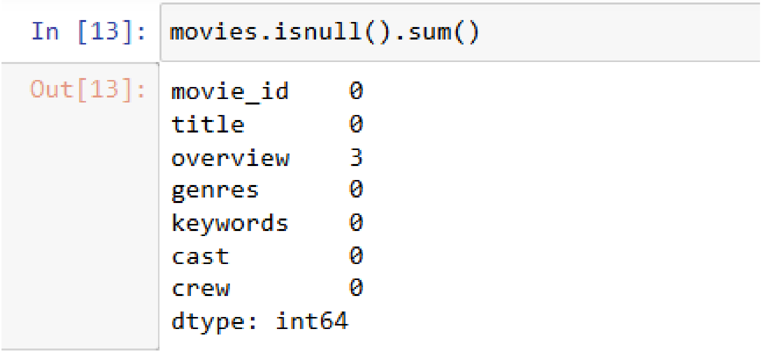
5.9 Viewing the first few rows of the 'movies' DataFrame



```
In [8]: credits.head()
Out[8]:
```

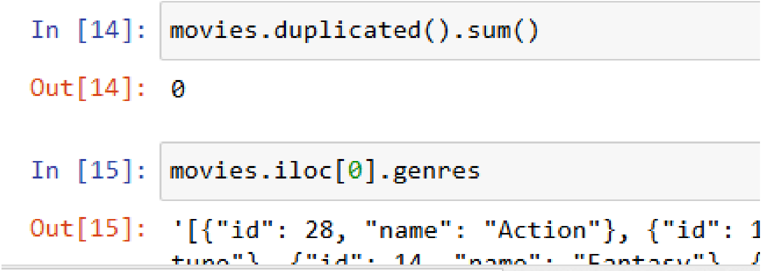
	movie_id	title	cast	crew
0	19995	Avatar	[{"cast_id": 242, "character": "Jake Sully", "...	[{"credit_id": "52fe48009251416c750aca23", "de...
1	285	Pirates of the Caribbean: At World's End	[{"cast_id": 4, "character": "Captain Jack Spa...	[{"credit_id": "52fe4232c3a36847f800b579", "de...
2	206647	Spectre	[{"cast_id": 1, "character": "James Bond", "cr...	[{"credit_id": "54805967c3a36829b5002c41", "de...

5.10 Handling missing values and duplicates in the 'movies' DataFrame



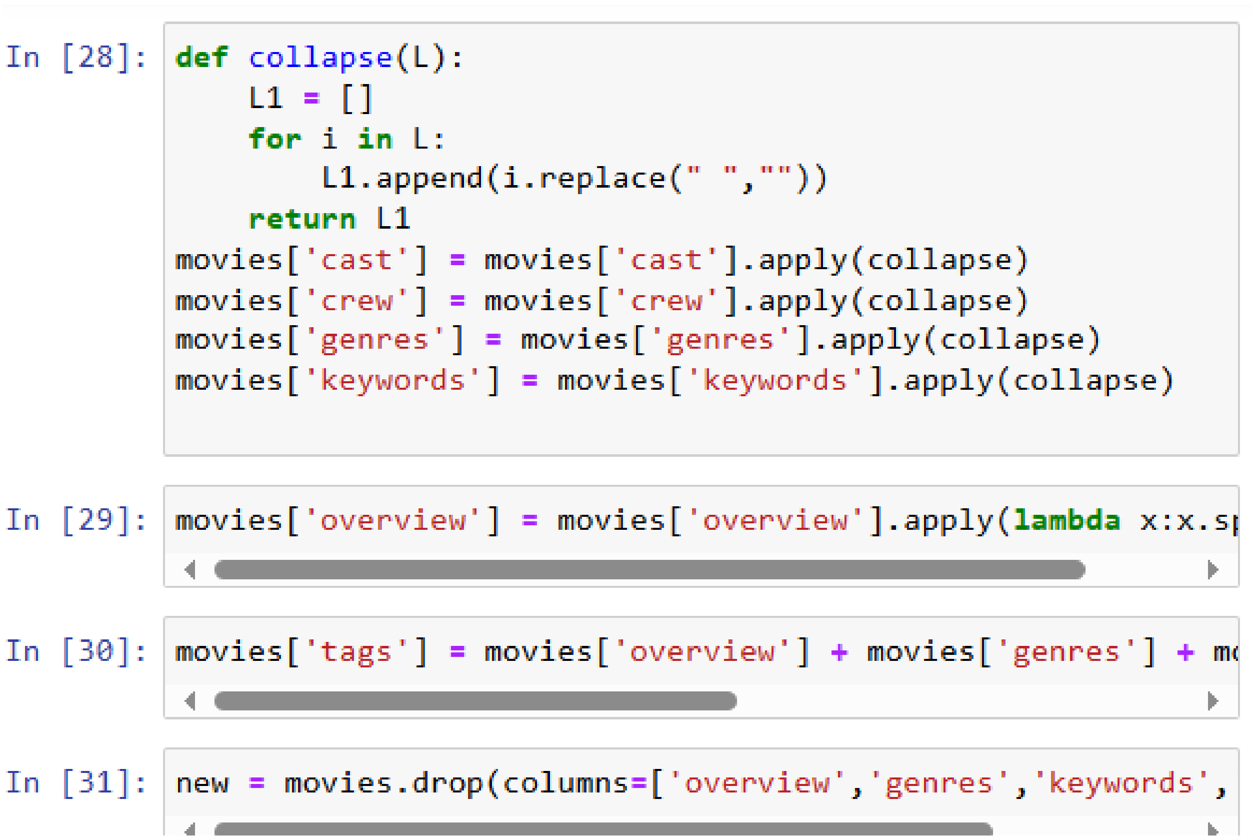
```
In [13]: movies.isnull().sum()

Out[13]: movie_id    0
         title       0
         overview    3
         genres      0
         keywords    0
         cast        0
         crew        0
         dtype: int64
```

movies.dropna(inplace=True)



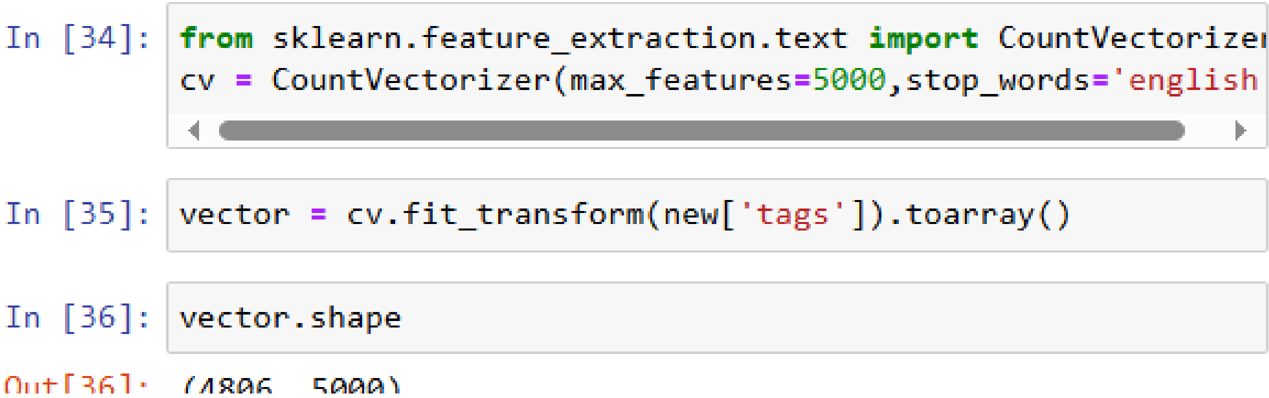
```
In [14]: movies.duplicated().sum()

Out[14]: 0

In [15]: movies.iloc[0].genres

Out[15]: '[{"id": 28, "name": "Action"}, {"id": 1
         ture"}, {"id": 14, "name": "Fantasy"}, {
```

5.11 Creating new DataFrame with Modifie features



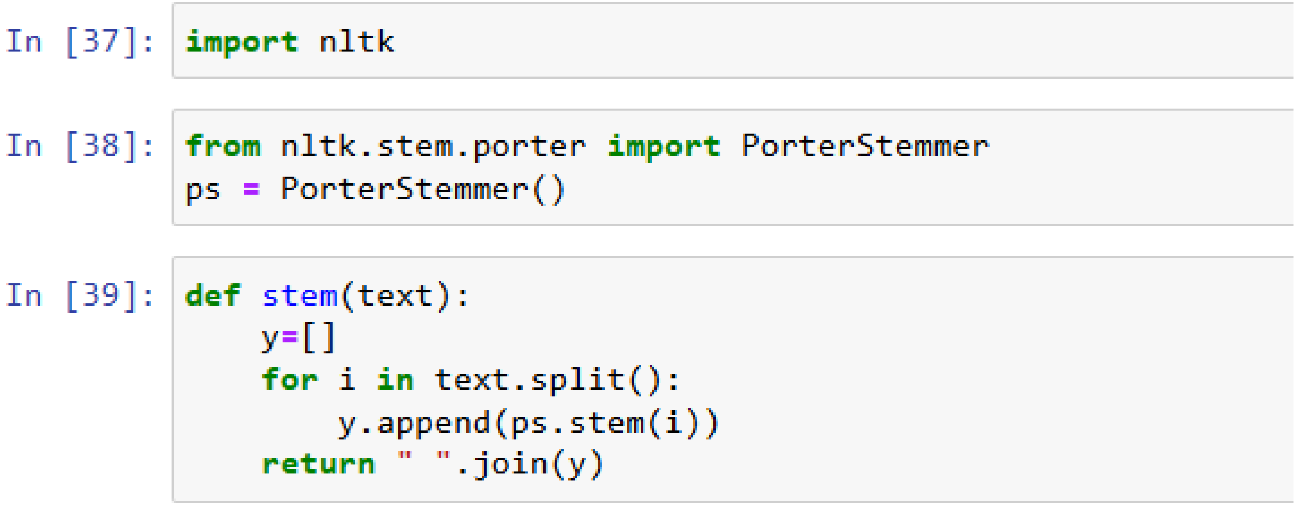
```
In [28]: def collapse(L):
             L1 = []
             for i in L:
                 L1.append(i.replace(" ",""))
             return L1
         movies['cast'] = movies['cast'].apply(collapse)
         movies['crew'] = movies['crew'].apply(collapse)
         movies['genres'] = movies['genres'].apply(collapse)
         movies['keywords'] = movies['keywords'].apply(collapse)

In [29]: movies['overview'] = movies['overview'].apply(lambda x:x.s

In [30]: movies['tags'] = movies['overview'] + movies['genres'] + mo

In [31]: new = movies.drop(columns=['overview','genres','keywords',
```

5.12 Feature extraction using CountVectorizer



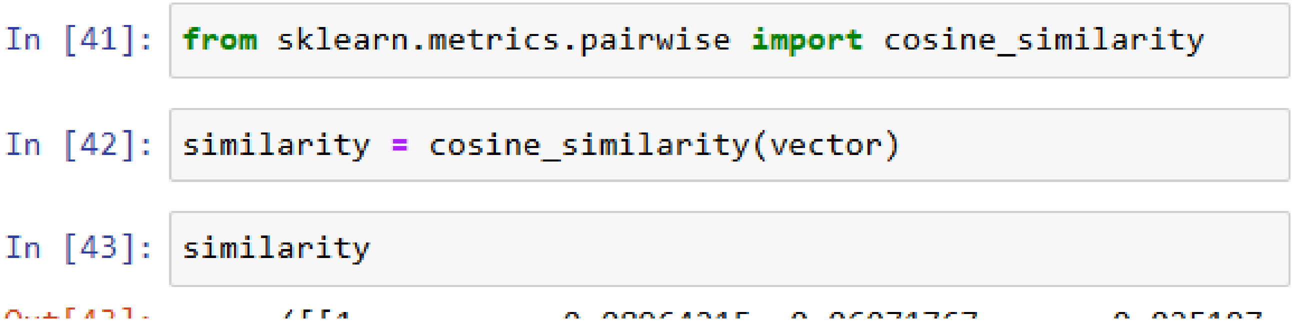
```
In [34]: from sklearn.feature_extraction.text import CountVectorize
         cv = CountVectorizer(max_features=5000,stop_words='english

In [35]: vector = cv.fit_transform(new['tags']).toarray()

In [36]: vector.shape

Out[36]: (4806, 5000)
```

5.13 Text pre-processing using NLTX



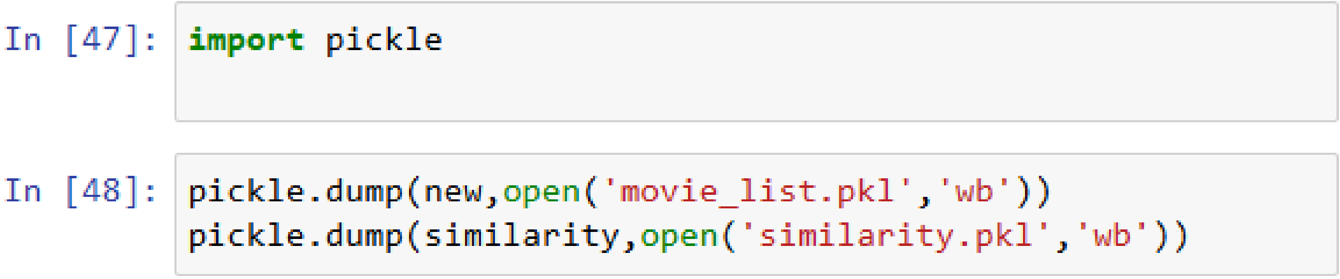
```
In [37]: import nltk

In [38]: from nltk.stem.porter import PorterStemmer
         ps = PorterStemmer()

In [39]: def stem(text):
             y=[]
             for i in text.split():
                 y.append(ps.stem(i))
             return " ".join(y)
```

5.14 Calculating similarity between movies Recommendation system



```
In [41]: from sklearn.metrics.pairwise import cosine_similarity

In [42]: similarity = cosine_similarity(vector)

In [43]: similarity

Out[43]: array([[1.        , 0.08964215, 0.06071767, ..., 0.025107
```

5.15 Saving processed data and similarity matrix



```
In [47]: import pickle

In [48]: pickle.dump(new,open('movie_list.pkl','wb'))
         pickle.dump(similarity,open('similarity.pkl','wb'))
```

6. Feasibility Report

A possibility ponder serves as a condensed outline of the complete handle, pointing to address key request such as recognizing the issue, investigating reasonable arrangements, and assessing the value of understanding the issue. Conducted once the issue is clearly characterized, this think about is vital to find out the achievability of the proposed framework, considering specialized, operational, and financial components. A comprehensive achievability ponder gives administration with a clear point of view on the proposed framework. To guarantee extend reasonability and recognize potential impediments, consider includes three fundamental angles:

6.1. Specialized Achievability:

This organize confirms the specialized reasonability of proposed frameworks, guaranteeing that all required advances for framework improvement are promptly accessible.

This stage verifies the technical viability of proposed systems, ensuring that all required technologies for system development are readily available. Factors such as the existence of necessary technology, system flexibility, guarantees of accuracy and reliability, and quick response to inquiries contribute to technical feasibility. Our project is technically feasible as all required technologies, including Android/iOS as the operating system, Dart as the language, MS-SQL Server as the database system, and MS-Word as the documentation tool, are readily accessible.

6.2. Economic Feasibility:

Financially, the **venture** is **regarded doable**, requiring no **extra monetary speculation** and having the potential to be completed **inside** a six-month **time allotment**. **Financial achievability includes** comparing the **monetary** benefits of the **unused framework** with the **related venture** costs
Contemplations incorporate the costs of **framework examination**, hardware/software, **improvement instruments**, and **upkeep**. Our **extend** is **financially attainable** due to **negligible improvement** costs compared to the **significant money related** benefits.

6.3. Operational Feasibility:

This phase assesses various operational factors of proposed systems, such as manpower and time, aiming to identify the solution with the least operational resource usage. The chosen solution should be operationally implementable and align with user objectives. Operational feasibility ensures that the proposed system can seamlessly integrate into current operations without causing issues. Our project is operationally feasible, meeting time and personnel requirements with a four-member team working on it for three months. Client involvement in planning and development further ensures operational acceptance.

In summary, the feasibility study rigorously evaluates technical, economic, and operational aspects to determine the viability of the proposed system, providing a solid foundation for decision-making by project management

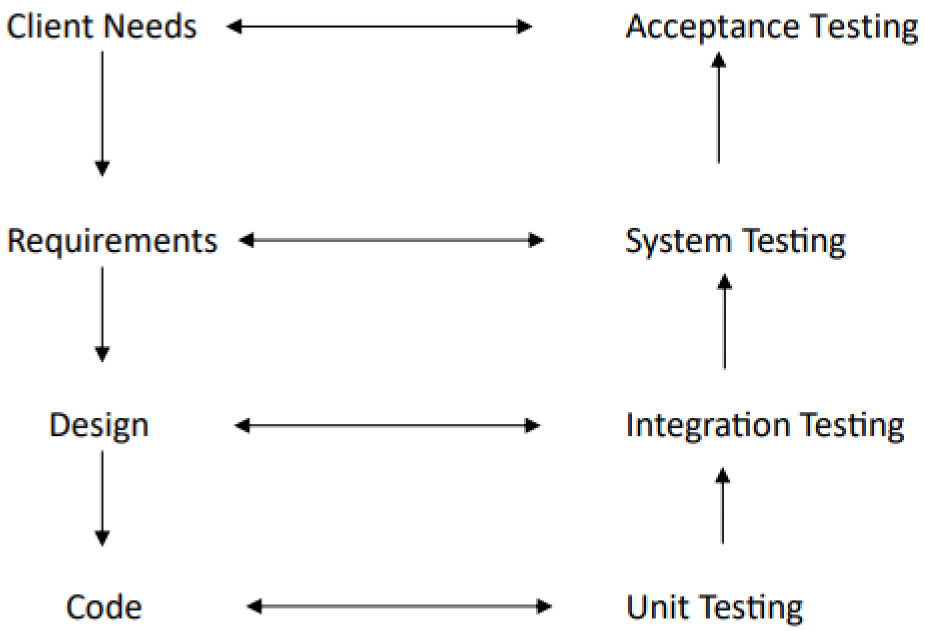
7. Testing

Testing is an integral part of icing the success of a design, especially when dealing with large-scale trials. The design attains success when all its factors serve duly in colourful aspects and produce the asked affair for different inputs. Thus, to achieve design success, thorough testing is imperative. The conducted testing, in this case, concentrated on System Testing, wherein the thing was to corroborate the satisfaction of stoner conditions.

The law for the new system was entirely developed using JAVA as the rendering language, and the front-end design interface employed Android Studio. Rigorous testing, involving druggies, was performed, checking all operations exhaustively to insure their functionality from every perspective. Despite encountering crimes in some operations, corrective measures were enforced before perpetration. The inflow of forms was set up to align seamlessly with the factual inflow of data.

7.1 Levels of Testing

Testing is organized into different situations to identify crimes in colourful phases, and these situations include: Client Needs Acceptance Testing Requirements System Testing Design Integration Testing Code Unit Testing A sequence of tests is executed for the proposed system before it undergoes stoner acceptance testing. The way involved in testing are as follows



7.1.1 Unit Testing

Focuses on vindicating the lowest unit of the software design, icing each module works satisfactorily during the programming stage. Modules are tested independently to validate their anticipated affair.

7.1.2 Integration Testing

Totally tests the program structure while uncovering crimes associated with module interfaces. Points to make a program structure by combining and testing all modules as a whole

7.1.3 System Testing

Ensures that the system functions directly and efficiently ahead of live operation commences. Vital for the success of the system, assuming that if all corridors are correct, the overall thing will be successfully achieved.

7.1.4 Confirmation Testing

Conducted after integration testing, validating whether the software functions in a manner anticipated by the client. Identifies diversions from specifications and creates an insufficiency list if necessary.

7.1.5 Affair Testing

Follows confirmation testing and involves testing the system's labors in the specified format. Checks both on-screen and published formats to ensure they meet stoner conditions.

7.1.6 Stoner Acceptance

Testing Key to the success of any system, this testing involves constant commerce with prospective druggies during development. One is the function or performance characteristics confirmed to specifications and are accepted and the other is a deviation from specification is uncovered and a deficiency list is created. Proposed system under consideration has been tested by using validation testing and found to be working satisfactorily. For the hard copy, the output comes as the specified requirements by the users. Hence output testing does not result in any corrections in the system.

7.2 Test Cases

All functional requirements are tested and are tabulated below:

Sl. No.	Er. No.	expected	obtained	result
1	FR1	Clients can enlist for an account. They have to be have secure get to to urge their claim motion picture offer.	Clients was able to enlist for an account. They have to be have secure get to to urge their claim motion picture offer.	PASS
2	FR2	Clients can seek for motion pictures by title. Other strategies or parameters ought to too be considered. Choose a motion picture they like.	Clients was able to seek for motion pictures by title. Other strategies or parameters ought to too be considered. Choose a motion picture they like.	PASS
3	FR3	The framework ought to give individualized film proposals. Suggestions ought to be based on client inclinations and seeing history. Perhaps indeed depending on the rating.	The framework was able to give individualized film proposals. Suggestions ought to be based on client inclinations and seeing history. Perhaps indeed depending on the rating.	PASS
4	FR4	The framework ought to utilize content-based sifting strategies to create film proposals. There are a few things to	he framework was able to utilize content-based sifting strategies to create film proposals. There are a few things to	PASS

		consider. These incorporate motion picture portrayals, catchphrases, and analogies.	consider. These incorporate motion picture portrayals, catchphrases, and analogies.	
5	FR5	Clients ought to be able to grant criticism on prescribed motion pictures. Like it or not by their evaluations. This input will assist refine the proposal prepare.	Clients was able to be able to grant criticism on prescribed motion pictures. Like it or not by their evaluations. This input will assist refine the proposal prepare.	PASS
6	FR6	Clients ought to be able to oversee their possess profiles. Inclinations and seeing history ought to be upgraded. An bookkeeping framework must be kept up.	Clients was able to be able to oversee their possess profiles. Inclinations and seeing history ought to be upgraded. An bookkeeping framework must be kept up.	PASS
7	FR7	The proposal framework ought to be consistently coordinates in a user-friendly site. This permits clients to connected with it rapidly.	The proposal framework was able to be consistently coordinates in a user-friendly site. This permits clients to connected with it rapidly.	PASS
8	FR8	The site interface ought to be responsive. This guarantees the leading	The site interface was able to be responsive. This guarantees the leading	PASS

		conceivable client involvement on all gadgets and screen sizes.	conceivable client involvement on all gadgets and screen sizes.	
9	FR9	The framework must run on a server optimized for execution. It should be flexible. It must be reliable and responsive to the user's demands.	The framework was able to run on a server optimized for execution. It should be flexible. It must be reliable and responsive to the user's demands.	PASS
10	FR10	Frameworks should be completely tried to guarantee that they are working. Make beyond any doubt the proposals are precise and secure.	Frameworks was able to completely tried to guarantee that they are working. Make beyond any doubt the proposals are precise and secure.	PASS

7.3 VALIDATION CRITERIA

1. In each form, no field which isn't null suitable should be left blank.

2. All numeric fields should be checked for non-numeric values. Also, textbook fields like names shouldn't contain any numeric characters.

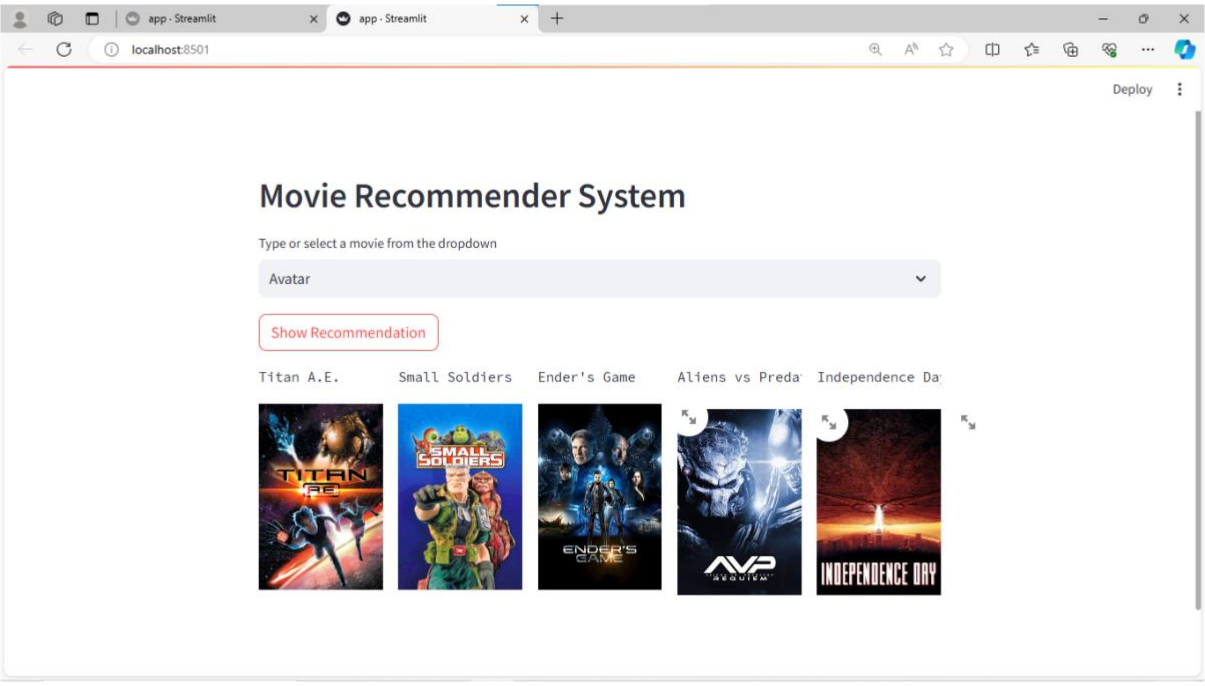
3. All primary keys should be automatically generated to help the stoner from entering any being key.

4. Use of error handling for each Save, Edit, cancel and other important operations.

5. Whenever the stoner Tabs out or Enter from a textbook box, the data should be validated and if it's invalid, focus should again be transferred to the textbook box with proper communication

8. Result

Upon doing research and implementing the project we have obtained the website as a result.



Movie Recommender System

Type or select a movie from the dropdown

Avatar

Show Recommendation

Titan A.E. Small Soldiers Ender's Game Aliens vs Preda Independence Da

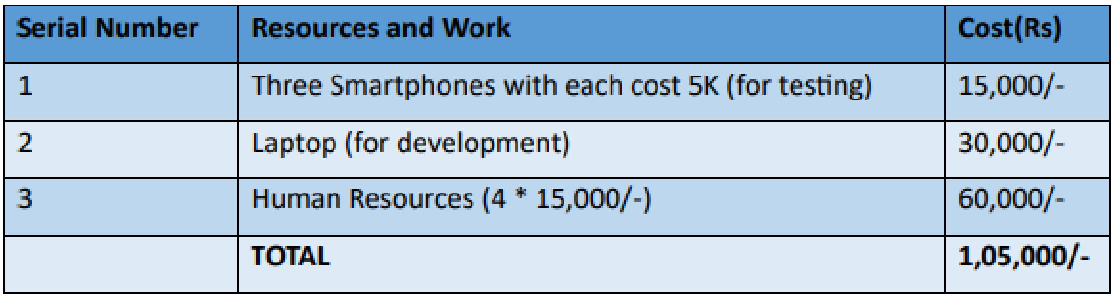
9. Project Costing

This chapter deals with costing of this project which gives an overall estimation of expenses that was required to complete this project. This Covers expenses of testing devices, Platform and Hardware cost, Human Resource Cost and a grand total of Entire cost.

9.1 Project Cost Estimation

The cost of project is summarised in a tabular form displayed below:

Table 4 Cost estimation table



Serial Number	Resources and Work	Cost(Rs)
1	Three Smartphones with each cost 5K (for testing)	15,000/-
2	Laptop (for development)	30,000/-
3	Human Resources (4 * 15,000/-)	60,000/-
	TOTAL	**1,05,000/-**

10. Conclusion

In conclusion, the Movie recommender system commences with an in-depth literature check that drew perceptivity from different sources, including IEEE papers, patented documents, estimable websites, and orally published books. This check served as a foundation for comparing operations, expounding their features, and informing the design of the Movie recommender system operation. The design strictly excavated into the background proposition, expounding the technology, frame, IDEs, and machines employed. This thorough disquisition aimed to grease the effective operation of these propositions and give a clear explanation for their selection. Later, the design's objects were fully outlined, aligning with the design's title and end. Styles and methodologies were detailed, forming a structured approach to achieving each ideal, and functional conditions were pulled for consecutive design completion. The visual representation of the design included the creation of colourful plates

— Use- case, Sequence, Class, Block, and Widget Hierarchy plates offering a comprehensive view from different perspectives. Performance of the design was showcased through Java and Dart programming languages, demonstrating impeccable commerce to realize the Movie recommender system operation. Rigorous testing validated the successful functioning of all functionalities. In the results section, the operation's different countries were captured through screenshots, each accompanied by an explanation of its significance. Performance analysis quantified the operation's effectiveness, pressing advancements compared to being request druthers. The project brings estimation handed perceptivity into the fiscal coffers demanded for design reconstruction. The design crowned in a befitting conclusion, recapitulating vital findings and outlining its compass for unborn developments. Movie recommender system stands as a testament to scrupulous exploration, effective performance, and a commitment to enhancing accuracy of recommendation.

11. References

[1.] Yehuda Korean, Robert Bell, and Chris Volinsky (2009). Matrix Factorization Techniques for Recommender Systems.

[2.] Yehuda Korean, Robert Bell, and Chris Volinsky (2008). Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model.

[3.] Hansaim Lim & Lei Xie (2018). Target Gene Prediction of Transcription Factor using a new Neighborhood-regularized Tri-factorization One-class Collaborative Filtering Algorithm.