

A
MINI PROJECT REPORT
ON
“Movie Recommendation System”

Submitted by
Suchi Sathavara (19IT402)
Dharitri Arya (19IT407)
Dhruvika Gajera (19IT414)

Prof. Trushna Patel
Faculty Guide



Information Technology Department
Birla Vishvakarma Mahavidyalaya Engineering College
(An Autonomous Institution)
AY: 2021-22, Semester VI



Birla Vishvakarma Mahavidyalaya Engineering College

(An Autonomous Institution)

Information Technology Department

AY: 2021-22, Semester VI

CERTIFICATE

This is to certify that project entitled with **“Movie Recommendation System”** has been successfully carried out by **Suchi Sathavara (19IT402), Dharitri Arya (19IT407), and Dhruvika Gajera (19IT414)** for the subject of **3IT31-Mini Project** under my guidance during the academic year 2021-22, Semester VI. The Mini Project work carried out by the students of 6th semester is satisfactory.

Date:

Prof. Trushna Patel

Faculty Guide
IT Department
BVM

Dr. Nilesh B. Prajapati

Course Coordinator
IT Department
BVM

Dr. Keyur

Brahmbhatt
Head of the Department
IT Department
BVM

ACKNOWLEDGEMENT

We are extremely grateful to **Dr. Indrajit Patel**, Principal, Birla Vishvakarma Mahavidyalaya Engineering College and **Dr. Keyur Brahmbhatt**, Head of the Information Technology Department, for providing all the required resources for the successful completion of our project. Special thanks is due to our **guide Prof. Trushna Patel** whose help, stimulating suggestions and encouragement helped us in all time of fabrication process and in writing this report. We also sincerely thanks for the time spent proof reading and correcting our many mistakes.

Many thanks go to the both guides who have given their full effort in guiding the team in achieving the goal as well as their encouragement to maintain our progress in track. Our profound thanks go to all classmates, especially to our friends for spending their time in helping and giving support whenever we need it in our project.

We express our thanks to project **coordinator (Dr. Nilesh B. Prajapati)**, all staff members and friends, for all the help and co-ordination extended in bringing out this project successfully in time.

We will be failing in duty if we do not acknowledge with grateful thanks to the authors of the references and other literatures referred to in this project. Last but not the least; we are very much thankful to our parents who guided us in every step which we took.

Thanking You,

Suchi Sathavara,

Dharitri Arya,

Dhruvika Gajera.

Report of Movie Recommendation System

Table of Content

Chapter 1: Introduction

- 1.1 Aim of the Project
- 1.2 Project Scope
- 1.3 Project Objective
- 1.4 Project Modules
- 1.5 Project basic requirements

Chapter 2: Analysis, Design Methodology and Implementation Strategy

- 2.1 Comparison of Existing Applications with your Project
- 2.2 Project Feasibility Study
- 2.3 Project Timeline chart
- 2.4 Detailed Modules Description
- 2.5 Project SRS
 - 2.5.1 Use Case Diagrams
 - 2.5.2 Class Diagrams
 - 2.5.3 Entity Relationship Diagrams
 - 2.5.4 Data Flow Diagram
 - 2.5.5 State diagram
 - 2.5.6 Activity diagram
- 2.6 Database design and Normalization
- 2.7 Database Relation Diagram
- 2.8 Template Design

Chapter 3: Implementation and Testing

- 3.1 Software and Tools
- 3.2 User Interface and Snapshot
- 3.3 Testing using Use Cases

Chapter 4: Conclusion and Future work

- 4.1 Conclusion
- 4.2 Future work

References

INDEX

Sr No.	Table	Page No.
1	Test Case 1	26
2	Test Case 2	26
3	Test Case 3	27
4	Test Case 4	28

Sr No.	Figure	Page No.
1	Timeline Chart	4
2	Use Case Diagram	6
3	Class Diagram	9
4	ER Diagram	10
5	Data Flow Diagram	12
6	State Diagram	15
7	Activity Diagram	17

ABSTRACT

Whether we're buying a book on Amazon, watching a movie on Netflix, or reading an article on Medium, recommender algorithms have shaped our online choices.

Here, we offer a movie recommendation system that can recommend movies to both new and existing users. It scans movie databases for all the necessary information for suggestion, such as Genre, Actors; Etc. Experiments on real data have shown that the proposed solution is efficient and effective. Movie recommendation systems aim to assist movie buffs by recommending what film to watch without requiring them to go through the time-consuming and complex process of selecting from a vast number of films ranging from thousands to millions.

Our goal is to reduce human effort by recommending movies based on the user's preferences. To address these issues, we developed a content-based system.

Chapter 1: Introduction

1.1 Aim of the Project

The primary goal of a movie recommendation system is to give users with accurate movie recommendations. A system will recommend movies to users depending on their taste.

1.2 Project Scope

Due to a lack of time and search challenges, many people have difficulty finding an alternate movie item. Also, getting movie recommendations from friends can take time. The system allows you to save a lot of time.

The system will look through the movies that the user has already watched and then recommend movies to the user based on that information.

1.3 Project Objective

The goal is to provide movie suggestions based on their similarity ratings. The suggested project's main goal is to find content that would be of interest to a single person.

The recommender system would be able to select the top 5 movies with the highest similarity scores and show them as the top 5 recommendations based on the similarity scores.

1.4 Project Module

- Data Processing
- Recommendation Engine
- Web service

1.5 Project basic Requirements

➤ Hardware

Processor: Intel core i5 or later version

Hard-drive: Minimum 256 GB SSD

Ram: 8 GB or more

Graphics Card: Minimum 2 GB

➤ **Software**

Operating System: Windows 10

Anaconda Distribution: Jupyter Notebook

PyCharm: Python, sklearn, NumPy, Streamlit, pandas, pickle, hashlib, SQLite3.

Chapter 2: Analysis, Design Methodology and Implementation Strategy

2.1 Comparison of Existing Application with your Project

A Movie Recommender System: MOVREC employing Machine Learning Techniques and A System on Machine Learning based Movie Recommendation System, for example, had a restricted dataset.

By covering a huge dataset and using content-based filtering, our system has added the positive qualities of existing systems while overcoming the shortcomings of existing systems.

The present algorithm can:

- Recommend more movies based on the past, as well as movies in similar genres.
- Can keep track of his or her favorite IMDB-rated movies, as well as the most popular movie genre among n people.

2.2 Project Feasibility Study

A feasibility study of the proposed system is to be carried out during system analysis. A basic understanding of the system's primary requirements is required for feasibility analysis.

The dataset must have a significant number of clusters of comparable people and things for movie recommendation algorithms to work.

➤ Technical Feasibility :

This research is being carried out to determine the system's technical requirements. Any system that is created should not place a large burden on the available technical resources. The designed system has a small set of requirements, as it only requires little or no adjustments to be implemented.

➤ Economic Feasibility :

This system will be cost-effective because it will not require any additional hardware or software.

➤ **Operational Feasibility :**

This system will run smoothly, so you can be confident that the system you've created is operationally viable.

➤ **Social Feasibility :**

This system will include a popularity component. As a result, the review and popularity can be seen by anybody. Thus, it is socially feasible.

2.3 Project Timeline Chart

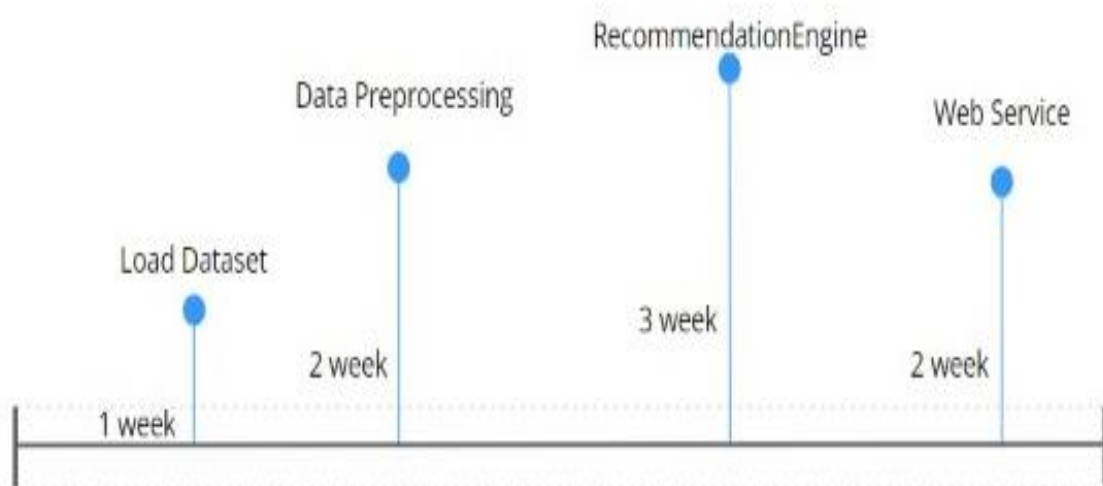


Figure 1: Timeline Chart

2.4 Detailed Modules Description

➤ **Data processing**

To obtained available dataset.

Perform various operations on it like data integration, data cleaning and data transformation.

Combining all this we can obtain required fields such as genre, cast, and so on.

➤ **Recommendation Engine**

It is main thing about system where with the help of k nearest neighbour machine learning algorithm we built recommendation engine.

The key thing where all the code for the recommender will be done is the **recommendation engine**.

➤ **Web service**

It is where user can access our recommendation system.

2.5 Project SRS

In numerous e-commerce applications, a recommendation system has become an essential component.

Recommender systems gather information about a user's preferences for various goods (e.g., movies, shopping, tourism, television, taxi) in one of two ways: implicitly or explicitly. Because the domain-specific object in our example is a movie, the major goal of our recommendation system is to filter and anticipate only those movies that a user would prefer based on some information about the user.

Every user receives generalized recommendations based on movie popularity and genre from the Simple Recommender.

2.5.1 Use Case Diagram

Use case diagrams are a frequent approach to communicating a software system's main functions.

At its most basic level, a use case diagram depicts a user's interaction with the system by illustrating the relationship between the user and the many Use cases in which the user is involved.

The use case diagram's purpose can be summarized as follows:

- To acquire a bird's eye view of a system, use this command.
- Determine the system's external and internal influences.
- It is used to collect system requirements.
- Demonstrate how the requirements interact with one another.

Symbols used in Use Case diagram:



Use case



Include



Association



Extend



Actor



Dependency



System



Generalization

Use Case Diagram of the System:

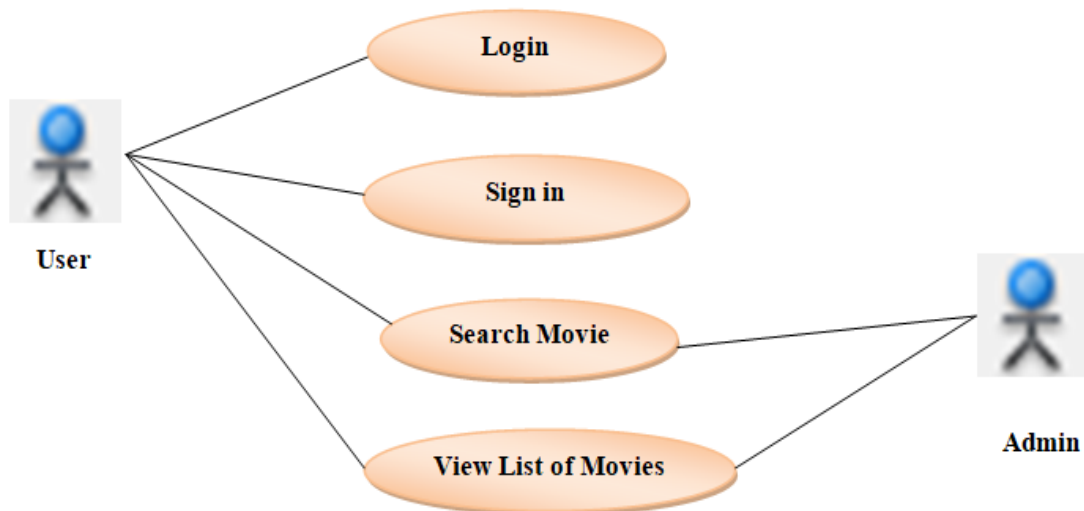


Figure 2: Use Case Diagram

If a person has already logged in to this system, he or she can Login. Otherwise, an error message will be displayed by the system.

Users must first sign in before being able to log into the system. After logging in, the user can search for a movie and then get a list of suggested films.

2.5.6 Class Diagram

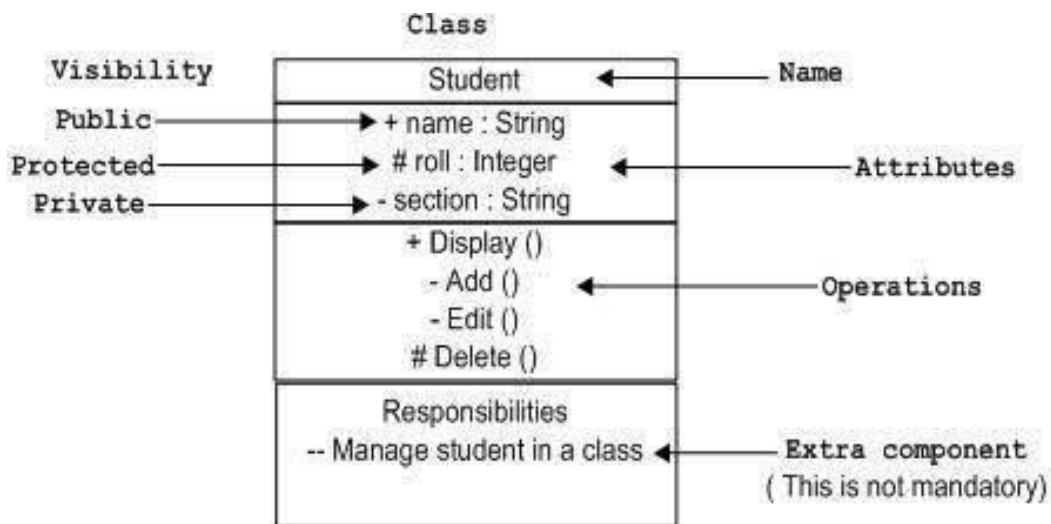
The class diagram is the main building block of object-oriented modeling. It is used both for general conceptual modeling of the system of the application, and for detailed modeling by translating the models into programming code. Class diagrams can also be used for data modeling.

The classes in a class diagram represent both the main elements, interactions in the application, and the class to be programmed.

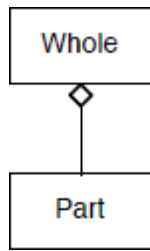
In the diagram, classes are represented by boxes that contain three compartments:

- The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized.
- The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lower case.
- The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lower case.

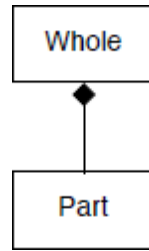
Symbols used in Class diagram:



Class



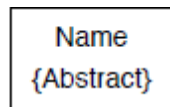
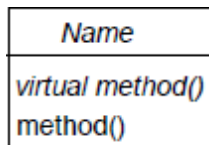
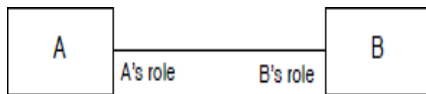
Aggregation



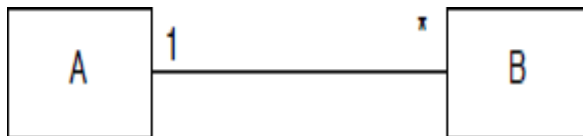
Composition



Association Inheritance



Abstract Class



Multiplicity in Aggregation,
Composition, or Association

Class Diagram of the system:

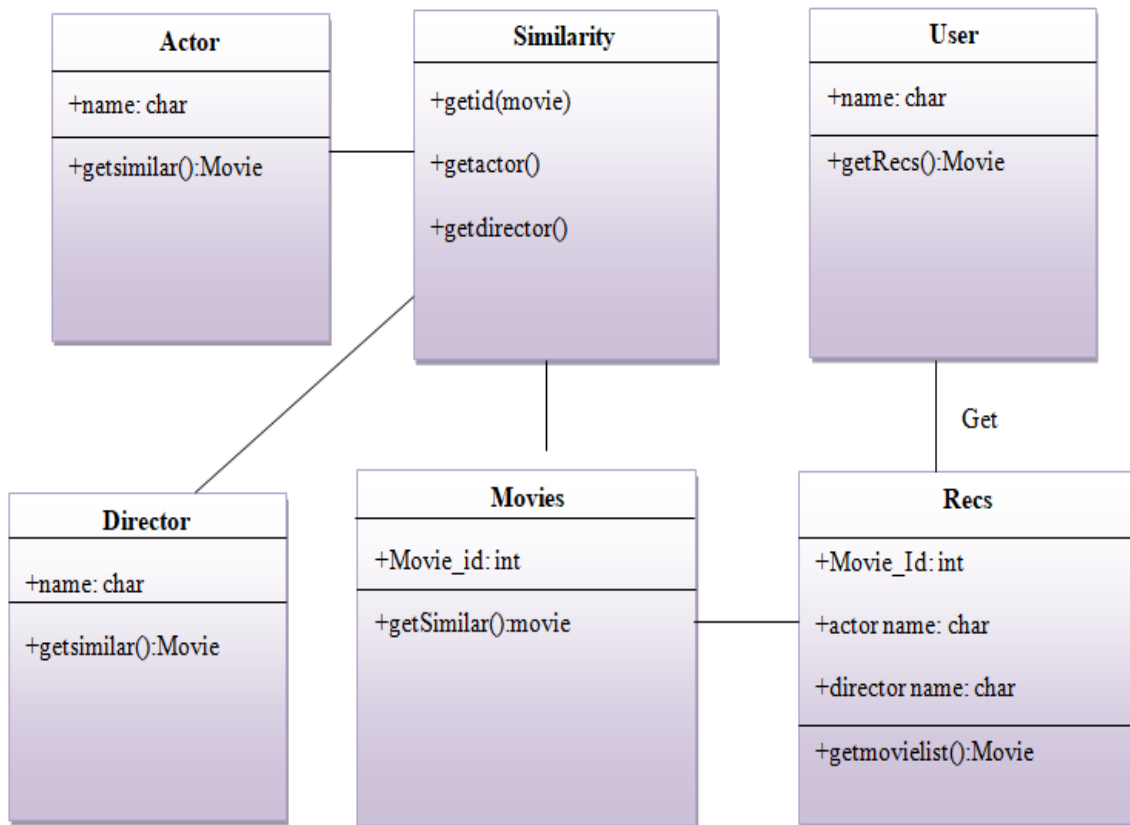


Figure 3: Class Diagram

Here the class diagram is given, having class names actor, similarity, user, director, movies and recs. In the class named actor, +name is the attribute and +getsimilar() is the operation and the same thing is common for all the remaining classes.

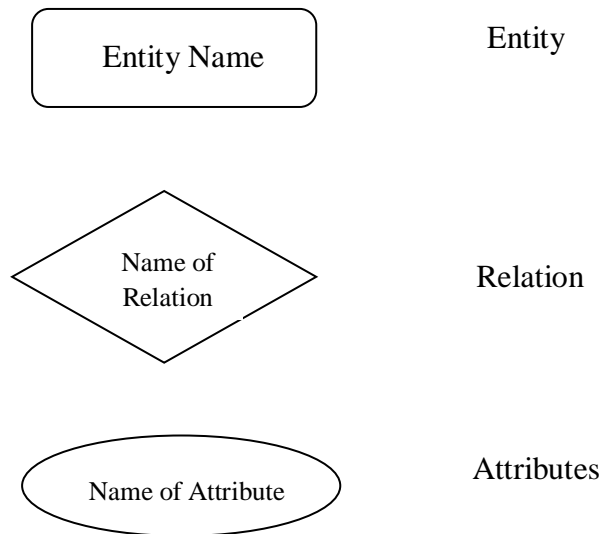
User enters the movie in recommender which has movie id ,actor and director so on the basis of their similarity , the similarity class will search among the 3 classes which are actor director and movies, for that all those 3 classes have the same operation i.e., +getsimilar():Movie.

2.5.3 Entity Relationship Diagram

The entity-relationship model is used to represent a logical design of a database to be created. In ER mode, real world objects (or concepts) are abstracted as entities, and different possible associations among them are modelled as relationships. We represent the attributes, entities and relations using the ER diagram.

Using this ER diagram, table structures are created, along with required constraints. Finally, these tables are normalized in order to remove redundancy and maintain data integrity.

Symbols used in ER diagram:



ER Diagram of the System:

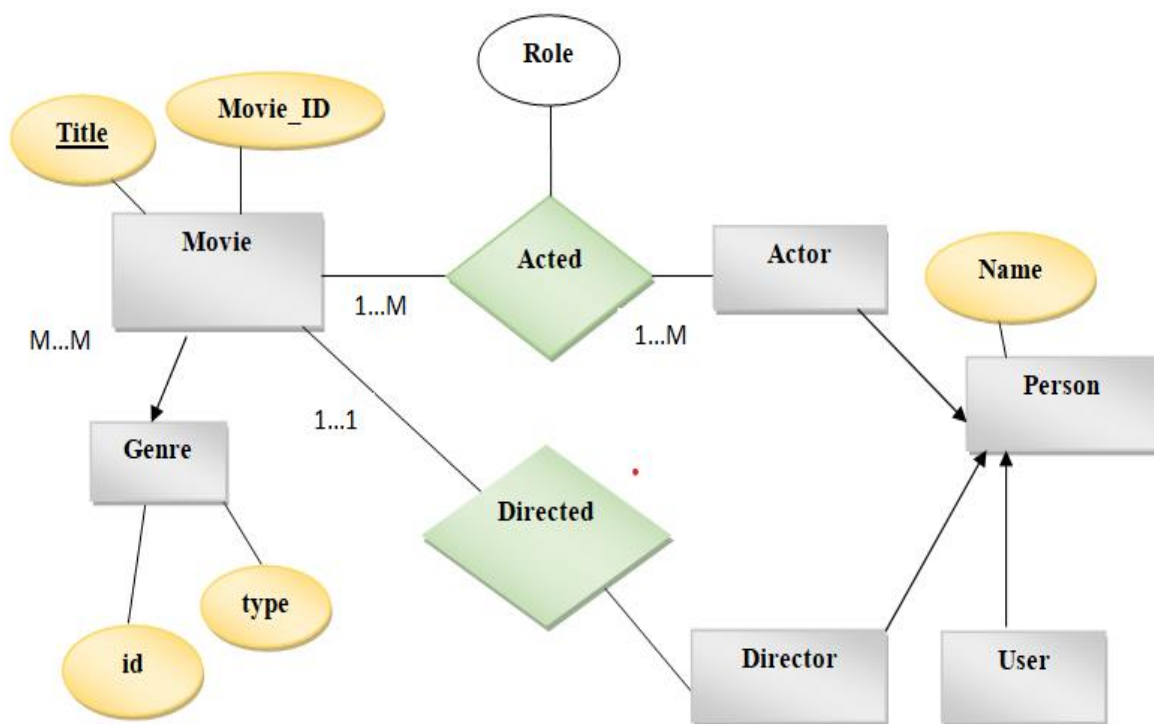


Figure 4: ER Diagram

The ER diagram shows the relationship between the attributes. Here the entity movie has 5 things i.e., genre, title, ID, its director and its actor.

One movie can have more than 1 genre and the same goes for genres so it has many to many relations. Genre has 2 things, i.e., type and ID. Now actor and director, a single movie can have more than one actor but a single director so there's one to many and one to one relations respectively.

Now they are both persons so are connected to that entity similar to the user who is also a person and is connected to name attribute.

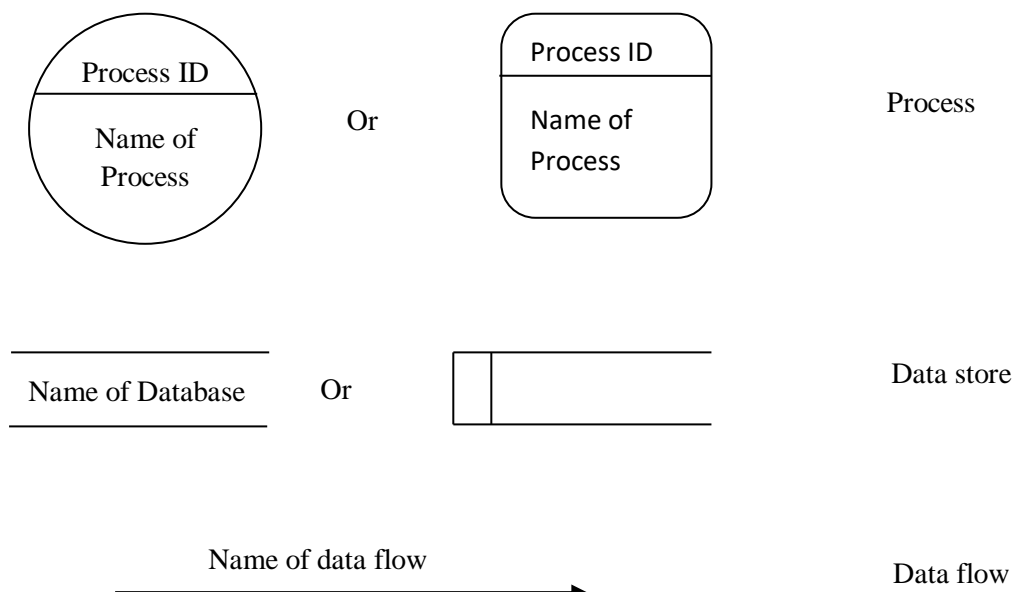
2.5.2 Data Flow Diagram

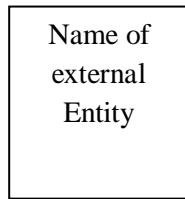
The Data Flow Diagram provides the functional overview of the system. The graphical representation easily overcomes any gap between 'user and system analyst' and analyst, system designer in understanding a system.

Starting from an overview of the system it explores detailed design of a system through a hierarchy.

DFD shows the external entities from which data flows into the process and also the other flows of data within a system. It also includes the transformations of data flow by the process and the data stores to read or write data.

Symbols used in Data Flow diagram:

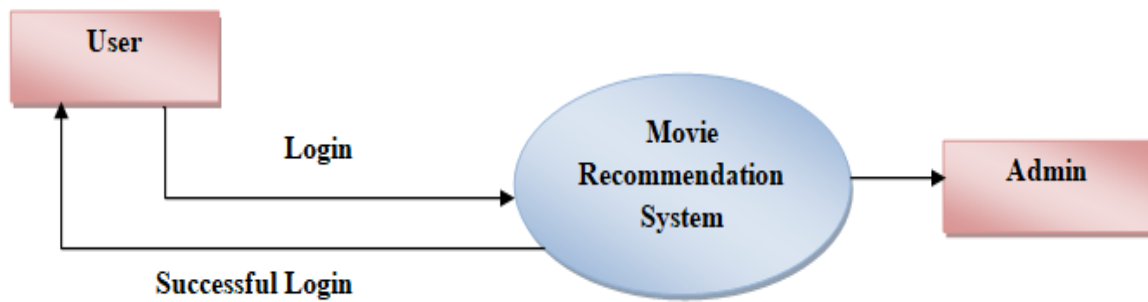




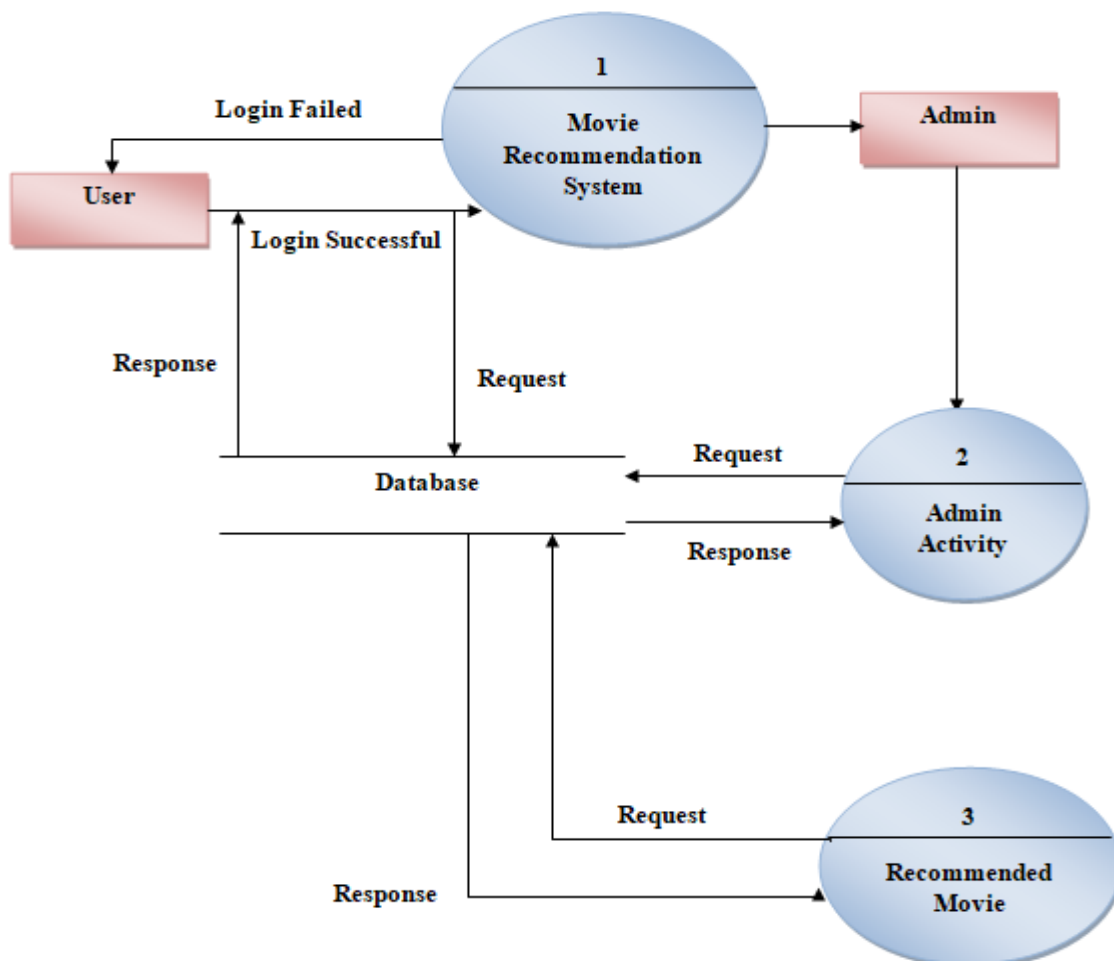
External Entity

Data Flow Diagram of the System:

Level 0:



Level 1:



Level 2:

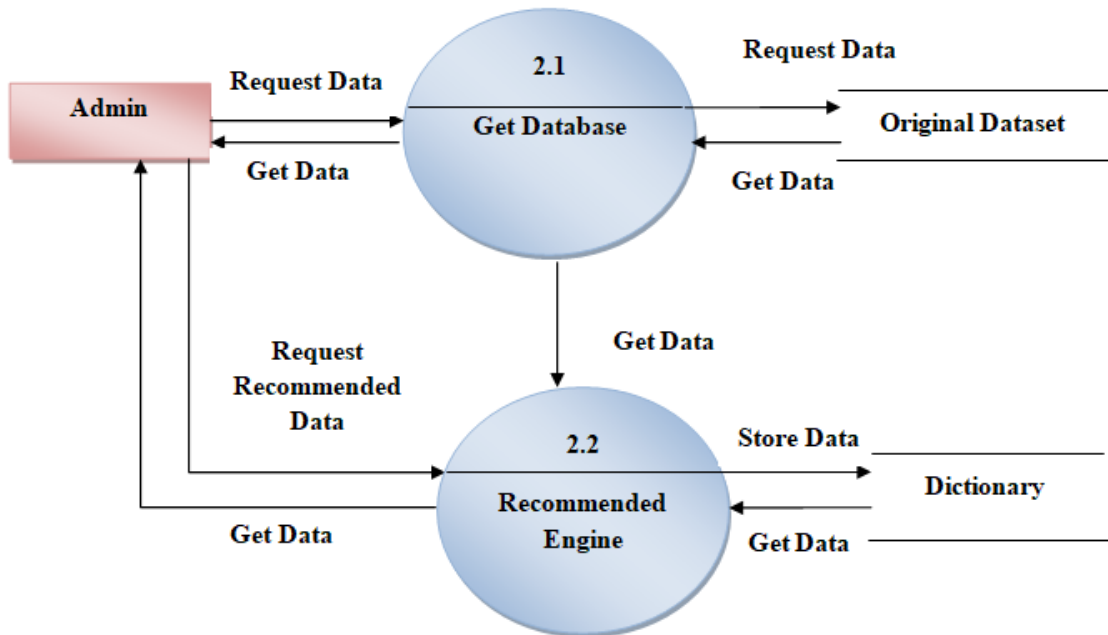


Figure 5: Data Flow Diagram

Level 0:

Here, the user is the database of the system, it logs into the system (process) whose information goes to another database called admin. It checks the information and if it's correct/valid, the user is logged in successfully into the system.

Level 1:

When the user logs in successfully, then the system informs the admin about it and sends the request to the database for the information that the user has input about the movie. Now the engine will give the relevant information to the database, which is then sent to the user.

When the user fails to login, then the system will not allow them to do any further activity and will be directed back to the login page.

Level 2:

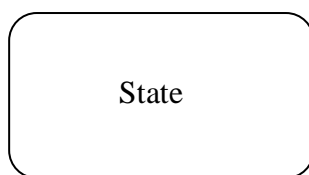
Now, when the user enters any information, the admin sends it to the database, which then requests that data from the dataset. The dataset sends the required data to the database which is then sent to the admin, which is then shown to the user.

After the user selects the movie, the admin requests the recommendation engine which finds the required data from the stored data of the dictionary and then sends it to the admin. Finally, the recommended movies are shown to the user.

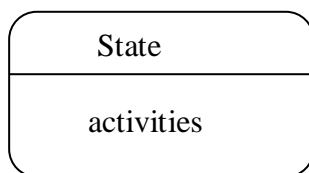
2.5.5 State Diagram

The behavior of classes in response to external stimuli is depicted in a state diagram. A state diagram, in particular, depicts the behavior of a single item in response to a series of system events.

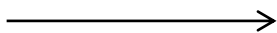
Symbols used in State diagram:



A Simple State



A State with internal Activity



Transition



Initial State



Final State

State Diagram of the System:

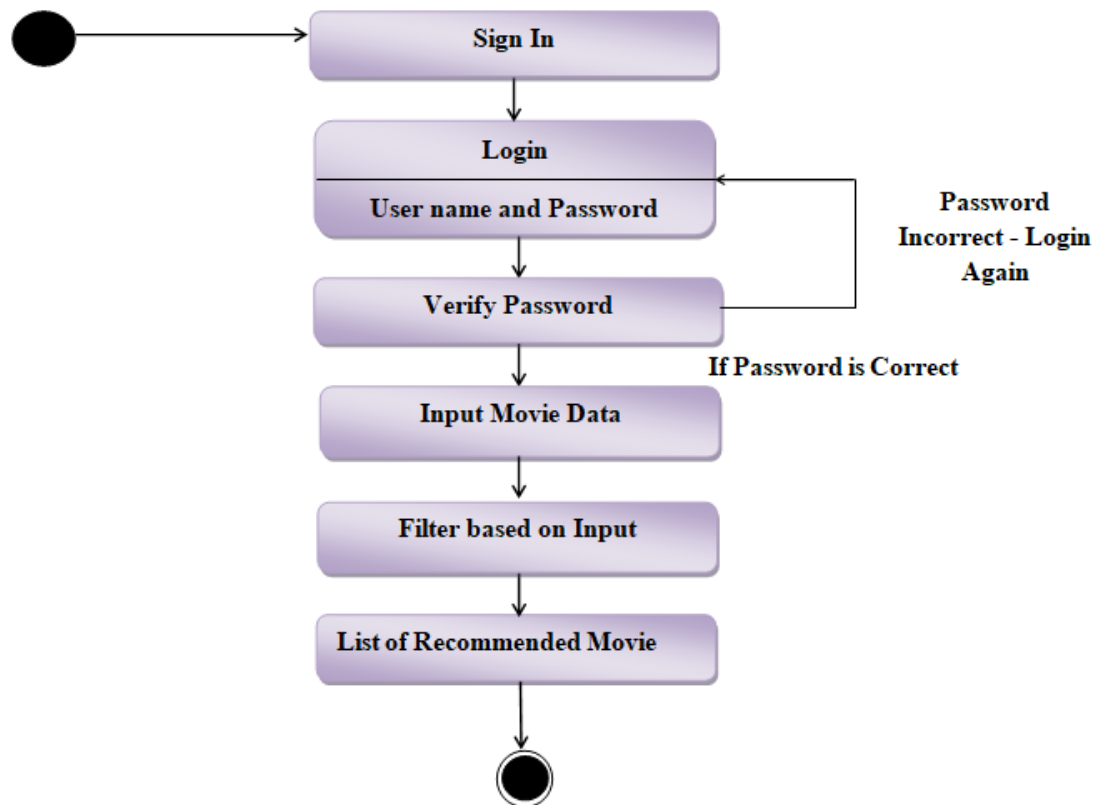


Figure 6: State Diagram

There are several states in this system, as depicted in the diagram above. First and foremost, the sign in state appears, requiring the user to sign in. After then, the user will be directed into the login state.

The system will check the password, and if it is correct, the user can log in and search for a movie. Otherwise, an error notice appears. Following the search, the system will filter the movies based on the information provided. Finally, it will make movie recommendations.

2.5.4 Activity Diagram

Activity Diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be describing as an operation of the system.

The basic purpose of activity diagram is to capture the dynamic behavior of the system. It is also called object-oriented flowchart.

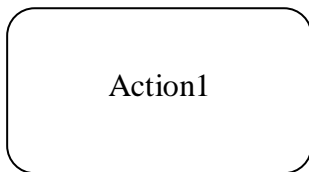
Activity diagram in UML allows you to create an event as an activity which contains a collection of nodes joined by edges. An activity can be attached to any modeling element to model its behavior. Activity diagrams are used to model,

- Use Cases
- Classes
- Interfaces
- Components
- Collaborations

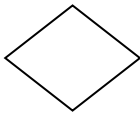
Symbols used in Activity diagram:



Initial-state



Action-box



Decision-box



Final-state

Activity Diagram of the System:

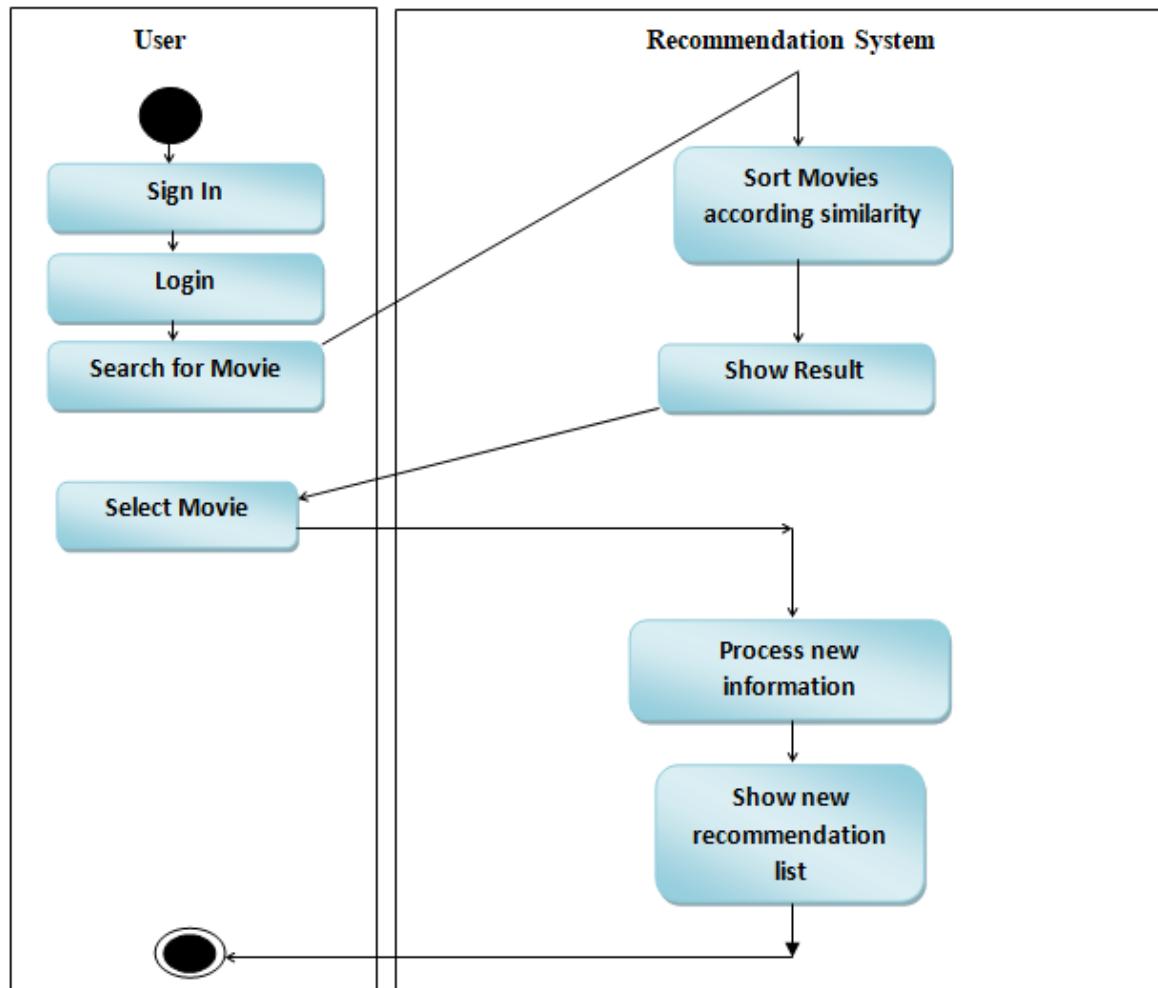
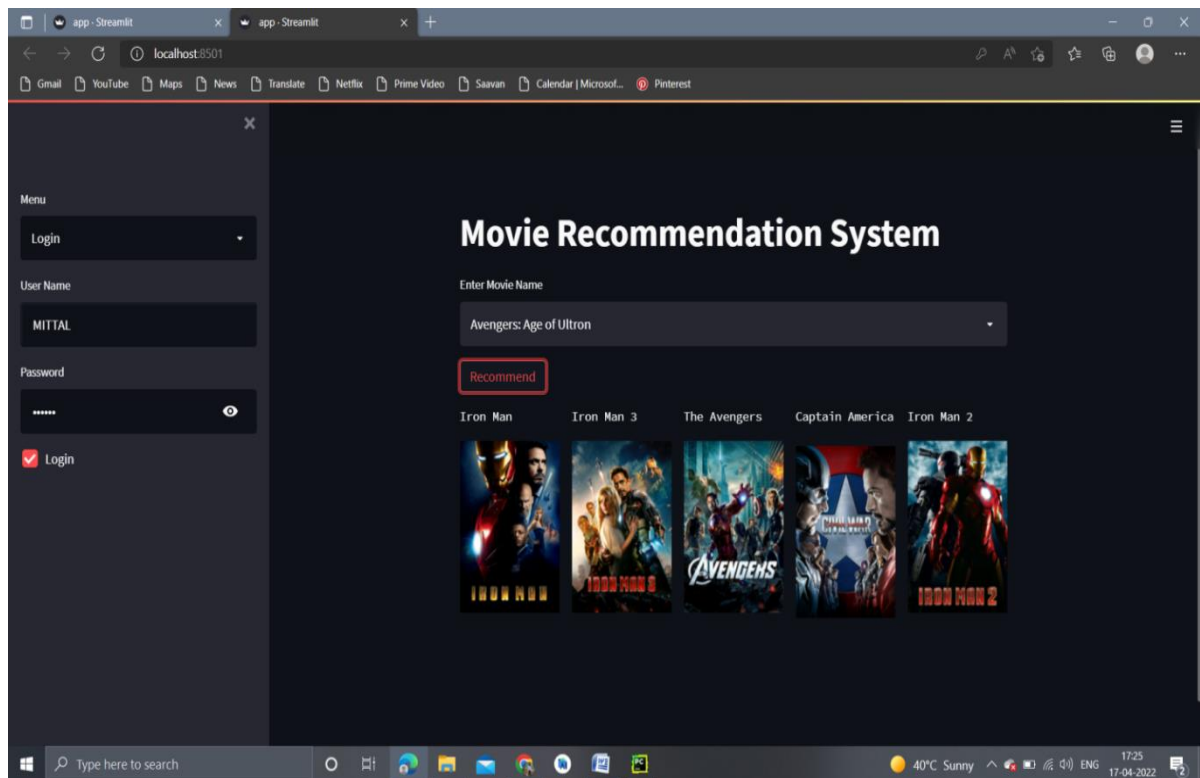


Figure 7: Activity Diagram

Here, we have a user as the starting point which signs in then logs into the system logs then, searches for a movie.

After this, the recommendation engine will sort the movies according to user, and then show the result. After that, the user will select the movie he/she is looking for and then the engine will process the new information, i.e., the selected movie and then it will show the final recommended movie list to the user.

2.7 Template Design



Chapter 3: Implementation and Testing

3.1 Software and Tools

Anaconda distribution:

Anaconda is a free and open-source distribution of the Python programming language for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management systems 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.

- **Jupyter Notebook:**

The Jupyter Notebook is an open-source web application that you can use to create and share documents that contain live code, equations, visualizations, and text.

Jupyter Notebook is maintained by the people at Project Jupyter. Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R.

PyCharm:

- **Python :**

For computation and analysis we need certain python libraries which are used to perform analytics. Packages such as SKlearn, NumPy, pandas, Matplotlib, Flask framework, etc. are needed.

- **SKlearn:**

It features various classification, regression and clustering algorithm including for vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

- **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.

Unlike NumPy library which provides objects for multi-dimensional arrays, Pandas provides in-memory 2d table object called Data frame.

- **Streamlit:**

Streamlit is an open-source python library that allows swift custom web app building revolving data science, machine learning, and much more. Streamlit aims at building and deploying applications without the necessity of any web development knowledge.

- **Pickel:**

Pickle in Python is primarily used in serializing and deserializing a Python object structure.

In other words, it's the process of converting a Python object into a byte stream to store it in a file/database, maintain program state across sessions, or transport data over the network.

- **Requests:**

The Requests library provides a simple API for interacting with HTTP operations such as GET, POST, etc.

The methods implemented in the Requests library execute HTTP operations against a specific web server specified by its URL.

- **Hashlib:**

Hashlib Python provides the most secure hash values which is very good in storing sensitive data, like passwords or bank details, so if the database is hacked or something other happens no one will be able to recreate the information from the hash values created by the hashlib python package. Even the site owner will not understand what this sensitive data means.

- **Sqllite3:**

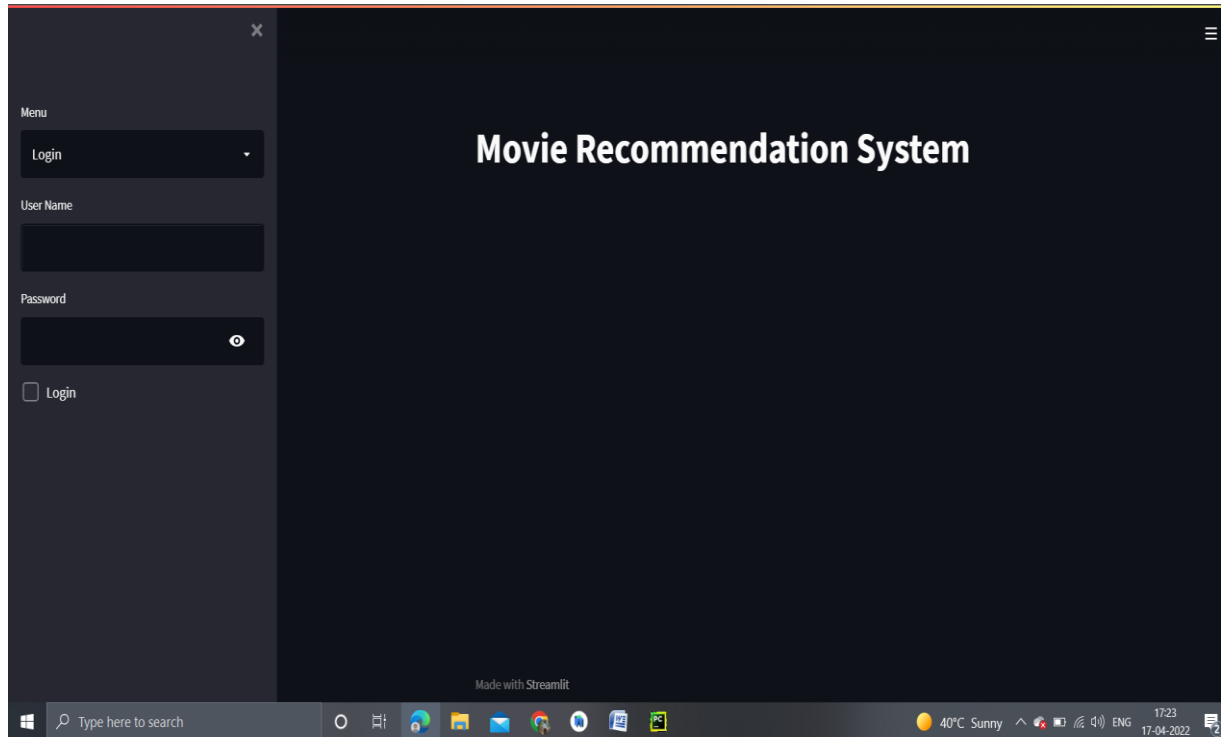
SQLite is a C library that provides a lightweight disk-based database that doesn't require a separate server process and allows accessing the database using a nonstandard variant of the SQL query language.

- **Pandas:**

Pandas is a python library used for data manipulation and statistical analysis. It is a fast and easy to use open-source library that enables several data manipulation tasks. These include merging, reshaping, wrangling, statistical analysis and much more.

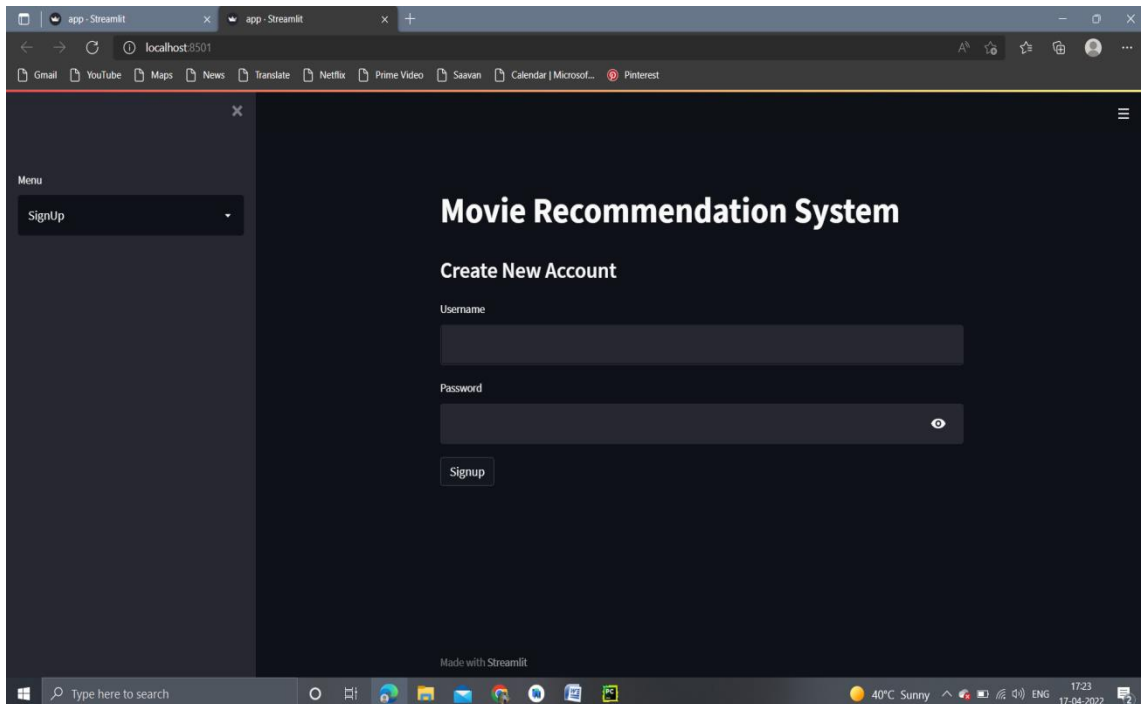
3.2 User Interface and Implementation:

- **Movie Recommendation System:**



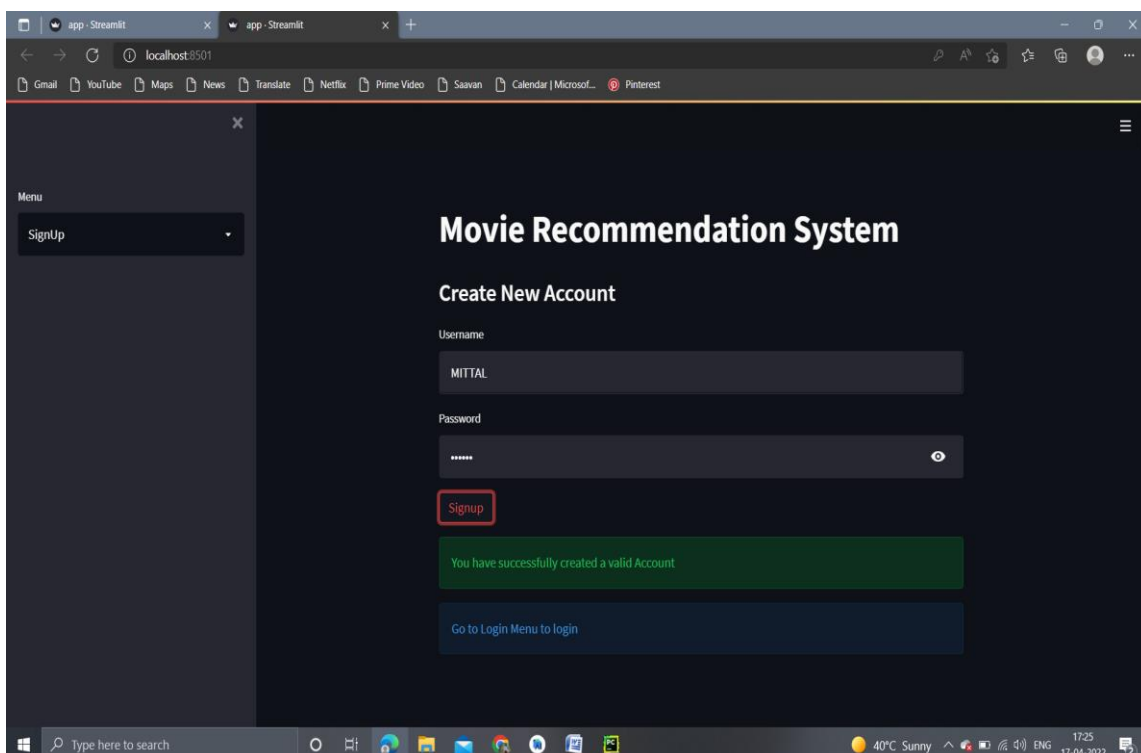
This is website view of movie recommendation system which we made. You can see left-hand side there is one Menu in which there are two option login and sign up.

- **Sign Up Page:**



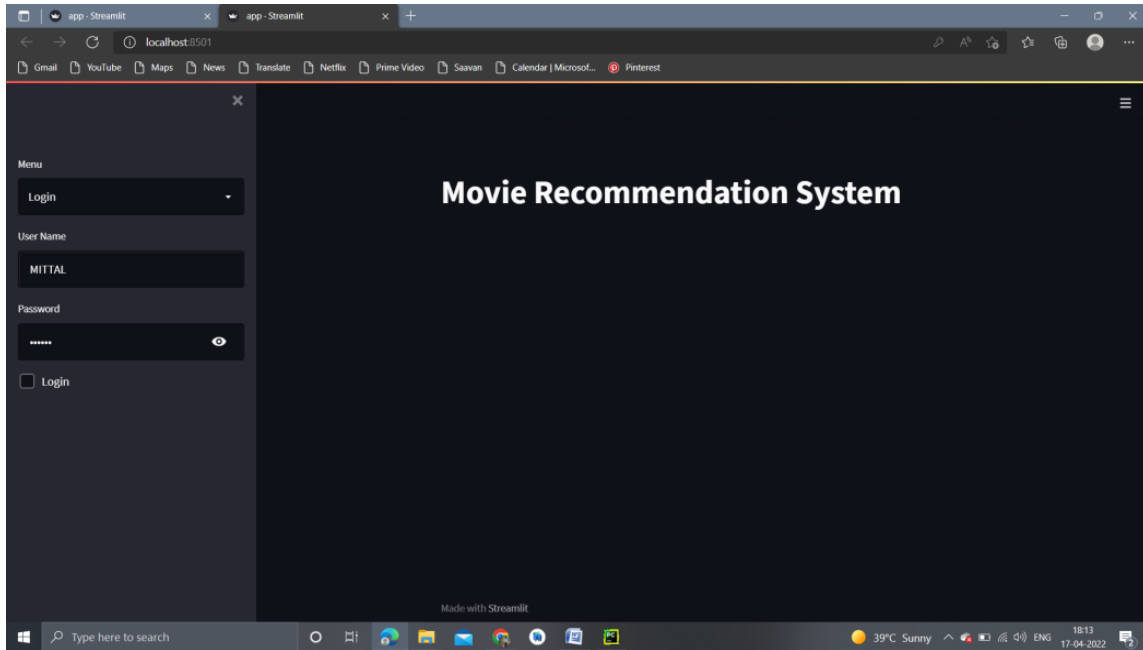
If you go in Menu and choose Signup option it will show you this output. To create a new account, you need to Sign Up.

- **Created Account Successfully:**



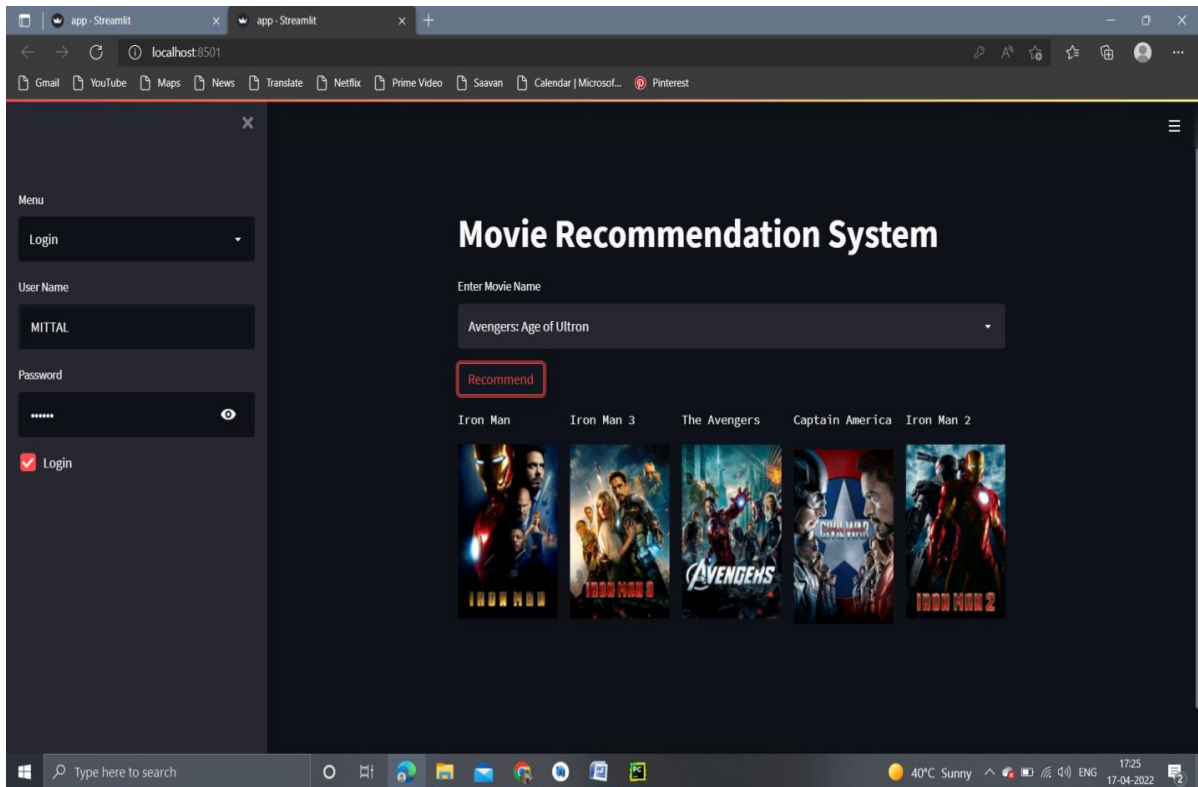
In the above screenshot you can see that we created an account successfully by entering our username and password. Clicking up on the Signup button we created a valid account successfully, a message will be shown.

- **Log In Page:**



After successfully creating an account, we will go to the Menu and choose the Login option. There we have to enter the correct Username and password. If you have just made new account your password will be shown automatically. There you only have to enter the correct username or it will show an error message to enter a valid password or username.

- **Recommendation Page:**



In login you have to check tick on Login checkbox. After successfully login we can get access of recommendation system.

In which you have to enter movie name and hit the Recommend Button you can get top 5 similar movies to the movie you enter.

Here we enter Avengers: Age of Ultron in name of movie. We want top 5 movies similar to it. Recommended system recommends top 5 movies and its poster below.

3.3 Testing using Use Case:

TC-01

Test Case ID	TC-01
Module to be tested	Login Page
Assumptions	---
Test Data	Check When passing a correct username and invalid password
Test Steps	1. Enter valid username 2. Enter incorrect password 3. Check on the login box
Expected results	User should not log in and should show proper error message
Result	Pass
Comments	On entering an invalid password, user not able to login application. It is required to enter the correct username and password.

TC-02

Test Case ID	TC-02
Module to be tested	Sign up
Assumptions	---
Test Data	Check if the user is able to sign up to the system if he enters the new user's name
Test Steps	1. Choose the sign-up option from the drop-down menu on

	<p>the side bar</p> <ol style="list-style-type: none">2. Enter the username and password3. Click on the sign-up button
Expected results	If the user-name is not taken the message of the successful registration should be displayed on the screen
Result	Pass
Comments	After the successful sign-up user is displayed the success message and asked to go to the login page

TC-03

Test Case ID	TC-03
Module to be tested	Sign-up page
Assumptions	---
Test Data	Check if user-name entered for the registration is already taken
Test Steps	<ol style="list-style-type: none">1. Choose the sign-up option from the drop-down menu on the side bar2. Enter the username which has been taken and password3. Click on the sign-up button
Expected results	The user should not be able to register for the system and displayed the error message to choose the different username
Result	Pass
Comments	When the username which has already been taken by someone else entered the error message has been displayed and user is asked to enter the different name

TC-04

Test Case ID	TC-04
Module to be tested	Recommendation page
Assumptions	User is already logged in
Test Data	Check that when user entered the movie name and click on recommend button, she gets the 5 recommended movies
Test Steps	<ol style="list-style-type: none">1. Enter the movie you have watched or like2. Click the recommended button
Expected results	User should get the list of 5 movies with their posters based upon the movie he entered
Result	Pass
Comments	On click the recommend button user is getting the result based upon the movie he entered with their posters.

Chapter 4: Conclusion and Future Work

4.1 Conclusion:

In this project we have implemented and learn the following things such as-

- Data Preprocessing
- Building Movie Recommendation System
- To find the Similarity Scores and Indexes.
- Compute Distance Between Two Vectors
- Cosine Similarity
- Content Based Filtering

To conclude, a recommender system powered by content-based filtering performed using the cosine similarity algorithm can make better recommendations for users by suggesting them movies that have similar key features like the IMDb votes, average IMDb rating, genre, release year, casts, directors, user tags, etc.

4.2 Future Work:

There are quite methods we can improve existing system. We would like to work on that in future.

- Use updated database.
- Here the search is case sensitive it can be improved.
- We can use different approaches which are more effective like hybrid filtering.
- We have plenty of features so the deep learning can beat the KNN algorithm in terms of accuracy.
- We should make mobile application of this system.

Various techniques are also discussed which works in giving recommendation. So, scope of any recommender system is to build a model in such a way that their user gets proper recommendation and efficiency of the system is maintained.

References:

<https://www.kaggle.com/tmdb/tmdb-movie-metadata>

https://youtu.be/HU_kd-1ulkQ

https://youtu.be/1xtrIEwY_zY

<https://blog.jcharistech.com/2020/05/30/how-to-add-a-login-section-to-streamlit-blog-app/>

https://link.springer.com/chapter/10.1007/978-981-13-1927-3_42

<http://www.jicrjournal.com/gallery/98-jicr-september-3208.pdf>