

“Next Up - Movie Recommendation System”

A PROJECT REPORT

Submitted by

Goutam Verma and Kashish Ahuja

Under the guidance of

Prof. Basant Namdeo

**In *partial fulfillment for the VIIth Semester Examination*
of**

M.Tech(IT)

of

**International Institute of Professional Studies
Devi Ahilya Vishvavidyalaya, Indore**

2022

DECLARATION

We hereby declare that the project entitled “Next up - Movie Recommendation System” which is submitted by us for partial fulfillment for the award of the degree of Master of Technology(IT) 5 Years of International Institute of Professional Studies, Devi Ahilya Vishwavidyalaya, Indore, comprises our own work and due acknowledgment has been made in the text to all other material used.

Signature of Students:

Date: 09/12/2022

Place: Indore

International Institute of Professional Studies

Devi Ahilya Vishwavidyalaya, Indore, M.P.

CERTIFICATE FROM GUIDE

It is to certify that dissertation on “Next up - Movie Recommendation System”, submitted by *Goutam Verma* and *Kashish Ahuja* for the partial fulfillment for the award of the degree of Master of Technology(IT) 5 years of International Institute of Professional Studies, Devi Ahilya Vishwavidyalaya, Indore has been completed under my supervision and the work is carried out and presented in a manner required for its acceptance.

Project Guide

Signature:

Name: Basant Namdeo

Date: 09/12/2022

International Institute of Professional Studies
Devi Ahilya Vishwavidyalaya, Indore, M.P.

CERTIFICATE

It is to certify that we have examined the project on "Next up - Movie Recommendation System", submitted by Goutam Verma and Kashish Ahuja for the partial fulfillment of the award of the degree of Master of Technology(IT) 5 years of International Institute of Professional Studies, Devi Ahilya Vishwavidyalaya, Indore and hereby accord our approval of it as a study carried out and presented in a manner required for its acceptance.

Internal Examiner

Signature: _____

Name : _____

Date : _____

External Examiner

Signature: _____

Name : _____

Date : _____

Table of Content	Page No.
1 Introduction	
1.1 Relevance of the project.....	10
1.2 Problem Description.....	13
1.3 Project Aim.....	14
1.4 Project Scope.....	14
1.5 Methodologies.....	15
1.6 Tools and Technologies.....	16
2 Feasibility Study	
2.1 Existing System and Limitation.....	17
2.2 Proposed System with Objectives.....	17
2.3 Feasibility Study.....	17
2.3.1 Economic Feasibility Study.....	18
2.3.2 Technical Feasibility Study.....	18
2.3.3 Behavioral Feasibility Study.....	18

3 System Analysis

3.1 Project Analysis.....	19
3.2 Preliminary Investigation.....	19
3.3 Requirement Specification	
3.3.1 Hardware Requirement.....	20
3.3.2 Software Requirement.....	20

4 System Design

4.1 Entities Definition.....	22
4.2 Use Case Description.....	22
4.3 Use Case Diagram.....	24
4.4 Workflow Diagram.....	25
4.5 Web Flow Diagram	26
4.6 Entity-relationship Diagram.....	27
4.7 User Interface.....	28

5 System Testing

5.1 Unit Testing.....	34
5.2 Integration Testing.....	34
5.3 System Testing.....	35
5.4 Test Cases.....	35
5.4.1 Continuous Integration.....	35
5.4.2 Continuous Deployment.....	35

6 Conclusions

6.1 Findings.....	37
6.2 Endpoints Output.....	38
6.3 Limitations.....	39
6.4 Scope for Future Prospects.....	39

7 Bibliography & References.....40

1. Introduction

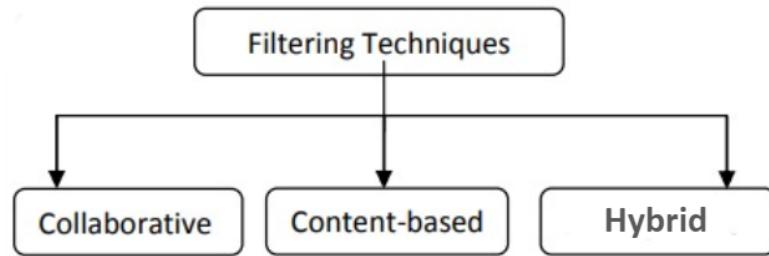
1.1. Relevance of the project

A recommendation system or recommendation engine is a model used for information filtering where it tries to predict the preferences of a user and provide suggestions based on these preferences. These systems have become increasingly popular nowadays and are widely used today in areas such as movies, music, books, videos, clothing, restaurants, food, places, and other utilities. These systems collect information about a user's preferences and behavior and then use this information to improve their suggestions in the future. Movies are a part and parcel of life. Various types of movies exist, some for entertainment, some for education, some for children, some for horror, and some for action. There are a variety of movie genres, such as comedy, thriller, animation, action, etc. Another way to distinguish between movies can be by releasing year, language, director, etc. Watching movies online, there are several movies to search for in our most liked movies. Movie Recommendation Systems help us to search our preferred movies among all of these different types of movies and hence reduce the trouble of spending a lot of time searching for our favorable movies. So, it requires that the movie recommendation system should be very reliable and should provide us with the recommendation of movies that are the same or most match our preferences. The movie Recommendation system is very powerful and important. A large number of companies are making use of recommendation systems to increase user interaction and enrich a user's shopping experience. Recommendation systems have several benefits, the most important being customer satisfaction and revenue.

Movie recommendation systems use a set of different filtration strategies and algorithms to help users find the most relevant films. The most popular categories of ML algorithms used for movie recommendations include content-based filtering and collaborative filtering systems.

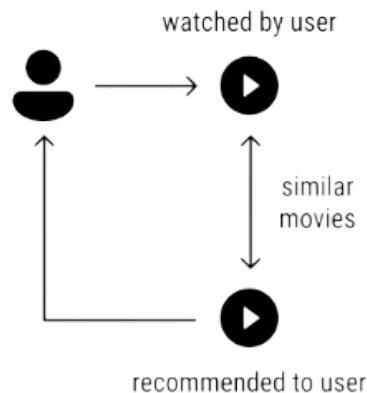
There are majorly three types of recommendation systems:

- Content-based recommendation
- Collaborative filtering based recommendation
- Hybrid recommendation



Content-based recommendation

A filtration strategy for movie recommendation systems, which uses the data provided about the items (movies). This data plays a crucial role here and is extracted from only one user. An ML algorithm used for this strategy recommends motion pictures that are similar to the user's preferences in the past. Therefore, the similarity in content-based filtering is generated by the data about past film selections and likes by only one user.



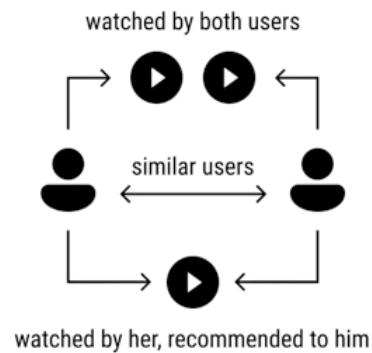
Collaborative-based recommendation

As the name suggests, this filtering strategy is based on the combination of the relevant user's and other users' behaviors. The system compares and contrasts these behaviors for the most optimal results. It's a collaboration of multiple users' film preferences and behaviors.

Collaborative filtering algorithms are divided into two categories:

User-based collaborative filtering: The idea is to look for similar patterns in movie preferences in the target user and other users in the database.

Item-based collaborative filtering: The basic concept here is to look for similar items (movies) that target users rate or interact with.



Hybrid recommendation

Hybrid Recommender System is more and more popular currently. Combining collaborative filtering and content-based filtering can be more effective in recent research.

1.2. Problem Description

Many times users face the problem of getting the right content to watch according to their current mood and choice of genre. It takes a lot of time to find the right content which usually leads to irritation and not wanting to watch anymore.

The scope of this project is to provide accurate movie recommendations to users. The goal of the project is to improve the accuracy, quality, and scalability of the movie recommendation system compared to pure approaches. This is done using the Hybrid approach by combining content-based filtering and collaborative filtering. To eradicate the overload of data, a recommendation system is used as an information filtering tool in social networking sites. As a result, there is a great deal of room to explore this area in order to improve the scalability, accuracy, and quality of movie recommendation systems. Movie recommendation systems are very important and powerful. But, due to the problems associated with a pure collaborative approach, movie recommendation systems also suffer from poor recommendation quality and scalability issues.

For building a recommender system from scratch, we face several different problems. Currently, there are a lot of recommender systems based on the content of information, so what should we do if the website does not have enough information about the movie? After that, we will solve the representation of a movie, which is how a system can understand a movie. That is the precondition for comparing similarities between two movies. Movie features such as genre, actor, and director are a way that can categorize movies.

So we get these questions:

- How to recommend movies when there is no movie information.
- What kind of movie features can be used for the recommender system?
- How to calculate the similarity between two movies.

1.3. Project Aim

The aim of the project is to:

- Offer generalized recommendations to every user based on movie popularity, genre, and year.
- Gives personalized recommendations based on the user's choice of genre and cast.
- Suggest similar movies that have a higher probability of being liked based on the movie selected by the user.
- Providing related content out of the relevant and irrelevant collection of items to users of online service providers.
- Improve the Quality of the movie Recommendation system.
- Improving the Accuracy of the recommendation system.
- Improving Scalability.
- Enhancing the user experience

1.4. Project Scope

The objective of this project is to provide accurate movie recommendations to users. The goal of the project is to improve the movie recommendation systems in terms of accuracy, quality, and scalability. This is done using the **content-based** approach of recommendation. To eradicate the overload of data, the recommendation system is used as an information filtering tool in social networking sites. Hence, there is a huge scope for exploration in this field.

1.5. Methodologies

In order to achieve the goal of the project, the first process is to do enough background study, so the literature study will be conducted. The whole project is based on a big amount of movie data so we choose the quantitative research method.

For computing similarity between the different movies in the given dataset efficiently and in the least time and to reduce the computation time of the movie recommender engine we used the cosine similarity measure.

Agile Methodology

- **Collecting the data sets:** import the TMDB datasets (tmdb_5000_movies.csv and tmdb_5000_credits.csv) and convert them into a data frame.
Datasets' link: <https://www.kaggle.com/datasets/tmdb/tmdb-movie-metadata>
- **Data analysis:** make sure that the collected data sets are correct and analyze the data in the CSV files. i.e. checking whether all the column fields are present in the data sets.
- **Pre-processing of data:** pre-process the data frame by selecting important columns and doing some processing on the columns (convert columns in a proper format, pick the top 3 casts from the cast column, pick director from the crew column), and merge all the columns into one tags column.
- **Algorithms:** In our project, we have only two algorithms one is text-vectorization and the other is cosine similarity used to build the machine learning recommendation model.
- **Training and testing the model:** once the implementation of the algorithms is completed, we have to train the model to get the result. We need to test it several times to check if the model is recommending the correct set of movies.
- **Improvements in the project:** In the later stage we can implement different algorithms and methods for a better and quick recommendation. We have used Annoy (Approximate Nearest Neighbors) mechanism for quickly getting similar movies.

Annoy link: <https://github.com/spotify/annoy>

1.6. Tools and Technologies

- Tech Stack and Software requirements:
 1. Frontend: HTML5, CSS3, JavaScript, BootStrap, jQuery
 2. Backend: Python flask
 3. Database: PostgreSQL, SQLite3
 4. ML model: Jupyter Notebook
 5. IDE: PyCharm
 6. Version Control: Git
 7. Deployment: Heroku
- Language specific requirements:
 1. Python: NumPy, Pandas, ast (Abstract Syntax Trees), pickle
 2. ML: NLTK (Natural Language Toolkit), Sklearn (scikit-learn)
 3. SQLite: sqlite3, SQLAlchemy

The code for the project can be found on GitHub.

Repository link: <https://github.com/kashishahuja2002/Minor-Project-Next-Up>

2. Feasibility Study

2.1. Existing System and Limitations

There are a few existing systems somewhat similar to our project. Example: pick a movie for me, IMDB recommendations.

Link: <https://pickamovieforme.com/>, <https://www.imdb.com/list/ls063596142/>

While both of them are great recommendation systems. Some of the noticeable limitations are

- Answering lots of questions before getting a recommendation.
- One movie recommendation at a time.
- No personalized recommendations.
- Recommendation based on current mindset, not on the similarity of the movie preferred.
- Bad user interface

2.2. Proposed system with objectives

The objective of our recommendation system to eliminate the current limitations are:

- Recommending multiple similar movies at a time.
- Gives personalized recommendations based on the user's choice of genre and cast.
- Suggest similar movies that have a higher probability of being liked based on the movie selected by the user.
- Providing related content out of the relevant and irrelevant collection of items to users of online service providers.
- Enhancing the user experience

2.3. Feasibility Study

A feasibility study is an analysis done to determine the viability of a project from an economic, legal, and technical perspective. Simply put, it gives us an insight into whether a project is doable or worth the investment.

Feasibility is the study of impact, which happens in the organization through the development of a system. The impact can be either positive or negative. When the positives nominate the negatives, then the system is considered feasible. The Recommendation System is purely feasible and it fulfills all three conditions given below:

1) Economic Feasibility

- a) This assessment performs a cost/ benefits analysis of the project before the financial resources are allocated. This type of study gives a clear-cut idea of project credibility (viability) as well as the economic benefits to the organization from the project.
- b) The development of this application is highly feasible. The organization needed not to spend much money on the development of the system. The only thing to be done is to make an environment for development with effective supervision. If we are doing so, we can attain the maximum usability of the corresponding resources. Even after the development, the organization will not be in a condition to invest more in the organization. Therefore, the system is economically feasible.

2) Technical Feasibility

The criteria for the technical feasibility of the Movie Recommendation System are :-

- a) **Easy to follow** - No previous knowledge required. The user will enter the movie name and an answer will be generated and displayed on the page.
- b) **Maintainability** - Since the project is developed keeping all the requirements in mind, therefore, no additional updatations will be required.
- c) **Portability** - The project is built using Version Control (Github) so anyone who has privileges can access the code using his/her machine with a stable internet connection.
- d) **Extensibility** - The project architecture is designed by keeping in mind the MVT (Model View and template) work pattern, therefore new functionalities can be easily integrated with the existing projects thus making the project extensible.
- e) **Reusability** - The extensibility and portability of this project would make it reusable software.
- f) **Serviceability** - This project will try to provide the maximum possible service to the user.

3) Behavioral Feasibility

- a) It evaluates and estimates the user attitude or behavior towards the development of a new system. This system is very user-friendly and its interface is simple enough so that everyone can utilize it.

3. System Analysis

System Analysis is a detailed study of the various operations performed by a system and their relationships within and outside of the system.

3.1. Project Analysis

- i. Project analysis is the process of examining the aspects of a project in detail. This is mainly to see to it that the project runs as expected by the Users. We have done studies to examine various aspects of the Movie Recommendation System.
- ii. Use-case diagrams and other UML diagrams as required.

3.2. Preliminary Investigation

The questionnaire was to know:

- How many people watch movies/series on OTT platforms?
- How do they select content to watch?
- Does the recommendations by OTT platforms works for them?
- How many users are interested in personalized recommendations?
- How many users are willing to pay for a recommendation system?
- If not, what do users do to solve their problems?

Statistical Analysis

Statistical analysis was done based on a questionnaire to know:

- Problems that users face in the currently available system.
- If the project is hosted will it be able to cover its operational cost?
- Features that can be added in the future.

Questionnaire

1. Do you watch movies or series on OTT (over-the-top platforms) or by downloading them?
Yes/No
2. In which age range do you fall?
 < 16
 16 - 23
 23 - 30
 30 - 50
 > 50

3. How do they select content to watch?
 Watch anything that appears
 Watch the latest content
 Watch according to genre
 Watch according to cast
 Watch according to popularity
 Watch according to the plot/story
4. Do you watch and like the content recommended by the OTT platforms?
Yes/No
5. Would you be interested in generalized and personalized recommendations?
Yes/No
6. Are you willing to pay for a recommendation system?
Yes/No
7. If not, what do you do to solve their problems?
 Watch anything
 Use free recommendation platforms

3.3. Requirement Specification

3.3.1 Hardware Requirements

- A PC with Windows/Linux OS
- Processor with 1.7-2.4GHz speed
- Minimum of 4GB RAM
- 2GB Graphic card

3.3.2. Software Requirements

- Text Editor (VS-code/WebStorm)
- Anaconda distribution package (PyCharm Editor)
- Python libraries

Anaconda distribution

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

Python libraries

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

SKlearn: It features various classification, regression and clustering algorithms including support 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 the NumPy library which provides objects for multi-dimensional arrays, Pandas provides an in-memory 2d table object called a Data frame.

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. Pandas provide an in-memory 2d table object called a Data frame.

Flask: It is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug.

4. System Design

4.1. Entities Definition

Next up is a movie recommendation system that offers generalized recommendations to every user based on movie popularity, genre, and year. The model also gives personalized recommendations based on the user's choice of genre and cast. Finally, the system suggests similar movies have a higher probability of being liked based on the movie selected by the user.

The entities and attributes of the movie recommendation system are:

- Movies
- Users
- Genre
- Cast
- Crew

4.2. Use Case Description

The use case is a description of all the ways an end-user wants to “use” a system. These “uses” are like requests of the system, and use cases describe what that system does in response to such requests. In other words, use cases describe the conversation between a system and its user(s), known as actors. This **Movie Recommendation System** has six major actors that describe how end-users want to use it.

1. Sign-up/Sign-in:

It is a user-initiated actor and its goal is to register new users or help existing users to get logged in. The postcondition of this actor is to register/login the users successfully and with no preconditions.

2. Select Movie:

It is a user-initiated actor, the main aim of this actor is to present recommended movies to users on the behalf of user selection. The precondition for this actor is

to select the genres/movie name/actor with the postcondition of successfully recommending movies.

3. Load/Update Database:

It is a system-initiated actor whose goal is to update/load the database after executing the operations. The postcondition of this actor is to visualize the accurate information to users.

4. View Movie Details:

It is a system-initiated actor, which makes a request to API for movie details(release date, description, casts, crew, etc). The precondition for this actor is to have a movie name and details will be displayed as postcondition.

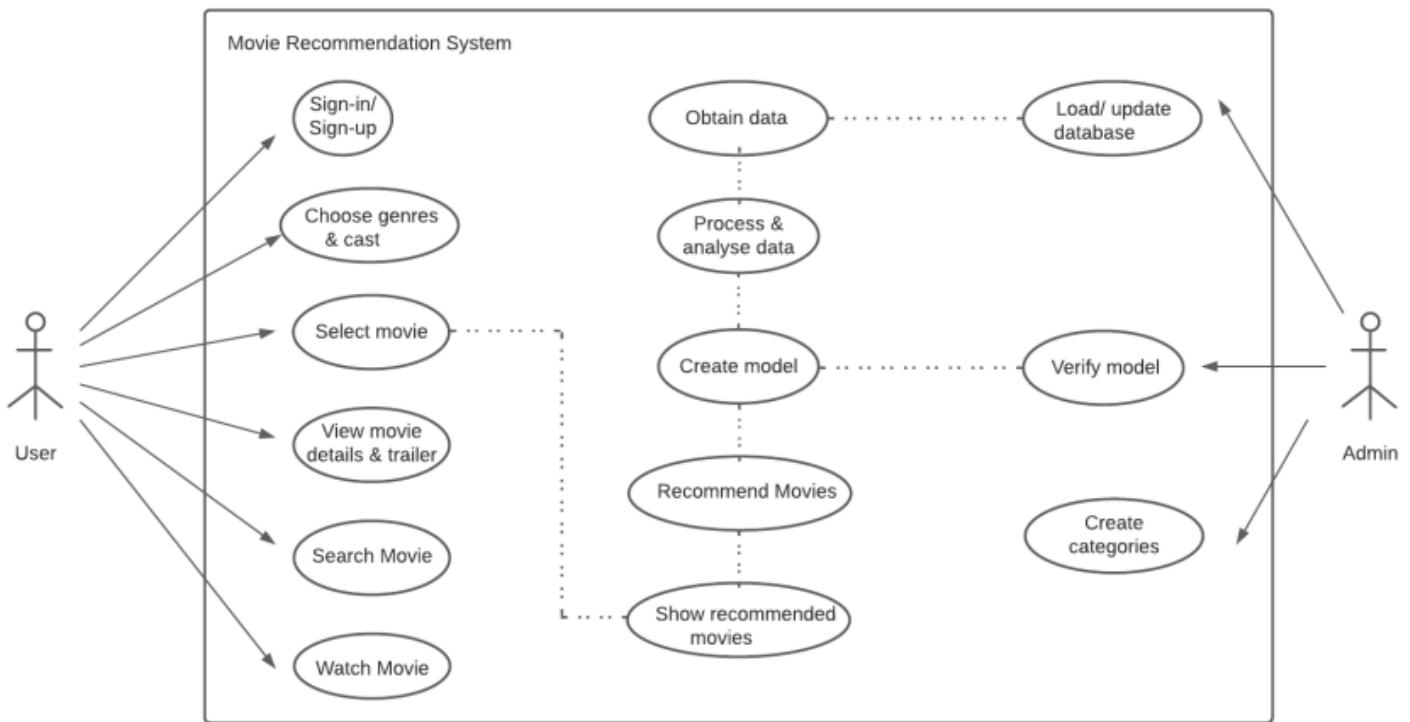
5. Search Movies:

It is a user-initiated actor. This actor completely works on data modeling and presents the recommended movies to users as a post-condition and requires a top 3 genre and top 5 casts as a pre-condition.

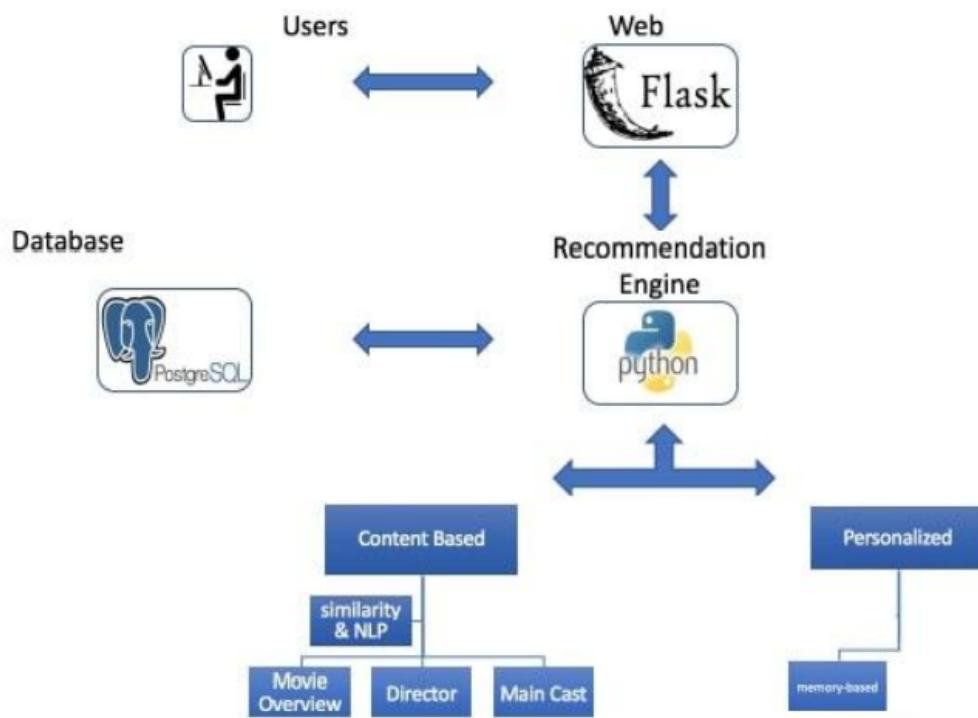
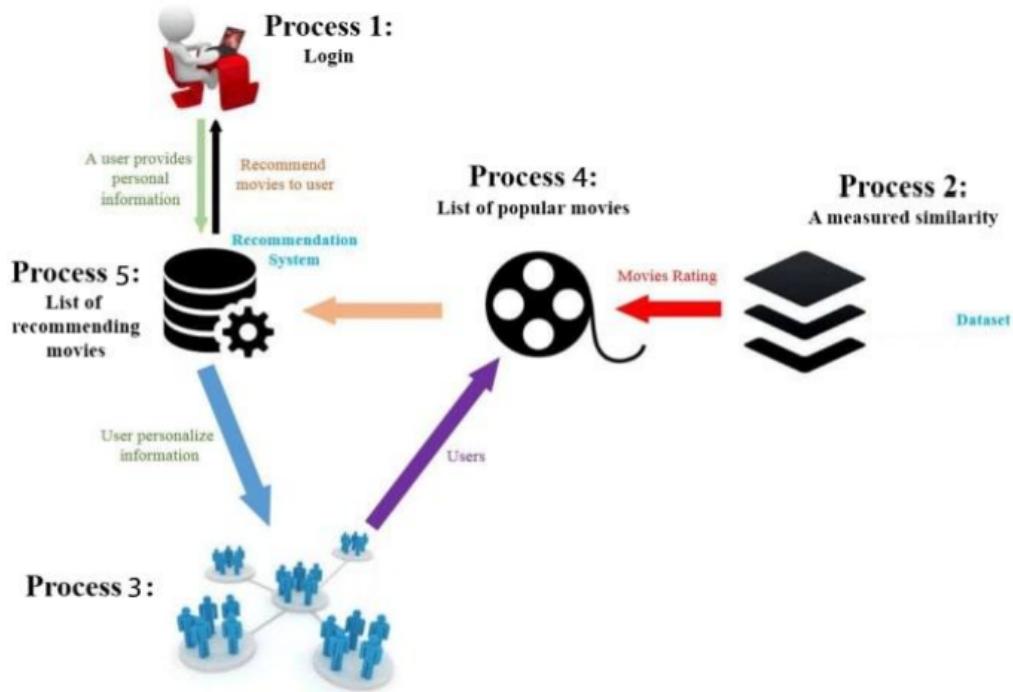
6. Watch Movies:

This actor is under development. It generates a redirecting link for users which gives access to watching movies directly. The precondition for this actor will be the movie name.

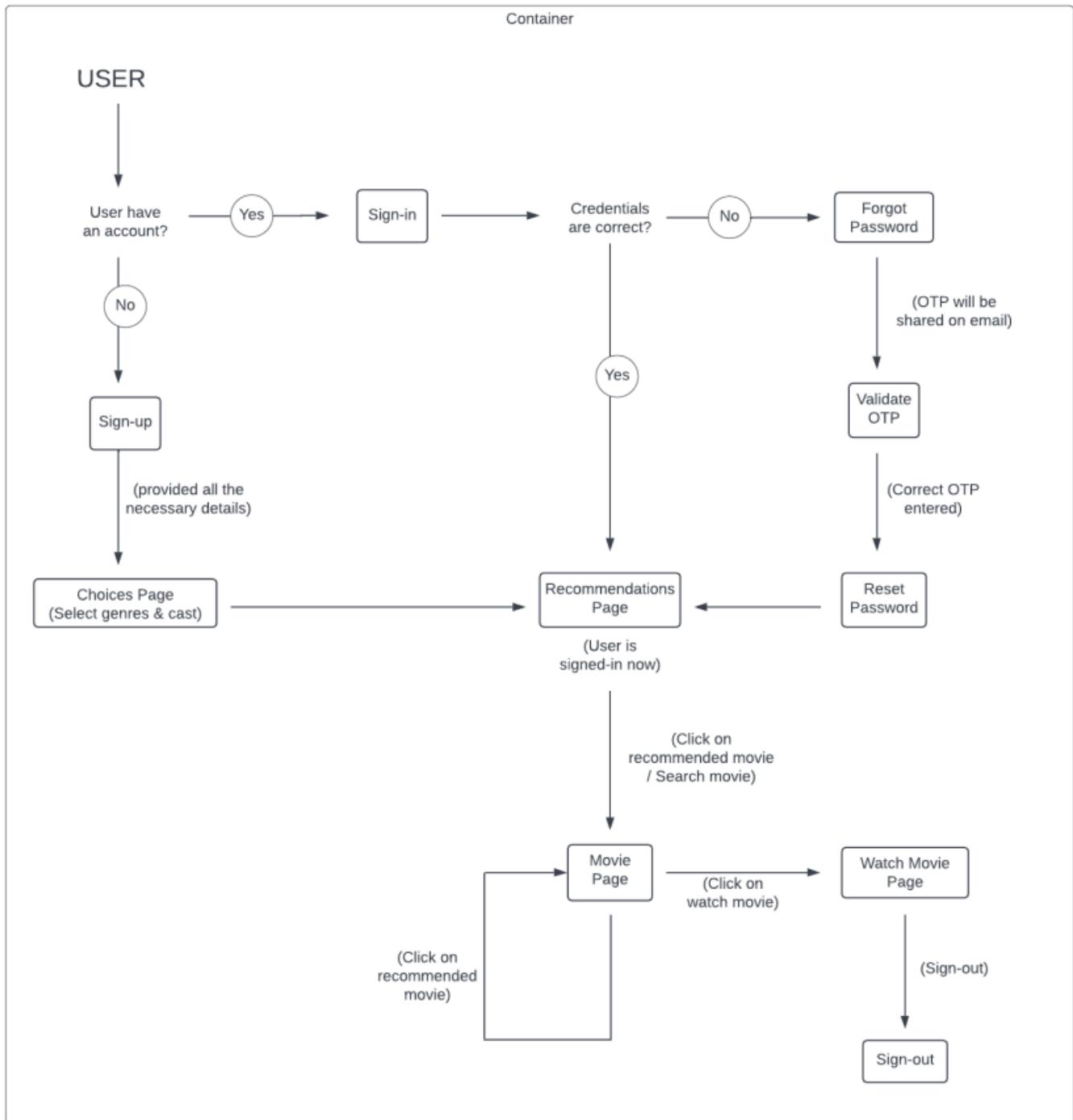
4.3. Use Case Diagram



4.4. WorkFlow Diagram



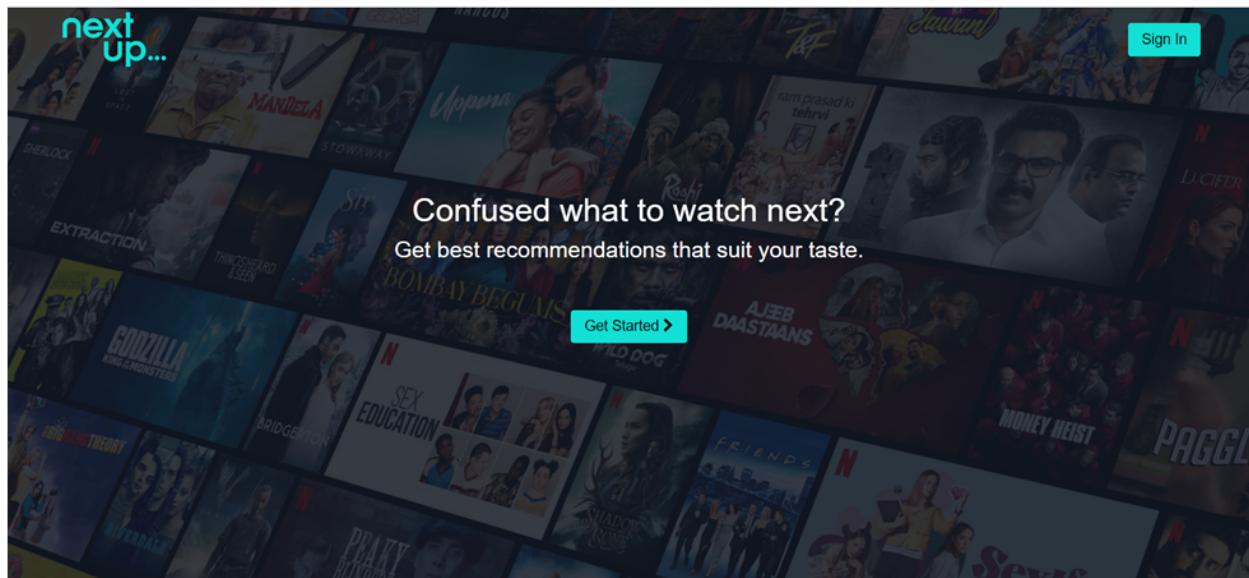
4.5. Web Flow Diagram



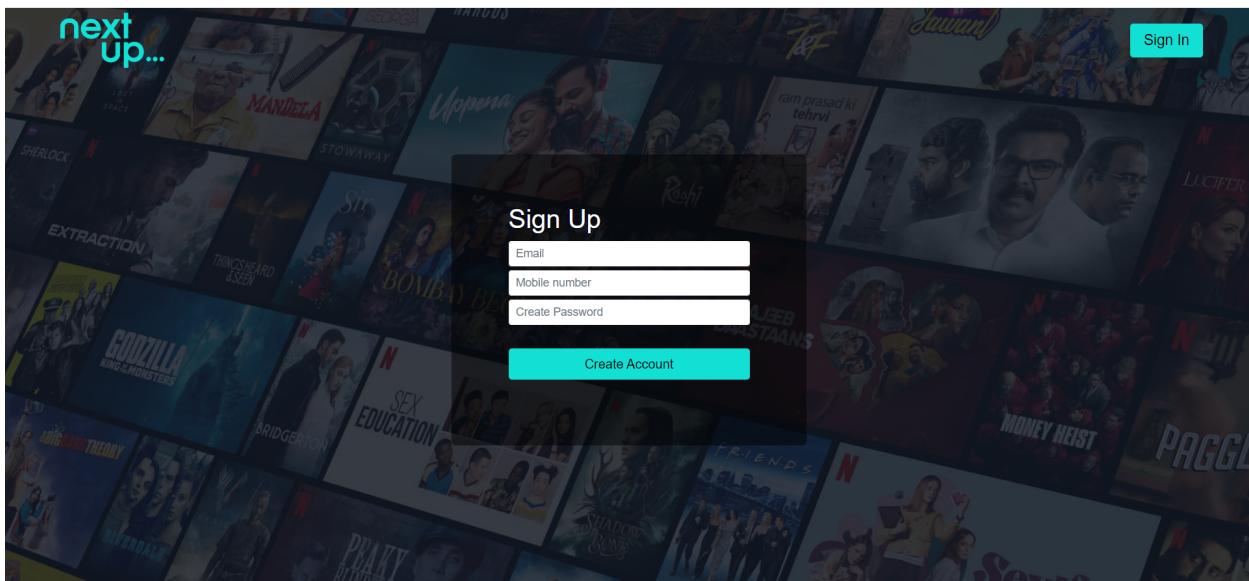
4.6. Entity Relationship Diagram (ER)

4.7. User Interface

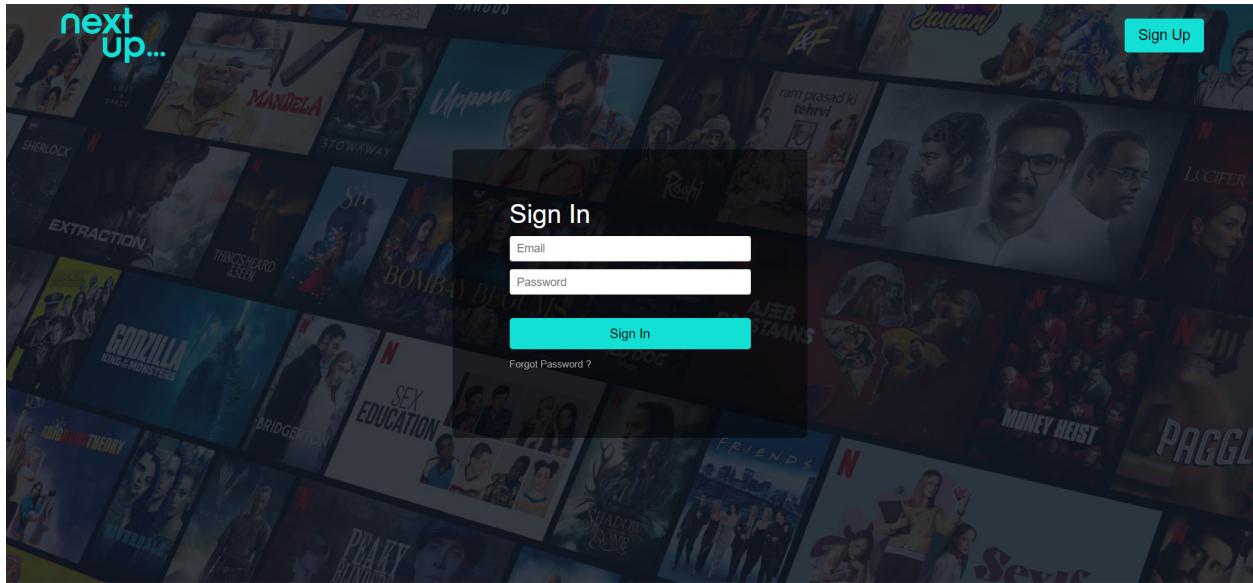
- Landing Page



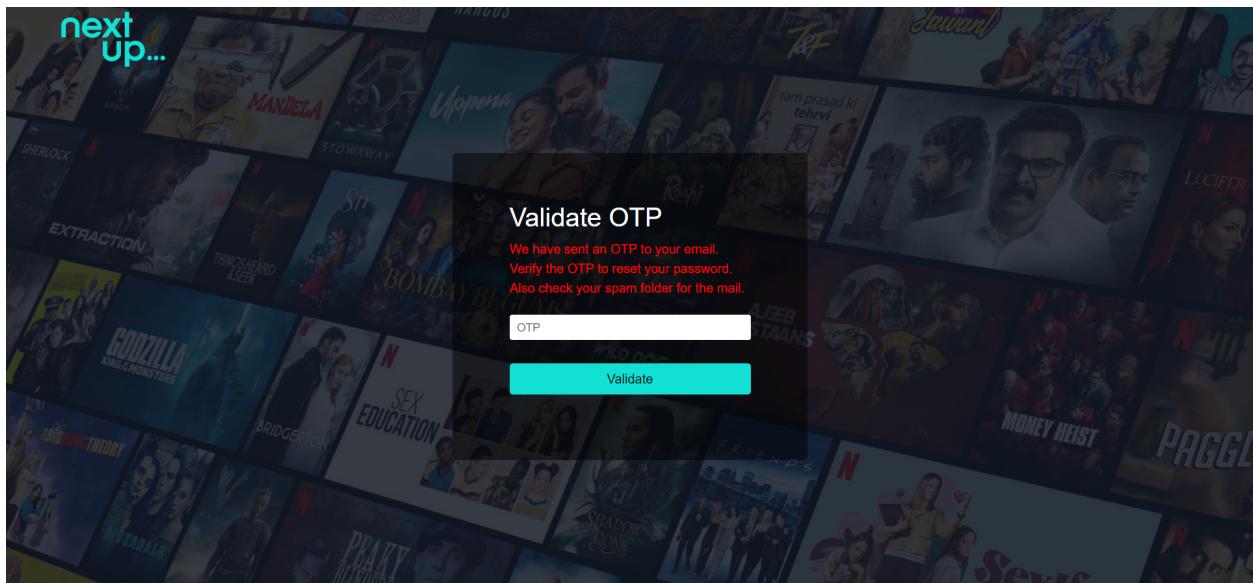
- Sign-up Page



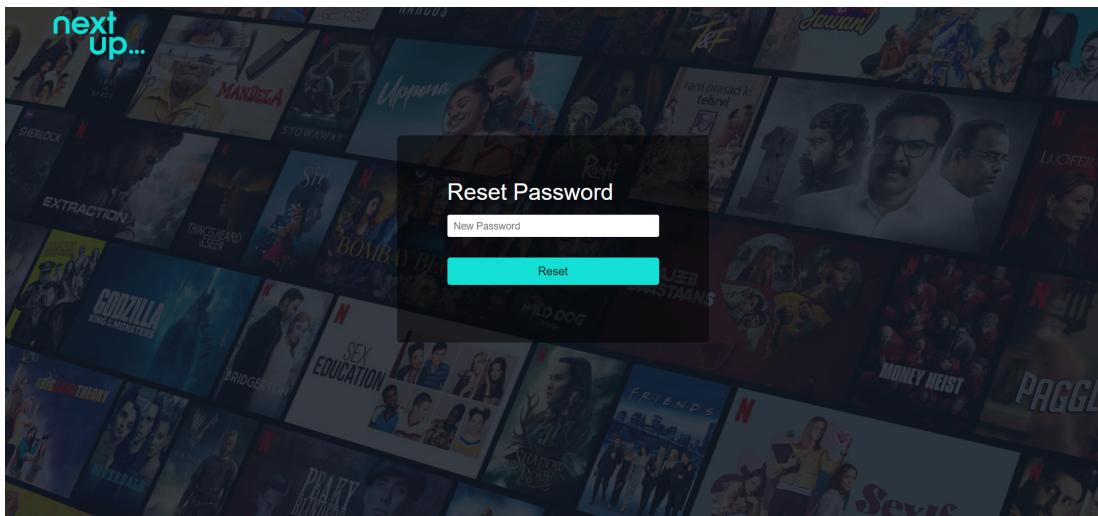
- Sign-in Page



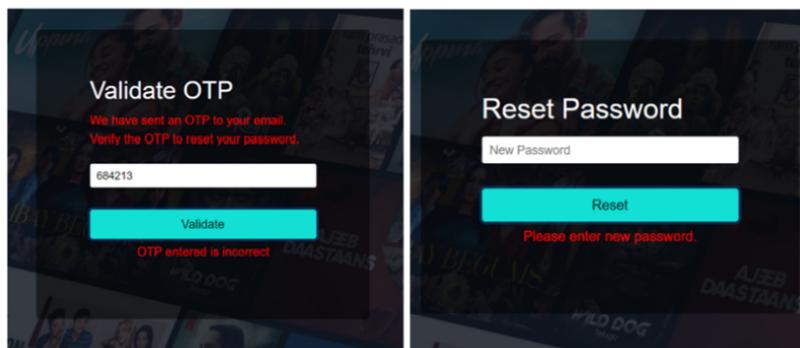
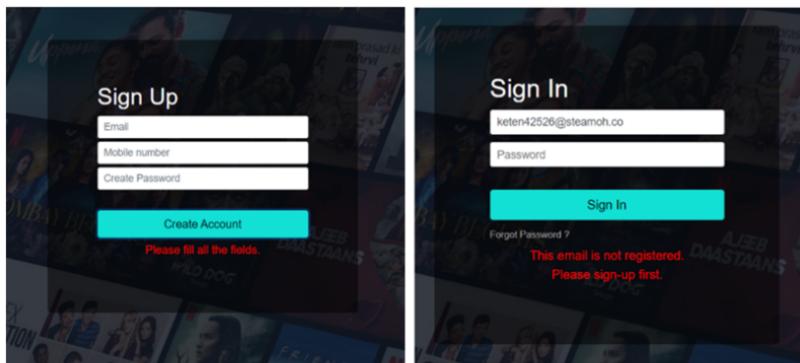
- OTP Validation Page



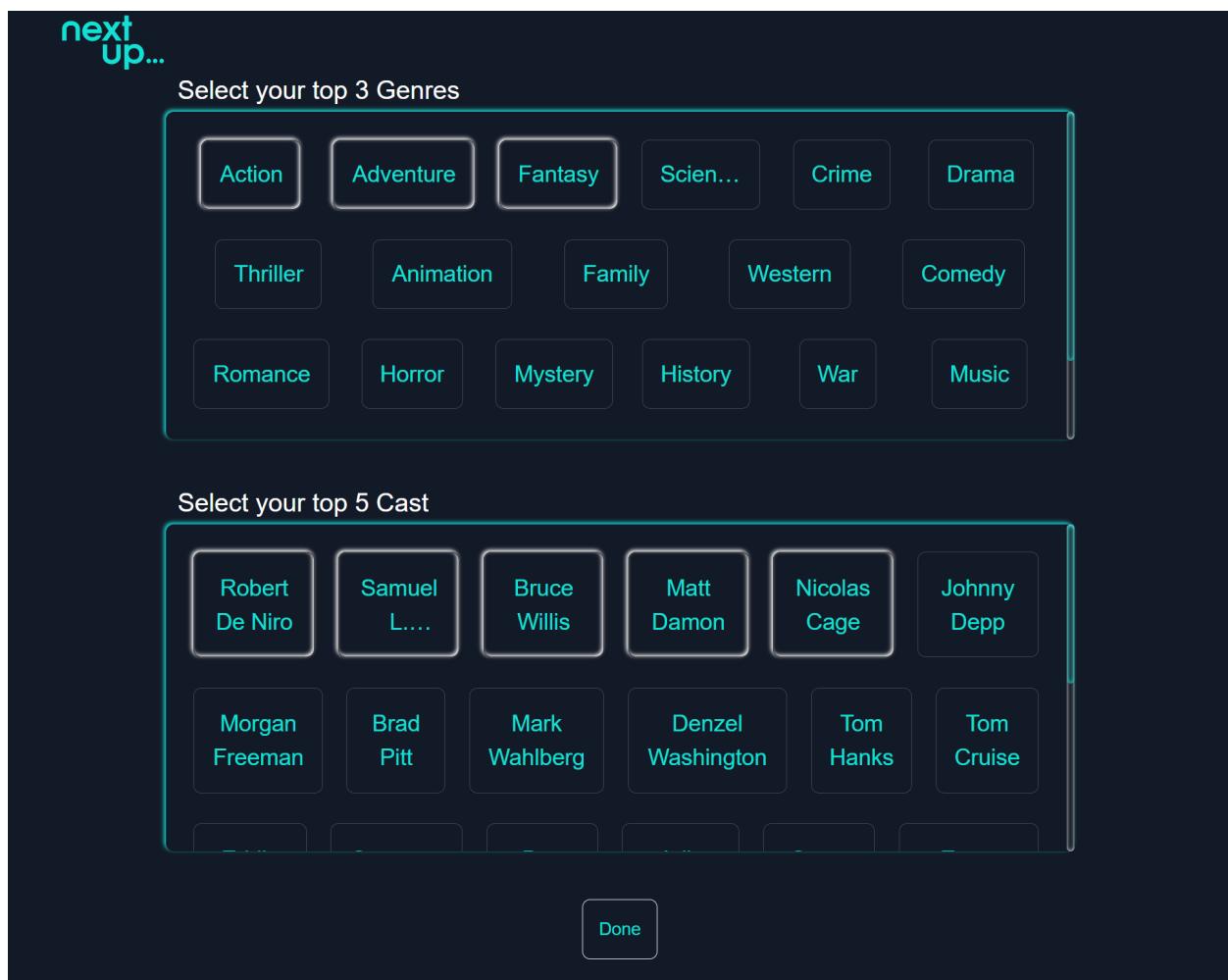
- Reset Password Page



- Restrictions and validations on the sign-up page, sign-in page, forgot password functionality, OTP validation page, and reset password page
 1. All fields not filled
 2. Email account already registered (sign-up), Email account not registered (sign-in)
 3. Incorrect password, OTP incorrect
 4. Email address not entered, OTP not entered, new password not entered



- Choices Page



- Recommendation System Page

**next
up...**

Search Movie  

The Martian

During a manned mission to Mars, Astronaut Mark Watney is presumed dead after a fierce storm and left behind by his crew. But Watney has survived and finds himself stranded and alone on the hostile planet. With only meager supplies, he must draw upon his ingenuity, wit and spirit to subsist and find a way to signal to Earth that he is alive.



Genre Action ▾



DEADPOOL

Deadpool



GUARDIANS OF THE GALAXY

Guardians of the Galaxy



MAD MAX FURY ROAD

Mad Max: Fury Road



JURASSIC WORLD

Jurassic World



PIRATES OF THE CARIBBEAN: THE CURSE OF THE BLACK PEARL

Pirates of the Caribbean: The Curse of the Black Pearl



DAWN OF THE PLANET OF THE APES

Dawn of the Planet of the Apes

Best Of Year 2016 ▾



DEADPOOL

Deadpool



CAPTAIN AMERICA: CIVIL WAR

Captain America: Civil War



BATMAN V SUPERMAN: DAWN OF JUSTICE

Batman v Superman: Dawn of Justice



X-MEN: APOCALYPSE

X-Men: Apocalypse



THE JUNGLE BOOK

The Jungle Book



SUICIDE SQUAD

Suicide Squad

© 2022 NextUp - Designed and Developed by Goutam Verma and Kashish Ahuja

- Movie Page

**next
up...**

Search Movie  

Minions

Before Gru, they had a history of bad bosses

Family . Animation . Adventure . Comedy

Sandra Bullock , Jon Hamm , Michael Keaton , Alison Janney

2015

Minions Stuart, Kevin and Bob are recruited by Scarlet Overkill, a super-villain who, alongside her inventor husband Herb, hatches a plot to take over the world.

[▶ Watch Movie](#)  



More Like This



DESPICABLE ME 2

Despicable Me 2



The Croods



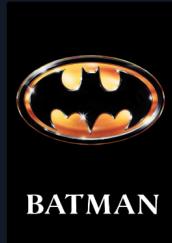
PENGUINS OF MADAGASCAR

Penguins of Madagascar



DESPICABLE ME

Despicable Me



BATMAN



STUART LITTLE 2

Stuart Little 2

© 2022 NextUp - Designed and Developed by Goutam Verma and Kashish Ahuja

- Watch Movie Page

**next
up...**

Search Movie  

Minions



© 2022 NextUp - Designed and Developed by Goutam Verma and Kashish Ahuja

5. System Testing

Once the code is generated, the software program testing begins. Different testing methodologies are available to unravel the bugs that were committed during the implementation phase. The purpose of the system testing is to consider all the likely variations to which it will be suggested and push the systems to limits. The testing process focuses on the logical intervals of the software ensuring that all statements have been tested and on the functional interval conducting tests to uncover errors and ensure that defined input will produce actual results that agree with the required results.

5.1. Unit Testing

Unit testing is the first level of testing and is often performed by the developers themselves. It is the process of ensuring individual components of a piece of software at the code level are functional and work as they were designed to. Developers in a test-driven environment will typically write and run the tests prior to the software or feature being passed over to the test team. Unit testing can be conducted manually, but automating the process will speed up delivery cycles and expand test coverage. Unit testing will also make debugging easier because finding issues earlier means they take less time to fix than if they were discovered later in the testing process. Test Left is a tool that allows advanced testers and developers to shift left with the fastest test automation tool embedded in any IDE.

5.2 Integration Testing

After each unit is thoroughly tested, we integrate it with other units to create modules or components that are designed to perform specific tasks or activities. These are then tested as a group through integration testing to ensure whole segments of an application behave as expected (i.e., the interactions between units are seamless). These tests are often framed by user scenarios, such as logging into an application or opening files. Integrated tests can be conducted by either developers or independent testers and usually consist of a combination of automated functional and manual tests.

This movie recommendation system is tested with black-box testing in which the functionalities of the application are tested without interfering with the internal structure of code, implementation details, and internal paths.

By testing this application our main focus is on the input and output of the system and it is entirely based on software requirements and specifications. It is also known as Behavioral Testing.

5.4 System Testing

System testing is a black box testing method, which we use to evaluate the completed and integrated system, as a whole, to ensure it meets specified requirements. The functionality of the software is tested from end-to-end and is typically conducted by a separate testing team from the development team before the product is pushed into production

5.5 Test Cases

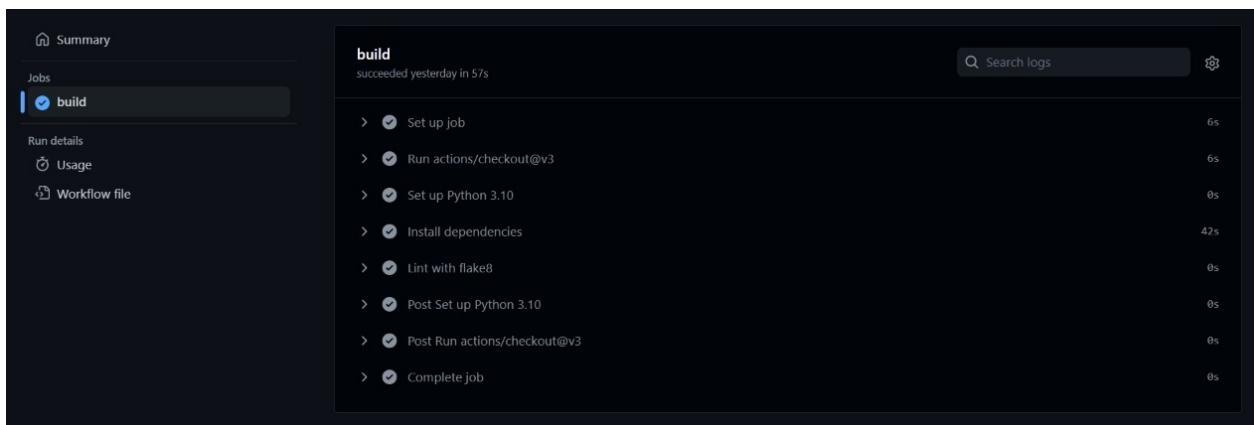
Test cases of the core of our project are available in the CI/CD pipeline of Movie Recommendation System. We implemented CI/CD for our system on the basis of our system requirements.

5.5.1. Continuous integration (CI)

We have added a YAML file that contains the jobs for preventing our code from frequently committing code to a shared repository. Committing code more often detects errors sooner and reduces the amount of code a developer needs to debug when finding the source of an error.

CI yaml file:

<https://github.com/kashishahuja2002/Minor-Project-Next-Up/blob/main/.github/workflows/python-app.yml>



5.5.2 Continuous Deployment(CD)

We delivered this movie recommendation system with features short release cycles, automation, and a direct connection to the source code repository. We designed it to enable teams to get incremental app updates to production quickly and safely. A successful CD process aligns project technology, processes, and automated activity. The Heroku platform is used in this project to maximize developer productivity and provide a great developer experience, from an app's first build to production and beyond. As part of this focus, Heroku extends the platform with a range of tools that support modern development practices, such as continuous delivery.



HEROKU

Personal > next-up-movies

GitHub kashishahuja2002/Microsoft-Intern-Engage main

Overview Resources Deploy Metrics Activity Access Settings

Activity Feed



15kashishahuja@gmail.com: Deployed cd2106e7

Yesterday at 4:43 PM · v19 · [Compare diff](#)



15kashishahuja@gmail.com: Build succeeded

Yesterday at 4:42 PM · [View build log](#)



15kashishahuja@gmail.com: Deployed 6f1ecd19

Dec 5 at 12:48 AM · v18 · [Roll back to here](#) · [Compare diff](#)



15kashishahuja@gmail.com: Build succeeded

Dec 5 at 12:47 AM · [View build log](#)



15kashishahuja@gmail.com: Deployed 2e141536

Dec 4 at 2:07 PM · v17 · [Roll back to here](#) · [Compare diff](#)



15kashishahuja@gmail.com: Build succeeded

Dec 4 at 2:06 PM · [View build log](#)

6. Conclusion

This **Movie Recommendation System** has become more and more important because of information overload. For a content-based recommender system specifically, we attempt to find a new way to improve the accuracy of the representative of the movie.

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

6.1. Findings

A total of 4 types of recommendation algorithms are implemented in the project:

- **Based on chosen genres and cast**

First, the movies which have the cast chosen by the user are filtered out by using a search algorithm. Then out of those, movies that have the genres chosen by the user are filtered out using a search algorithm. The resultant movies are sorted based on the frequency of chosen cast appearing in a movie and secondly based on popularity. The top 3 movies are recommended to the user.

- **Popular movies based on the genre**

First, the movies which have the genre chosen by the user are filtered out by using a search algorithm. Then movies are sorted based on their popularity. The top movies are recommended to the user for that genre.

- **Popular movies based on the year**

First, the movies which are released in the year chosen by the user are filtered out by using a search algorithm. Then movies are sorted based on their popularity. The top movies are recommended to the user for that genre.

- **Similar movie recommendation**

For recommendations of similar movies, a content-based recommendation system. For recommendation, the system takes into account movie titles, genres, starring cast, keywords, overview, and the director. I have implemented Cosine Algorithm after the vectorization of movies. It is achieved by using Annoy (Approximate Nearest Neighbors) mechanism.

6.2. Endpoint Outputs

1. Sign-up and Sign-in functionality.
2. Forgot password (resetting password) functionality.
3. OTP validation. The user receives a mail containing OTP for validation before resetting the password.
4. The User's credentials are stored in the database.
5. Completely responsive front end.
6. A total of 4 types of recommendations:
 - a. Recommended movies based on the user's chosen genres and casts.
 - b. Most popular movies are based on different genres.
 - c. Most popular movies are based on different years.
 - d. Recommended movies similar to the user's selected movie.
7. Movie details and trailers linked for each movie.
8. Watch a movie option.
9. Option to like or dislike a movie.
10. Client-side session tracking.
11. Continuous deployment

The code for the project can be found on GitHub.

Repository link: <https://github.com/kashishahuja2002/Minor-Project-Next-Up>

6.3. Limitations

There are a few limitations of the movie recommendation system:

- Scalability

As the number of users grows, the algorithms suffer scalability issues. If you have 10 million customers and 100,000 movies, you would have to create a sparse matrix with one trillion elements.

- The lack of the right data

Input data may not always be accurate because humans are not perfect at providing ratings. Users' behavior is more important than ratings.

- Cold Start Problem

It needs enough users in the system to find a match. For instance, if we want to find similar users or similar items, we match them with the set of available users or items. At initial stage for a new user, his profile is empty as he has not rated any item and the system do not know about his taste, so it becomes difficult for a system to provide him recommendation about any item. Same case can be with new item, as it is not rated by any user because it's new for the user. Both these problem can be resolved by implementing hybrid techniques

6.4. Scope for Future Prospects

With enough time frame and dedication, the project can be developed further. The bigger picture we had in mind while designing this project includes the below points.

- Like/Dislike

The option to like or dislike a movie adds the movie to the users' like/dislike list. As of now, we are just accumulating the data. This can be further extended by using the like/dislike list to recommend movies to the user.

- Watch Movie

The watch movie option currently displays the same movie intro for all the movies. In the future, it can be customized according to the movie selected.

- Collaborative Filtering

The model currently uses a content-based recommendation system. It can be converted into a hybrid system by adding a collaborative filtering mechanism.

7. Bibliography & References

- Types of Filtering Techniques (IJERT)
https://www.ijert.org/research/recommender-systems-types-of-filtering-techniques-IJERT_V3IS110197.pdf
- Movie Recommendation System Report (New York University)
https://www.ijert.org/research/recommender-systems-types-of-filtering-techniques-IJERT_V3IS110197.pdf
- Movie Recommendation System Research (VTU)
http://203.201.63.46:8080/jspui/bitstream/123456789/6218/1/PR3215%20-%20movie_recommendation_on_system-report%20-%20PAVAN%20KUMAR%20P%20B.pdf
- IEE Movie Recommendation using Machine Learning
<https://ieeexplore.ieee.org/document/9702472>
- Movie Recommendation System Project
https://youtu.be/1xtrIEwY_zY
- Annoy (Approximately nearest neighbor) c++ library with python binding
<https://github.com/spotify/annoy>