



Home

Film

Theater

# CINIMATCH

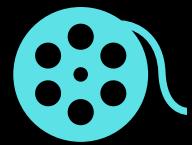
## MOVIE RECOMMENDATION SYSTEM

Project Overview and Implementation

Chaithra N  
21BTRCL026



FILM

[Home](#)[Film](#)[Theater](#)

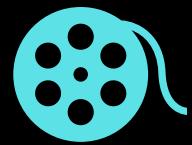
# PROBLEM STATEMENT

**Description:** Build a simple movie recommendation system.

**Functionalities:**

1. Users can submit any number of movies and the system will store them.
2. Users can search for movies and receive recommendations based on genre or ratings.
3. Users can delete a movie from the system.

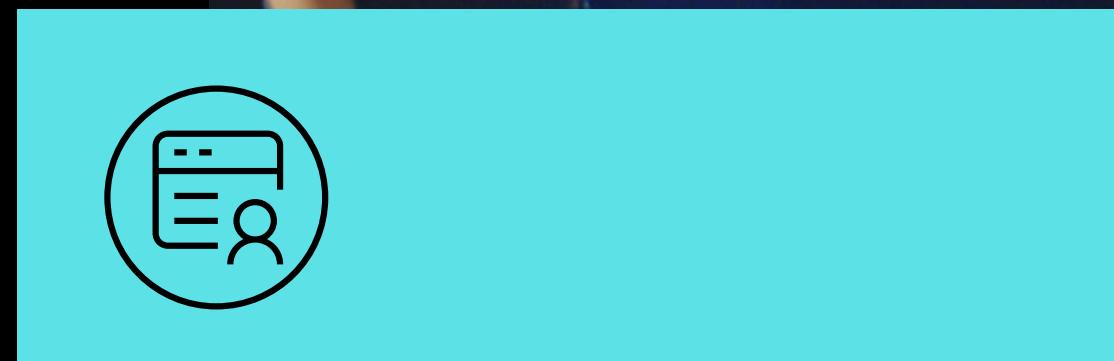




Home

Film

Theater

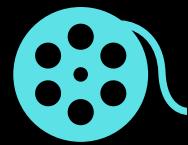


# CONCEPTS USED

Core Concepts:

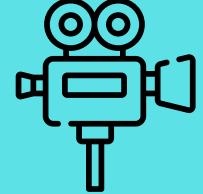
- Sorting algorithms - quicksort
- Searching algorithms - python search (linear)
- Data structures: lists, dictionaries

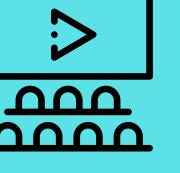
FILM

[Home](#)[Film](#)[Theater](#)

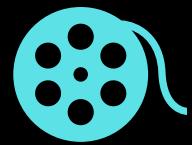
# MOVIE RATING SYSTEM

- Add new movies with details like title, genre, language, actors, etc.
- Calculate unique ratings based on predefined criteria.
- Display the calculated ratings.

 Title, genre, language, actors, box office collection, review, OTT details, budget, shoot days, new films signed, watched status.

 calc\_weight():  
Calculates the weight-based rating for a movie.



[Home](#)[Film](#)[Theater](#)

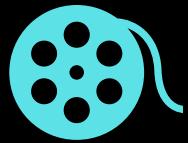
# PROGRAM STRUCTURE

## ■ Classes and Methods:

- Rating class to calculate movie ratings based on various attributes.
- Methods to handle movie attributes like genre, language, actors, box office collection, etc.

## ■ Static Method

run\_all\_methods to apply all rating calculations.



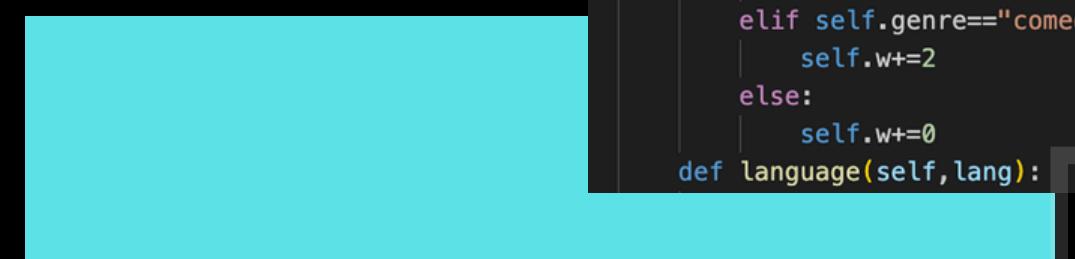
# CODE SNIPPET

## - RATING CLASS

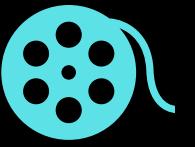
[Home](#)[Film](#)[Theater](#)

```
import re
import inspect
class Rating():
    ratings=[]
    def __init__(self, title, genre, lang, male_actor,
                 female_actor, boc, review, days_ott, ott_platform,
                 budget, shoot_days, next_film,watched):
        self.genre=genre
        self.title=title
        self.lang=lang
        self.male_actor=male_actor
        self.female_actor=female_actor
        self.boc=boc
        self.review=review
        self.days_ott=days_ott
        self.ott_platform=ott_platform
        self.budget=budget
        self.shoot_days=shoot_days
        self.next_film=next_film
        self.watched=watched
        self.w=0

    def gen(self,genre):
        if self.genre=="sci-fi":
            self.w+=2
        elif self.genre=="drama":
            self.w+=2
        elif self.genre=="romance":
            self.w+=1
        elif self.genre=="rom-com":
            self.w+=1
        elif self.genre=="action":
            self.w+=1
        elif self.genre=="tragedy":
            self.w+=1
        elif self.genre=="comedy":
            self.w+=2
        else:
            self.w+=0
    def language(self,lang):
```



FILM

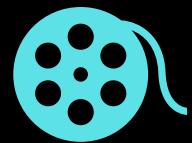
[Home](#)[Film](#)[Theater](#)

```
print("IT ALL STARTS HERE")
print("WELCOME TO CINIMATCH")
def add_movie():
    movies={}
    m=int(input("Enter the number of movies you wish to enter"))
    for i in range(m):
        movie={}
        title=input("\nEnter the movie {} name: ".format(i+1))
        movie['title']=title
        genre=input("Genre: ").lower
        movie['genre']=genre
        lang=input("Language:").lower()
        movie['lang']=lang
        male_actor=input("Male actor's name:").lower()
        movie['male_actor'] = male_actor
        female_actor=input("Female actor's name:").lower()
        movie['female_actor'] = female_actor
        boc=int(input("Enter box office collection in crores:"))
        movie['box_office_collection']= boc
        review=str(input("Enter critics review:")).lower()
        movie['review'] = review
        days_ott=int(input("No.of days to release in ott:"))
        movie['days_to_release_in_ott'] = days_ott
        ott_platform=input("Ott platform:").lower()
        movie['ott_platform'] = ott_platform
        budget=int(input("Enter budget of the movie in crores:"))
        movie['budget'] = budget
        shoot_days=int(input("No.of days to shoot the movie:"))
        movie['shoot_days'] = shoot_days
        next_film=int(input("No.of new films signed:"))
        movie['next_film'] = next_film
        watched=input("Have you watched the movie?").lower()
        movie['watched'] = watched
        movies[i+1] = movie
Rating.run_all_methods(title, genre, lang, male_actor, female_actor,
movies[i + 1] = movie
return movies
```

# CODE SNIPPET

## - MOVIE OPERATIONS

FILM

[Home](#)[Film](#)[Theater](#)

# DEMONSTRATION

## Adding Movies:

- Input details for new movies.

## Searching Movies:

- Search by title or genre.

## Recommending Movies:

- Recommend top N movies based on ratings.

## Deleting Movies:

- Delete movies by title.

```
❸ (venv) (base) chaithra@Chaithras-MacBook-Pro ~ % /Users/chaithra/The-Cookbook/venv/bin/python /Users/chaithra/cinimatch.py
IT ALL STARTS HERE
WELCOME TO CINIMATCH
Menu
1. Add new movies
2. Display movies
3. Search movies
4. Recommend moovies
5. Delete movies
What do want to do today?1
Enter the number of movies you wish to enter1

Enter the movie 1 name: chaithra
Genre: action
Language:kannada
Male actor's name:ganesh
Female actor's name:chaithra
Enter box office collection in crores:500
Enter critics review:amazing
No.of days to release in ott:20
Ott platform:netflix
Enter budget of the movie in crores:9
No.of days to shoot the movie:30
No.of new films signed:5
Have you watched the movie?yes
Rating: 4
Movies added successfully
Menu
1. Add new movies
2. Display movies
3. Search movies
4. Recommend moovies
5. Delete movies
What do want to do today?2

Movie 1:
Title: chaithra
Genre: <built-in method lower of str object at 0x10155d6b0>
Lang: kannada
Male_actor: ganesh
Female_actor: chaithra
Box_office_collection: 500
```



Home

Film

Theater

# CONCLUSION

## Summary:

- Implemented a simple movie recommendation system with core functionalities.
- Utilized sorting, searching algorithms, and data structures effectively.

## Future Enhancements:

- Improve recommendation algorithms.
- Add user authentication and personalized recommendations.





# THANK YOU

---

FILM