A

PROJECT REPORT ON

MOVIE RECOMMENDATION AND SENTIMENT ANALYSIS

By

HARSH PATEL(CE-116)(19CEUON067)
ANSH SHAH(CE-139)(19CEUOS095)

B.Tech CE Semester-VI
Subject: System Design Practice

Guided by:
Prof.Pandav K. Patel
Assistant Professor
Dept. of Comp. Engg.

Faculty of Technology
Department of Computer Engineering
Dharmsinh Desai University

Faculty of Technology
Department of Computer Engineering
Dharmsinh Desai University

CERTIFICATE

This is to certify that the practical / term work carried out in the subject of

System Design Practice and recorded in this journal is the

bonafide work of

HARSH PATEL(CE-116)(19CEUON067)
ANSH SHAH(CE-139)(19CEUOS095)

of B.Tech semester **VI** in the branch of **Computer Engineering**

during the academic year **2021-2022**.

Prof. Pandav K. Patel	Dr. C. K. Bhensdadia,
Assistant Professor,	Head,
Dept. of Computer Engg.,	Dept. of Computer Engg.,
Faculty of Technology	Faculty of Technology
Dharmsinh Desai University, Nadiad	Dharmsinh Desai University,Nadiad

Contents

Abstract

- Movie Recommendation and Review’s Sentiment Analysis is a webapp that Provides Movie Info Like Title, Poster, Cast & Crew, Ratings, Plot, Genre etc. In Addition to it, user can also see Movie Reviews and Sentiment of the review next to it. also, User Can See List of Similar Movies according to Genre, Actor, director which are Recommended to him/her.

- Also Admin Can Add Movies To Dataset, Delete Movies which are in the dataset but are not required and Update Movies when some changes are required in their respective entry.

Introduction:

- Movie Recommendation and Review Sentiment Analysis is a webapp targeting users who are interested in watching movies. It is a Content based movie recommendation webapp. On this webapp, there are two types of Users:
    1.) Admin
    2.) End User

- Admin Takes care of operations like Add Movies, Delete Movies, Update Movies. End User can Search a movie based on a language and Its Information Like Title, Plot, Ratings, Cast, Genre, etc gets displayed. End User can see movie reviews with its sentiment and also Movies Recommended Also End User can click on Individual Cast to Get Details About That Particular Cast (Used TMDB API for this) This All Functionalities are applicable to Admin too as he can be one of the end User.

- Technologies and Tools Used:

    - ➢ Python
    - ➢ Flask
    - ➢ Machine Learning Algorithms
    - ➢ TMDB API
    - ➢ Kaggle Datasets
    - ➢ JavaScript & JQuery

Software Requirement Specification:

R.1: Admin:

Description: This Module Manages functionalities of admin.

- R.1.1: Admin Authentication
  Description: Checks if credentials entered by admin are correct or incorrect.
  Input: Username and Password.
  Output: If the entered username or password are incorrect, incorrect credentials message is shown.
- R.1.2: Admin Login
  Description: If Credentials entered by admin are correct, he is redirected to welcome page
  Input: Username and Password.
  Output: if the entered username and password are correct, admin is redirected to welcome page.
- R.1.3: Admin Logout
  Description: session is terminated on clicking it and admin is redirected to homepage.
  Input: Logout Button Click
  Output: Redirect to Homepage and Session is terminated (Backend)
- R.1.4: Navigation Bar
  Description: Redirection to a particular page.
  Input: Link clicking by admin.
  Output: Redirect to that particular page.

R.2: Manage Movies:

Description: This Module manages Functionalities that are related with movie and dataset which is done by the admin.

- R.2.1: Add Movie
  Description: Adds a movie to dataset if it is not present in dataset else displays required validation message.
  Input: Movie name, Director, Actor1, Actor2, Actor3, Genre, Language. (Used to get Poster of movie By TMDB API) Passed as String Arguments.
  Output: Required Validation Message is shown (Success Message on success of adding movie else Movie already exist message is displayed).
- R.2.2: Remove Movie
  Description: Removes a movie from dataset if it is present in dataset else displays required validation message.
  Input: Movie name, Director name, Language. (Used to get Poster of movie By TMDB API) Passed as String Arguments.
  Output: Required Validation Message is shown (Success Message on success of deleting movie else Movie does not exist message is displayed).

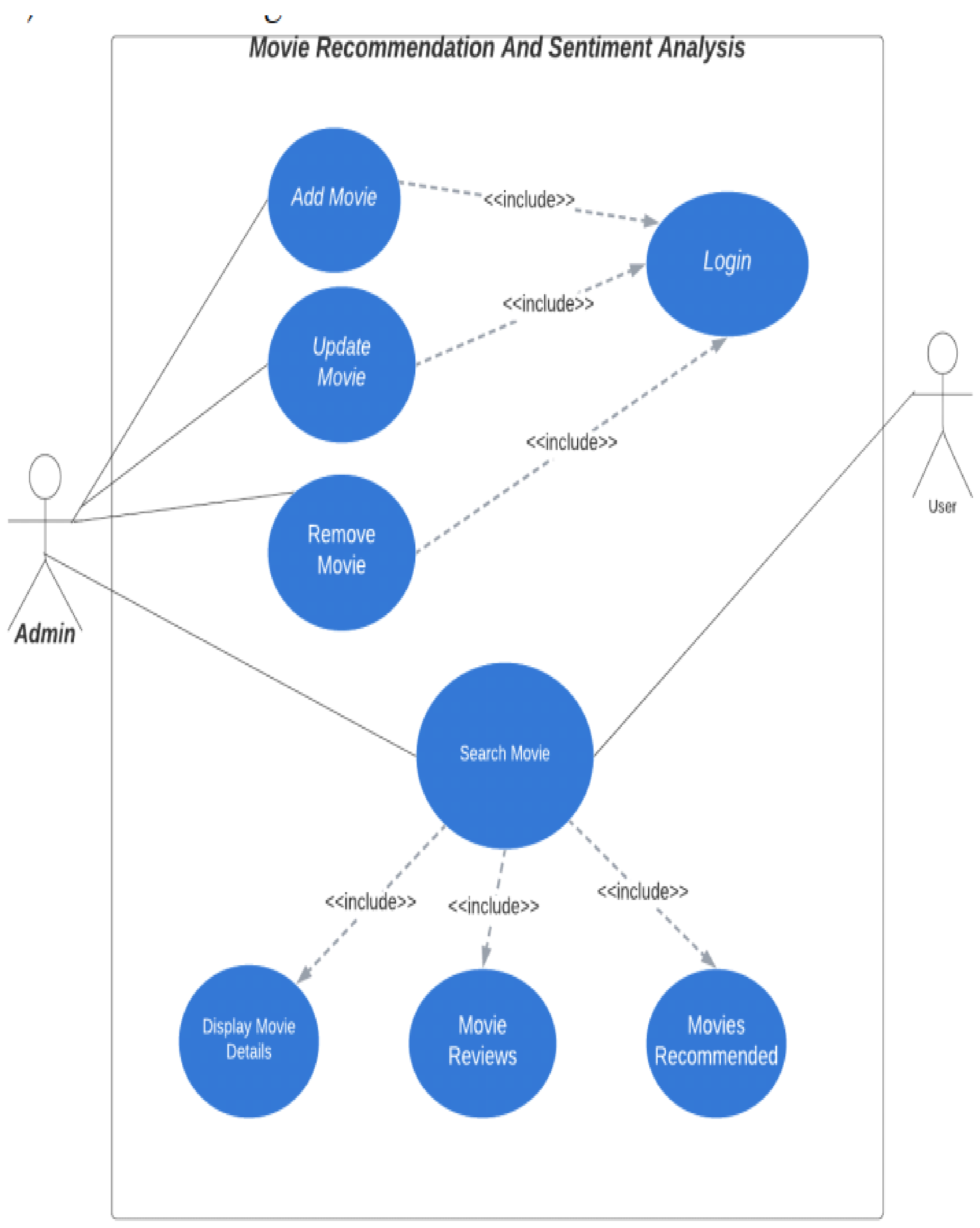
- R.2.3: Update Movie
  Description: Updates a movie to dataset if it is present in dataset else displays required validation message.
  Input: Movie name, Director name Language. (Used to get Poster of movie By TMDB API) Passed as String Arguments.
  Output: Required Validation Message is shown (Success Message on success of updating movie else Movie does not exist message is displayed).

R.3: Function of movies

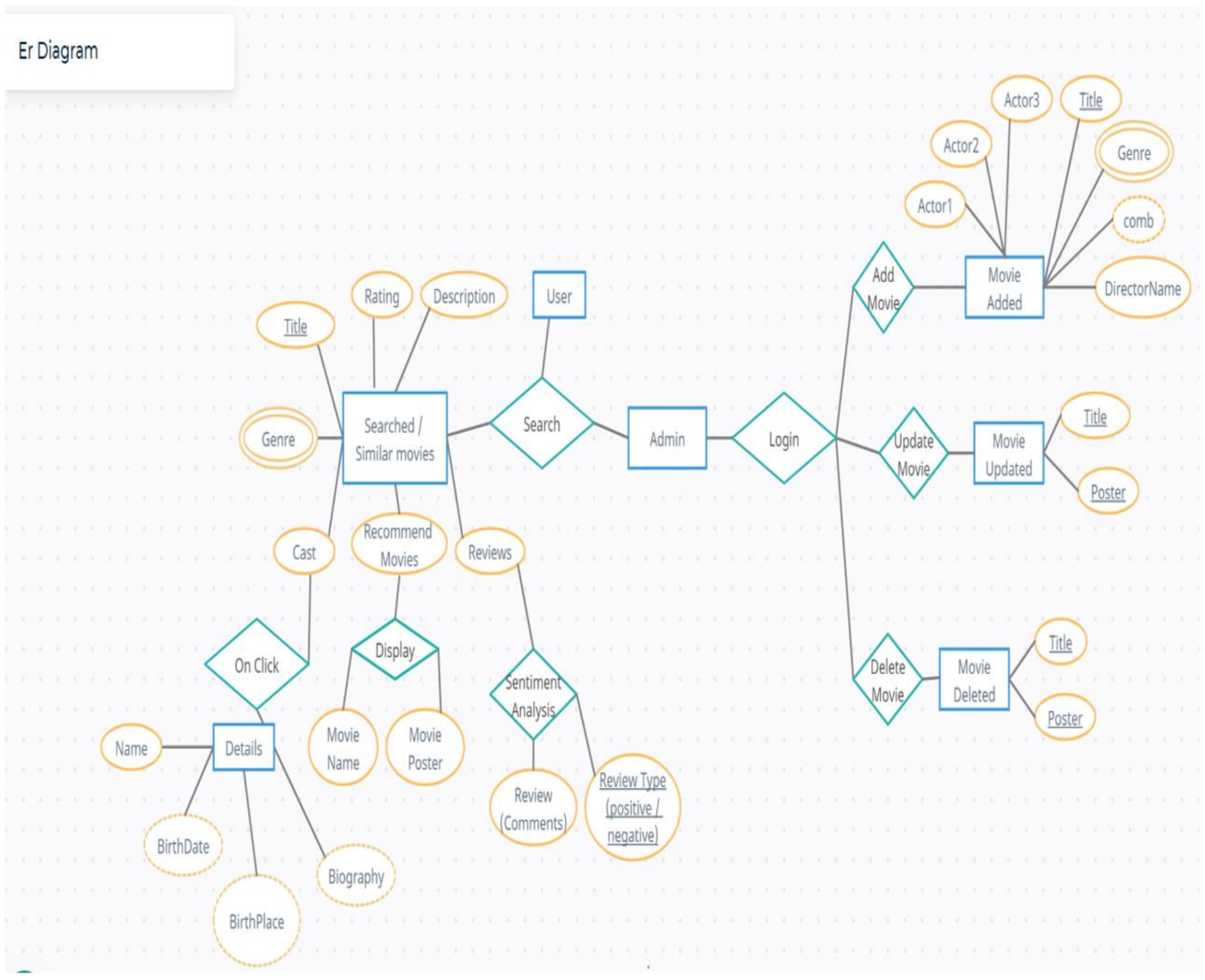
- R.3.1: Search Movie
  Description: Lets User search a movie.
  Input: Movie name, Language. (Used to get Poster of movie By TMDB API and Separator for movie with same title and different language) entered by User.
  Output: Movie Info, Cast & Crew, Reviews with Sentiment, Recommendations are shown if movie exist in dataset else Movie does not exist message is shown.
- R.3.2: Recommend Movie
  Description: Returns a list of Movies present in dataset which are similar to given movie title.
  Input: Movie name Passed as String Arguments.
  Output: List of Movie names similar to passed movie title if it exists in dataset else movie does not exist message is shown.
- R.3.3: Review Sentiment Analysis
  Description: Predicts whether a piece of string passed as input is more positive or negative.
  Input: Movie Review String (Fetched Through Web Scraping) Passed as String Arguments.
  Output: If string is empty, string cannot be empty message is displayed else output is between 0 and 1(if >=0.5 it is rounded off to 1 else 0 is kept).
- R.3.4: Web Scraping Reviews
  Description: Fetching Review from IMDB Site Using web Scraping.
  Input: IMDB-ID of Movie Passed as String Arguments.
  Output: Reviews of Movie inserted in a list.

Design:

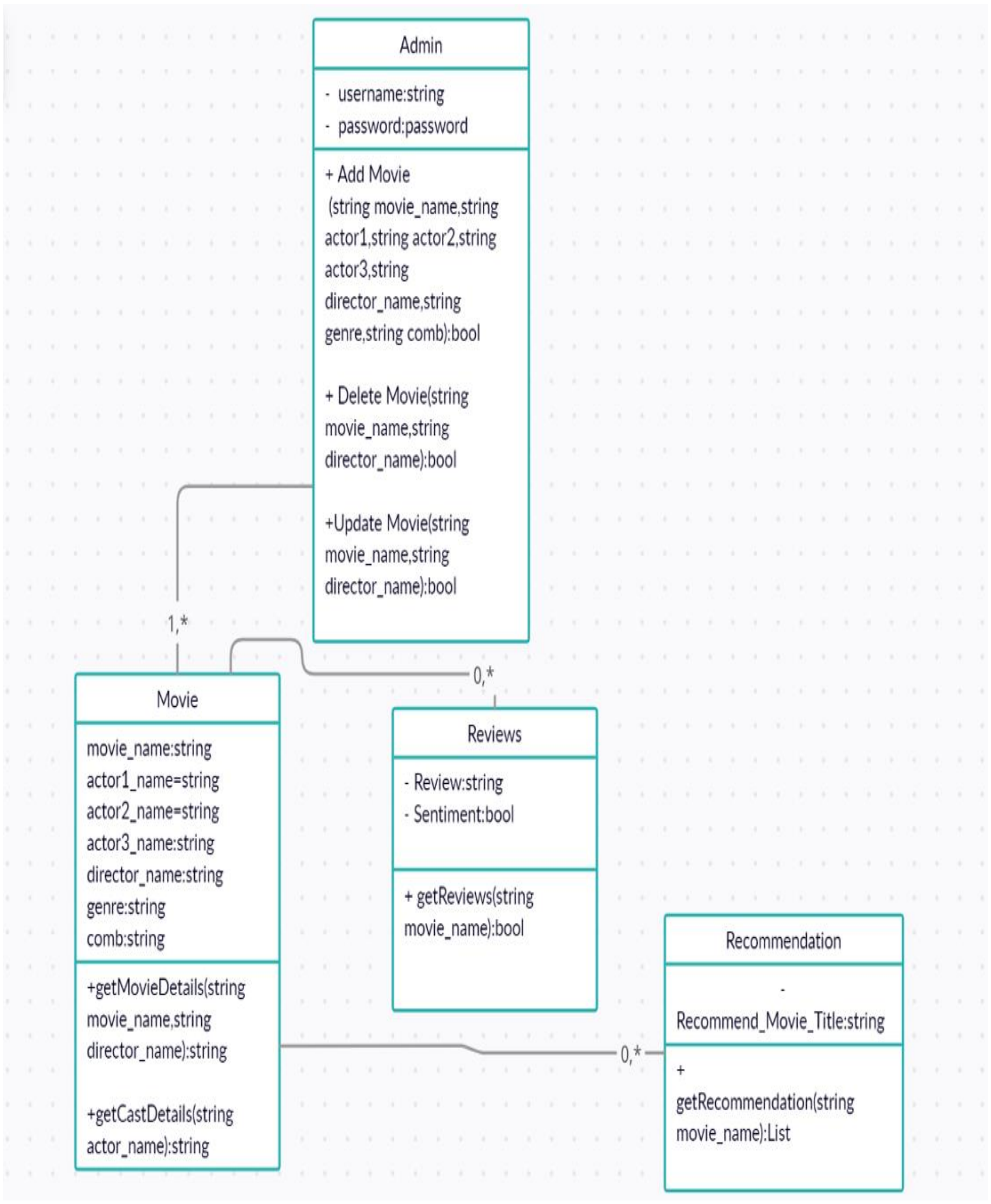
Use Case diagram:



ER Diagram:


Er Diagram
Actor3
Title
Actor2
Genre
Actor1
comb
Add Movie
Movie Added
DirectorName
Rating
Description
User
Title
Searched / Similar movies
Search
Admin
Login
Update Movie
Movie Updated
Title
Genre
Poster
Cast
Recommend Movies
Reviews
On Click
Display
Sentiment Analysis
Delete Movie
Movie Deleted
Title
Poster
Name
Details
Movie Name
Movie Poster
Review (Comments)
Review Type (positive / negative)
BirthDate
Biography
BirthPlace

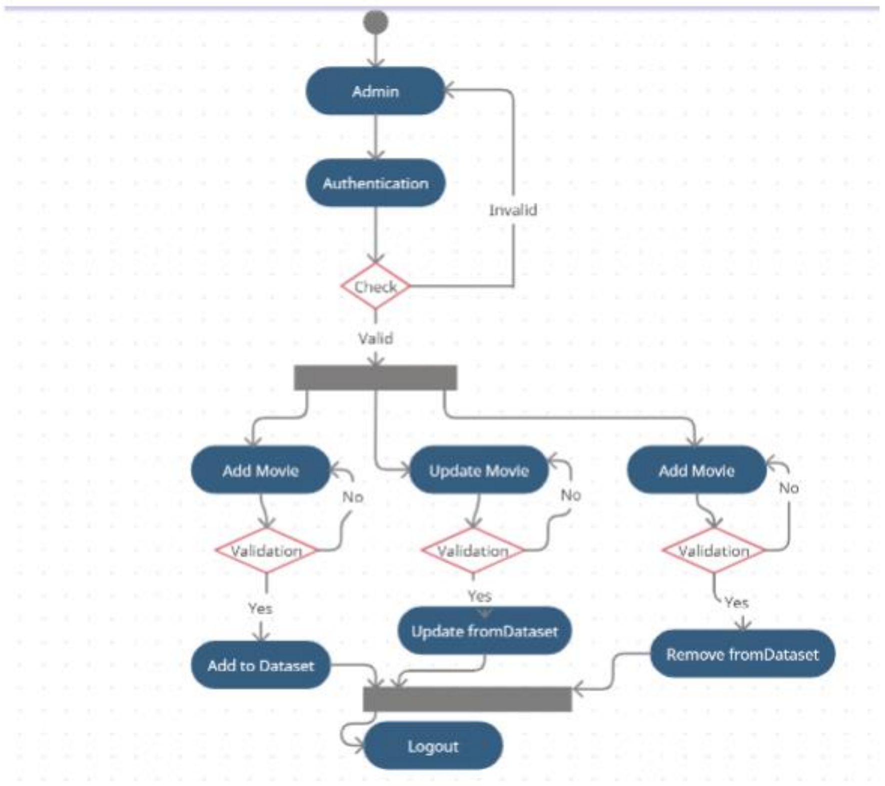
Class Diagram:



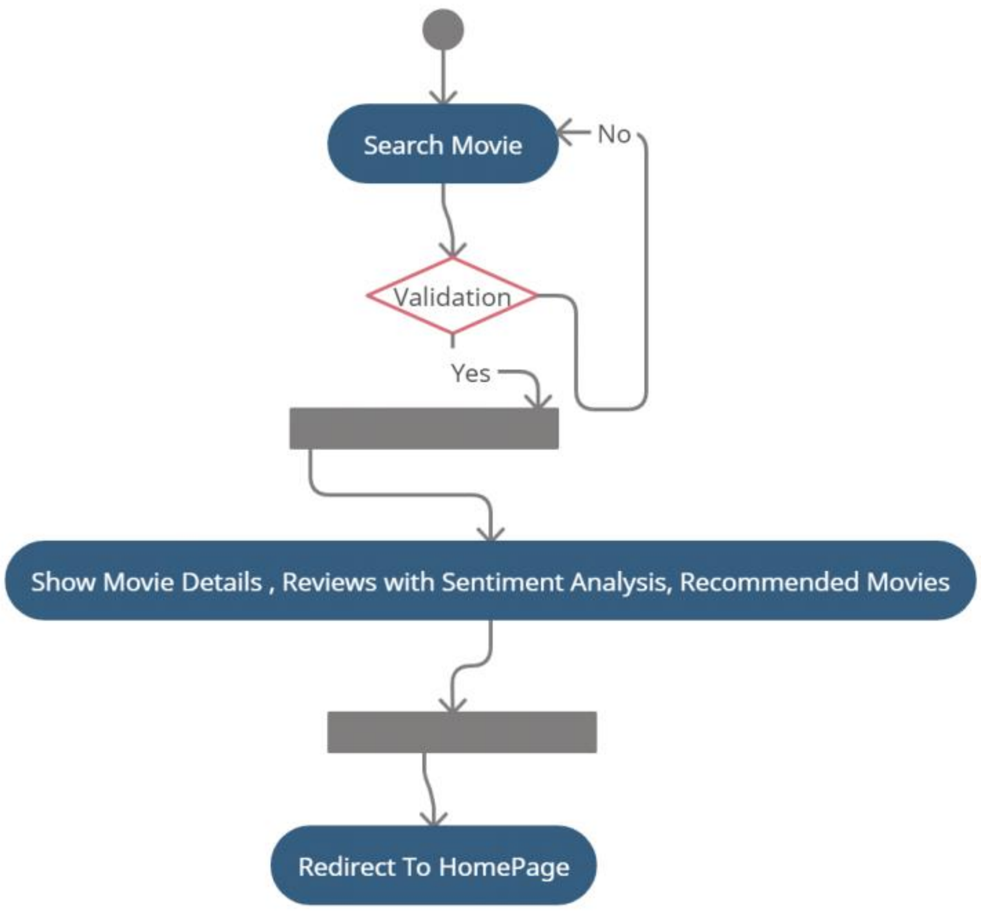
Admin
- username:string
- password:password
+ Add Movie (string movie_name,string actor1,string actor2,string actor3,string director_name,string genre,string comb):bool
+ Delete Movie(string movie_name,string director_name):bool
+Update Movie(string movie_name,string director_name):bool
1,*
0,*
Movie
movie_name:string
actor1_name=string
actor2_name=string
actor3_name:string
director_name:string
genre:string
comb:string
+getMovieDetails(string movie_name,string director_name):string
+getCastDetails(string actor_name):string
Reviews
- Review:string
- Sentiment:bool
+ getReviews(string movie_name):bool
0,*
Recommendation
- Recommend_Movie_Title:string
+ getRecommendation(string movie_name):List

Activity Diagram (Admin + User):

Admin:



User:


Search Movie
No
Validation
Yes
Show Movie Details , Reviews with Sentiment Analysis, Recommended Movies
Redirect To HomePage

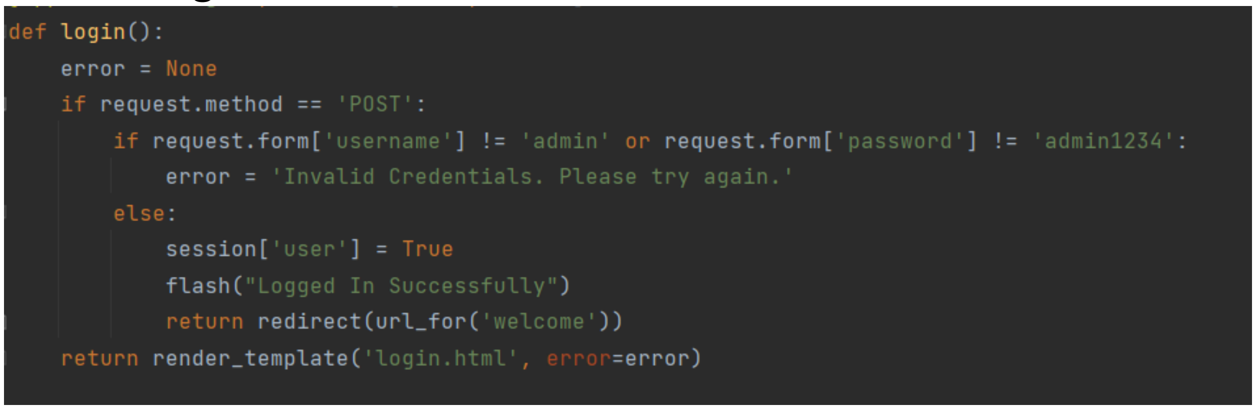
Implementation-Details:

1.) Modules:

- Admin Module:
  - Admin has to be logged in to add Movies, Delete Movies, Update Movies. In Other Words, Admin's Session must be set.
  - This Module has a major functionality of login for admin. If Admin Username and password entered are correct, admin is Logged in to the WebApp else Validation error is thrown.
- Manage-Movies Module:
  - This Module has 3 major functionalities i.e. Add, Delete and Update Movie. Here, Admin if logged in is able to do the Add, Delete, Update operations.
- Function-of-Movies Module:
  - This Module is User-End Module. It has a functionality of searching movie given Movie Title and language.
  - It also has a functionality of predicting if a given string is more on positive side or negative side.
  - It also has a functionality of Recommending similar movies if a movie title is given as a parameter.
  - It also has a functionality of getting reviews from IMDB-ID of movie got using TMDB API and web scraping reviews by imdb-ID

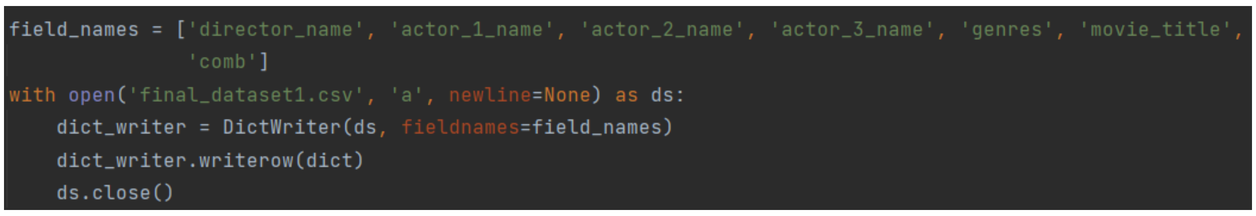
2.) Important Function Prototypes & Snippets of Code Of them:

- Admin-Login function:



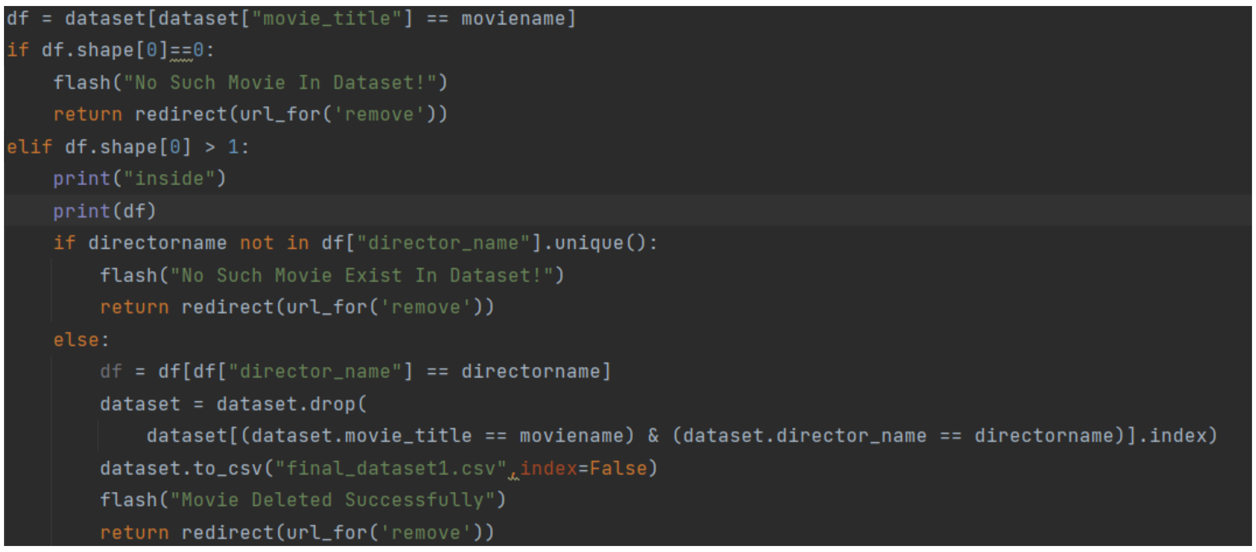
```
def login():
    error = None
    if request.method == 'POST':
        if request.form['username'] != 'admin' or request.form['password'] != 'admin1234':
            error = 'Invalid Credentials. Please try again.'
        else:
            session['user'] = True
            flash("Logged In Successfully")
            return redirect(url_for('welcome'))
    return render_template('login.html', error=error)
```

- Add-Movie Function:



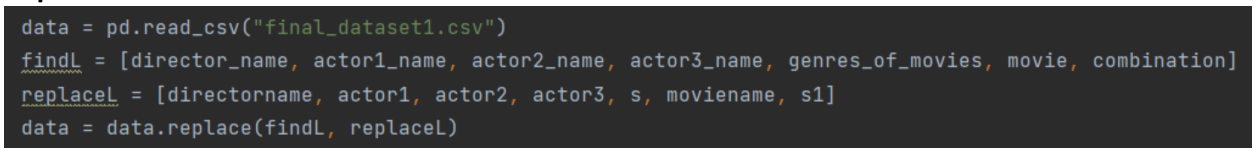
```
field_names = ['director_name', 'actor_1_name', 'actor_2_name', 'actor_3_name', 'genres', 'movie_title',
               'comb']
with open('final_dataset1.csv', 'a', newline=None) as ds:
    dict_writer = DictWriter(ds, fieldnames=field_names)
    dict_writer.writerow(dict)
    ds.close()
```

- Delete-Movie Function:



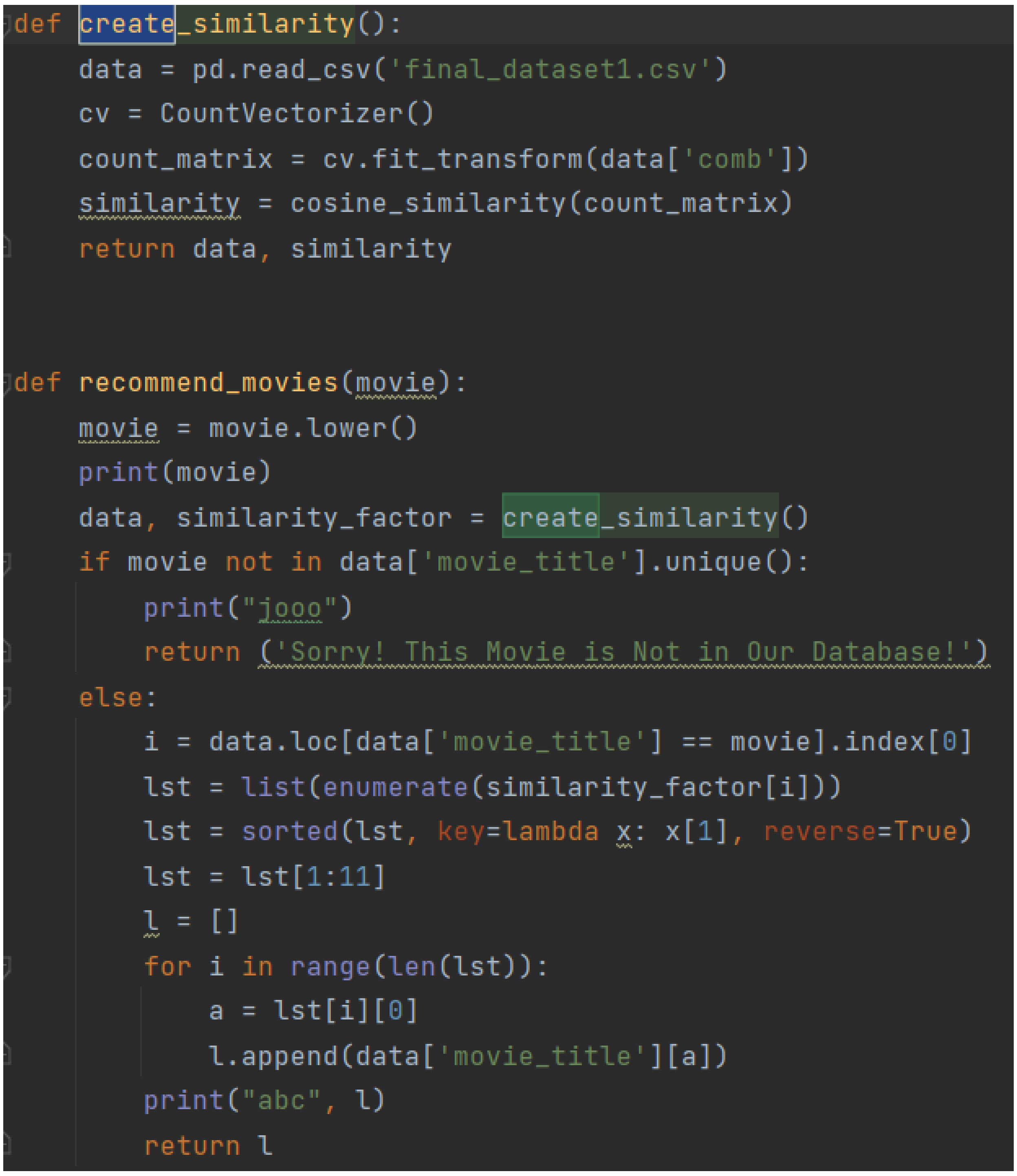
```
df = dataset[dataset["movie_title"] == moviename]
if df.shape[0]==0:
    flash("No Such Movie In Dataset!")
    return redirect(url_for('remove'))
elif df.shape[0] > 1:
    print("inside")
    print(df)
    if directorname not in df["director_name"].unique():
        flash("No Such Movie Exist In Dataset!")
        return redirect(url_for('remove'))
    else:
        df = df[df["director_name"] == directorname]
        dataset = dataset.drop(
            dataset[(dataset.movie_title == moviename) & (dataset.director_name == directorname)].index)
        dataset.to_csv("final_dataset1.csv",index=False)
        flash("Movie Deleted Successfully")
        return redirect(url_for('remove'))
```

- Update-Movie Function:



```
data = pd.read_csv("final_dataset1.csv")
findL = [director_name, actor1_name, actor2_name, actor3_name, genres_of_movies, movie, combination]
replaceL = [directorname, actor1, actor2, actor3, s, moviename, s1]
data = data.replace(findL, replaceL)
```

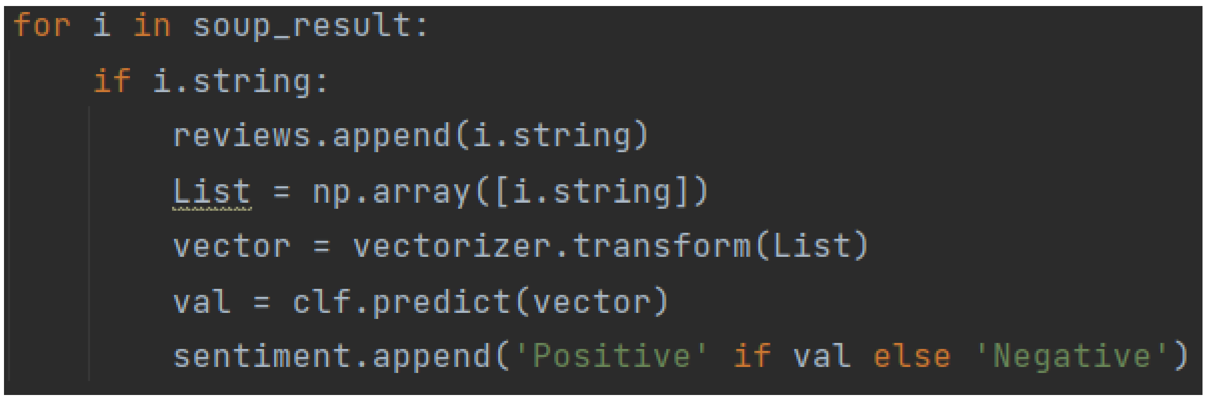
- Movie-Recommendation:



```
def create_similarity():
    data = pd.read_csv('final_dataset1.csv')
    cv = CountVectorizer()
    count_matrix = cv.fit_transform(data['comb'])
    similarity = cosine_similarity(count_matrix)
    return data, similarity


def recommend_movies(movie):
    movie = movie.lower()
    print(movie)
    data, similarity_factor = create_similarity()
    if movie not in data['movie_title'].unique():
        print("jooo")
        return ('Sorry! This Movie is Not in Our Database!')
    else:
        i = data.loc[data['movie_title'] == movie].index[0]
        lst = list(enumerate(similarity_factor[i]))
        lst = sorted(lst, key=lambda x: x[1], reverse=True)
        lst = lst[1:11]
        l = []
        for i in range(len(lst)):
            a = lst[i][0]
            l.append(data['movie_title'][a])
        print("abc", l)
        return l
```

- Review-Sentiment Analysis:



```
for i in soup_result:
    if i.string:
        reviews.append(i.string)
        List = np.array([i.string])
        vector = vectorizer.transform(List)
        val = clf.predict(vector)
        sentiment.append('Positive' if val else 'Negative')
```

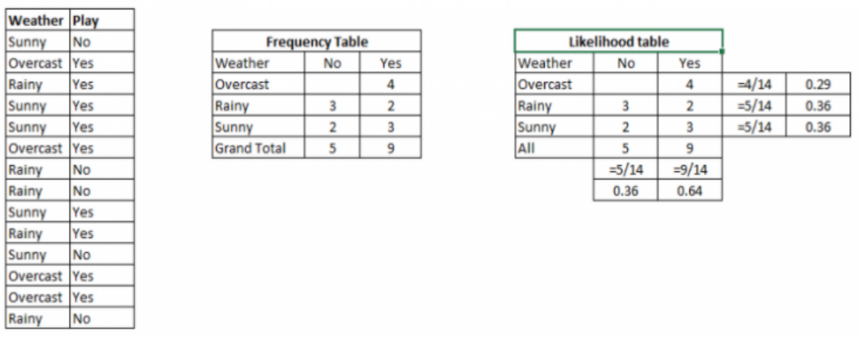
3.) Explain Algorithms Used If Any with Example:

I) Naïve Bayes Algorithm:

- The formula for Bayes' theorem is given as:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

- **P(A|B) is Posterior probability**: Probability of hypothesis A on the observed event B.
- **P(B|A) is Likelihood probability**: Probability of the evidence given that the probability of a hypothesis is true.
- **P(A) is Prior Probability**: Probability of hypothesis before observing the evidence.
- **P(B) is Marginal Probability**: Probability of Evidence.
- Let us Take an Example to understand it:



Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

- **Here, we are given Dataset, Its frequency table with respective Probabilities.**

- **Problem:** Players will play if weather is sunny. Is this statement is correct?
- P(Yes | Sunny) = P(Sunny | Yes) * P(Yes) / P (Sunny).

- Here we have

  P (Sunny |Yes) = 3/9 = 0.33,

  P(Sunny) = 5/14 = 0.36,

  P(Yes)= 9/14 = 0.64.

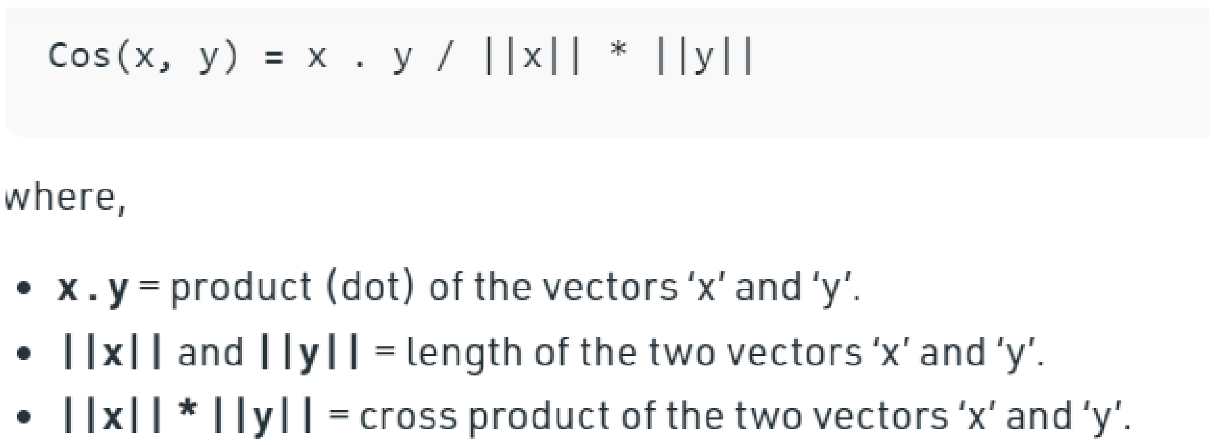
  Now,

  P (Yes | Sunny) = 0.33 * 0.64 / 0.36 = 0.60, which has higher probability. It is Greater than 0.5 so is rounded to 1, so we can say given statement is true.

II)<u>Movie Recommendation Using Cosine Similarity</u>:

- **Cosine similarity** is a metric, helpful in determining, how similar the data objects are irrespective of their size.

- The formula to find the cosine similarity between two vectors is:



```
Cos(x, y) = x . y / ||x|| * ||y||
```

where,

- **x . y** = product (dot) of the vectors 'x' and 'y'.
- **||x||** and **||y||** = length of the two vectors 'x' and 'y'.
- **||x|| * ||y||** = cross product of the two vectors 'x' and 'y'.

- Let Us take an example:

- The 'x' vector has values, **x = { 3, 2, 0, 5 }**

- The 'y' vector has values, **y = { 1, 0, 0, 0 }**

- The formula for calculating the cosine similarity is **Cos(x, y) = x . y / ||x|| * ||y||**

- x . y = 3*1 + 2*0 + 0*0 + 5*0 = 3

- ||x|| = √ (3)^2 + (2)^2 + (0)^2 + (5)^2 =   6.16

- ||y|| = √ (1)^2 + (0)^2 + (0)^2 + (0)^2 = 1

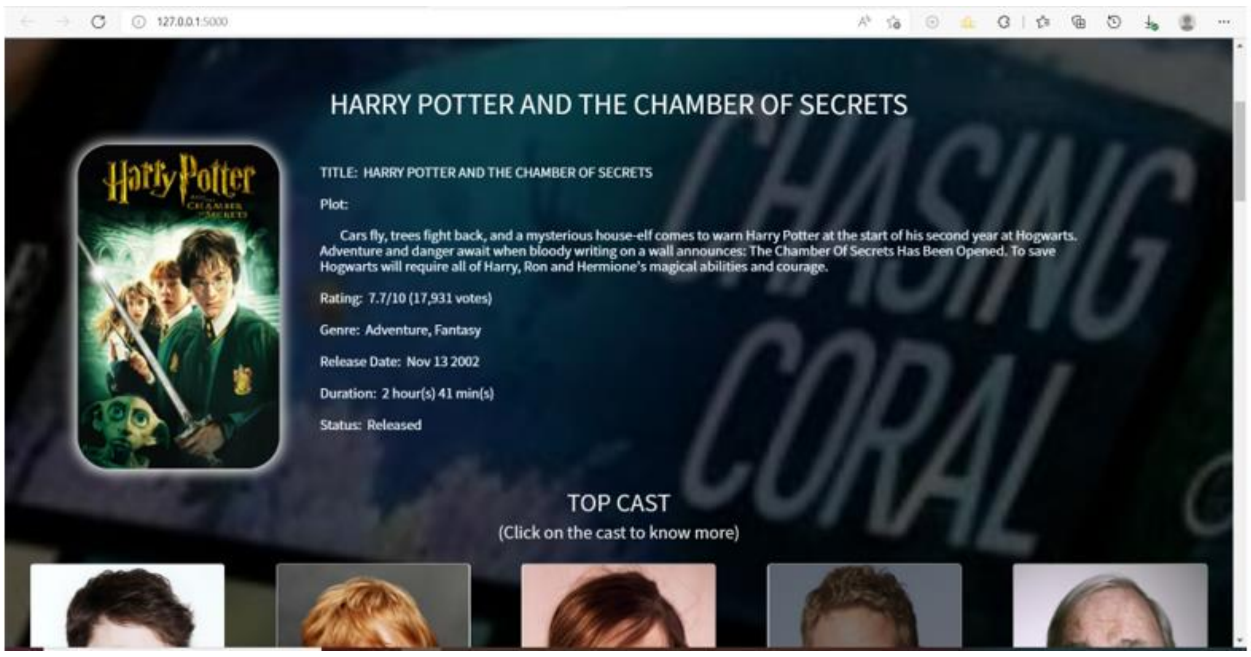
- ∴ Cos(x, y) = 3 / (6.16 * 1) = 0.49

- But Our Dataset Entries are in string form, so to convert it in string form we use scikit learn Library fit and transform method.

Testing:

Module	Field-Names	Expected output	Actual output
Admin-Login	Username & Password (Incorrect)	Incorrect credential Message	Incorrect credential Message
Admin-Login	Username & Password (correct)	Login Successful Message	Login Successful Message
Admin-CRUD	Add-Movie (Already Existing Movie)	Duplicate Entry Message	Duplicate Entry Message
Admin-CRUD	Add-Movie (New Movie)	Movie Added Successfully message	Movie Added Successfully message
Admin-CRUD	Remove-Movie (Not existing in Dataset)	Movie Not in Dataset Message	Movie Not in Dataset Message
Admin-CRUD	Remove-Movie (existing in Dataset)	Movie Deleted Message	Movie Deleted Message
Admin-CRUD	Update-Movie (Not Existing in Dataset)	Movie Not in Dataset Message	Movie Not in Dataset Message
Admin-CRUD	Update-Movie (Existing in Dataset)	Movie Updated Message	Movie Updated Message
Sentiment Analysis	Giving empty string as input	No Sentiment	No sentiment
Recommendation	Empty String as Movie Title/Movie title not in dataset	No Movies Recommended/validation error	No Movies Recommended/ validation error
Sentiment Analysis	Giving Proper String as input	Good or Bad	Good or Bad
Recommendation	Movie Title present in Dataset	List of Movies similar to given movie also present in dataset	List of Movies similar to given movie also present in dataset

Screenshots:



127.0.0.1:5000

HARRY POTTER AND THE CHAMBER OF SECRETS

TITLE: HARRY POTTER AND THE CHAMBER OF SECRETS

Plot:

Cars fly, trees fight back, and a mysterious house-elf comes to warn Harry Potter at the start of his second year at Hogwarts. Adventure and danger await when bloody writing on a wall announces: The Chamber Of Secrets Has Been Opened. To save Hogwarts will require all of Harry, Ron and Hermione's magical abilities and courage.

Rating: 7.7/10 (17,931 votes)

Genre: Adventure, Fantasy

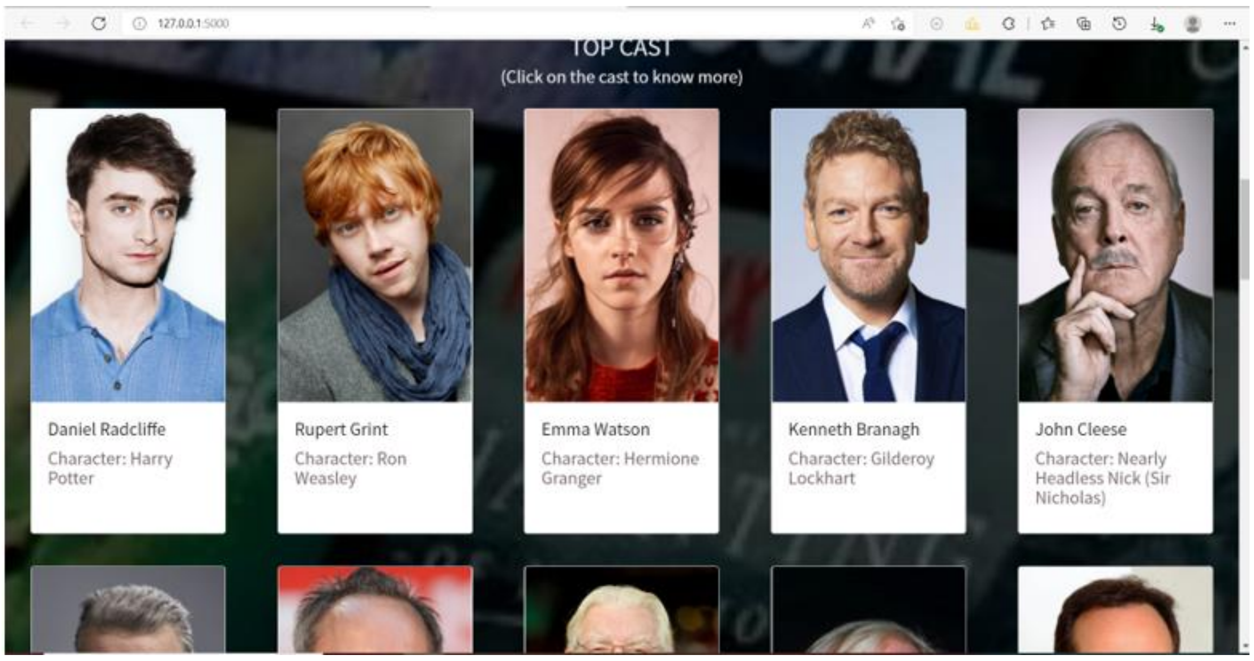
Release Date: Nov 13 2002

Duration: 2 hour(s) 41 min(s)

Status: Released

TOP CAST

(Click on the cast to know more)



127.0.0.1:5000

TOP CAST

(Click on the cast to know more)

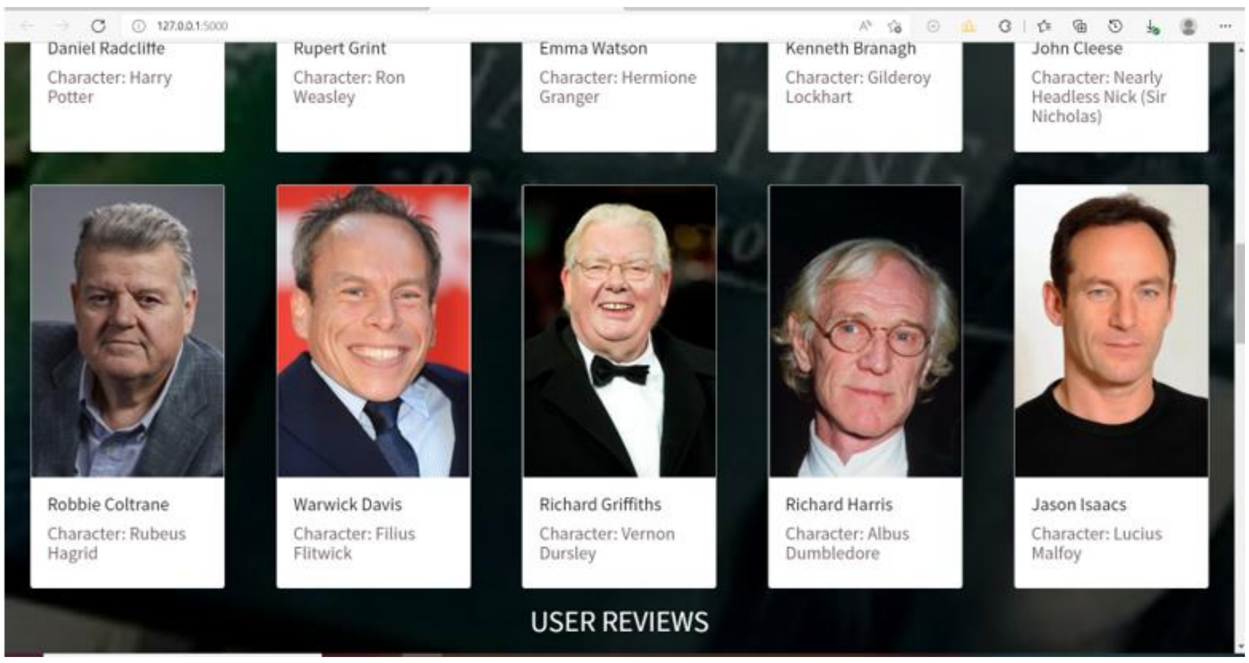
Daniel Radcliffe
Character: Harry Potter

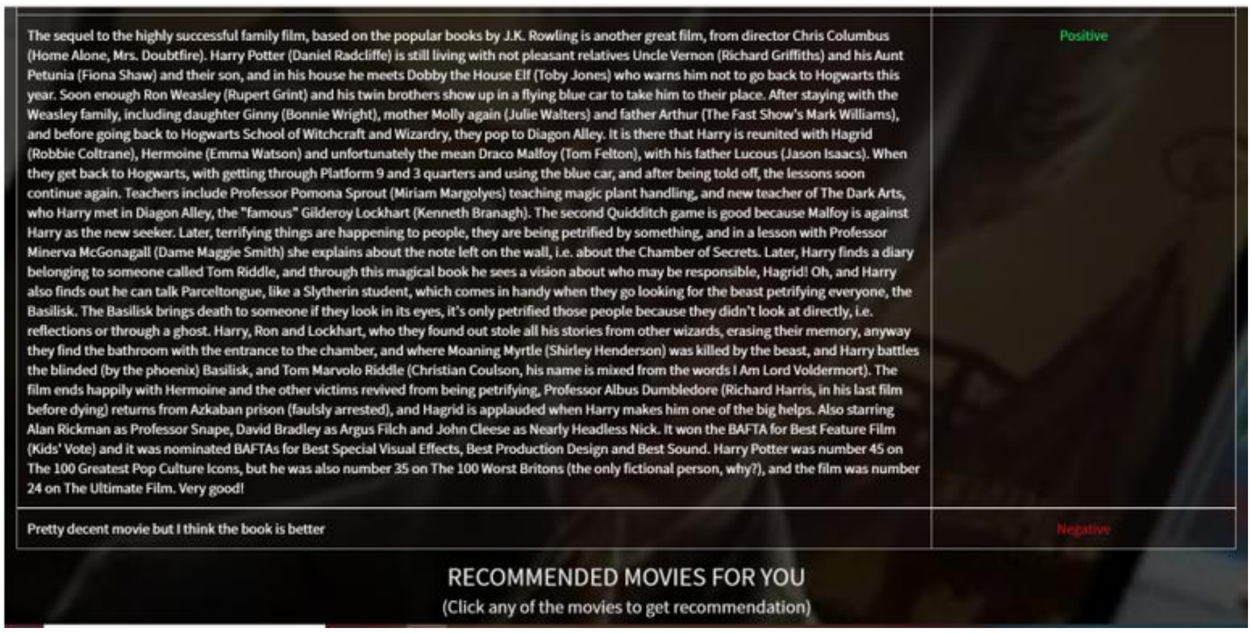
Rupert Grint
Character: Ron Weasley

Emma Watson
Character: Hermione Granger

Kenneth Branagh
Character: Gilderoy Lockhart

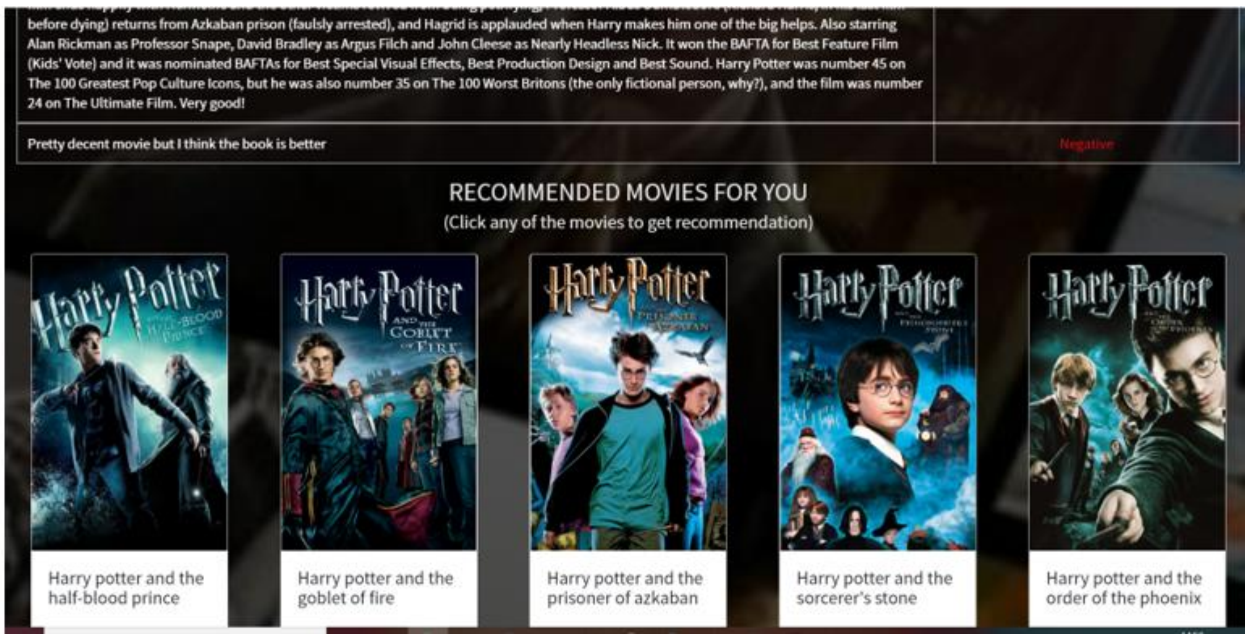
John Cleese
Character: Nearly Headless Nick (Sir Nicholas)





The sequel to the highly successful family film, based on the popular books by J.K. Rowling is another great film, from director Chris Columbus (Home Alone, Mrs. Doubtfire). Harry Potter (Daniel Radcliffe) is still living with not pleasant relatives Uncle Vernon (Richard Griffiths) and his Aunt Petunia (Fiona Shaw) and their son, and in his house he meets Dobby the House Elf (Toby Jones) who warns him not to go back to Hogwarts this year. Soon enough Ron Weasley (Rupert Grint) and his twin brothers show up in a flying blue car to take him to their place. After staying with the Weasley family, including daughter Ginny (Bonnie Wright), mother Molly again (Julie Walters) and father Arthur (The Fast Show's Mark Williams), and before going back to Hogwarts School of Witchcraft and Wizardry, they pop to Diagon Alley. It is there that Harry is reunited with Hagrid (Robbie Coltrane), Hermoine (Emma Watson) and unfortunately the mean Draco Malfoy (Tom Felton), with his father Lucous (Jason Isaacs). When they get back to Hogwarts, with getting through Platform 9 and 3 quarters and using the blue car, and after being told off, the lessons soon continue again. Teachers include Professor Pomona Sprout (Miriam Margolyes) teaching magic plant handling, and new teacher of The Dark Arts, who Harry met in Diagon Alley, the "famous" Gilderoy Lockhart (Kenneth Branagh). The second Quidditch game is good because Malfoy is against Harry as the new seeker. Later, terrifying things are happening to people, they are being petrified by something, and in a lesson with Professor Minerva McGonagall (Dame Maggie Smith) she explains about the note left on the wall, i.e. about the Chamber of Secrets. Later, Harry finds a diary belonging to someone called Tom Riddle, and through this magical book he sees a vision about who may be responsible, Hagrid! Oh, and Harry also finds out he can talk Parceltongue, like a Slytherin student, which comes in handy when they go looking for the beast petrifying everyone, the Basilisk. The Basilisk brings death to someone if they look in its eyes, it's only petrified those people because they didn't look at directly, i.e. reflections or through a ghost. Harry, Ron and Lockhart, who they found out stole all his stories from other wizards, erasing their memory, anyway they find the bathroom with the entrance to the chamber, and where Moaning Myrtle (Shirley Henderson) was killed by the beast, and Harry battles the blinded (by the phoenix) Basilisk, and Tom Marvolo Riddle (Christian Coulson, his name is mixed from the words I Am Lord Voldermort). The film ends happily with Hermoine and the other victims revived from being petrifying, Professor Albus Dumbledore (Richard Harris, in his last film before dying) returns from Azkaban prison (faulsly arrested), and Hagrid is applauded when Harry makes him one of the big helps. Also starring Alan Rickman as Professor Snape, David Bradley as Argus Filch and John Cleese as Nearly Headless Nick. It won the BAFTA for Best Feature Film (Kids' Vote) and it was nominated BAFTAs for Best Special Visual Effects, Best Production Design and Best Sound. Harry Potter was number 45 on The 100 Greatest Pop Culture Icons, but he was also number 35 on The 100 Worst Britons (the only fictional person, why?), and the film was number 24 on The Ultimate Film. Very good!	Positive
Pretty decent movie but I think the book is better	Negative

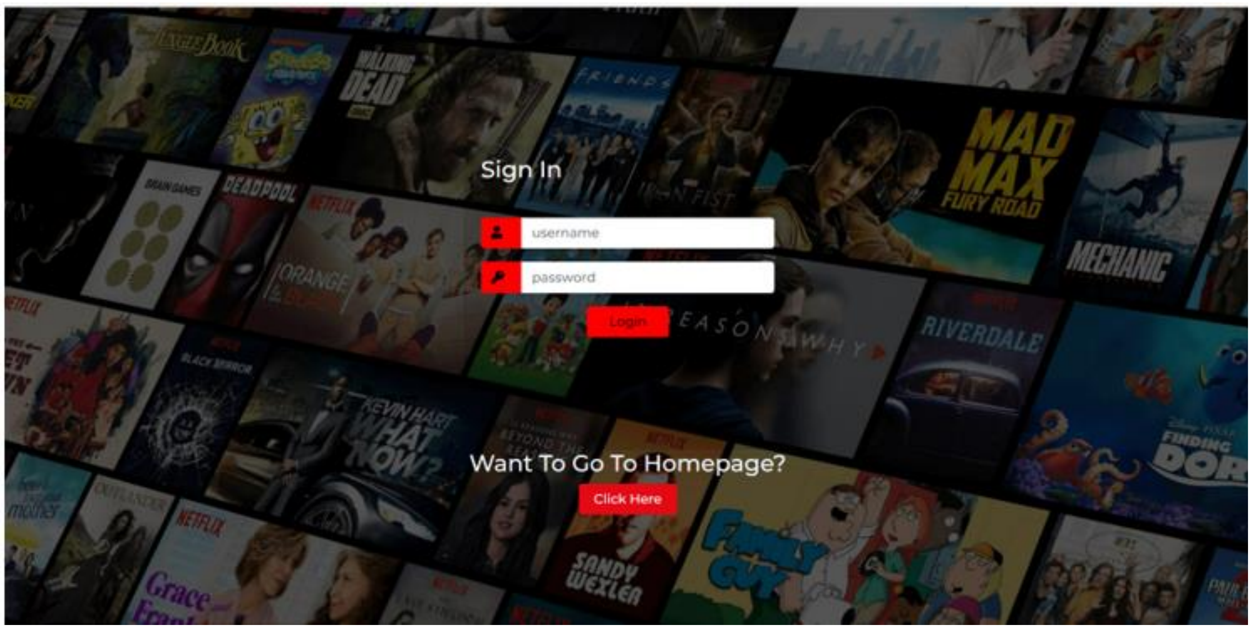
RECOMMENDED MOVIES FOR YOU
(Click any of the movies to get recommendation)

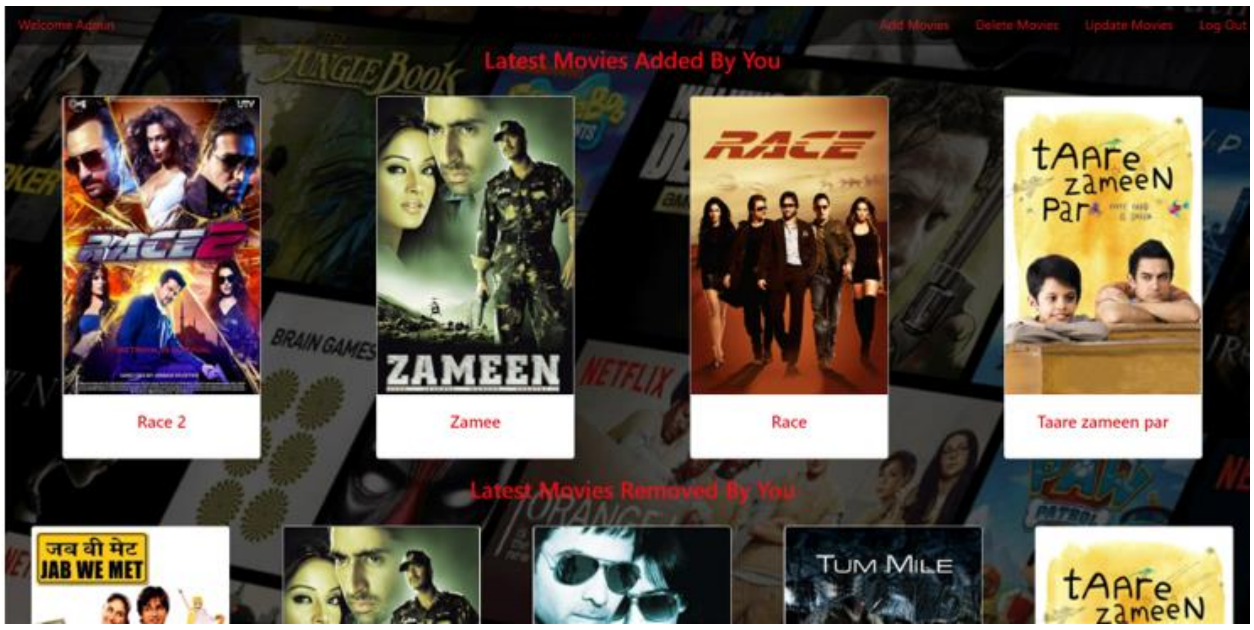

before dying) returns from Azkaban prison (faulsly arrested), and Hagrid is applauded when Harry makes him one of the big helps. Also starring Alan Rickman as Professor Snape, David Bradley as Argus Filch and John Cleese as Nearly Headless Nick. It won the BAFTA for Best Feature Film (Kids' Vote) and it was nominated BAFTAs for Best Special Visual Effects, Best Production Design and Best Sound. Harry Potter was number 45 on The 100 Greatest Pop Culture Icons, but he was also number 35 on The 100 Worst Britons (the only fictional person, why?), and the film was number 24 on The Ultimate Film. Very good!

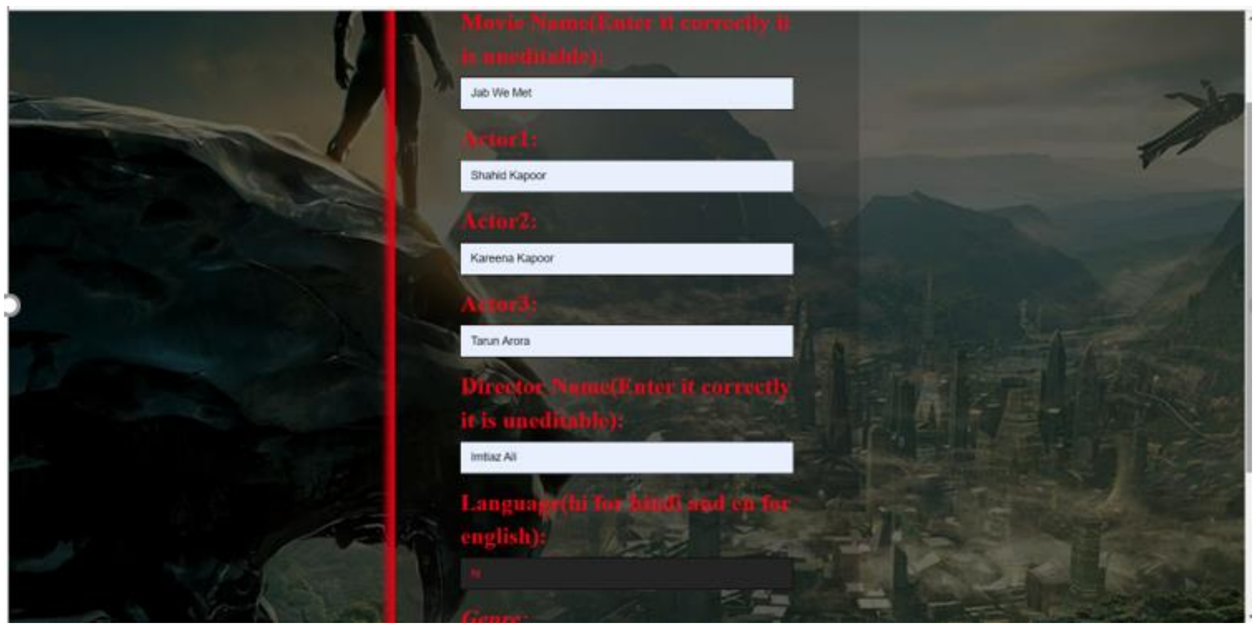
Pretty decent movie but I think the book is better — Negative

RECOMMENDED MOVIES FOR YOU
(Click any of the movies to get recommendation)

Harry potter and the half-blood prince
Harry potter and the goblet of fire
Harry potter and the prisoner of azkaban
Harry potter and the sorcerer's stone
Harry potter and the order of the phoenix


Sign In
username
password
Login
Want To Go To Homepage?
Click Here


Welcome Admin
Add Movies
Delete Movies
Update Movies
Log Out
Latest Movies Added By You
Race 2
Zamee
Race
Taare zameen par
Latest Movies Removed By You



Movie Name(Enter it correctly it is uneditable):

Jab We Met

Actor1:

Shahid Kapoor

Actor2:

Kareena Kapoor

Actor3:

Tarun Arora

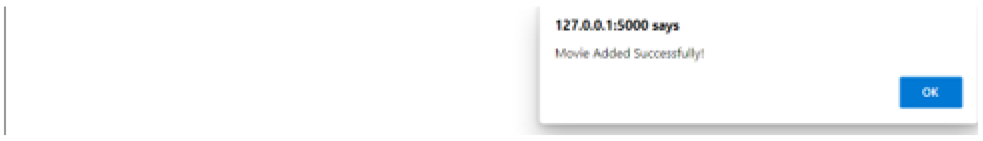
Director Name(Enter it correctly it is uneditable):

Imtiaz Ali

Language(hi for hindi and en for english):

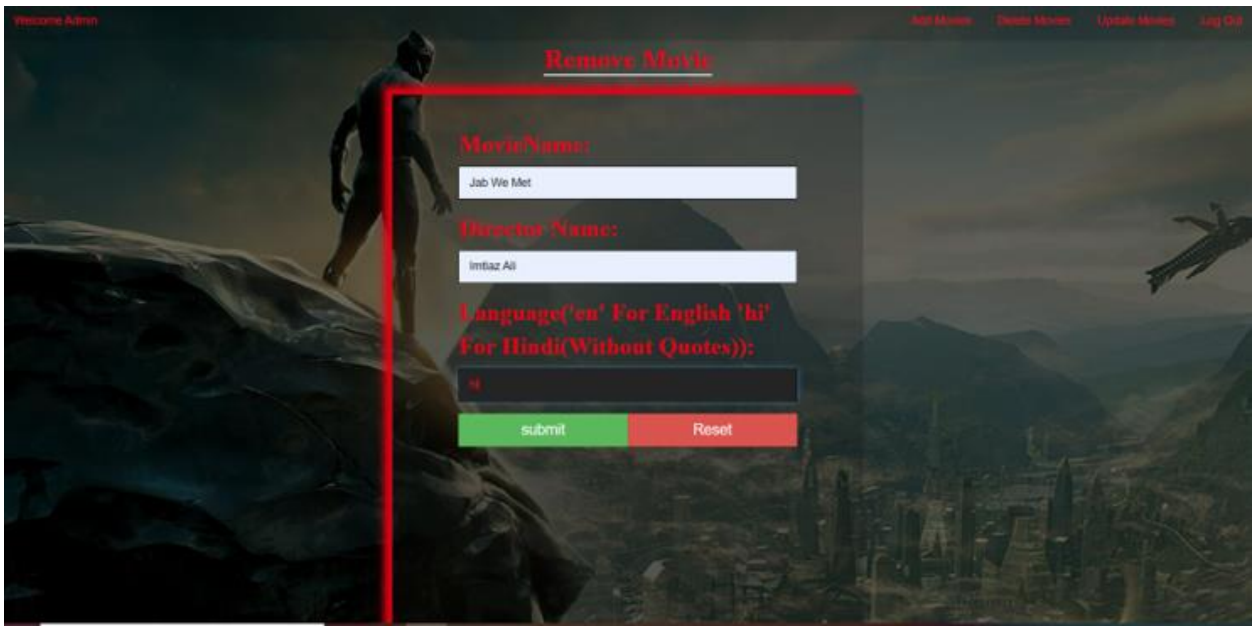
hi

Genre:



127.0.0.1:5000 says

Movie Added Successfully!



Remove Movie

MovieName:

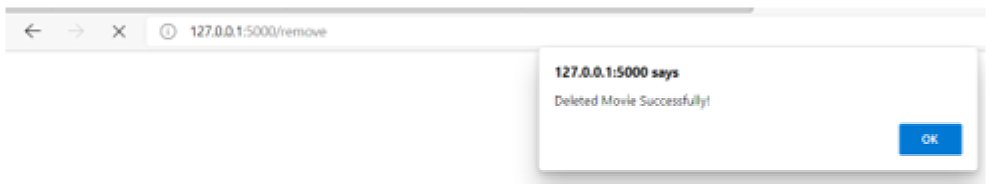
Jab We Met

Director Name:

Imtiaz Ali

Language('en' For English 'hi' For Hindi(Without Quotes)):

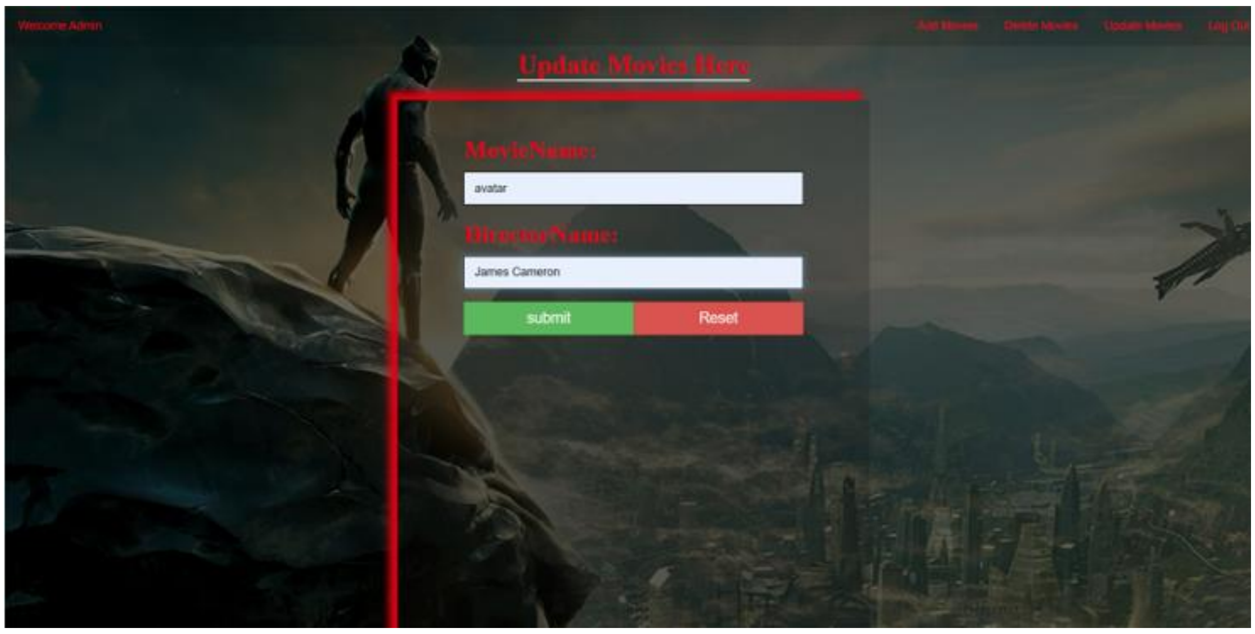
hi



127.0.0.1:5000/remove

127.0.0.1:5000 says

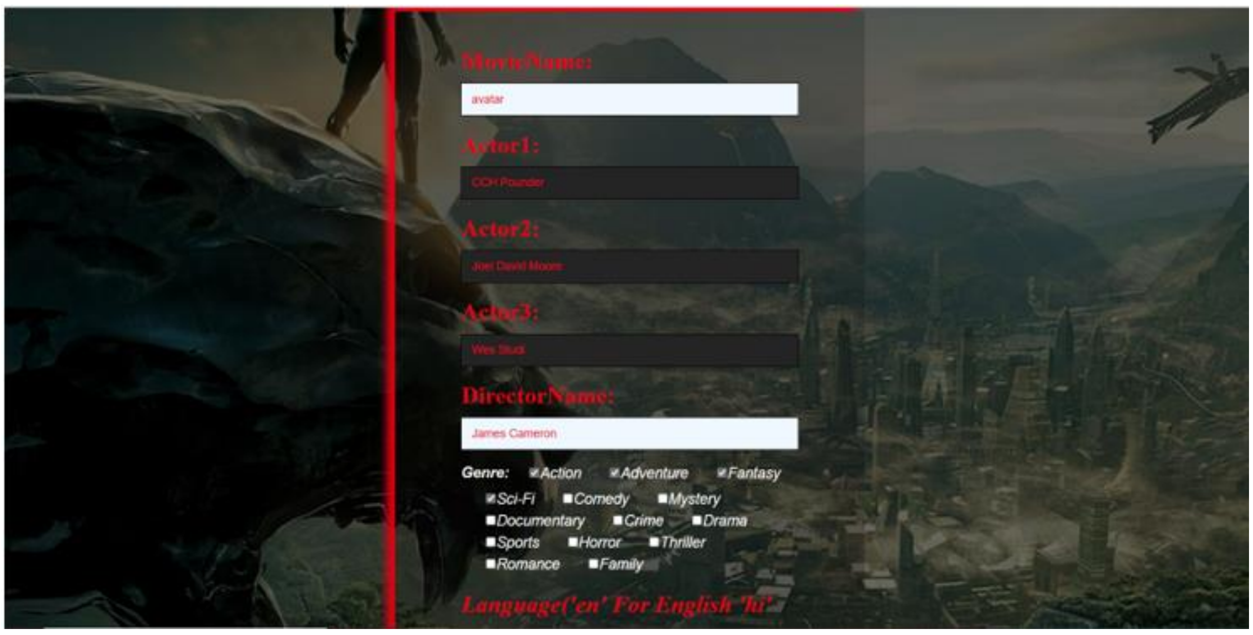
Deleted Movie Successfully!



Update Movies Here

MovieName: avatar

DirectorName: James Cameron


MovieName: avatar

Actor1: CCH Pounder

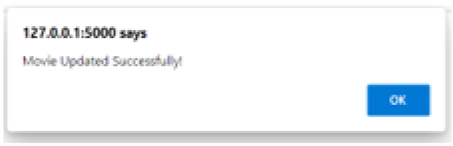
Actor2: Joel David Moore

Actor3: Wes Studi

DirectorName: James Cameron

Genre: [✓ Action] [✓ Adventure] [✓ Fantasy] [✓ Sci-Fi] Comedy Mystery Documentary Crime Drama Sports Horror Thriller Romance Family

Language('en' For English 'hi'


127.0.0.1:5000 says

Movie Updated Successfully!

Conclusion:

- So, We Have Implemented all the functionalities mentioned in SRS Document that are as follows:

    1. Admin-Login: It lets the admin to enter the system to Add, Delete and Update movies.
    2. Add-Movie: it Lets admin add a movie to dataset with proper validations.
    3. Remove-Movie: it Lets admin remove a movie from dataset with proper validation.
    4. Update-Movie: it Lets admin update a movie which is in dataset with proper validations.
    5. Admin-Authentication: it Verifies if Credentials entered by admin are correct or Incorrect.
    6. Search-Movie: it Lets user search a movie based on its title and language.
    7. Recommend-Movie: it Returns a list of movies similar to given movie name passed as string input.
    8. Review-Sentiment-Analysis: it Returns whether a review passed as string input is Good or Bad.
    9. Web-Scraping-Reviews: it Returns a list of reviews fetched through web scraping on imdb website by using IMDB-ID of the movie.

Limitation and Future Extension:

Limitations:

- One of the limitations of our project is that, it covers only Two language of movies that is English and Hindi. Other Limitation is that it does not cover all movies of these Languages Due to RAM Crash while performing Cosine Similarity.

Future Extensions:

- We can Add all movies and make it supportable for all languages by thinking of some alternative of cosine similarity. We can make it mobile responsive. We can make User Module who can rate, write reviews etc.

Bibliography:

1.) https://flask.palletsprojects.com/en/2.0.x/
2.) Cosine Similarity - GeeksforGeeks
3.) Learn Naive Bayes Algorithm | Naive Bayes Classifier Examples (analyticsvidhya.com)
4.) Naive Bayes Classifiers - GeeksforGeeks
5.) jQuery API Documentation
6.) API Overview — The Movie Database (TMDB) (themoviedb.org)
7.) 3. Numpy — Pandas Guide documentation
8.) pandas - Python Data Analysis Library (pydata.org)
9.) scikit-learn: machine learning in Python — scikit-learn 1.0.2 documentation
10.) Template Designer Documentation — Jinja Documentation (2.11.x) (palletsprojects.com)