MOVIE RECOMMENDATION SYSTEM

DIVESH SONI *VINAY KUMBHAR* *NIKITA NAIR*
Rochester Institute of Technology, Rochester, NY

♦ BACKGROUND

- Recommendation system comes in handy when there are a lot of options to choose from.
- Detecting the frequently watched movies by a user can help to determine the pattern of the movies to the user's liking.
- Users are presented with an option to choose a movie of their liking based on these patterns.
- This project aims at detecting the pattern of the user based on the previously watched movies in order to recommend movies according to the users' liking.
- A movie recommendation system helps the user select from a range of options presented to him based on analyzing the preference of the user.

♦ COLLABORATIVE FILTERING

• Collaborative filtering approach is employed to formulate a list of similar movies to every other movie in the data set.

•Item-based collaborative filtering is used to calculate the similarity of each movie in the data set to every other movie that exists in the data set employing the item-based collaborative filtering paradigm.

• Pearson Correlation Coefficient gives the most accurate results for filtering for our sparse dataset.

• The coefficient considers the users that have rated both the items and calculates the sum and sum of squares of the ratings for both items and the sum of products of the pair of ratings by each for both the items under consideration

♦ DESIGN

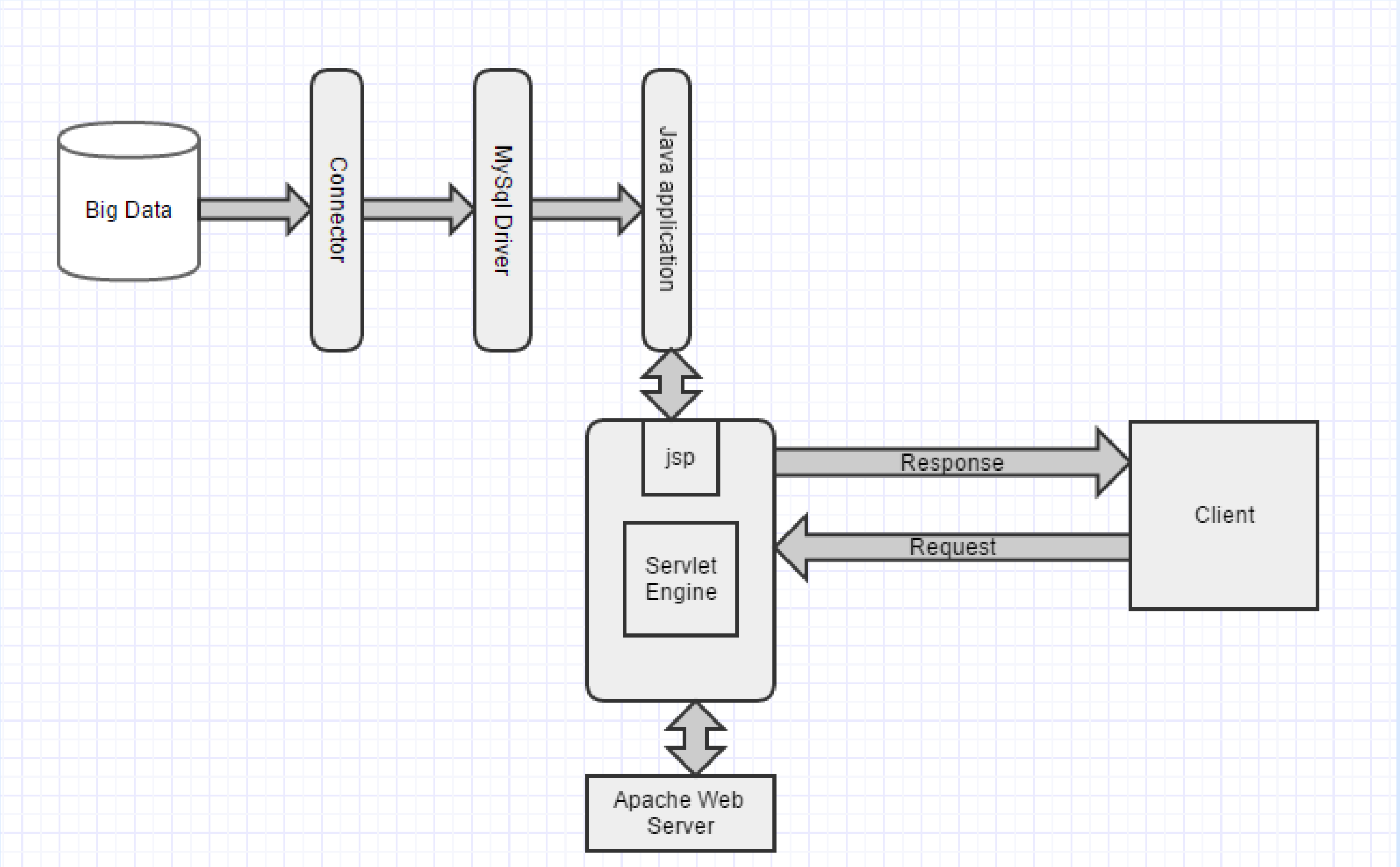
• Data is stored in MySQL database ,accessed using XAMPP server.

• XAMPP provides Apache Tomcat and MySQL servers, which makes data accessible to the Java application using JDBC.

• Java application also requires mysqlconnector which is an open source java connector. The integration of client side and back end is done using Java Server Pages (JSP).

• Servlets are hosted on Apache Tomcat server.

• The output of these servlets is then dynamically loaded on the client side to display the recommendation list



System Design

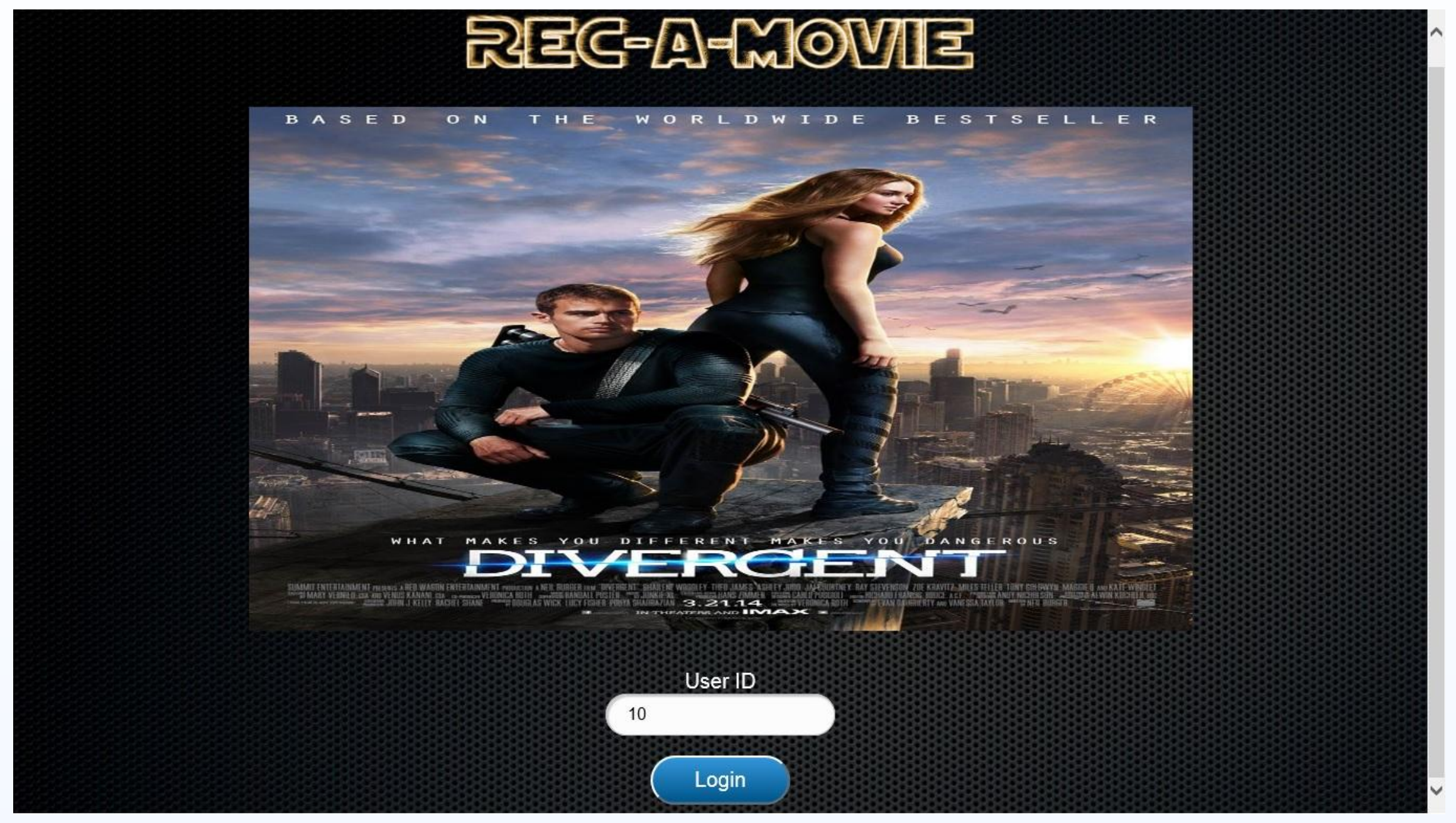
♦ IMPLEMENTATION

• Java was used for the cleaning of the data set and the user data file was converted for the ratings.

• Pearson Correlation Coefficient is used for calculating similarity between two movies for the Item-based Collaborative Filtering.

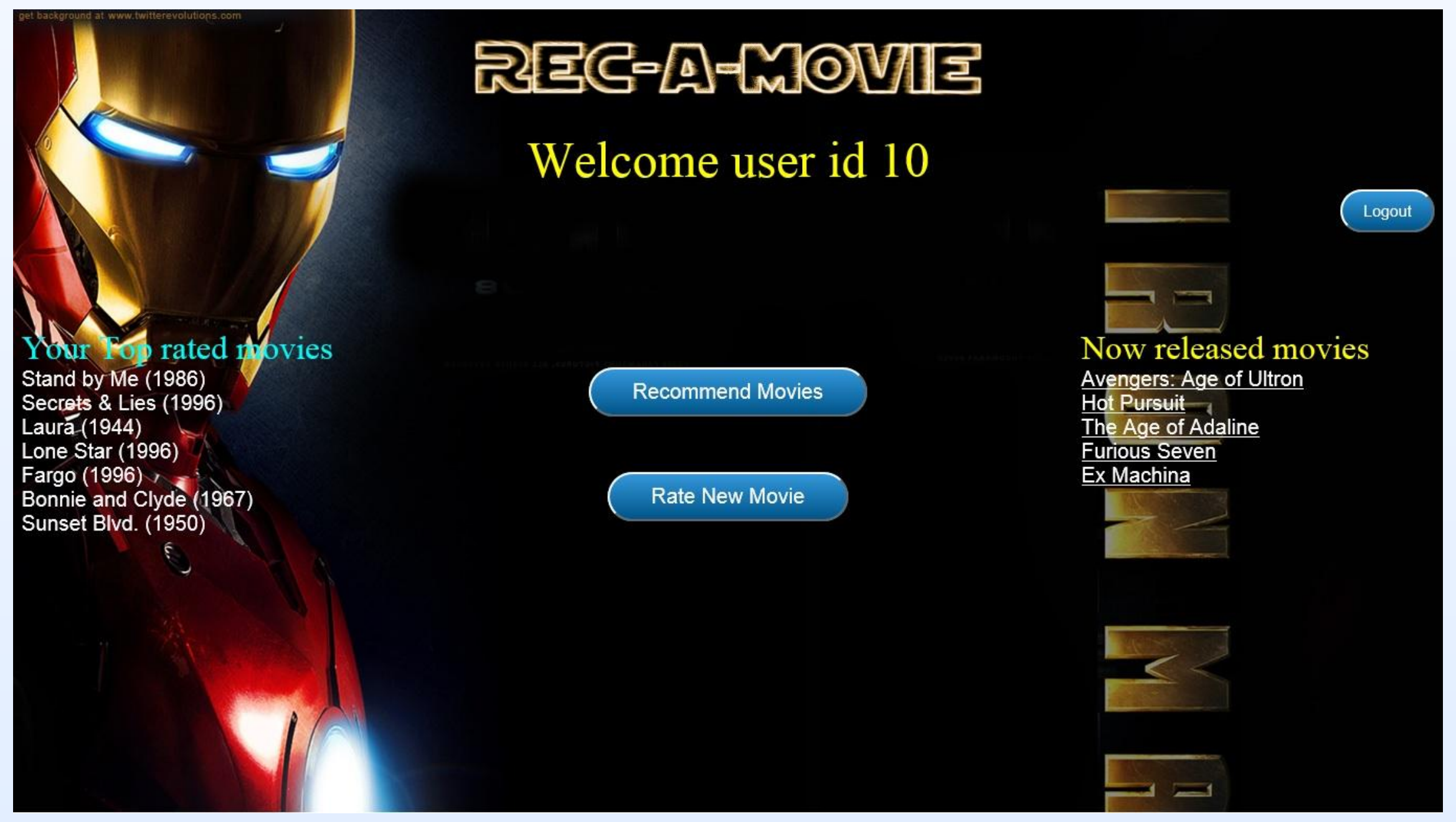
• Considering a particular user's top rated movies, movies similar to those are recommended to that user.

•New movies can be added by users along with their ratings or ratings for previously not reviewed movies can be provided by the users.


REC-A-MOVIE
BASED ON THE WORLDWIDE BESTSELLER
WHAT MAKES YOU DIFFERENT MAKES YOU DANGEROUS
DIVERGENT
3.21.14
Login

User ID: 10

Home Page

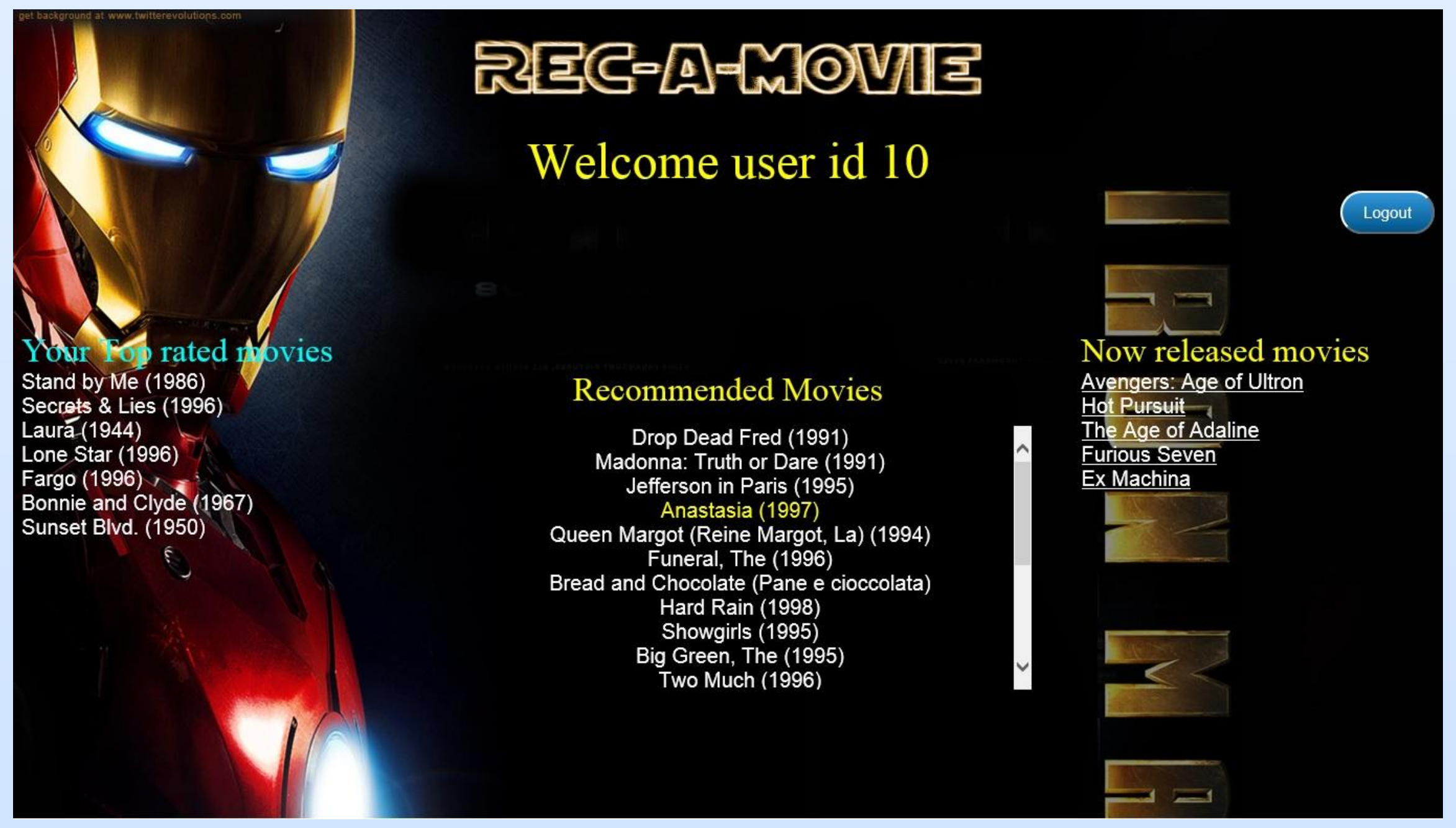


Welcome user id 10

Your Top rated movies
Stand by Me (1986)
Secrets & Lies (1996)
Laura (1944)
Lone Star (1996)
Fargo (1996)
Bonnie and Clyde (1967)
Sunset Blvd. (1950)

Now released movies
Avengers: Age of Ultron
Hot Pursuit
The Age of Adaline
Furious Seven
Ex Machina

Recommended movies by User 10



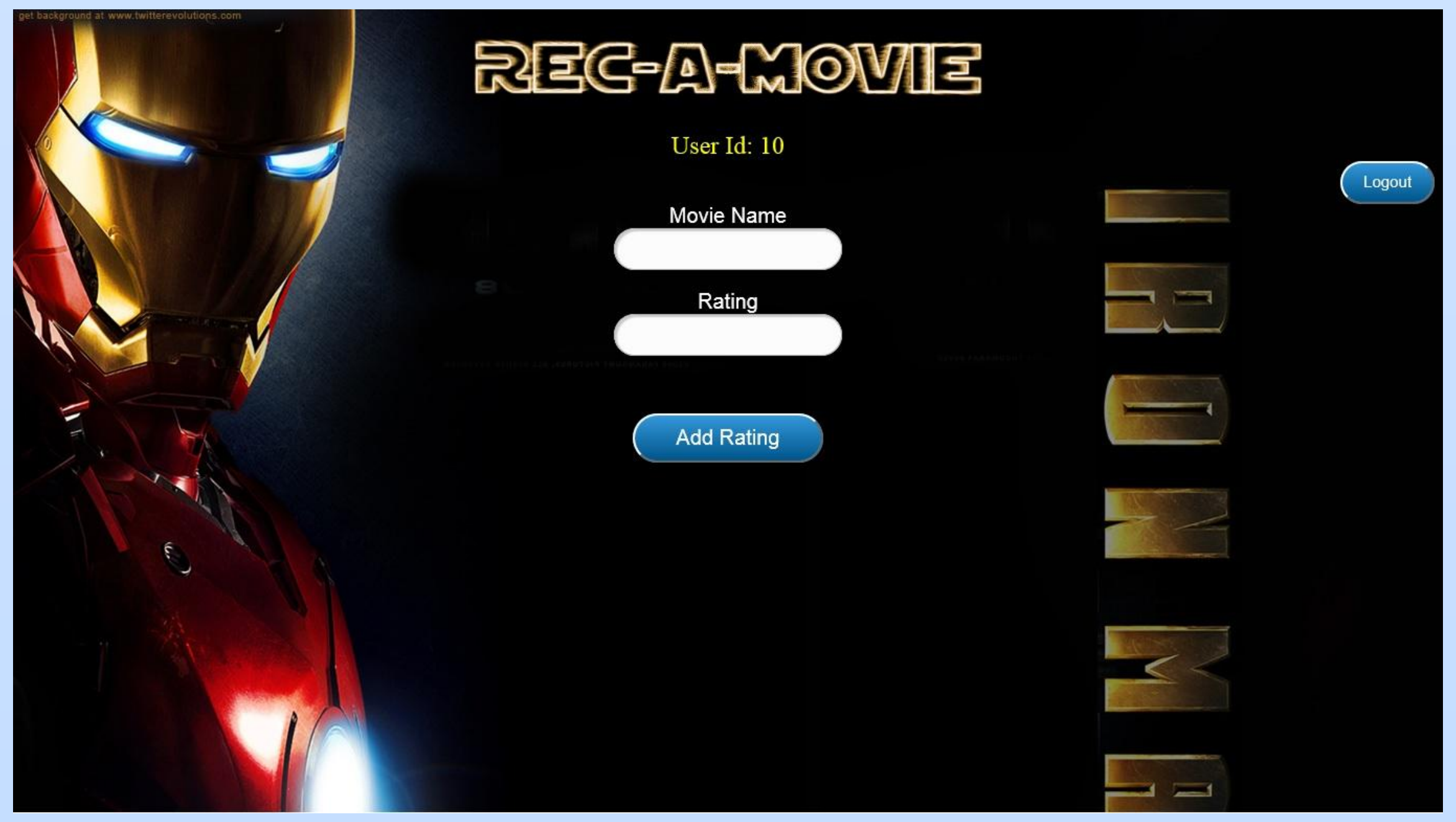
Welcome user id 10

Your Top rated movies
Stand by Me (1986)
Secrets & Lies (1996)
Laura (1944)
Lone Star (1996)
Fargo (1996)
Bonnie and Clyde (1967)
Sunset Blvd. (1950)

Recommended Movies
Drop Dead Fred (1991)
Madonna: Truth or Dare (1991)
Jefferson in Paris (1995)
Anastasia (1997)
Queen Margot (Reine Margot, La) (1994)
Funeral, The (1996)
Bread and Chocolate (Pane e cioccolata)
Hard Rain (1998)
Showgirls (1995)
Big Green, The (1995)
Two Much (1996)

Now released movies
Avengers: Age of Ultron
Hot Pursuit
The Age of Adaline
Furious Seven
Ex Machina

Recommended Movies for User 10



User Id: 10
Movie Name:
Rating:

User adding new review

♦ LESSONS LEARNED

• This project gave us an insight into handling large amounts of data.

•The problem of handling sparse data set as ours, lead us to explore various techniques to improve accuracy while dealing with it. We used Pearson Correlation Coefficient to improve accuracy.

• Filtering and indexing were implemented, that helped in the efficient processing of data.

•The use of indexing for specific columns that were a part of the 'where' clauses reduced the computational time by a tenth of the time that was required to run those queries.

♦ CONCLUSION

• Using the data set with information pertaining to movies, and ratings for a subset of those movies from an enormous user-base, our recommendation system predicts a set of movies that is likely to be top-rated by that particular user.

• Item-based collaborative filtering is used, which gives the most similar movies to each movie in the data set.

♦ FUTURE WORK

• Using query optimization, the computational time to calculate the similarity can be reduced .

• The current trending movies can be included by analyzing Twitter feeds and "trending" Facebook topics

♦ REFERENCES

1.GroupLens Research Group. MovieLens 100k
http://grouplens.org/datasets/movielens/

2. GroupLens Research Group. MovieLens 100K
http://files.grouplens.org/datasets/movielens/ml-100k/u.item