ReadMe

***Objective:*** To create a data-driven information System

***Goal:*** Create an application which enables users to access the huge movie database of IMDB and Netflix. The user can easily explore, keep track of, and discover movies and television content.

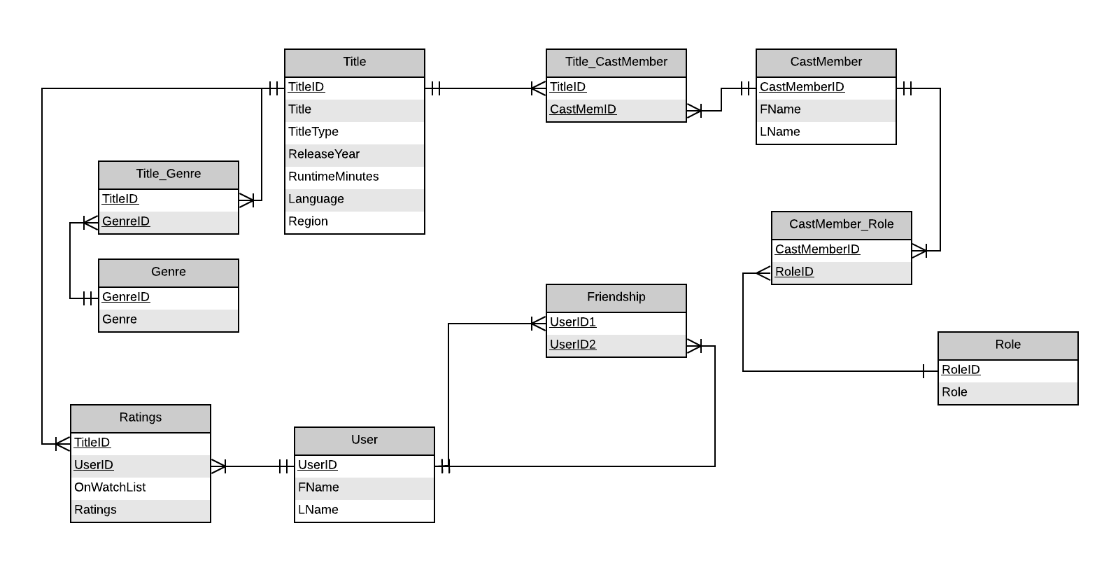
***Dataset***

1. The IMDB data set: <https://datasets.imdbws.com/>
2. Information about the data: <https://www.imdb.com/interfaces/>
3. Netflix <https://www.kaggle.com/netflix-inc/netflix-prize-data#movie_titles.csv>

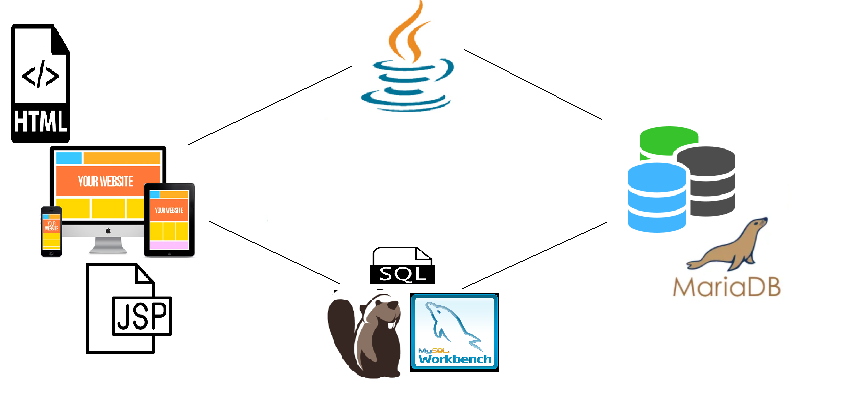
***Value Created***

1. Personal
   1. Allow users to track movies that they have seen and wish to see and associated personal ratings.
   2. Allow users to easily explore the dataset by querying via actors, TV shows, writers, producers, and any other person entity. Allow the user to view aggregate data for each.
   3. Give actors an average rating that users can see by averaging the ratings of the movies they acted in
2. Social
   1. See what your friends are watching
   2. Allow you to view a friend’s reviews, etc.
3. Recommendation Engine
   1. Suggest new movies and TV shows based on the user’s history and preferences
   2. Generate recommendations based on queries. For example, if a user likes a particular genre, look up highly rated movies in that genre.
   3. Show personalized recommendations in one pane and also “what your friends are watching” in another panel.
   4. Combine the netflix database (682 MB) to provide recommendations to users based on other individuals’ ratings

***Entity - Relationship Diagram***



***Architecture***



***Installation***

1. ***Setting up clusters***

We installed a centos OS on AWS. By using instructions from (<https://galeracluster.com/library/training/tutorials/aws-galera-cluster.html>) we setup 3 instances of a MariaDB Galera cluster.

1. ***Creating the interface***

We first designed wireframes of user interface and created HTML code based on that.A simple front end UI was developed with GIFs integrated to the application. CSS, HTML and JSP tags were used to develop the application.

***Data Cleaning***

These are the requirements we followed for cleaning the data.

1. Keep only the columns that are stated in the ERD
2. Combine the various title tables for normalization purposes.
3. Create keys for the genre and role tables so that looking up data is easier.
4. Remove duplicate records
5. For the ratings table, combine the IMDB and Netflix dataset and only keep the titles associated with Netflix. Then merge the combined table with given ratings table.
6. Create first and last names for the users.

We mainly used R to clean the data for these reasons:

* R was able to split a “cell” with multiple values. Example: A title can have multiple genres. The genres listed were “action, documentary, comedy”. R is used to split the data into 3 seperate rows (“action”, “documentary”, “comedy”)
* R can be used to work with large amounts of data

***Data loading***

We used mysql workbench and DBeaver as tools to load data into nodes, which is faster.

***How to use***

* Login to the application by username.
* The application then lets you:
  + Search for a movie
  + Search for an actor
  + Check out movies you’ve rated
  + Obtain friend recommendations
  + Capture your movie watchlist
* Users can then logout of the application

***Code references & Instructions***

1. We used eclipse as our IDE to write our Java and JSP code
2. In order to implement a Login functionality, servlets where used. The following link was used as a template to implement this feature: (<https://krazytech.com/programs/a-login-application-in-java-using-model-view-controllermvc-design-pattern>)
3. HTML and css tags were referenced from: <https://www.w3schools.com/>
4. JSP tags and functionality was referenced from: <https://www.tutorialspoint.com/jsp/jsp_database_access.htm>
5. GIFs have been used in the application, they are referenced from:
   1. <https://media1.giphy.com/media/pOTQnulT6tlrhYfNKU/giphy.gif>
   2. <https://cdn.dribbble.com/users/1446436/screenshots/5516645/my-movie-desktop_watchlist_dribbble.gif>
   3. <https://thumbs.gfycat.com/ReflectingNeglectedBug-size_restricted.gif>
   4. [https://img.buzzfeed.com/buzzfeed-static/static/2018-09/24/8/asset/buzzfeed-prod-web-02/anigif\_sub-buzz-2655-1537793305-5.gif?downsize=700:\*&output-format=auto&output-quality=auto](https://img.buzzfeed.com/buzzfeed-static/static/2018-09/24/8/asset/buzzfeed-prod-web-02/anigif_sub-buzz-2655-1537793305-5.gif?downsize=700:*&output-format=auto&output-quality=auto)
   5. <https://media.tenor.com/images/7778d8e5b309d6b331ca1f06ff886559/tenor.gif>
   6. <https://media3.giphy.com/media/BWySufD6KWQzC/giphy.gif>