* 1. COMP 2406 Final Project Report
  2. Movie Database

Thomas Soubliere 101144900

Wenyu Zhang 100941511

Date: dec. 6th, 2020

Fall, 2020

1. Running instructions:

a. The openStack address and password are as below:

IP: 134.117.134.114

VM\_USERNAME: student

VM\_PASSWORD: student

* 1. b. To start the database.

node database-generator

* 1. c. No extra initialization required
  2. d. To run the server.
     + 1. node server
  3. **2. Provide a summary of what functionality you have implemented successfully and what functionality you have not implemented.**

User Accounts

* Change between a ‘regular’ user account and a ‘contributing’ user account.
* View and manage the people they follow.
* View and manage the other users they follow.
* View recommended movies.
* Search for movies by title, name, and/or genre keyword, at minimum.

Viewing Movies

* See the basic movie information
* See the genre keywords and allow the user to navigate to search results that contain movies with that genre keyword.
* See the director, writer, and actors the movie has, which should also allow the user to navigate directly to each person’s page.
* See a list of similar movies to this one
* See movie reviews that have been added for the movie.
* Add a review

Viewing People

* See a history of all of this person’s work. Each movie entry should allow the user to navigate to that movie’s page.
* See a list of frequent collaborators of this person. That is, a list of people this person has worked with the most, according to your database information.
* Choose to follow this person.
* If a user follows a person, the user should receive a notification any time a new movie is added to the database that involves this person, or any time this person is added to an existing movie.

Viewing Other Users

* See a list of all of the reviews this user has made and be able to read each full review.
* See a list of all of the people this user has followed and be able to navigate to each person’s page.
* Choose to follow this user. If a user X follows a user Y, user X should receive a notification any time user Y creates a new review.

Contributing Users

* Add a new person to the database by specifying their name.
* Add a new movie by specifying all of the minimum information required by your system, including at least one writer, director, and actor.
* When viewing a movie, be able to edit the movie by adding actors, writers, and/or directors.

REST API

* GET /movies – Allows searching for movies in the database.
* GET /movies/:movie – Allows retrieving information about a specific movie with the unique ID movie, assuming it is a valid ID.
* POST /movies – Allows a new movie to be added into the database. It will accept a JSON representation of a movie and is responsible for checking the data is valid before adding it to the database.
* GET /people – Allows searching for people within the movie database.
* GET /people/:person – Retrieves the person with the given unique ID, if they exist.
* GET /users – Allows searching the users of the application.
* GET /users/:user – Get information about the user with the given unique ID, if they exist.
  1. **3. Describe any extensions you included beyond the required specification.**

1. A mongoose database is included in the project to store movies, people and user’s data.
   1. **4. Discuss any design decisions you made that you believe increase the overall quality of your system. Some important things to think about in this regard include the scalability, robustness, and user experience.**
   2. **5. Discuss any improvements to your system that you think could be made to increase its overall quality. This is an opportunity to demonstrate your understanding of course concepts that you feel were not adequately demonstrated in your project implementation.**
2. Improve the UI design
3. Allow users to upload their profile photo
   1. 6. Identify any modules, frameworks, or other tools that you used and justify their use.
   2. 7. What do you like most about your project? What would you say is the best feature(s)?