Richard Kim

101138475

COMP2406

Final Report

1. Instructions for initializing:

1) Unzip the downloaded zip file

2) Travel to the directory in cmd

3) Create a directory (folder) where the database will stay (call it database).

4) Use the command 'npm install' to install all dependencies listed in package.json

5) Run mongod using command "mongod --dbpath=database"

6) If mongod is running successfully, open a second command, travel to the directory.

7) Run the command "npm run start" to initialize the database and start the server. This special command was set up so TAs do not need to write long file names.

8) go to http://localhost:3000/ to start navigating the web application

2. Functionality

Most of the functionality mentioned in the specification has been implemented successfully.

- A user can sign up for an account.
- A user can log in if he/she has an account.
- Once a user log in, they may visit their profile page. Click the href ‘Profile’.
  - User may switch between “regular” and “contributing” account.
  - User may follow people they want and manage their list of people followed. They can also navigate to the personal page of any person they have followed. They can press the ‘Unfollow’ button to unfollow them anytime.
  - User may follow users they want and manage their list of users followed. They can also navigate to the user page of any user they have followed. They can press the ‘Unfollow’ button to unfollow them anytime.
  - User may view the list of movies they have watched (the movies they selected to be in their ‘watched list’. They can also navigate to the movie page of any movie they have ‘watched’. They can press the ‘Delete’ button to remove the movie from the watched list anytime.
  - User may view 5 movies recommended to them, based on the people they

followed. More detailed explanation of the algorithm will follow later.

  - User may view notifications sent to them. They will be notified if a person they followed has a new movie in the database (a new movie featuring the person is added), or a user they follow wrote a new review. User may navigate to the movie page the notification was about.

- Any users, either logged in or not logged in may use the movie search function. Click the href "Search" to view the list of movies in the database. The list is broken up to pages, with 10 movies per page. User may search for movies based on 1) the title of the movie, 2) or actors in the movie, 3) or genre of the movie.
  - If a user inputs multiple values to search for, note that the database will search for 'and' of those parameters. For example, searching for title of "hi" and genre of "War" will result in movies with title similar to "hi" with genre including "War".
  - The program searches for movies with similar title inputed.
  - The program requires exact name of actors and genres, separated by commas with no space in between. This means that actors need to have first letter of both first and last name capitalized, and genres also need to have first letter capitalized. For example, "First Last", and "Action". I did not have enough time to fully perfect it.
  - Click on hrefs 'Previous' and 'Next' to navigate to previous, and next pages of movies.
- Users either logged in or not, may view specific movies in their individual page.
  - See the basic movie information, including: the title, release year, average review rating, runtime, and plot.
  - See each of the genre keywords for the movie and allow the user to navigate to search results that contain movies with the same genre keyword.
  - See the director, writer, and actors the movie has, and be able to navigate directly to each person's page.
  - See a list of similar movies to this one and allow the user to navigate to the page for any of those movies. Similar movies were found based on similar genre; a more detailed explanation of algorithm will follow.
  - If the user is logged in, they may add the movie to their 'watched list'. This list will be viewable from the user's profile.
  - See movie reviews that have been added for the movie.
  - If a user is logged in, they may add a basic review by specifying a score out of 10. Or they may add a full review by specifying a score out of 10, a summary, and a full review text.

- Users may view specific users in their individual page by either: 1) getting the unique id of the user from mongo shell and writing it in the url (http://localhost:3000/users/:id), or more easily, view a specific review, and if you like the review, click the href that will take you to the user's page. Follow the user if you want to receive notifications when this user writes more reviews. Of course, to follow the user, you will need to be logged in.

- Users either logged in or not, may visit another user's page. Where they may:
  - 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.
  - See the user's 'watch list' and navigate to any of the movies listed.
  - If user is logged in, they may follow this user.

- Users either logged in or not, may view individual people's page. Where they may:
  - See a history of all of this person's work. Each movie entry allows the user to navigate to that movie's page.
  - The history of the person's work should be separated into categories for directed, written, and acted in. If the person has not directed, wrote, or acted in a movie (e.g., they have not acted in a movie), then that category is not shown on the page.
  - See top 5 list of frequent collaborators of this person. If there are less than 5 collaborators, only show however many there are.
  - If logged in, user may follow this person. If a user follows a person, the user receives a notification any time a new movie that includes this person is added to the database.

- All users are set to regular users as default. They may change their account type to be a contributor in their profile page, while being logged in. If a user is a contributor, they may:
  - Navigate to an "Add Person" page and add a new person to the database by specifying their name. If the name already exists, the user is not able to add the new person.
  - Navigate to an "Add Movie" page and add a new movie by specifying the title, release year, runtime, plot, genre keywords, and at least one writer, director, and actor. Unfortunately, the dynamic search function is not implemented.

- Unfortunately, there are not other additional functions implemented that are not mentioned in the assignment specification.

3. Critique of overall design

Overall, this application follows the RESTful design structure very well. All requests are categorized very RESTfully, into those most related. Get, post, put, delete requests are sent accordingly. The coding is also quite nicely done, with comments where to help user understand what is going on. Proper status codes are used in appropriate areas, and errors are all handled very RESTfully. Asynchronous operations are also very well implemented, with minimization of required data transfer.

While the efficiency of this program is not the worst, it is also not the best. There definitely are more efficient methods to build same functionality. This program does have some latency, and with a bigger database, it will only increase.

4. Algorithms

Movie recommendation algorithm reads the people the user is following and bases the recommendation from there. The program picks 5 movies that features any of the person the user is following and recommends it to the user.

Similar movie algorithm reads the genre of a given movie and finds other movies that contains any of the genre included in the original movie. I thought the genre is what defines the movies the best, and so concluded that similar movies should be movies in similar genre.

Frequent collaborator algorithm reads all the movies a given person has worked on. Then, the program finds all the people who worked for those movies, and lists each person depending on how many times this person’s name is counted. Thereby, ensuring people who shows up most in all the movies a specific person has worked on to be shown. Then top 5 people with names most shown up are selected. Top 1 will probably be the person him/herself, so the person given is excluded from their own list of top 5 collaborators.