

Table of contents

1. Launching instructions
2. Summary of functionality
3. Additional features
4. Design critique
5. Design of movie algorithms

1. To install dependencies, use the “npm install” command. Afterward, create a directory called database and run the mongo daemon. Navigate to the database-init directory and run “node database-initializer.js”. After the database is initialized, navigate back to the main directory to run “node MovieServer.js”. Once all that is done, go to <http://localhost:3000> to access the homepage.

2. Every required functionality was successfully implemented including the mongo database. A user is able to navigate to the movies tab and find a specific movie based on query parameters. Doing so will provide them a paginated list of movies that match the query. When a user navigates to a specific movie, they are able to view all information about the movie. A user may also add the movie to their watched list and may also remove it from the list via the movie page or their profile. Every person that worked on the movie is listed and the user is able to navigate to a page for that particular person. When viewing the person, the user is able to see a list of works that this person has done. In addition, a list of the top 5 people that this person has worked with will also be available to the user, and they can also navigate to that persons page if they click on their name. A user is also able to follow this person and they will receive a notification whenever a new movie is added with that person. Users can also follow other users and will receive notifications when the user creates a new review. When viewing other users, they are able to view their watched movies list as well as see who they followed and what movies they reviewed. When a user views their own profile, they are able to manage people they follow as well as movies in their watch list. They can also see a list of recommended movies. Users can also switch to a contributing account and add movies and people to the database.

3. An additional feature that was added was the ability to search for people and users on the server by specifying a name. Another feature added is the ability to navigate to the user profile that created the review for a movie.

4. In terms of aspects done well, our design is very RESTful. It uses four routes; movies, people, users, and accounts, and each link used is organized into those four routes. On top of that, we use proper HTTP status codes when delivering responses. Since our code is broken down into many files, it is quite scalable as we could easily add more functionality on top of editing current functionality. Our application also handles errors to prevent it from crashing the server, in addition to proper use of asynchronous functions. In terms of qualities that can be improved, there is a lot of repeated code such as querying the database as well as sending responses. One way this could be improved is by using predefined functions that can be repeated as many times as needed. Overall, proper planning and design would have also helped mitigate repeated code.

5. The similar movie algorithm is implemented on the basis of each movie's genre. The first step of the process is to loop through the current movie genre array that the user is on. Then the code will loop through each movie in the db and check if any of the genres in that movie matches with any of the genres of the movie the user is on, and pushes the loop through the movie to the similar movie array. While the loop checks if any of the genres matched it also checks if the movie that was not pushed to a similar movie array before. Once that algorithm is executed only the first five similar movies will be sent to the movie so that the user can see the similar movies. The recommended movies algorithm is based on the movies added to the users watched list. The server takes each unique genre in each of the movies in the list and checks if they intersect with the genres of other movies in the database. In general, the algorithm will take the first five movies in the database that have intersecting genres with the users watch movies list.