# Movie Database Project Report

## By Alexander Iwoh

**Intro**

For my website I used HTML, CSS, Javascript and a variety of Node.js modules. I mainly used "Express" to handle my server because Express handles a lot of tasks in an elegant way for me, such as: creating my server, handling routes, compiling template engines alongside "Pug", parsing requests, handling sessions, authentication, and authorization, etc. I truly enjoyed using Express and can see that this it will be around for quite some time. It is robust and powerful, making web design less tedious than it could be. For future improvement, I would consider using a database like MongoDB for my server instead of storing all my information in RAM like it currently does. The database would make modification and retrieval of data faster and more transferable to other programmers who take over my website. When entering the website from localhost:3000, the user is directed to the main.html page.

**Assumptions**

When making my website I assumed that the initial movie data would be provided for us in a JSON file. I could then parse this JSON file and add the movie info into my server's "logic.js" file. I ended up adding 100 or so movies from the JSON file into my logic.js file in order to have an initialized movie server. I also assumed that the actor/director/writer (movie attribute) names were unique, which made searching and retrieval of data much easier.

**Constraints**

For the project we were constrained to Node.js and the requirement that all resources must be served by our server. As a result, we could not use outside sources for our data.

**User Account**

The website allows a user to create a new account on the login.html page, which can be navigated to by clicking "Login" on the sidebar from different pages. On the login.html page, the user is met with several textboxes and buttons grouped into sections. The first section ("Login to Account") allows the user to type in a username and password, then press the "Login" button to enter. If there are no matching usernames and passwords in the server, the user is alerted of their failure to log in. Up to 30 accounts have already been created as part of the server initialization. Some examples are:

- Name: a, username: a, password: a
- Name: TA, username: TA, password: TA
- Name: Dave McKenney, username: dave, password: dave

You can type these in the credentials boxes to login or create your own account in the second section ("Create New Account"). The user can fill in the required information then press the "Create" button to create an account. Note that the system will alert you if the name, username or email already exists in the server. So, you must create a unique username, name, email, etc. Finally, you can scroll to the "Forgot your Password" section at the bottom, which allows you to type in an email. The server will then match that email (if it exists) to a password on the database and alert the user of that password. (This is a very rudimentary interaction and is only meant for testing. In real life (in industry) we would actually send the username and password to the email) If no such email exists in the server, the user is alerted. Finally, note that you can automatically navigate to each of the sections by clicking the "Login", "Create Account", and "Forgot Password?" links on the sidebar.

When pressing "My Account" it will take you to your account.html page which shows some of your information, if you are logged in. If you aren't logged in, you'll be redirected to the login.html page. On your account page you will see your name and credentials as well as a stock person image. On the bottom you'll see a list of notifications from users you follow. It will show notifications when they've added a movie or review. On the right hand side you will see a button to switch your account type to and from contributor status (accounts start off as regular accounts) and a link to "Add Movie/Person", which takes you to addMovie.html page where you can add a Movie or a Person (provided you are a contributor). To see more information about your account you can press the "Edit/See More" link underneath your credentials. Note, this will be a running theme: in the various browsing pages you can click "Edit/See More" after selecting a user, movie, or person, to go their individual detailed pages. When you press Edit/See More from your account page, you'll be taken to a template-engine generated user page with your information (the same information that would be made public). There you can see users that you are following, your followers, people you follow, and the reviews you've written. Because the names of the followers and people are all links, you can navigate to any of their personal pages as well.

**Viewing Movies**

There are several ways to view movies in this app. At any point you can click the "Movies" link on the sidebar to go navigate to the Browse Movies page. On the bottom you will see a scrollable list of all movies in the server. This may look daunting, however, there is a "Filter" on the side that allows you to type in values for "Title", "Genre", "Year", and "minRating". As you type into the textboxes, the scrollable list will dynamically change, presenting you with a shrunken list that satisfies the criteria in the filter. Another way to filter your results is to type in a query after "/movies" or after "/Content/movies.html". Having a query where "?title=x&genre=x&year=x&minrating=x" will take you to the same movies.html page except with the scrollable list already filtered and the query parameters already entered in the Filter text boxes, based on your query string values. When you click on one of the movies in the scrollable list, you will be shown a preview of its data directly in the text pane above. You can then click the "Edit/See More" link to go to the movie's individual page. Upon selecting a movie from the scrollable list, the top right side will show a pane where you can add information to create a review for that movie (provided you are logged in and are a contributor). You can click the "Add Review" button to add the movie, provided the textboxes are filled, otherwise you'll be alerted. Also, entering incompatible values (like letters for numerical scores) will alert the user to change the value. If you are not logged in, you'll be redirected to the login.html page. If you are logged in but not a contributor, you'll be redirected to your account.html page so that you can click the "Contributor" button yourself.

When you click "Edit/See More" to go to the individual movie page, you'll see the movie's name, genres, year, running time, plot, directors, cast, writers, average score, and written reviews. The listed genres are all links, so clicking on them to redirect you to the movies.html page except the query parameter will be “?genre=x” where “x” is whatever genre you clicked.

The average review score for the movie is the average of all its reviews, and yes, even including the reviews that had “x%” or “x/10”. We wrote a code snippet to parse the different review score types, convert them to numbers out of 10, and then calculate an aggregate review score for the movie.

**Viewing People**

You can click the “People” link on the sidebar to navigate to the Browse People page. There you will also see a scrollable, list, except of People this time. You will see a filter underneath the list to filter the scrollable list by name. Like before you can type in a query into the “/people” or “/Content/people.html” to refresh the page with an already filtered scrollable list. By clicking a person in the list, you’ll see a preview in the pane above and an “Edit/See More” link, as well as a button to follow or unfollow the person on the top right. You can only follow a person if you’re logged in, so if you aren’t you will be redirected to the login page. Clicking the “Edit/See More” will take you to the Person’s individual page where you can see their name, movies, collaborators (People they’ve worked with) and followers (users that follow them). The stated movies, collaborators, and users are all links that will take you to the movie, person, and user pages respectively.

**Viewing Other Users**

By clicking the “Accounts” link on the sidebar, you will navigate to the accounts.html page, where there is also a scrollable list, except this time with users on the server. There is also a filter on the bottom to filter the list by name, and you can also add a query to the “/users” or “/Content/accounts.html” to start with the pre-filtered list. By clicking a user in the list, you’ll see a preview of their info. On the top right, will appear a pane to add or remove this user as a friend (you can’t add yourself of course). This will only work if you’re logged in, otherwise you will be redirected to the login page. You can click the “Edit/See More” to go to the individual user page. On the individual user page, you will see users that they are following, their followers, people they follow, and the reviews they’ve written. Because the names of the followers and people are all links, you can navigate to any of their personal pages as well.

**Adding a Movie or Person**

When logged in to your account.html page you can click the “Add Movie/Person” link on the right side to go to the addMovie.html page. Here there are two sections: the first section for movie addition, and the second for Person addition. Both sections have textboxes to take in required information, and you will be unable to add any movie on person without filling in all required boxes for that section with appropriate values. Clicking the “Add Movie” button will create a movie on the server as well as any new people that are not already in the server. Clicking the “Add Person” button will create a person if there are not already in the server.

**REST API**

My API is handled by routes named: people-router.js, accounts-router.js, movies-router.js, and login-router.js. When a GET /movies request is made, movies-router.js will return a list of all movies in the database as a JSON object (if ‘application/json’ is the ‘Accept’ header type) or it will redirect the user to the movies.html page with all the movies in a scrollable list. If query parameters are added, then the returned list will be filtered based on the query parameters before being sent as JSON or redirecting the user to the movies.html page with a filtered scrollable list. The acceptable query parameters are: *title*, where the router will include a movie in the results if the movie name includes that *title* string.; *genre*, where the router will include the movie if the movie's genre includes the *genre* string; *year* which will include only results where the movies' release year matches; and *minrating*, which will include results where the movie's average score exceeds the minimum specified rating. A request of GET /movies/:ID will either return a JSON object containing the movie object (and its information) with that ID (which is used by the scrollable movie list to display the preview), or will redirect the requester to the individual page for that movie. A request of POST /movies/addMovie allows for a movie to be created by taking in an object as the request body: the object is in the format "obj = {txtName: "", txtGenres: "", txtYear: *Whole Number*, txtDirectors: "", txtCast: "", txtWriters: "", txtReview: "", txtRating: "x/100"};", where the txtRating is out of 100, in string format. A request to GET /people will take you to the people.html page with the scrollable list and the GET /people/:person will take you to the person's individual page with *person* referring to the person ID. Likewise, GET /users will take you to the accounts.html page with the scrollable list of users and GET /users/:user will take you to their individual page with *user* referring to their ID. Not that for both users and people pages, you can add a query parameter "?name=x" to filter the results as described above. The GET requests to users and people will also return JSON in cases where the 'Accept' header is 'application/json'. The JSON will contain relevant information such as a user's reviews.

**Future Changes**

If given more time, I would have liked to explore using databases to speed up the server and make it more modular. I would have also liked to add a deep recommendation system for users, people, etc. I would also have liked to have the ability to send emails to users who would like a subscription. Hopefully with more web development practice I will achieve such dreams.

**Final Notes**

Also included in the folder is a Readme.txt file that details the purpose of each file in the server directory as well as the directory tree for navigation. Please read that file to learn more about the file usage in this server.