

Project Report

Successful Functionality

User Accounts

- Provide a way for users to create new accounts by specifying a unique username and password.
- A user is able to log in and out of the system using their username and password.
- Within a single browser instance, only a single user can be logged in at one time.
- All newly created accounts are regular users but can be upgraded upon sign up or at any time on the users profile page.
- Users can change between a 'regular' user account and a contributing user account .If a user changes account types, it only affects their ability to carry out an action in the future (adding/editing movies and people).
- Users can view and manage the other users they follow. Users are able to navigate to the user page of any user they follow and can stop following any user that they have followed.
- Search for movies by title and/or genre keyword. The user is able to navigate to the movie page for any of the search results.

Viewing Movies

- See the basic movie information, including at minimum: the title, release year, average rating, runtime, and plot.
- See the movie genre
- See one director, writer and actor
- Navigate to people's profile pages
- See movie reviews that have been added for the movie.
- Add a basic review by specifying a score out of 10.
- Add a full review by specifying a score out of 10, a brief summary, and a full review text.

Viewing People

- Each movie entry on the persons page allows the user to navigate to that movie's page.

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, the user is able to edit the movie by adding ONE actor, writer, and director.

REST API

- **GET/movies**– Allows searching for movies in the database. Returns an array of movies that match the query constraints. Support title and genre query parameters.
- **GET/movies/:movie** – Allows retrieving information about a specific movie with the unique ID *movie*, assuming it is a valid ID. Returns the basic movie information (title, year, actors, etc.) and also information about the reviews of the movie.
- **POST/movies** – Allows a new movie to be added in the database. It accepts a JSON representation of a movie and validates data with Bootstrap form validation, and Javascript middleware
- **GET/people**–Allows searching for people within the movie database by their name or role and returns any person in the database whose name contains the given *name or role* parameter.
- **GET/people/:person** – Retrieve the person with the given uniqueID, if they exist, including their name and role
- **GET/users** – Allows searching the users of the application. Support an optional *username* query parameter. If the query parameter is included, it returns any user in the database whose name contains the given *username* parameter.
- **GET/users/:user** – Get information about the user with the given uniqueID, if they exist. Contains their username, their account status (regular user or contributing user) and the reviews the user has made.

Unsuccessful Functionality

User Accounts

- View and manage the movie people they follow.

Viewing Movies

- Users CAN see one genre key word, but it does not allow the user to navigate to search results that contain movies with that genre keyword.
- Users CAN see the director, writer, and actor the movie has, but does not allow the user to navigate directly their specific page, but they Can navigate each person in the databases page
- Users CAN see a list of all the movies and navigate to their pages, but not a list of similar movies to the one they are viewing

Viewing People

- Users cannot see a history of all of this person's work, but can see movie entries which allow the user to navigate to that movie's page.
- Users cannot see a list of frequent collaborators of this person
- Cannot follow people.
- Users cannot add people to a movie if that person exists in the database already, the can add them even if they don't exist

REST API

- Cannot search by year or migrating
- If no value is given for the search parameter, the page does not return all movies all, returns a blank page instead
- User search is not done in a case-insensitive nature. If no parameter is given, no users are returned instead of all

Extensions

- The interface adapts well to varying screen sizes (use the Chrome developer tools 'device toolbar' to test).
- The interface is very impressive visually.
- A database is used to store the system's data.
- Uses flash to send users alerts of successful task completion (ie. login, logout, sign up, create movie/person/review) and unhandled errors are shown on an error page that is visually impressive
- Uses Passport to authenticate user login and sign up

- Bootstrap frontend validation, Joi schema validation

Design Decisions

- Modular code design and MVC design allowed me to make changes to database or other components of my application

Future Improvements

- Improve on business logic, especially for working with databases. One of the main things I was unable to accomplish was recommendations, frequent collaborations, and properly adding People models to movies.

Frameworks and Modules

- Passport
- Bootstrap Joi