

Movie recommendation and social media application.

MODULE	CM3203 One Semester Individual Project
PROJECT TITLE	Creating a movie review based social media and recommendation system.
AUTHOR	Tyler Brown
SUPERVISOR	Walter Colombo
MODERATOR	Victor Gutiérrez Basulto

Acknowledgement

I would like to thank the following people:

- My supervisor, Walter Colombo for the constant support and help on the project and making himself available to guide me through the project and providing useful resources.
- My family, for the support and patience while conducting the project and helping with my time management.
- The following close friends for being part of the study and completing a survey to determine the project's success:
 - Ben Mann
 - Kurt Bradbury
 - Nathaniel Partington
 - Lewis Bowden
 - Malachy Jennings
 - Jack Bowns

Table of Contents

Contents

Acknowledgement	2
Table of Contents	3
1 Introduction.....	4
2 Background	5
2.1 Existing Technologies	5
2.2 Recommender Systems	9
2.3 User Interface and User Experience	12
3 Design and Approach.....	13
3.1 Technology Stack	13
3.2 Gathering Data	15
3.3 Database Design	16
3.4 UI/UX Design.....	18
3.5 Security and Authentication	21
4 Implementation	22
4.1 Login and Registration	22
4.2 Retrieving data	26
4.3 User Inputs	36
4.4 Recommendation System	40
4.5 Database Implementation	42
4.6 Issues.....	43
5 Evaluation	45
5.1 Survey Design	45
5.2 Survey Feedback	46
5.3 Comparison with Existing Technologies	47
6 Conclusion and Future Work	48
6.1 Key Findings	48
6.2 Limitations.....	49
6.3 Future Work	49
7 Reflection on Learning.....	50
References.....	51

1 Introduction

The world is currently dominated by digital media and streaming platforms, as media is enhanced the needs for personalized movie recommendation system and review based systems are inclining as Brits spend more than 100 days of their live trying to determine what TV show or movie to watch, changing what film they chose 9 times before deciding on one. (Alice Hughes, 2021). This project aims to address this demanded by creating a piece of software gives the user a platform to share, and catalogue their opinions and receive recommendations based on said opinions on a user-friendly, communal website where the focus is instead on receiving movie critics ratings and global statistics on how well a movie is performing global, but how you close friends perceived them and how your opinion differs to those closer to you.

The primary objective of the project is for the platform to allow users to discover, rate and discuss movies with the focus on social interaction being paramount, with the social media aspect and recommendation algorithm, the project strives to enhance usability and user experience within the online movie-fans community. Key features the project aims to include, should the project currently implement and develop these key features to a professional standard with a focus on user experience, the project can be deemed successful:

- **Personalized Recommendations:** Using an algorithm based on the users' input, whether than be friendships or reviews to give a set of movies that the applications predict the user should enjoy.
- **Social Relationships:** Creating a following system where users can choice who they wish for their account to be tracking and following, showing them that users' ratings and movies should they follow them.
- **Movie Search:** Allow the user to search for any movie that is currently available and see information relating to that movie, that may help decide whether the user wants to watch that movie or not.
- **Rating/Review System:** The user should be able to input a value on how much they feel the movie deserved in addition to a short amount of text that they can add to discuss what about the movie they enjoy or discuss any problems they felt the movie had.
- **User Profiles/Statistics:** Users should be able to create a customized profile displaying key information about themselves and their movie preferences to allow other users to easily find and distinguish friends or people of similar tastes in movies as themselves.

To successfully carry out this project, research must be done on already existing software that is like that of the project. An in-depth background search can help the project develop without previous tests of other systems where the project can use the success of other sites. This research can help implement features that are loved and robust and reduce wasting time on features that do not do so well. Research into review systems, like IMDB, Movie Database and Rotton Tomato's. In addition to a deep dive into other social media-based systems, like Letterboxd, Serialz and backlogd. By researching a wider range of systems, more information can be collected on what works and what does not. Research into different recommendation systems are also necessary to guarantee that the user receives a relevant and accurate recommendation, research into collaborative filtering, content-based filtering, and AI-based filtering to determine which is more relevant for this project. Development a robust, accessible, and user-friendly system requires a thorough examination into different technique of UI and UX design that can help the system feel more professional.

Tyler Brown

C21059883

The development of this project will require a prioritization of user privacy and security. The project must aim to ensure that minimal user data be stored, and any required personal data (like emails) be store in an efficient and safer manner with no risk of leaks. Password protection will need to be developed with some sort of hashing algorithm be implemented before passwords are stored.

The way this project can be determined a success is using a small sample of test users to test the systems effectiveness. By giving a small sample of test users anonymously answer questions after using the system. These questions will not require any personal data and only relate to the system. By doing a survey in this way, problems and successes can be determined by the results and future improvements and plans can be made to further develop the software. Users should thoroughly test the software and not only give feedback on UI, UX, recommendation accuracy, ease of use, in addition to any errors or features that didn't work as intended.

2 Background

This section will go into detail on specific research required to complete this project. All information and in-depth dive into different technologies, methods and approaches that should be considered before starting the project, a dive into already existing technologies, different recommendations algorithms as well as different user interface and user experience techniques that can be used for ease of use and aesthetics.

2.1 Existing Technologies

A dive into already similar technologies is require learning from, as any improvements and features used in this system could give the idea of what works a what does to work, as well as what the most usable approaches and system are. Since my system plans to be similar to these, however the plan is to implement a more social aspect to the system where the focus is on users sharing rather than users entering data.

IMDB

IMDB is one the most popular platforms online, it is a database of information and one of the largest at that, that stores information related to films, TV shows, podcasts, video games and other streaming content. It also stores all the cast, production crew and biographies as well as a load of narrative information on each piece of media.

Tyler Brown
C21059883

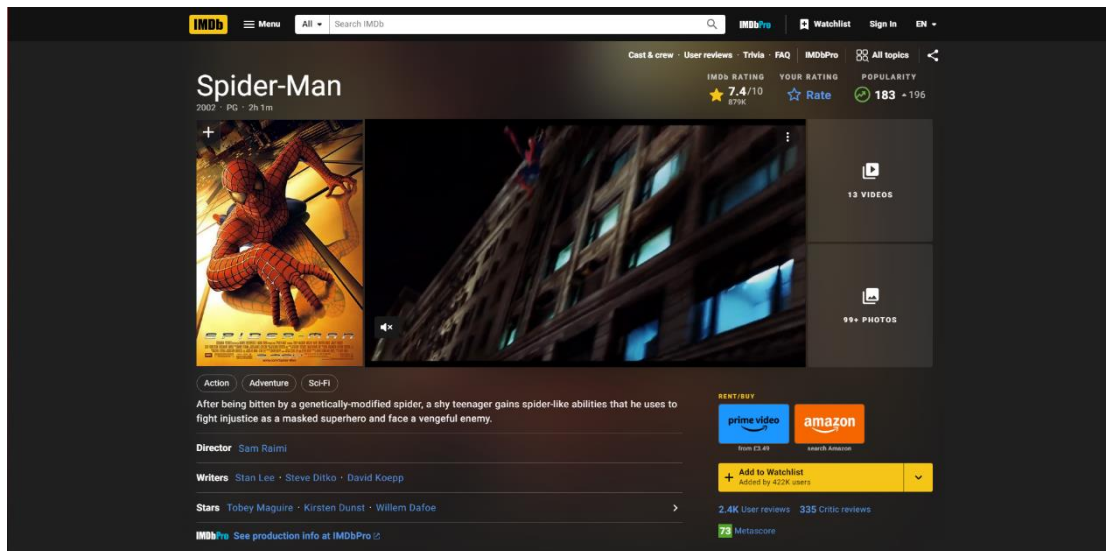


Figure 1 - IMDB Webpage on specific movie.

Some of the key features IMDB includes are, movies listings/search where users can browse and find the almost any movie or media they want and be provided with an abundance of information about said content, links to streaming platforms where users can find out where and how much it costs for them to gain access to this content. One of the main features of this platform that sets it apart is the separation of user reviews and ratings to critics' reviews or rating, this distinction makes the application more of a professional rating rather than a social communal rating system removing the social aspect despite comments are forum-based pages being present on the site. Despite the removal of the communal feel to the rating system, IMDB does have a lot of commenting and forum-based sections on their sites where users can discuss leave opinions and talk about different aspects of the movies and content they love, although the lack in connections between users is clear it does have some social media aspects to it. (IMDB, 2024)

Ease of Use	7
Aesthetic	5
Functionality	6
Accuracy	10
Responsiveness	9
Overall	7.4

The site is relatively easy to use, however the sheer amount of information displayed can be overwhelming which a load of things being clickable on the page, the page is pretty monotone giving it a boring, deflated look to it. Functionally the site works fine however some slowness on scrolling and clicking due to the number of images and information loading. The site is accurate with most

modern systems using the same database for their own information with most recommendations seeming related to the content being searched. The site acts responsively to the device but can become cluttered and crammed on smaller devices.

Rotten Tomatoes

Rotten tomatoes are described as a review-aggregation system for media content like films and shows, it is probably the most trusted rating system with its rating system, tomato meter being one of the most famous online being used but most critics and find to gauge how well a piece of media preformed.

Tyler Brown
C21059883

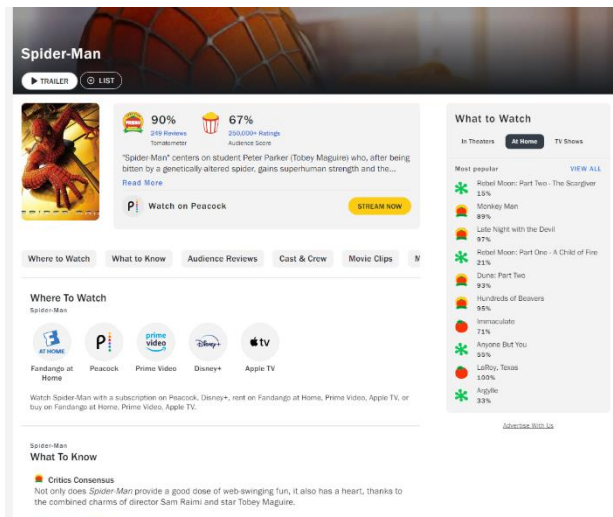


Figure 2 - Rotten tomatoes webpage

Again, one of the key features of rotten tomatoes is that the user reviews and critic reviews are separate, the tomato meter being critics reviews and another rating is the audience scores, however users do have to be verified to allow their score to count towards it. This site also allows users to click the “Buy Tickets” button which takes users to another site that shows where people can watch it and for how much allowing them to purchase tickets for it.

With a site like this it is important to consider how much of an influence a large rating system like this can influence sales on given movies, if the site is slightly bias, it can really affect how a movie sells as it is one of the leading applications to give users feedback on a movie choice they plan to watch. This site instead of having a recommendation system as a “What to watch” system which shows you what current films are in circulation or in cinema. In addition to this is also shows you upcoming shows for you to look forward to and plan a trip to the cinema. (Rotten Tomatoes, 2024)

The site is not intuitive to use but is not overly complicated, the site has a lot of tabs and drop-down menus that sometimes make it hard to find what you’re looking for. Aesthetically the site looks very old fashion with some outdate icons and colour schemes that gives it a very old look with a lot of white space. Information is on a professional grade accurate. The site works as intended responsively but some boxes overlap creating an ugly look to the site.

Ease of Use	5
Aesthetic	4
Functionality	8
Accuracy	10
Responsiveness	7
Overall	6.8

Letterboxd

Letterboxd compared to the other is a more social media movie review sharing platform rather than a professional critical reviewing platform. It known for having a large audience of casual watchers rather than movie critics who job it is to put their opinion out to the public. This makes for a friendlier and more accurate results when it comes to ratings due to it being more about how the public views the films rather than a small selection of movie critics.

Tyler Brown
C21059883



Figure 3 - Letterboxd movie webpage

Some of the key features Letterboxd advertises is its importance with sharing opinions and favourites with your friends / followers. One of the ways they do this is by adding a diary system where you can log and track where and when you watch films so you can reference them in the future and so that you can discuss with you friends when you watched it with the additional feature of rewatching a film adding a separate review and rating for each time you log it, in case your opinion has changed. Another key feature they have is the ability to create lists, in a list you can pick a certain theme, genre, actor, director, studio and so on, or even just your favourites and place a plethora of films into a list where you can discuss in the comments and share you list with others. You can also make these lists a ranked list so you can order the films from the best to worst.

Letterboxd's recommendation system is based on the users' friendships under a section in the home page called "popular with friends" which shows films that have been commonly watch but some or most of the people you are following in addition to its recommendation new releases to you in case you were unaware of their releases. It also shows you the most recent diary entries of you friend to see what they a watching in case you wanted to share an opinion with them if you have already watch it. (Letterboxd, 2024)

Ease of Use	8
Aesthetic	8
Functionality	8
Accuracy	7
Responsiveness	10
Overall	8.2

Letterboxd is incredible easy to use with most of the options available to you being very intuitive and simplistic in nature. The aesthetic of the site is very appealing with very clear borders between different section and appropriate highlighting on the most important parts of the page, borders are well rounding to give an appealing casual look. Most of the information is accurate however, a plethora of irrelevant media is accessible through this, and despite the site displaying that it's a film reviewing site some TV shows and YouTube shorts do make it through into the database, possible confusing users who are looking for something similar but only see a show.

2.2 Recommender Systems

There are many ways to decide what movies to display and recommend to users, there are simpler ways and there are more complex methods and algorithms that use different data to form a list of movies the user may like, the more complex algorithms and maths behind the system, usually the more accurate and more likely the user will be of liking and watching the film. Simpler methods of recommendation can include systems like:

- New films, recommend new films that have recently come to cinema to users who like to watch the newest films when they come out. Users can have a calendar of movie release in cinemas in their country. This implementation can easily be combined with other systems to only recommend new movies that the system thinks the user may enjoy rather than just the newest releases alone.
- Friends' activity, recommend films that users have recently been watching, users that the person has added to their friends list. The user can have an activity page where it shows all the films that their friends have watched and what they rated it so the user knows, if they often share opinions, that the movie might or might not be to their liking.
- Currently Popular, in larger systems, a recommendation system based on a larger audience where the system will calculate the most watched films over a specific time frame, often a week. Instead of a personal recommendation system this is instead global where all recommendations will be the same.
- Highest ratings, this system will be a dynamic system which changes based on the average ratings of movies. Recommendations like this only recommend films that are generally liked by most people scoring the highest.

Collaborative Filtering

Collaborative Filtering is a method of recommendation that uses only the users input of which it is going to recommend too. (Victor Dey 2021) In this instance that would mean it uses already entered ratings and review to create a list of recommendations like the ones the user likes by then looking at other users in the system and finding what other movies they also liked. Most of these systems also try to group users together who have similar tastes recommending only movies from users that have like a lot of the same stuff as them, rather than just one or two. This ensures that the user has a heightened chance of enjoying the movie as opposed to if it just found a random user who also rate the movie high as that user may have a broader preference in movies. Collaborative filtering is used by a lot of modern-day streaming platforms like Amazon, YouTube, Netflix and many more, these intelligent systems have a large user-base for which to use this type of recommendation system, meaning more accurate but complex results.

(Abhinav Ajitsaria, 2021)

User-Based filtering

	Film 1	Film 2	Film 3	Film 4	Film 5	Euc. Dis.
User1	4.5	5	2	3	1	-
User2	3	2.5	5	3.5	1	4.21
User3	4	4	3	1	1.5	2.55

Tyler Brown
C21059883

User-based filtering is used for when a system has more users than items (films in this

User4	3	5	2	3	1	1.58
User5	1	4.5	4	5	4	5.43

instance.) The algorithm itself works in a resource intensive way, as it will need to compare all users to find similar uses. This is done by create a matrix of users versus movies with the rating each user made for each item like the follower, highlighted in blue is the current user the system is trying to recommend too:

The system then gets the Euclidean distance between each vertex in the table, so for User1 vs User2, User1 vs User3.... Etc. So in this example the user with the closest opinions and reviews would be user 4 (highlighted in orange), due to it having the lowest Euclidean distance to the original user. (Abhinav Ajitsaria, 2021)

The system would then use that users top rated movies to find a movie that the current user is yet to watch, and tries to predict if the user would rate it high, but find the topmost similar users and using the following formula that calculates the average rating of all similar users, which we can assume the users rating will be similar:

$$Predicted\ Rating = \left(\sum_{u=0}^n User\ u\ Rating \right) / n$$

Where u iterates through all that are similar, and n is the total number of users. (Abhinav Ajitsaria, 2021)

Item-Based filtering

Item-based fundamentally works in the same way as User-based but for when there are a lot. more items rather than films. In this instance item-based filter works by instead of calculating the distance between two users, it instead finds the distance between two films, but using all users' ratings it can find films that are rating it a similar way to a film the user already likes. The film highlighted in blue is the one the systems trying to find recommendations based on and orange being the closest film. In this instance the system things that because that all the users rating film 5 in a similar way to film 1, that it should suggest film 5 to people who have enjoyed film 1.

	Film 1	Film 2	Film 3	Film 4	Film 5
User1	4.5	5	2	3	4
User2	3	2.5	5	3.5	2
User3	4	4	3	1	4
User4	3	5	2	3	3.5
User5	1	4.5	4	5	2
Euc. Dis.	-	4.0	4.6	5.2	1.58

The problem with collaborative filtering is that due to the complexity of the calculations and the amount of data being accessed users may experience some latency when loading pages as they wait for the applications to calculate the recommendations, another problem is what they call cold-start, which is when an application is launched there is not enough data to accurately preform these calculations, the application does not have access to the history of users before the application was made.

Content-based Filtering

Content-based filtering differs from collaborative by instead of using other items in the set to find similar items, it uses the items attributes or the users' attributes to instead recommend to the user. It these circumstances, instead of the recommendations being based on previous ratings the user has enter, it is based on other entered information the user enters. This could be

Tyler Brown
C21059883

information about the user themselves or a calculation on the users account in general. So, the user could enter their favourite genre, actor, director, or other similar information; or the recommendations can be based on the user's person, so sex, age, hobbies, job, or other personal information. This could be calculated also, so the application could notice that the user most rated genre is horror, so it would recommend horror films, or that they tend to watch a of films with a specific actor or from a specific decade.

This compared to collaborative filter is very easy on the system due to its focus on only accessing the user needing the recommendation rather than comparing everyone in the database. It differs in that the user has a lot more control over the recommendation the system is suggesting, by changing information on their profile they can influence what kind of recommendation they get. So, for instance the user could start to dislike horror films, all they need to do to stop getting them recommended is to remove the genre from their favourite films. This differs from other methods as other methods use information that the user cannot change, like already enter ratings and review, which although can theoretically be changed the user should feel they need to change their opinion on previous entered ratings just to change their preferences.

The difference in computation power required allows the recommendation system to be more dynamic, changing with new releases, the user can get new recommendations for their preferences depending on what is new in the cinema in a specific category of films, this dynamic approach to recommendations allows the application to feel up to date new, and removes any old fashion feeling to it whilst also keeping old classics in rotation that the user is yet to watch. (Victor Dey 2021)

Using this method fixes the cold star problem due to the limit information applications have on startup is irrelevant as on account creation the user can enter their preferences and have recommendations lined up for them without even entering any ratings or review (Mark Milankovich, 2015.)

This method in recommending movies can often limit the user to discovery into new genres and gets blocked away from certain content to it simply being not in their preferences, regardless of whether they may like it or not. This bit of randomness is requiring sometimes due to the risk of repeating similar recommendations become stale or boring to the user. This is known as limited serendipity and is often a huge problem for content-based algorithms. By having a system that sometimes recommends an anomaly that is irrelevant to any of the users' preferences is important, so this level of randomness needs to be addressed when people use these kinds of systems. Another limitation could be the accuracy of the users' input, if a user is unsure on their preferences, they could feel that the recommended movies are inaccurate and find the system useless or dislike using it very much by placing the trust in the user to enter accurate information this must be considering.

Generative AI approach

Although this project is likely not to include this type of recommendation due to the time restraints and requirements of an AI-based recommendation system being too high to fit within the scale and scope of the project it is still worth mentioning due to the rising popularity of artificial intelligence with the development of models like ChatGPT. AI base recommendations will have the most accuracy due provided given enough example data, the AI may use its own method of recommendations or may use some hybrid form of both collaborative and content-

Tyler Brown
C21059883

based to return the recommendation. On the contrary, the system will have issue with it, the base requirement of large amounts of data aside, the system will also be susceptible too bias's depending on the information fed from it. The system will also have a large maintenance requirement and filtering system to ensure it does not error or go out of data. This can result in system not giving correct accurate recommendations or other issues, if not giving a large or specific enough stream of data. (Thomas Peam 2023)

Developing a language model however is not a substitute for one of these other systems, and although the AI can influence the suggestions, fundamentally it would be done in the same way as one of the other two approaches, except the language models, there is a lot more work and data required to train the model which is not suitable for such a small-scale academic project.

Which system is more suitable?

Out of the different systems researched above, it seems that the most usable and suitable algorithm to use for this project would be the content-based filtering, where user recommendations are based on the own users entered information rather than the information entered by other users and friends. This is because of the amount of data required for collaborative filtering and limitations on Generative AI recommendations. The scenarios will have limited data and will have the user enter information about their own preferences, so it stands that this be the only suitable option.

2.3 User Interface and User Experience

This section will detail different methods and strategies to ensure that the visual experience and feel of the site remains easy and appealing to the users. By learning strategies to organise information and increase accessibility on my site, the project can reach a larger audience and account for other demographics who would not have an enjoyable experience otherwise.

Information Organization

Due to the nature of the project this application will be dealing with a large amount of information, so it must be that the way this information is displayed and organized to the user be an essential part of the web pages are laid out. Portraying the webpage in an informative and intuitive way will help the user read and find the most important bits of information and leave avoid any unnecessary or irrelevant information that they do not need. In this instance the most important information for movies is a general bit of information about the movie, user reviews and ratings, in addition to an average rating score. We can also leave out unnecessary information such as production crews on the movie, except for the director, larger descriptions of the movie, studio names and funding. All this information seems unnecessary for this style of application.

In addition to this an important aspect is to ensure that the layout of information remains consistent throughout the entire website so not to confuse users, following the flow of information, displaying more important information higher up in the page, with a more visible styling to it (Sally Wolleat 2024). This allows the user to just glance at the site and take away the information they need while also providing the means to delve into finer details should they need without it being shoved in their face.

Tyler Brown

C21059883

Accessibility

To make a website accessible is to make it available and easy to use not only for most people but for people with disabilities that may usually prohibit them from using the internet. Some of the way we can account for accessibility is to ensure that the colour scheme we use is using contrasting colours for visually impaired people to easily see and distinguish between foreground objects and background objects, making the most important parts of the page easy to read, and hence easy to use for those who struggle. Like this it is also important not to portray information using colour, an example for this would be asking the user to select, the blue box, red box, or yellow box, this poses an issue for colour blind people, and it is much more accessible to use large bold numbers to label or group things that the user needs to select (Web Accessibility Initiative, 2015)

3 Design and Approach

This section will discuss the design methods I took and the approach to specific problems that helped plan and make necessary preparations for the different technologies and methods I will use when implementing my project.

3.1 Technology Stack

The different technologies used play a huge role in success of the project, picking the incorrect language or technology could result in making the project harder than it must be, so considering the circumstances of the project in addition to my own knowledge on each technology the following systems and languages are used. The design this project will be in will be a modular design where the front end and backend are separated which allows for easier maintenance without taking the entire site down, if for example the backend needed to be tweaked a little.

Front-end

The front-end of the application, will be a blend of different technologies. These technologies used are going to be specifically chosen to ensure that the project is done efficiently, and these technologies must have specific capabilities that allow for all the features and system that are planned for the project and because it is the front end must allow for a clean, professional looking design. React JS and Node JS will be the primary programming language that the front end uses, however due to the nature of React JS the front end will be using HTML as a subsidiary of react JS from within the JavaScript code itself. This is how the design of the website and scripting of the front end will be designed. The reasoning behind using React JS and Node JS is due to the nature of these technologies, being marginally quicker than most other front-end frameworks for this type of project, more precisely 14% and 19% faster than the next best technology respectfully, the next best technology being jQuery. (Shahil Modan, 2023) HTML being the only front-end markup language that is universally supported and pretty much the only one used online today. The dynamic design behind these technologies allows for a responsive and interactive design too the project and allows users to interact with elements its real time without page refreshes or loading times. In addition, React, the project will also use React Routing to allow for the user to accurately navigate through the site using the URL allowing for them to send links to friends of their profile or of a film they enjoyed, should they not want to interact through the site itself.

Tyler Brown
C21059883
[Backend](#)

The design of the backend is of the most important part of this project as it is responsible for most of the processing and data management, the technologies must have the capacity to process large amounts of data and effectively interact with the front end in a secure, responsive, and reliable way, allowing for an efficient interaction between the database also. The technology being used in this project is a complicated one due to the specific needs of it, the project will include a Rest API for a package using python, for specifically using Flask module within python to create an API that the front end can make requests too to retrieve data about the user or films that the user is trying to find. The importance of this technology being secure and quick has a heavy influence on the success of the application and how the users interact with it despite the users not directly interacting with the backend themselves. With in the python and flask itself we use another technology called CORS, this is a technology that uses cookies and GET requests for user authentication and to remember what user was logged in user a key, within the cookies of the user accessing them, meaning the user will stay logged in if they access the website from the same device and browser.

Included in the backend is the storage system, in this project SQL Database is used, for testing purposes only a file is used, but during deployment an SQL Server will also be deployed for safety and security purposes. But the use of SQL allows the data to be managed and store in an appropriate manner to the project, efficiently dealing with data requests the server has, organising data and returning it to the user in a useful way, without any of the sensitive data. SQL is one of the most well-known easier to use data management languages and since I have prior knowledge of it this is what I will be using.

Other Tools

Other technologies which will be used but are not as vital to the process of creating the project are as follows:

- Text Editor, VSCode
The importance of a good text editor can make the process of development either a lot easier or a lot worse. VSCode has been a personal favourite, due to its intuitiveness and effectiveness when it comes to shortcuts and fundamental systems that allow the user to develop features with ease, and with the extensions systems, you can add anything to the editor.
- API Interaction, postman.
To test my backend thoroughly and to interact with a back end that is yet to have to a front end we need some way of making requests to the rest API and see how it responds to then build a front end around the response. So, this project will be using postman to do this. Postman allows us to receive and make requests to the API without the fuss of coding anything extra just to test it.
- Version Control, Git.
Version control is typically used when multiple developers are working on a project at a time, however I used GitHub, with git in order to back up and document my project, so that I could occasionally work on it from multiple devices at different places without the need of carrying the physical storage medium with me, this will also help me protect the project in the future should I need it later on to reference or look back on.
- Encryption/Hashing, BCrypt

Tyler Brown
C21059883

Due to the nature of the backend and frontend only communicating through requests it is important to make sure that all sensitive data is encrypted, this is done in python where all data is encrypted before transmitting. Additionally, this technology is used within python to hash any passwords before they are stored to ensure the integrity of user data.

- Figma.
Used to create data flow and prototyping images for the design of the project, a proper design which is important to the success of development.
- DB Browser.
Used to simply view the database system in an easily readable format for testing and development purposes.

3.2 Gathering Data

The sheer amount of data required to make a movie system this large is almost impossible, there are resources everywhere of datasets online that give you access to this kind of data, as it is impossible to collect it yourself as an individual, however a more efficient way is to use a technology that already has access to this kind of data, an API whose entire job is to be a large database of movie and celebrity and use their information to create my own application.

The Movie Database has an API that I can access that will give me all sorts of information relating to each movie, including production crew, cast, general information, genres and even provides images to cover art and images of the actors themselves in the movies, this kind of database is perfect for this project as all the information required is one API request away and can easily be referenced in the applications database using their own unique ID. Doing movie data and statistics this way saves on storage and a lot of effort collecting the data, however it does mean that the system is reliant on another system being functional.

When referring to user data, any data collected on the data will be data manually entered by them themselves, rather than collected through other more invasive means, only the most essential data is collected on users, such as usernames, email addresses, passwords, and other review-based information the user intentionally places into the system themselves. Any sensitive data collected is treated carefully and hashed. Data is entered through specific forms throughout the website like the registration form or review creation form where data is censored when it is personal or private like passwords.

Tyler Brown
C21059883

3.3 Database Design

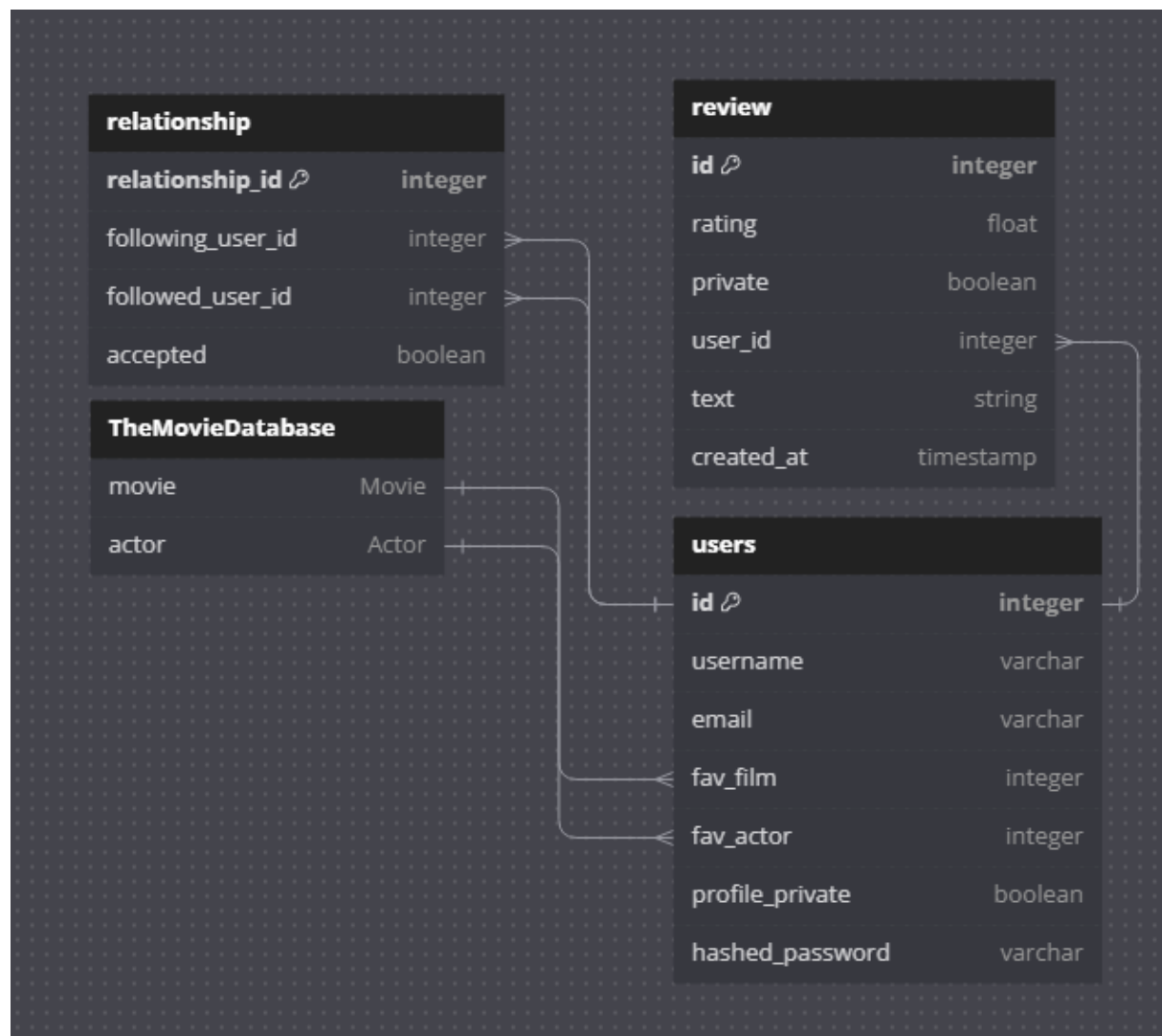


Figure 4 - Relational database diagram.

When designing the information technology behind a system, databases are the thing you need to design so carefully and meticulously that you use as little storage in the most usable way for your application. When designing it is a priority to decide what database system you want to use, and what management system you want to you, the design of both is dependent on the type and amount of data you wish to store, as well as the knowledge of the developer in each field. In this project the database system that will be used is a relational database as we want to ensure that every review stores have its own link to a user and any follow request has its own individual follower. In terms of management system, the project will use is a little tricky since we are making the backend through python, however fundamentally the management system will be MySQL just through a separate module that integrates it into the flask application called Flask-Alchemy. This allows for easy interactions between the relational database and the application itself, as instead of using SQL Queries, we instead use python queries that represent the SQL queries saving time and resources when using it due to it being more efficient to run instead of implementing my own system to run SQL Queries.

Tyler Brown
C21059883

In this design the simpler it was made effectively the more efficient it becomes, by overcomplicating it by adding countless tables and relationship increase the amount of work later, so by confining the system to 3 tables and 1 external database as mentioned above, we keep it easy to read and easy to use. The user's table has embedded into it all the information about a user's profile as well as the user themselves, this is due to the only thing needed for a user's profile being, their favourite actor and film, a list of reviews they have left which has its own table, and a user's username. The favourite actor although does not necessarily have a relationship with TheMovieDatabase, it stores the ID of the actors and films from that database, essentially making it the same thing. That also if their profile is not private. The review table itself links back to the user table to refer to the creator of the review, this table only has a rating, text for the review, the author, privacy values and a timestamp of creation. The following system has its own table, which stores the follower, the person they are following, and whether it has been accepted. To remove any redundant data within this database no information on the movies or actors are stored, instead any time this information is needed, the system makes an API request instead. With this design of database, the application should be scalable to a larger user base, the limitation on data redundance and organization of user information allows for more user traffic and interactions with the database, this also helps that we are using a REST API instead so only one application is interacting with the database.

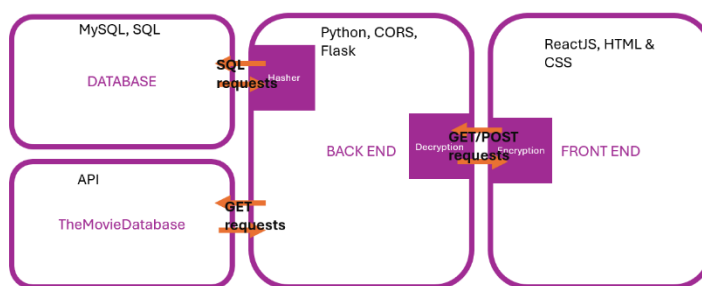


Figure 5 - Representation of dataflow and modular design.

The flow of data has been designed in a way to ensure that any sensitive data is protected, due to the backend front end system being separated, all sensitive data needs to be properly handled, the data flows from the front end into the back end, and vice versus using GET or POST Requests. The backend then will

make a request to the movie database when information is required. When data is stored or retrieved about users or reviews, a SQL request is made to the database. This modular design allows for easy and efficient data flow.

Tyler Brown
C21059883

3.4 UI/UX Design

User Flow

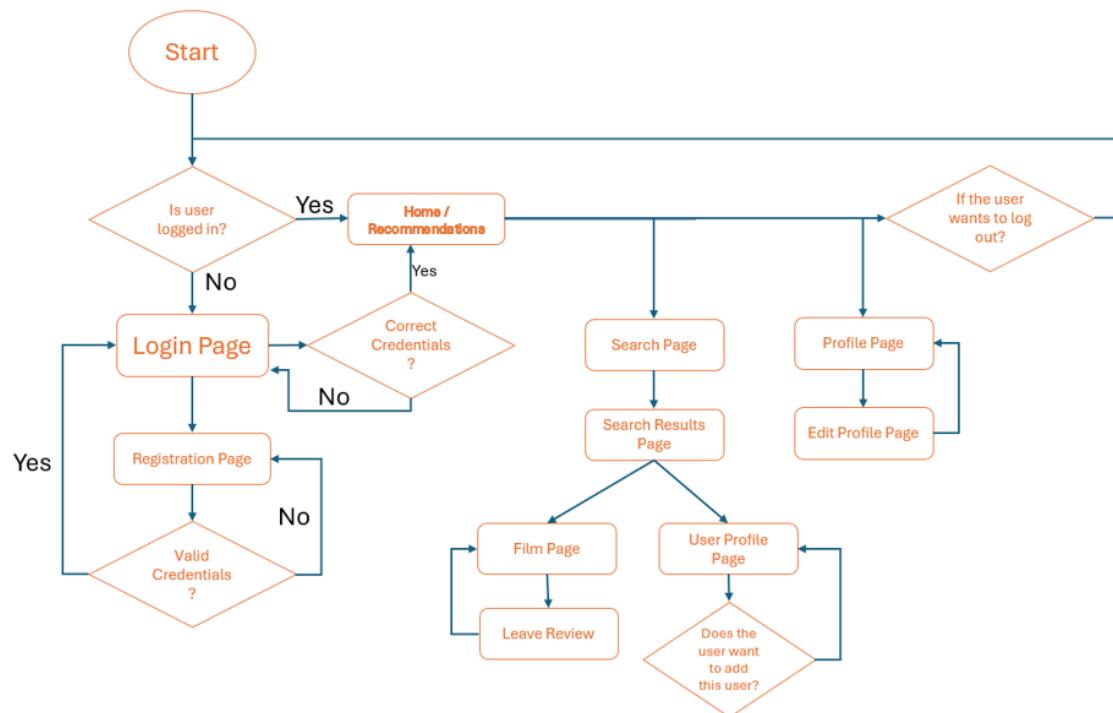


Figure 6 - data flow diagram and user flow.

When creating the flow of user traffic in a site with such complexity, it is quite important to ensure that users do not get stuck in a loop of any sorts and that the flow of the user is very intuitive and easy to anticipate where the user will end up, when flowing through the site the user should end up on the correct pages and all related pages are linked to the parent page that is relevant, for example users should not be able to select a film from the login menu as this simply makes no sense. Navigation bar links connection is not represented in the flow diagram due to the navigation bar always being visible.

Feedback loops that are necessary in this system are used for login purposes and for when the user needs to iterate through a large amount of films or users they want to add or review, for example if the user has just created an account the chances are they have quantities of films they need to search for, so instead of return to the homepage, after users have searched for a film, they are returned to the search bar page. The login loop is there purely for credential purposes, users are only stuck in a loop if the credentials entered are incorrect, where they then have the option to register anyway.

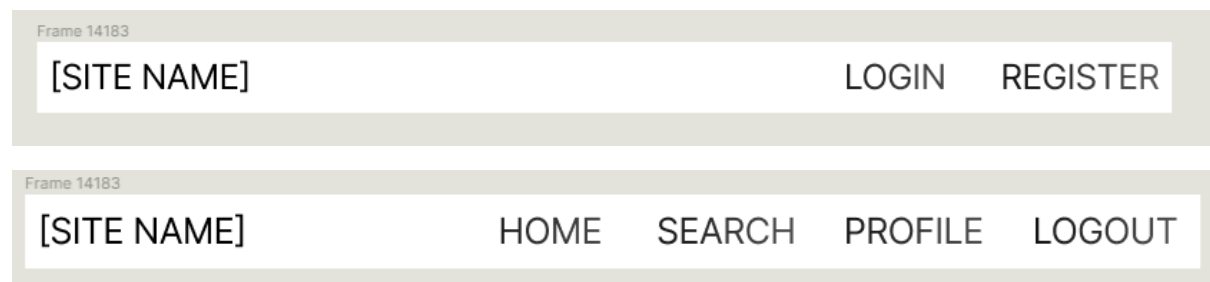
Navigation

When deciding the style of navigation you want, a factor of high importance is the intrusiveness of the navigation bar vs the importance of easy navigation within the site itself, for much simpler system a navigation bar sometimes just is not needed. However, for this application it is essential that users have the option to always traverse the site, no matter where they are. So, the thing left to decide is the style of nav bar, whether to have a less intrusive drop-down

Tyler Brown
C21059883

navigation system or to only include a more permanent top bar navigation system. For this application the number of options in the bar is few, so having top bar navigation is not as intrusive as if we were to have more, a navigation bar with few links makes no sense to have it as a drop-down system.

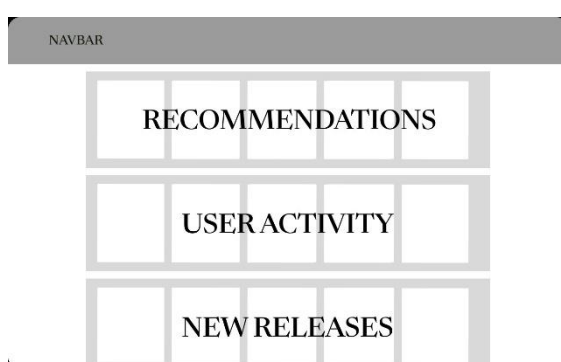
The contents of the nav bar need to be simple and easy to understand, having few links contributes to this. The links users have in the navigation bar should be the most traversed areas of the site, so designing it should be easy. Users will mostly use links, home, search, and profile. These three links will always be accessible to the user, as these are the three main areas of the site, however some links should be accessible in certain circumstances. These links include logout links if the user is logged in, and a login/register button for when the user is not. These two links allow the user to navigate from logging in to registering easily.



Another design choice that could be implemented is when clicking the name of the site that they be rerouted to the homepage of the site, this is useful however to keep the navigation of intuitive, the HOME button on the nav bar will remain also.

Wireframes / Prototyping UI

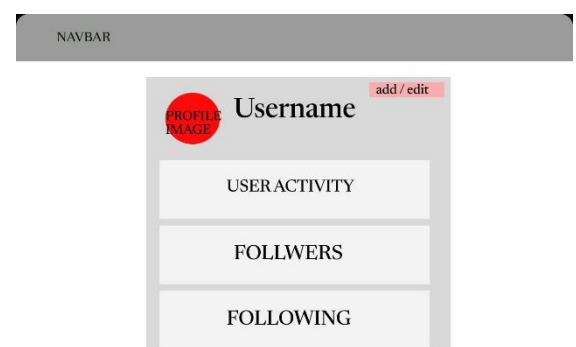
Before beginning development, it is a good idea to develop some wireframes and prototype of the site you plan to create so that when you begin development you have some sort of frame of reference to begin from, preventing you from being lost and allowing you to focus more on the fundamentals of the site, and functionality rather than cosmetic problems.



For the home page, I thought it was important for this to be the place where the recommendations are stores, in addition to this, the home page will also be the place where the “feed” is showing users activity and brand-new releases in the film industry. This is the place to go if users want to find something to watch. Although the new releases and user activity are low priority feature. Each section will be separated, to easily

distinguish between them.

The user profile will be how user distinguish and interact with each other, there will be 4 sections to the user profile, all of which will be clickable to take you further into what each section will entail, one section for information on the user, one for activity, one for followers, and one for following. Also there will be a



Tyler Brown

C21059883

button, this button will change depending on the circumstances, Edit if the user is on their own profile, Add if the user has not added them, and remove if the user is already following them.

When designing the search system, the decisions been made to instead of having a sperate page for the results that the page be a dynamic page that updates the current page with the results, this however will leave whitespace before the user search but will reduce loading times and page refreshes. Also, combining both types of searches into one bar reduces page redundancy and ease of use, both users and films can be searched for in one menu without the need of redirecting the user

between a different search system.

The creation of the UI of a login page is very simple, only 3 things are required, username, password and login button, passwords will be censored as is the norm when developing password systems. The user will not need their email address to login, this is a design choice, users are distinguished by their username not their email and their email is used just for protection against account loss.

The registration system is also very similar to design, except the user needs to enter additional information to confirm and initiate their account. Email will need to be entered for security reasons, and the password of the user will need to be confirmed to ensure that the correct one is entered, and a mistake is not made, preventing account loss.

User Persona's and Profiles

With socially oriented applications the need for user distinguishing is ever important. The users need to have enough customization for other users to decide whether they want to o follow the user or not in a system where you prioritized user security this can sometimes be difficult. To give users, the ability to discover each other, the design of user profiles will in turn have systems in place to distinguished each other. Other than just the films they review, user should be able to select their favourite film or actor, this will further customize their profile. The profile icon of users will be determined by their favourite actor and will be the most recent portrait of them available to the system. The profile's background will be the backdrop of their favourite film and

Tyler Brown

C21059883

the user will have a section on their profile that tells the reader a little bit about their favourite film.

Since the target demographic would be a lot focussed on younger people who enjoy movies this customization and favourites feature should be very popular, the more customizations the user has the better the users perceive their profiles as being pleasant to look at as they can make it look exactly the way they like it. Another feature that could be implemented is the user's favourite genre or director that would be displayed on their profile although this is not a priority.

3.5 Security and Authentication

User authentication is an often, difficult process to implement into systems, this however, does not take away from its importance. The need to ensure that the user is who they say they are effects not only the user experience of the system but also data security and integrity. That is why in this system the CORS (Cross-origin resource sharing) mechanism will be used. CORS is described as “an HTTP-header based mechanism that allows a server to indicate any origins other than its own from which a browser should permit loading resources.” (Mozilla 2021.) This means that the user login system will be handles by both frontend and backend, where the user on-login will receive a cookie in return which will in turn serve as a authentication key for the browser, ensuring that the user can stay logged in not only as they use the website but also if they try open the site on the same browser again in the future.

User authentication and data integrity also requires data validation, to ensure the user is authentic and is not someone else, the data the user inputs into the system will need to meet specific requirements. In this instance, data validation will be run on any email address to ensure it is in fact an email, usernames to ensure that the user does not already exist, passwords to make sure the pass is secure and safe to use and other minute details where data validation is run to ensure the data simply exists. These are called format check, uniqueness checks, format check again and validation check. Also, another consistency check is performed on the registration process to ensure that the password and password confirmation, are both identical.

Most of the security aspects of the application are managed by a system and python module named BCrypt, where BCrypt will ensure that any receiving data is encrypted when sent through a post request. This encrypt adds a level of security, due to the separation and modularity of the front and back end, this link is susceptible to attacks and leaks, encryption adds a complexity to those trying to maliciously interact with this data. Additionally, the importance of ensure the databases are not storing passwords comes into view when you consider data leaks and cyber-attacks on the data itself. This calls for a hashing system which again is handles by BCrypt, where passwords are irreversibly encrypted and instead of decrypting to check against the password entered, the password is again hashed and compared to the already hashed password in the system, this method is safe as it is mathematically compared instead of a simple consistency check as both these hashed password will not be identical even though the password entered to get these hashed values back where.

Tyler Brown
C21059883

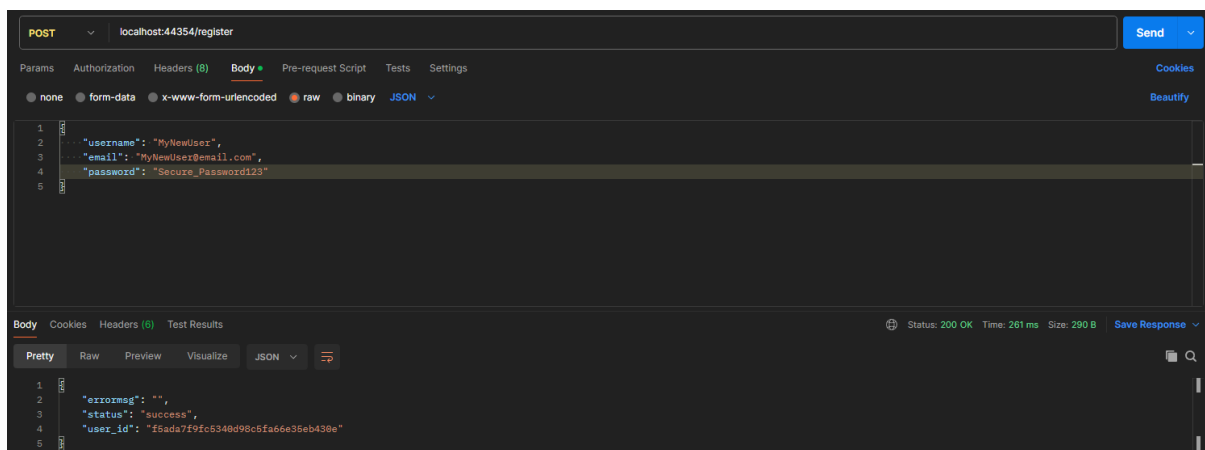
4 Implementation

This section includes an in depth discuss on the development and process of implementing my designs and solutions to the project itself, in the form of screenshots to demonstrate and a discuss on the decisions made in each section of the application. Two types of screenshots exist in this section, one which looks quite code heave but however is just an API response from my backend, and a screenshot of an actual webpage to show the final product after the implementation of both front and backends.

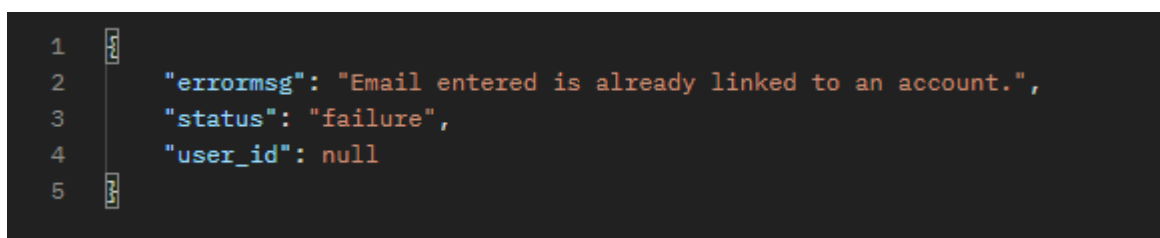
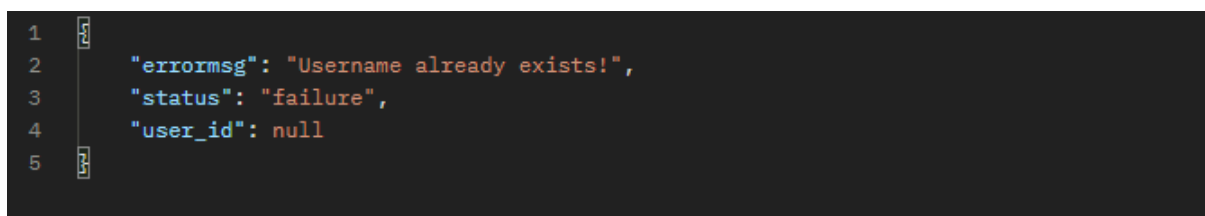
4.1 Login and Registration

Looking back on figure 6 this represents the first section of the flow, where the user is looped until valid credentials are made or until the user creates an account and then logs in, also includes the logout section in the top right corner of figure 6 too.

POST Register



Here shows the registration system in the backend, the user inputs the username, email and password into a post request to the “register” end point, where the system then does its necessary checks, where if it is successful the backend will return a status success message including the user id, this allows the front end to know that an account has been made and to continue to the login page. Password confirmation consistency checks and format checks are done through the front end.

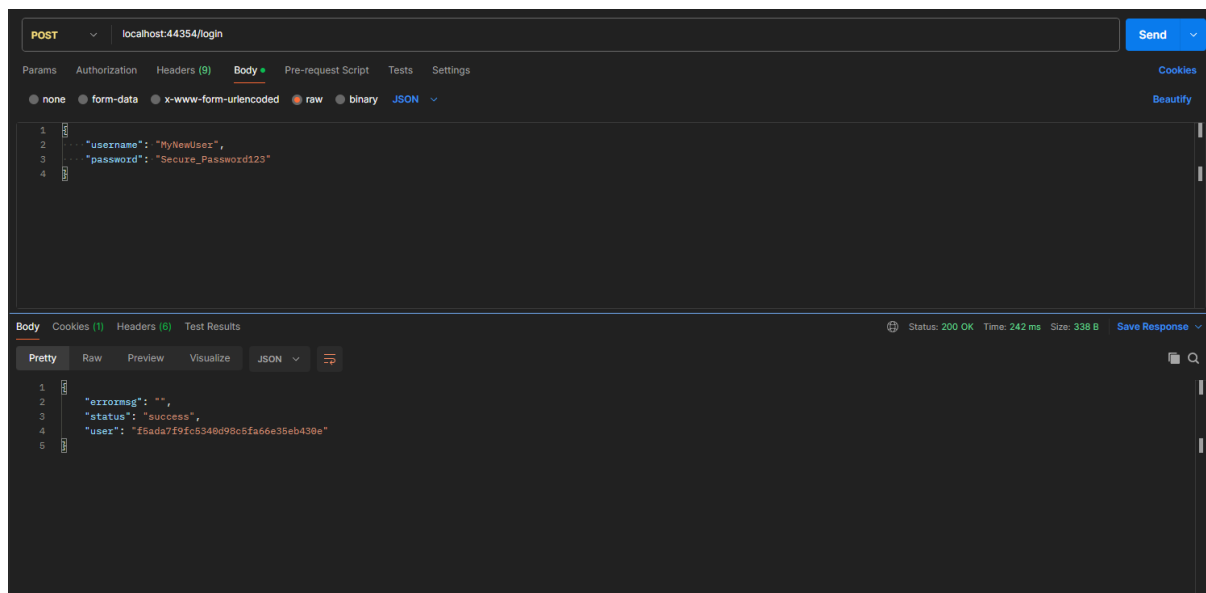


Tyler Brown

C21059883

If the system finds another user with the same username an account cannot be created, so the user is sending a failure status message with an error message telling the user that they need to choose a different name, this will be used by the front end to show the user where account create failed. If an email address is already linked to an account, it is impossible to use the email address for user validation, due to this the system does not let two accounts exist with the same email, so the system will return another failure message, with an error message saying that the email is already in use. Both error messages return a null value instead of a user id, this is for consistency in the return messages, because of this the attribute is still sent out.

POST Login and Cookies



In this demonstration the user login system from the backend is shown, the user inputs the username and password as attributes in a post request to the login endpoint, where the user if login should be successful, is return with two sets of data, the first set of data being the API response, this being a success status and the user id of the user logged in. The second set of data return is shown below, and is in the form of a cookie, this cookie is a CORS implementation and allows the backend to know exactly what session the user is coming from keeping the user logged in and adding a level of authentication to the system, this feature is completely hidden from the user, and posses no complication for the user themselves.

Name	Value	Domain	Path	Expires	HttpOnly	Secure
session	TYIT7ayR-Nit...	localhost	/	Session	true	false

```
1
2 "errorMsg": "User doesnt exist.",
3 "status": "failure",
4 "user": null
5
```

Tyler Brown
C21059883

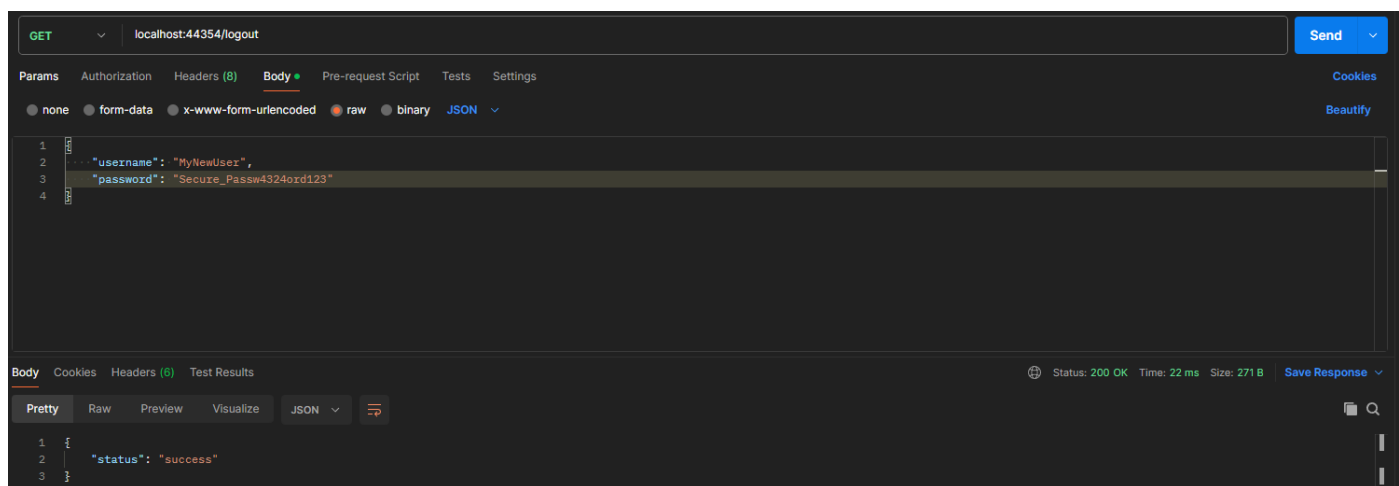
```

1  {
2    "errorMsg": "Password was incorrect.",
3    "status": "failure",
4    "user": null
5  }

```

If for some reason the user is trying to log into an account that simply does not exist or happens to enter the incorrect username, the user is prompted with a failure message, with an error message telling them this. Also, if the user does not enter the correct password the system also declines them entry into the account and prompts them with a different error message and failure status telling them that it was not right.

GET Logout



The logout system is a simple one as there is no data validation or alternative scenarios that the system needs to consider, it is simply a way to reset the session, which is done regardless of whether the user is logged in or not. So, by making a post request to the logout end point the user is always return with a success message, however if there is an active cookie session in the user's browser, the backend will make sure that it no longer views this authentication as valid, effectively resetting the entire user session. In this image the username and password are given, however this serves no purpose and is simply a result of attributes being left over in postman.

The screenshot shows a web application interface with a dark blue header. On the left, the text 'SCREENSCORE' is visible. On the right, there are links for 'LOGIN' and 'REGISTER'. The main content area is dark blue and contains a centered, light beige registration form titled 'Register'. The form has four input fields: 'Username' (containing 'MyNewUser'), 'Email' (containing 'MyNewUser@MyNewUser.com'), 'Password' (masked with asterisks), and 'Password Confirmation' (also masked with asterisks). A red 'Register' button is at the bottom of the form.

The registration page has been slightly changed to original planned to make room for data validation messages and to fit the overall theme of the site. The registration panel, like most panels has been centred on the page to make the site look symmetrical and aesthetically pleasing to users. The user has 4 option boxes, as planned, username, email, and two passwords field for conformation and like discuss the password fields are censored.

This screenshot shows the registration form with a red error message at the top: 'Username already exists!'. The 'Username' field contains 'tylerbrown101'. The 'Email' field contains 'MyNewUser@MyNewUser.com'. The 'Password' and 'Password Confirmation' fields are empty. The red 'Register' button is at the bottom.

This screenshot shows the registration form with a red error message at the top: 'Passwords do not match.'. The 'Username' field contains 'tylerbrown1012'. The 'Email' field contains 'tylerbrown101@gmail.com'. The 'Password' and 'Password Confirmation' fields are empty. The red 'Register' button is at the bottom.

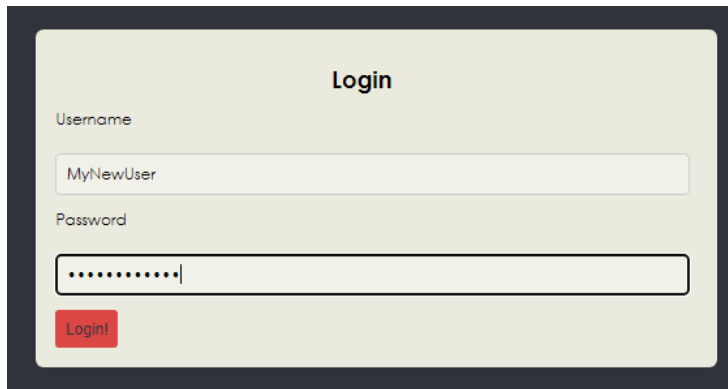
This screenshot shows the registration form with a red error message at the top: 'Password should be 8 or more characters.'. The 'Username' field contains 'tylerbrown1012'. The 'Email' field contains 'tylerbrown101@gmail.com'. The 'Password' and 'Password Confirmation' fields are empty. The red 'Register' button is at the bottom.

This screenshot shows the registration form with a red error message at the top: 'Email entered is already linked to an account.'. The 'Username' field contains 'tylerbrown1012'. The 'Email' field contains 'tylerbrown101@gmail.com'. The 'Password' and 'Password Confirmation' fields are empty. The red 'Register' button is at the bottom.

Tyler Brown
C21059883

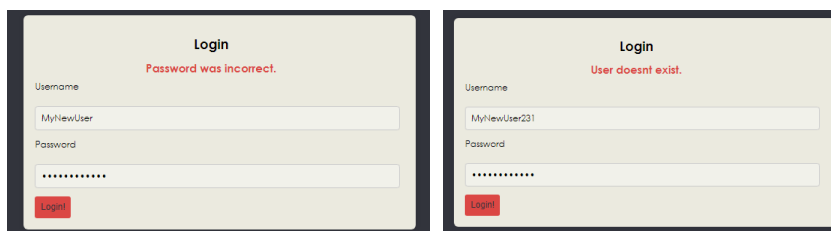
These screenshots show the data validation techniques in the regiser system, each case has its own message, to inform the user of how they need to change their details to successfully create an account. The message is in red to show that it is an issue with their information and appears above all of the fields to ensure the user reads it and does not glimpse over it and miss it. These 4 validations are for when the username is already being used, password are not the same, password does not meet the security standards, and if the email you used has already been linked to an account

Login



The login form is titled "Login". It has two input fields: "Username" and "Password". The "Username" field contains the text "MyNewUser". The "Password" field contains a masked password represented by eight dots ".....". Below the password field is a red button labeled "Login!".

The login page also needs a lot of validation due to it being the front of whether the user can enter the site or not, so it is important to ensure the user should have access to the site, this is shown in the design where the user is requiring entering a username and password field to gain access to their account. As seen above the field the user is currently typing in gets highlighting with a bolder border so the user can easily determine where they are typing.



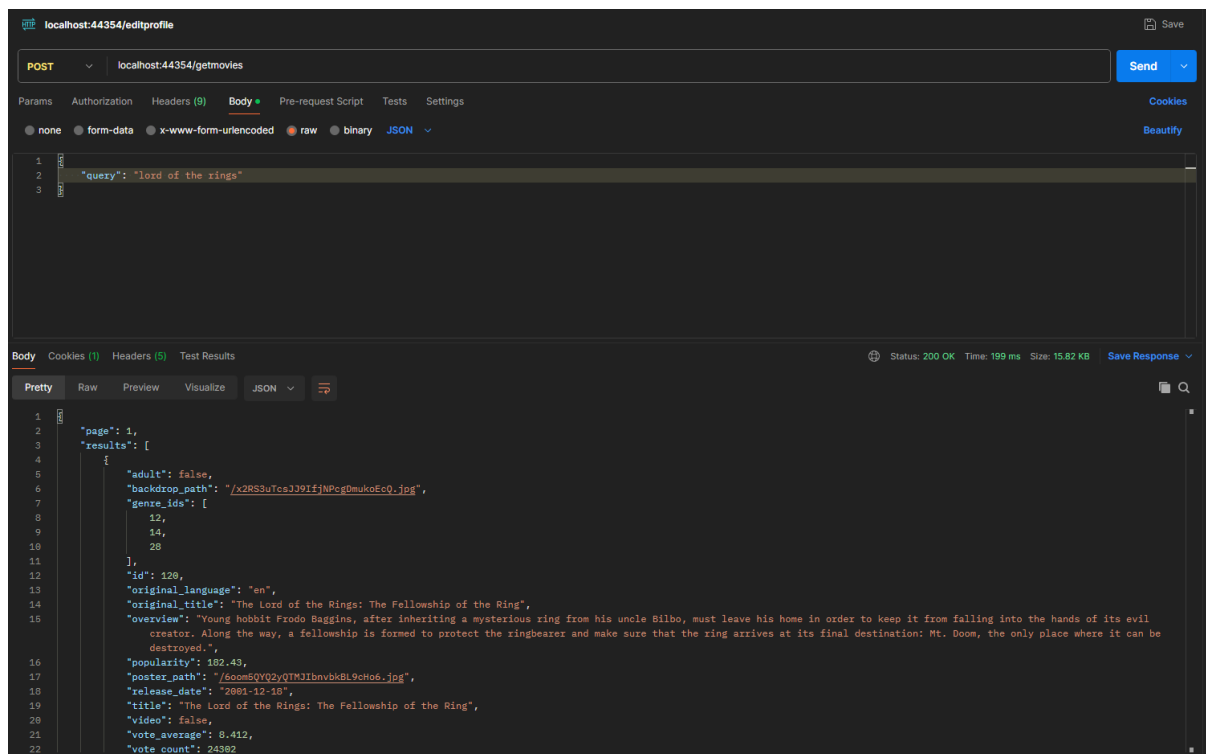
Two screenshots of the login form showing validation messages. The left screenshot shows the message "Password was incorrect." in red text above the password field. The right screenshot shows the message "User doesnt exist." in red text above the username field. Both screenshots show the login form with the "Login!" button.

The validation methods in this are to ensure that the details are correct, these validations make direct comparisons with the database to find a user with that username, and if they exist whether the password matches. The two messages the user can get returned.

4.2 Retrieving data

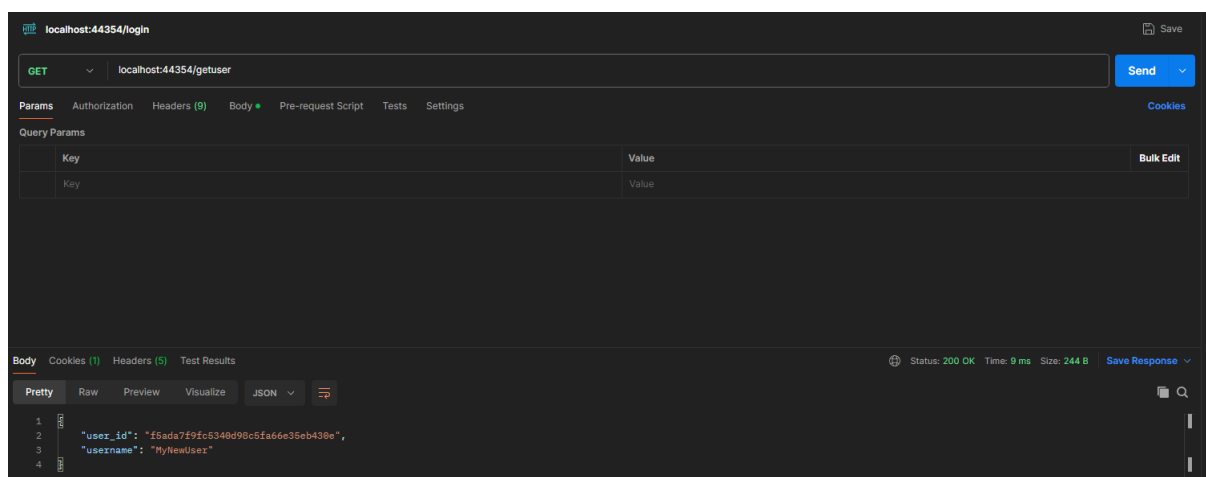
This system mostly applies to the middle portion of figure 6, where all the pages are dedicated to searching through data and providing the user with information about movies or users and user reviews.

Tyler Brown
C21059883
POST getmovies



This feature is used when the user wants to retrieve the information and results on a search of all movies, this system requires a POST request with a query of the film the user wants to search for, in this instance, lord of the rings is searched, and a plethora of results are given, the user is returned with a list of movies, each movie containing vital information about each of them, the most importance of which being a backdrop path file and original titles of the films, in addition to the post path too. Which all will be used for display purposes in the front end.

GET getuser



This API request is used for validation for the front end, it is used to ensure and get basic information about the current logged in user, this although simple will be the most used end

Tyler Brown

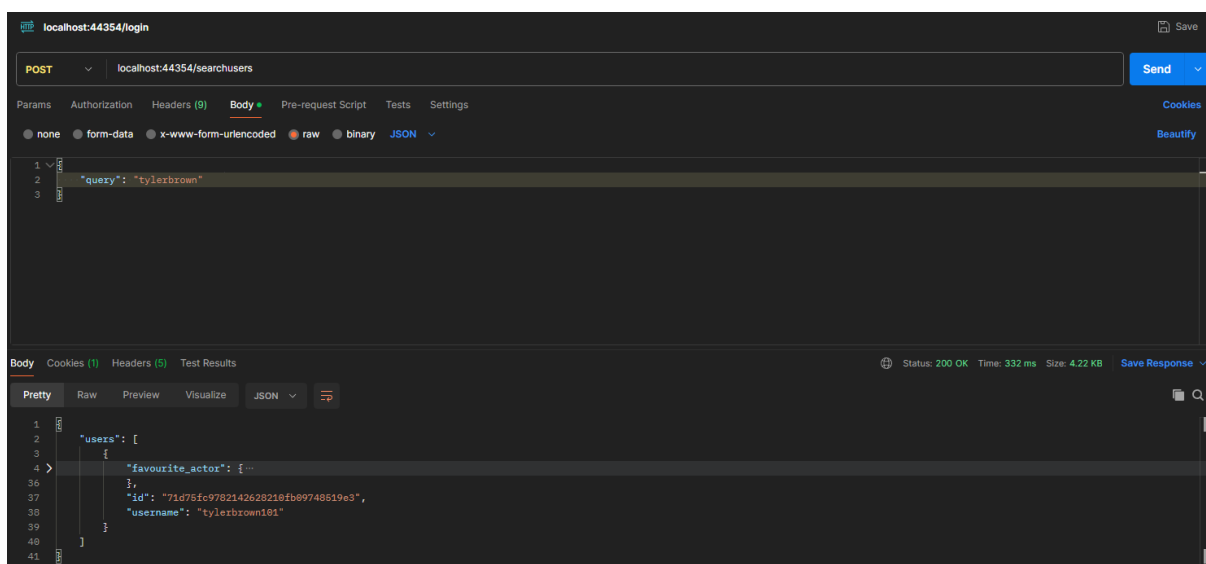
C21059883

point due to every page needing to ensure the user is still logged in. By making a get request the user will get returned with the user id and username of the current logged in user.

```
1 {
2   "status": "failure",
3   "user_id": null,
4   "username": null
5 }
```

However, if the user is not logged in, they are returned with a status failure message, and null values of the user id and username, this can then be used by the front end to return the user to the login page, and out of the main content of the website.

POST searchusers



The search features tend to be the most complicated API requests, this is due to not only the amount of processing required by the backend but by the fact that the backend also needs to access an external API, this gives the system a reliability on this API and results in this endpoint having the highest latency and performance issues other than the recommendation one. The user is asked to input a query that is then compared to the database of usernames, the user then is returned with a list of users, their favourite actors so the user can be shown the profile pictures, the id of the user and the username for display purposes.

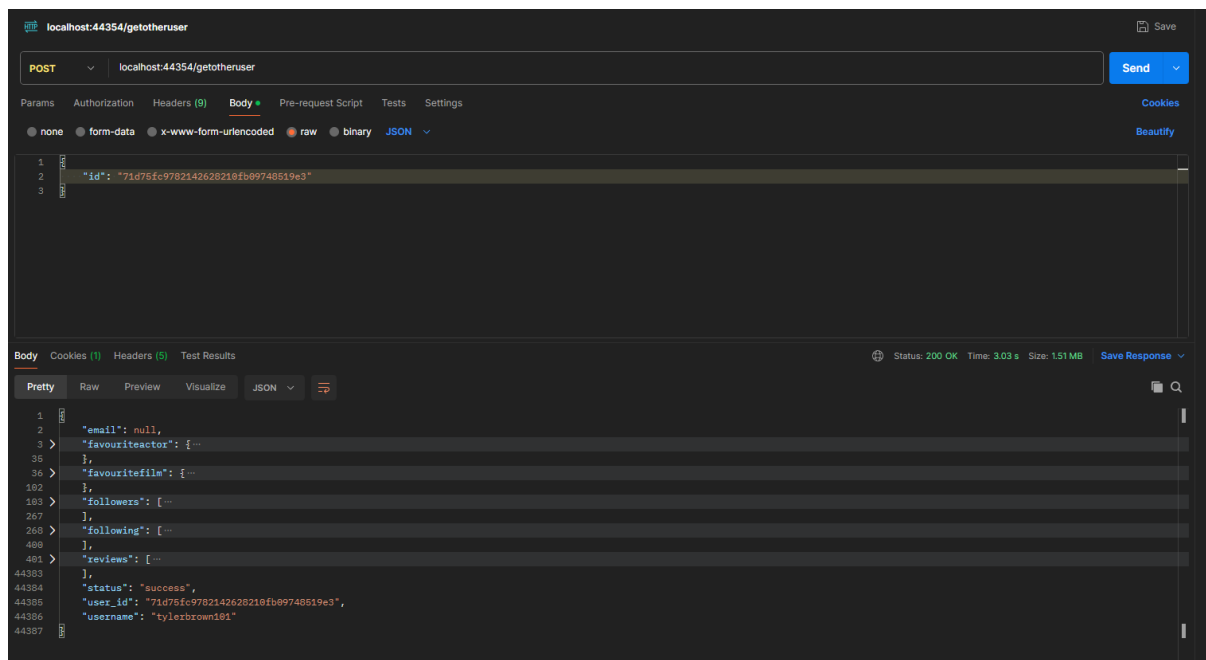
```
1 {
2   "users": []
3 }
```

If the user should enter a query that has no results the user is returned with an empty list, although no error message is returned, this is because technically there was not an error, but simply the results of the search are empty, and this will be used appropriately to display the lack of search results.

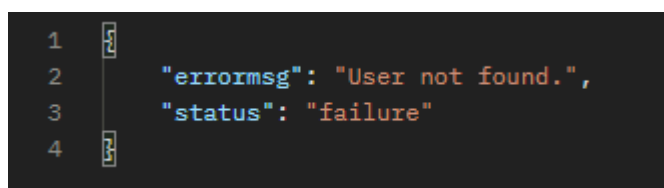
Tyler Brown

C21059883

POST getotheruser



Get other user, endpoint differs from get user by the amount of censorship and privacy settings involved, for example, if you are set on private, you still should be able to view your own profile, the solution to this in my eyes was to instead of producing a check on the current user, but to instead implement a different system to get your own user versus other users. This other user's endpoint is used by adding an attribute that has the users id that you want the information on, the user is then responded with a plethora of information that is used to create the user profile, this information includes an email although this is null due to privacy reasons, the favourite actor and film, a list of followers and followings, the user id and username, and finally the status that shows the front end that the request was successful.

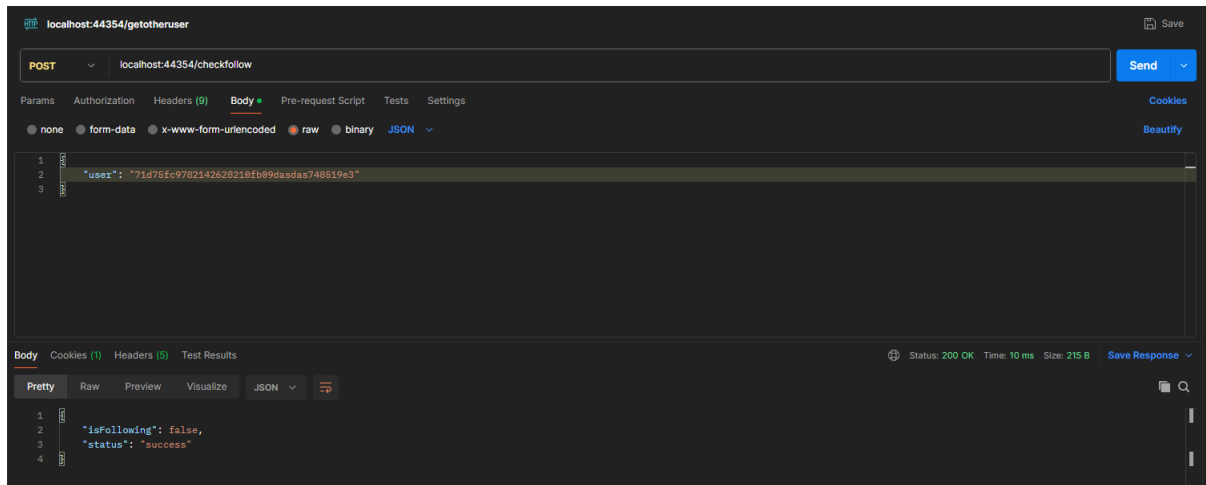


This error message of user not found appears for multitude of reasons, the first being as simple as the user does not exist, although this only occurs if the user manually changes the URL of the profile, they are on since the user will only access other users through search tools or other users' profiles. Another reason for this message is if the user they are trying to search for has their profile set to private.

Tyler Brown

C21059883

POST checkfollow

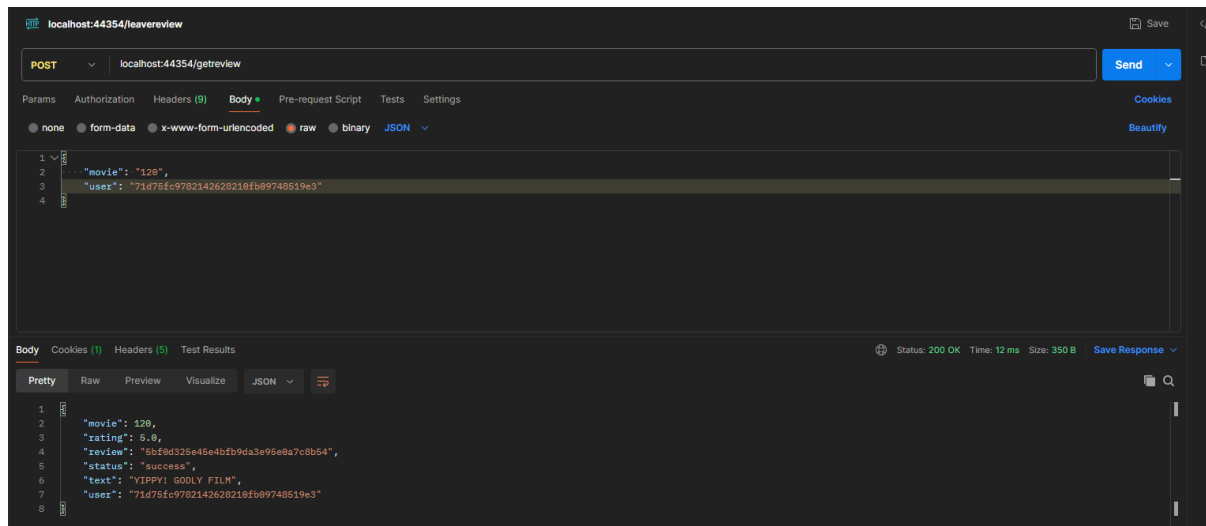


Tyler Brown

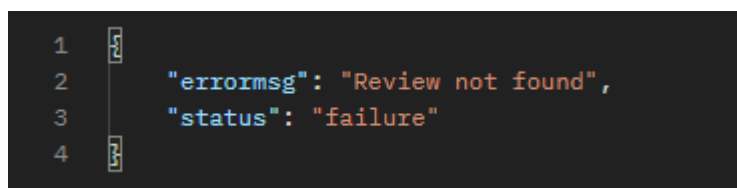
C21059883

generate movie information, genres, studios, images, and budgeting information. If the user enters a invalid movie ID which again will only be possible by changing the URL, the user will simple be responded with a status failure message, which will allow for a redirection to the home page.

POST getreview

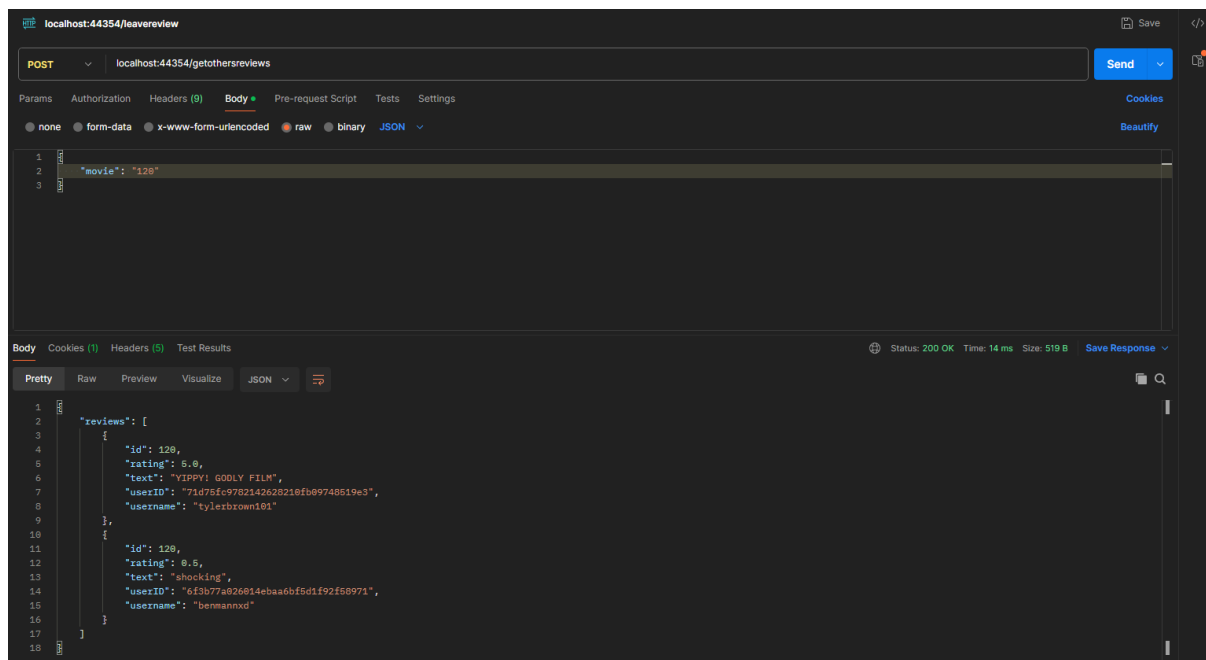


The review system in the backend has a endpoint that allows the user to retrieve a specific review, the requirements for retrieving this review is the user id of the user that posted the review and the movie id of the movie, since the user can only have one review per movie, this will show the specific one they are searching for, they are then provided with the move id to verify the movie the review is on as well as the user. Then the review content like the text field, rating and review ID for the front end to display to the user, a status message is also provided.

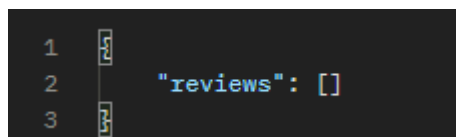


If the user is not placed a review on that movie or the user is not found or even if the movie does not exist the user is given a failure message that tells them the review could not be located and backend has to record of it, although this would not happen much is safe to say it is still implemented for the extreme circumstances.

Tyler Brown
C21059883
POST getothersreview

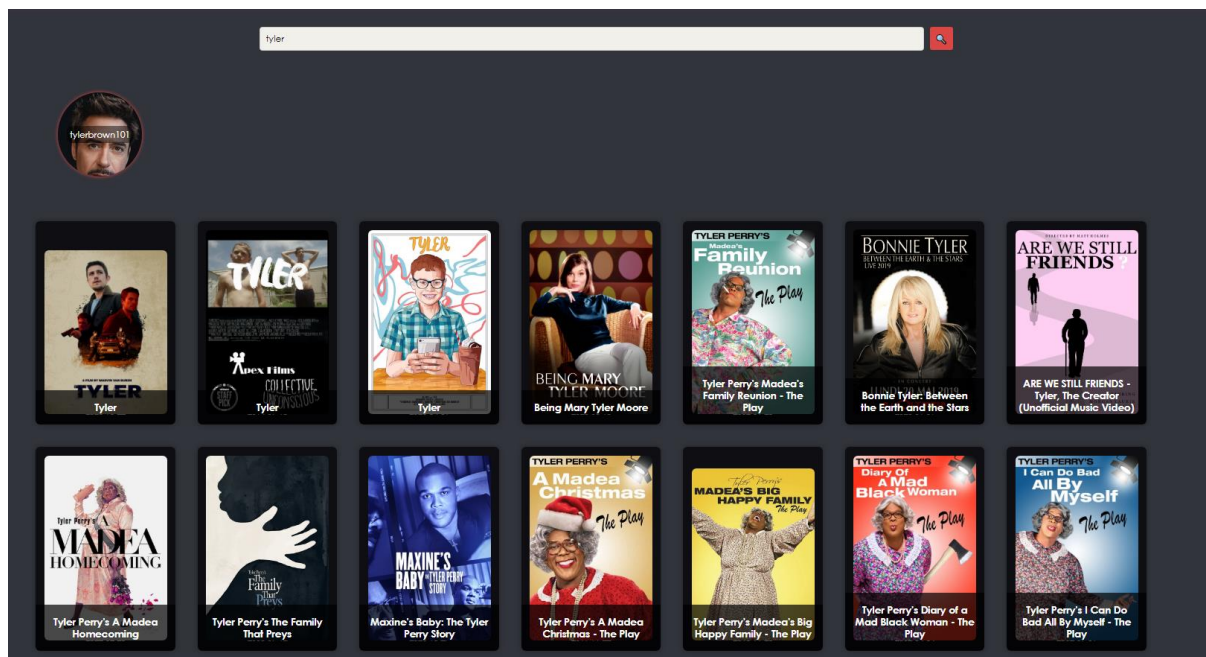


This endpoint is like the previous one except it is used for retrieving any information in regard to reviews about a specific movie where only the id of the movie is required. The user is then given a list of reviews on that movie that are public, each review having the username, user id, text, rating, and movie id, although redundant it is useful for debugging to display the movie id.



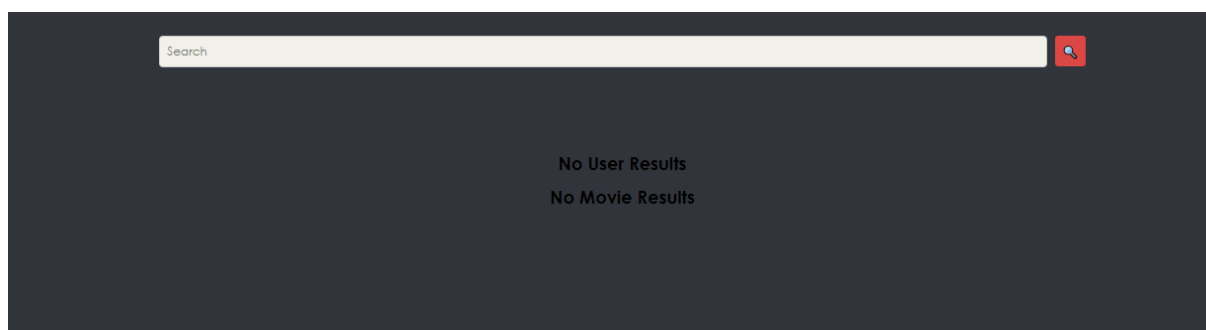
If not review is found on that movie, then an empty list is returned to the user, instead of an error message as nothing errored as per say but only the results are empty, the use can use this to display a message regarding the fact that no reviews exist.

Tyler Brown
C21059883
Search



The search function is probably the most complicated page since, not only are there two different searches going on, but the API request takes while also. The user enters a query into the search bar at the top and retrieves users matching the name, but also movies that match it also, in this instance I searched my name and retrieve my own account as well as plenty of movies that had my name in the title. By giving the two different types of responses different shapes, users having a rounded shape and films having rectangular shapes, I am ensuring users are seeing that one is a user and other is not, this makes sure they do not try click on for example my profile that shares a name with the film they are trying to look for.

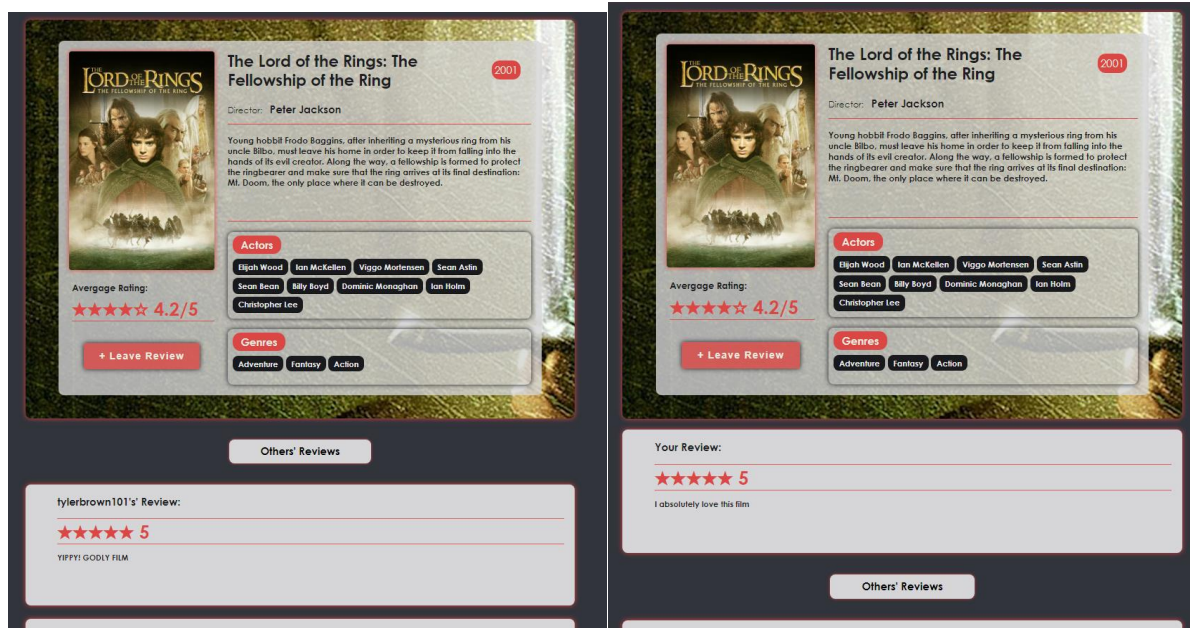
In this menu after clicking the search button you remain on the same page, this dynamic design of the page updating with the results rather than, redirecting to a page with results, gives the site a less clunky more refined feel to users, with less time between pages the user feels that they are using the site effectively by it loading responsively.



The one alternative case is when no results are found. Although this could have been made to look better, a simple message is show either telling the user no users were found or no movies where, or both in the event of nothing being found in the database. Either one of these messages can appear, or they can appear at the same time where they are placed on top of each other.

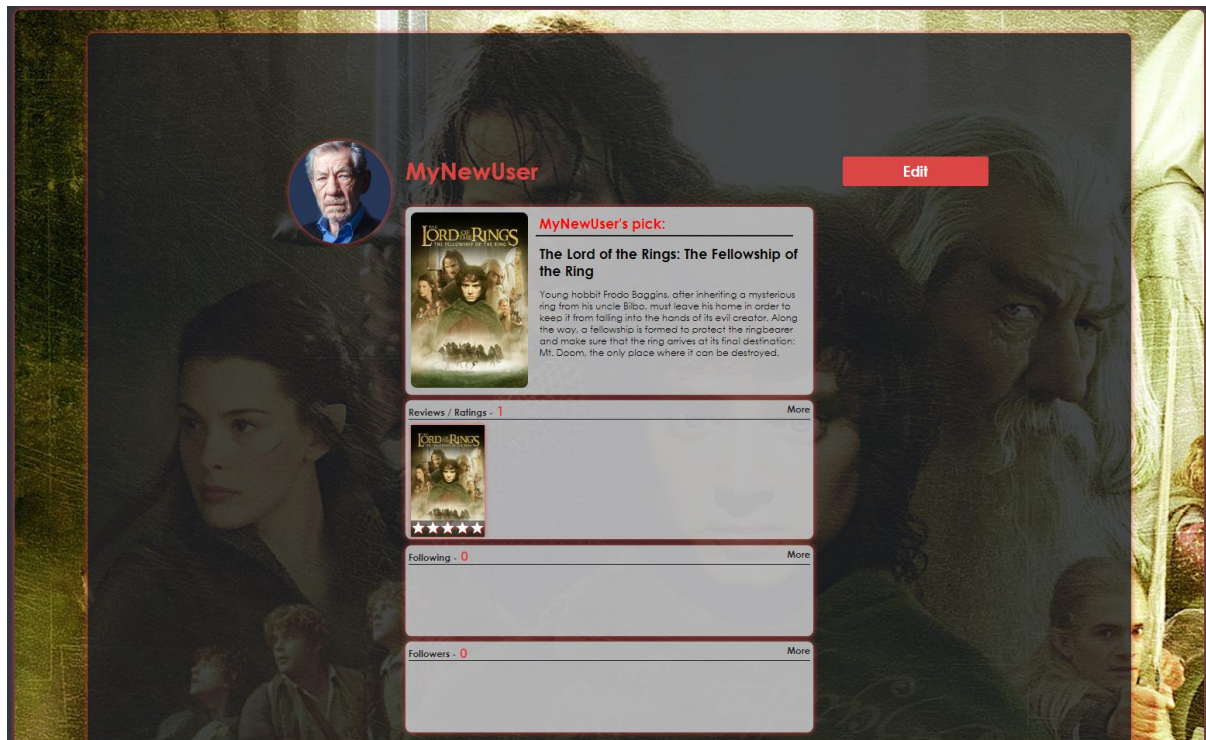
Tyler Brown
C21059883

Movie

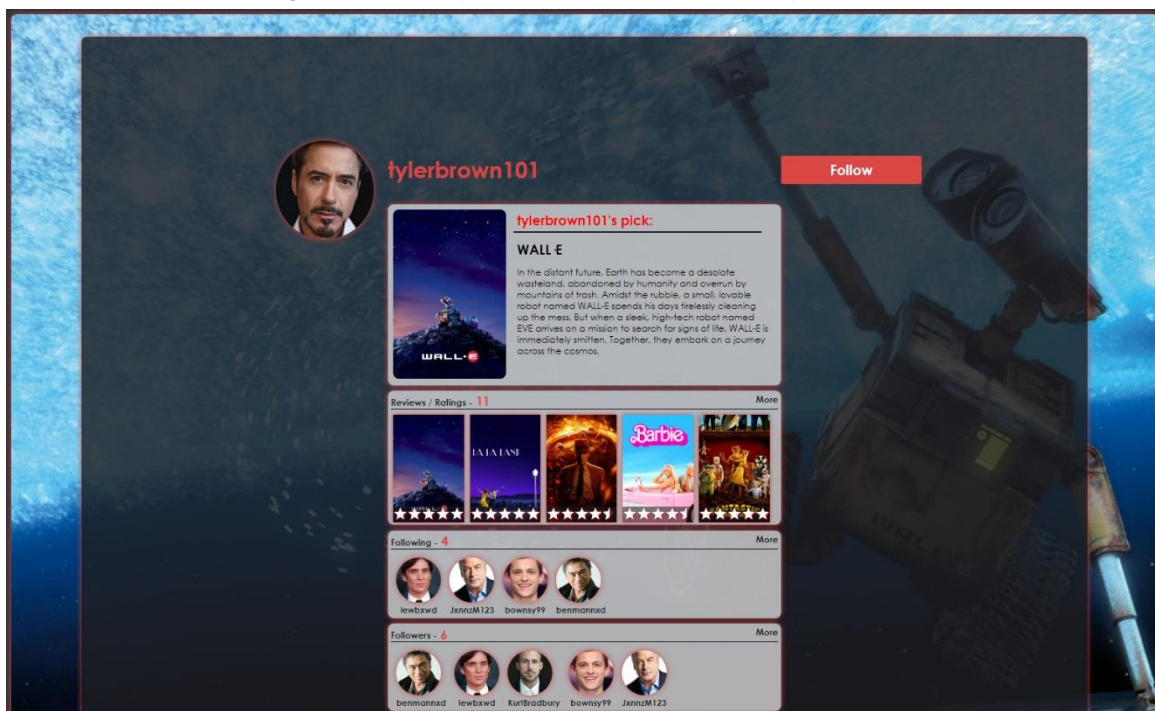


These pages are the pages with the most CSS styling and HTML intensive code in the entire system, this is due to the sheer amount of information it must display. The page is split into 3 sections, the movie panel, the user review panel, and the other users review, the later 2 can disappear providing that no review exist, leaving just the movie panel. The movie panel is nested into another panel that's background is set to the backdrop art of the movie, giving each individual movie a more distinctive looking page, and recognizable one. The movie panel is separate in a wealth of ways using CSS Grids, these different segments include, title and core information, where the title, director, and year of the film is displayed. A description of the movie separated by two thin red lines for segmentation, where room is left in case a larger description exists. The limit of these descript is 500 words where it is then cut off and replaces with an ellipsis. An actor panel, and genre panel where each element is listed out for the user. A post panel for the user to see the box art of the film. Finally, a panel left for button and review information for the user, allowing the user to review this film and to see the average rating of it.

User Profile



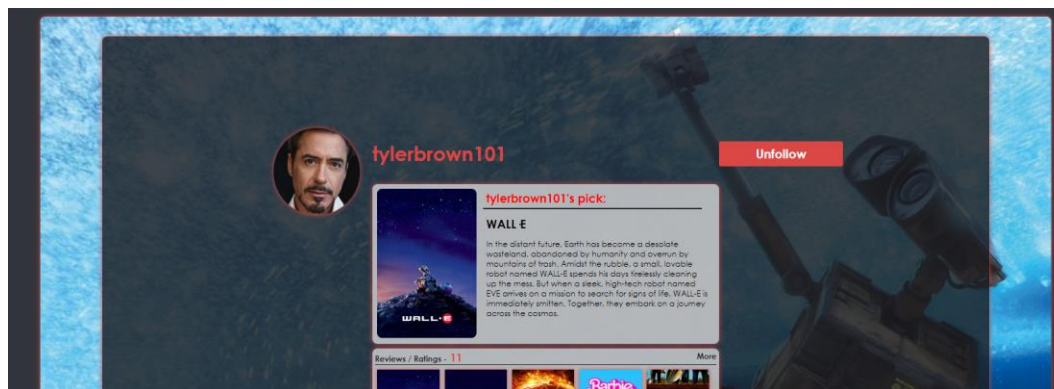
The user profile has customization, so it is only natural that it looks the most interesting and has the most colour and sections to edit. The username and profile are displayed at the top as well as a button which changes its function depending on what user you're viewing. Other sections include a user's favourite panel where information on the user's favourite film is displayed, the users' recent reviews and ratings, a panel for followings and finally a panel for followers. Additionally, the background of the panel is set to the backdrop of the users' favourite film,



Tyler Brown
C21059883

giving the profile that custom feel that the user can change. The red button can be three things, the other two are shown below, however this one says edit since it is the users own profile.

This image shows a user that is not the users own profile, so the red button has change to a follow button. This user profile is also a lot more filled out so it easier to see how a profile will look of a frequent user. Ever follower and following user is show by their name and profile picture, every review and rating show the rating the user gave. Each section has a more button which although is not functional currently, will lead to a page that is an extended version of each panel, where all the content is show instead of the recent few.



This profile is the same as before but shows of the third and final form of the button on the top right, this one is show to a user that the user is already following that is not their own profile. It serves as an unfollow button and allows the user to remove the relationship between them. Another feature I wanted to add to this page though is a block button for users that do not want that user to view any of their information either, however this feature is for the future.

4.3 User Inputs

The lower half of figure 6 is represented by this implementation sections and shows any form of data inputted into the system by the user, this includes profile customizations or movie reviews and rating. However, it also includes navigation as the user is inputting where they need to go.

Navigation

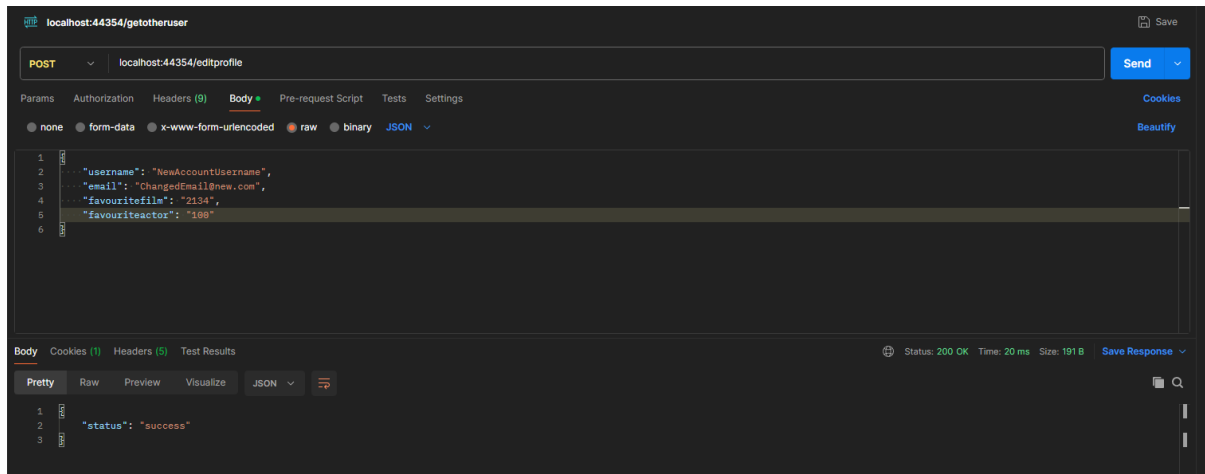


When implementing the navigation system, I wanted to ensure that it was dynamically changing with the current needs of the user, this means that the bar only shows options currently available to the user. In this instance the bar on every page meaning that when the application makes an API request to check if the user is logged in, it does it on every page, this adds a lot of traffic but however is quite necessary. One the user is obtained, the system then makes the choice of which of the two nav bars to show to us, the bar with just login and register for users who are yet to have an account or that are logged out, and the bar with home, search, the username, and logout options for user who are logged in.

Tyler Brown

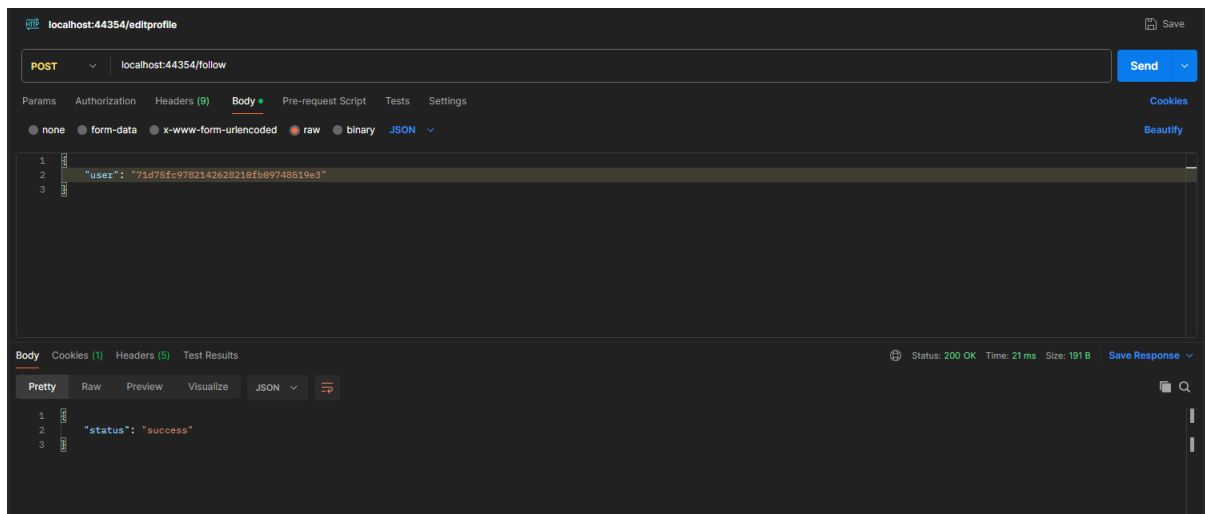
C21059883

POST editprofile



This is the edit profile API interaction and shows the user entering all the information on the user again, this however will only be different if the user changes information in the front end, the user makes a post request with the username, email, favourite actor id and the favourite film id, where a success status is returned. It is not possible to have a failure scenario in this as all data validation is done in the front end, where users are only given the choice of valid options in the for of drop downs or validation checks.

POST follow



This is the following implementation that I made and includes both follow and unfollow functionalities. It serves as a toggle of following status where the user will follow the user if they are not already and unfollow them if they already are, this works in tandem with the check follow endpoint. The user inputs a user id attribute and is return with a status message, this will simply change to failure if the user they are trying to follow or unfollow does not exist although this would only happen in very specific circumstances.

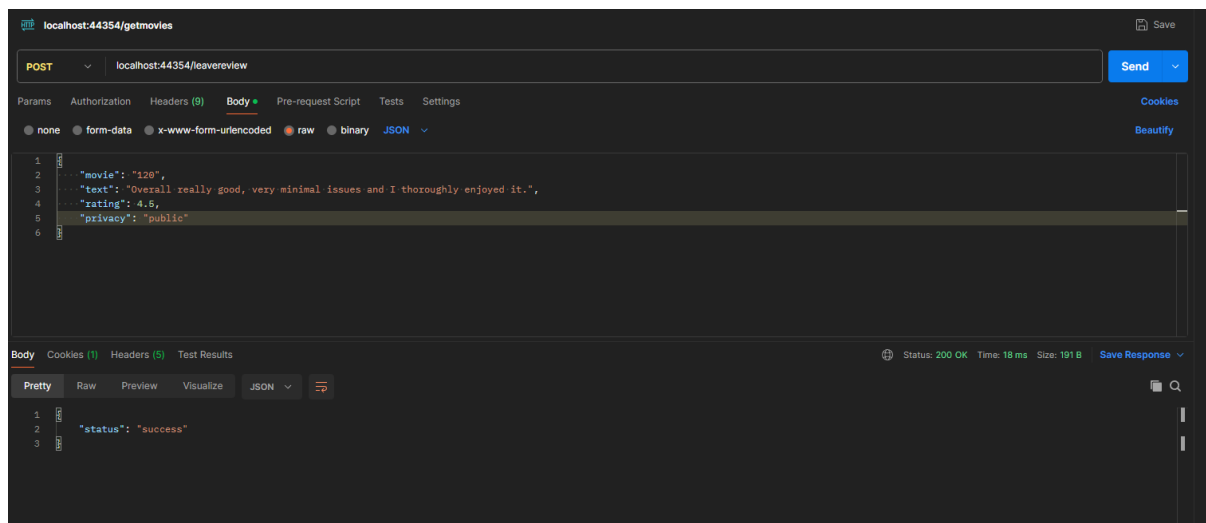
Tyler Brown

C21059883

If the user searches for a film that the system cannot find and results for the user is prompted with the API's response, the Movie database API returns a single page entry since only one is needed to show that no results exist, it also returns an empty list and the total number of pages and results, this most likely would not be used but is still useful to include the full API's response for debugging purposes.

```
1  {
2    "page": 1,
3    "results": [],
4    "total_pages": 1,
5    "total_results": 0
6  }
```

POST leavereview

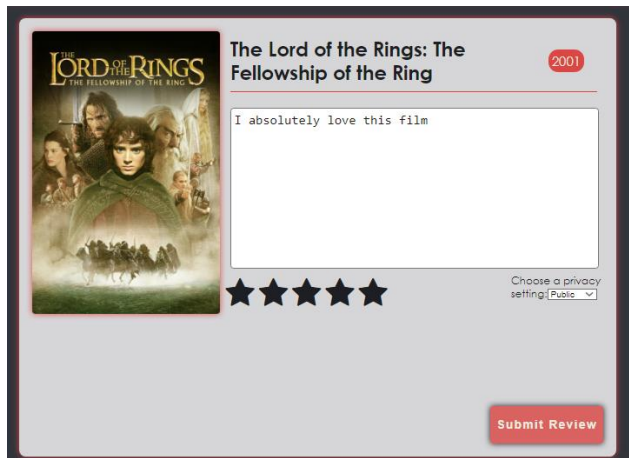


This backend functionality is probably one of the most important ones to exist as it is where most of the information inputted by the user other than account information, will be processed through. The user needs to enter the movie id of the movie they are currently viewing, a text field for a review discussion, the rating out of 5 (decimals being allowed, although users would not be able to select anything that is not a multiple of 0.5, so no validation is needed,) and finally the privacy setting of the review. This needs to be entered via a POST request, and the user is returned with a successful status message, that notifies the front end to return the movie screen.

```
1  {
2    "status": "failure"
3  }
```

If any information that the user has inputted is invalid, including the text being too long, improper rating, or even an invalid movie, the user is returned with a failure message which can be used to notify the user to return the review.

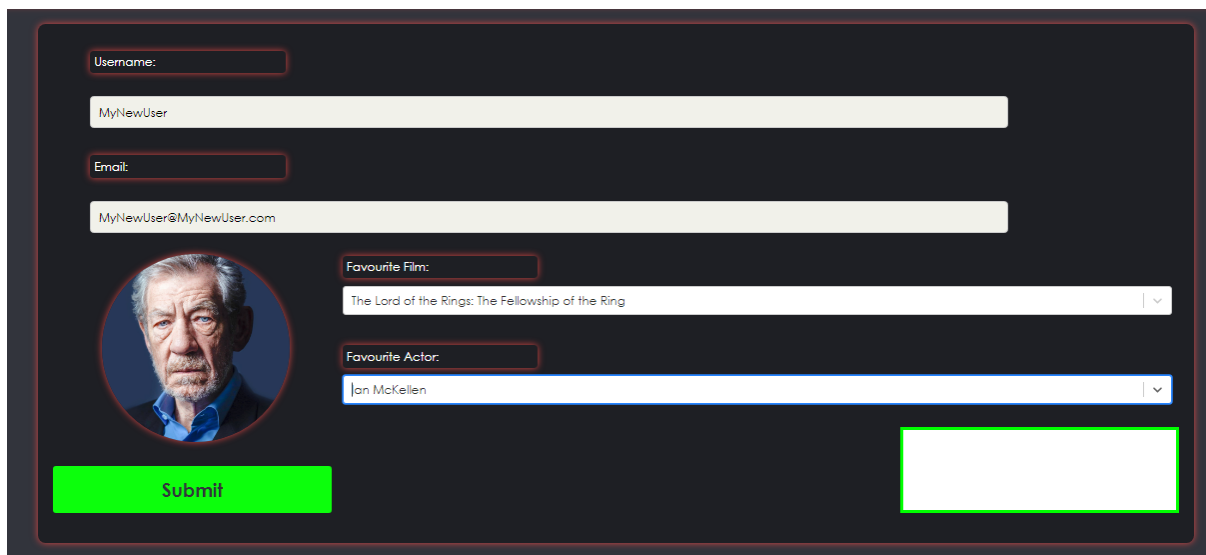
Tyler Brown
C21059883
Leave Review



The screenshot shows a web form for leaving a review. On the left is a movie poster for 'The Lord of the Rings: The Fellowship of the Ring'. To the right of the poster, the title 'The Lord of the Rings: The Fellowship of the Ring' and the year '2001' are displayed. Below the title is a text input field containing the text 'I absolutely love this film'. Underneath the text field is a star rating slider with five stars, all of which are filled. To the right of the stars is a dropdown menu labeled 'Choose a privacy setting' with 'Public' selected. At the bottom right of the form is a red button labeled 'Submit Review'.

This is the review menu of the system and is used for the user to enter information about how much and what they liked about the film they selected. The initial plan was to have a lot more options and information to enter the system about each review. Information that includes perhaps a breakdown of the rating, more information displayed, the date you watched it, etc. However, time limitations made it limited to the systems I could implement, however I did leave space for future improvements. This page has basic information like the title and year of the film as well as the poster, but the rest is user entered information like the review text and rating which is in the form of a star selection slider, and finally a privacy setting. Clicking submit review will place the review into the system and take the user back to the film page.

Edit Profile



The screenshot shows a web form for editing a profile. It has a dark background. On the left is a circular profile picture of an older man with a beard. To the right of the picture are four input fields: 'Username' (containing 'MyNewUser'), 'Email' (containing 'MyNewUser@MyNewUser.com'), 'Favourite Film' (containing 'The Lord of the Rings: The Fellowship of the Ring'), and 'Favourite Actor' (containing 'Ian McKellen'). Below the 'Favourite Actor' field is a green button labeled 'Submit'. In the bottom right corner, there is a white rectangular box with a green border.

This is the final page to talk about and is the profile editing page, where users can change certain aspects of their profile. They can change their username and email address in case they lose access to it, or if they online persona changes. They can choose their favourite film which will change the background and users pick panel on their profile. However, they can also choose their favourite actor, this process will change their profile picture to a picture of the actor, more specifically the most recent photo in the system of them. This will be used to identify them visually in searches for them. The green and white box in the bottom right was a section I

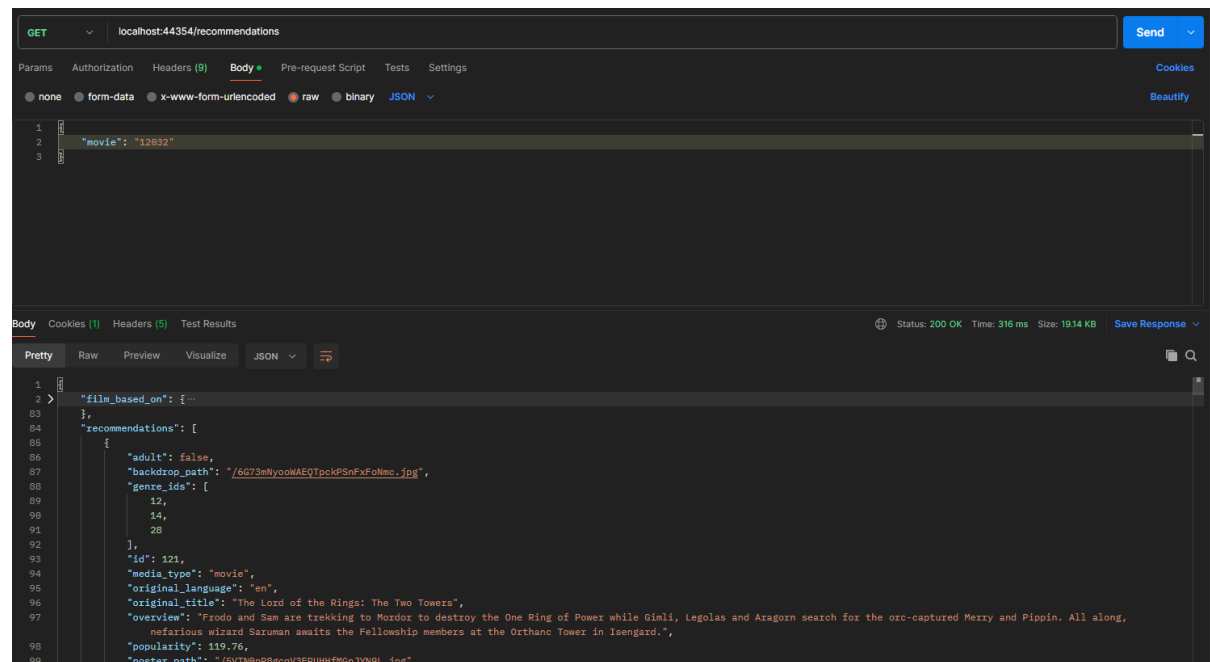
Tyler Brown
C21059883

reserved for privacy settings, but this is a feature that is not fully implemented so this was placed as a placeholder in the CSS grid for future implementation.

4.4 Recommendation System

The representation for this section on figure 6 is a single panel, which is the homepage / recommendation panel, as all recommendation will immediately be shown to the user on login on the home page and separating these pages seem redundant to me.

Recommendation



The recommendation implementation is by far the most complicated and comprehensively design endpoint in the backend as lots of calculations are preformed to decide what the user should be given. The user doesn't need to enter a movie id this is a left-over attribute in post man and serves no purpose, but they make a GET request, where the server then retrieves the user data of the currently logged in user.

This system uses a combination of collaborative filter and content-based filter to ensure that the system works in all scenarios. Since the data in the system currently is not enough for collaborative filtering, the most suitable system is content based, however I have implemented some collaborative features into the algorithms. The system first gets a list of the current users highly rated movie reviews and compares it to the average review of the movie, it comes with an average and calculated value on these two values to find what are the best movies this user has reviewed, ensuring that the general populations opinion and the current user's opinion are considered. Out of all these values the system pick at random one of the highest valued ones, I called this effective rating, the top highest effective rating are then put through a randomizer where one of them are picked, this is to ensure that the system updates a fresh set of recommendations each time so that the user is not just met with the same recommendations every time they load into the site. Using this movie that is selected randomly then the system will then try and find movies that are similar using the API. The API works by then using the genre of the films, directors, studios, actors or even the collection of films to find films that have the

Tyler Brown
C21059883

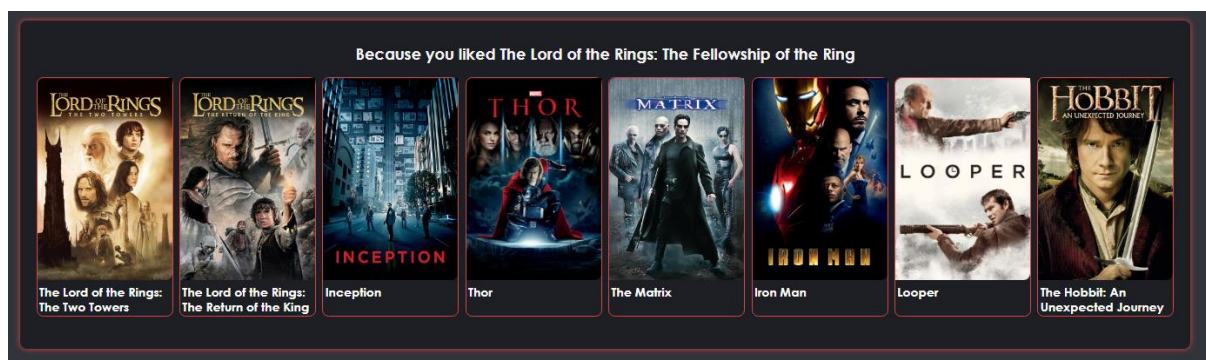
most similarities. In the instance above it makes sense that the Lord of the Rings: The Fellowship of the rings, has been selected, and the recommendation consist of the rest of the trilogy as well as the hobbit trilogy.

The user is then sent all the recommendations where they then select the top 8 recommendation to be displayed. The information provided by each movie includes all information from the API although not all will be used, but again is useful for debugging. Most importantly, names, backdrops/posters, etc are used on the recommendation page.

This system is a compromised to the limiting factors and situation of the project, limited data, limited time and limiting testing makes it hard to implement a fully accurate collaborative filtering system, since the system does not store all the movie data themselves it is hard to compare the movies in a vector like collaborative filtering does.

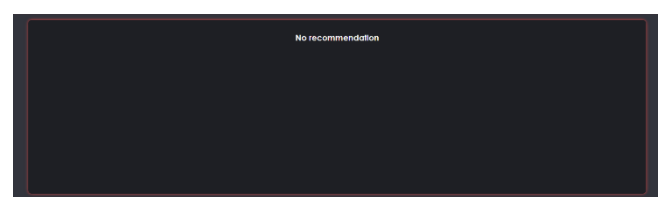
If not, reviews are found which only occurs on a new account where no reviews are left, the user is then met with an empty list on return from the API.

Homepage / Recommendations



This page is likely the page with the longest loading times due to the amount of time it takes to get a response from the backend, since this is the place where most of the calculations happen since its pull recommendations. Originally, I planned for this page to have more, like new releases and friend activity, however due to time constraints this is something I could not get around to and prioritize the recommendations panel. Here we can see at the top it tells you what film it is basing the recommendations on, so in this instance the only high-rate film on the account is lord of the rings, so it is recommending the other parts of the trilogy, other fantasy words, and the hobbit. These recommendations make sense. Clicking any one of these films will take you to the page with all the information about them. The only issue with this page is the name banner of each film, I do not like how it looks and was unable to come up with anything I was happy with, as it sometimes has issues with longer film names, as you can see the films view now barely fit, however this is an extreme case and does not happen often.

The only alternate case rarely happens and only occurs on newer accounts, and in this instance the application does not have any reviews or profile information on the user to generate any recommendations, so instead it just notifies the user that they do not have any recommendations for them.



Tyler Brown
C21059883

4.5 Database Implementation

Note: All data in these tables are example data, test users were given fake accounts although names are real for testing purposes, any other information is fake, and users were notified to not use real emails or passwords. No review is accurate and are simply made up.

All screenshots are taken using DB Browser, this is only used for display purpose to represent the records in the tables, this application was not used for development.

Table: user							
id	username	email	private	favouritefilm	favouriteactor	hashed_password	
Filter	Filter	Filter	Filter	Filter	Filter	Filter	
1 71d75fc9782142628210fb09748519e3	tylerbrown101	tylerbrown101@gmail.com	public	10681	3223	\$2b\$12\$YSdi3RAmxatoDsJbN87tMu33Ji8oTHow1J...	
2 6f3b77a026014ebaa6bf5d1f92f58971	benmannxd	benmannxd@benmannxd.com	public	940721	138435	\$2b\$12\$t0/x4Q95kykBgMjKwVJpHeTc3uZ/...	
3 d7d801337139465da02c13dc27923339	lewbxwd	lewbxwd@yahoo.co.uk	public	872585	2037	\$2b\$12\$9NZD/...	
4 c6d32a6c73014a47b0c4cb5c99a30e1a	KurtBradbury	KurtBradbury@gmail.com	public	920	30614	\$2b\$12\$g/...	
5 2a89ccc14b614ec6b165276ffa89ec5	bownsy99	bownsy99@gmail.com	public	24	1136406	\$2b\$12\$mp4H/...	
6 9ff4ffa40b1e461da3a316b92a4aa954	JxnnzM123	JxnnzM@hotmail.com	public	44896	7447	\$2b\$12\$PeMlo9fHMHMcN1/pru5ku1/...	
7 f5ada7f9fc5340d98c5fa66e35eb430e	NewAccountUsername	ChangedEmail@new.com	public	2134	100	\$2b\$12\$Fp.oCFmM5/...	
8 852df1a9e68046088e5ec5fc81edc2c	MyNewUser2	MyNewUser2@email.com	public	NULL	NULL	\$2b\$12\$ezh.vX8dOfkZFcWRpFDCFu8.CzVfWzwlU...	

This table is implemented as planned however, the privacy settings on the site will remain a last resort feature if given enough time, however this field is implemented into the table in case that system does get around to be implemented on the front end. Hashed password field has all the passwords of the user in a format that is irretrievable to anyone even in the case of a data leak. Favourite actor and film contain the IDs of these film and although are not a foreign field from within this database, they are from the movie database API, these ID's are unlikely to change.

Table: review						
reviewID	movieID	userID	text	rating	privacy	
Filter	Filter	Filter	Filter	Filter	Filter	
1 5bf0d325e45e4bfb9da3e95e0a7c8b54	120	71d75fc9782142628210fb09748519e3	YIPPI! GODLY FILM	5.0	public	
2 91fdf162544f4da0ac5ee0b674125c00	120	6f3b77a026014ebaa6bf5d1f92f58971	shocking	0.5	public	
3 bf18fd7ec62743b19da47f1b17fe8064	123	d7d801337139465da02c13dc27923339	awesome	4.5	public	
4 5bce69d10b7147a19052ff3dc12aa114	872585	d7d801337139465da02c13dc27923339	:O	5.0	public	
5 dc26493e8f5d4ecdb15c7c6383aa3817	121	71d75fc9782142628210fb09748519e3	AWESOME!!!!	5.0	public	
6 9c3f9d11c2e444a9bd10ed8d8fb0c53a	122	71d75fc9782142628210fb09748519e3	Almost perfect	4.5	public	
7 5fb34c3e4e0849a0ae7dbc30c334b948	122917	71d75fc9782142628210fb09748519e3	Pretty good!	3.5	public	
8 d3b40428e1f540ceb2021ce01b34d6e6	49051	71d75fc9782142628210fb09748519e3	Very good!	4.0	public	
9 e2de9721844a46c389e23974682b3722	57158	71d75fc9782142628210fb09748519e3	:YAY	2.5	public	
10 f7f5da074a2d44c28cf2d578d1cd610	920	c6d32a6c73014a47b0c4cb5c99a30e1a	Best film to hit my eye balls	5.0	public	
11 fef43620a7a94e00a5c60e527033ac6d	49013	c6d32a6c73014a47b0c4cb5c99a30e1a	Second best thing to hit my eye balls.	4.5	public	
12 97b5cf35fd45441db5c55e80cbe216fb	10315	71d75fc9782142628210fb09748519e3	CUTE FOX!	5.0	public	
13 e8e1e88643f848999e00b9483c4eadc4	346698	71d75fc9782142628210fb09748519e3	KEN!	4.5	public	

	friendshipID	senderID	recieverID	accepted
	Filter	Filter	Filter	Filter
1	1	6f3b77a026014ebaa6bf5d1f92f58971	71d75fc9782142628210fb09748519e3	TRUE
2	2	d7d801337139465da02c13dc27923339	71d75fc9782142628210fb09748519e3	NULL
3	4	6f3b77a026014ebaa6bf5d1f92f58971	d7d801337139465da02c13dc27923339	NULL
4	5	c6d32a6c73014a47b0c4cb5c99a30e1a	71d75fc9782142628210fb09748519e3	NULL
5	6	NULL	NULL	NULL
6	7	NULL	NULL	NULL
7	9	2a89ccc14b614ec6b165276fffa89ec5	71d75fc9782142628210fb09748519e3	NULL
8	11	71d75fc9782142628210fb09748519e3	d7d801337139465da02c13dc27923339	NULL
9	12	6f3b77a026014ebaa6bf5d1f92f58971	2a89ccc14b614ec6b165276fffa89ec5	NULL
10	13	9ff4ffa40b1e461da3a316b92a4aa954	71d75fc9782142628210fb09748519e3	NULL
11	14	9ff4ffa40b1e461da3a316b92a4aa954	2a89ccc14b614ec6b165276fffa89ec5	NULL
12	15	9ff4ffa40b1e461da3a316b92a4aa954	c6d32a6c73014a47b0c4cb5c99a30e1a	NULL
13	16	71d75fc9782142628210fb09748519e3	9ff4ffa40b1e461da3a316b92a4aa954	NULL
14	17	71d75fc9782142628210fb09748519e3	2a89ccc14b614ec6b165276fffa89ec5	NULL
15	18	2a89ccc14b614ec6b165276fffa89ec5	d7d801337139465da02c13dc27923339	NULL
16	19	2a89ccc14b614ec6b165276fffa89ec5	9ff4ffa40b1e461da3a316b92a4aa954	NULL
17	20	2a89ccc14b614ec6b165276fffa89ec5	c6d32a6c73014a47b0c4cb5c99a30e1a	NULL
18	21	71d75fc9782142628210fb09748519e3	6f3b77a026014ebaa6bf5d1f92f58971	NULL
19	22	f5ada79fc5340d98c5fa66e35eb430e	71d75fc9782142628210fb09748519e3	NULL

This table does look messy however this is because only long random IDs are stored, the ID of each user in each relationship are stored as well as the ID of the actual relationship. The accepted feature was implemented in case the decision was made to make a request system; however, this feature was never implemented, and I resorted to the less convoluted following/follower system.

4.6 Issues

- CORS and authentication

Initially the first few weeks of the project was delayed due to this problem. When trying to initial implement a session system, I was trying to implement it in the innate flask session system. This however, due to flask limitation does not work when using flask as a REST API, so it initially took me a while to figure this out and a lot more time to figure out a solution where I wrote the following code to fix this problem, by creating a client that retrieves the API request and implementing cores in the back end.

```
1 import axios from "axios";
2
3 export default axios.create({
4   withCredentials: true
5 })
```

```
CORS_HEADERS = 'Content-Type'
SESSION_TYPE = "redis"
SESSION_PERMANENT = False
SESSION_USE_SIGNER = False
SESSION_REDIS = redis.from_url("redis://127.0.0.1:6379")
```

By import CORS we are effectively replacing the session function in flask.


```
3 from flask_session import Session
4 from flask_cors import CORS, cross_origin
```

This solution was very technical and took a lot of work to figure out, but thanks to masses of research I managed to figure it out. (Saad Irfan, 2022)

- Decision to make users not be able to use their own profile pictures.
This problem when developing was more of an ethical problem I needed to consider. By giving users, the option to input their own profile pictures would result in the site needing a moderation system on user profiles, not to mention the database storage size would increase marginally, and the complexity of the system would be increased. So, by limiting profile pictures to actors, the user has the same level of customization, as all the actors are relevant to the site, while also not having the capability of uploading malicious or rude images. This also helps with the fact that users own faces on the site possess an ethical and privacy threat, even if they choose to upload it themselves, this problem was an easy choice to make, however is important to consider the reasoning behind it.
- API get requests not retrieving.
Initially when working with the backend post requests, I was getting issues that the code on the backend would run but nothing would be returned, this is since a thing called a preflight request was not being return to the client, this mean that the client then does not trust the API so does not allow for the actual response to reach it. This was a huge issue that took hours of work to solve and hours of research, but however some minor tweaks to the API client above adding the with Credentials parameter served as a fix, this minor fix solve a massive problem and should not have taken a long as it did to solve.
- React looping and erroring the API out.
Inside react, I used the use Effect function to run code on the page loading. This however implemented an issue where the code would infinitely loop, causing API requests to be spammed to our backend, this would time out the session as the backend saw the request as malicious and spam, effectively crashing the entire site. The simple solution to this problem was adding an empty array into the arguments of this function. This would tell react that this effect does not depend on any parameters or states, so it does not rerun itself, solving the problem. This issue was completely down to the limited knowledge I had on react, however I am grateful to have learned the solution to this common problem.

Overall, the issues that this have occurred over this project has mostly been limitations on my own knowledge required for the technical, programming aspects of this project, as a lot of these technologies are fresh and new to me so it would only make sense that a lot of what I have implemented be new to me and a great learning experience.

5 Evaluation

The purpose of this section is to emphasize and explore the success of this project by creating and deploying a survey to a series of users who tested my applications and compare the application and results to already existing technologies. Discovering whether the project's success or not, was a result of my own technical limitations or proficiency, or whether it was because of other limiting factors.

5.1 Survey Design

The design of the survey needs to make sure that enough and thorough questions are asked to strong reflect on the applications usability and overall functionality. By researching the users experience from users themselves, we get an example of how this system would perform in real word deployment, which by any means it is not ready for, but it is nice to how some indicator and how it has been produced so far. The following paragraphs will explore each question and explain the design choices of each of them, and what I expect the results should be or how they should be useful to me.

The first question the test users where asked was a rating question where the users where ask to rate the aesthetics and overall look of the site out of ten. This question gives the user a platform to judge and criticise the look of the website, this may be the colour, information organisation, panelling, the symmetry of the site or other aesthetic factors of the site. This will allow for minor changes are tweaks to the site and gives me the opportunity to understand what works and what does not work too people using the internet, before deploying it on a large scale.

The next question is similar to the last but instead asks the person to rate the user experience and user flow of the applications, this means that the user rates how easy they interacted and flowed through the site, this has a bigger impact on the result of the project due to the user experience having a lot more impact on the success of a application than the aesthetics of a site, although this is important too. This will allow me to further rethink or even reimagine an approach to the user flow if users do not like it.

The next question asked considers accessibility and asks the user to rate how accessible the application is and whether the application considers users with challenges or difficulties when accessing the site, which will tell me whether the application needs to cater for more demographics and needs more accessibility features.

The fourth and likely the most important one since this is the main feature of the project is a question that ask the users whether the recommendations given to them, they deem valuable insight and good suggestions to what films they should consider watching them. Asking them to be rating this system out of 10, 10 being that these films are exactly the films they like 1 being completely irrelevant films they dislike.

The next and fifth question relates to the retrieval of data. Users are asked whether the film page portrays enough information about each film or whether they needed more information about them, although I believe this question will be answers mostly with yes, as a lot of Information is portrayed, it is always good too just double check.

Next question is a bit of a repeat however, it is worded slightly differently so users may be able to understand what it is asking more. It asks users whether the website is easy to use and intuitive,

Tyler Brown
C21059883

which is like accessibility and user flow, however this one specifically will tell me whether the site needs to have explanations or tutorials, or whether I need to make it slightly more obvious what the sites usage should look like.

The final two questions are text field questions, where users are given a space for errors they find, this is to ensure than any errors within the system can be fixed and to figure what areas of the technical implementation I failed on. They are also given an area for any suggestions or future features they wish to see implemented, which will be useful for future work.

5.2 Survey Feedback

The survey went out to 6 of my close friends who were asked to be critical and fair to the system and judge it without bias, each where sent a word file with all the relevant questions as well as a brief about the project and what they should do to test it. I gave it out and receive all the results in about a week of sending them to them, I did not rush them as I had the time to wait, however they all where very timely and considerate that the project was time limited.

Questionnaires were submitted alongside this.

The results of the survey are shown below.

Q1 Aesthetics	Q2 UX	Q3 Accessibility	Q4 Recommendations	Q5 Informative	Q6 Intuition
Average: 7.67	Average: 8.33	Average: 7	Average: 9.33	All YES	All YES

Errors:

- “The site crashes occasionally when logging in”.
- “Site crashed a few times while I was loading up”.
- “Not being able to add my own profile picture instead of my favourite actor.”
- “Nothing so far”
- “Site does not seem responsive to mobile devices; Site sometimes loads twice when loading user profiles”.

Suggestions:

- “to compare what films you have seen with other friends”
- “add series to the website”
- “Further profile customisation”
- ““Adding a comment section so people can leave their thoughts on other peoples’ reviews.”
- “Site has all that it needs as far as I’m concerned”.

When looking at the results of this survey, the overall scores and responses are quite positive. Users thought the aesthetics of the site looks quite well, although there is room for improvement, even with them noting that.

Tyler Brown
C21059883

These results show that in a lot of the aspect of the application preform quite well and are very appealing to the test users I gave it too. However, we must consider also that these results may have a slight bias since they were close friends testing and are all from a single demographic. However, they were asked to be critical and fair when reviewing the application. The aesthetic of the application mainly responds to colouring, organisation and panelling within the application as seems like it is rated high so I would deem the look of the application very successful with room for improvement. The user flow and user experience seem to be very well implemented by these results; users have even specified they like how they are immediately met with recommendations. Accessibilities is quite high although more could be done to improve who and how users can use my application, users say the contrast of content to background really helps the visual of the application and helps impaired people. The highest score of them all would be the recommendation system, users felt that this system accurately represented their tastes and all though they would enjoy the films that were recommended to them. A worry of mine when developing was what and how much information I should use and display to the user, so I am very happy to hear that all the users felt that the film page was informative enough and did not require any more information. Finally, all users had no problem figuring out how to use the site and found it very simple and intuitive to use, without the need of any tutorial or simplification. To summarise, the project appears to be successful, and users enjoyed the use of the system.

5.3 Comparison with Existing Technologies

The most similarity I see between my application and other technologies is with letterboxd, however I do see some similarities to all of them. This is bound to happen due to the nature of the applications, them all having a similar purpose; however, all these systems use collaborative filtering since they have larger data bases. Whereas my application has a unique personalized response, I would like to think that my application has a personalized touch to it that the other differ from having a lot of profile customizations and a more social and communal aspect to it rather than letterboxd, rotten tomatoes and IMDB. But does lack the list making and sharing techniques used in these systems.

The survey indicated that my site preforms and feels a lot better than rotten tomatoes did during my research as that site was very poorly designed, and although is functional is very counter intuitive and complex to use, as well as it looks very old fashion and outdates. It also indicates that my review system seems to be very accurate, possibly better than these system due to its capability of running on a smaller set of data, these systems are professional and large scale system so its very good to hear that my system at least is on par with this system given the time frame and scale, the system may not be as well thought through as the other or completely accurate and refined, but it seems all the user receive recommendations with accurate predictions.

It is very unfair to compare these applications to my applications to mine as they were developed by a full team of developers and full teams over the span of possibly years. They also have the reputation of being accurate and well-respected sources of information whereas mine was limited to 12 weeks with me as the sole developer who has limited knowledge. However, comparing does help me see where I can improve and where I went wrong, so comparing these I will need to take this into account, and considering the differences in times and developers, I

believe the solution was successful and a great job was done developing this application, as its very close to being a professional application that could be deployed.

6 Conclusion and Future Work

To conclude the entire project, I created a Movie recommendation system which also prioritize a social media influence too it where user interaction is prioritized and encouraged to all users, I created both a backend and front end using python and JavaScript where each would interact with each other to retrieve data, user customization was an aspect I focussed on to ensure user recommendations where refined and accurate when calculated, to determine the success of the project I performed a survey to a group of close friends who thoroughly tested my system by creating accounts, following each other and rating different movies to then see if the recommendations where accurate or not. The results of which seems to show a success in development. An abundance of research was preformed in this project where I explored different movie recommendation algorithms and different approaches to providing the user with suggestions, research was done into database design and user interface and flow to guarantee users leaving my site which a positive experience, the following sections explore the key findings and limitation I had in the project.

6.1 Key Findings

Here are some of the key findings and discoveries made by developing this system and what they mean, and whether this finding was to be expected, overall contributing to the project's success:

- When it comes to developing user profiles and user accounts, the development of specific customizations to the users are account are very appreciated, users really value the level of uniqueness and personality they can give to their profile distinguishing them from other users also making them easier to discover, which again is something that is important to an application that you search for users in.
- How important real world social reactions can be to the success of social interactions between users in applications like this, users need to be specifically looking for people or have some way to discover user to have the chance of finding someone they want to interact with.
- The importance of a larger data set for more complex recommendation systems, collaborative filtering requires a lot of users who have already review a lot of the movies available to the user, or no similar users will be found.
- How accessibility features and to ensure an inclusive design is implemented can affect user experience and usability of a system, which will eventually affect the overall traffic and use a system gets.
- When considering user security and data integrity, the implementation of proper hashing algorithms, user authentication and encryption, we need to take into consideration that it increases the complexity and harshness of the development process. This took most of the project time, and proves to still be an important point, hence why so much time must be dedicated to ensuring that no information is leaks and users are safe to enter their information into your site.

Tyler Brown
C21059883

- Implementation of an external API or resource makes your system reliant on it, since my application used the movie database API any time that the system would error could be due to the API erroring itself, this issue cannot be resolved by me but instead I could catch this error, and display the issue to the user, effectively making any issue the API my systems issue, without having the abilities to amend it. So, this is the sacrifice made for the uses the API brings to the table.

6.2 Limitations

Throughout the project there where a few times that the development and design would run into some limitations and technical issues, these are only natural in a project of this nature and it is up to the developer's skill and expertise to find alternative routes or solutions to these limitations, here are some of the limitation I ran into and how I solved or avoided them.

- The time restraints on the project being only 12 weeks really became a problem as a lot of what I was implementing was often falling behind on schedule, 3 months is not enough to develop a fully implemented system at a professional level, especially by a single developer, the work around for this was the way I prioritized tasks, removing features that were not as important to ensure the project came to a working prototype before the end, rather than an unfinished system with more implementations.
- The technical knowledge of one person is not going to be enough for a full stack development process, that of a student is especially not going to be enough for a design that would compete in a competitive market, I can not know enough to make a perfect system with all the technologies I know, the way around this was to use a blend of technical skills I am already proficient in and one I could learn during the project, not only providing new learning opportunities but also allowing me to refine my knowledge.
- The amount of data I could obtain on each movie was a problem from the very beginning of the project, the initial idea was to find some way of obtaining all this information and have it in my own system so that I could rely on my own data base and implementations to work on however the more I looked into it, the more it seemed impossible as there were simply too many movies out there to store information on, a solution I did find to this was the API, this limits my design around their system but also allows for more information to be used about each movie.
- Limitations on what algorithms I could use due to user entered data became an issue and limitation where it was obvious the application was not going to have enough users for a more complex designed, accurate recommendation system, so the only possible solution is to use less complicated easier algorithms that used less information to come up with recommendations for the user.

Despite all these limitations I was still able to produce and build, a fully efficient system with basic functionalities, with a user friendly interface and an easy to use system according to the survey results and feedback given to me by users, these limitations were roadblocks that did not limit the success of the project, but rather just made it more challenging.

6.3 Future Work

The future of my project is full with refinement and increase and the number of implementations and improvements I was initially planning to make, a deeper search into the colouring and UI of

Tyler Brown

C21059883

the site is likely needed to improve the aesthetics a little bit. However other than that the implementations of the following features would be a smart move:

- Fulling implementing a privacy system where user can choose to hide or stop people from finding or seeing their profile or review by changing a setting, users could also choose an unlisted section where it would not come up in search but can be used to navigate to if the user has the link.
- Commenting system to implement a more social aspect to each of the review where user can discuss in a forum type setting about the film they are reviewing or adding a rating too.
- Although initially custom profile pictures seem to cause an issue, a lot of the survey results said that users would prefer to upload their own profile pictures despite the privacy issues they cause. So, this may be a good option for users to have should they choose.
- A more refined recommendation system in the future would seem to be beneficial should the application get a larger user base, enough to start using a more collaborative filtering-based system.

7 Reflection on Learning

To consider what I have learned from this project I also need to ensure I understand what knowledge I already had and compare it to what skills were required to develop the project to see what I had learned throughout. The course I am studying has provided a great deal of knowledge throughout the years to assist in this project, it was a large influence on why I choose some of these technologies.

The technological skills used in this project that I already knew were python and more specifically the use of flask, this skill set was taught during year one of my course and was a technology that I really enjoyed using which is why I choose to use it here, as well as it seems to fit the specific requirements. However, I did not learn to use it in the way I used it here, as a REST API. This factor allows me to refine and broaden my knowledge on python and flask in a way that is not usually done, as instead of just refining my knowledge on the syntax and error handling of it, I increase the ways I am able to use and the amount scenarios I can use python and flask in, learning this technique makes python a lot more usable for more problems in the computer science industry for me, where I initially view python as a very math heavy language where most of what it is used for is language models and maths or physics simulations, but now I see it in a way that allows it to be used in web development.

Now the technologies I had no knowledge of prior to this project was ReactJS, this language of web design, as always been something I have wanted to learn, so I decided that the project was going to be using it so that I would learn it. Throughout development I watched countless tutorials and read through countless documents not only learning the syntax of React and JavaScript but also learning techniques to avoid and what techniques are used in specific scenarios, I learnt about use Effects and use State parameters, React's own form of HTML and CSS. The want to learn this technology was a huge factor that kept me motivated and kept my unwavering attention on developing throughout. I was eager to prove that I could learn it in time to use it here. Additionally, CORS and Redis was something I had never even heard of let alone

Tyler Brown

C21059883

considered using, but the requirements of the system, needed me to learn about these technologies so over the starting weeks I learnt how to implement these systems into my code for my authentication and data integrity.

The general skills I used and refined in this project was a great help my overall performance and happiness during this project, I learnt how to better time manage and organise the tasks I had to ensure that any deadlines I had whether they be academic deadlines or my own personal deadlines where met, limiting myself to how far back I could fall behind. Discussing in detail my project constantly to my supervisor ensure the project is following on track as well as collecting survey results help improve my communication skills effectively allowing me to collect information and receive constructive feedback on how to better the system.

References

Abhinav Ajitsaria. Build a Recommendation Engine with Collaborative Filtering. Available at: <https://realpython.com/build-recommendation-engine-collaborative-filtering/> [Accessed on: 17/04/2024]

Alice Hughes 2021. Britons spend more than 100 days of their lives deciding what TV shows and films to watch. Available at: <https://www.independent.co.uk/news/uk/home-news/britons-choosing-tv-shows-movies-b1915481.html> [Accessed on: 06/04/2024]

IMDb 2024. The worsts most popular and authoritative source of movie, TV, and celebrity content. Available at: <https://www.imdb.com> [Accessed on: 12/04/2024]

Letterboxd 2024. Social platform for sharing your taste in film. Available at: <https://letterboxd.com> [Accessed on: 14/04/2024]

Mark Milankovich. The Cold Start Problem for Recommender Systems. Available at: <https://medium.com/@markmilankovich/the-cold-start-problem-for-recommender-systems-89a76505a7> [Accessed on: 18/04/2024]

Mozilla 2022. Cross-origin resource sharing. Available at: https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS#see_also [Accessed on: 1/05/2024]

Rotten Tomatoes 2024. Home of the Tomatometer, the most trusted measurement of quality for movies & TV. Available at: <https://rottentomatoes.com> [Accessed on: 13/04/2024]

Saad Irfan 2022. How to handle CORS in flask? Available at: <https://rapidapi.com/guides/handle-cors-flask> [Accessed on: 07/03/2024]

Sally Wolleat 2024. What are some effective ways to organize and present information on a website? Available at: <https://www.linkedin.com/advice/1/what-some-effective-ways-organize-present-information> [Accessed on: 19/04/2024]

Shahil Modan 2023. The benefits of React JS and reasons to choose it for your project. Available at: <https://www.peerbits.com/blog/reasons-to-choose-reactjs-for-your-web-development-project.html> [Accessed on: 30/04/2024]

Tyler Brown

C21059883

Thomas Peham 2023. Why don't consumers trust AI-Driven recommendations? Available at: <https://builtin.com/articles/consumers-lack-trust-in-ai-recommendations> [Accessed on: 08/05/2024]

Victor Dey 2021. Collaborative Filtering Vs Content-Based Filtering for Recommender Systems. Available at: <https://analyticsindiamag.com/collaborative-filtering-vs-content-based-filtering-for-recommender-systems/> [Accessed on: 17/04/2024]

Web Accessibility Initiative 2015. Designing for Web Accessibility. Available at: <https://www.w3.org/WAI/tips/designing/> [Accessed on: 25/04/2024]