

Course code COMP9900
Course title Computer Science Project
Project title Movie Finder System
Group name 9900H14AKuangbiao

Boning Wang	z5364629	z5364629@ad.unsw.edu.au	Front-end
Yihao Zhou	z5327558	z5327558@ad.unsw.edu.au	Front-end
Shaoming Li	z5328070	z5328070@ad.unsw.edu.au	Back-end
Younian Zhang	z5262910	z5262910@ad.unsw.edu.au	Front-end
Xiandong Cheng	z5342690	z5342690@ad.unsw.edu.au	Back-end

Table of Contents

1. Overview	1
1.1 System Architecture	1
1.2 Software Architecture	2
1.3 Project Objectives	8
2 Functionalities	9
2.1 Authority	9
2.1.1 Register.....	9
2.1.2 Login	9
2.1.3 Logout.....	10
2.1.4 User Profile.....	11
2.2 Movie.....	13
2.2.1 Home Page	13
2.2.2 Search and Browse	15
2.2.3 Movie Details.....	15
2.2.4 Review and Rating	17
2.2.5 Wish list	18
2.2.6 Banned list.....	20
2.2.7 Recommendation	23
2.3 News.....	25
2.3.1 News list	25
2.3.2 News details	26
2.4 Discussion	28
2.4.1 Discussion Page	28
2.4.2 Discussion Editing Page	29
2.4.3 Comments of Discussion	30
3 Third-party functionalities	31
3.1 Django REST Framework (DRF)	31
3.2 django-filter	31
3.3 React.....	31
3.4 React Router	31
3.5 MUI.....	31
3.6 Swiper.....	32
3.7 Jest	32

4 Implementation challenges.....	33
4.1 Front-end Part.....	33
4.1.1 Challenge of Jest mocks global.fetch return undefined	33
4.1.2 Challenge of updating React components in Jest.....	33
4.2 Back-end Part.....	33
5 User manual	37
5.1 Setup the code environment	37
5.2 Building the system.....	37
5.3 Running the system.....	38
5.4 Using the system	38
5.4.1 Login Page, Register Page and Home Page	38
5.4.2 BROWSE	38
5.4.3 Movie details page	39
5.4.4 WISHLIST	39
5.4.5 NEWS.....	39
5.4.6 DISSCUSSION	39
Reference	40

1. Overview

With the increasing consumption level and the process of building film and TV culture, the movie industry is in a booming stage. With new movie-going users and new movies on the shelves every day, more and more people are choosing to relax by watching movies. Movies are a very popular art, but there are good and bad movie quality.

For viewers with limited money and time, how to judge the quality of a movie depends heavily on the reviews of popular movie review sites. The current international mainstream websites lack innovation and are limited only to the aggregation of movie information, unable to make recommendations based on users' preferences. The abstraction of a single scene, the lack of content that can generate topics, and the weak perception among users consume their imagination and cannot give the expected interactive feedback. How to let users find their favourite movies in such a huge scale requires the system to transform from passively receiving user requests to actively sensing users' information needs, thus saving users' time and increasing the stickiness of the system. Our team planned to develop a movie review site that would give users a stronger need to discover movies and share them. For targeted users who know they want to watch a certain movie, they can directly query or fuzzy match to find a movie based on the movie name provided by the user; for non-targeted users, to recommend personalized and quality content through self-selected tags and let the system automatically give users recommendation predictions. Establish a personal centre for users to manage their favourite movies and add a forum function to establish a social circle among users to find people with similar movie watching interests to watch movies together and communicate with each other to increase user stickiness.

Nowadays, movie resources are an important part of network resources, and as the number of movie resources on the network becomes more and more huge, it is urgent to design a personalized movie recommendation system. Therefore, this system aims to recommend movies with a high degree of compatibility with each user's interests. The website is a social movie review website with personalized recommendation and sharing functions, mainly targeting young people who have a strong need to discover movies and share their opinions. The system includes the implementation of a front-end display interface for movies, movie ratings, movie collections, and recommendation algorithms. The website provides a lot of information about current and upcoming movies, and on top of that, personalized recommendation and community features are introduced to meet users' interests and social needs.

1.1 System Architecture

The architecture of the system can be generally divided into 3 layers: Presentation Layer, Business Layer, Data Layer which shows in the graph below.

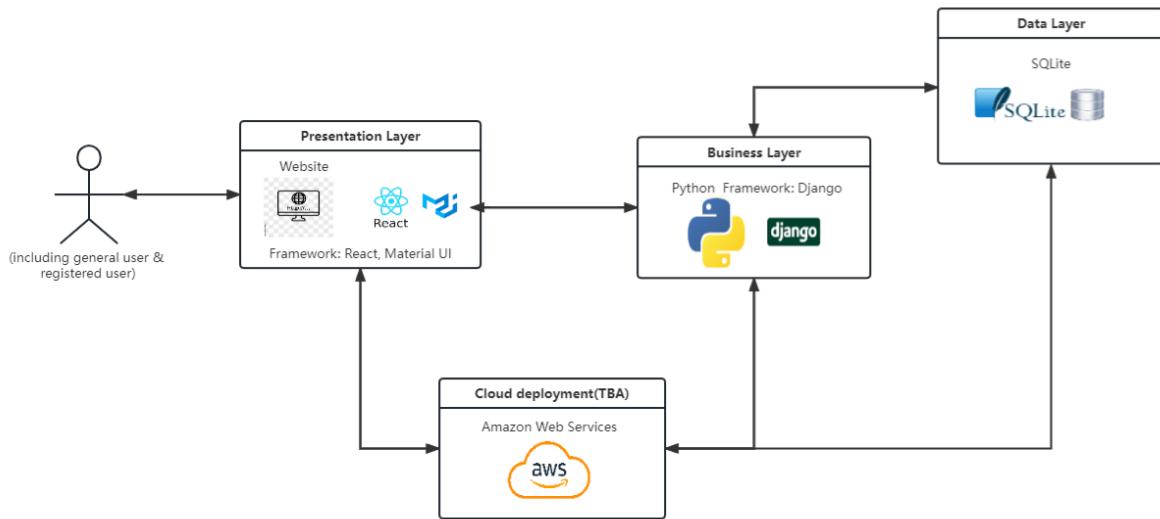


Figure 1 – The architecture of the system

1.2 Software Architecture

1.2.1 Data Tier

As a lightweight relational database, SQLite has many advantages, making it an ideal choice for databases in some specific scenarios. Here are the advantages of SQLite and the reasons why you choose it as your database:

1. **Lightweight:** SQLite is a very lightweight database, its entire core is only a few hundred KB, no additional installation and configuration is required. SQLite is a very good choice for projects that require a lightweight solution.
2. **Serverless:** SQLite is a serverless database, meaning it doesn't need to run on a separate server process or system. This makes SQLite ideal for projects that require a low-cost, low-maintenance solution.
3. **Cross-platform:** SQLite supports multiple operating systems, including Windows, macOS, Linux, etc., making it easy to deploy and use on different platforms.
4. **Transaction Support:** SQLite supports ACID transactions, which means it ensures data consistency and integrity, keeping data safe even in the event of a system crash or power failure.
5. **Self-contained:** SQLite databases are self-contained, which means that all data is stored in a single file for easy backup and migration.
6. **High performance:** Although SQLite is a lightweight database, its performance is still very good, especially in read operations. SQLite is an excellent choice for applications that need to query and retrieve data quickly.

To sum up, SQLite is lightweight, serverless, cross-platform, transaction-supported, self-contained, and high-performance, making it an ideal choice for many projects. Especially for those projects that require rapid development, low cost and low maintenance, SQLite can provide an easy-to-use database solution.

The process of organizing the display information according to the data organization model requires a lot of resources, and the design work is also very tedious, which results in low efficiency. The E-R method is easy to solve this problem, entity-relationship method is referred to as E-R method, the tool used in this method is the E-R diagram, entities, entity attributes and inter-entity relationships make up the E-R diagram.

There are 7 entities in the database, which are user profile, user wish list, movie, movie review, movie news, forum article, and forum comment.

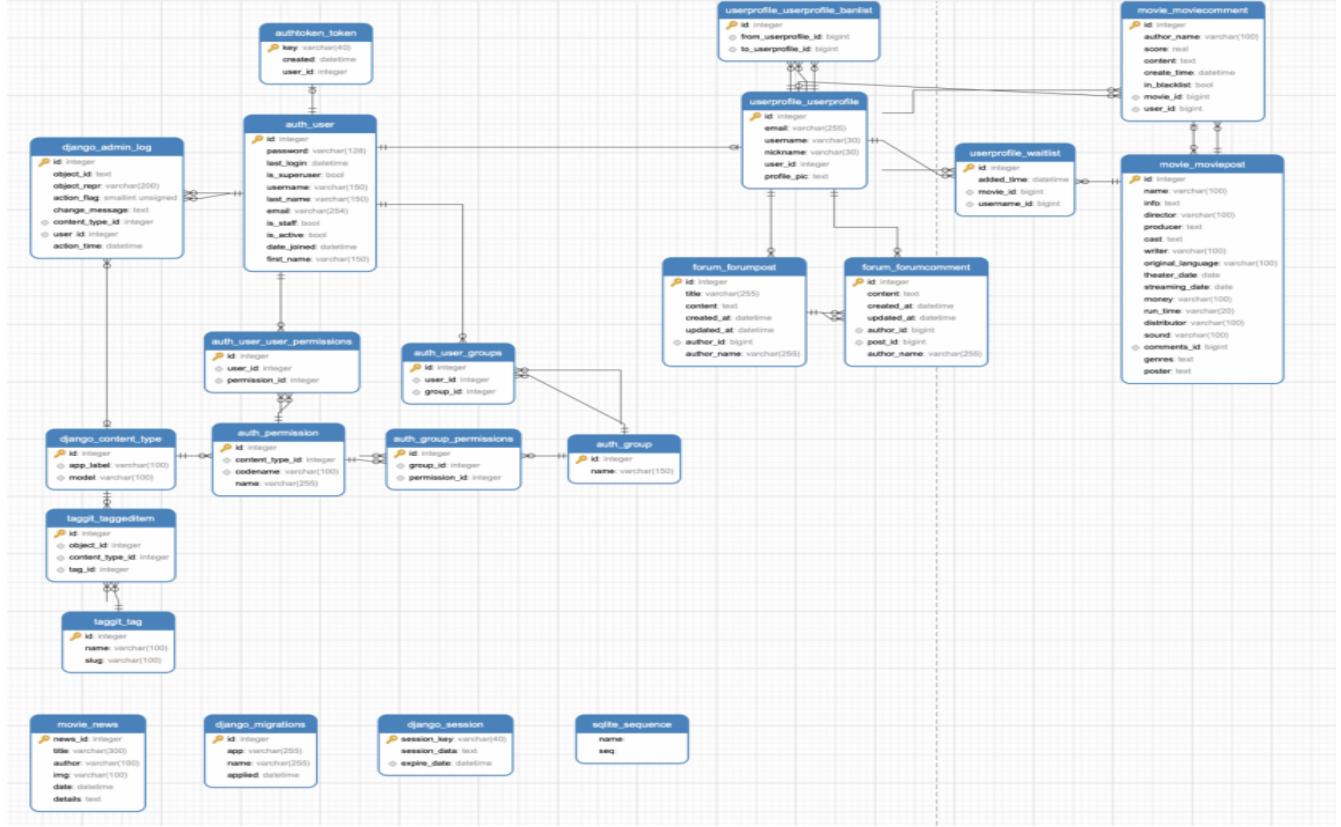


Figure 2 - System E-R diagram

UserProfile				
	Data Type	Primary Key	Allow empty	Function Description
id	integer	YES	NO	Basic user information fields
email	varchar	NO	NO	Basic user information fields
username	varchar	NO	NO	Basic user information fields
nickname	varchar	NO	YES	User information fields
profile_pic	text	NO	YES	Additional user information fields
banlist	integer	NO	NO	User blacklist
waitlist	integer	NO	NO	User waitlist for movies

WaitList				
	Data Type	Primary Key	Allow empty	Function Description
id	integer	YES	NO	user_id
username	bigint	NO	NO	username
movie	bigint	NO	NO	Movie name
added_time	datetime	NO	NO	Time for movies to be added to the collection

MoviePost				
	Data Type	Primary Key	Allow empty	Function Description
id	integer	YES	YES	movie_id
name	varchar	NO	YES	Name of the movie
poster	text	NO	YES	Movie poster (as a text field)
genres	text	NO	YES	Genre of the movie (using Django-taggit module)
info	text	NO	YES	Info about the movie
director	varchar	NO	YES	Director of the movie
producer	text	NO	YES	Producer of the movie
cast	text	NO	YES	Cast of the movie
writer	varchar	NO	YES	Writer of the movie
original_language	varchar	NO	YES	Original language of the movie
theater_date	date	NO	YES	Release date (in theaters)
streaming_date	date	NO	YES	Release date (streaming)
money	varchar	NO	YES	Box office (gross USA)
run_time	varchar	NO	YES	Runtime of the movie
distributor	varchar	NO	YES	Distributor of the movie
sound	varchar	NO	YES	Sound mix of the movie
comments_id	bright	NO	YES	Comments for the movie

MovieComment				
	Data Type	Primary Key	Allow empty	Function Description
id	integer	YES	YES	moviecomment_id
auther_name	varchar	NO	YES	Author name of the comment
score	real	NO	NO	Score given for the movie (as stars)
content	text	NO	NO	Content of the comment
create_time	datetime	NO	YES	Time when the comment was created
in_blacklist	bool	NO	NO	Flag to indicate whether the author of the comment is in the blacklist
user	bigint	NO	NO	user_id
movie	bigint	NO	NO	Name of the movie

News				
	Data Type	Primary Key	Allow empty	Function Description
news_id	integer	YES	NO	ID of the news
title	varchar	NO	NO	Title of the news
author	varchar	NO	NO	Author of the news
img	varchar	NO	NO	Image of the news
date	datetime	NO	NO	Date when the news was created
details	text	NO	NO	Details of the news

ForumPost				
	Data type	Primary Key	Allow empty	Function Description
id	integer	YES	NO	user_id
title	varchar	NO	NO	Forum article title
content	text	NO	NO	Forum Article Body
created_at	datetime	NO	NO	Article creation time
update_at	datetime	NO	NO	Article update time
author_id	bright	NO	NO	Article author id
author_name	varchar	NO	NO	Author's name

ForumComment				
	Data type	Primary Key	Allow empty	Function Description
id	integer	YES	NO	user_id
author_name	varchar	NO	NO	Commenter's name
post_id	bigint	NO	NO	Commenter id
content	text	NO	NO	Comment content
created_at	bigint	NO	NO	Comment time
upload_at	datetime	NO	NO	Comment update time
author_id	varchar	NO	NO	Author id

Table 1 - System data sheet design

1.2.2 Application Tier

The application layer (Application Tier) is the core part of the multi-tier architecture system, responsible for processing business logic and interacting with the data layer. In our project, the application layer is mainly responsible for processing requests from users, executing corresponding business logic, and communicating with the data layer to realize data storage and retrieval.

The following are the main components of the application layer:

1. API (Application Programming Interface): API defines the interface for interaction between applications, allowing clients to communicate with backend services. In our project, we use the Django REST framework to create and manage RESTful APIs, providing clients with ways to interact with server-side data.
2. Business logic: Business logic is the core part of processing user requests. It performs corresponding operations based on user input and returns results. In our project, the business logic mainly includes the creation, query, update and deletion of entities such as movies, news, and reviews, as well as the implementation of the recommendation system.
3. Data Access: The application layer needs to communicate with the data layer for data storage and retrieval. In our project, we use Django ORM (Object-Relational Mapping) as a data access tool, which maps tables in the database to Python objects, simplifying the interaction process with the data layer.
4. Permissions and authentication: In order to protect data security and privacy, our application layer needs to implement permissions and authentication functions. We use the authentication and permissions module provided by Django REST framework to ensure that only authenticated users can access protected resources and limit the scope of their operations according to their permissions.

In summary, the application layer plays a vital role in our project, it handles user requests, executes business logic, communicates with the data layer, and implements permissions and authentication functions. By using tools such as Django and Django REST framework, we can easily build a scalable and easy-to-maintain application layer.

1.3 Project Objectives

We conclude 9 project objectives of this system.

PO1. Search

The Movie Finder System must allow user to search for movies based on keywords, that match the movie name, description, or genre, with listed results showing matching movie names, and their latest average rating.

PO2. Movie details

The Movie Finder System must provide users with detailed information about each movie, including the movie name, description, genre, cast, director, latest average rating, and all associated reviews.

PO3. Recommendation

The Movie Finder System must have a recommendation system that suggests movies similar to those the user like.

PO4. Wish list

The Movie Finder System must be able to allow users to create and manage their wish lists, including adding or removing movies.

PO5. Review and rating

The Movie Finder System must provide a review and rating system that allows users to rate and leave reviews for movies they have watched.

PO6. Browse

The Movie Finder System must enable users to browse movies by director and genre.

PO7. Banned list

The Movie Finder System must be able to allow users to create and manage their banned lists, including adding or removing other users.

PO8. News

The Movie Finder System must provide a news section for users to stay up to date on the latest movie- related news.

PO9. Discussion

The Movie Finder System must provide a platform for users to share their thoughts and opinions.

2 Functionalities

2.1 Authority

2.1.1 Register

Users can click the top-right button to navigate to the registration page when they want to register a new account. You can easily join this movie finder system on the registration page by simply inputting some basic information. The registration page is designed with a clear and concise layout, providing you with the required fields such as username, password, and email address. Once completed, click the green signup button will navigate to the login page.

localhost:3002/sign-up

Figure 3 - Registration page

The frontend file address of the registration page is shown below:

Name	Address
SignUpPage.js	/frontend/src/pages/SignUpPage.js

Table 2 - The address of sign up page

The API for fetching signup data is shown below:

API name	URL
user	http://127.0.0.1:8000/user/register/

Table 3 - The URL for sign up page

2.1.2 Login

Users can click the top-right button to navigate to the login page when they want to log in to the movie finder system. Users need to fill in the two input fields with the correct username and password. If the username does not exist or the password is not correct, there will be an alert information popup. Once login succeeds, it will navigate to the home page.



Sign in to Moviehub

Username or email address

Password

New to Moviehub? [Create an account.](#)

Figure 4 - Log in page

The frontend file address of the login page is shown below:

Name	Address
LogInPage.js	/frontend/src/pages/LogInPage.js

Table 4 - The address of login page

The API for fetching login data is shown below:

API name	URL
user	http://127.0.0.1:8000/user/login/

Table 5 - The URL for login page

2.1.3 Logout

Users can log out by clicking on the user icon to open the drop-down menu and then clicking on the logout button to log out. Once the user logout succeeds, the navigation bar will back to the visitor status. Users can not access the wish list banned list and profile after logout.



NFW

Figure 5 - The dropdown menu on the home page

The API for fetching logout data is shown below:

API name	URL
user	http://127.0.0.1:8000/user/logout/

Table 6 - The URL for logout

2.1.4 User Profile

The user profile displays the username, user email, and avatar, as well as all the movie reviews the user posted.

The user can change the user's email and avatar by clicking the edit button.

younian	newemail@gmail.com	Edit
---------	--------------------	------

My reviews:

Comment: I was genuinely entertained by directors Christopher Berkeley and Sam Liu's 2023 animated movie. And if you are a "Batman" fan, then you should check this out. Score: 8 18/04/2023	Edit
Comment: Throughout the film several characters appear and this is the coolest part of the film. The curiosity to know what the characters are like in this universe is really cool Score: 6 18/04/2023	Edit

Figure 6 - User profile page

The frontend file address of the profile page is shown below:

Name	Address
ProfilePage.js	/frontend/src/pages/ProfilePage.js

Table 7 - The address of profile page

The API for fetching profile data is shown below:

API name	URL
user	http://127.0.0.1:8000/user/profile/{username}

Table 8 - The URL for profile page

Once the user clicks the edit button, there will be a popup interface user can input a new email address and avatar.

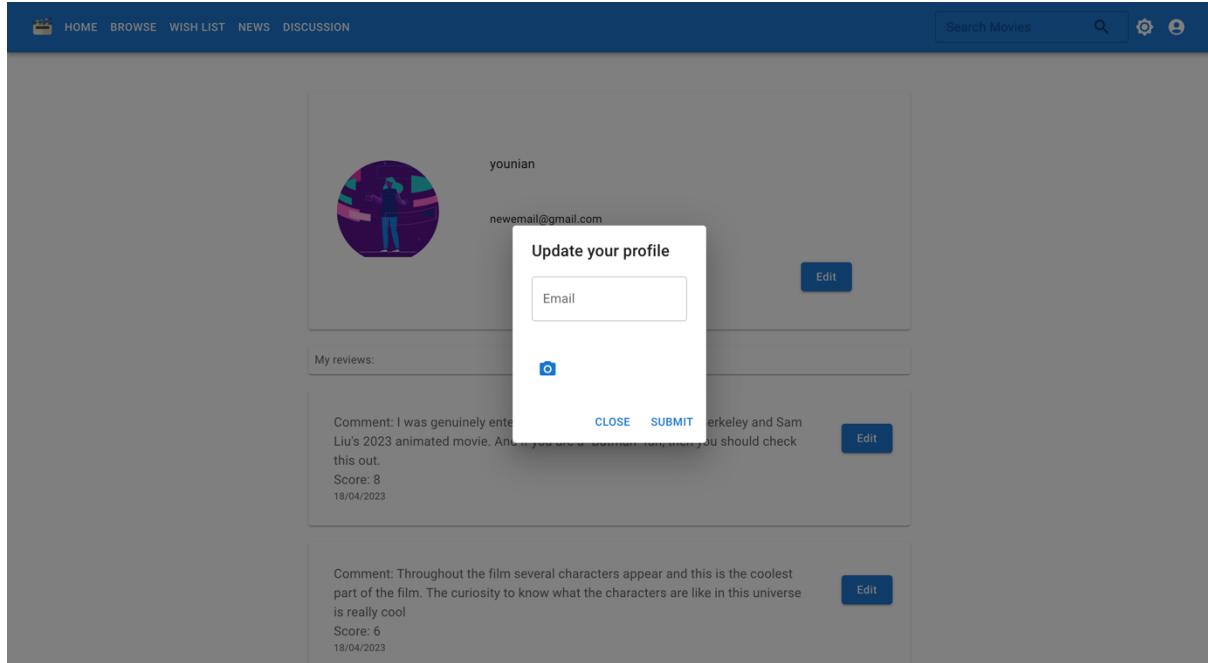


Figure 7 - The interface for editing profile

The users can edit posted comments by clicking the edit button on the comment box.

The screenshot shows a user profile page. At the top, there's a navigation bar with links for HOME, BROWSE, WISH LIST, NEWS, and DISCUSSION. On the right side of the header are search and settings icons. Below the header, the user's profile picture (a person standing in front of a screen) is displayed, along with the name "younian" and the email "newemail@gmail.com". A blue "Edit" button is located at the bottom right of this section. Below this, a section titled "My reviews:" lists two comments. The first comment is: "Comment: I was genuinely entertained by directors Christopher Berkeley and Sam Liu's 2023 animated movie. And if you are a "Batman" fan, then you should check this out." It includes a score of 8 and the date 18/04/2023. The second comment is: "Comment: Throughout the film several characters appear and this is the coolest part of the film. The curiosity to know what the characters are like in this universe is really cool." It includes a score of 6 and the date 18/04/2023. Each comment has its own "Edit" button at the bottom right, with the one for the first comment circled in red.

Figure 8 - The user profile page

Once the user clicks the edit button, there will be a popup interface user can input a new score and comment on the movie. Users can find this edited comment in the review area for that movie on the movie details page.

The screenshot shows the same user profile page as Figure 8, but with a modal dialog box overlaid. The dialog is titled "Edit your comment" and contains a form field with the placeholder "Comment". Above the form, it says "Movie name: Batman: The Doom That Came to Gotham" and "Score". At the bottom of the dialog are "CANCEL" and "SUBMIT" buttons. In the background, the user's profile picture, name, email, and the two reviews from Figure 8 are visible, though slightly dimmed by the overlay.

Figure 9 - The interface of editing comments

2.2 Movie

2.2.1 Home Page

After the users log in, the Home Page is the first page displayed in the screen. The Home Page is not in the project objectives, but it provides some movie information to entertain the user:

1. Big screen posters

The big screen posters show three newest movies in the Home Page. Users can click the blue arrow to switch to different newest movie.



Figure 10 - The big screen posters

2. New movie list

Below the big screen posters is the new movie list. This movie list shows movies by their created time from new to old. Users can click the blue arrow to switch to different movie.

NEW



Figure 11 - The new movie list

3. Top movie list

Below the new movie list is the top movie list. This movie list shows movies by their score from high to low. Users can click the blue arrow to switch to different movie.

TOP



Figure 12 - The top movie list

2.2.2 Search and Browse

Related to: PO1 and PO6

Our team developed a search bar for users to search movies with key words related to the movie name, movie genres, movie directors and so on. Users can type some key words in the search bar, and then click the search icon or type the Enter key in their keyboard to search movies. After clicking the search icon, the website will redirect users to the Browse Page. In this page, related movies will be shown in this page if their information contains the key words. This feature addresses the project objective – Search.

Users can click the browse button on the top of the website to go to the Browse Page. In the Browse Page, there are two selecting boxes: the directors selecting box and the genres selecting box. With these two selecting boxes, users can browse any movies from any directors and genres. This feature addresses the project objective – Browse.

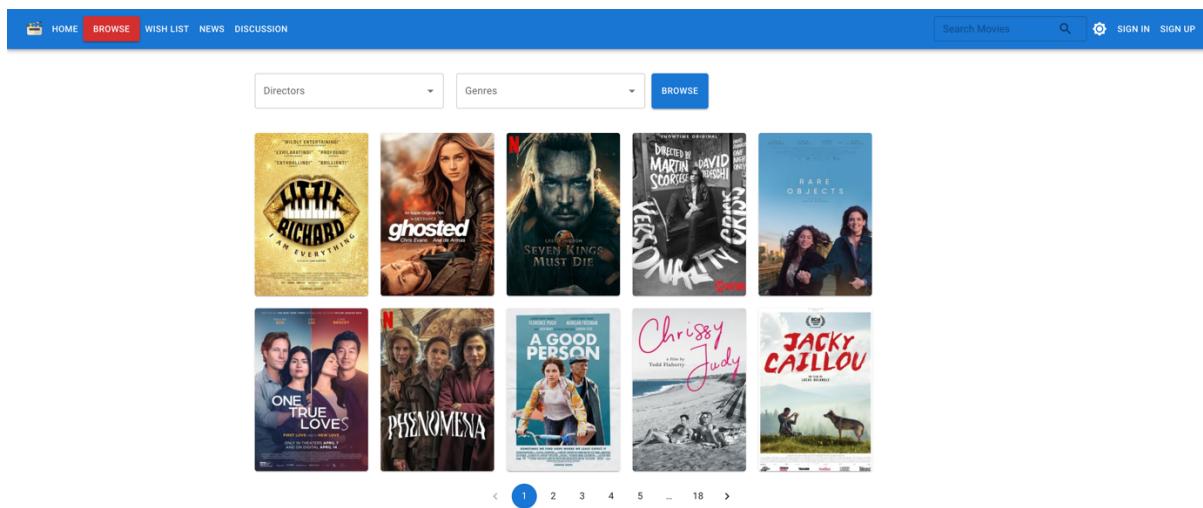


Figure 13 - The Browse Page

2.2.3 Movie Details

Related to: PO2

When the user clicks the picture of the movie shown in the home page or browse page, it will be routed to the movie details page. At the top of this page, the details of this movie will be shown.

The layout of the details part: The poster is displayed on the left and other details information are shown on the right. Through this layout, users can see the poster of the movie more clearly. The atmosphere and information of the movie are delivered to the user visually. Then other details are displayed on the right side in order, and users can see the corresponding information according to their own needs.

Movie Details: A Good Person

Rating:	4.33333333333333
Genre:	Drama
Original Language:	English
Director:	Zach Braff
Producer:	Zach Braff,Pamela Koffler,Christina Piovesan,Florence Pugh,Noah Segal,Christine Vachon
Writer:	Zach Braff
Streaming date:	2023-04-11
Runtime:	2h 9m
Distributor:	United Artists Releasing

[Add to my wish list](#)

Figure 14 - Example of details part

The items of the movie details will be listed in the table below.

Items	Introduction
Poster	The visual representation of the movie, deliver the movie style to attract the audience.
Rating	A number indicates the general perception of the movie.
Genres	Shows the theme and the style of the movie.
Original language	The initial spoken language of the movie's dialogue.
Director	The boss of the movie, responsible for selecting script, selecting actors, shooting and post processing.
Producer	Responsible for managing the budget, schedule and marketing.
Writer	Responsible for the movie script and dialogue.
Streaming date	The date that movie is first play to the public in the cinema.
Run time	The total duration of the movie.
Distributor	The platform that brings the movie to the audience.

Table 9 - The items and brief introduction for movie details

The several categories details for each movie are list above, but in some condition the data maybe NULL or empty, and the page will automatically hide these details and only show exist details. A comparison will be shown below.

Movie Details: Personality Crisis: One Night Only



Rating:	7.75
Genre:	Biography, Documentary, Music
Original Language:	English
Director:	Martin Scorsese,David Tedeschi
Producer:	Margaret Bodde,Martin Scorsese,Justin Wilkes,Sara Bernstein
Streaming date:	2023-04-14
Runtime:	2h 7m

[Add to my wish list](#)

Movie Details: A Good Person



Rating:	4.33333333333333
Genre:	Drama
Original Language:	English
Director:	Zach Braff
Producer:	Zach Braff,Pamela Koffler,Christina Piovesan,Florence Pugh,Noah Segal,Christine Vachon
Writer:	Zach Braff
Streaming date:	2023-04-11
Runtime:	2h 9m
Distributor:	United Artists Releasing

[Add to my wish list](#)

Figure 15 - The comparison to show how the system hide the unknown data

The frontend file address of the movie details is shown below:

Name	Address

DetailsPage.js	/frontend/src/pages/DetailsPage.js
----------------	------------------------------------

Table 10 - The address of movie details

The API for fetching movie details data is shown below:

API name	URL
Details	http://127.0.0.1:8000/doc/movie/{id}/

Table 11 - The URL for movie details

The code for fetching details data is shown below:

```
export async function apiGetMovieID(movieID, wish=false) {
  const init = {
    method: "GET",
    headers: {
      "Content-Type": "application/json",
    },
  };
  // console.log(`${URL}${URL_GET_MOVIEID.replace("id", movieID)}');
  const response = await fetch(
    `${URL}${URL_GET_MOVIEID.replace("id", movieID)}`,
    init
  );
  const data = await response.json();
  // console.log(response);
  // console.log(data);
  if (wish) {
    | return data
  }
  return [response.ok, data];
}
```

Figure 16 - The code to fetch details data

2.2.4 Review and Rating

Related to: PO5

Our team developed the review functionality and the rating functionality in the Movie Details Page.

My Review

Rating

☆☆☆☆☆☆☆☆☆☆

comment

Submit

OtherReview

-  Anonymous User
Rating: 6.5 create time: no recording time

"...doesn't quite know what to do with all the conflicts that made Richard Wayne Penniman tick and holler. But neither did he."
-  Anonymous User
Rating: 7.5 create time: no recording time

In covering all the bases, the film's energy can't help but flag in the later acts. But Cortes has made an impressive music history that restores a "king" to his rightful place in rock royalty
-  younian
Rating: 10 create time: 18/04/2023

good!

Figure 17 - The review and rating functionalities in the Movie Details Page

As the figure shows above, the movie finder system allow user to see other users' review and rating. It also allows users to leave their own review and rating of the selected movie. When users submit their review and rating, the score of the selected movie will be changed due to the rating system. Also, the review system allows users to update their review and rating of the movie. The review and rating functionalities address the project objective – Review and Rating.

2.2.5 Wish list

Related to: PO4

The screenshot shows a navigation bar with links: HOME, BROWSE, WISH LIST (which is highlighted with a red box), NEWS, and DISCUSSION. To the right is a search bar labeled "Search Movies" with a magnifying glass icon, and some user settings icons.

My wish list:

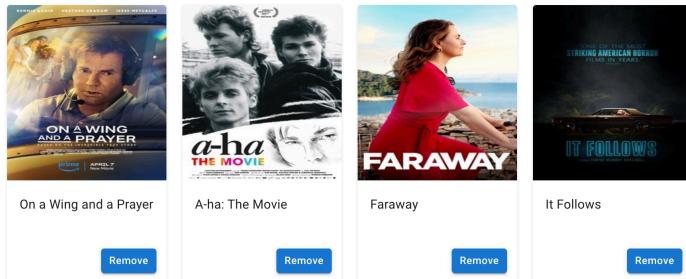


Figure 18 - The wish list page

The frontend file address of the wishlist page is shown below:

Name	Address
WishListPage.js	/frontend/src/pages/WishListPage.js

Table 12 - The address of wish list page

The API for fetching wishlist data is shown below:

API name	URL
user	http://127.0.0.1:8000/user/waitlist/

Table 13 - The URL for wishlist page



Recommendation:

Figure 19 - The details page

2.2.6 Banned list

Related to: PO7

Users can view their own banned list by clicking the banned list button in the drop-down menu of the user icon.



NFW

Figure 20 - The dropdown menu on the main page

On the banned list page, the user can choose to remove the banned user by clicking the remove button. If the user wants to view the banned user's profile, they can simply click the username, and it will navigate to the banned user's profile.

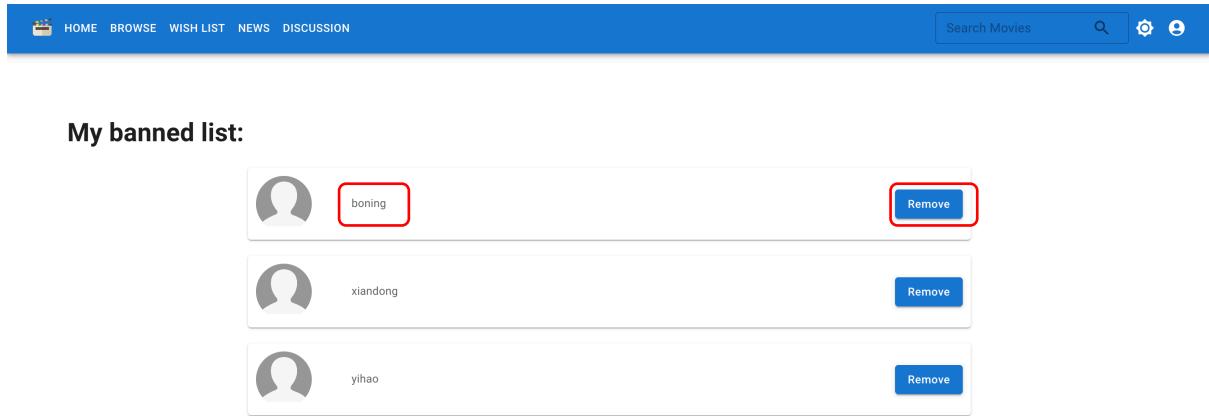


Figure 21 - The banned list page

The frontend file address of the banned list page is shown below:

Name	Address
BannedListPage.js	/frontend/src/pages/BannedListPage.js

Table 14 - The address of banned list page

The API for fetching banned user data is shown below:

API name	URL
user	http://127.0.0.1:8000/user/ban/

Table 15 - The URL for banned list page

Logged-in users can view other user's profile page, which displays all the movie reviews posted by that user. If the user doesn't like boning's comments, the user can block boning by clicking the add to banned list button. Also, the user can check the details of boning's

comment by clicking the comment field.

The screenshot shows a movie profile page. At the top, there is a navigation bar with links for HOME, BROWSE, WISH LIST, NEWS, and DISCUSSION. On the right side of the navigation bar are search and settings icons. Below the navigation bar is a user profile section featuring a circular profile picture of a building labeled 'UNSW'. To the right of the profile picture is the username 'boning'. Underneath the profile picture are two buttons: 'Wish list' and 'Add to banned list', with 'Add to banned list' being highlighted with a red border. Below this section is a 'Reviews:' heading followed by a comment card. The comment card contains the following text:
Comment: This movie tries to be too much like Gotham by Gaslight and it just doesn't work. The story is not very interesting and just throws in random things to introduce characters that reference the comics.
Score: 4
18/04/2023

Figure 22 - The profile of boning

Once the user adds boning to the banned list, all boning's comments will be hidden as the below picture shows.

This screenshot shows the same movie profile page for 'boning' as in Figure 22, but after the user has been banned. The interface elements are identical to Figure 22, including the navigation bar, user profile section, and the first comment card. However, the second comment card, which was visible in Figure 22, is now completely hidden, replaced by a single line of text: 'Comments have been hidden'.

Figure 23 - The profile page after banned

Logged-in users can view Boning's wishlist by clicking the wishlist button. Users can add movies they like to their own wishlist or remove disliked movies from their wishlist.

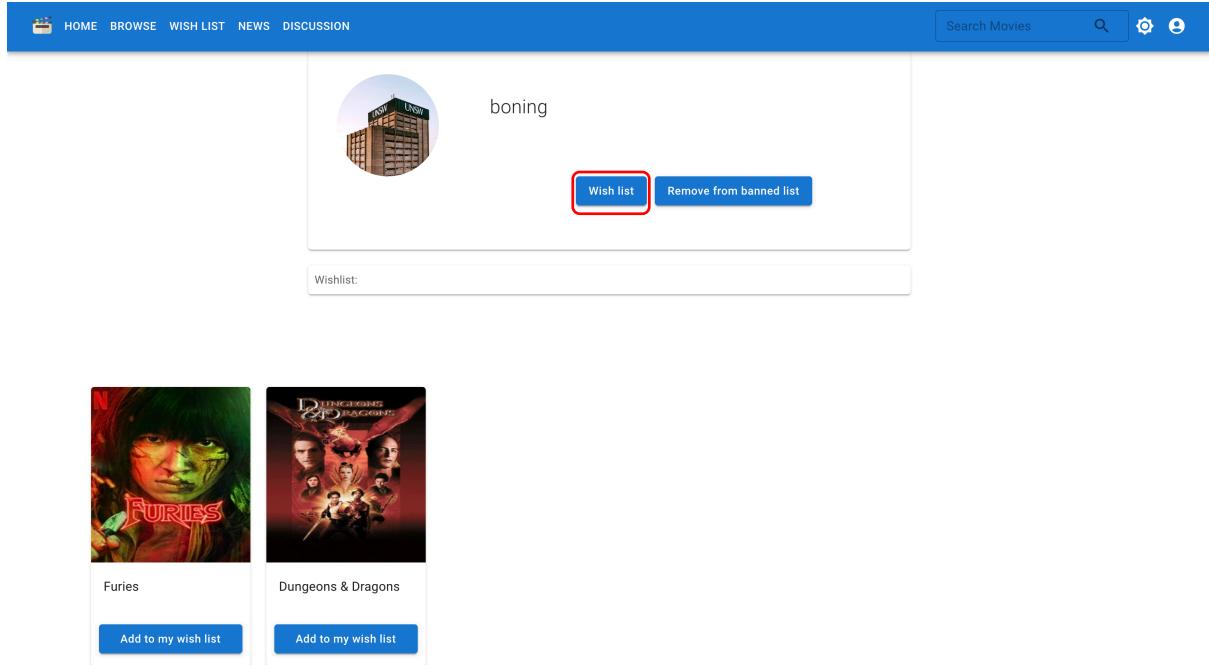


Figure 24 - The profile page of boning after showing his wishlist

2.2.7 Recommendation

Related to: PO3

In the details page, the recommendation part is displayed below the movie details. In this part, the movie finder system will recommend some similar movies for the user that they may like. And the system uses an algorithm to generate 15 recommended movies for the user.

The layout of the recommendation part: 15 recommended movies will be displayed in a line, and there are two direction buttons for user to switch. The movie in far left is the most recommended one and as going to right, recommendation decrease.

Recommendation:



My Review

Figure 25 - Example of recommendation part

The recommendation algorithm is:

This website has a function of personalized recommendation. When a user views the movie details, the system will get the movies in the user's wish list, select the movies with the same genre as the movies in the wish list, and then recommend them to the user in the order from highest to lowest movie rating. If the movies in the user's wish list contain multiple genres, the system will randomly select several genre movies and then recommend them to the user.

in the order of highest to lowest movie rating. When the user browses the site as a visitor or when there are no movies in the wish list, the system will randomly select 15 movies to recommend to the user.

```

def recommend_movies(user_id):
    # 1. Gets the user's UserProfile
    user_profile = UserProfile.objects.filter(user_id=user_id).first()
    if not user_profile:
        return MoviePost.objects.order_by("?")[:15]

    # 2. Get the movie in the user's waitlist
    waitlisted_movies = user_profile.waitlist.all()
    if not waitlisted_movies:
        return MoviePost.objects.order_by("?")[:15]

    pk = []
    genres = []
    for movie in waitlisted_movies:
        pk.append(movie.pk)

    _genres = [i.strip() for i in movie.genres.split(",")]
    genres.extend(_genres)

    # 3. Get all movies of the same type as the movie in the waitlisted_movies
    q_list = [Q(genres__icontains=genre) for genre in genres]
    similar_movies = MoviePost.objects.filter(reduce(lambda x, y: x | y, q_list)).exclude(pk__in=pk).distinct()

    # 4. Calculate a rating for each movie
    movies_with_score = []
    movies_pk = []
    for movie in similar_movies:
        comments = MovieComment.objects.filter(movie=movie).all()
        if comments:
            # Calculate the average rating
            score = sum(comment.score for comment in comments)/len(comments)
            movies_with_score.append((movie, score))
            movies_pk.append(movie.pk)

    # 5. Sort by average rating from highest to lowest
    if movies_with_score:
        movies_with_score = sorted(movies_with_score, key=lambda x: x[1], reverse=True)
        _count = 15 - len(movies_with_score)
        # Not enough for 15 mandatory completions
        if _count > 0:
            movies_with_score.extend([(movie, 0) for movie in MoviePost.objects.exclude(pk__in=movies_pk).order_by("?")[:_count]])
        else:
            # If there are no movies, 15 movies are returned randomly
            movies_with_score = [(movie, 0) for movie in MoviePost.objects.all()]

    # 6. Returns the MoviePost object querysets
    return [movie[0] for movie in movies_with_score[:15]]

```

Figure 26 - Implementation of the recommendation function

The recommendation is also in the details' JavaScript file since it is a part of the details page, and the frontend file address is shown below:

Name	Address
DetailsPage.js	/frontend/src/pages/DetailsPage.js

Table 16 - The address of recommendation part

How to interact the frontend and backend: When the user is in the details page, the frontend will get the movie id. And then send it to the backend, the backend will search the information of this movie in the database and apply the recommendation algorithm. Then it will outcomes a list of movies and then the frontend could get that and display.

The API for fetching recommended movie is shown below:

API name	URL
Details	http://127.0.0.1:8000/doc/movie/reco_movie/

Table 17 - The URL for recommendation part

The code for fetching recommendation data is shown below:

```

export async function apiGetRecommendationMovies(token) {
  const init = {
    method: "GET",
    headers: {
      "Content-Type": "application/json",
      Authorization: `Token ${token}`,
    },
  };
  const response = await fetch(`${URL}${URL_GET_RECOMMENDATION_MOVIES}`, init);
  const data = await response.json();
  // console.log(response);
  // console.log(data);
  return [response.ok, data];
}

```

Figure 27 - The code to fetch recommendation data

2.3 News

Related to: PO8

The system has an news page which is routed by clicking the news button in the navigation bar and it shows the top headlines. Then when user clicks one of the news, the page will show the news details in a pop-up window.

2.3.1 News list

The layout of the news list: The news page will show the news title on the left and the release date on the right. The news are displayed by date in descending order, which means the most recent news will be shown in the top of this page. Each page will show 8 news since display too much news in one page may make the users feel visually fatigued and unable to find the news, they are interested in. To handle this problem, switch page button is provided in the bottom and the user can click it to switch the page.

Top Headlines

Mia Goth Joins Marvel's Blade, and More Movie News	April 15, 2023
The Game of Thrones Universe: Spin-Offs, Prequels, and Animated Series in Development, in Production, and Ordered to Series	April 15, 2023
Edgar Ramirez Enjoys Playing the Dark Comedy in Florida Man	April 15, 2023
The Mandalorian: Heirs to the Empire Emerge in the Penultimate Episode of Season 3	April 15, 2023
Morgan Freeman's Five Favorite Films	April 15, 2023
Everything We Know About Blue Beetle	April 15, 2023
What to Watch This Week: Renfield, Barry, and More	April 15, 2023
The Marvelous Mrs. Maisel Star Rachel Brosnahan, Creator Amy Sherman-Palladino on Midge's Sacrifices in Her Rise to Fame	April 15, 2023

[Previous page] Page 1 of 4 [Next page]

Figure 28 - Example of news list

The frontend file address of the news list is shown below:

Name	Address
NewsPage.js	/frontend/src/pages/NewsPage.js

Table 18 - The address of news list

The API for fetching news list data is shown below:

API name	URL
News	http://127.0.0.1:8000/doc/movie/news/

Table 19 - The URL for news list

The code for fetching news list data is shown below:

```
export async function apiGetNews() {
  const init = {
    method: "GET",
    headers: {
      "Content-Type": "application/json",
    },
  };
  const response = await fetch(`#${URL}${URL_GET_NEWS}`, init);
  const data = await response.json();
  // console.log(response);
  // console.log(data);
  return [response.ok, data];
}
```

Figure 29 - The code to fetch news list data

Above is the first part of the news page, and another part is the news details. When the user clicks one of the news, a window pops up on this page to show the news details, which will be introduced later. Moreover, to make it easier for users to see which news they are currently choosing, I added an effect, shading the background color around the selected news. When the user hovers the cursor in one of the news. The effect will be shown like the following figure.



Figure 30 - The effect of selecting a news

2.3.2 News details

The layout of the news details: The news details are displayed in a pop-up window for users' convenience since they may just want to browse the content briefly. The pop-up window contains the news title, author, release date, and the content of the news. The user can scroll down to see all the details, and at the bottom of this window, there is a close button for user to get back to the news list. Also, the user can click anywhere besides the pop-up window to close the window.

Top Headlines

- Mia Goth Joins Marvel's Blade, and More Movie News
- The Game of Thrones Universe: Spin-Offs, Prequels, and Animated Series
- Edgar Ramirez Enjoys Playing the Dark Comedy in Florida Man
- The Mandalorian: Heirs to the Empire Emerge in the Penultimate Episode
- Morgan Freeman's Five Favorite Films
- Everything We Know About Blue Beetle
- What to Watch This Week: Renfield, Barry, and More
- The Marvelous Mrs. Maisel Star Rachel Brosnahan, Creator Amy Sherman-Palladino on Midge's Sacrifices in Her Rise to Fame**

Author: Whitney Friedlander
Date: April 15, 2023



The Marvelous Mrs. Maisel Star Rachel Brosnahan, Creator Amy Sherman-Palladino on Midge's Sacrifices in Her Rise to Fame

Miriam "Midge" Maisel, Rachel Brosnahan's perfectly poised heroine from The Marvelous Mrs. Maisel, came up in the era of "three before 30." The phrase, or at least how the Emmy-winning Prime Video series deploys it, is meant to refer to children: As in, Midge was raised to get married before, during, or in lieu of college and have three children before she turned 30. That's what her parents, Abe (Tony Shalhoub) and Rose (Marin Hinkle) expected of her, and that's what she did when she married Joel (Michael Zegen). (Photo by Philippe Antonello/Prime Video) She was two-thirds of the way through that kid goal when he abruptly left her on Yom Kippur, a

April 15, 2023

Figure 31 - Example of news details

Top Headlines

- Mia Goth Joins Marvel's Blade, and More Movie News
- The Game of Thrones Universe: Spin-Offs, Prequels, and Animated Series
- Edgar Ramirez Enjoys Playing the Dark Comedy in Florida Man
- The Mandalorian: Heirs to the Empire Emerge in the Penultimate Episode
- Morgan Freeman's Five Favorite Films
- Everything We Know About Blue Beetle
- What to Watch This Week: Renfield, Barry, and More
- The Marvelous Mrs. Maisel Star Rachel Brosnahan, Creator Amy Sherman-Palladino on Midge's Sacrifices in Her Rise to Fame**

Author: Whitney Friedlander
Date: April 15, 2023

something like that because that's how you can talk to somebody that you have that sort of relationship with." (Photo by Philippe Antonello/Prime) This closeness is something that the Sherman-Palladinos also appreciate in their own lives, especially as they've seen so many of the actors they've worked with achieve their own levels of fame. The *Maisel* writers have a habit of bringing in actors they've worked with on their other series, such as *Gilmore Girls* alum Milo Ventimiglia and Kelly Bishop and *Bunheads* actress Bailey De Young. This season sees more familiar faces from the Sherman-Palladino-verse, including Sean Gunn, Danny Strong, and Sutton Foster. Palladino said he specifically wrote the parts that Gunn and Strong play this season. He knew they were both busy with other projects, so he needed something that would more substantial than a three-line appearance. "You save those special parts for those special guys and you bring you bring you bring your buddies back who you know can do it," Palladino said and continued that "Some things come very quick and easy to us when we're writing. And those never came quick and easy." (Photo by Philippe Antonello/Prime) And there is a special kind of actor who can handle this kind of material. *Maisel*, like other shows associated with the couples, is known for a particular kind of rapid-fire cadence and suave demeanor. But it takes a lot of work to make Midge and company sound so polished. Palladino said it "takes a village" of stand-up comics to offer input for Midge's seemingly off-the-cuff monologues and stand-up routines. "They have they have the rhythms kind of in their DNA that we don't quite have the experience for," he continued. She added that "especially this year, where anytime she was on stage was so important because it's one of the last times we're going to see her on stage, I think it just carried even more angst and weight." Just as fame is an illusion, so is the idea that it's easy to be this marvelous. 92% The Marvelous Mrs. Maisel: Season 5 (2023) premieres its first three episodes on April 14 on Prime Video. Subsequent episodes roll out weekly. On an Apple device? Follow Rotten Tomatoes on Apple News.

[Close]

April 15, 2023

Figure 32 - Example of news details with the "close" button

The frontend file address of the news details is same as news list, and it is shown below:

Name	Address
NewsPage.js	/frontend/src/pages/NewsPage.js

Table 20 - The address of news details

The API for fetching news details is shown below:

API name	URL
News details	http://127.0.0.1:8000/doc/movie/news/{id}/

Table 21 - The URL for news details

The code for fetching news details data is shown below:

```

export async function apiGetNewsID(newsID) {
  const init = {
    method: "GET",
    headers: {
      "Content-Type": "application/json",
    },
  };
  // console.log(`[${URL}${URL_GET_MOVIEID.replace("id", movieID)})}`);
  const response = await fetch(
    `${URL}${URL_GET_NEWS_BY_ID.replace("id", newsID)})`,
    init
  );
  const data = await response.json();
  // console.log(response);
  // console.log(data);
  return [response.ok, data];
}

```

Figure 33 - The code to fetch news details data

2.4 Discussion

Related to: PO9

The system has a discussion page which is routed by clicking the button in the navigation bar to show some thoughts of the user for some specific movies. And it is like a forum allowing users to share their thoughts for some specific movies. And first an article list will be displayed in this page, and there is a button for user to create their own forum in the lower right corner of the discussion page. When the user clicks one of the forums, they will be routed to a new page to write some comments and ideas.

2.4.1 Discussion Page

The layout of the articles list: The page shows a lot of articles, and they are sort by the release time in descending order. For each article, the box shows the publisher and release time, as well as the title and their idea. In the bottom of this page, there are two buttons for user to switch the page and see the earlier article.

The screenshot displays a discussion forum interface with four visible posts:

- yihao** (April 18, 2023 at 02:37 PM): The opening was somewhat similar, but the rest?
Damn. Calling this trash doesn't do it justice. This was bad enough to be something from Marvel's animated films which are just as bad.
- younian** (April 18, 2023 at 02:36 PM): Terrible. Did they read the comic?
They brought no emotion to the voices, nothing. Did the writers read the comic?
- xiaodong** (April 18, 2023 at 02:29 PM): Questions about DCAU
First, this is without a doubt, the worst DCAU film ever. Other than the reliable John DiMaggio, who were these people?
- shaoming** (April 18, 2023 at 02:28 PM): Batman discussion
Without knowing more than how these versions of the characters came about, they seem detached from the story. Some of the smaller characters are also used more like a narrative device than a character.

At the bottom of the page, there are navigation links: Previous page, Page 1 of 2, Next page, and a blue 'Create a forum' button.

Figure 34 - Example of article list

The frontend file address of the articles list is shown below:

Name	Address
------	---------

DiscussionPage.js	/frontend/src/pages/DiscussionPage.js
-------------------	---------------------------------------

Table 22 - The address of the articles list

The API for fetching the articles list is shown below:

API name	URL
Forum list	http://127.0.0.1:8000/doc/forum/posts/

Table 23 - The URL for the articles list

The code for fetching the articles list data is shown below:

```
export async function apiGetAllDiscussions(token) {
  const init = {
    method: "GET",
    headers: {
      "Content-Type": "application/json",
      Authorization: `Token ${token}`,
    },
  };
  const response = await fetch(`${URL}${URL_GET_ALL_DISCUSSIONS}`, init);
  const data = await response.json();
  // console.log(response);
  // console.log(data);
  return [response.ok, data];
}
```

Figure 35 - The code to fetch the articles list data

2.4.2 Discussion Editing Page

Users can create a discussion by clicking the bottom-right button on the discussion page.

The screenshot shows a discussion page with the following comments:

- yihao** (April 18, 2023 at 02:37 PM): **The opening was somewhat similar, but the rest?**
Damn. Calling this trash doesn't do it justice. This was bad enough to be something from Marvel's animated films which are just as bad.
- younian** (April 18, 2023 at 02:36 PM): **Terrible. Did they read the comic?**
They brought no emotion to the voices, nothing. Did the writers read the comic?
- xiandong** (April 18, 2023 at 02:29 PM): **Questions about DCAU**
First, this is without a doubt, the worst DCAU film ever. Other than the reliable John DiMaggio, who were these people?
- shaoming** (April 18, 2023 at 02:28 PM): **Batman discussion**

At the top left, there is a small text "localhost:3002/discussion". At the top right, there is a search bar with "Search Movies" and a magnifying glass icon, along with other icons for settings and user profile.

Figure 36 - The discussion page

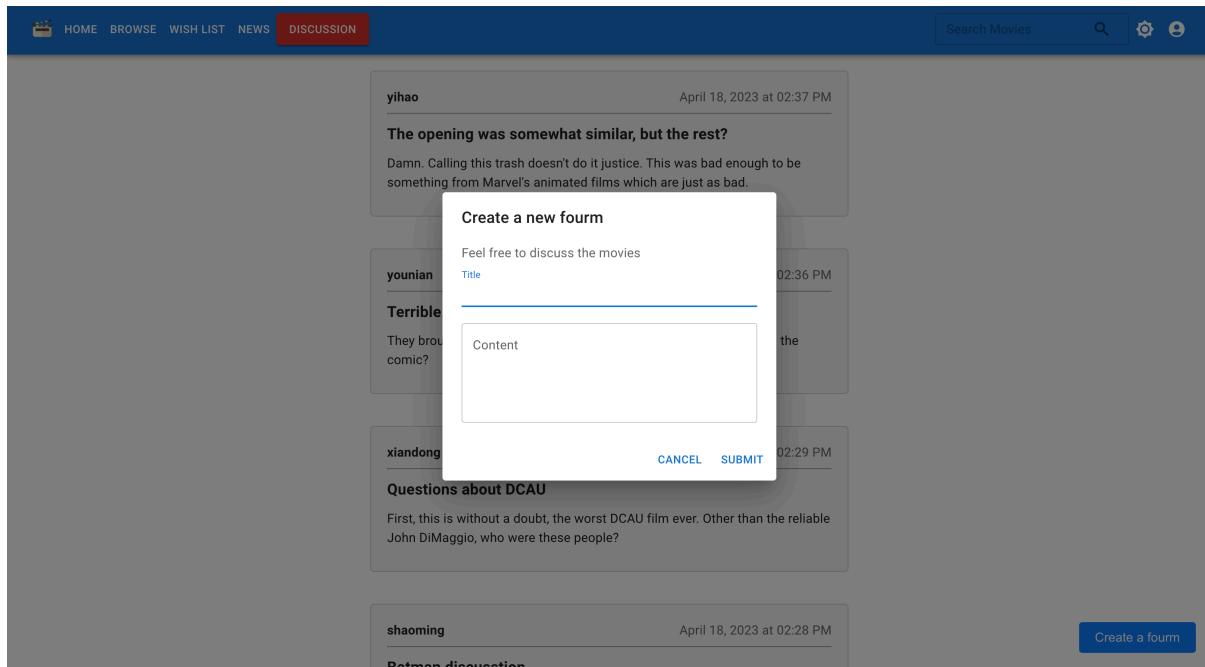


Figure 37 - The interface of creating a forum

2.4.3 Comments of Discussion

Related to PO9

Our team developed a comments of discussion feature to allow users to comment a specific discussion of movies. Users can click one of the discussions to navigate to the Discussion Details Page. Below the content of the discussion is the comment functionality. Users can leave comments to a specific discussion to express their opinions. This comment functionality is one part of the project objective – Discussion.

Write your comment:

submit

Comments:



boning

create time: 20/04/2023

Yes, you are right.

Figure 38 - The comment functionality of the Discussion

3 Third-party functionalities

3.1 Django REST Framework (DRF)

Django REST Framework is a powerful and flexible Web API framework for building Web APIs for a variety of clients. In your project, DRF is mainly used to create and manage API views, serializers, filters, etc. For example:

`MovieListAPIView`, `MovieDetailAPIView`, `MovieCommentListAPIView`, etc. are DRF-based view classes for processing API requests and returning appropriate responses.

`MovieCommentSerializer`, `MovieListSerializer`, `MoviePostSerializer`, etc. are serializer classes used to convert data models into JSON format for use in API responses.

3.2 django-filter

`django-filter` is a library for filtering operations on Django querysets. In your project, it is used in conjunction with DRF to implement the filtering function of the API view. For example:

In `MovieListAPIView`, `DjangoFilterBackend` is used as the filtering backend, allowing to filter movies based on name, genre, director and original language.

3.3 React

React is one of the most popular JavaScript frameworks in the frontend area. It can let us build user interfaces out of individual pieces called components [1].

Our team choose this framework to build the Movie Finder System. For example, the files such as `HomePage.js`, `BrowsePage.js` and other components are using the React framework to write the codes of the pages.

3.4 React Router

`React Router` is a popular library for routing in `React.js` applications. `React Router` allows users to navigate between different pages or views of an application without refreshing the page, making it easy to create complex multi-page web applications [2].

Our team choose this library to build the router of the website. The config of the router is defined in the `router.js` file.

3.5 MUI

`MUI` (Material-UI) is a popular React component library that provides pre-designed, customizable, and reusable UI components. With MUI, developers can quickly and easily build modern, responsive web applications that are visually appealing and user-friendly [3]. Our team choose this React component library to build the UI of this website. For example, the files such as `Header.js`, `Movie.js` and other components are using the MUI components to build the UI.

3.6 Swiper

Swiper is the most modern free mobile touch slider with hardware accelerated transitions and amazing native behaviour. It is intended to be used in mobile websites, mobile web apps, and mobile native/hybrid apps [4].

Our team use this library to build the sliders of the big screen posters, the new movie list, the top movie list, and the recommendation movie list in the Movie.js file.

3.7 Jest

Jest is a delightful JavaScript Testing Framework with a focus on simplicity [5].

Our team use this testing library to write the unit tests such as movie.test.js and discussion.test.js. With the unit tests, we can ensure that this Movie Finder System is robust.

4 Implementation challenges

4.1 Front-end Part

4.1.1 Challenge of Jest mocks global.fetch return undefined

When I mock the global.fetch, it returns undefined which means I fail to mock this function. To solve this problem, I search for this issue in Stackoverflow and find out that the problem is caused by Create-React-App has resetMocks set to true. This means that my mock is reset before the test runs. To solve this issue, I move the mock inside my test function, and it work.

4.1.2 Challenge of updating React components in Jest

When I try to render some React components in Jest, there are some issues:

```
console.error
Warning: An update to PageLogin inside a test was not wrapped in act(...).

When testing, code that causes React state updates should be wrapped into act(...):

act(() => {
  /* fire events that update state */
});
/* assert on the output */

This ensures that you're testing the behavior the user would see in the browser. Learn more at https://reactjs.org/link/wrap-tests-with-it
at PageLogin

 38 |     } else {
 39 |       console.log(data);
> 40 |       setOpenNotice(true);
|       ^
 41 |       setNotice({
 42 |         level: NOTICE_LEVEL.ERROR,
 43 |         message: data.error,
 44 |
 45 |       })
 46 |     }
 47 |   );
 48 | }

at printWarning (node_modules/react-dom/cjs/react-dom.development.js:67:30)
at error (node_modules/react-dom/cjs/react-dom.development.js:43:5)
at warnIfNotCurrentlyActingUpdatesInDEV (node_modules/react-dom/cjs/react-dom.development.js:24064:9)
at setOpenNotice (node_modules/react-dom/cjs/react-dom.development.js:16135:9)
at handleSubmit (src/pages/PageLogin.jsx:40:9)
```

Figure 39 - The error of rendering component in the unit testing

To solve this issue, I need to use the async/await, like:

```
await act ( async () => render(<YourComponent />));
```

4.2 Back-end Part

4.2.1.1 Write a web crawler to collect rotten tomatoes's movie information data, add download movie poster function.

obstacles:

The poster is not directly stored on the website with the img tag, so it cannot be downloaded directly

4.2.1.2 overcome:

Obtain the request link of the poster image by capturing the packet, and obtain the poster by using request.get() and the link

4.2.2.1 Crawler add collect movie information data function.

obstacles:

Movie information is stored in different tags, and extracting it directly will cause the text to be very confusing due to extra spaces, line breaks, and tabs.

4.2.2.2 overcome:

Process text through regularization and remove redundant symbols

4.2.3.1 Crawler add collect movie's cast information function.

obstacles:

The names stored in the actor information are all NA, and the actor information cannot be obtained directly, for example:

```
{'@type': 'Person', 'name': 'NA', 'sameAs':  
'https://www.rottentomatoes.com/celebrity/1168177-ben_whishaw', 'image':  
'https://resizing.flixster.com/SkJDWYbLIELQyVBd4srJEZNn1iQ=/100x120/v2/https://flxt.msimg.com/assets/275095\_v9\_bc.jpg'}
```

4.2.3.2 overcome:

Use regex to get the actor's name by processing the link of the actor's page, and crop the actor's name with a unique ID

4.2.4.1 movie information data

obstacles:

Different movies have different information. Some animated movies have no author, some musicals have no director, and some have no streaming release time. It is difficult to divide the string by ordinary methods.

4.2.5.2 overcome:

Use different beautifulsoup tag search methods to search, and then process the corresponding string

4.2.5.1 Add User Register function

obstacles:

"Bad Request: /userprofile/register/"

"KeyError: 'username'", send post request and register data, then server return this error.

"{"password": ["This field is required."]}"

4.2.5.2 overcome:

"KeyError: 'username'" :

1. has try fix view.py->register() and serializer.py->create() from username=validated_data['user']['username'] to username=validated_data['username'], delete user, but didn't solve this problem
2. try to change model, add user field, but it need post user string.
3. try to change request json structure:

```
{  
    "user": {  
        "username": "testuser",  
        "email": "testuser@example.com",  
        "password": "testpassword123",  
        "first_name": "Test",  
        "last_name": "User"
```

```

    },
    "date_of_birth": "1990-01-01",
    "country": "Australia",
    "city": "Sydney"
}
but still because of user
Finally, rewrite view.py->register() and serializer.py->create()

```

4.2.6.1 Add Swagger API documents:

obstacles:

1. Error:'AutoSchema' object has no attribute 'get_link'
2. still have other errors
3. drf-yasg didn't support the latest version drf

4.2.6.2 overcome:

1. Using an older version of django-rest-swagger, the attribute 'get_link' has been renamed to "get_operation"
2. The author of django-rest-swagger has given up updating this module, which has been deprecated in the django official guide, so use drf-yasg
3. use the drf-yasg2 module

4.2.7.1 Finish python data crawler to collect rottentomatoes movie data and comments

obstacles:

1. Hard to input data in django ORM

4.2.7.2 overcome:

1. try to use scrapy
2. try to use scrapy-djangoitems
3. abandon to use ORM, use other method, save to json

4.2.8.1 Add movie data

obstacles:

1. Hard to input data in django ORM

4.2.8.2 overcome:

1. use sqlite3 module connect to database and input data

4.2.9.1 Rewrite UserProfile & Movie models

obstacles:

1. Models rasie AttributeError: 'str' object has no attribute '_meta'
2. Models rasie SystemCheckError: System check identified some issues:

ERRORS:

userprofile.Waitlist.email: (fields.E303) Reverse query name for 'userprofile.Waitlist.email' clashes with field name 'userprofile.UserProfile.waitlist'.

HINT: Rename field 'userprofile.UserProfile.waitlist', or add/change a related_name argument to the definition for field 'userprofile.Waitlist.email'.

4.2.9.2 overcome:

1. The error is in the `through_fields` parameter of the `waitlist` field in the `userprofile/models.py` file. Between `MoviePost` and `UserProfile`, `through_fields` should be a tuple where the first element should be a field from the `UserProfile` model, not `email`.
2. According to the error message, the `email` field of the `Waitlist` model defines the same reverse query name as the `waitlist` field of the `UserProfile` model, so the `related_name` parameter of the `waitlist` field needs to be changed to another name to avoid name conflicts.

4.2.10.1 Register password different from login password

Obstacle:

Used this password when registering, but the login shows that the password is incorrect

4.2.10.2 overcome:

Because at the time of registration, the password is encrypted using a hash function. Django uses the PBKDF2 password hashing algorithm by default, which hashes the original password multiple times and adds salt to it and stores it in the database. Therefore, when you use the `check_password` function for password comparison, you need to pass in the original password instead of the encrypted password. The `check_password` function will return True if the original password is correct.

So give up storing passwords in `userprofile`, and directly use the `auth_user` built-in module to store uid and password

5 User manual

We have uploaded our final submission to our team's GitHub classroom account on time.

The link to our submission on GitHub is:

<https://github.com/unsw-cse-comp3900-9900-23T1/capstone-project-9900h14akuangbiao>

We setup, build, and run this system on a **virtual machine** based on the **Lubuntu 20.4.1 LTS** virtual machine image [here](#).

The VM image should be imported and run using:

VirtualBox 6.1.22 or later (for Mac/Windows/Linux) available [here](#).

Username: **lubuntu**

Password: **lubuntu**

(To become root, use **sudo su**)

5.1 Setup the code environment

Before you can build and run the Movie Finder System, you need to make sure that you have installed the following software/tools in the machine.

0. Firstly, update the **lubuntu**

```
$ sudo apt update  
$ sudo apt upgrade  
$ sudo apt install -y curl
```

1. Python

First, make sure you have Python installed. We recommend using Python 3.10.11 or higher. You can download Python from the official Python website:

<https://www.python.org/downloads/>

```
$ sudo apt install python3-pip  
$ python3 --version  
Python3.8.10
```

2. Node

Notice: the version of the node must larger than 17.

```
$ curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash -  
$ sudo apt install -y nodejs  
$ node --version  
v18.16.0
```

5.2 Building the system

After install all needed software and tools, you can now build the Movie Finder System by following the code below:

1. Download the system code from the website: <https://github.com/unsw-cse-comp3900-9900-23T1/capstone-project-9900h14akuangbiao>

Extract the zip into a directory capstone-project-9900h14akuangbiao:

2.

```
$ unzip capstone-project-9900h14akuangbiao-main.zip
```

3. cd into to the directory capstone-project-9900h14akuangbiao

```
$ cd capstone-project-9900h14akuangbiao/
```

4. Install the dependences of the backend:

```
$ pip3 install -r requirements.txt
```

5. Install the dependences of the frontend:

```
$ cd frontend
```

```
$ npm install
```

5.3 Running the system

After building the system, now you can run the Movie Finder System!

1. Run the backend

At the system root directory capstone-project-9900h14akuangbiao/:

```
$ python3 manage.py runserver
```

2. Run the frontend

At the system root directory capstone-project-9900h14akuangbiao/:

```
$ npm start
```

5.4 Using the system

5.4.1 Login Page, Register Page and Home Page

1. Register, Login, Logout

The top right corner of the home page provides a user login and registration function. For new users, you can register your account in this website and log in. After logging in, the same position of the navigation bar will be switched to logout button.

2. Home Page

After login, the Home page are shown to users. Users can see a big screen posters, a new movie list and top movie list.

5.4.2 BROWSE

1. Search movies by keywords

Searching movie information by keywords is a very important function in APP. The website supports searching movie information by keywords, including movie cast list, movie shooting country, movie director list, movie category, movie classification and rating, movie release date, movie title, etc.

2. Filter movies by tags

Enter the movie tags you are interested in in the filter box or select them in the drop-down menu, such as "Romance", "Action", "Suspense", "Thriller", etc, "Suspense", then click the search button. You can further filter the results based on factors such as director and actors. You will then see a list of movies related to that tag.

5.4.3 Movie details page

- 1. Show movie information**

Select one of the movies and click on its name to view detailed information about the movie, including the movie cover title and director, cast, plot summary, release date, duration, rating, and audience reviews.

- 2. Movie Collection**

Users can add or un-bookmark movies in the detail screen and view the bookmarked movies in the wishlist page to easily find and watch them in the future.

- 3. Recommend movies of the same genre or director**

In the movie details page, some movies of the same genre will be provided as recommendations to help users better discover the content they are interested in. If you like a movie, you can add it to your favorites, and the system will recommend other movies worth knowing according to the movie genre.

- 4. Review and Rating**

You can enter messages in the message board to exchange ideas, review movies, express opinions and rate the current movie.

5.4.4 WISHLIST

- 1. Favourite/unfavorite movies**

Here users can add their favourite movies to their wish list by selecting them in the details page. After you add them, they are automatically saved, and you can see the movies you have added to your wish list in your wish list after you log in next time. After a while, you can delete your favourite movies if you don't like them.

5.4.5 NEWS

- 1. News Browse**

You can view the list of news information and news content information.

5.4.6 DISCUSSION

- 1. Post Browsing**

Any type of user who enters the site has the right to browse posts. If you see a topic you are interested in, you need to register as a member of the site before you can post your opinion.

- 2. Posting**

After logging in, users can post posts.

- 3. Replying to posts**

In this system, except for visitors, all users with permission can reply to posts. It is to provide a discussion environment for people, so the post reply function is used to express their views on the same topic, so that people can communicate better.

- 4. Post management module**

In this system, moderators and administrators have the authority to manage posts, which mainly includes viewing post details, editing posts and deleting posts. The difference is that moderators only have the right to manage posts in the section they belong to, while administrators can manage all posts in the forum.

Reference

- [1] Meta. (2023). React - A JavaScript library for building user interfaces. React. <https://reactjs.org/>
- [2] Remix Software, Inc. n.d. React Router: Declarative Routing for React.js. React Router. <https://reactrouter.com/web>
- [3] MUI (Material-UI). (2023). Material-UI: React components for faster and easier web development. Material-UI. <https://material-ui.com/>
- [4] Swiper. (2023). Swiper - The Most Modern Mobile Touch Slider. Swiper. <https://swiperjs.com/>
- [5] Meta. (2023). Jest - JavaScript Testing Framework. Jest. <https://jestjs.io/>