COMP9900 Information Technology Project

Final Report

Project name: FilmFinder

Team: COMP9900-H12A-0xdeadbeef

Members:

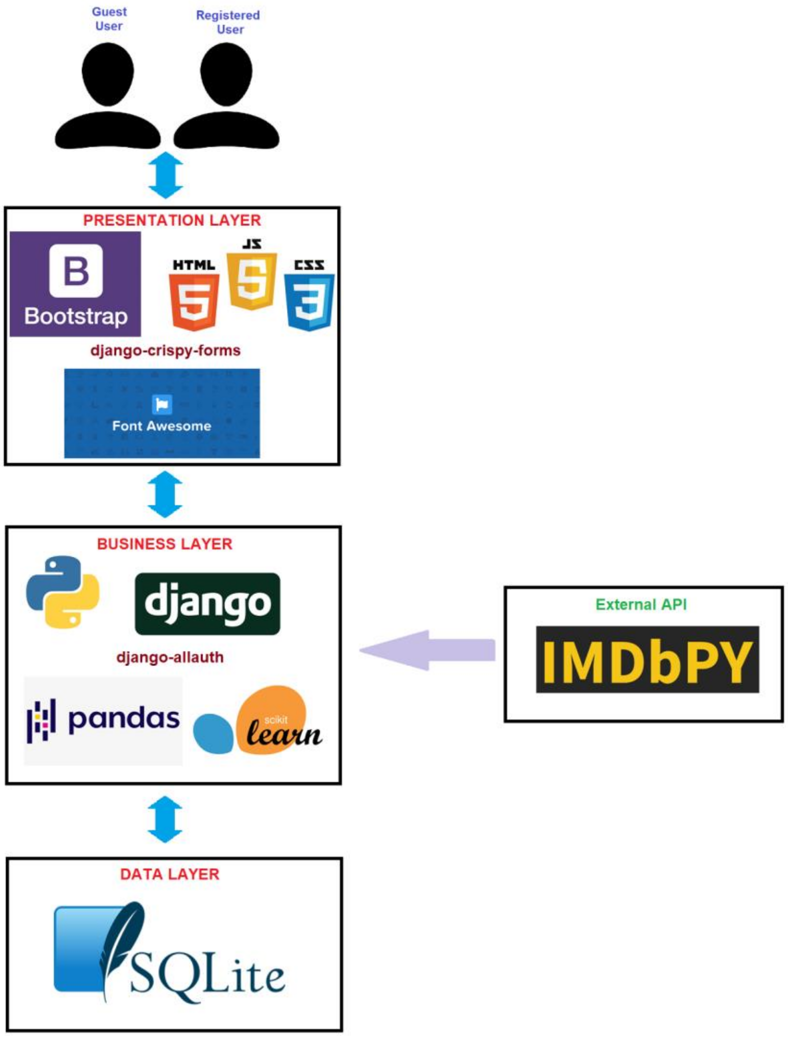
Name	E-mail	StudentID	Role
An Wei	an.wei@student.unsw.edu.au	z5215719	Developer
Jerry Edackatt	j.edackatt@student.unsw.edu.au	z5229867	Developer
Luyuan He	luyuan.he@student.unsw.edu.au	z5247759	Scrum Master, Developer
Xiao Han	xiao.han8@student.unsw.edu.au	z5203242	Developer
Yuta Sato	y.sato@student.unsw.edu.au	z5186797	Developer

Submission date: 16/11/2020

Table of Contents

1. Overview

The following section gives an overview of the structure of the project and the technologies employed in its architecture during its development. The figure below is an outline of the overall design of the project:



1.1. External actor interaction

The external actors in the FilmFinders project are the guest users and registered users. Both of these groups of users may interact with the platform via the presentation layer at the front-end. The guest users can access a great number of features while the registered users can enjoy some additional features. Both groups of users can browse through, search and view movies details including titles, genres, cast, reviews, etc. In addition to that, registered users can submit their reviews and ratings for any given movie, add movies to their personalized wish lists and block (or “ban”) users whose opinions they wish to ignore on FilmFinders. If a guest user chooses to use these features, they must log in using their account credentials. If they do not already have an account, they may choose to register an account on FilmFinders via the signup page.

1.2. Presentation layer

The presentation layer is the layer that is responsible for interacting with the end-users directly, i.e. the front-end. Bootstrap was used as the primary front-end framework to enable flexibility while working with CSS, HTML and JavaScript (including jQuery scripts) leading to responsive and dynamic web pages. The project also uses *Django-crispy-forms* which uses Bootstrap classes by default. This allowed for better and more convenient management of online web forms. Further polishing of fonts, buttons, etc. was possible with Bootstrap using frameworks like Font Awesome and other native Bootstrap features. The combination of these frameworks based on a Bootstrap foundation allowed the project to better serve the needs of the end-user in a simple yet effective manner.

1.3. Business layer

The business layer is the implementation of the core functionality and logic of the FilmFinders project. It is responsible for communicating between the presentation layer, the database and the integrated APIs. It is also equipped with the task of returning appropriate responses based on the requests sent as a result of the user interaction with the presentation layer. The business layer was implemented in Python 3.7 with Django 3 as the primary web framework for the project. It uses the WSGI web server since it is the default server for Django (in addition to being the Python standard). With Django handling a large chunk of the web application, some additional frameworks and technologies were integrated into the project. *Django-allauth* was used to incorporate third-party authentication for Twitter and Google. It eliminated the need for direct interaction with the Google and Twitter APIs as it organically fit into the Django environment. Moreover, to offer movie recommendations to the users based on a variety of (personalized) criteria, machine learning was applied. This was achieved with the help of the *scikit-learn* and *pandas* libraries. Finally, the database of movies was populated with the aid of the *IMDbPY* package which retrieved all relevant information related to the movies including titles, posters, cast, genres, etc.

1.4. Data layer

The data layer is managed by SQLite. Django supports SQLite as the database backend by default. Besides the aforementioned purpose of storing all the movie details fetched by the *IMDbPY* package, the database was also used for storing user data including login credentials, wish lists, block lists, average movie ratings, user-submitted reviews, etc. The use of SQLite as the embedded database backend ensured that FilmFinders remained a lightweight and portable project enhancing its overall performance.

2. Third-party functionalities

2.1. 3-clause BSD License

All functionalities provided by the Django web framework, scikit-learn and pandas use this license. They are provided "AS IS" and they are free for us to use.

django.urls.path https://docs.djangoproject.com/en/3.1/topics/http/urls/

This is used to construct an URL pattern which Django uses to process a request.

django.urls.include https://docs.djangoproject.com/en/3.1/topics/http/urls/

This is used to "include" other urlconf modules for our urlpatterns.

django.urls.reverse_lazy
https://docs.djangoproject.com/en/3.1/ref/urlresolvers/#reverse_lazy

This is used on our password change and reset features. To provide a reversed URL as the URL attribute of a generic class-based view.

django.urls.reverse https://docs.djangoproject.com/en/3.1/ref/urlresolvers/#reverse

This is used to return an absolute path matching a given view and optional parameters.

django.shortcuts.render
https://docs.djangoproject.com/en/3.1/topics/http/shortcuts/#render

This is used to combine a given template with a given context dictionary and returns an HttpResponse object with that rendered text.

django.shortcuts.redirect
https://docs.djangoproject.com/en/3.1/topics/http/shortcuts/#redirect

This is used to return an HttpResponseRedirect to the app appropriate URL for the arguments passed.

django.shortcuts.get_object_or_404
https://docs.djangoproject.com/en/3.1/topics/http/shortcuts/#get-object-or-404

This is used to make sure the object exists. If it does not, an Http404 will be raised. We use this to validate movies and user models.

django.utils.http https://docs.djangoproject.com/en/3.1/ref/utils/#module-django.utils.http

We use the urlsafe_base64_encode method from this package to encode a byte string to a base64 string for use in URLs. This is used to generate a user password reset URL.

django.utils.translation https://docs.djangoproject.com/en/3.1/topics/i18n/translation/

This package is used to make Django translation strings that will be translated into the end user’s language. This is used in our project to record the movie review creation time based on the user's time zone.

django.contrib.auth https://docs.djangoproject.com/en/3.1/ref/contrib/auth/

This is used for Django’s authentication system. We used it to handle our user account management module including user creation, user login, user logout, user password change and user password reset.

django.contrib.messages https://docs.djangoproject.com/en/3.1/ref/contrib/messages/

This is used to show one-time notification messages across our project. We used these messages to inform the users if their operation was successful or not.

django.core.mail https://docs.djangoproject.com/en/3.1/topics/email/

This is used to send password reset emails to registered users if they provided valid email addresses.

django.core.exceptions https://docs.djangoproject.com/en/3.1/ref/exceptions/

This is used to raise validation errors on our new user registration form.

django.core.paginator https://docs.djangoproject.com/en/3.1/topics/pagination/

This is used on our homepage to split all of our movie entries into several pages with navigation links.

django.db.models https://docs.djangoproject.com/en/3.1/topics/db/models/

This is used to easily access and integrate with our SQLite database where we stored all the data. Each model is mapped as a single table and each attribute of the model represents a field in the database.

django.http.HttpResponse https://docs.djangoproject.com/en/3.1/ref/request-response/#httpresponse-objects

This is used to respond to simple strings to a request.

django.http.Http404 https://docs.djangoproject.com/en/3.1/topics/http/views/#the-http404-exception

This is used to raise a 404 page in our project for a variety of situations.

django.http.HttpResponseRedirect https://docs.djangoproject.com/en/3.1/ref/request-response/#django.http.HttpResponseRedirect

This is used to handle situations where we want to redirect users to other pages within the response.

django.forms.ModelForm
https://docs.djangoproject.com/en/3.1/topics/forms/modelforms/#modelform

This can be used to create a simple form class from a Django model. We used it to create the user rating and review form for movies.

django.forms.Textarea
https://docs.djangoproject.com/en/3.1/topics/forms/modelforms/#modelform

This is used to provide a text input field for our users to write their reviews in.

django.forms.EmailField https://docs.djangoproject.com/en/3.1/ref/forms/fields/#emailfield

This is used to include an Email field on our new user registration form.

django.template.Library https://docs.djangoproject.com/en/3.1/howto/custom-template-tags/

This is used to create custom template filters.

sklearn.feature_extraction.text.CountVectorizer https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

This is used to produce a matrix (count vector) from a collection of text (words)
If a user defines a vocabulary parameter, it maps a text followed by a dictionary defined by the parameter.

sklearn.metrics.pairwise.cosine_similarity https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.html

This is used to calculate the cosine similarity between two vectors.

pandas https://pandas.pydata.org

This is used in the movie recommendation system to handle data processed by scikit-learn.

2.2. GPL license

All functionalities provided by IMDbPY use a GPL license. We can freely use them.

IMDbPY https://imdbpy.github.io

This is used to retrieve movie and related information from the International Movie Database (IMDb) to our local database.

2.3. MIT license

All functionalities provided by django-crispy-forms, django-allauth, Bootstrap and jQuery use MIT license. There are no restrictions on how we use them.

django-crispy-forms https://github.com/django-crispy-forms/django-crispy-forms

This is used to make all of our forms look nice and crispy.

django-allauth https://github.com/pennersr/django-allauth

This is used to enable users to log in to our website and use their social network accounts.

Bootstrap **https://getbootstrap.com**

This is used to render our user interfaces to look nicer and more modern.

jQuery **https://jquery.com**

This is used to add some dynamic elements to our web pages such as the movie detail page.

2.4. Freemium

All functionalities provided by Font Awesome use Freemium license. We can freely use the basic features of Font Awesome, which is enough for our project.

Font Awesome **https://fontawesome.com**

This is used to add some icons like the heart icon to our web pages.

3. Functionalities

3.1. New User Registration

New user registration functionality can be used by new visitors who do not have accounts with us. They can register accounts with our website to use the full features. This functionality is supported by the Django user authentication system. A registration form is provided for inputting a username, an email address and a password. There are instructions for the input formats and restrictions for each field. Upon submission, a new user account will be created if all the inputs are valid and the user will be directed to our homepage. Specific errors will be given for each field if the inputs are not valid, so the user knows how to change it. User credentials are stored in our database with their passwords hashed and salted.

3.2. Login Using Social Network Accounts

Instead of registering new accounts with our website, new visitors could save registration time by using their social network accounts to log in directly. They will be directed to the selected social network's authorization page. We support Google and Twitter accounts. Using Facebook accounts to login is also implemented, however, Facebook forces websites using its API to use HTTPS protocol. Since Django is run on a local server using the HTTP protocol, login by Facebook was later removed.

Corresponding project objectives: novelty.

3.3. Login and Logout

Both login and logout options are provided on the navigation bar on top of each page of our website. Registered users can log in into our website as long as they provide valid pairs of username and password. After logging in, user sessions are stored for two weeks so they do not have to login again if they re-visit our website. Users can manually log out from our website any time to close their sessions.

3.4. Password Change and Reset

Registered users can change their passwords. They have to provide their valid old passwords to change it to a new one. If a user happens to forget their password, they can request for a password reset by providing the email address used during registration. After checking with our record to make sure an account with this email address exists, an instruction email will be sent to that email address with a one-time use password reset link. The user will then be able to input a new password.

Corresponding project objectives: novelty.

3.5. Browse for Movies

Users can browse all the movies we have in our database on the homepage. The movies are sorted by their average rating. If multiple movies have the same average rating, they will be sorted alphabetically by their title. If a user is logged in and has blocked other users using the blocklist feature, the ratings made by those blocked users will not affect the average ratings. Users can filter these movies by their genres, directors and average rating ranges. If a movie does not have any reviews yet, its average rating will be shown as a 0. Upon clicking each movie, users will be directed to that movie's detail page. The pagination provided divides all movies into pages and each page displays 12 movies.

Corresponding project objectives:

1) The platform must also allow film finders to browse movies by director or genre, with movies sorted from most popular to least popular based on the movie average rating, and movies with an equal rating being sorted alphabetically (movies with no rating would be treated as having an average rating of 0).
2) The reviews written by the banned review writer(s) will not influence the average rating for the FilmFinder who banned the review writer(s).

3.6. Search for Movies

A search box is provided on the navigation bar on top of each page of our website. Users can input keywords here and search for movies they are interested in. A list of movie results will be returned based on the movie name, description, or genre matches with the keywords. Specifically, while searching for movie description matches, the order of keywords does not matter because users would not know how exactly they are ordered in the movie descriptions. The result list displays a title, an average rating and a cover image for each movie and it is sorted by average rating. If a user is logged in and has blocked other users using the blocklist feature, the ratings made by those blocked users will not affect the average ratings. Upon clicking each movie's title in the list, users will be directed to that movie's detail page.

Corresponding project objectives:

1) Film finders must be able to search for movies they are interested in by keywords that match the movie name, description, or genre, with listed results showing matching movie names, and their latest average rating.
2) The reviews written by the banned review writer(s) will not influence the average rating for the FilmFinder who banned the review writer(s).

3.7. Rate and Review Movies

Registered users can give a rated review on each movie's detail page. A user can only leave one rated review for each movie. A rating is an integer from 0 to 5. A review is a text field where users can type in for a maximum of 500 characters. A reminder tooltip under the text field will indicate how many more characters are allowed. Users are also able to edit and delete their reviews on the movie's detail page.

If the current user is not authenticated, this feature is disabled. Instead, there will be help texts to prompt the user to log in or register first.

The reviews for each movie are displayed on that movie's detail page.

Corresponding project objectives:

1) A film finder must also be able to leave a review for any given movie they browse on the platform, where such a review includes text as well as a rating from 0 to 5. Film finders must also be able to view the wishlist of any other film finder that has left a review for a given movie.
2) Editing and deleting review functionalities are novelty.

3.8. Wishlist

Each registered user has a wishlist to record movies they find interesting across the platform. This list can be accessed from the navigation bar. They can add movies into this list from the movie detail pages and delete movies already in the wishlist. Each user's wishlist is viewable by other registered users from the movie detail page if that user has left a review for that movie.

If the current user is not authenticated, this feature is disabled. Instead, there will be help texts to prompt the user to log in or register first.

Corresponding project objectives:

1) Film finders must be able to add any movie they come across on the platform to a wishlist of movies they'd like to watch and must also be able to remove movies from this wishlist.
2) Film finders must also be able to view the wishlist of any other film finder that has left a review for a given movie.

3.9. Blocklist

Each registered user has a blocklist to block other users that they do not want to see. All the reviews by those blocked users will not show for the owner of the blocklist. All the ratings made by those blocked users will not affect the average rating of each movie for the owner of the blocklist. Registered users can unblock other users in their blocklist.

If the current user is not authenticated, this feature is disabled, instead there will be help texts to prompt the user to login or register first.

Corresponding project objectives:

The platform must also allow film finders to add any review writer(s) they wish to their “banned” list, meaning that the reviews written by the banned review writer(s) will not be seen or influence the average rating for the FilmFinder who banned the review writer(s).

3.10. Similar Movie Recommendations

On each movie’s detail page, users are able to view other similar movies recommended by our platform. There are three different attributes all users can select including genre, director and cast. If a user is authenticated, one extra attribute, user reviews will be shown. Each attribute will display top five similar movies.

Corresponding project objectives:

The full details page for a given movie must also show recommendations for other movies that are “similar”, where such recommendations must be based on at least review history, and a selection of multiple attributes that a user can select (e.g., genre and/or director).

3.11. Movie Details Page

Each movie has its own details page. This page includes full details of this movie, all the reviews associated with it and similar movies recommended by our system.

Corresponding project objectives:

A film finder must be able to view the full details of any movie they come across on the platform, including the movie name, description, genre, cast, director, latest average rating, and all associated reviews.

3.12. Watch It Now on Streaming Services

Each movie’s details page also contains a ‘Watch it now’ button which will direct users to a service provided by justwatch.com. It will list all the available streaming services that have this move. Users can choose their favorite services to watch this movie.

Corresponding project objectives: novelty.

4. Implementation Challenges

4.1. Recommendation system

Nowadays, recommendation systems are used everywhere on the internet and are capable of predicting user preference of items by user history or various other categories such as genre, creator, country of origin, etc. The reason recommendation systems are needed for applications like online shopping or streaming services is that there are too many options to choose from compared to the past. For example, Netflix has a system to help users choose movies they want to watch from an enormous library. In this project, we have applied a simpler version of this system. For recommendation systems, there are two basic categories: Content Based Filtering and Collaborative Filtering (see figure below).

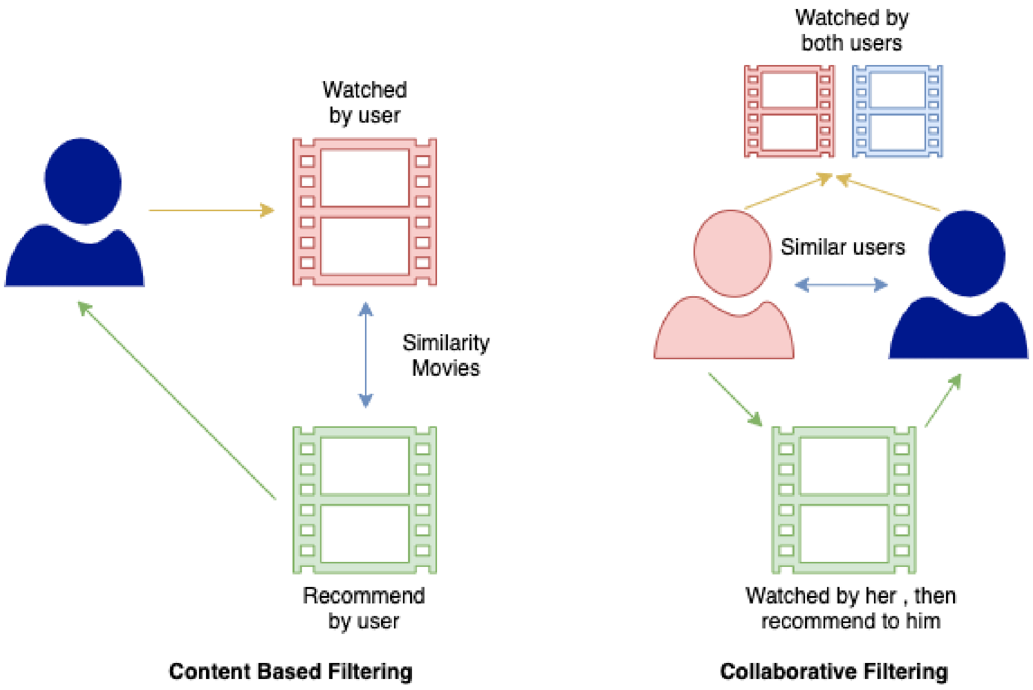


Fig. Two types of recommendation system

We use the former method of recommendation system for genre-, director- and cast-based recommendations. The latter method is used for recommendation by user reviews.

4.1.1. Content Based Filtering

It is commonly used in building recommendation systems by Machine Learning. We use this technique for recommendation by genre to calculate the cosine similarity between movies. However, when using Machine Learning, preprocessing raw data is a necessity. Therefore, we use pandas and scikit-learn libraries for this process (movie/views.py).

Firstly, we need to define genre categories such as

genre_list = ['Fantasy', 'Mystery', 'Sport', 'History', 'Biography', 'Crime', 'Sci-Fi', 'Film-Noir', 'Western', 'Animation', 'Romance', 'War', 'Family', 'Thriller', 'Documentary', 'Musical', 'Comedy', 'Drama', 'Horror', 'Music', 'Adventure', 'Action'] .

It converts the array of each movie genre data to count vectors.

For example, if the movie “Harry Potter and the Sorcerer's Stone” genre is ‘Adventure’, ‘Family’ and ‘Fantasy’ then this count vector is

[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0]

(the corresponding index in the count vector is set to 1 for the relevant genres and 0 for others).

Next it calculates the cosine similarity between the movie on the current page and all other movies. Finally, it takes the top 5 similar movies from this list (the closer the movie similarity to 1, the more similar the movie is). For recommendation systems by directors, almost all movies have 1 director, therefore, it makes movie sets using Django template (objects.filter) to collect movies directed by the same director (except for the current movie). Then if the number of movies is more than 5, it picks random 5 movies from this set. For recommendation systems by casts, a hash (dictionary) is used for each movie and the count value is incremented by 1 for every common cast member between the current movie and the rest. The top 5 movies with the highest number of common cast members are then chosen.

4.1.2. Collaborative Filtering

Unlike Content-based Filtering, Collaborative Filtering is related to other users' ratings and reviews. It does not rely on the content itself but on how other users rate the product. In this project, a simpler version of this system is implemented. Firstly, it checks whether the current logged in user has given a high rating (more than 4 out of 5) for the current movie. Next, it checks all other reviewers who have also rated the current movie highly and stores such users' names in a list. It then queries the database about other movies these users have rated highly. After obtaining this set of movies, if the size of the set is more than 5, it picks 5 movies from this set at random. (Flow chart below)

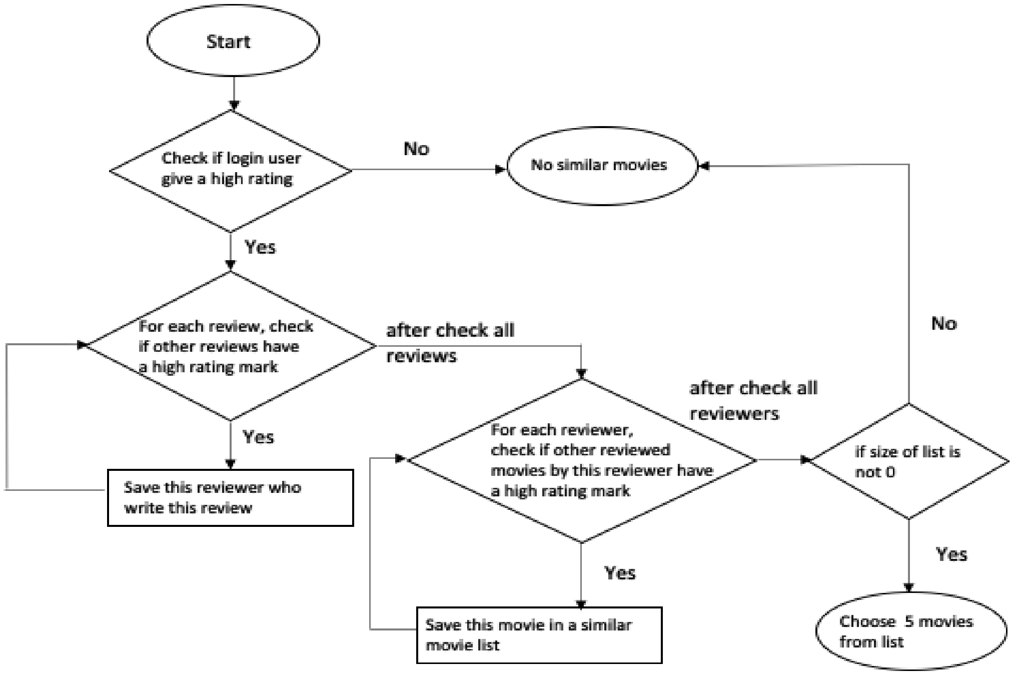

Start
Check if login user give a high rating
No
No similar movies
Yes
For each review, check if other reviews have a high rating mark
after check all reviews
Yes
Save this reviewer who write this review
For each reviewer, check if other reviewed movies by this reviewer have a high rating mark
after check all reviewers
if size of list is not 0
No
Yes
Save this movie in a similar movie list
Yes
Choose 5 movies from list

Fig. recommendation by user review

4.2. Average Rating Calculation

For each movie, the application shows an average rating. This average rating is calculated by

$$average\ rating = \frac{sum\ of\ rating\ score}{the\ number\ of\ reviewers}$$

However, since each user has a block list, if the user is currently logged in, the system needs to consider this while displaying the average. In this case, it requires a slightly different calculation:

$$average\ rating = \frac{sum\ of\ individual\ rating\ score\ from\ unbanned\ reviewers}{the\ number\ of\ reviewers\ -\ the\ number\ of\ banned\ reviewers}$$

Originally, every time the user visited the website, the system calculated the average in real time. However, we noticed that this was a costly operation and was one of the biggest reasons the web application was slow. Moreover, recalculating the average rating for registered users while ignoring the ratings of blocked users hindered performance further. This was because the original implementation took O(n*m) time where *n* is the number of movies in the database and *m* is number of blocked users.

Therefore, we decided to implement an alternative method to calculate the average rating. Instead of calculating the average each time, we decided to store the basic average score for each movie in the database and use this value to produce the average rating for guest users and users with no ban lists/block lists.

For the users with a block list, we implemented a many-to-many relation (both attributes are the foreign keys of the user database). First, the algorithm looks at the list of banned users and gathers a dictionary of movies that have been reviewed by them and the rating the blocked users have given for these movies. It then only recalculates the movies that are in that dictionary. This significantly reduces the size of *n* as the algorithm no longer has to cycle through each movie in the database to recalculate the average but only a subset of the total number of movies. After this change to the algorithm, it calculates the average rating as follows:

$$average\ rating\ = \frac{average\ rating\ *\ number\ of\ reviews\ -\ individual\ rating}{number\ of\ reviews\ -\ 1}$$

$$number\ of\ reviews\ =\ number\ of\ reviews\ -\ 1$$

As a result of this restructuring of our algorithm, the average processing time was cut down in half, if not more.

5. User Document/Manual

5.1. How to Set Up Project

To set up the project, you must have Python3.7 or above installed. If you don't, you can find it here: https://www.python.org/downloads/. If you set up everything on a CSE machine, Python3.7.3 has been installed on the path /usr/bin/python3.7. When you use the command 'python3', the terminal will refer to this path automatically. The following instructions assume the use of unix-based OS.
Extract the project zip and cd into it:

- *unzip project.zip && cd project*

Then install the required python framework Django and other dependencies.

- *python3 -m pip install -r requirement.txt*

5.2. Initializing the Website Server

You can start the server by calling the following command:

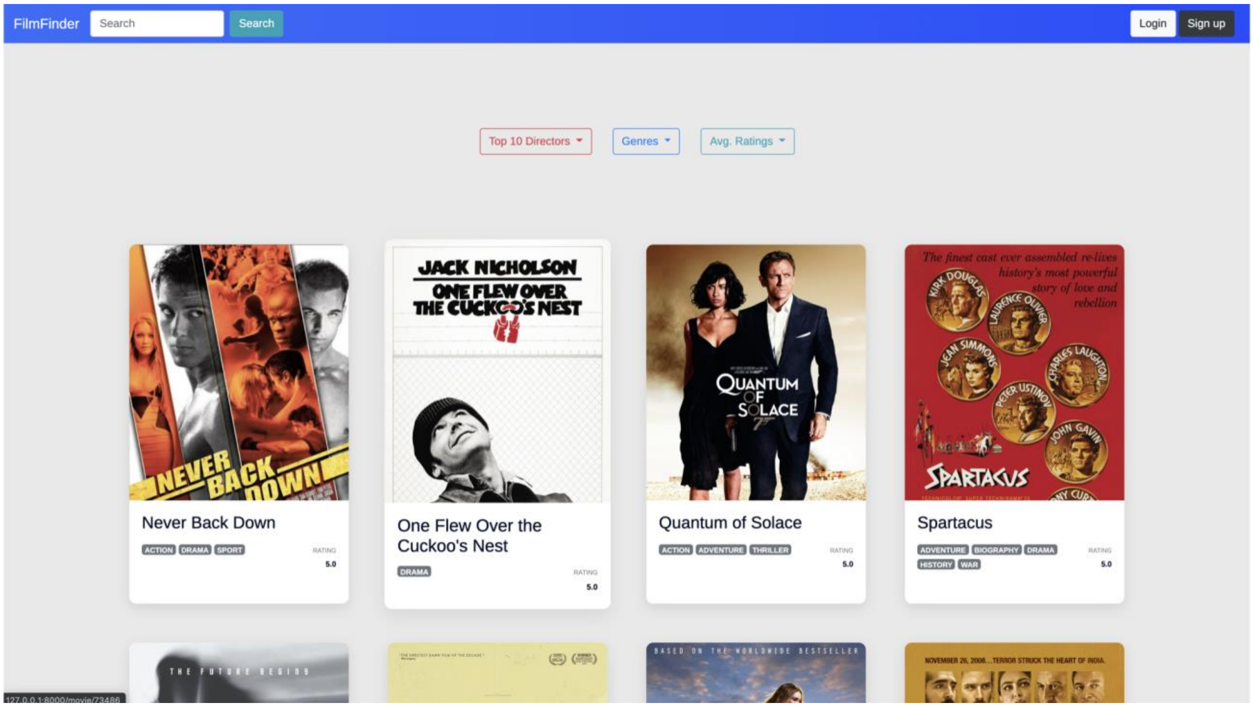
- *python3 manager.py runserver --insecure*

5.3. Stopping the Website Server

To stop the server, you can use the keyboard shortcut CTRL+C.
Information
Python3.7 or above is required for running the website server. Please make sure the Python version has been updated to 3.7 or above instead of Python2.x.
After initializing the web server, go to the URL, http://127.0.0.1:8000.

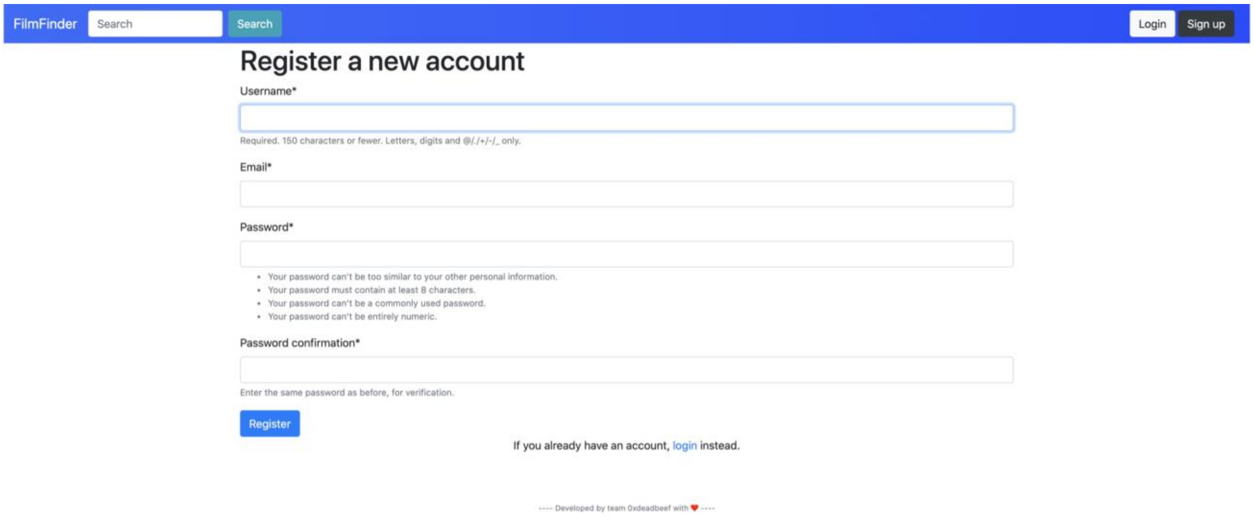
5.4. Register and login

On the upper right corner of the FilmFinder homepage, there are two buttons to click, “Login” and “Sign Up” respectively. If users do not have an account yet, they can register first.



Click the “Sign Up” button to be redirected to a registration page, where you can choose a preferred username and a personal email address to register a new account for our FilmFinder.

There are some requirements for username, like less than 150 characters and letters, digits and @/./+/-/_ only. For security purposes, users are not allowed to use common, simple or numeric passwords. In case users have got one account but click the “Sign Up” button accidentally, they can find out a link located on the bottom of this page, which will lead users back to the login page.



Register a new account

Username*

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Email*

Password*

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

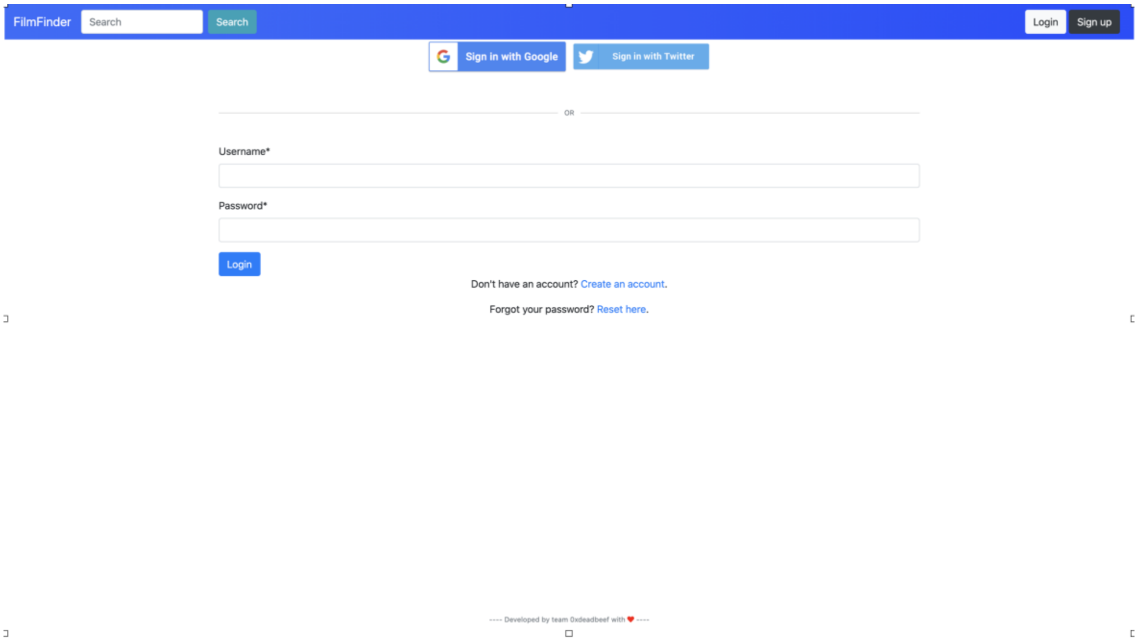
Password confirmation*

Enter the same password as before, for verification.

If you already have an account, login instead.

---- Developed by team 0xdeadbeef with ❤ ----

On the login page, users can choose either an account of FilmFinder that has been created before or an account of a third-party platform to login.



Username*

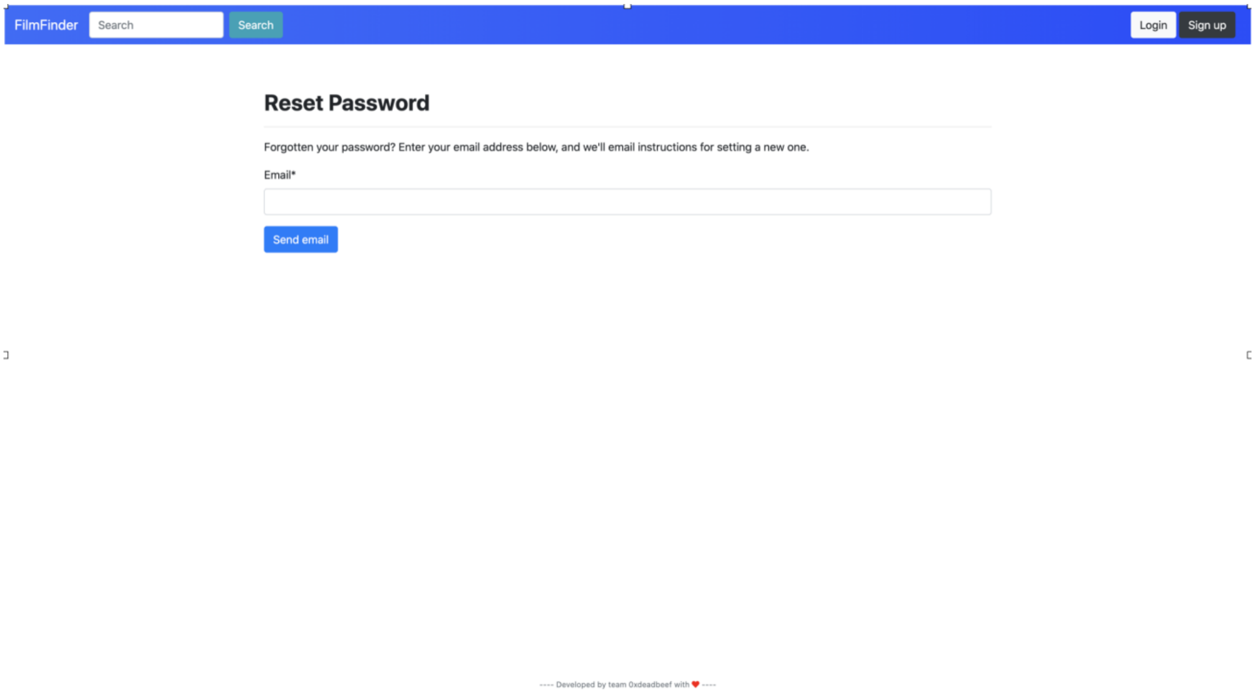
Password*

Don't have an account? Create an account.

Forgot your password? Reset here.

---- Developed by team 0xdeadbeef with ❤ ----

In case users forget their password, we provide a "reset here" link where you can fill in the email address you used for the account registration. An instruction email will be sent to that email address with a one-time use password reset link



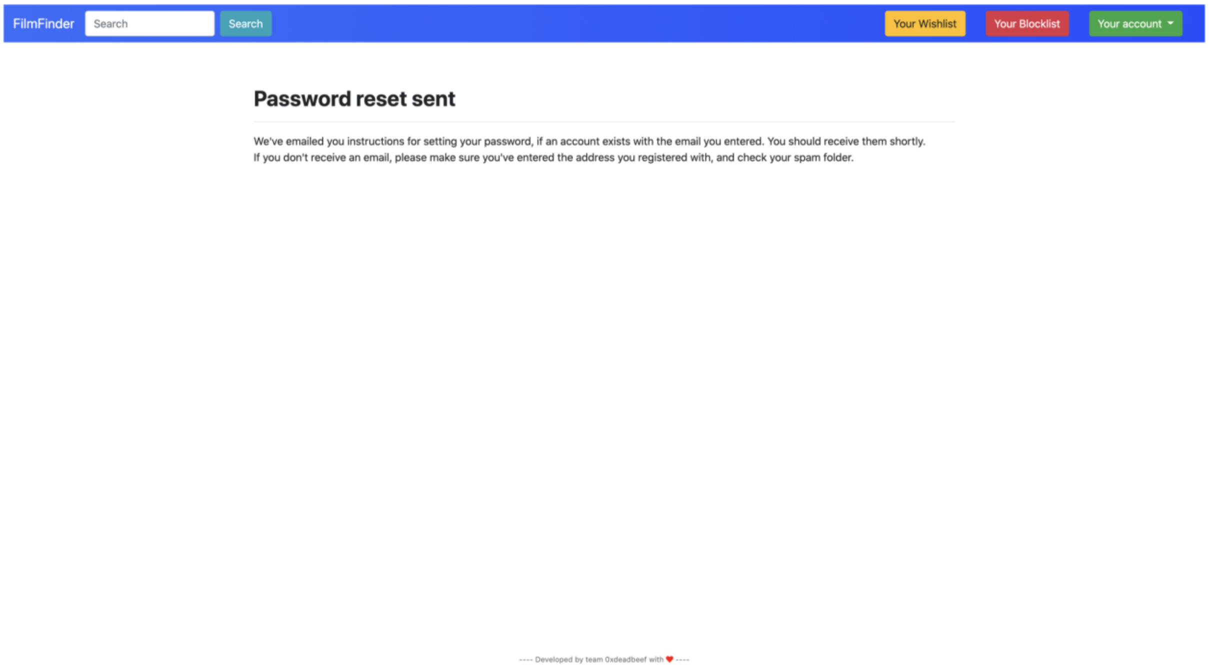
Reset Password

Forgotten your password? Enter your email address below, and we'll email instructions for setting a new one.

Email*

---- Developed by team 0xdeadbeef with ❤ ----

After a user submits the correct email address, the user will be redirected to the page attached below, informing them of the password reset email sent.

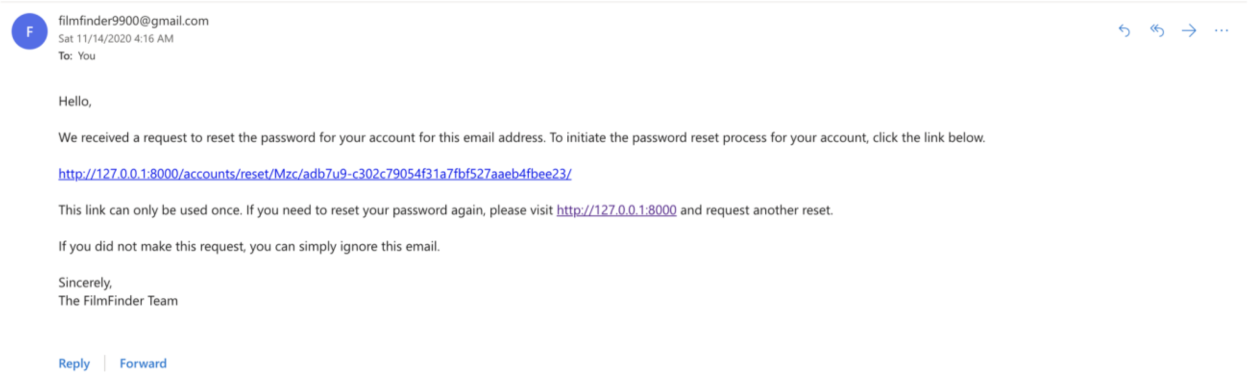


Password reset sent

We've emailed you instructions for setting your password, if an account exists with the email you entered. You should receive them shortly.
If you don't receive an email, please make sure you've entered the address you registered with, and check your spam folder.

---- Developed by team 0xdeadbeef with ❤ ----

Following the instructions in the password-reset email, users can reset the password.



filmfinder9900@gmail.com
Sat 11/14/2020 4:16 AM
To: You

Hello,

We received a request to reset the password for your account for this email address. To initiate the password reset process for your account, click the link below.

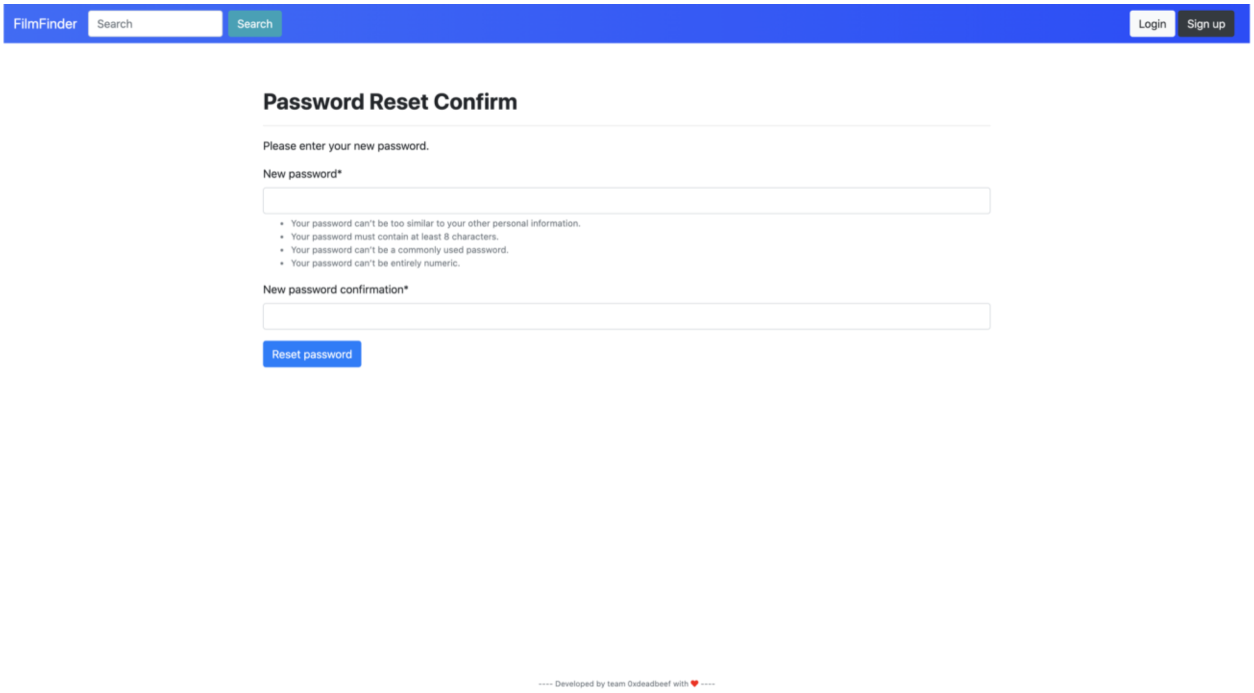
http://127.0.0.1:8000/accounts/reset/Mzc/adb7u9-c302c79054f31a7fbf527aaeb4fbee23/

This link can only be used once. If you need to reset your password again, please visit http://127.0.0.1:8000 and request another reset.

If you did not make this request, you can simply ignore this email.

Sincerely,
The FilmFinder Team

After clicking on the reset link, the user can choose a new password as shown below.



Password Reset Confirm

Please enter your new password.

New password*

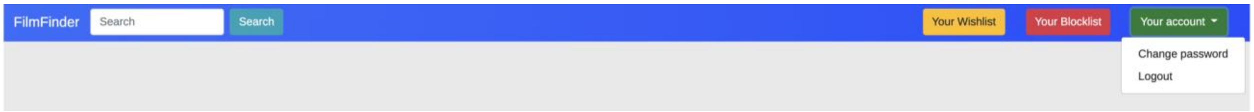
- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

New password confirmation*

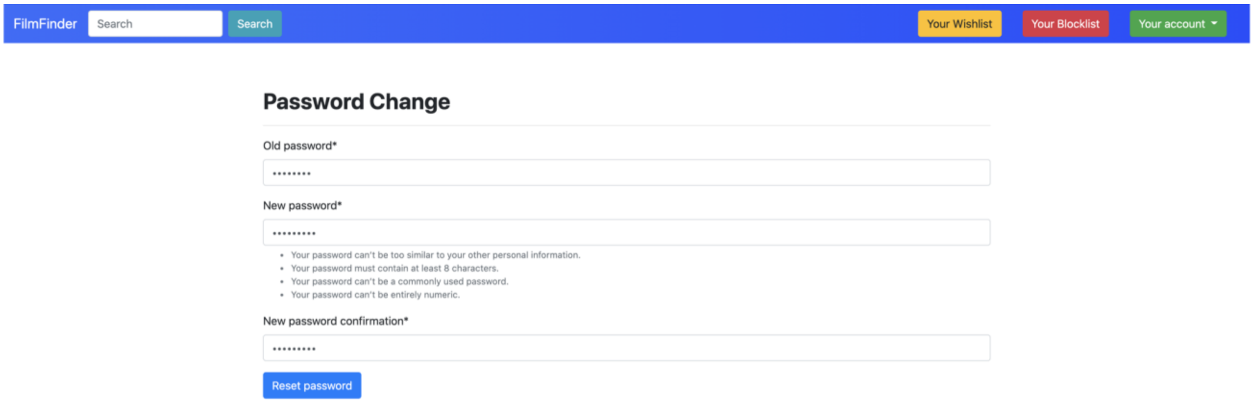
---- Developed by team 0xdeadbeef with ❤ ----

5.5. Account management

After a user successfully logs in, they will see three new buttons, “Your Wishlist”, “Your Blocklist” and “Your account”, on the top right corner of the home page. Current wishlist and blocklist will be empty for a new user, but they can add their favorite movies onto the wishlist and put users that they choose to block onto the blocklist. “Your account” is a dropdown button, with options to change the password and logout.



Unlike password reset, changing the password requires the user to enter the old password in addition to choosing a new password.



Password Change

Old password*

••••••••

New password*

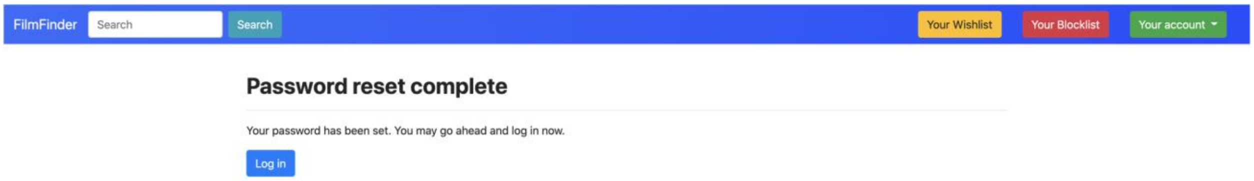
•••••••••

- Your password can’t be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can’t be a commonly used password.
- Your password can’t be entirely numeric.

New password confirmation*

•••••••••

After successfully changing the password, users will see the page below. Click the “login” button, they will be redirected back to the home page.

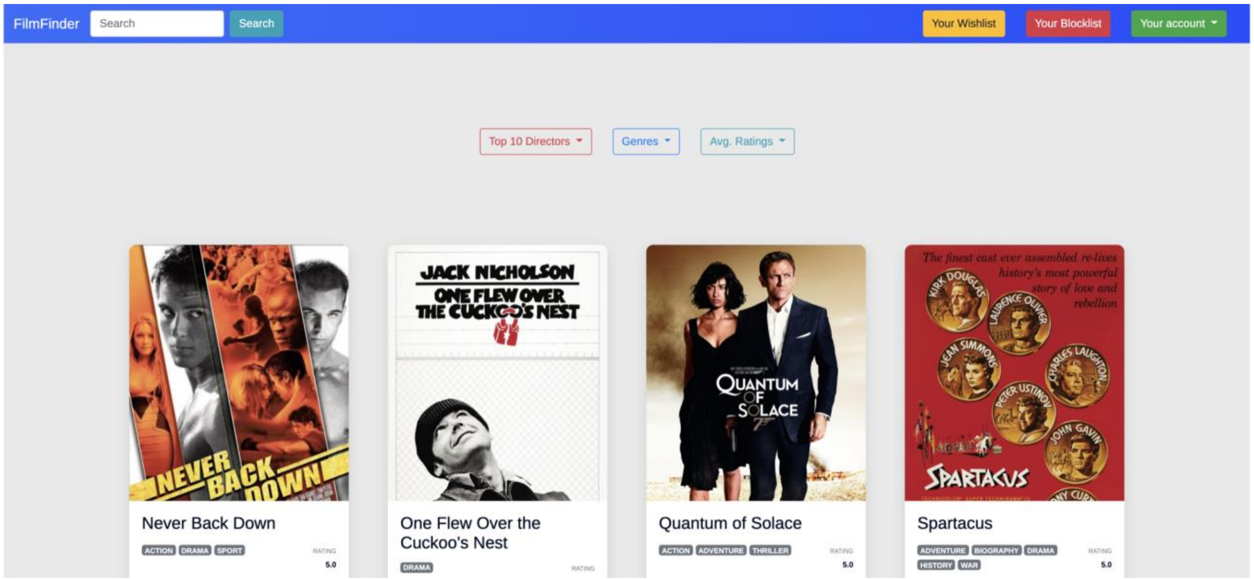


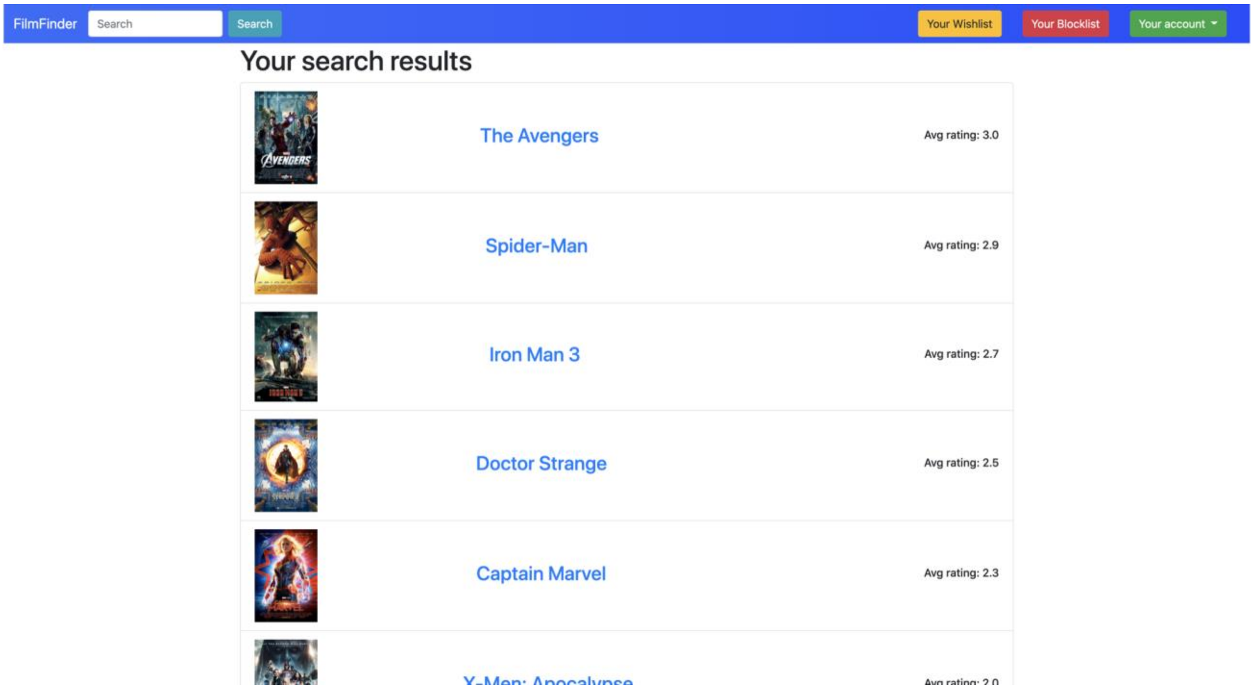
Password reset complete

Your password has been set. You may go ahead and log in now.

5.6. Search function

In the left top corner of the home page, there is a search bar where users can find any movie by typing in some keywords. For example, if users want to search for some Marvel movies, and type in “Marvel” and hit the search button, relevant search results will be returned.





Your search results

The Avengers Avg rating: 3.0

Spider-Man Avg rating: 2.9

Iron Man 3 Avg rating: 2.7

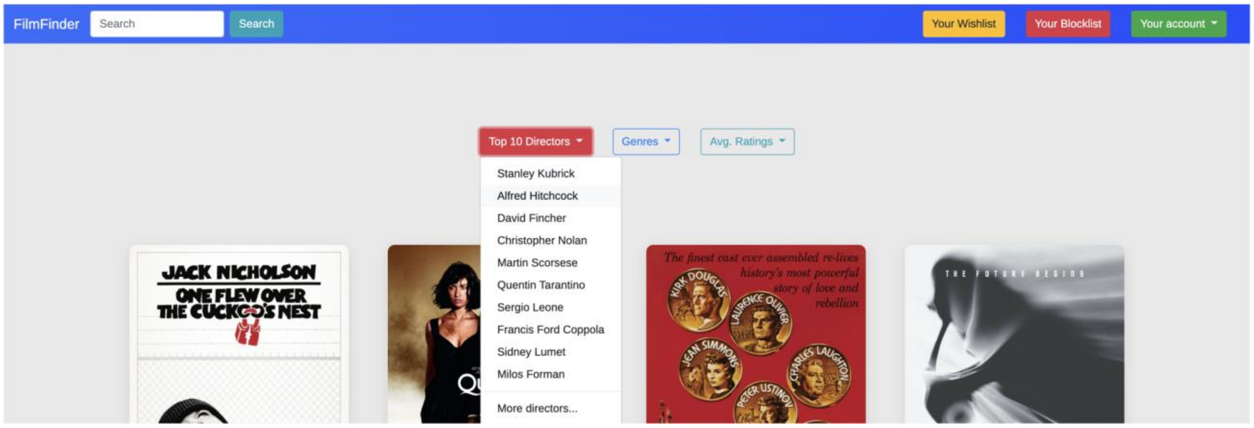
Doctor Strange Avg rating: 2.5

Captain Marvel Avg rating: 2.3

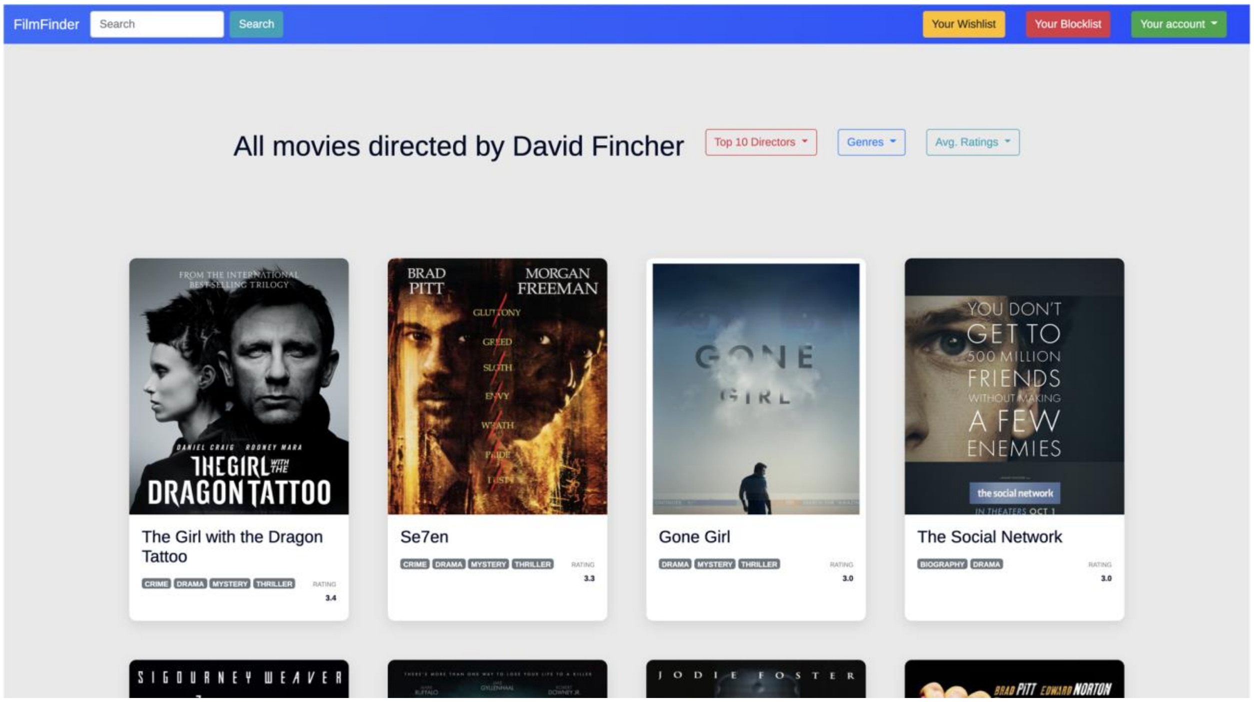
X-Men: Apocalypse Avg rating: 2.0

5.7. Search movies by categories

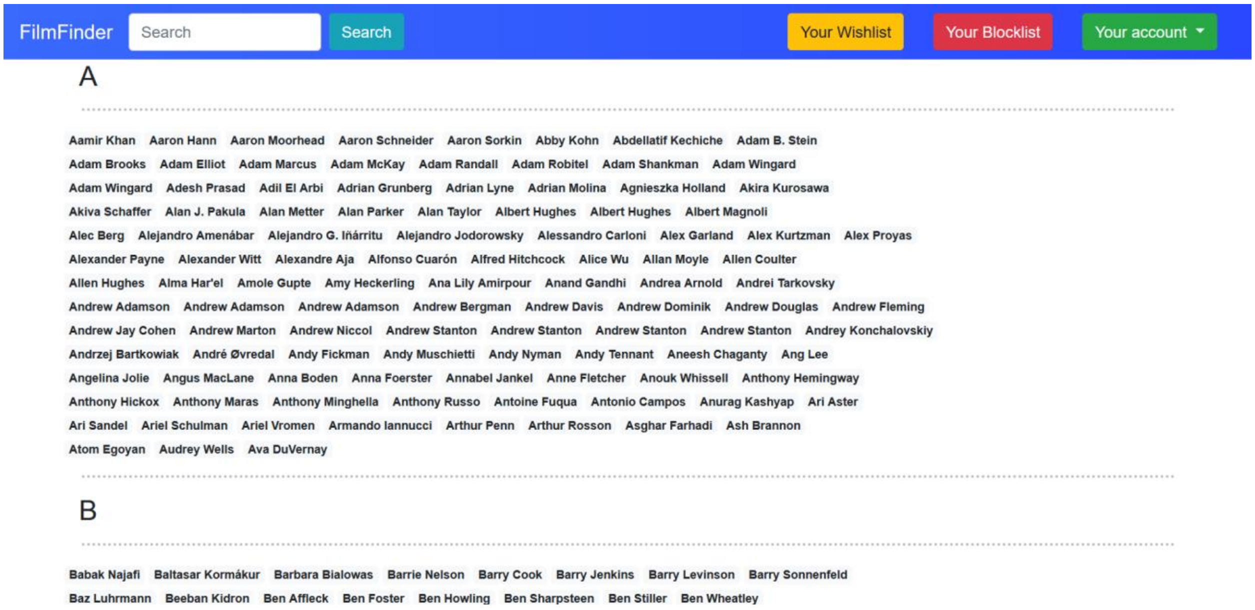
We sort movies by three categories: “top 10 directors”, “genres” and “average ratings”. If users want to find movies of top 10 most popular directors, just click the “top 10 directors” dropdown button, and users will see the 10 directors on the list.



For example, a user is interested in David Fincher. After clicking his name on the list, the user will be redirected to a new page, where all of his movies will show up in alphabetical order.



If users want to see all of the directors, they can click “More directors…” on the dropdown list. They can see all of the directors listing on the web page. If users want to check a specific director’s movies, they can click on the director’s name on the page.



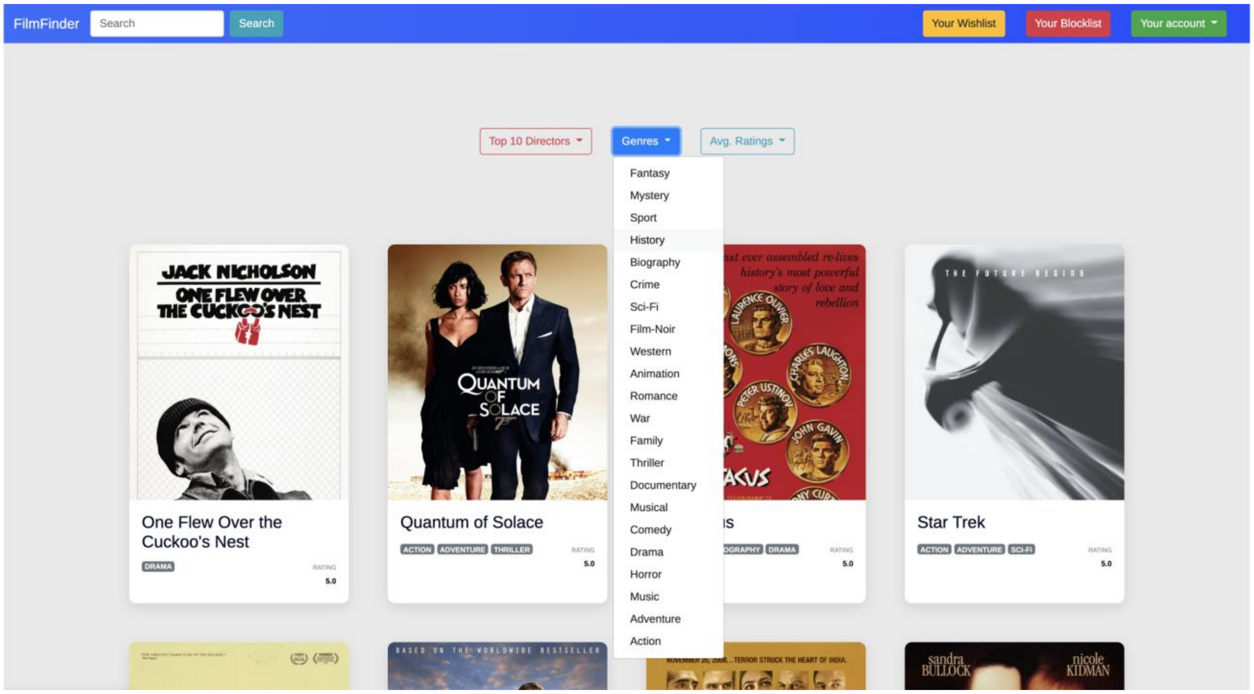
A

Aamir Khan Aaron Hann Aaron Moorhead Aaron Schneider Aaron Sorkin Abby Kohn Abdellatif Kechiche Adam B. Stein
Adam Brooks Adam Elliot Adam Marcus Adam McKay Adam Randall Adam Robitel Adam Shankman Adam Wingard
Adam Wingard Adesh Prasad Adil El Arbi Adrian Grunberg Adrian Lyne Adrian Molina Agnieszka Holland Akira Kurosawa
Akiva Schaffer Alan J. Pakula Alan Metter Alan Parker Alan Taylor Albert Hughes Albert Hughes Albert Magnoli
Alec Berg Alejandro Amenábar Alejandro G. Iñárritu Alejandro Jodorowsky Alessandro Carloni Alex Garland Alex Kurtzman Alex Proyas
Alexander Payne Alexander Witt Alexandre Aja Alfonso Cuarón Alfred Hitchcock Alice Wu Allan Moyle Allen Coulter
Allen Hughes Alma Har'el Amole Gupte Amy Heckerling Ana Lily Amirpour Anand Gandhi Andrea Arnold Andrei Tarkovsky
Andrew Adamson Andrew Adamson Andrew Adamson Andrew Bergman Andrew Davis Andrew Dominik Andrew Douglas Andrew Fleming
Andrew Jay Cohen Andrew Marton Andrew Niccol Andrew Stanton Andrew Stanton Andrew Stanton Andrew Stanton Andrey Konchalovskiy
Andrzej Bartkowiak André Øvredal Andy Fickman Andy Muschietti Andy Nyman Andy Tennant Aneesh Chaganty Ang Lee
Angelina Jolie Angus MacLane Anna Boden Anna Foerster Annabel Jankel Anne Fletcher Anouk Whissell Anthony Hemingway
Anthony Hickox Anthony Maras Anthony Minghella Anthony Russo Antoine Fuqua Antonio Campos Anurag Kashyap Ari Aster
Ari Sandel Ariel Schulman Ariel Vromen Armando Iannucci Arthur Penn Arthur Rosson Asghar Farhadi Ash Brannon
Atom Egoyan Audrey Wells Ava DuVernay

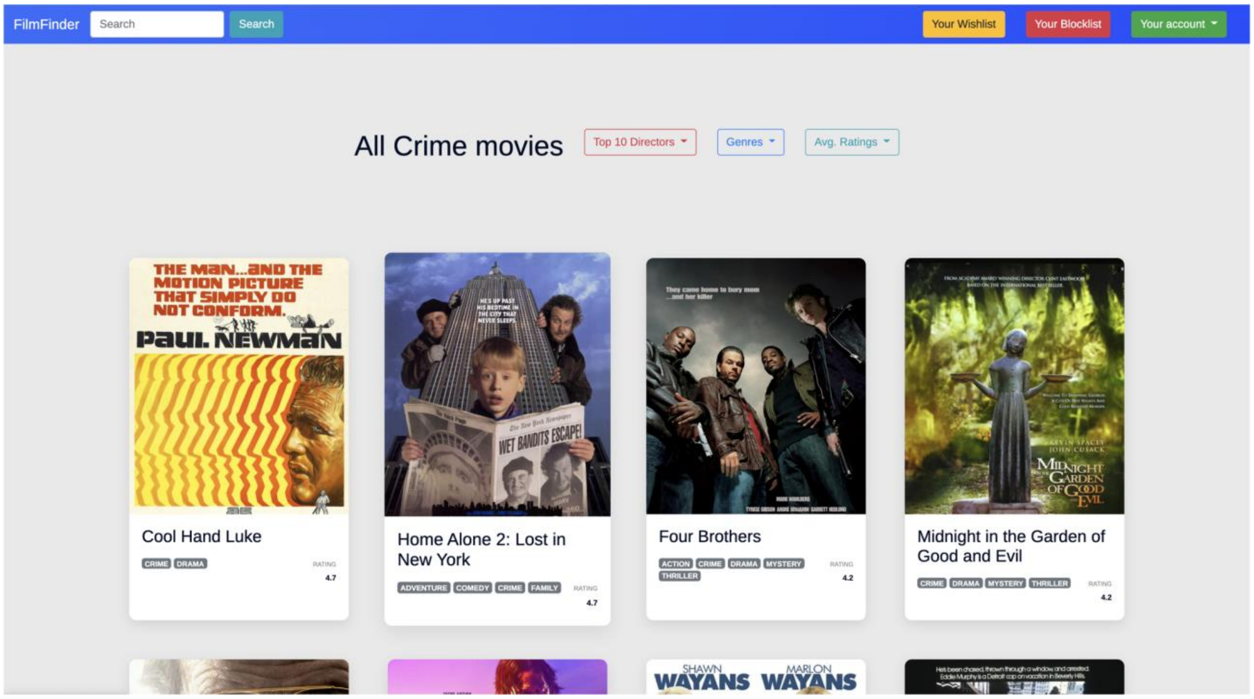
B

Babak Najafi Baltasar Kormákur Barbara Bialowas Barrie Nelson Barry Cook Barry Jenkins Barry Levinson Barry Sonnenfeld
Baz Luhrmann Beeban Kidron Ben Affleck Ben Foster Ben Howling Ben Sharpsteen Ben Stiller Ben Wheatley

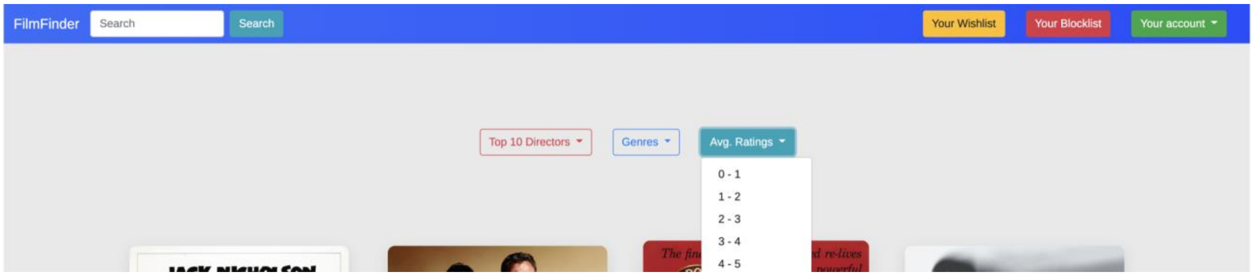
Besides the “top 10 directors” category, we also classify movies by genres, like sport, history, crime and etc.



If users search for crime movies, all movies of this genre will be listed on the page.



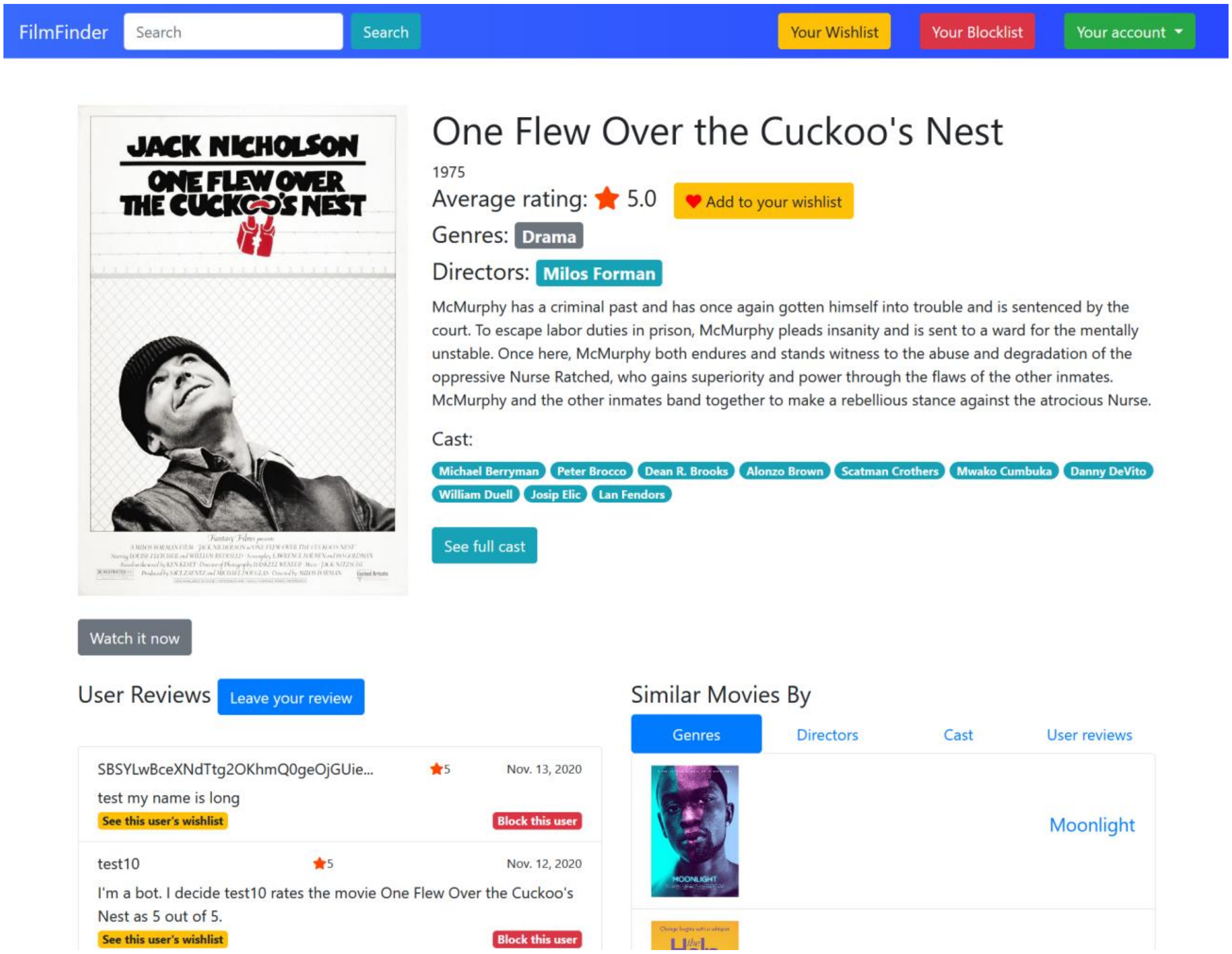
Users can also search movies by average rating.



5.8. Review and rate

Visiting the movies' detail pages, users will see some detailed information about each movie, like average rating, the genre it belongs to, directors, the plot, cast, etc. If they click on "See full cast", they will see all the cast members who play a role in this movie.

There is a section on the page to show reviews from other users.



One Flew Over the Cuckoo's Nest

1975

Average rating: ★ 5.0

Genres: Drama

Directors: Milos Forman

McMurphy has a criminal past and has once again gotten himself into trouble and is sentenced by the court. To escape labor duties in prison, McMurphy pleads insanity and is sent to a ward for the mentally unstable. Once here, McMurphy both endures and stands witness to the abuse and degradation of the oppressive Nurse Ratched, who gains superiority and power through the flaws of the other inmates. McMurphy and the other inmates band together to make a rebellious stance against the atrocious Nurse.

Cast: Michael Berryman, Peter Brocco, Dean R. Brooks, Alonzo Brown, Scatman Crothers, Mwako Cumbuka, Danny DeVito, William Duell, Josip Elic, Lan Fendors

User Reviews

SBSYLwBceXNdTtg2OKhmQ0geOjGUie... ★5 Nov. 13, 2020

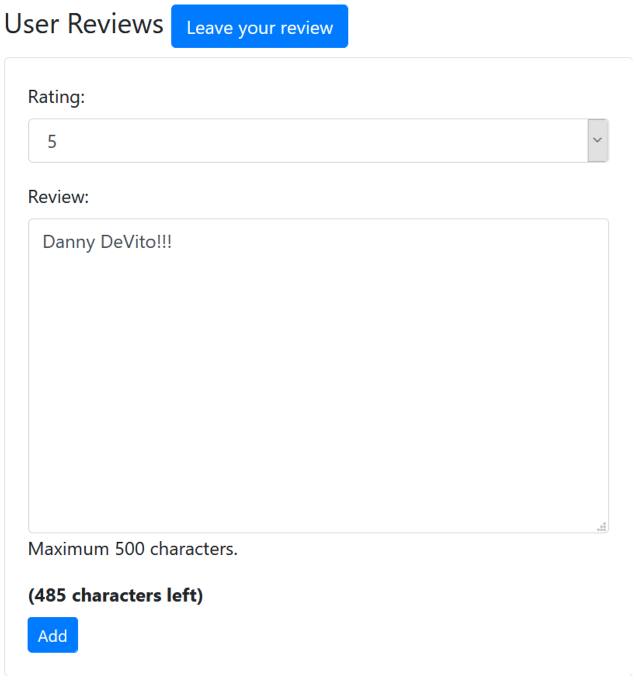
test my name is long

test10 ★5 Nov. 12, 2020

I'm a bot. I decide test10 rates the movie One Flew Over the Cuckoo's Nest as 5 out of 5.

Similar Movies By

Clicking on “Leave your review” will reveal a dropdown list and a text field, where users can leave a rating for this movie and submit a review. The character count will indicate how many more characters the users are permitted in the review. After finishing reviewing and rating, hit “add”, and the review will be added to the movie’s detail page. At the same time, after a new rating is submitted, the average rating of this movie will be re-calculated and updated.



User Reviews

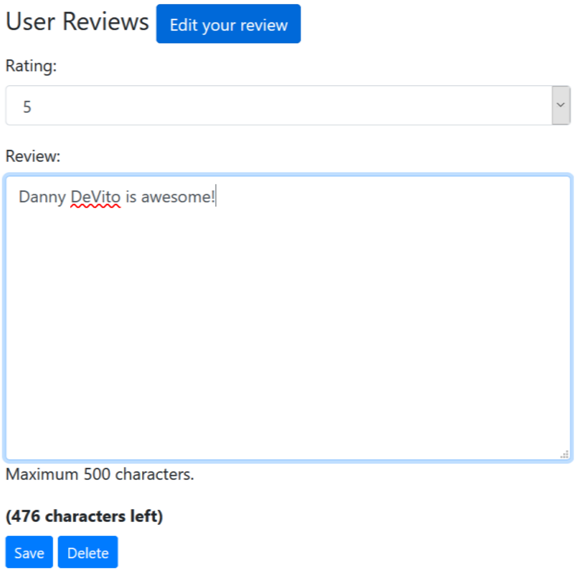
Rating: 5

Review: Danny DeVito!!!

Maximum 500 characters.

(485 characters left)

A user is also able to edit their previously submitted review by clicking on the “Edit your review” button which replaces the “Leave your review” button after the user has submitted a review.



User Reviews

Rating: 5

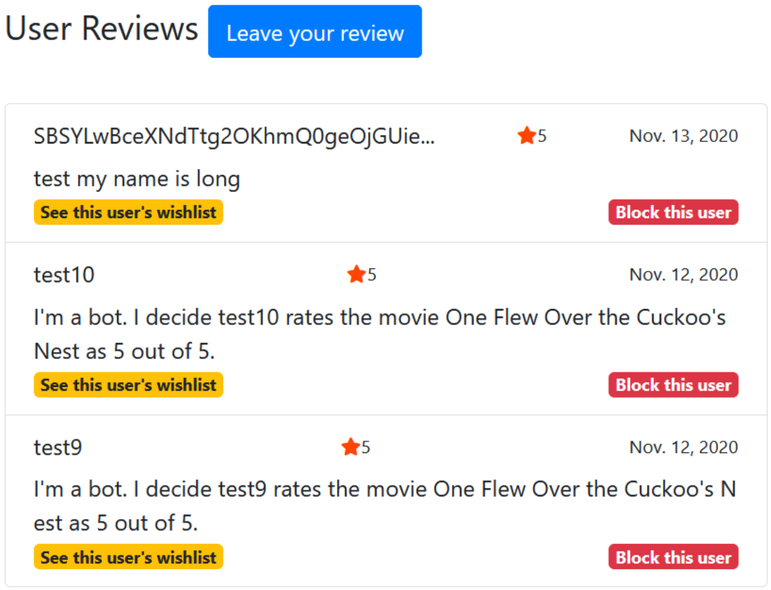
Review: Danny DeVito is awesome!

Maximum 500 characters.

(476 characters left)

5.9. Blocklist function

Under every user’s review (except for the current user), there is a button called “Block this user”. For example, the current user really likes a movie, but another user does not. They hold two totally different opinions. The current user might think this user’s opinions may have a negative impact on their FilmFinder experience. The current user can block this other user and add their username to the current user’s blocklist. This will prevent the user from seeing the blocked user’s reviews and the average rating will not consider the blocked user’s rating in its calculation. The figures below show the reviews before and after blocking a user.



SBSYLwBceXNdTtg2OKhmQ0geOjGUie... ★5 Nov. 13, 2020

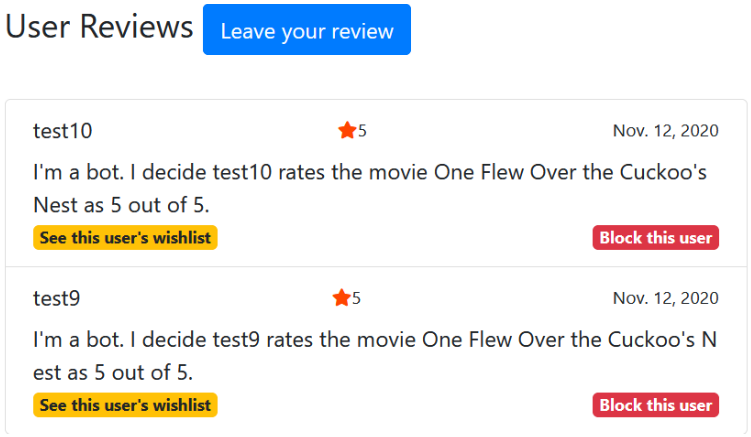
test my name is long

test10 ★5 Nov. 12, 2020

I'm a bot. I decide test10 rates the movie One Flew Over the Cuckoo's Nest as 5 out of 5.

test9 ★5 Nov. 12, 2020

I'm a bot. I decide test9 rates the movie One Flew Over the Cuckoo's N est as 5 out of 5.



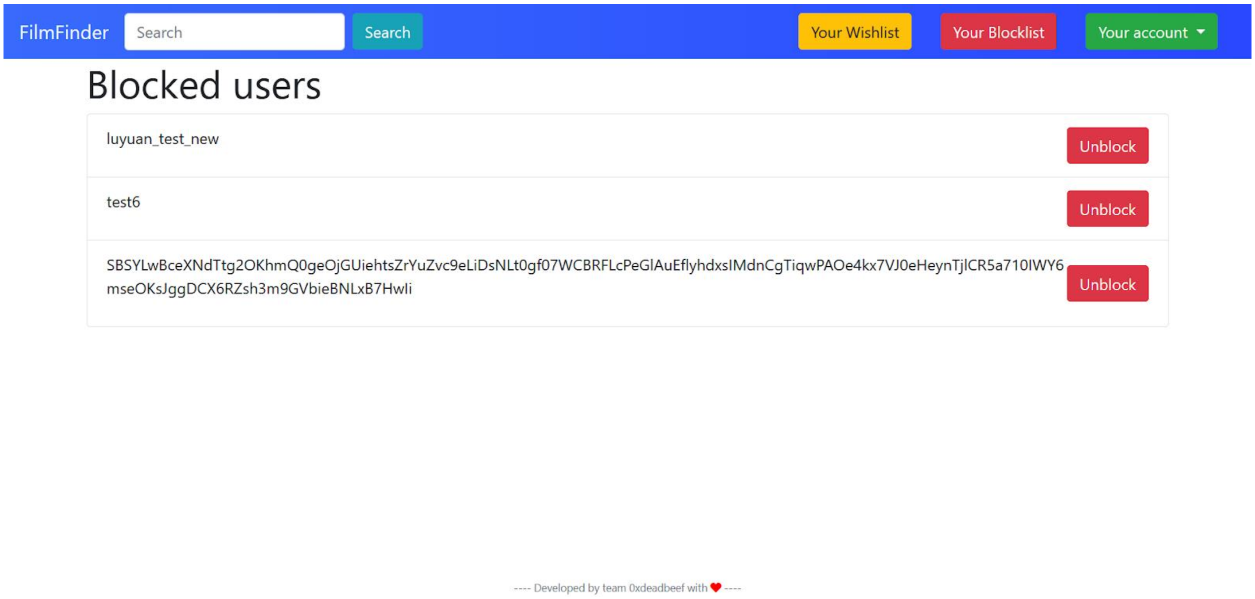
test10 ★5 Nov. 12, 2020

I'm a bot. I decide test10 rates the movie One Flew Over the Cuckoo's Nest as 5 out of 5.

test9 ★5 Nov. 12, 2020

I'm a bot. I decide test9 rates the movie One Flew Over the Cuckoo's N est as 5 out of 5.

The user can also navigate to the block list to view all the users they have blocked by clicking on the “Your Blocklist” button at the top of the page on the navigation bar. They may also unblock a user on the list if they so choose by clicking the “Unblock” button to the right of the blocked user’s username.



Blocked users

luyuan_test_new

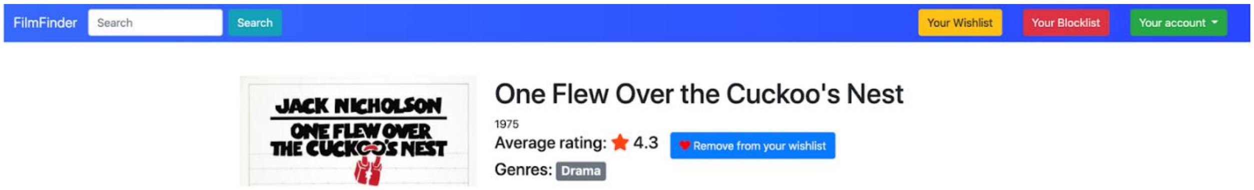
test6

SBSYLwBceXNdTtg2OKhmQ0geOjGUiehtsZrYuZvc9eLiDsNLt0gf07WCBRFLcPeGlAuEflyhdxsIMdnCgTiqwPAOe4kx7VJ0eHeynTjlCR5a710IWY6mseOKsJggDCX6RZsh3m9GVbieBNLxB7Hwli

---- Developed by team 0xdeadbeef with ♥ ----

5.10. Wishlist function

Next to the average rating, there is a “Add to your wishlist” button. If users want to watch this movie later, they can add it onto the wishlist, making it convenient for users to check it out later. After users have added it, the button will switch to “Remove from your wishlist” automatically. Users can remove the movie by clicking it.



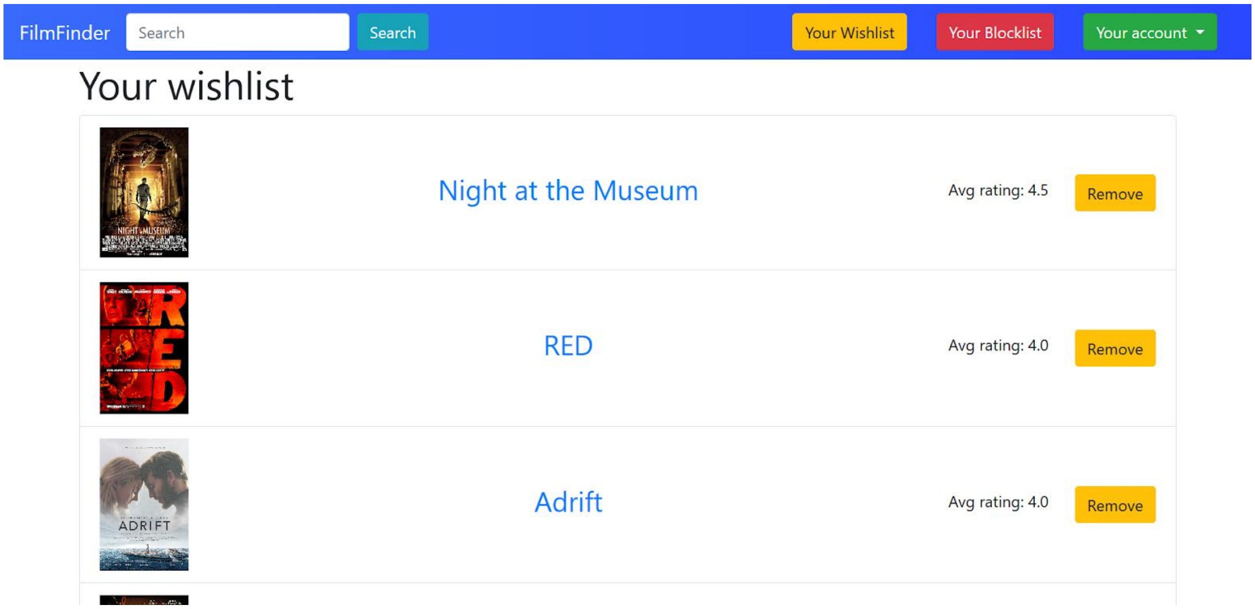
One Flew Over the Cuckoo's Nest

1975

Average rating: ★ 4.3

Genres: Drama

A user can navigate to their wish list to view all other movies on the wish list by clicking on the "Your Wishlist" button at the top of the page on the navigation bar. They may also remove a movie from the wish list on that page by clicking the "Remove" button next to the movie's average rating.

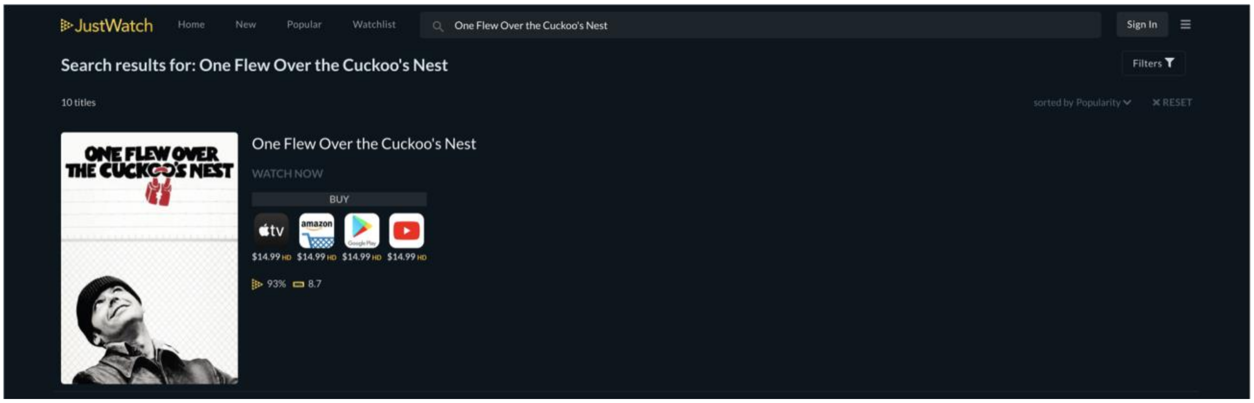


Your wishlist

Night at the Museum	Avg rating: 4.5	Remove
RED	Avg rating: 4.0	Remove
Adrift	Avg rating: 4.0	Remove

5.11. Watch it now

If users want to watch this movie online, they can try "watch it now", which will lead users to JustWatch website. They can check where this movie is available for viewing.



Search results for: One Flew Over the Cuckoo's Nest

10 titles

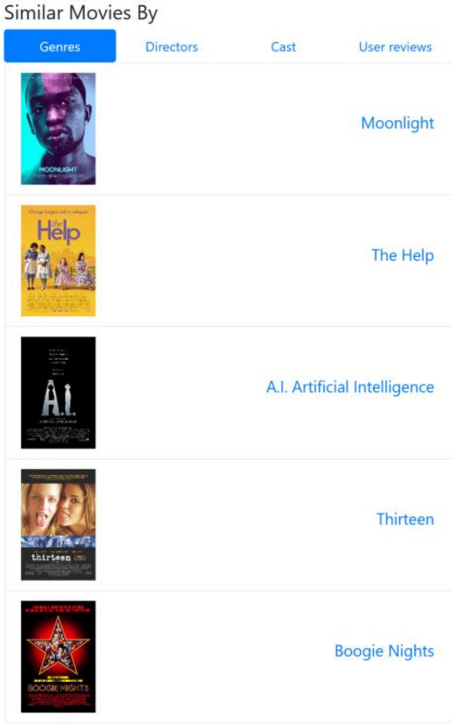
One Flew Over the Cuckoo's Nest

$14.99 HD $14.99 HD $14.99 HD $14.99 HD

93% 8.7

5.12. Movie recommendation

According to the current movie a user is browsing, we recommend some similar movies to the user based on genres, directors, cast and user reviews. The user can easily seek out some similar movies based on the recommendations.



Similar Movies By

Genres | Directors | Cast | User reviews

Moonlight

The Help

A.I. Artificial Intelligence

Thirteen

Boogie Nights

References

"Documentation." *Django*, docs.djangoproject.com/en/3.1/.

"10 Custom Bootstrap Navbar Examples." *Ordinary Coders*, 4 Aug. 2020, www.ordinarycoders.com/blog/article/codepen-custom-bootstrap-navbars.

Dianes, Jose A. "Building Data Products with Python: A Wine Review Website Using Django and Bootstrap." *Codementor*, 1 Sept. 2015, www.codementor.io/@jadianes/get-started-with-django-building-recommendation-review-app-du107yb1a.

Ghosh, Arun. "Redirect to Next after Login in Django." *Stack Overflow*, 18 July 2016, stackoverflow.com/a/38431227.

"A Guide to User Registration, Login, and Logout in Django." *Ordinary Coders*, 12 Aug. 2020, www.ordinarycoders.com/blog/article/django-user-register-login-logout.

"How to Use Django Messages Framework." *Ordinary Coders*, 2 June 2020, www.ordinarycoders.com/blog/article/django-messages-framework.

"How to Use the Extends and Include Django Template Tags." *Ordinary Coders*, 20 Apr. 2020, www.ordinarycoders.com/blog/article/extend-and-include-django-template-tags.

"Improve Signup Rates with Django-Allauth." *Ordinary Coders*, 29 June 2020, www.ordinarycoders.com/blog/article/django-allauth.

Katigaridi, Katerina. "Django-Allauth Tutorial (Part 1): Setup & Facebook Login (Upd. August 2020)." *Django Investigations*, 17 Oct. 2020, djangokatya.wordpress.com/2020/08/12/another-django-all-auth-tutorial/.

Reddy, Sathwick. "Implementing a Recommendation System on IMDB Dataset through Machine Learning Techniques." *Medium*, Medium, 20 June 2020, medium.com/@sr7037/implementing-a-recommendation-system-on-imdb-dataset-through-machine-learning-techniques-47d0a86da9df.

"Render a Django Bootstrap Form." *Ordinary Coders*, 21 Apr. 2020, www.ordinarycoders.com/blog/article/render-a-django-form-with-bootstrap.

"Reset User Passwords in Django." *Ordinary Coders*, 5 June 2020, www.ordinarycoders.com/blog/article/django-password-reset.

Vazquez-AbramsIgnacio, Ignacio. "How to Send Email via Django?" *Stack Overflow*, 16 June 2011, stackoverflow.com/a/6367023.