

CS699 Project

Streaming Playlist

22M0828 (Dhruv)
22M2117 (Anmol)

Description:

As movie streaming is becoming more and more popular, we are seeing more and more players entering the market. Due to this, movies and tv shows are divided amongst various streaming platforms like Netflix, Amazon Prime Video, Hotstar etc. Due to this it is getting harder to find which content is available at which streaming platform. Also, as content is being divided amongst these services, there is no way to create a playlist containing content from different platforms.

Aim: We aim to create a web application through which users can know which content is available on which platform, and create custom playlists which can also be shared with the world.

What did we achieve:

- User management.
- TMDb APIs to search for Movies and TV Shows.
- Playlists containing any combination of movies, TV shows or specific episodes.
- Managing Playlist: option to add/delete/shuffle/share playlists.
- Everyone can view all public playlists.
- Know which streaming services is providing specific movie/series.
- Brief description of a particular movie including its casts and genre content.

Future work:

- Making the pages more interactive and visually appealing.
- Strict user control over modification policies of the playlists.
- Option to have a private playlists.
- Play the content directly from the playlist.
- Integrating OMDb API for get details of even more content on the web.

Documentation of directories, files, main libraries/functions

- `manage.py` : is the main django file to start the server, serve the files online.
- `home-app`: control the index page of the website, displays latest movies/shows.
- `playlist-app`: control playlist management (Creation, Updating, Deletion).
- `search-app`: search through the TMDb dataset to give the required results.
- `templates-folder`: using django templating to control the responses of the html pages.
- `users-app`: user management system (Sign in, Sign up, Sign out etc).

Documentation of algorithms (if any)

No such hard named algorithm were used.

It is a simple python-based server which uses the capabilities of the APIs and the libraries which it provides to create a single-landing platform which can display the details of all the various types of movies and TV shows available online on different platforms.

And indeed provide with functionalities to create and share playlists which contains content from various platforms.

Tools used:

- Python
- HTML/CSS
- GIT
- Postgres
- VS code
- Bash
- Makefile

Compilation/running instructions

For first-time:

- `pip install -r requirements.txt` (Install dependencies use Conda env)
- `python manage.py createsuperuser` (Make admin user)
- `python manage.py migrate` (To make tables in the db)
- `python manage.py runserver` (Start the server)

Else:

- `python manage.py runserver`

Ending Notes:

We were just trying and playing around with the tools learned in software lab to make meaningful web applications which provides flexibility and cool functionality to our users.

We learned how python can be used as a backend framework for web servers.

We learned more about how REST API works as most of the communication on www happens through various APIs.

We learned about templating and how can we use that to control html pages to render the required content on the browser.

Thank You



Trending Movies



Black Adam (2022)

Rating : 6.8



Smile (2022)

Rating : 6.9



Black Panther: Wakanda
Forever (2022)

Rating : 7.5



Sniper: The White Raven
(2022)

Rating : 7.4



Terrifier 2 (2022)

Rating :

Trending TV



The Walking Dead (2010)

Rating : 8.1



Chucky (2021)

Rating : 7.9



La Reina del Sur (2011)

Rating : 7.8



House of the Dragon (2022)

Rating : 8.5



Now what (2022)

Rating : 4



Sign Up

Already have an account? Then please [sign in](#).

E-mail*

Username*

Password*

Password (again)*



Successfully signed in as dhruv.



dhruv

My Info

E-Mail

Searching for Harry

Found 40 results

Movies (20 results)



Harry Potter and the
Philosopher's Stone (2001)

Rating : 7.9



Harry Potter and the Half-
Blood Prince (2009)

Rating : 7.7



Harry Potter and the Goblet of
Fire (2005)

Rating : 7.8



Harry Potter and the Chamber
of Secrets (2002)

Rating : 7.7



Harry Potter and the
Prisoner of Azkaban (2004)

Rating : 7.7

[Back to season](#)[Stream at:](#)

Hotstar

Episode 6: The Princess and the Queen 2022

[Add to Playlist](#)

- [Overview](#)
- [Cast & Crew](#)

Ten years later. Rhaenyra navigates Alicent's continued speculation about her children, while Daemon and Laena weigh an offer in Pentos.

cast

[Full Cast & Crew](#)

**Paddy
Considine**

King Viserys
I Targaryen



Matt Smith

Prince
Daemon
Targaryen



**Emma
D'Arcy**

Princess
Rhaenyra
Targaryen



**Olivia
Cooke**

Alicent
Hightower



Rhys Ifan

Ser Otto
Hightower

Director:[Miguel Sapochnik](#)**Writer:**[Sara Hess,](#)**Stars:**[Paddy Considine,](#) [Matt Smith,](#) [Emma D'Arcy,](#)
[Olivia Cooke,](#)**Genres:**[Sci-Fi & Fantasy,](#) [Drama,](#) [Action & Adventure,](#)**Air Date:**

2022-09-25



Create New Playlist

1: my_fav

2: new

[Create New Playlist](#)

my_fav

[Shuffle Playlist](#)[Delete Playlist](#)**1: Episode 6: The Princess and the Queen (House of the Dragon - Season 1) (2022)**

Ten years later. Rhaenyra navigates Alicent's continued speculation about her children, while Daemon and Laena weigh an offer in Pentos.

Stream at: Hotstar[Delete](#)**2: Episode 10: The Black Queen (House of the Dragon - Season 1) (2022)**

Mourning a tragic death, Rhaenyra tries to hold the kingdom together and Daemon prepares for war.

Stream at: Hotstar[Delete](#)**3: Episode 4: King of the Narrow Sea (House of the Dragon - Season 1) (2022)**

Daemon returns to court after the successful completion of the battle. Princess Rhaenyra must deal with potential suitors in order to keep her line of succession.

Stream at: Hotstar[Delete](#)

CS699 Project

Streaming Playlist

22M0828 (Dhruv), 22M2117 (Anmol)

Description:

As movie streaming is becoming more and more popular, we are seeing more and more players entering the market. Due to this, movies and tv shows are divided amongst various streaming platforms like Netflix, Amazon Prime Video, Hotstar etc. Due to this it is getting harder to find which content is available at which streaming platform. Also, as content is being divided amongst these services, there is no way to create a playlist containing content from different platforms.

Aim:

We aim to create a web application through which users can know which content is available on which platform, and create custom playlists which can also be shared with the world.

Motivation:

This project was conceptualized when we noticed that very low information is available online about where to find whichever content we want to see, and there was no way to create rewatchable playlists which we can see whenever we want. Even though some platform-finding services exist, they cater mostly to the US and European markets, often ignoring the Indian market entirely. We wanted to create something which solves this problem in the Indian context.

Example Use Case:

A person likes watching sitcom TV Shows, with his favourites being F.R.I.E.N.D.S and How I Met Your Mother. Sometimes he just wants to watch any random episode of either of them without any effort. As both of these TV Shows exists on different streaming platforms (Netflix and Hotstar resp.), There is no way to achieve this without manually selecting any random episode.

With our project, he can just create a playlist consisting of his favourite sitcoms, and play episodes randomly, or through any of the other sorting mechanisms we may use like ratings, episode length, popularity etc.

What did we achieve:

- User management.
- TMDb APIs to search for Movies and TV Shows.
- Playlists containing any combination of movies, TV shows or specific episodes.
- Managing Playlist: option to add/delete/shuffle/share playlists.
- Everyone can view all public playlists.
- Know which streaming services is providing specific movie/series.
- Brief description of a particular movie including its casts and genre content.

Future work:

- Making the pages more interactive and visually appealing.
- Strict user control over modification policies of the playlists.
- Option to have a private playlists.
- Play the content directly from the playlist.
- Integrating OMDb API for get details of even more content on the web.

Documentation of directories, files, main libraries/functions:

- manage.py : is the main django file to start the server, serve the files online.
- home-app: control the index page of the website, displays latest movies/shows.
- playlist-app: control playlist management (Creation, Updating, Deletion).
- search-app: search through the TMDb dataset to give the required results.
- templates-folder: using django templating to control the responses of the html pages.
- users-app: user management system (Sign in, Sign up, Sign out etc).

Documentation of algorithms (if any):

No such hard named algorithm were used.

It is a simple python-based server which uses the capabilities of the APIs and the libraries which it provides to create a single-landing platform which can display the details of all the various types of movies and TV shows available online on different platforms.

And indeed provide with functionalities to create and share playlists which contains content from various platforms.

Tools used:

- Python
- HTML/CSS
- GIT
- Postgres
- VS code
- Bash
- Makefile

Compilation/running instructions:

For first-time:

- `pip install -r requirements.txt` (Install dependencies use Conda env)
- `python manage.py createsuperuser` (Make admin user)
- `python manage.py migrate` (To make tables in the db)
- `python manage.py runserver` (Start the server)

Else:

- `python manage.py runserver`

Ending Notes:

We were just trying and playing around with the tools learned in software lab to make meaningful web applications which provides flexibility and cool functionality to our users.

We learned how python can be used as a backend framework for web servers.

We learned more about how REST API works as most of the communication on www happens through various APIs.

We learned about templating and how can we use that to control html pages to render the required content on the browser.

Thank You