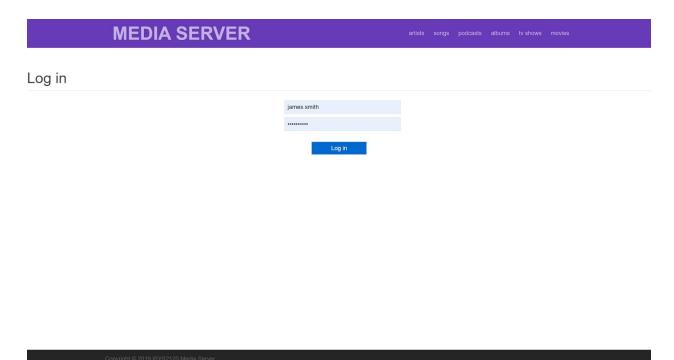
# ISYS2120 Group2 Assignment 3

# Report

# **Group members**

Zhanqiu Chen Chongbo Hou Cheng Zeng Robert Bai 1. User login by using username and password



- (a) User subscribed Podcasts
- (b) User Playlists
- (c) User current in-progress items

# Welcome, james.smith

### Your playlists

Playlist ID	Playlist Name	Items
1	james.smith's random movie collection	50
2	james.smith's random song collection	30
3	james.smith's random podcast episode collection	357
4	james.smith's random tv episode collection	64
5	iames.smith's random collection of media	42

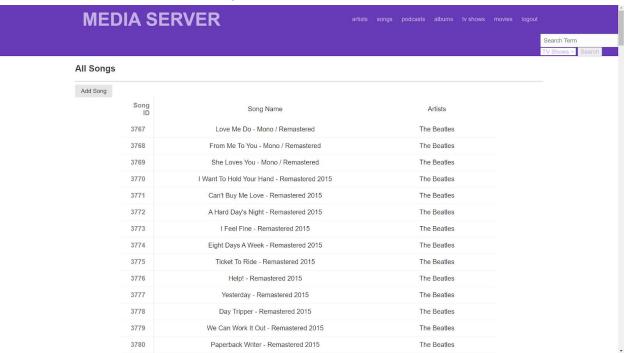
#### **Your Podcasts**

Podcast ID	Title	URI	Last Updated
2	Conversations	http://www.abc.net.au/local/podcasts/conversationspodcast.xml	2019- 10-25
3	Hamish & Andy	https://www.omnycontent.com/d/playlist/820f09cf-2ace-4180-a92d- aa4c0008f5fb/0786e9ff-0519-43d3-a0cd-aa4c0176ad25/2bf97b40-59cd- 424a-ba35-aa4c0176ad25/podcast.rss	2019- 10-23
6	Bad Batch	https://rss.art19.com/bad-batch	2019- 10-25
7	A Perfect Storm: The True Story of The Chamberlains	https://rss.acast.com/a-perfect-storm	2019- 10-24
11	Dr Karl Podcast	http://www.abc.net.au/science/k2/podcast/drk_rss.xml	2019- 10-24
12	TED Talks Daily	http://feeds.feedburner.com/TEDTalks_audio	2019- 10-25
13	Office Ladies	https://rss.art19.com/office-ladies	2019- 10-23

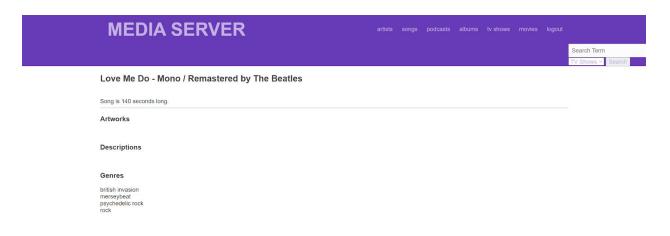
#### Your most recent viewed items

Media ID	Playcount	Progress	Lastviewed	Storage Location
3891		95.19	2019-10- 23	https://p.scdn.co/mp3-preview/9670fda6c30abbced8a997fa42528c16a85952e4? cid=118e90cc8bcc44758238928a7b3e60b3
3816		33.56	2019-10- 22	https://p.scdn.co/mp3-preview/190abd8904d0c9b01567195f4cc36572bac4e01e? cid=118e90cc8bcc44758238928a7b3e60b3
2526		38.68	2019-10- 22	http://traffic.libsyn.com/joeroganexp/p721.mp3?dest-id=19997
890		90.34	2019-10- 21	/files/the-office-(us)-s04-e15.mp4
3935		66.45	2019-10- 21	https://p.scdn.co/mp3-preview/e2f5edb569c73916428ec0a2e0b56a9f777851dd? cid=118e90cc8bcc44758238928a7b3e60b3
3071		51.64	2019-10- 21	http://traffic.libsyn.com/joeroganexp/p210.mp3?dest-id=19997
2069		68.71	2019-10- 20	http://traffic.libsyn.com/joeroganexp/p1120.mp3?dest-id=19997
			2019-10-	https://p.scdn.co/mp3-preview/60cd920247214f6d4bf7d8b0dc8c64f8e8a04e042

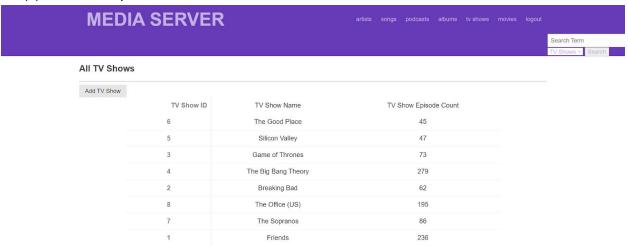
# 2. When a user clicks on a song



- (a) Song name
- (b) Song Artists
- (c) Song length
- (d) Song Artworks, Descriptions, Genres



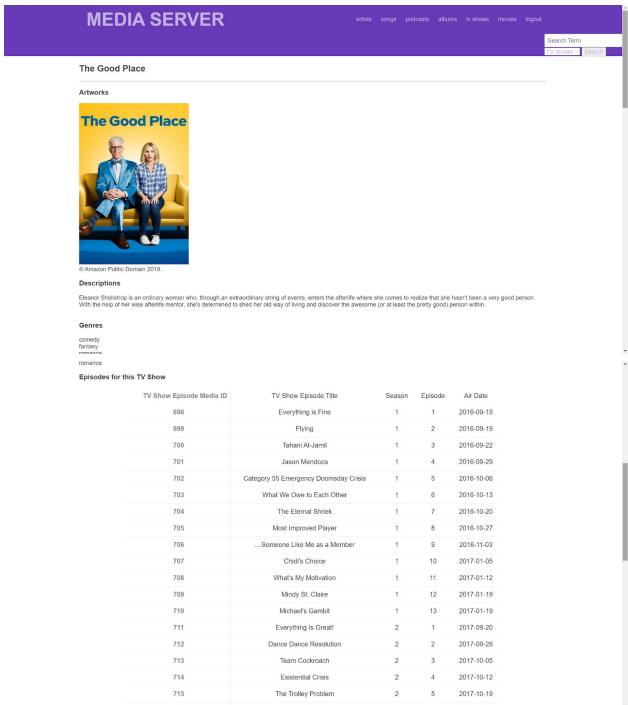
- 3. When a user clicks on the 'tv shows' nav item
  - (a) TV Show ID
  - (b) TV Show NAME
  - (c) TV Show Episode Count



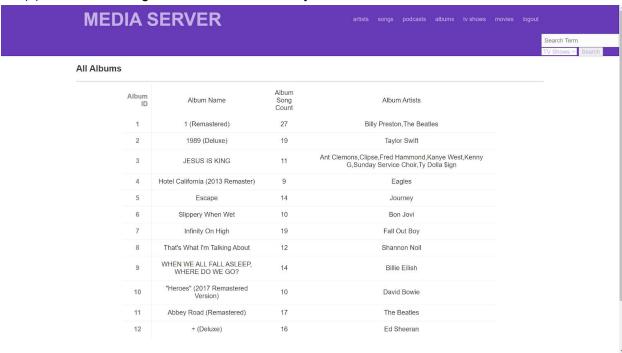
opyright © 2019 ISYS2120 Media Server

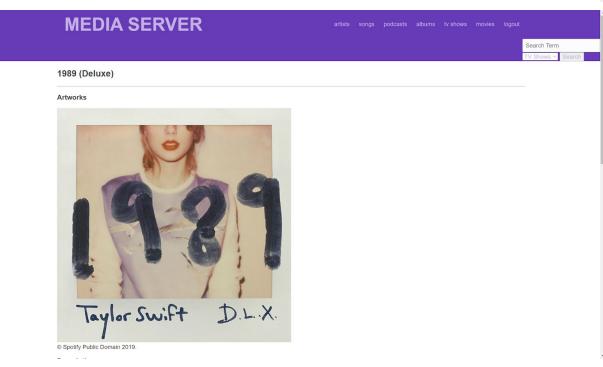
# 4. When a user clicks on a single tv show

- (a) TV Show Name
- (b) TV Show Artworks, Descriptions, Genres



- 5. When a user clicks on a single Album
  - (a) Album Name
  - (b) Album Artworks, Descriptions
  - (c) A list of all songs in an album ordered by track number





Song ID	Song Title	Artists
3794	Welcome To New York	Taylor Swift
3795	Blank Space	Taylor Swift
3796	Style	Taylor Swift
3797	Out Of The Woods	Taylor Swift
3798	All You Had To Do Was Stay	Taylor Swift
3799	Shake It Off	Taylor Swift
3800	I Wish You Would	Taylor Swift
3801	Bad Blood	Taylor Swift
3802	Wildest Dreams	Taylor Swift
3803	How You Get The Girl	Taylor Swift
3804	This Love	Taylor Swift
3805	I Know Places	Taylor Swift
3806	Clean	Taylor Swift
3807	Wonderland	Taylor Swift
3808	You Are In Love	Taylor Swift
3809	New Romantics	Taylor Swift
3810	l Know Places - Voice Memo	Taylor Swift

Artists

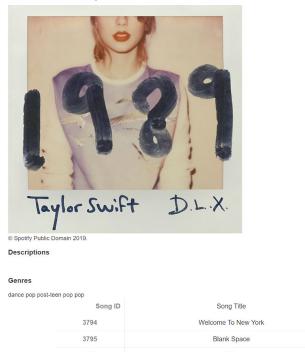
Taylor Swift

Taylor Swift

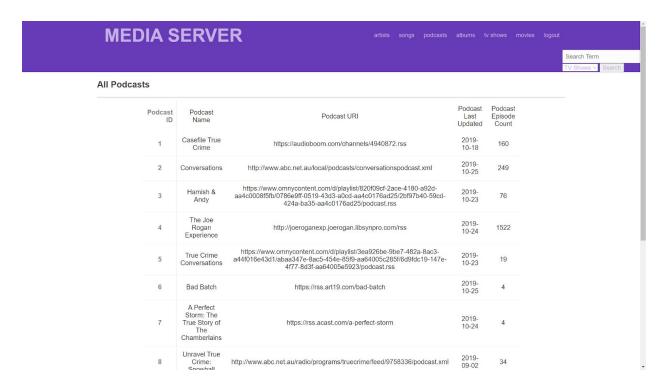
6. A user can see all the genres for an Album in part (5)

Descriptions

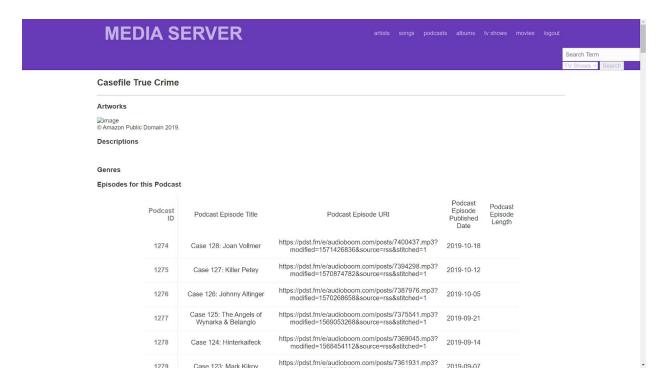
Genres



- 7. When a user clicks on a single podcast from the 'podcasts' list
  - (a) Podcast ID
  - (b) Podcast Name
  - (c) Podcast URI
  - (d) Last Updated



- (e) Podcast Artworks, Descriptions, Genres
- (f) A list of all podcast episode in this podcast ordered by descending publication date



- 8. When a user clicks on a single podcast episode from the list in part (7)
  - (a) Podcast Episode ID
  - (b) Podcast Episode Title
  - (c) Podcast Episode URI
  - (d) Podcast Episode Date Published
  - (e) Podcast Episode Length
  - (f) Podcast Episode Artworks, Descriptions, Genres

# **MEDIA SERVER**

artists songs podcasts albums tv shows movi

Podcast ID episode title: Case 128: Joan Vollmer published on 2019-10-18

Length 4215

 $\label{lem:url:mp3:modified=1571426836&source=rss\&stitched=1} In the content of the content of$ 





© Amazon Public Domain 2019.

#### Description

On Sunday, January 10 1993, 54-year-old pig farmer Raiph Vollimer and his 49-year-old wife Joan returned home from a week-long interstate trip...\nEpisode narrated by the Anonymous Host (https://witter.com/casefile/host)/Episode researched and written by Erin Murro in/Creative Director. Milly Raiso (https://millp/cuseraso.com/in/This episode' sponsors/si-HerperJusion (https://www.herperJusion.com/i) — Get 20% OFF your first order and FFRES shipping with prome code CASEFILE\*n/Lip/Recynding (https://www.ziprerulier.com/casefile) — Post your jobs for FREE/n/hative (http://www.ziprerulier.com/casefile) — Post your jobs for FREE/n/hative (http://www.ziprerulier.com/) — Get 20% of Fry your first order with promo code CASEFILE\*n/hative (http://www.ziprerulier.com/) — Get 20% of Fry your first order with promo code CASEFILE\*n/hative (http://www.ziprerulier.com/) — Get 20% of Fry your first order with promo code CASEFILE\*n/hative (http://www.ziprerulier.com/) — Get 20% of Fry your first order with promo code CASEFILE\*n/hative (http://www.ziprerulier.com/) — Get 20% of Fry your first order with promo code CASEFILE\*n/hative (http://www.ziprerulier.com/) — Get 20% of Fry your first order with promo code CASEFILE\*n/hative (http://www.ziprerulier.com/) — Get 20% of Fry your first order with promo code CASEFILE\*n/hative (http://www.ziprerulier.com/) — Get 20% of Fry your first order with promo code CASEFILE\*n/hative (http://www.ziprerulier.com/) — Get 20% of Fry your first order with promo code CASEFILE\*n/hative (http://www.

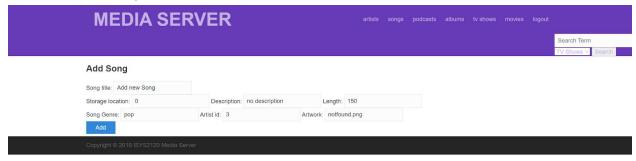
Copyright

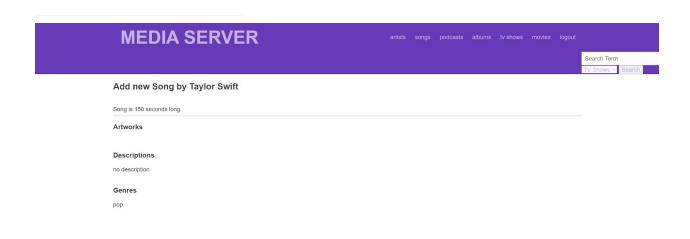
Genre

The website then displays the metadata associated with this episode.

Podcast.html and podcastep.html is modified to link the podcast page to the podcast episode page, and also made it possible to display all the meta data for each episodes.

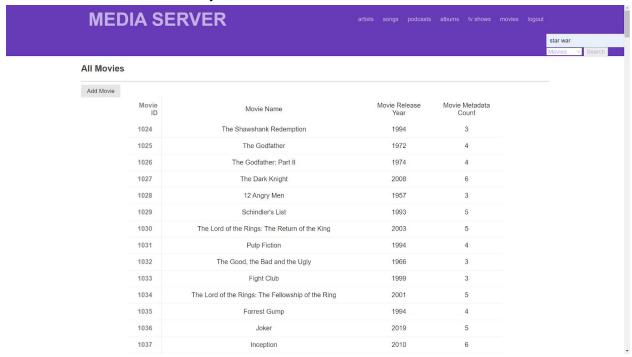
# 9. Insert a new Song





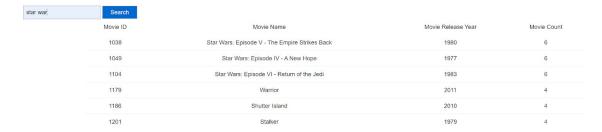
This will add a new song to the database by the user

# 10. A user can search movies by their titles.





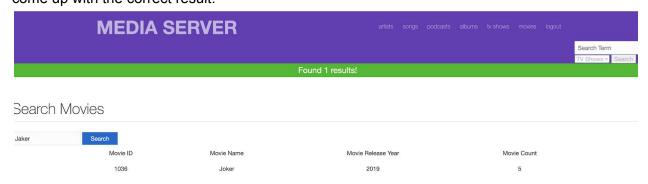
# Search Movies



# New functionalities of the project

We added three new search functions (search song, search podcast, search artist).

Also, we have implemented fuzzy Search, and have modified pre-existing(search TV shows and search Movies) and new Search() functions to represent the fuzzy search function in action. For example, if a user wants to search for Joker, but misspells it as "Jaker", the search will still come up with the correct result:



All searches(song, podcast, artist, movie and TV show) support fuzzy search. We use SOUNDEX and SIMILARITY function in public schema to implement fuzzy search.

The new search functions:

#### **Podcast**

We have modifies files and functions as below:

1, Add Podcast:

listpodcasts.html

An HTML page to display all the podcasts

createpodcast.html

An HTML page to add a single podcast to the database

add podcast to db() in database.py

The python code to connect the SQL code to the connection cursor

add\_podcast() in routes.py

#### 2. Search Podcast:

search\_podcasts.html

An html page that allows searching for podcasts

find matchingpodcasts() in database.py

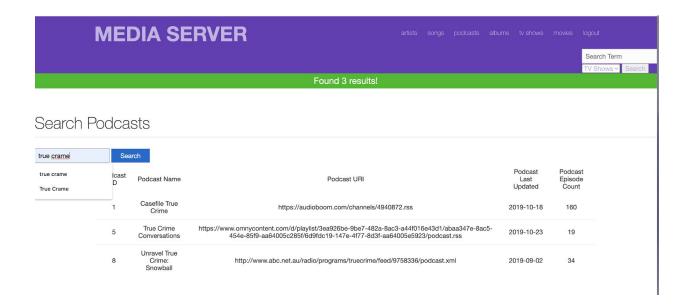
SQL query inside python to execute the query using a cursor

search\_podcasts() in routes.py
Python code to link to the database and extract the search term entered by the user

get\_allpodcasts() in database.py SQL query that executes as a python cursor to display all the podcasts

Top.html to add the podcast option

						True <u>Cra</u>
All Podcast	s					true crame
F	Podcast ID	Podcast Name	Podcast URI	Podcast Last Updated	Podcast Episode Count	
	1	Casefile True Crime	https://audioboom.com/channels/4940872.rss	2019- 10-18	160	
	2	Conversations	http://www.abc.net.au/local/podcasts/conversationspodcast.xml	2019- 10-25	249	
	3	Hamish & Andy	https://www.omnycontent.com/d/playlist/820f09cf-2ace-4180-a92d-aa4c0008f5fb/0786e9ff-0519-43d3-a0cd-aa4c0176ad25/2bf97b40-59cd-424a-ba35-aa4c0176ad25/podcast.rss	2019- 10-23	76	
	4	The Joe Rogan Experience	http://joeroganexp.joerogan.libsynpro.com/rss	2019- 10-24	1522	
	5	True Crime Conversations	https://www.omnycontent.com/d/playlist/3ea926be-9be7-482a-8ac3- a44f016e43d1/abaa347e-8ac5-454e-85f9-aa64005c285f/6d9fdc19-147e- 4f77-8d3f-aa64005e5923/podcast.rss	2019- 10-23	19	
	6	Bad Batch	https://rss.art19.com/bad-batch	2019- 10-25	4	
	7	A Perfect Storm: The True Story of The Chamberlains	https://rss.acast.com/a-perfect-storm	2019- 10-24	4	
	8	Unravel True Crime: Snowball	http://www.abc.net.au/radio/programs/truecrime/feed/9758336/podcast.xml	2019- 09-02	34	
	9	Inside the Big Day Out	http://www.abc.net.au/radio/programs/inside-the-big-day-out/feed/11547726/podcast.xml	2019- 10-22	4	
	10	Full Story	https://www.theguardian.com/australia-news/series/full-story/podcast.xml	2019- 10-24	5	
	11	Dr Karl Podcast	http://www.abc.net.au/science/k2/podcast/drk_rss.xml	2019- 10-24	121	



### Artist

We have modifies files and functions as below:

1, add artist:

add\_artist\_to\_db() in database.py
The python code to connect the SQL code to the connection cursor
add\_artist() in routes.py

### 2, Search artist::

search\_artist.html
An html page that allows searching for artist

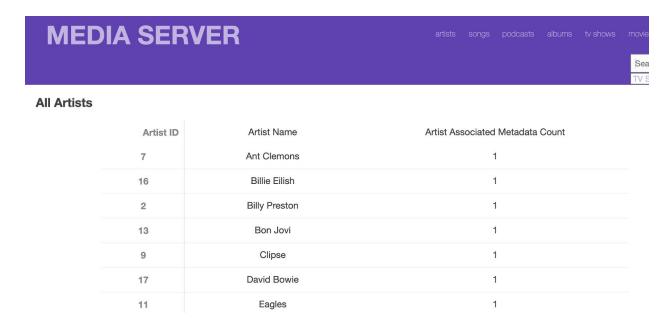
find\_matchingartists() in database.py
SQL query inside python to execute the query using a cursor

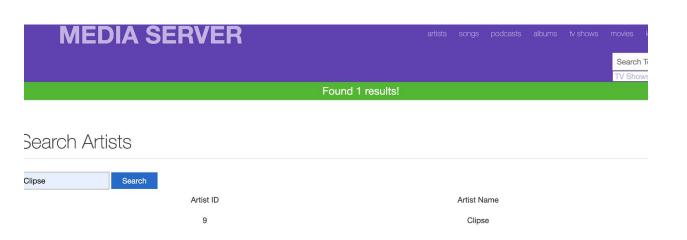
search\_artists() in routes.py

Python code to link to the database and extract the search term entered by the user

get\_allartists() in database.py SQL query that executes as a python cursor to display all the artist

Top.html to add the artist option





### Song

We have implemented the extension of the Search to provide the functions for the users to search songs by titles.

To achieve this function, we have modified some files and create some functions:

1, Search Song:

search\_songs.html
An html page that allows searching for songs

find\_matchingsongs() in database.py SQL query inside python to execute the query using a cursor

search\_songs() in routes.py
Python code to link to the database and extract the search term entered by the user

get\_allsongs() in database.py SQL query that executes as a python cursor to display all the songs

Top.html to add the song option

