Assignment

In this assignment, you will be implementing a simple web-app that consists of backend and frontend components.

Backend Component:

You will create a web application server by using Flask, which will have a single web-service named “ tracks “ This HTTP service will accept a music genre type as input and do the following:

1. Pick a random artist associated with that genre from the " genres.json " file, which will be provided.

2. Filter the most popular 10 tracks (songs) of that artist from the first 50 tracks obtained from Spotify Search API (eg. "rock" genre will retrieve the most popular 10 track info of "Led Zeppelin", which is randomly chosen).

API-doc of your “tracks” service:

HTTP Request Type:	GET
Description:	Returns a list of 10 tracks, sorted by popularity.
Query parameters:	genre: Name of the genre that is desired by the user.
URL Path:	http://IP_ADDR:PORT/tracks/{genre}
Response Content-Type:	application/json

Sample request-response:

Request URL:	http://localhost:8080/tracks/rock

Response:	[
	{
	"artist": "Led Zeppelin",
	"track": "Babe I’m Gonna Leave You",
	"album_image_url": "https://i.scdn.co/image/12b952dadc9614a8893be2eb517c6c5aea9f
	0ded",
	"preview_url": "https://p.scdn.co/mp3-preview/8455599677a13017978dcd3f4b2109
	37f0a16bcb?cid=67a9bc7ccdc2455fb6894e770fef2557",
	},
	...
	...
	{
	"artist": "Led Zeppelin",
	"track": "Kashmir - 1990 Remaster",
	"album_image_url": "https://i.scdn.co/image/28af597d86bf7bcbf9513695b04cf7047ad7a
	e1f",
	"preview_url": "https://p.scdn.co/mp3-preview/13b2c47c98132f80a5ce12d8a4ceb1
	92ba7a4345?cid=67a9bc7ccdc2455fb6894e770fef2557",
	}
	]

Notes:

1. Your web-service will need to generate an access token to use Spotify Search API . It is more convenient and easy to use Client Credentials Flow to generate a new access token for server-to-server authentication.
2. Don't waste your time judging the popularity for tracks by yourself, Spotify Search API provides that for you.

Front End Component:
You will implement a basic user interface on web-browser which will consume your " tracks "service. This UI will include a textbox & search button, where the user can type the genre. After you retrieve the 10-track list, you will visualize these tracks in a table showing the artist name, the track name, the image of the track’s album cover and a redirect link to the 30-seconds preview url, one track for each row.