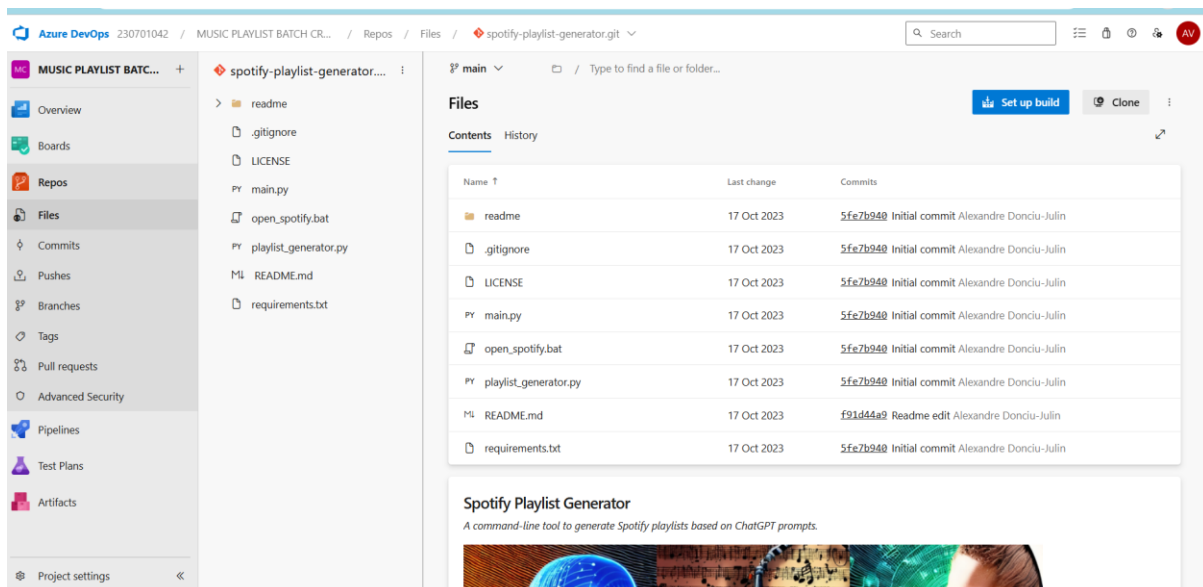
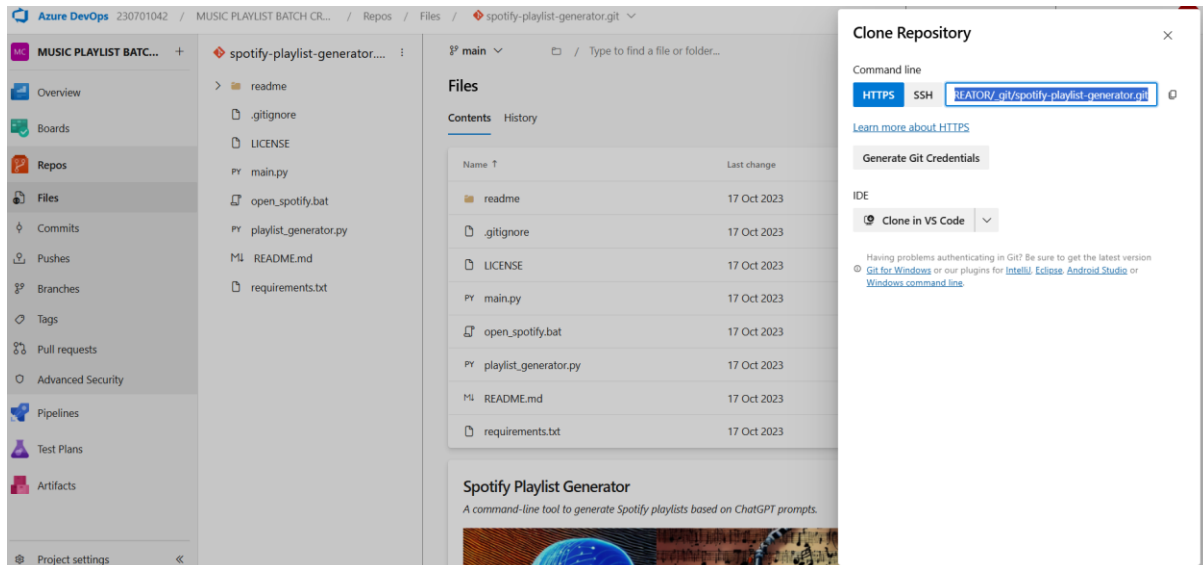


Ashna .V

230701042

Ex.no:7

7.)CLONE REPOSITORY (music playlist batch creator):



The image shows a GitHub repository page for 'spotify-playlist-generator'. The top navigation bar includes 'Azure DevOps', '230701042', and a breadcrumb trail: 'Music Playlist Batch CR...' / 'Repos' / 'Files' / 'spotify-playlist-generator.git'. A search bar is on the right. The left sidebar shows the repository's file structure: 'Overview', 'Boards', 'Repos', 'Files' (selected), 'Commits', 'Pushes', 'Branches', 'Tags', 'Pull requests', 'Advanced Security', 'Pipelines', 'Test Plans', and 'Artifacts'. The main content area has tabs for 'Contents' and 'History'. The 'Contents' tab is active, displaying the project title 'Spotify Playlist Generator', a description 'A command-line tool to generate Spotify playlists based on ChatGPT prompts.', and a featured image showing a blue digital head, headphones, and a woman's profile. Below the image is the 'Project description' section, which explains the project's goal: to learn how to interact with the OpenAI API and use the Spotify module to create playlists. It mentions a tutorial by 'Cait Steele: Mastering OpenAI Python APIs' and describes the project's features, including automatic playlist creation and the ability to blacklist artists. The 'Prerequisites' section is partially visible at the bottom, stating the need for an OpenAI and a Spotify Developer account.

MUSIC PLAYLIST BAC...

Overview

Boards

Repos

Files

Commits

Pushes

Branches

Tags

Pull requests

Advanced Security

Pipelines

Test Plans

Artifacts

Project settings

spotify-playlist-generator....

readme

.gitignore

LICENSE

main.py

open\_spotify.bat

playlist\_generator.py

README.md

requirements.txt

ContentsHistory

Prerequisites

To use this tool, you need both an OpenAI and a Spotify Developer account.

OpenAI

It's the company who created and offers access to different ChatGPT models. Go to the [OpenAI website](#) and create an account. It comes with some free credit to start and test the API. See [documentation](#) to generate your own API-Key.

Spotify

Create a [Spotify account](#) if you don't already have one. It's free. After that, login to [Spotify for Developers](#) and create a project. In the project settings, you can generate the client-ID and -Secret needed to connect and send requests to Spotify. See the [documentation](#) for more info.

Once you have your own credentials to access both APIs, you can add them to `playlist_generator.py`. I am using the [keyring](#) module to store and retrieve mine, but you can use an `env` file, store them in environment variables or just use them directly in the code. In this last case, be careful not to share your code with anyone!

```
# Set openai key
openai.api_key = keyring.get_password('openai_key', 'main_key')
# Set spotify client id and secret
client_id=keyring.get_password('spotify', 'client_ID')
client_secret=keyring.get_password('spotify', 'client_secret')
```

Spotify Desktop

If you want to be able to play songs during the playlist creation (in the interactive mode for instance), you will need [Spotify Desktop](#) installed on your machine. If you want to skip this part, comment all calls to the method `play_song_in_spotify()` in `playlist_generator.py`, as it will not be able to ask your Spotify player to play the given songs.

Installation

Clone project and install required libraries (in a virtual environment for instance)

```
git clone https://github.com/alexndjulin/spotify-playlist-generator.git
python -m venv venv
```

Azure DevOps

230701042 / MUSIC PLAYLIST BATCH CR... / Repos / Files / spotify-playlist-generator.git

MUSIC PLAYLIST BATC... +

Overview

Boards

Repos

Files

Commits

Pushes

Branches

Tags

Pull requests

Advanced Security

Pipelines

Test Plans

Artifacts

Project settings <

spotify-playlist-generator....

> readme

.gitignore

LICENSE

main.py

open\_spotify.bat

playlist\_generator.py

README.md

requirements.txt

Contents History

Clone project and install required libraries (in a virtual environment for instance)

```
git clone https://github.com/AlexJulin/spotify-playlist-generator.git  
python -m venv venv  
venv\Scripts\activate # on windows  
source venv/bin/activate # on macOS/linux  
pip install -r requirements.txt
```

How to use it

The most convenient way to start the playlist generator is to call main.py.

```
python main.py
```

It will ask for the 4 required input variables:

- Playlist Prompt (-p):**  
A short description sent to ChatGPT to generate the playlist. Example: *Best songs for a road trip with your best friend.*
- Playlist Length (-l):**  
How many songs should be in the playlist. Default is 10.
- Playlist Name (-n):**  
The playlist name on Spotify. Optional, leave blank to use the prompt as name. Example: *Road Trip Songs.*
- Interactive Mode (-i):**  
Y to activate or N to deactivate the interactive mode. When active, each song will be played in Spotify and the user can choose to Add the song, blacklist it or blacklist the artist.

```
Playlist Prompt: Best songs for a road trip with your best friend  
Playlist/Batch Length: 10  
Playlist Name (leave blank to use prompt): Road Trip Songs  
Interactive Mode? [y]yes or [n]: n  
Creating playlist. Please wait...  
>> end of playlist creation
```

Additionally you can call playlist\_generator.py and pass these arguments in the command line directly. Don't forget the quotes around the strings.

Clone project and install required libraries (in a virtual environment for instance)

```
git clone https://github.com/alexjdulin/spotify-playlist-generator.git
python -m venv venv
venv\Scripts\activate # on windows
source venv/bin/activate # on macOS/linux
pip install -r requirements.txt
```

### How to use it

The most convenient way to start the playlist generator is to call `main.py`

```
python main.py
```

It will ask for the 4 required input variables:

- **Playlist Prompt (-p):**  
A short description sent to ChatGPT to generate the playlist. Example: *Best songs for a road trip with your best friend.*
- **Playlist Length (-l):**  
How many songs should be in the playlist. Default is 10.
- **Playlist Name (-n):**  
The playlist name on Spotify. Optional, leave blank to use the prompt as name. Example: *Road Trip Songs.*
- **Interactive mode (-i):**  
Y to activate or N to deactivate the interactive mode. When active, each song will be played in Spotify and the user can choose to Add the song, blacklist it or blacklist the artist.

```
Playlist Prompt: Best songs for a road trip with your best friend
Playlist/Batch Length: 10
Playlist Name (leave blank to use prompt): Road Trip Songs
Interactive Mode? [y]es or [n]o: n
Creating playlist. Please wait...
>> end of playlist creation
```

Additionally you can call `playlist_generator.py` and pass these arguments in the command line directly. Don't forget the quotes around the strings.

Azure DevOps 230701042 / MUSIC PLAYLIST BATCH CR... / Repos / Files / spotify-playlist-generator.git

MUSIC PLAYLIST BATC... +

- Overview
- Boards
- Repos
- Files
- Commits
- Pushes
- Branches
- Tags
- Pull requests
- Advanced Security
- Pipelines
- Test Plans
- Artifacts
- Project settings

spotify-playlist-generator....

- readme
- .gitignore
- LICENSE
- main.py
- open\_spotify.bat
- playlist\_generator.py
- README.md
- requirements.txt

Contents History

Additionally you can call playlist\_generator.py and pass these arguments in the command line directly. Don't forget the quotes around the strings. The interactive mode flag is TRUE if you pass -i, FALSE if you don't pass it:

```
# Interactive flag off
python playlist_generator.py -p "Best songs for a road trip with your best friend" -l 10 -n "Road Trip Songs"
# Interactive flag on
python playlist_generator.py -p "Best songs for a road trip with your best friend" -l 10 -n "Road Trip Songs" -i
```

**Automatic Playlist Creation**

When interactive mode is off, the playlist is created and filled automatically. It is great to create quick playlists, without worrying too much about the contents... Providing that you trust ChatGPT's suggestions, which can be sometimes surprising!

Output when interactive mode is off:

```
----- SPOTIFY PLAYLIST GENERATOR -----
Prompt: Best songs for a road trip with your best friend
Length: 10
Name: Road Trip Songs
Interactive: False
-----
1. Journey - Don't Stop Believin' | Escape (Bonus Track Version)
2. Smash Mouth - All Star | Astro Lounge
3. Bruce Springsteen - Born to Run | Born To Run
4. Guns N' Roses - Sweet Child O' Mine | Appetite For Destruction
5. John Denver - Take Me Home, Country Roads - Original Version | Poems, Prayers and Promises
6. ABBA - Dancing Queen | Arrival
7. Bon Jovi - Livin' On A Prayer | Slippery When Wet
8. Queen - Bohemian Rhapsody | Bohemian Rhapsody (The Original Soundtrack)
9. Lynyrd Skynyrd - Sweet Home Alabama | Second Helping (Expanded Edition)
10. Whitney Houston - I Wanna Dance with Somebody (Who Loves Me) | Whitney
-----
```

Spotify should open automatically, the songs are added to the playlist and the first one from the list should play when done.

Azure DevOps 230701042 / MUSIC PLAYLIST BATCH CR... / Repos / Files / spotify-playlist-generator.git

MUSIC PLAYLIST BATC... +

- Overview
- Boards
- Repos
- Files
- Commits
- Pushes
- Branches
- Tags
- Pull requests
- Advanced Security
- Pipelines
- Test Plans
- Artifacts
- Project settings

spotify-playlist-generator....

- readme
- .gitignore
- LICENSE
- main.py
- open\_spotify.bat
- playlist\_generator.py
- README.md
- requirements.txt

Contents History

**\_Road Trip Songs**

AlexdJulin • 10 songs, 44 min 36 sec

#	Title	Album	Date added
1	Don't Stop Believin' Journey	Escape (Bonus Track Version)	2 hours ago
2	All Star Smash Mouth	Astro Lounge	2 hours ago
3	Born to Run Bruce Springsteen	Born To Run	2 hours ago
4	Sweet Child O' Mine Guns N' Roses	Appetite For Destruction	2 hours ago
5	Take Me Home, Country Roads - Original Version John Denver	Poems, Prayers and Promises	2 hours ago

Azure DevOps 230701042 / MUSIC PLAYLIST BATCH CR... / Repos / Files / spotify-playlist-generator.git

MUSIC PLAYLIST BATC... +

- Overview
- Boards
- Repos
- Files
- Commits
- Pushes
- Branches
- Tags
- Pull requests
- Advanced Security
- Pipelines
- Test Plans
- Artifacts
- Project settings

spotify-playlist-generator....

- readme
- .gitignore
- LICENSE
- main.py
- open\_spotify.bat
- playlist\_generator.py
- README.md
- requirements.txt

Contents History

**\_Rainy Mood**

AlexdJulin • 6 songs, 27 min 58 sec

#	Title	Album	Date added
1	Lonely Day System Of A Down	Hypnotize	42 seconds ago
2	Crying in the Rain a-ha	East of the Sun, West of the Moon	37 seconds ago
3	Rainy Night in Georgia Brook Benton	Brook Benton Today	27 seconds ago
4	Boulevard of Broken Dreams Green Day	Boulevard of Broken Dreams	23 seconds ago
5	Riders on the Storm The Doors	L.A. Woman	13 seconds ago
6	Everybody Hurts R.E.M.	Automatic For The People	9 seconds ago

MC MUSIC PLAYLIST BATC... +

Overview

Boards

Repos

Files

Commits

Pushes

Branches

Tags

Pull requests

Advanced Security

Pipelines

Test Plans

Artifacts

spotify-playlist-generator....

> readme

.gitignore

LICENSE

PY main.py

open\_spotify.bat

PY playlist\_generator.py

M↓ README.md

requirements.txt

