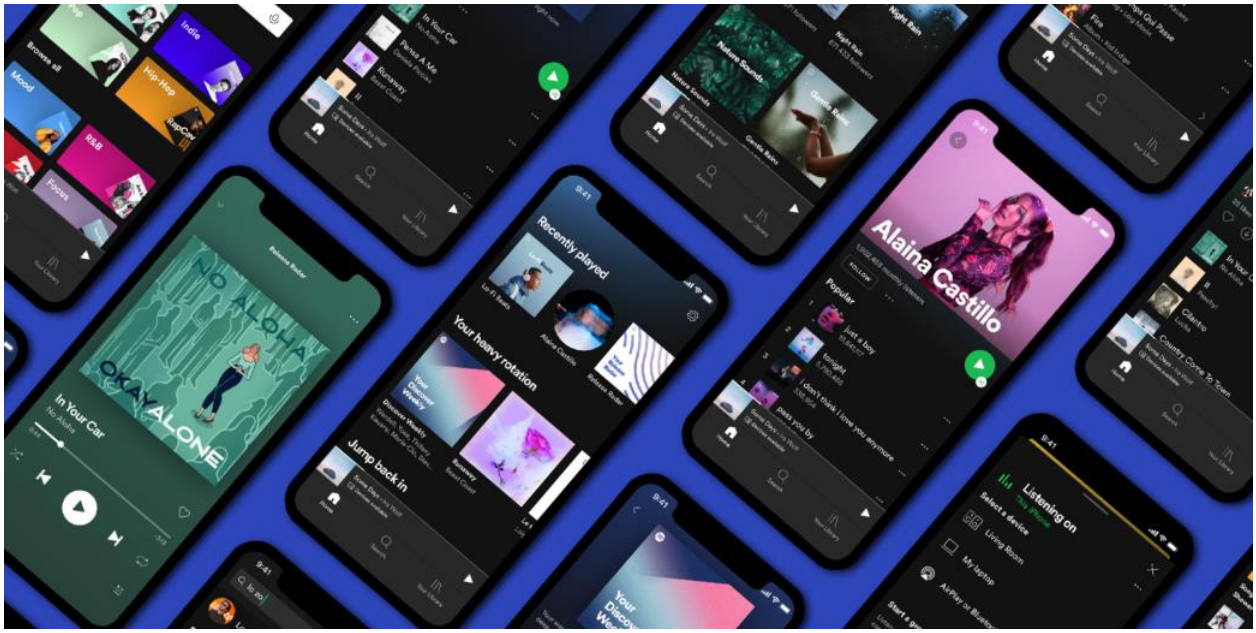
Data Structures

MP3 Music Player



This project would mimic a very famous media player app known as “spotify”, which is very common among youth for music playing.
We aim at creating your own music player in which you can create, edit and delete your own playlist, we can also shuffle the songs or play the songs on loop. Moreover , we can see our search history just by going to the recently played option available. You will be using various data structures such as circular and doubly linked list, stack, queues, arrays, trees.

You will provide the following functionalities:

1. Creating a playlist:

   We need to add the playlist name to be created, After entering the playlist name, now we can add songs in our own playlist by just typing ‘a’. A list of all the songs available will be shown and you can add songs in the playlist.

2. Adding songs in playlist:

A list of latest hits should be displayed to the user, user should be able to choose the ID of the song to be added to his/her playlist and then a message should be displayed saying “Song successfully added”

3. Deleting songs from the playlist:

If somehow the playlist is empty or you have deleted all the songs from the playlist, it will display the message that the playlist is empty.

If the song to be deleted isn’t found, then it displays the message- ‘Song doesn’t exist”.

Finally, if the song is found and can be deleted, a message should be displayed saying : “Song deleted successfully ”

4. Display entire playlist:

To display the playlists that the user created including:

1. Playlist Name
2. Playlist Songs

5. Display Total number of songs in the playlist

Counts the number of total songs in the playlist and prints it

6. Shuffle:

For shuffling, there are three different types to shuffle playlist:

1. Pre Order
2. Post Order
3. In Order

7. Search any song on the app(Ascending):

1. By song name
2. By Artist name

8. Track your search history:

The search history should display the recent history of search from recent one to oldest

Data Structures

9. Last played song

   The last song that was played by the user

10. Sort

    1. Sorting based on number of plays
       a. Keeping track of the number of times every song is played, and sorting them accordingly, from most played to least played
       b. Reverse sort, To reverse the previous sorting from the least played to the most played songs
    2. Recently played Songs

       From the most recent to less recently played songs.

The main should contain all the above mentioned requirements as an option that the user chooses from after he/she logs into the system.

Bonus Part:

Doing the project with a file named musics.txt that contains a list of songs.

Below is a list of functions you might need to do:

1. An empty file playlist.txt that will be for the user.
2. tofile() – Function to work on playslist.txt.
3. add_node_file() – Function that adds songs to the playlist to linked list from the data passed in addplaylist() function.
4. delete_file() – Function to delete song from text file playlist.txt.
5. printlist() – Function that displays the input songs of the playlist.
6. addplaylist() – Function that opens text file playlist.txt and passes data to add_node_file() function.
7. play() - Function to search input song and show if it can be played. It then passes the song to push() function to be added to recently played list.