Project PlaylistApp

Team member: Lan Luo, Yang Ming, Ziang Xu

Summary:
Create a popular playlist application that accepts song playlists along with their popularities, stores the most popular 1024 playlists, and i) suggests playlists given a song and ii) suggests songs when the user starts typing a song name.

App form: website
Reason: Simple, appropriate to accomplish the functions

Using language: Python
Reason: Simple, all members have background of python

Accomplishment:
Until recent days, we have made a simple website and designed algorithms to sort the songs and playlists and search them.

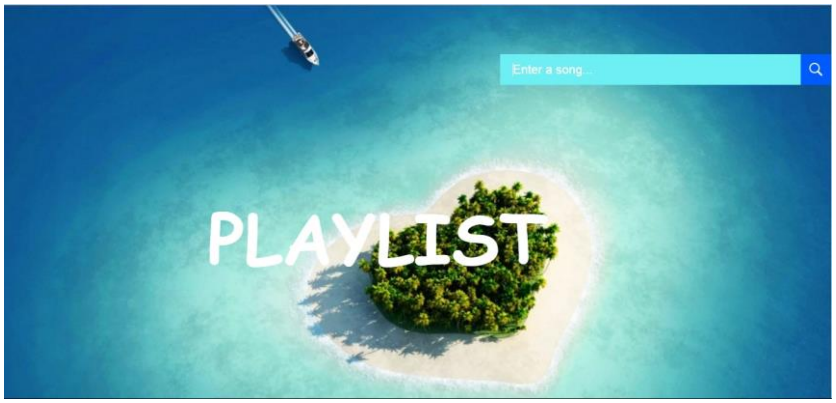
1. Algorithm:

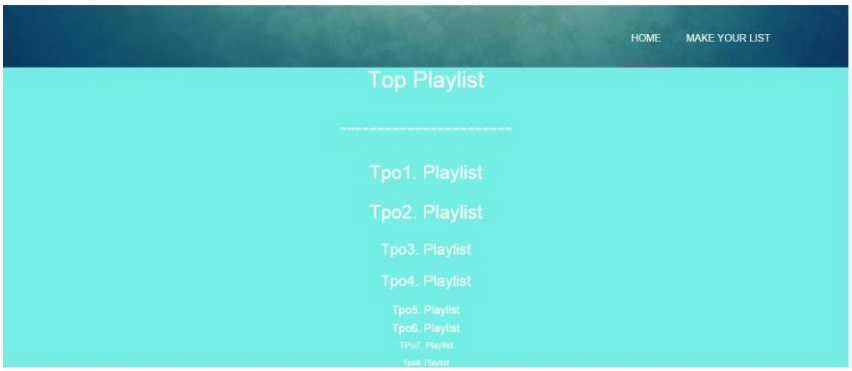
(1) Restore:
We decided to use dictionary to accomplish restoring the data.

(2) Sort: using quicksort
Quicksort has a higher efficiency in most cases and it causes less data movement.

(3) Search: using trie
Advantages of trie:
The lookup time is predictable which is O(k) where k is the size of keys;
Lookup can take less than k time if the song we look up is not in the dictionary;
Deletion is straightforward in trie;
We can quickly look up prefixes of keys, enumerate all entries with a given prefix. This is very suitable for accomplish the searching function of the project.

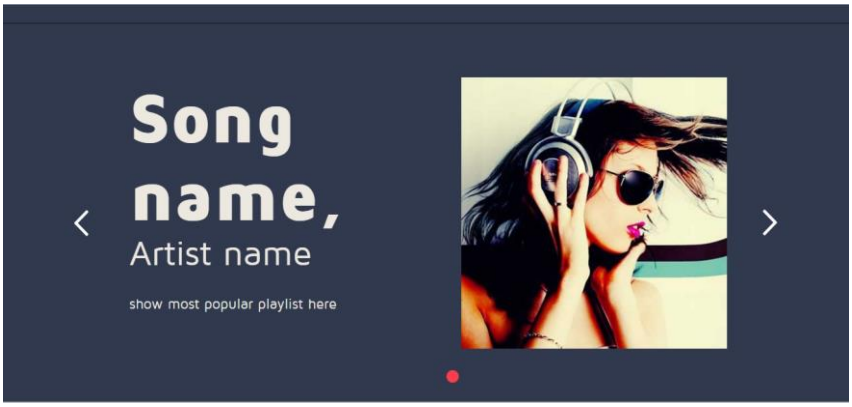
2. Website:
Webpage 1:





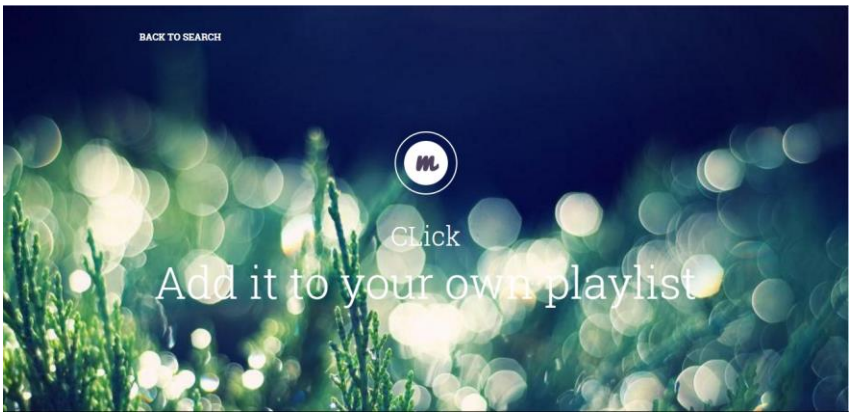
Webpage 1 gives the function of searching a song on the top and shows top 8 most popular playlists on the bottom.

Webpage 2:



Webpage 2 shows a single playlist along with its popularity.

Webpage 3:



Webpage 3 gives the function of adding a new playlist on the top and shows the most popular playlist the song searched in webpage one is in.

3. Database:

We have set up the environment of the database, we are going to add the data in it in the few days.

Plan:

1. Finish the program
2. Accomplish design and layout of the website
3. Transmit the data into the database and upload it into AWS
4. Connect the website with the program