

# Music Collaboratory Web Application

---



As I get better, you get better! As I learn, you learn!

---

## Table of Contents

These section headers will be used to reference location of destination for an easy use.

- [Music Collaboratory Web Application](#)
  - [Table of Contents](#)
  - [Problem Description](#)
    - [Languages/Technologies](#)
  - [How To Use](#)
    - [API Reference](#)
    - [Site Demo](#)
  - [References](#)
  - [License](#)
  - [Author Info](#)

---

## Problem Description

Suppose you are travelling with friends or families and for the purpose of entertainment you are playing music. What other platform is better than Spotify (i.e. if you have premium account, P.S: you don't want to listen to the ads). But it's becoming very difficult as everyone has their own taste and choices and depending on the current mood one might not feel like listening to a particular song, so, what's the easy way out? One can take the majority of the votes, calculate it manually and change or skip the song but this will be very hectic and it will ruin the experience of the trip itself, of which one would not want to have an experience of.

In such a situation, you wouldn't want to be the person who had called shotgun previously. For such a scenario, you need to create a system that is hassle-free and makes the job easier by building a collaborative system where votes can be taken to play or skip a song.

Expectations from this system will be to get votes for songs to skip or keep playing, and in such a way that it uncovers all the perspective of managing the system itself. It will be sighted to show:

- Control the music in some sense of unity?
- Security features? So, that only a person with the code can join the group?
- Different permissions will be given to all the members and whether the host can decide the roles of different members involved?

## Languages/Technologies

Frontend:

- HTML5
- Javascript
- React.js
- Configuration-Files
- CSS3
- SVG

Backend:

- Django(Python)

Database:

- Django Database Adminsator (This section will be updated with time.)

[Back To The Top](#)

---

## How To Use

- Each technology has it's own directory.
- Navigate to the same and you will find the code related to the same.

## API Reference

```
const learn_build_teach = "AWESOME!!!";
```

## Site Demo

[NYC-Times Critics Pick](#)

[Back To The Top](#)

---

## References

- [YouTube](#)
- [Coursera](#)

[Back To The Top](#)

---

## License

### MIT License

Copyright (c) [2020] [eseus-coder] Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "nyctimes-critics-pick"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

[Back To The Top](#)

---

## Author Info

- GitHub - [eseus-coder](#)

[Back To The Top](#)