Music Player Help document.

Hardware requirements

In order to run the program you are required to have a monitor, keyboard, mouse and a basic computer with a sound card ,CPU,RAM, motherboard, power supply and a hard drive/SSD with preferably windows 10 operating system.

Setting up the node project

There are a few installations that need to be used in order to start the music.

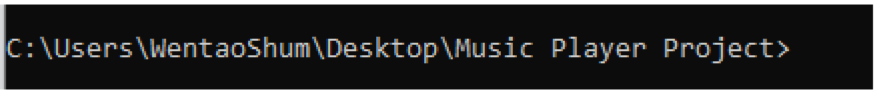
Firstly you will need a browser (eg. Google Chrome, Firefox and Microsoft Edge)

In order to set up the project you will be required to install node.js which can be found here.

https://nodejs.org/en/download/

If you open command prompt (open as admin)

And open up your following directory(where you stored the music player) of where your music player is (use cd to open up a directory).



```
C:\Users\WentaoShum\Desktop\Music Player Project>
```

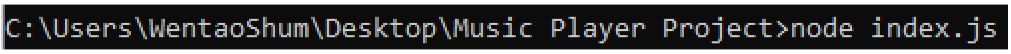
Type in the following commands

```
npm install express --save npm
npm install body-parser --save npm
npm install cookies-parser -- save npm
npm install multer --save
```

npm install ejs
npm install pg
npm install fs

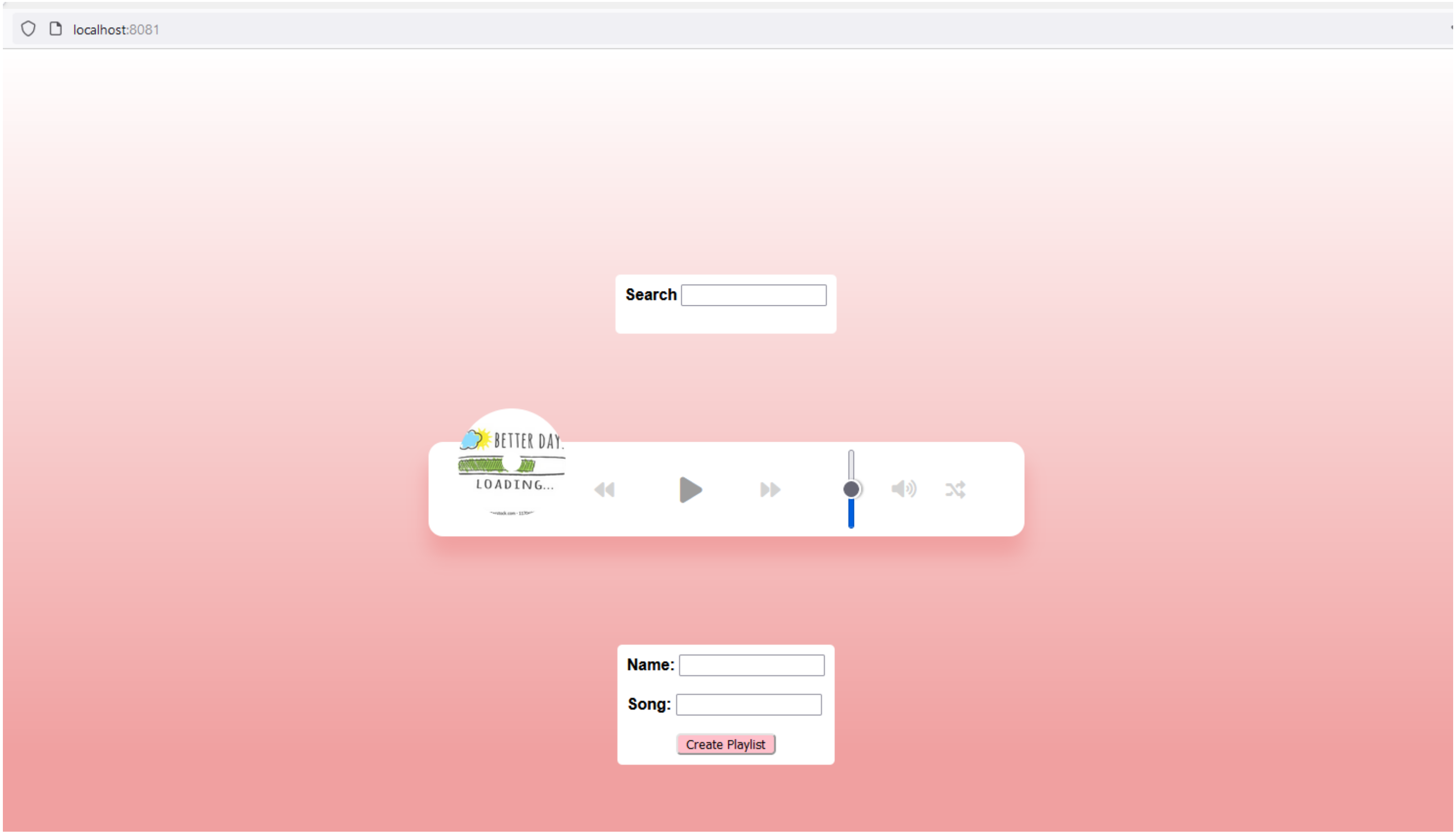
These are the Node packages that you will need in order to run the music project.

And in order to start the music player local server type in cmd the following command as shown below node index.js



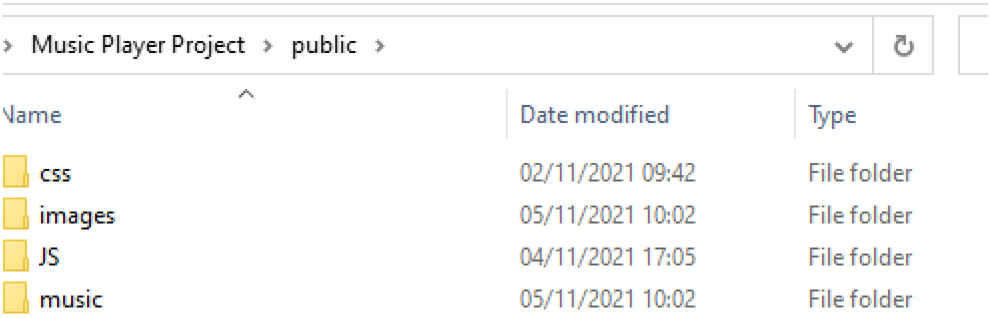
```
C:\Users\WentaoShum\Desktop\Music Player Project>node index.js
```

This will allow you to then navigate to http://localhost:8081/ within your browser (Note: make sure that port 8081 is cleared)
And here is the music player.



Adding music/images

In order to provide music/images to the music player all you need to go to is the Music Player Project -> public ->image/music. And add in your files here in order for the music to play and display an image as the spinning vinyl (if you want a song to have a certain vinyl they must be both called the same e.g dubstep.mp3 will have the spinning vinyl of dubstep.jpg.



› Music Player Project › public ›

Name	Date modified	Type
css	02/11/2021 09:42	File folder
images	05/11/2021 10:02	File folder
JS	04/11/2021 17:05	File folder
music	05/11/2021 10:02	File folder

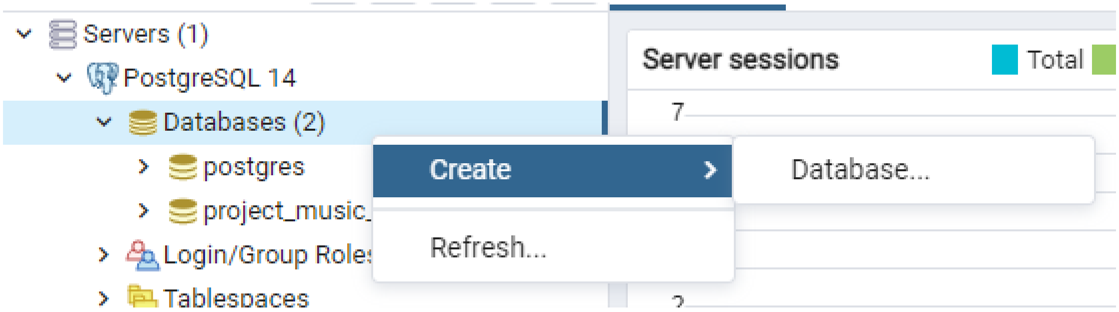
Note: that music will only accept .mp3 files and images will only accept .jpg files, if any other file types are included these may not work as intended.

Databases

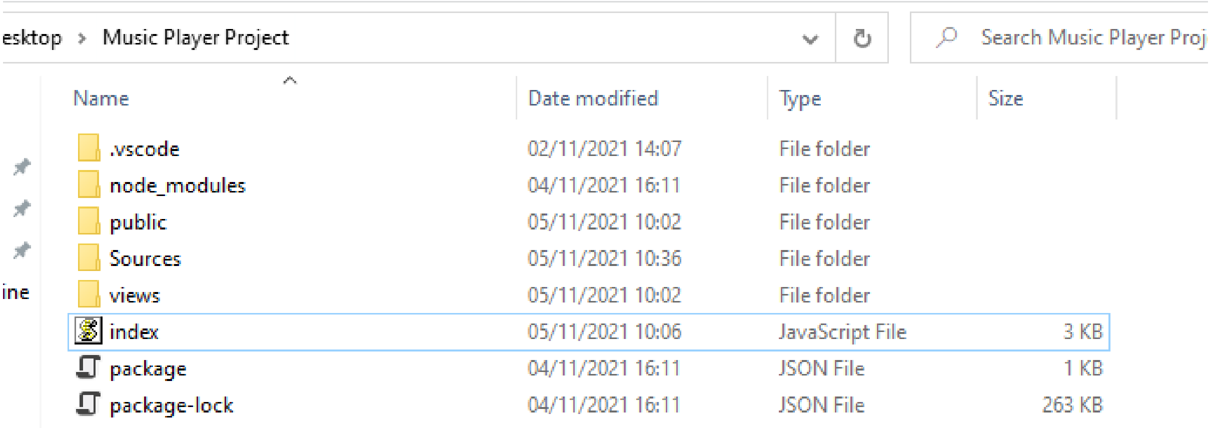
In order to view databases you will need to install Postgres
https://www.postgresql.org/download/
And pgAdmin
https://www.pgadmin.org/download/

Next you would like to create your database after setting up pgAdmin:

Within pgAdmin navigate to PostgreSQL and right click databases and "Create" "Database…" and give the database a name.



Once complete in order to setup your database to the music player, navigate to index.js within the music player

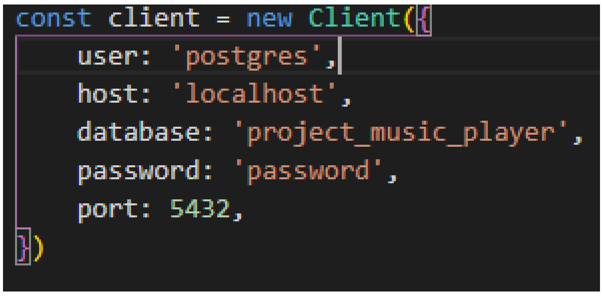


Name	Date modified	Type	Size
.vscode	02/11/2021 14:07	File folder	
node_modules	04/11/2021 16:11	File folder	
public	05/11/2021 10:02	File folder	
Sources	05/11/2021 10:36	File folder	
views	05/11/2021 10:02	File folder	
index	05/11/2021 10:06	JavaScript File	3 KB
package	04/11/2021 16:11	JSON File	1 KB
package-lock	04/11/2021 16:11	JSON File	263 KB

And using your IDE (e.g notepad++, visual studio code)

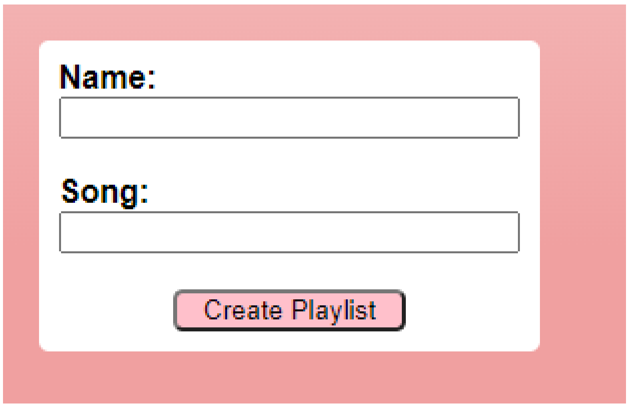
Change the following to match your postgres settings

(User = will most likely be postgres if left as default during setup)
(host = most likely localhost)
(database = *INSERT your database name here*)
(port = 5432 if left as default during setup)



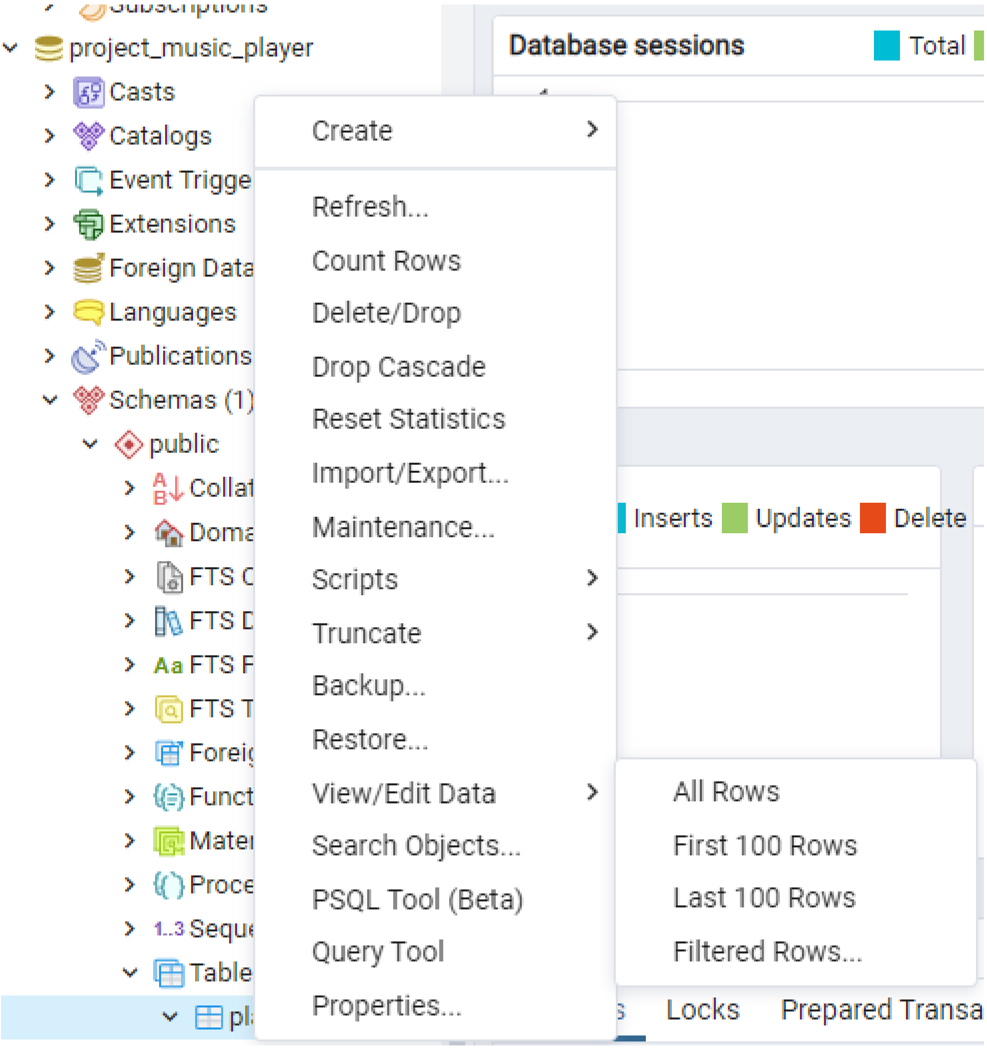
```
const client = new Client({
    user: 'postgres',
    host: 'localhost',
    database: 'project_music_player',
    password: 'password',
    port: 5432,
})
```

In order to see and populate your database startup the music player (shown previously node index.js)
This will automatically create the table of “playlists” for you and in order to populate tables input data within the playlist form that looks like below.

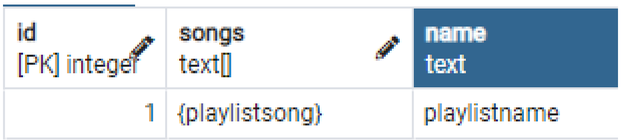


(Note: The playlist form does not have a high level of validation, duplicate information can be inputted and there is no minimum and maximum length working. (this needs to be worked on))

To see the results of the database go to pgAdmin and look at the *your database name*, right click the playlist table and then “view/edit data”, “all rows”



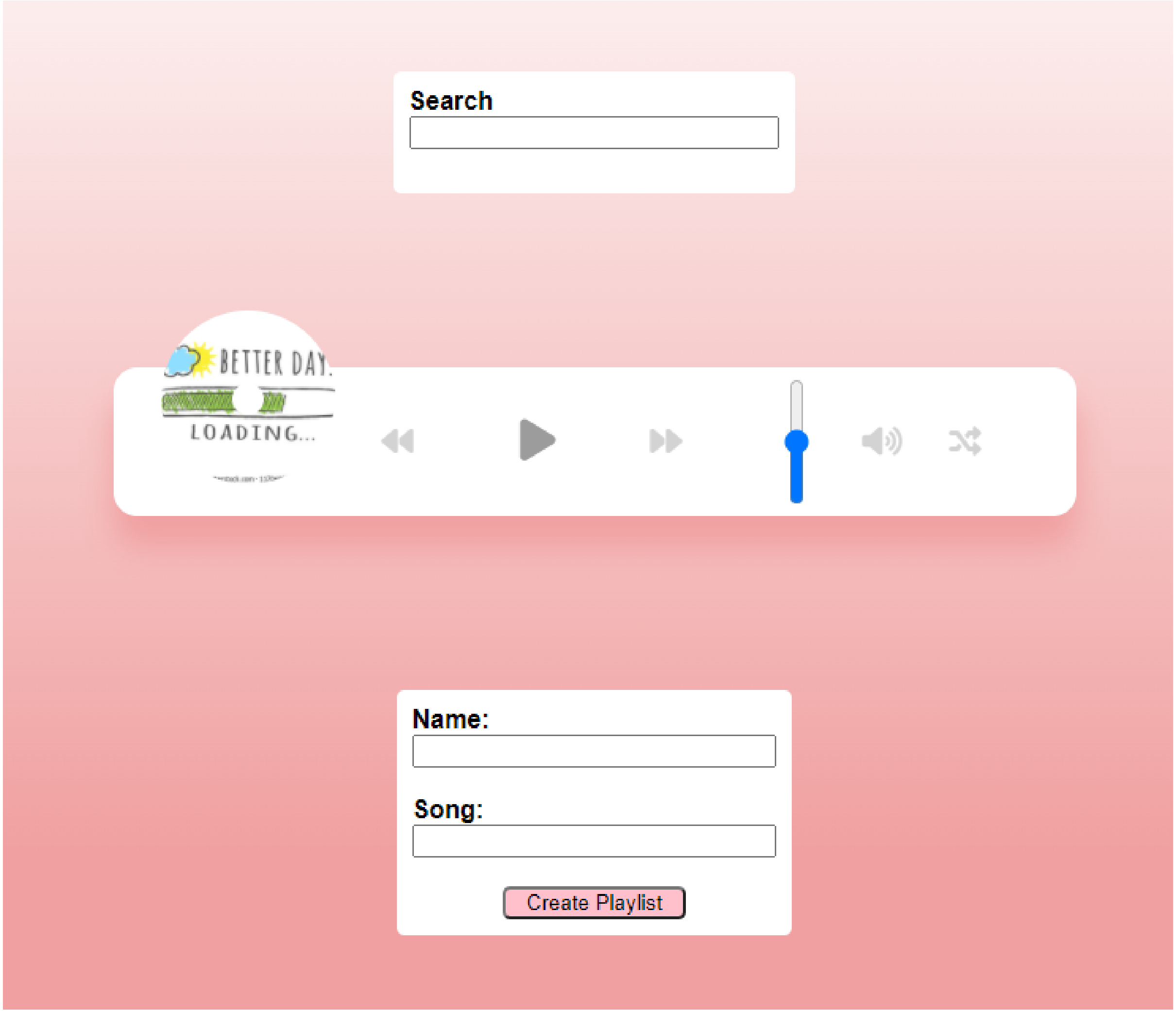
Within Data Output you can see your inputted information here.



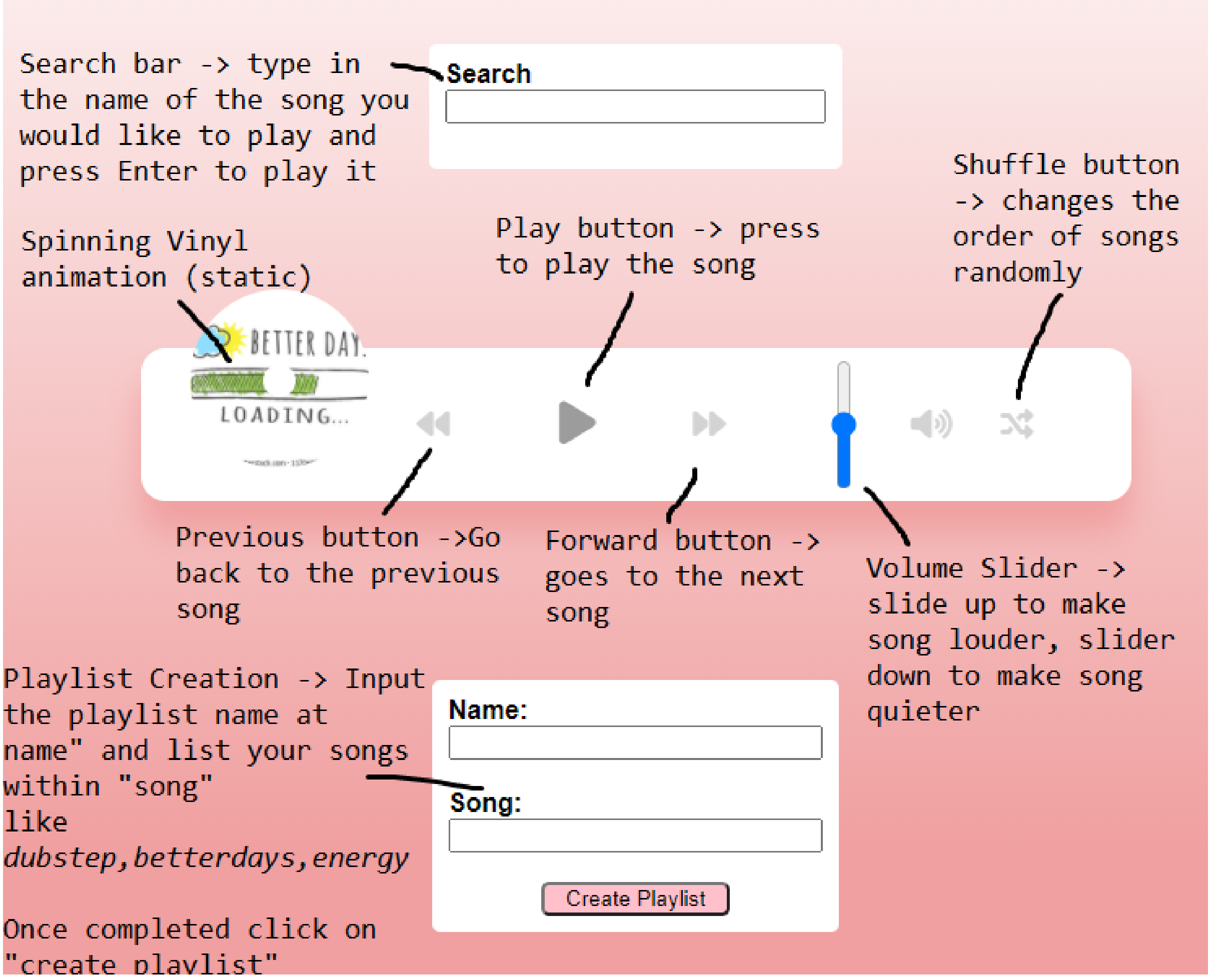
id [PK] integer	songs text[]	name text
1	{playlistsong}	playlistname

How to use Music player

This is how the music player is displayed on first load (it will always display the first song in alphabetical order)



I have created an annotated diagram of the first screen in order to describe what each part of the page does and how to use it.



Search bar -> type in the name of the song you would like to play and press Enter to play it

Spinning Vinyl animation (static)

Play button -> press to play the song

Shuffle button -> changes the order of songs randomly

Previous button ->Go back to the previous song

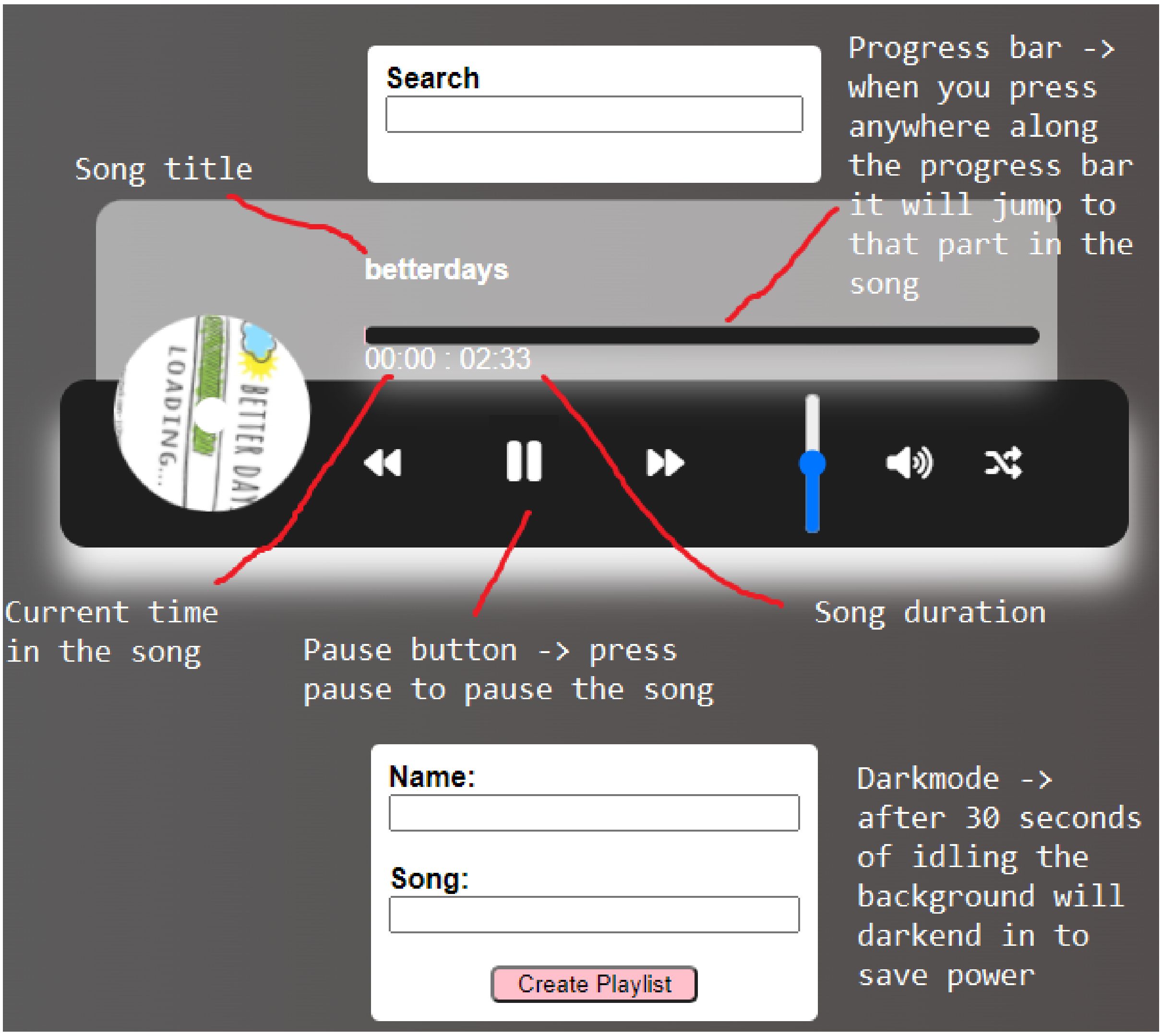
Forward button -> goes to the next song

Volume Slider -> slide up to make song louder, slider down to make song quieter

Playlist Creation -> Input the playlist name at name" and list your songs within "song" like *dubstep,betterdays,energy*

Once completed click on "create playlist"

And here is an annotated diagram of when you press play, I’ve also included what darkmode looks like when you’ve been idling for over 30 seconds.



Song title

Progress bar -> when you press anywhere along the progress bar it will jump to that part in the song

Current time in the song

Pause button -> press pause to pause the song

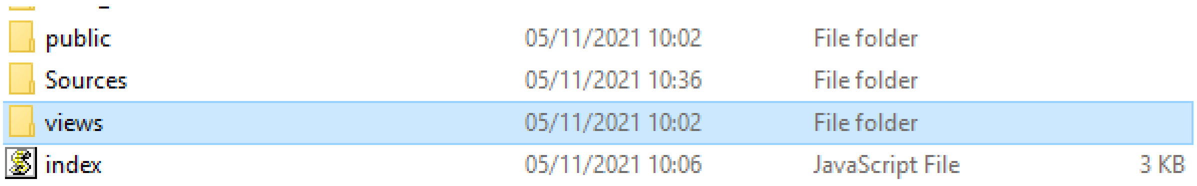
Song duration

Darkmode -> after 30 seconds of idling the background will darkend in to save power

Development structure

This is a detailed guide for developers to understand how to develop on this music player further.

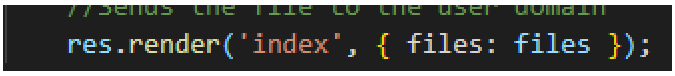
Firstly is the file structure index.js should be left out in root, views contains index.ejs (which allows for html markup in plain javascript). And public contains all the Javascript, CSS, images and music.



public	05/11/2021 10:02	File folder	
Sources	05/11/2021 10:36	File folder	
views	05/11/2021 10:02	File folder	
index	05/11/2021 10:06	JavaScript File	3 KB

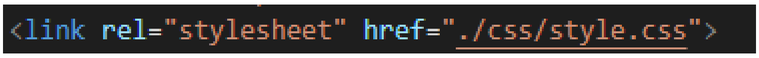
Index.js is in the root folder as it creates the connection to the local server, this also contains database connections. If any extra node packages need to be installed for future development, define them here in index.js

views/index.ejs is where HTML is written any future elements that are required to be added are done so here, also any variables that need to be passed from index.js to any other JS functions need to be defined within view/index.ejs using <*%= variable from index.js %*> in a <script> tag (don't forget as well to define the variable you'd like to pass through from index.js with

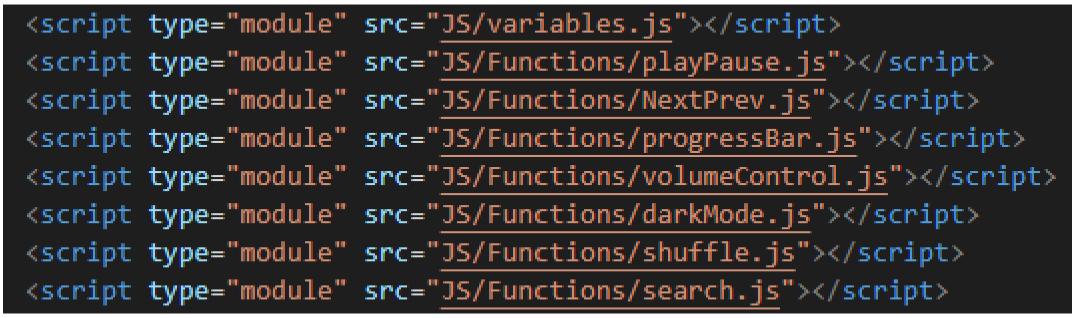


```
    //sends the file to the user domain
    res.render('index', { files: files });
```

. Also within index.ejs any further JS or CSS style sheet to be added is also done here. Shown Below.

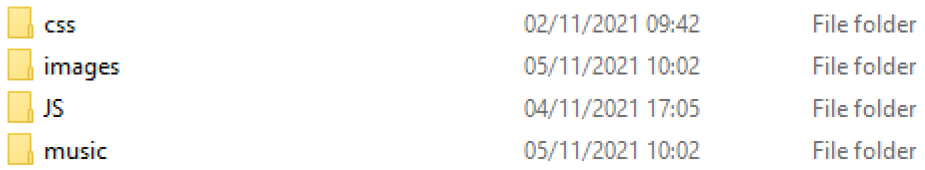


```
<link rel="stylesheet" href="./css/style.css">
```



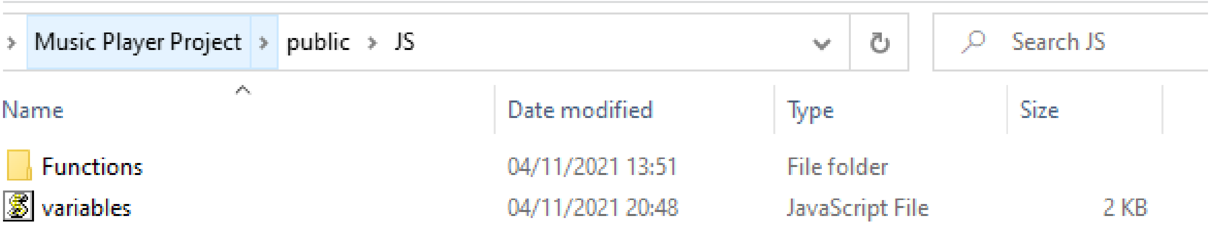
```
  <script type="module" src="JS/variables.js"></script>
  <script type="module" src="JS/Functions/playPause.js"></script>
  <script type="module" src="JS/Functions/NextPrev.js"></script>
  <script type="module" src="JS/Functions/progressBar.js"></script>
  <script type="module" src="JS/Functions/volumeControl.js"></script>
  <script type="module" src="JS/Functions/darkMode.js"></script>
  <script type="module" src="JS/Functions/shuffle.js"></script>
  <script type="module" src="JS/Functions/search.js"></script>
```

Within the public directory is the previously mentioned CSS, JS, images and music folder. Images and Music are where .mp3 and .jpg files go respectively. CSS contains style.css, where any styling is created, any future style sheet should be placed here.



css	02/11/2021 09:42	File folder
images	05/11/2021 10:02	File folder
JS	04/11/2021 17:05	File folder
music	05/11/2021 10:02	File folder

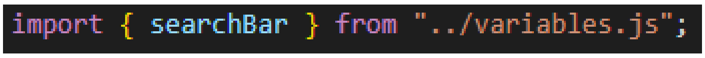
Within JS folder is anything Javascript relating to client side goes here.



Name	Date modified	Type	Size
Functions	04/11/2021 13:51	File folder	
variables	04/11/2021 20:48	JavaScript File	2 KB

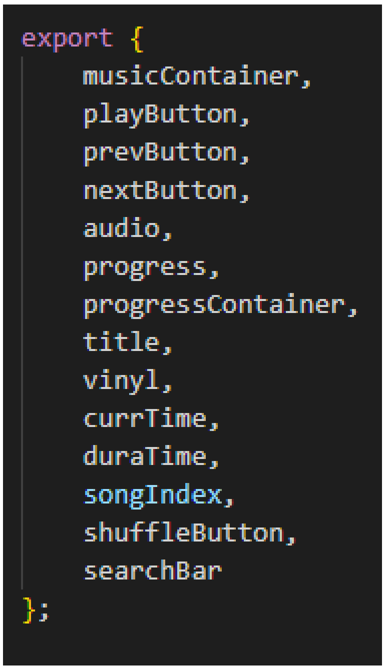
Within variables.js is where all variables will be defined, all future variables should be defined here and imported and exported as shown within the file.

Example of an import:



```
import { searchBar } from "../variables.js";
```

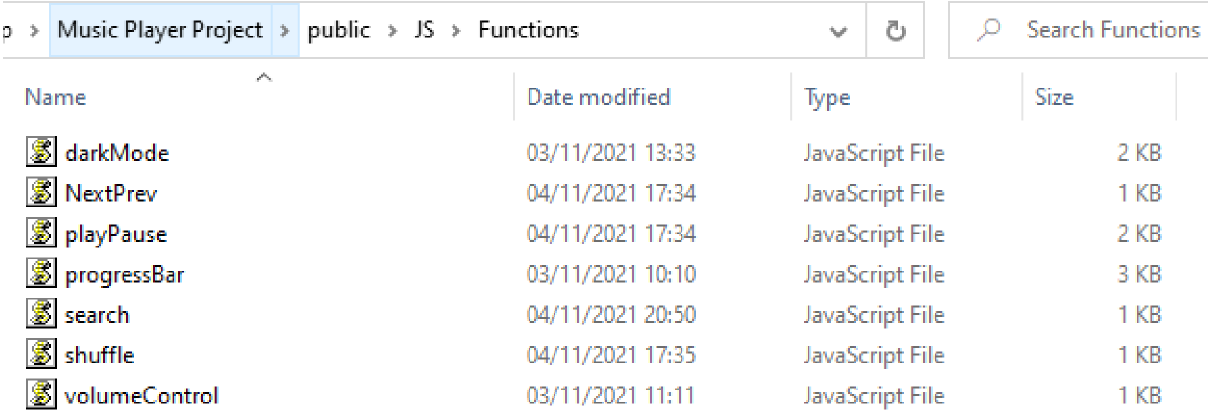
Example of how to export:



```
export {
    musicContainer,
    playButton,
    prevButton,
    nextButton,
    audio,
    progress,
    progressContainer,
    title,
    vinyl,
    currTime,
    duraTime,
    songIndex,
    shuffleButton,
    searchBar
};
```

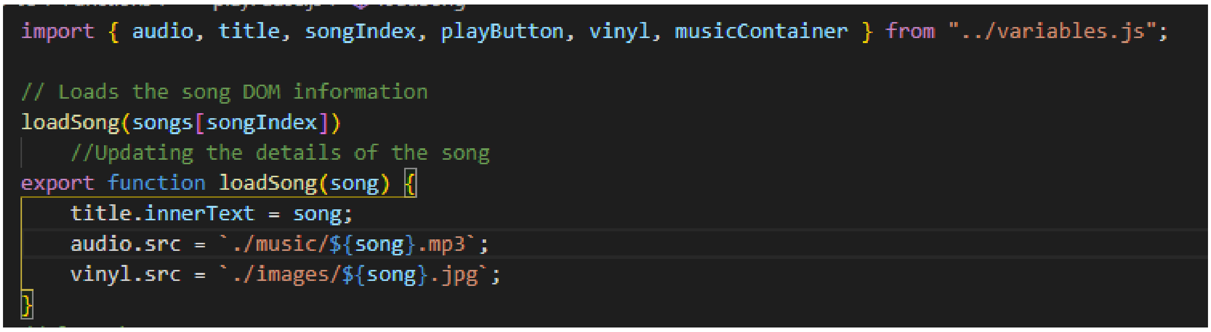
Within Functions its functionality should be divided into separate files (this is to optimize debugging).

Start off each file by importing any variables which are required and then write the function.



Name	Date modified	Type	Size
darkMode	03/11/2021 13:33	JavaScript File	2 KB
NextPrev	04/11/2021 17:34	JavaScript File	1 KB
playPause	04/11/2021 17:34	JavaScript File	2 KB
progressBar	03/11/2021 10:10	JavaScript File	3 KB
search	04/11/2021 20:50	JavaScript File	1 KB
shuffle	04/11/2021 17:35	JavaScript File	1 KB
volumeControl	03/11/2021 11:11	JavaScript File	1 KB

This example is search.js and javascript should be structured like this.
Any functions that need to be exported can be exported with export in front of the function.



```
import { audio, title, songIndex, playButton, vinyl, musicContainer } from "../variables.js";

// Loads the song DOM information
loadSong(songs[songIndex])
    //Updating the details of the song
export function loadSong(song) {
    title.innerText = song;
    audio.src = `./music/${song}.mp3`;
    vinyl.src = `./images/${song}.jpg`;
}
```

Known issues

Version 1.0

- Database validations have issues of allowing for duplication of data and there is no set minimum and maximum length.
- Search function can allow for invalid song names, this can be fixed by passing in a check to see if the song name exists before trying to play the song.