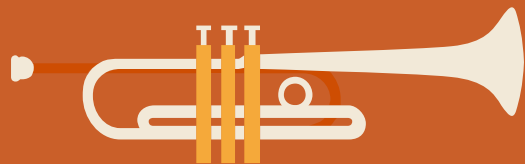
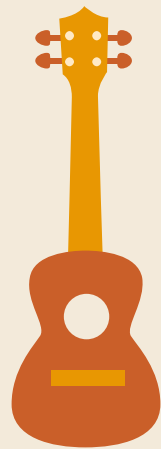


Project: *Music Player* *Presentation*



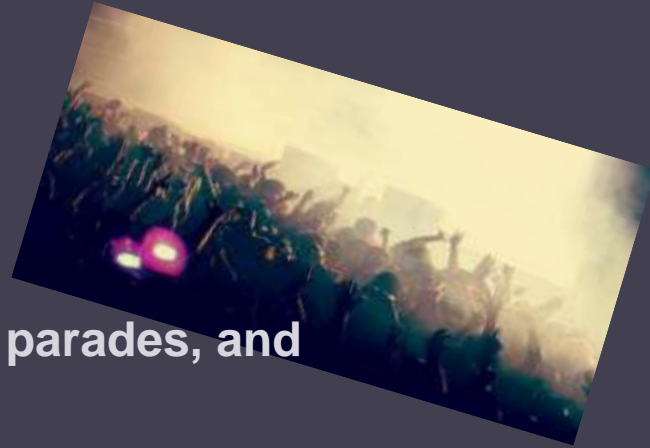
Binding People Together

- o **Concerts, clubs, parties, etc.**

allow people to bond and experience music together

- o Music plays an important role in special occasions such as **weddings, parades, and religious festivities**

- o "...means of giving voice to their common experience and to give vent to anger, grief and frustration at their mistreatment."



Evoke Strong Emotions/Feelings:



- Motivation
- Sadness
- Euphoria



About Project:

This project [Music Player in JavaScript] has several features like you can loop, repeat or shuffle a song, play/pause a song or play the next or previous song. You can view your songs list and also know which song is currently playing and you can also select the song from the list to play.



About The Code~

To create this project [Custom Music Player in JavaScript]. First, we created four files, HTML File, CSS File, and JavaScript File.

First, we created an HTML file with the name of index.html and paste the given codes in your HTML file. Remember, you've to create a file with .html extension and the images that are used on this music player won't appear.

Code Continues...

Second, we created a CSS file with the name of style.css and paste the given codes in your CSS file.

Then Third, a JavaScript file with the name of music-list.js and paste the given codes in your JavaScript file. In this JavaScript file, we store all music details using the array.

~And Last

Last, create a JavaScript file with the name of script.js and paste the given codes in your JavaScript file.

Input:html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content=
      "width=device-width, initial-scal
      e=1.0"
6    <title>
      Music Player | CodingNepal</title>
7    <link rel="stylesheet" href=
      "style.css">
8    <link rel="stylesheet" href=
      "https://fonts.googleapis.com/ico
      n?family=Material+Icons"
9  </head>
10 <body>
11   <div class="wrapper">
12     <div class="top-bar">
13       <i class="material-icons">
          expand_more</i>
14       <span>Now Playing</span>
15       <i class="material-icons">
          more_horiz</i>
16     </div>
17     <div class="img-area">
18       <img src="" alt="">
19     </div>
20     <div class="song-details">
21       <p class="name"></p>
22       <p class="artist"></p>
23     </div>
24     <div class="progress-area">
25       <div class="progress-bar">
26         <audio id="main-audio"
          src=""></audio>
27       </div>
28       <div class="song-timer">
29         <span class=
          "current-time">0:00</span>
30         <span class=
          "max-duration">0:00</span>
```

```
31     </div>
32   </div>
33   <div class="controls">
34     <i id="repeat-plist" class=
      "material-icons" title=
      "Playlist looped">repeat</i>
35     <i id="prev" class=
      "material-icons">skip_previous</i>
36     <div class="play-pause">
37       <i class=
      "material-icons play">play_arrow
38     </div>
39     <i id="next" class=
      "material-icons">skip_next</i>
40     <i id="more-music" class=
      "material-icons">queue_music</i>
41   </div>
42   <div class="music-list">
43     <div class="header">
44       <div class="row">
45         <i class=
          "list material-icons">queue_music
46       </div>
47       <span>Music list</span>
48       <i id="close" class=
          "material-icons">close</i>
49     </div>
50     <ul>
51       <!-- here li list are coming fro
          m js -->
52     </ul>
53   </div>
54 </div>
55 <script src="js/music-list.js"
56 ></script>
57 <script src="js/script.js"></
58 script>
59 </body>
60 </html>
```


Input:css

```
1  @import url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700&display=swap');
2  *{
3    margin: 0;
4    padding: 0;
5    box-sizing: border-box;
6    font-family: "Poppins",
    sans-serif;
7  }
8  *::before, *::after{
9    padding: 0;
10   margin: 0;
11 }
12 :root{
13   --pink: #000000;
14   --violet: #9f6ea3;
15   --lightblack: #515C6F;
16   --white: #ffffff;
17   --darkwhite: #cecaca;
18   --pinkshadow: #ffcbbd;
19   --lightbshadow: rgba(0,0,0,
20   0.15);
21 }
22 body{
23   display: flex;
24   align-items: center;
25   justify-content: center;
26   min-height: 100vh;
27   background: linear-gradient(
28     var(--pink) 0%, var(--violet)
29     100%);
30 }
31 .wrapper{
32   width: 380px;
33   padding: 25px 30px;
34   overflow: hidden;
35   position: relative;
36   border-radius: 15px;
37   background: var(--white);
38   box-shadow: 0px 6px 15px var(
39     --lightbshadow);
40 }
41 .wrapper i{
42   cursor: pointer;
43 }
44 .top-bar, .progress-area
45 .song-timer,
```

```
41   .controls, .music-list .header,
42   .music-list ul li{
43     display: flex;
44     align-items: center;
45     justify-content: space-between
46   ;
47 }
48 .top-bar i{
49   font-size: 30px;
50   color: var(--lightblack);
51 }
52 .top-bar i:first-child{
53   margin-left: -7px;
54 }
55 .top-bar span{
56   font-size: 18px;
57   margin-left: -3px;
58   color: var(--lightblack);
59 }
60 .img-area{
61   width: 100%;
62   height: 256px;
63   overflow: hidden;
64   margin-top: 25px;
65   border-radius: 15px;
66   box-shadow: 0px 6px 12px var(
67     --lightbshadow);
68 }
69 .img-area img{
70   width: 100%;
71   height: 100%;
72   object-fit: cover;
73 }
74 .song-details{
75   text-align: center;
76   margin: 30px 0;
77 }
78 .song-details p{
79   color: var(--lightblack);
80 }
81 .song-details .name{
82   font-size: 21px;
83 }
```

```
81 .song-details .artist{
82   font-size: 18px;
83   opacity: 0.9;
84   line-height: 35px;
85 }
86 .progress-area{
87   height: 6px;
88   width: 100%;
89   border-radius: 50px;
90   background: #f0f0f0;
91   cursor: pointer;
92 }
93 .progress-area .progress-bar{
94   height: inherit;
95   width: 0%;
96   position: relative;
97   border-radius: inherit;
98   background: linear-gradient(
99     90deg, var(--pink) 0%, var(
100    --violet) 100%);
101 }
102 .progress-bar::before{
103   content: "";
104   position: absolute;
105   height: 12px;
106   width: 12px;
107   border-radius: 50%;
108   top: 50%;
109   right: -5px;
110   z-index: 2;
111   opacity: 0;
112   pointer-events: none;
113   transform: translateY(-50%);
114   background: inherit;
115   transition: opacity 0.2s ease;
116 }
117 .progress-area: hover
118 .progress-bar::before{
119   opacity: 1;
120   pointer-events: auto;
121 }
122 .progress-area .song-timer{
123   margin-top: 2px;
124 }
```

Input:css

```
121 }
122 .song-timer span{
123   font-size: 13px;
124   color: var(--lightblack);
125 }
126 .controls{
127   margin: 40px 0 5px 0;
128 }
129 .controls i{
130   font-size: 28px;
131   user-select: none;
132   background: linear-gradient(
133     var(--pink) 0%, var(--violet)
134     100%);
135   background-clip: text;
136   -webkit-background-clip: text;
137   -webkit-text-fill-color:
138     transparent;
139 }
140 .controls i:nth-child(2),
141 .controls i:nth-child(4){
142   font-size: 43px;
143 }
144 .controls #prev{
145   margin-right: -13px;
146 }
147 .controls #next{
148   margin-left: -13px;
149 }
150 .controls .play-pause{
151   height: 54px;
152   width: 54px;
153   display: flex;
154   cursor: pointer;
155   align-items: center;
156   justify-content: center;
157   border-radius: 50%;
158   background: linear-gradient(
159     var(--white) 0%, var(--darkwhite)
160     100%);
161   box-shadow: 0px 0px 5px var(
162     --pink);
163 }
164 .play-pause::before{
165   position: absolute;
166   content: "";
```

```
161   height: 43px;
162   width: 43px;
163   border-radius: inherit;
164   background: linear-gradient(
165     var(--pink) 0%, var(--violet)
166     100%);
167 }
168 .play-pause i{
169   height: 43px;
170   width: 43px;
171   line-height: 43px;
172   text-align: center;
173   background: inherit;
174   background-clip: text;
175   -webkit-background-clip: text;
176   -webkit-text-fill-color:
177     transparent;
178   position: absolute;
179 }
180 .music-list{
181   position: absolute;
182   background: var(--white);
183   width: 100%;
184   left: 0;
185   bottom: -55%;
186   opacity: 0;
187   pointer-events: none;
188   z-index: 5;
189   padding: 15px 30px;
190   border-radius: 15px;
191   box-shadow: 0px -5px 10px rgba
192     (0,0,0,0.1);
193   transition: all 0.15s ease-out
194   ;
195 }
196 .music-list.show{
197   bottom: 0;
198   opacity: 1;
199   pointer-events: auto;
200 }
```

```
201   color: var(--lightblack);
202 }
203 .header .row i{
204   cursor: default;
205 }
206 .header .row span{
207   margin-left: 5px;
208 }
209 .header #close{
210   font-size: 22px;
211   color: var(--lightblack);
212 }
213 .music-list ul{
214   margin: 10px 0;
215   max-height: 260px;
216   overflow: auto;
217 }
218 .music-list
219 ul::-webkit-scrollbar{
220   width: 0px;
221 }
222 .music-list ul li{
223   list-style: none;
224   display: flex;
225   cursor: pointer;
226   padding-bottom: 10px;
227   margin-bottom: 5px;
228   color: var(--lightblack);
229   border-bottom: 1px solid
230     #E5E5E5;
231 }
232 .music-list ul li:last-child{
233   border-bottom: 0px;
234 }
235 .music-list ul li .row span{
236   font-size: 17px;
237 }
238 .music-list ul li .row p{
239   opacity: 0.9;
240 }
241 .audio-duration{
242   font-size: 16px;
243 }
244 ul li.playing{
245   pointer-events: none;
246   color: var(--violet);
247 }
```

Input:Music List

```
1  let allMusic = [  
2    {  
3      name: "Harley Bird - Home",  
4      artist: "Jordan Schor",  
5      img: "music-1",  
6      src: "music-1"  
7    },  
8    {  
9      name:  
10     "Ikson Anywhere - Ikson",  
11     artist: "Audio Library",  
12     img: "music-2",  
13     src: "music-2"  
14   },  
15   {  
16     name: "Beauz & Jvna - Crazy",  
17     artist: "Beauz & Jvna",  
18     img: "music-3",  
19     src: "music-3"  
20   },  
21   {  
22     name: "Hardwind - Want Me",  
23     artist: "Mike Archangelo",  
24     img: "music-4",  
25     src: "music-4"  
26   },  
27   {  
28     name: "Jim - Sun Goes Down",  
29     artist: "Jim Yosef x Roy",  
30     img: "music-5",  
31     src: "music-5"  
32   },  
33   {  
34     name: "Lost Sky - Vision NCS"  
35     ,  
36     artist: "NCS Release",  
37     img: "music-6",  
38     src: "music-6"  
39   },  
40 ];
```

Input:JavaScript

```
1 const wrapper = document.  
2   querySelector(".wrapper"),  
3   musicImg = wrapper.querySelector  
4   ("img-area img"),  
5   musicName = wrapper.  
6   querySelector(".song-details .name"),  
7   musicArtist = wrapper.  
8   querySelector(".song-details .artist"),  
9   playPauseBtn = wrapper.  
10  querySelector(".play-pause"),  
11  prevBtn = wrapper.querySelector(  
12    "#prev"),  
13  nextBtn = wrapper.querySelector(  
14    "#next"),  
15  mainAudio = wrapper.  
16  querySelector("#main-audio"),  
17  progressArea = wrapper.  
18  querySelector(".progress-area"),  
19  progressBar = progressArea.  
20  querySelector(".progress-bar"),  
21  musicList = wrapper.  
22  querySelector(".music-list"),  
23  moreMusicBtn = wrapper.  
24  querySelector("#more-music"),  
25  closeMoreMusic = musicList.  
26  querySelector("#close");  
27  
28 let musicIndex = Math.floor(  
29   Math.random() * allMusic.length  
30 ) + 1;  
31 isMusicPaused = true;  
32  
33 window.addEventListener("load",  
34   ()=>{  
35     loadMusic(musicIndex);  
36     playingSong();  
37   });  
38  
39 function loadMusic(indexNum){  
40   musicName.innerText = allMusic  
41   [indexNum - 1].name;  
42   musicArtist.innerText =  
43   allMusic[indexNum - 1].artist;  
44   musicImg.src = `images/${  
45   allMusic[indexNum - 1].src  
46   }.jpg`;  
47   mainAudio.src = `songs/${  
48   allMusic[indexNum - 1].src  
49   }.mp3`;  
50  
51 //play music function  
52 function playMusic(){  
53   wrapper.classList.add("paused")  
54 };  
55 playPauseBtn.querySelector("i")  
56 .innerText = "pause";  
57 mainAudio.play();  
58  
59 //pause music function  
60 function pauseMusic(){  
61   wrapper.classList.remove(  
62     "paused");  
63   playPauseBtn.querySelector("i")  
64 .innerText = "play_arrow";
```

```
41   mainAudio.pause();  
42 }  
43  
44 //prev music function  
45 function prevMusic(){  
46   musicIndex--;  
47   //decrement of musicIndex by 1  
48  
49   //if musicIndex is less than 1 then  
50   musicIndex will be the array  
51   length so the last music play  
52   musicIndex < 1 ? musicIndex =  
53   allMusic.length : musicIndex =  
54   musicIndex;  
55   loadMusic(musicIndex);  
56   playMusic();  
57   playingSong();  
58 }  
59  
60 //next music function  
61 function nextMusic(){  
62   musicIndex++;  
63   //increment of musicIndex by 1  
64  
65   //if musicIndex is greater than  
66   array length then musicIndex will  
67   be 1 so the first music play  
68   musicIndex > allMusic.length  
69   ? musicIndex = 1 : musicIndex  
70   = musicIndex;  
71   loadMusic(musicIndex);  
72   playMusic();  
73   playingSong();  
74 }  
75  
76 // play or pause button event  
77 playPauseBtn.addEventListener(  
78   "click", ()=>{  
79     const isMusicPlay = wrapper.  
80     classList.contains("paused");  
81  
82     //if isPlayMusic is true then call  
83     pauseMusic else call playMusic  
84     isMusicPlay ? pauseMusic() :  
85     playMusic();  
86     playingSong();  
87   });  
88  
89 //prev music button event  
90 prevBtn.addEventListener("click",  
91   ()=>{  
92     prevMusic();  
93   });  
94  
95 //next music button event  
96 nextBtn.addEventListener("click",  
97   ()=>{  
98     nextMusic();  
99   });  
100 }
```

```
101 //update progress bar width according  
102 to music current time  
103 mainAudio.addEventListener(  
104   "timeupdate", (e)=>{  
105     const currentTime = e.target.  
106     currentTime;  
107     //getting playing song current  
108     time  
109     const duration = e.target.  
110     duration;  
111     //getting playing song total  
112     duration  
113     let progressWidth = (  
114     currentTime / duration) * 100;  
115     progressBar.style.width = `${  
116     progressWidth}%`;  
117  
118     let musicCurrentTime = wrapper.  
119     querySelector(".current-time"),  
120     musicDuration = wrapper.  
121     querySelector(".max-duration");  
122     mainAudio.addEventListener(  
123     "loadeddata", ()=>{  
124       // update song total duration  
125       let mainAddDuration =  
126       mainAudio.duration;  
127       let totalMin = Math.floor(  
128       mainAddDuration / 60);  
129       let totalSec = Math.floor(  
130       mainAddDuration % 60);  
131       if(totalSec < 10){  
132         //if sec is less than 10 then  
133         add 0 before it  
134         totalSec = `0${totalSec}`;  
135       }  
136       musicDuration.innerText =  
137       `${totalMin}:${totalSec}`;  
138     });  
139  
140     // update playing song current  
141     time  
142     let currentMin = Math.floor(  
143     currentTime / 60);  
144     let currentSec = Math.floor(  
145     currentTime % 60);  
146     if(currentSec < 10){  
147       //if sec is less than 10 then  
148       add 0 before it  
149       currentSec = `0${currentSec}`;  
150     }  
151     musicCurrentTime.innerText =  
152     `${currentMin}:${currentSec}`;  
153   });  
154  
155 // update playing song current  
156 time on according to the progress  
157 bar width  
158 progressArea.addEventListener(  
159   "click", (e)=>{  
160     let progressWidth =  
161     progressArea.clientWidth;  
162     //getting width of progress bar  
163     let clickedOffsetX = e.offsetX  
164     ; //getting offset x value  
165     let songDuration = mainAudio.  
166     duration;  
167     //getting song total duration  
168  
169     mainAudio.currentTime = (  
170     clickedOffsetX / progressWidth  
171     ) * songDuration;  
172     playMusic();  
173     //calling playMusic function  
174     playingSong();  
175   });  
176 }
```

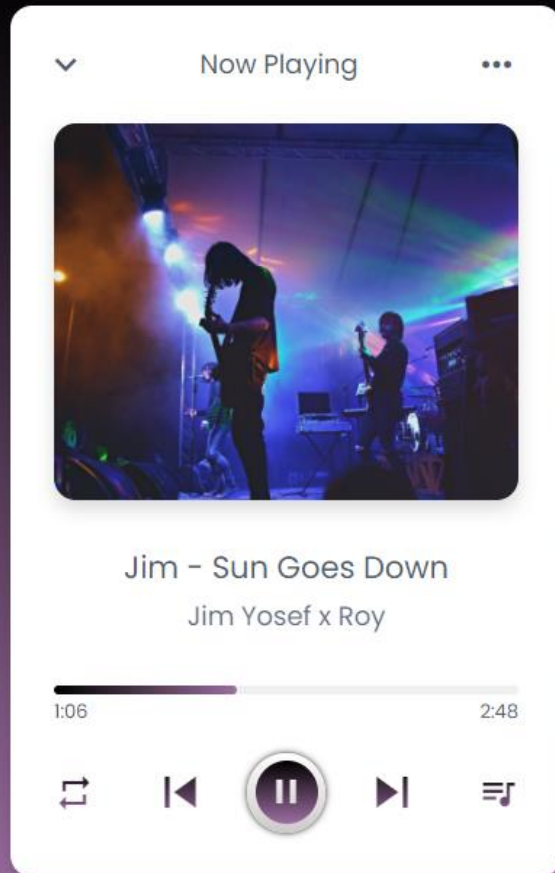
Input:JavaScript

```
121 //change loop, shuffle, repeat i
122 const repeatBtn = wrapper.
123   querySelector("#repeat-plist");
124   repeatBtn.addEventListener(
125     "click", ()=>{
126       let getText = repeatBtn.
127         innerText;
128       //getting this tag innerText
129       switch(getText){
130         case "repeat":
131           repeatBtn.innerText =
132             "repeat_one";
133           repeatBtn.setAttribute(
134             "title", "Song looped");
135           break;
136         case "repeat_one":
137           repeatBtn.innerText =
138             "shuffle";
139           repeatBtn.setAttribute(
140             "title", "Playback shuffled");
141           break;
142         case "shuffle":
143           repeatBtn.innerText =
144             "repeat";
145           repeatBtn.setAttribute(
146             "title", "Playlist looped");
147           break;
148       }
149     });
150 //code for what to do after son
151 g ended
152 mainAudio.addEventListener(
153   "ended", ()=>{
154     // we'll do according to the ico
155     n means if user has set icon to
156     // loop song then we'll repeat t
157     he current song and will do acco
158     rdingly
159     let getText = repeatBtn.
160       innerText;
161     //getting this tag innerText
162     switch(getText){
163       case "repeat":
164         nextMusic();
165       //calling nextMusic function
166       break;
167       case "repeat_one":
168         mainAudio.currentTime = 0
169       ;
170       //setting audio current time t
171       o 0
172       loadMusic(musicIndex);
173       //calling loadMusic function wit
174       h argument, in the argument ther
175       e is a index of current song
176       playMusic();
177       //calling playMusic function
178       break;
179       case "shuffle":
180         let randIndex = Math.floor
181         ((Math.random() * allMusic.
182         length) + 1);
183       //genereting random index/numb w
184       ith max range of array length
185       do{
186         randIndex = Math.floor((
187         Math.random() * allMusic.length
188         ) + 1);
189       }while(musicIndex ==
190       randIndex);
191       //this loop run until the next r
192       andom number won't be the same o
193       f current musicIndex
194       musicIndex = randIndex;
195       //passing randomIndex to musicIn
```

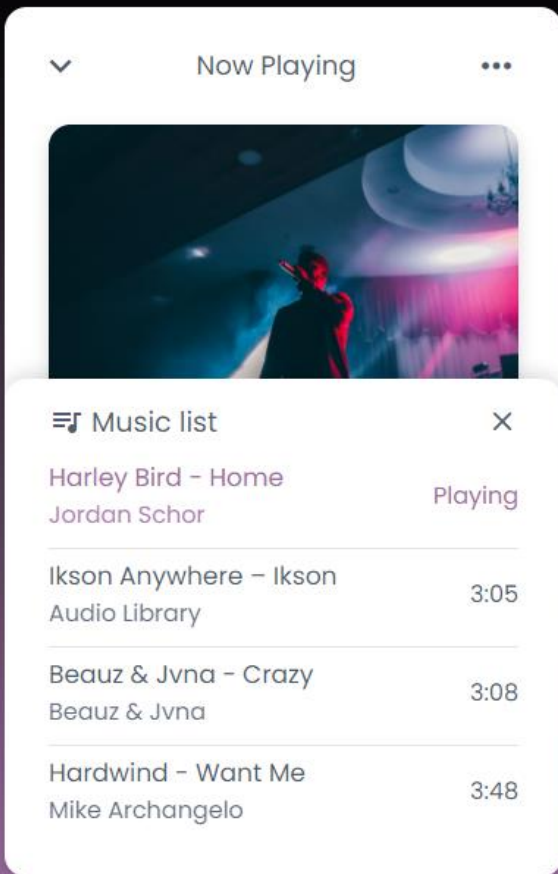
```
161 dex
162 loadMusic(musicIndex);
163 playMusic();
164 playingSong();
165 break;
166 }
167 });
168 //show music list onlick of mus
169 ic icon
170 moreMusicBtn.addEventListener(
171   "click", ()=>{
172     musicList.classList.toggle(
173     "show");
174   });
175   closermoreMusic.addEventListener(
176   "click", ()=>{
177     moreMusicBtn.click();
178   });
179   const ulTag = wrapper.
180     querySelector("ul");
181   // let create li tags accordin
182   g to array length for list
183   for (let i = 0; i < allMusic.
184     length; i++) {
185     //let's pass the song name, arti
186     st from the array
187     let liTag = `<li li-index="${i
188       + 1}">
189       <div class="ro
190       w">
191         <span>${
192         allMusic[i].name}</span>
193         <p>${allMusic[
194         i].artist}</p>
195       </div>
196       <span id="${
197         allMusic[i].src}
198       " class="audio-duration">3:40</s
199       pan>
200       <audio class="${
201         allMusic[i].src}" src="songs/${
202         allMusic[i].src}.mp3"></audio>
203       </li>`;
204       ulTag.insertAdjacentHTML(
205       "beforeend", liTag);
206       //inserting the li inside ul tag
207       let liAudioDuartionTag = ulTag.
208         querySelector(`#${allMusic[i].
209         src}`);
210       let liAudioTag = ulTag.
211         querySelector(`.${allMusic[i].
212         src}`);
213       liAudioTag.addEventListener(
214       "loadeddata", ()=>{
215         let duration = liAudioTag.
216         duration;
217         let totalMin = Math.floor(
218         duration / 60);
219         let totalSec = Math.floor(
220         duration % 60);
221         if(totalSec < 10){
222           //if sec is less than 10 then ad
223           d 0 before it
224           totalSec = `0${totalSec}`;
225         };
226         liAudioDuartionTag.innerText
227         = `${totalMin}:${totalSec}`;
228         //passing total duation of song
229         liAudioTag.
230         setAttribute("t-duration", `${
231         totalMin}:${totalSec}`);
232         //adding t-duration attribute wi
```

```
201 //adding t-duration attribute wi
202 th total duration value
203 });
204 //play particular song from th
205 e list onlick of li tag
206 function playingSong(){
207   const allLiTag = ulTag.
208     querySelectorAll("li");
209   for (let j = 0; j < allLiTag.
210     length; j++) {
211     let audioTag = allLiTag[j].
212     querySelector(".audio-duration"
213     );
214     if(allLiTag[j].classList.
215     contains("playing")){
216       allLiTag[j].classList.
217       remove("playing");
218       let adDuration = audioTag.
219       getAttribute("t-duration");
220       audioTag.innerText =
221       adDuration;
222     }
223     //if the li tag index is equal t
224     o the musicIndex then add playin
225     g class in it
226     if(allLiTag[j].getAttribute(
227     "li-index") == musicIndex){
228       allLiTag[j].classList.add(
229       "playing");
230       audioTag.innerText =
231       "Playing";
232     }
233     allLiTag[j].setAttribute(
234     "onclick", "clicked(this)");
235   }
236 }
237 //particular li clicked function
238 function clicked(element){
239   let getLiIndex = element.
240   getAttribute("li-index");
241   musicIndex = getLiIndex;
242   //updating current song index wi
243   th clicked li index
244   loadMusic(musicIndex);
245   playMusic();
246   playingSong();
247 }
```


Output:



Output:



Presented By:

Name-Tejas Saxena
Roll no.-CSE-19/083

Name-Tarun Saxena
Roll no.-CSE-19/082

Thanks