iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: VI Month of publication: June 2025

DOI: https://doi.org/10.22214/ijraset.2025.72753

www.ijraset.com
Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 13 Issue VI June 2025- Available at www.ijraset.com*

# Research Paper on a Music Player Application (ECHOIC)

Anish Prasad Rajak[1], Abdullah Salim[2], Abhishek, Priyanka Chakraborty[3], Debajyoti Ghosh[4], Bonhhi Bose[5], Dr. Availi Banerjee[6]
*Guru Nanak Institute of Technology Kolkata, India*

***Abstract: This presents the design and implementation of a Music Player Application that enables users to play, organize, and manage their local audio files efficiently. The primary objective of this project is to develop a lightweight, user-friendly, and functional music player that offers essential playback features such as play, pause, stop, next, previous, shuffle, repeat, and volume control***
***The application also includes features such as playlist creation, song categorization (by artist, album, or genre), and a visually appealing user interface for enhanced usability. The project emphasizes modular code design, smooth audio playback, and Technologies used in the development include [mention tech stack, e.g., React Native/Flutter for the front-end, Node.js/Django for the back-end, and MongoDB/MySQL for the database]. The project follows software development best practices such as MVC architecture, RESTful APIs, and responsive design principles. This application serves as a practical implementation of key computer science concepts including object-oriented programming, file handling, multimedia APIs, and user interface design. The project not only showcases technical skills but also addresses user-centric design and performance optimization.***

## I. PROJECT OBJECTIVES

When you have completed this module you will be able to-:
Basic functions such as playing music are totally free, but you can also choose to upgrade to Echoic . Either way, you can:
Choose what you want to listen to with Browse and Search
Find what you're looking for with Search, including:
1.Songs 2.Albums 3.Artists 4.Playlist
On mobi1e,tab1es and desktop you can also use Search to browse categories such as genres, moods charts, and new releases.
Get recommendations from personalized features, such as Discover Weekly, Release Radar, and Daily Mix.

Or Search the name of any playlist made for you. Build collection. Find Made for you playlist in Home . or podcast, you can find it in Your Library.
See what friends, artists, and celebrities listen to

Follow artists to receive notifications and never miss a new release.
Go to the artist's profile.
Select Follow.
Follow friends to see what they're listening to in Friend Activity.
Create your own Radio station.

Keep the mood going. Echoic radio creates a collection of songs based on anyartist, album, playlist, or song of your choice. It even updates over time to keep fresh.
Go to anyartist, album, playlist, or song.
Select Go to radio.
*Concerts Alerts: Notifyusers about live concerts of favourite artists in your area when you sign up for an account and setup notifications.
*Dual audio mix: Dual audio mixing is the process of mixing two audio sources or creating Separate mixes for different audiences.

## II. SYSTEM PLANNING

### A. *Survey of Technologies*

#### 1) *Front End*

##### a) *HTML*

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.
Web browsers receive HTML documents from a web server or from local storage
and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.
HTML elements are the building blocks of HTML pages. With HTML
constructs, images and other objects such as interactive forms maybe embedded into the rendered page.
HTML provides a means to create structured documents bydenoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items.
HTML elements are delineated bytags, written using angle brackets. Tags such as and <input /> directly introduce content into the page. Other tags such as surround and provide information about document text and mayinclude other tags as sub- elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.
HTML can embed programs written in a scripting language such as JavaScript, which affects the behaviour and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintained of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

##### b) *CSS*

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technologyof the World Wide Web, alongside HTML and JavaScript.
CSS is designed to enable the separation of presentation and content, including layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting byspecifying the relevant CSS in a separate .css file which reduces complexity and repetition in the structural content as well as enabling the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.
Separation of formatting and content also makes it feasible to present the same
markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.
The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable. The CSS specifications are maintained bythe World Wide Web Consortium (W3C).
Internet media type (MIME type) text/css is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.

##### c) *React.JS*

ReactJS is JavaScript libraryused for building reusable UI components.
According to React official documentation, following is the definition —
React is a libraryfor building composable user interfaces. It encourages the creation of reusable UI components, which present data that changes over time. Lots of people use React as the V in MVC. React abstracts away the DOM from you, offering a simpler programming model and better performance. React can also render on the server using Node, and it can power native apps using React Native. React implements one- way reactive data flow, which reduces the boilerplate and is easier to reason about than traditional data binding.
React Features

- JSX — JSX is JavaScript syntax extension. It isn't necessary to use JSX in React development, but it is recommended.
- Components — React is all about components. You need to think of everything as a component. This will help you maintain the code when working on larger scale projects.

- Unidirectional data flow and Flux — about your app. Flux is a pattern that helps keeping your data un React Advantages
- Uses virtual DOM which is a JavaScript object. This will improve apps performance, since JavaScript virtual DOM is faster than the regular DOM.
- Can be used on client and server side as well as with other frameworks.
- Component and data patterns improve readability, which helps to maintain larger apps.

React Limitations
- Covers onlythe view layer of the app, hence you still need to choose other technologies to get a complete tooling set for development.
- Uses inline templating and JSX, which might seem awkward to some developers.

*d) Language used JavaScript*

JavaScript (JS) is a lightweight, interpreted, or just-in-time compiled
programming language with first-class functions. While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat. JavaScript is a prototype-based, multi-paradigm, single-threaded, dynamic language, supporting object-oriented, imperative, and declarative (e.g. functional programming) style.

JavaScript runs on the client side of the web, which can be used to design / program how the web pages behave on the occurrence of an event.

JavaScript is an easy to learn and also powerful scripting language, widely used for controlling web page behaviour.

JavaScript is not "Interpreted Java". In a nutshell, JavaScript is a dynamic scripting language supporting prototype based object construction. The basic syntax is intentionally similar to both Java and C++ to reduce the number of new concepts required to learn the language.

Language constructs, such as if statements, for and while loops, and switch and try catch blocks function the same as in these languages (or nearly so).

JavaScript can function as both a procedural and an object oriented language. Objects are created programmatically in JavaScript, byattaching methods and properties to otherwise emptyobjects at run time, as opposed to the syntactic class definitions common in compiled languages like C++ and Java.

*2) BackEnd*

*a) Node.js*

A common task for a web server can be to open a file on the server and return the content to the client.

Here is how PHP or ASP handles a file request:
- Sends the task to the computer's file system.
- Waits while the file system opens and reads the file.
- Returns the content to the client.
- Ready to handle the next request.

Here is how Node.js handles a file request:
- Sends the task to the computer's file system.
- Ready to handle the next request.
- When the file system has opened and read the file, the server returns the content to the client.

Node.js eliminates the waiting, and simply continues with the next request.

Node.js runs single-threaded, non- blocking, asynchronously programming, which is very memory efficient.

What Can Node.js Do?
- Node.js can generate dynamic page content
- Node.js can create, open, read, write, delete, and close files on the server
- Node.js can collect form data

- Node.js can add, delete, modifydata in your database

What is a Node.js File?

- Node.js files contain tasks that will be executed on certain events
- A typical event is someone trying to access a port on the server
- Node.js files must be initiated on the server before having anyeffect
- Node.js files have extension ".js".

*b) Npx:*

NPX (NPM Package Runner) Commands
List of useful npx (NPM Package Runner) commands.

## III. REQUIREMENT AND ANALYSIS

### *A. Problem Definition*

The biggest drawback is the low audio quality,
MP3 uses the lossy
algorithm which deletes the lesser audible music content to reduce the file size, thus compromising on the music quality, Music piracy increased to a greater extent,
Cheaper or free duplicate versions of the original music files are available on the Internet for download .
There are some **disadvantages of the existing system.**

- The sound qualityof the MP3 format is not as good as that of the CD , So , CD players provide clearer audio than do MP3 players , Although MP3s can be compressed at a higher bit rate , Most are encoded at 128 kilobits per second , compared with CDs , on which the listener receives sound at 196 kilobits per second , about 50 per cent higher .
- The data is susceptible to losses due to the malware or virus attacks , The people who used the file-sharing service , They had their computers accessed by the hackers , MP3 players are generally more expensive than CD players.
- MP3 compression maydiscard as much as 90 percent of the data from the original recording without a significant drop in sound quality , Nevertheless , The listeners with the exceptional hearing or high-end earphones maydetect slight differences between the MP3 file & the original uncompressed CD recording .
- Unlike CDs , The albums on MP3s cannot be resold , When the people purchase the song from iTunes or another online MP3 store, Theyare not so much buying the song as indefinitely leasing it , This may limit the ability of the owners of MP3 players to refresh their libraries frequently , unlike owners of CD players, they cannot legally trade their songs for new ones.

### *B. Requirement Specification*

Echoic is immediately appealing because you can access content for free bysimply signing up using an Email address or by connecting with Facebook, Gmail Account. If you're not keen on monthly subscription fees for Echoic, or just want to dip your toe in and test it out, it's out, it's easy to get started and there's no commitment.
You can find out the main differences between Echoic Free and Premium in our separate feature but as a quick summary, the free version is ad- supported, much like radio stations. The free version of Echoic can be accessed on PC, laptop and mobile phone, but the full service needs a Echoic Premium subscription.

## IV. MODULES OF PROPOSE SYSTEM

### *A. Registration*

Using this module customer can register or login into the system in order to use that system. User can search the for Music and create it's own playlist.

### *B. Music-Streaming*

The streaming method doesn't require the downloading of the entire file.
The audio that the user requests is delivered to him in small packets to play the music instantly.

### *C. Search*
The entire idea of a music streaming application is to give the listeners the opportunityto search for the type of music theywant to listen to as per their mood.

### *D. Playlists*
What could be a better option that giving your users a platform where theycan create a list of all their preferred tracks in a single spot, classified according to their mood.

### *E. Social-Sharing*
It is a well- known saying that the success your application gets is directly related to the promotions it gets on social networking websites.

### *F. Offline-Mode*
This feature permits users to listen to their favourite music even without the internet connection. It utilizes the local storage of the device to cache the audio information.

### *G. Push Notifications*
Push-Notifications are not just a must-have feature, however, the most helpful feature through which you can stun your audience by giving them astonishing offers.

### *H. Payment*
Payment can be done byusing a credit card, debit card, internet banking, online. Payment entry is highly secure and trusted.

## V. SOFTWARE AND HARDWARE REQUIREMENT

### *A. Hardware Requirement*
Hardware requirement for this system are as Follows:

| | Processor | RAM | Disk Space |
|---|---|---|---|
| Client side | Intel P4 or equivalent | 512MB | 2GB |
| | Intel P4 or equivalent | 512MB | 1GB |
| Server side | Server Environment Capable H/w | 2GB | As per the size of requirements DBMS |

| | | |
|---|---|---|
| FROW T EN D | Htm 15,css,J S, React. I S | |
| BACK END | Nodejsnpx,yarn | |
| SYSTEM | Windows 1 0 | |

*B. System Design*

Data Design-:

Registration table-:

| ATTRIBUTE |
|---|
| First Name |
| Last Name |
| Phone no |
| Email lD |
| Password |
| Conform Password |

Login Table-:

| A FTRIBUTE |
|---|
| User naive |
| PaS5WOrd |
| Forget Password |

Playlist Table-

| ATTRIBUTE |
|---|
| Create PlayliSt |
| Choose Categries |
| Add Music |
| List Music |
| PAYMENT T |
| ATTRIBUTE |
| First Name |
| Last Name |
| Total Amount |
| Card Number |
| Cvv Number |
| Id number |
| Address |

FEEDBACK TABLE-:

| ATTRIBUTE |
|---|
| NAME |
| EMAIL ID |
| PHONE NO |
| DESCRIPTION |

ADMIN TABLE-:

| ATTRIBUTE |
|---|
| NAME |
| PASSWORD |

Data Integrity and Constraints:
REGISTRATION TABLE

| ATTRIBUTE | DATA TYPE | €ONSTR NT |
|---|---|---|
| First Name | Varchar(30) | Not Null |
| Last Name | Varchar(30) | Not Null |
| Phone No | Long{10) | Pñmmykey |
| Email Id | Varchar(50) | Not null |
| City | Varchar(30) | Not Null |
| Password | Varchar{2O) | Not Null |
| Conf¥m Password | varchar{20} | uoi ai |

£A i TAB&•

| ATTRIBUTE | DATA TYPE | CONSTRAINT |
|---|---|---|
| User name | Varchar(30) | Primary key |
| Password | Varcfzar{20) | Not null |

PAYMENTTA8L1-

| ATTRIBUTE | DATA TYPE | CONSTRAINT |
|---|---|---|
| Total Amount | Int(10) | Not null |
| Atm Card No | Long (15) | Not null |
| Cw No | Long (15) | Not null |
| Book id | Int(4) | Primary Key |

FEEDBACK TABLE:-

| ATTRIBUTE | DATA TYPE | CONSTRAIN T |
|---|---|---|
| Name | Varchar (30) | Not n ull |
| Email IN | Varchar (20) | Nol nul |
| Phone no | Int(10) | Pr im ary key |
| Descr ipt ion | Varchar (90) | Not nul |

ADMIN TABLE:-

| ATTRI BUTE | DATA TYPE | CONSTRAI NT |
|---|---|---|
| User name | Varchar(30) | Primary key |
| Password | Varchar( 30) | Not nul |

Test Cases

| SR NO | **Form Name** | **Test c»d I"‘** | **Step or Procedure** | **Input Test D"‘** | **Expected Result** | **Actual Output** | **Pass/ Fail** |
|---|---|---|---|---|---|---|---|
| I | Login | Check login with val id input | Wrong username with correct password | User name: admin PASS:admin | Display Message: “Invalid Username OF Password | Display Message “lnvalid Usemame O¥ Password | PASS |
| 2 | Login | Check login with val id input | If Numbers are inserted | User name: admin PASS:admin | Display Message: “Invalid Username Password | Display Message: “lnvalid Usemame Password | PASS |
| 3 | User | Check Alphabetic Values | If mobile number is morethan 10 digit | 9653329853 | Display Message: “only Characters are allowed” | Display Message: “Only Characters are allowed, | PASS |
| 4 | User | Check Email id | Wrong username with correct password | Name: rahul456 | Display Message: “Enter 1 0 digit number only” | Display Message: “Enter 1 0 digit number only” ” | PASS |
| 5 | User | Check Emailid | If mobile number is more than 10 digit | 828666425 | Display Message: “Phone number cannot be less than l o digit” | Display Message: “Phone number cannot be less than l o digit” ” | PASS |
| 6 | User | Check Email id | If @mail.coin is not speciJ\ed | nik@gmail.com | Display Message: “Email is expected | Display message : “ Email is Expected | PASS |

## VI. SYSTEM CODING, IMPLEMENTATION AND TESTING

```
  <!DOCTYPE html>
  <html lang="en">
  <head>
  <meta charset="UTF-8" />
  <meta
  ort"

  name="viewp

  content="widt
  h=device-width,                                   initial- scale=1.0"/>
  <title>ECHOIC</title>
<style> body ( margin: 0;
  font-family: Arial, sans-serif; background-color: #121212; color: #ffffff;
  display: flex;
  flex-direction: column; height: l00vh;
  .header (
  background-color:#1db954; padding: 15px 20px;
  text-align: center; font-size: 1.5rem;

container ( display:flex; flex: 1;
  overflow: hidden;

  .sidebar (
  background-color: #181818;
  width: 20%;
  min-width: l50px; padding: 20px;
  box-sizing: border-box; display: flex;
  flex-direction: column;

.sidebar nav ul ( list-style: none; padding: 0;
.sidebar nav ul li ( margin: 15px 0; padding: 10px;
  background-color:#282828; text-align: center;
  border-radius: 5px; cursor: pointer;
  transition: background 0.3s;

.sidebar nav ul li:hover  ( background-color:#383838;

.main-content ( flex: 1; padding: 20px;
  box-sizing: border-box; overflow-y: auto;

.track—list, .playlist { margin-bottom: 30px;
.track-item p ( margin: 5px 0; font-size: 1.1rem;

  audio (
  width: 100%;
  margin: 15px 0;

  button (
  background-color: #1db954;
  border:none; color: #ffffff;
  padding: 10px20px; font-size: lrem; cursor: pointer; border-radius: 5px; margin-top: 10px;

  button:hover (
  background-color: #1ed760;

.search-box input ( width: 100%; padding: 10px;

  font-size: lrem; border-radius: 5px; border: none; outline: none; margin-bottom: 10px;

  .playlist ul {
  list-style: none; padding: 0;

.playlist ul li ( padding: 10px; margin-bottom:5px;
  background-color: #282828;
  border-radius: 5px; cursor: pointer;
  transition: background 0.2s;

.playlist ul li:hover  ( background-color:#383838;
  @media (max-width: 768px) (
  .container (
  flex-direction: column;

.sidebar ( width: 100%;
  flex-direction: row;
  justify-content: space-around;

.sidebar navul ( display: flex;
  justify-content:space-around; width: 100%;

.sidebar navul li ( margin: 0; padding: 10px; flex: 1;
  font-size: 0.9rem;
  </style>
  </head>
  <body>
  <header class="header">ECHOIC</header>

  <div class="container">
  <aside class="sidebar">
  <nav>

  dli onclick="alert('Home clicked')">Home</li> dli onclick="alert('Search clicked')">Search</li>
  <li onclick="alert('Library clicked')">Library</li>
         <
         </ul
  </nav>
  </aside>
```

```
<main class="main-content">
<section class="track-list">
<h2>Now Playing</h2>
<div class="track-item">

<p><strong id="track-title">Track Titled/strong /pt Up id="track-artist">Artist</pt
</div>
<audio id="audio-player" controls>
<source id="audio-source" src="" type="audio/mpeg" /> Your browser does not support the audio element.
</audio>

          <li

<button onclick="playPauseAudio()"
>Play/Pause</button>
</section>

<section class="playlist">
<h2>Playlist</h2>
<div class="search-box">
<input                                   type="text"
id="search-

Source')">Sample Beat 2</li>
onclick="playTrack('https:
//samplelib.com/lib/previe w/mp3/sample- 9s.mp3', 'Sample Beat 3', 'Free Source')">Sample Beat 3</li>

</section>
</maim
</div>
input"

placeholder="

<script>

Search tracks..."
oninput="filterPlaylist()" />
</div>
<ul id="playlist">

onclick="playTrack('https:
//samplelib.com/lib/previe w/mp3/sample- 3s.mp3', 'Sample Beat 1', 'Free Source')">Sample Beat 1</li>

onclick="playTrack('https:
//samplelib.com/lib/previe w/mp3/sample- 6s.mp3', 'Sample Beat 2', 'Free
function playTrack(trackPath, trackTitle, CackArtist) {const audiosource = document.getElementById('audio-source'); audioSource.src =
trackPath;

document.getElementById('track-title').innerText = trackTitle; document.getElementById('track-artist').innerText = trackArtist;
```

```
  const player = document.getElementById('audio-player'); player.load();
  player.play();


  function playPauseAudio() {
  const player = document.getElementById('audio-player'); if(player.paused) {
  player.play(); else player.pause();


  function filterPlaylist() (
  const input input').value.toLowerCase();
  document.getElementById('search-
  const items = document.querySelectorAll('#playlist li'); items.forEach((item) => {
  const text = item.textContent.toLowerCase(); item.style.display= text.includes(input) ?'' : 'none'',

  </script
  </body>
  </html>
  <!DOCTYPE html>
  <html 1ang="en">
  <head>
  <meta charset="UTF-8" />
  <meta name=”viewport" content="width=device-width, initial-sca1e=1.0"/>
  <tit1e>ECHOIC</tit1e>
<sty1e> body ( margin: 0;
  font-family: Arial, sans-serif; background-color: #121212; color: #ffffif,
  display: flex;
  flex-direction:    column;
  height: l00vh;

  .header (
  background-color: #1db954 padding: 15px 20px;
  text-align:  center;
  font-size: 1.5rem;

.container  ( display: flex; flex: l;
  overflow: hidden;

  .sidebar (
  background-color: #181818;
  width: 20%;
  min-width: l50px; padding: 20px;
  box-sizing: border-box;
  display: flex;
  flex-direction: column;

.sidebar nav ul ( list-style: none; padding: 0;
.sidebar nav ul li ( margin: 15px 0; padding: 10px;
  background-color: #282828; text-align: center;
  border-radius: 5px; cursor: pointer;
  transition: background 0.3s;

.sidebar nav u11i:hover  ( background-color: #383838;


.main-content ( flex: 1; padding: 20px;
  box-sizing: border-box;
  overflow-y: auto;


.track-list, .playlist  ( margin-bottom: 30px;

.track-item p { margin: 5px 0; font-size: 1.1rem;
  audio (
  width: 100%;
  margin: 15px 0;
  button (
  background-color: #1db954; border: none;
  color: #fIfffP, padding: 10px 20px; font-size:  lrem; cursor: pointer; border-radius: 5px; margin-top: 10px;

  button:hover {
  background-color: #1ed760;

.search-box input ( width: 100%; padding: 10px; font-size:  lrem; border-radius: 5px; border: none; outline: none;
  margin-bottom: 10px;
  .playlist   u1 {
  list-style: none;
  padding: 0;

.playlist ulli ( padding: 10px; margin-bottom: 5px;
  background-color: #282828; border-radius: 5px;
  cursor: pointer;
  transition: background 0.2s;

.playlist ulli:hover  ( background-color: #383838;

  @media (max-width: 768px) (
  .container (
  flex-direction: column;

.sidebar ( width: 100%;
  flex-direction: row;
  justify-content: space-around;


.sidebar nav ul ( display: flex;
  justify-content: space-around; width: 100%;

.sidebar nav ul li ( margin: 0; padding: 10px; flex: 1;
  font-size: 0.9rem;

  </styled
  </head>
  <body>
  <header c1ass="header">ECHOIC</header>
  <div c1ass=”container">
  <aside class=”sidebar">
  <nav>
  full
  <1i   onc1ick="alert('Home   c1icked')">Home</1i>
  <1i   onc1ick="alert('Search   c1icked')">Search</1i>
  <li   onc1ick="alert('Library   c1icked')">Library</1i>
  </ul>
  </nav>
  </aside>

  <main class=”main-content">
  <section      c1ass="track-1ist">
  <h2>Now P1aying</h2>
  <div class="track-item">
  <p><strong                                    id="track-tit1e">Track     Tit1e</strong></p>
  <p id="track-artist">Artist</p>
  </div>
  <audio id="audio-player” controls>
  <source id="audio-source" src="" type="audio/mpeg" /> Your browser does not support the audio element.
  </audio>
  <button onclick="p1ayPauseAudio()">P1ay/Pause</button>
  </section>


  <section c1ass="p1ay1ist">
  <h2>P1ay1ist</h2>
  <div                              c1ass="search-box">
          <input                    type="text"        id="search-input"        placeholder="Search        tracks..."
  oninput=”filterPlaylist()" />
  </div>
  <u1 id="p1ay1ist">
          <li                       onc1ick="p1ayTrack('https://sample1ib.com/lib/preview/mp3/sample-3s.mp3',    'Sample Beat 1', 'Free
  Source')”>Samp1e Beat 1</1i>
          <1i                       onc1ick="p1ayTrack('https://samp1e1ib.com/lib/preview/mp3/sample-6s.mp3',    'Sample Beat 2', 'Free
  Source')">Samp1e Beat 2</1i>
          <1i                       onc1ick="p1ayTrack('https://samp1e1ib.com/lib/preview/mp3/sample-9s.mp3',    'Sample Beat 3', 'Free
  Source')">Samp1e Beat 3</li>
  </ul>
  </section>
  </main>
  </div>
  <script>
  function p1ayTrack(trackPath, trackTitle, trackArtist) {
```

```
const audiosource = document.getE1ementById('audio-source'); audioSource.src = trackPath;

document.getE1ementById('track-title').innerText = trackTitle; document.getE1ementById('mack-artist').innerText = trackArtist;
const player = document.getE1ementById('audio-player'); player.load();
player.play();

function playPauseAudio() (
const player = document.getE1ementById('audio-player'); if(player.paused) (
player.play();
} else ( player.pause();

function filterPlaylist() (
const input = document.getE1ementById('search-input').value.toLowerCase(); const items = document.querySelectorA11('#p1ay1ist li');
items.forEach((item) => (
const text = item.textContent.toLowerCase(); item.style.display= text.includes(input) ? '' : 'none“,

</script>
</body>
</html>
```

## VII. TEST RESULT

| Sr. No. | TEST CONDITION | STEPS OR PROCEDURE | INPUT TEST DATA | EXCEPTED RESULT | ACTUAL OUTPUT | PASS/ FAIL |
|---|---|---|---|---|---|---|
| l | Check whether product is added properly to specific category | Add product from admin dashboard and it will display in user dashboard | Admin Dashboard Add product wood guitar in guitar category With all details. | Guitar.jsp page Here guitar images and all details are showu . | Guitar.jsp page Productname: wood guitar Price:4Sd9<br><br>Data is displayed in proper layout | Pass |
| 2 | Check whether one or more products are added in the cart or riot | User can check products and click on add to cart button it will add products in cart, user can add one or more products in Can | User dashboard User check product and add the product in cart and go back for a adding more products | User dashboard (cart details) Here all the products are displayed in cart table | User dashboard (cart details) Here are all the products are displayed which is added from the user and the total amount is displayed | Pass |
| 3 | Check whether after user has done payment user can get mail or not | After user has done the payment process user can get mail from admin about the delivery details. | After admin get payment from user admin send the delivery details mail to user. | User get mail from admin about delivery details | User get mail from admin about delivery details | Pass |

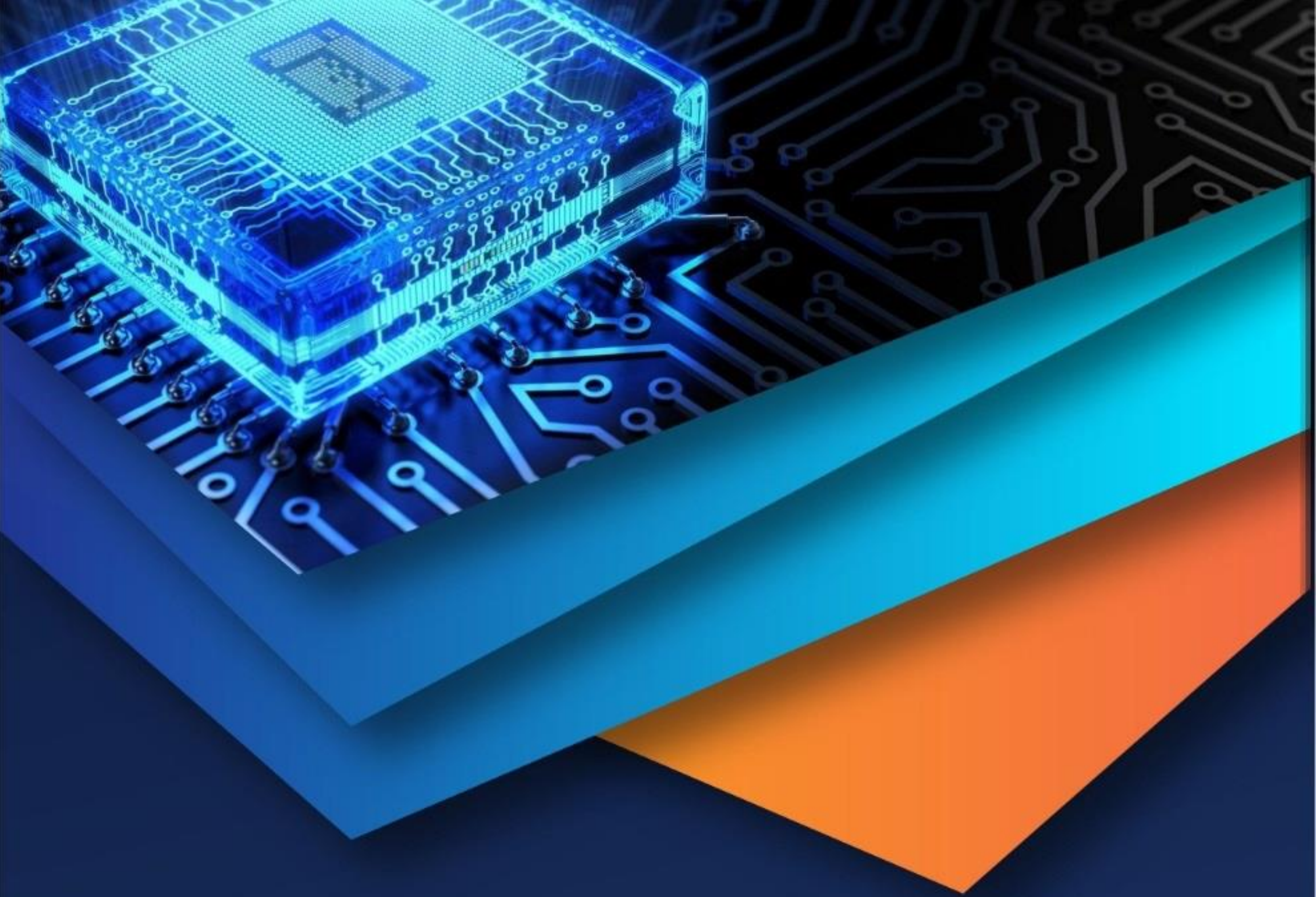
## VIII. MODIFICATIONAND IMPROVEMENT

During the development, several opportunities for enhancement were identified to improve functionality, usability, and overall user experience. The following modifications and improvements are either implemented during the project or proposed for future updates:

1) Enhanced User Interface (UI): The interface was redesigned with a more modern and minimalistic layout, making navigation intuitive and visually appealing. Improvements include updated icons, animations, and a responsive layout for various screen sizes.
2) Improved Audio Controls: The application now includes smoother transitions between tracks, volume control integration, and notification- based playback control, enhancing ease of use during multitasking.
3) Playlist and Queue Functionality: A custom playlist feature was added, allowing users to create, edit, and delete playlists. A dynamic play queue was also introduced for better control over song order during playback.
4) Search and Filter Options: A real-time search bar and filters (by genre, artist, album, etc.) were added to help users quickly locate specific songs in their library.
5) Background PlaySupport: The player continues functioning even when the app is minimized or the screen is locked, improving user convenience.
6) Metadata and Album Art Display: The app now fetches and displays metadata such as album name, artist, and album art, enhancing the listening experience.
7) Bug Fixes and Performance Optimization: Several bugs related to file access,
8) audio lag, and UI glitches were resolved. The application was also optimized for faster loading times and lower memory usage.
9) Securityand Permissions Handling: Improvements were made to better handle app permissions and ensure user data privacy, especially when accessing media files.

## REFERENCES

References that helped me building this website:
[1] www.w3schools.com (For HTML, CSS and JS)
[2] www.tutoria1spoint.com
[3] www.envanto.com (For website UI inspiration)
[4] www.carbonmade.com
[5] www.behance.net
[6] React- The Complete Guide (incl Hooks ,React Router, Redux)



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)