HTML

File Name : INDEX.HTML

CODE :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="mystyle.css">
</head>
<body>

    <header>
        <h1>Music Database</h1>

     <label for="select">Search for Music</label>
<div>
  <button id="searchButton1" style="display: none">Search by
Lyrics</button>
  <button id="searchButton2" style="display: none">Search by
Artist</button>
  <button id="searchButton3" style="display: none">Search by
Song</button>
  <button id="searchButton4" style="display: none">Search by
Movie</button>
</div>


    </header>

                <main>
                    <!-- Popular Artists Section (Unmodified) -->
                    <section id="popular-artists" class="section">
                        <h2>Popular Artists</h2>
```

```
                    <div class="artist-list">
                        <!-- Add popular artist items here -->
                         <div class="item">
                            <img
src="https://encrypted-tbn0.gstatic.com/images?q=tbn:

ANd9GcTkSFPWAI9FU0LYbqgBZ6Ryt1eOBQrh-NvdUQ&usqp=CAU" alt="Singer 1">
                            <div class="item-info">
                                <h3>Taylor Swift</h3>
                            </div>
                        </div>
                        <div class="item">
                            <img
src="https://i.scdn.co/image/ab6761610000e5ebdf9a1555f53a20087b8c5a5c"
alt="Singer 2">
                            <div class="item-info">
                                <h3>Kishore Kumar</h3>
                            </div>
                        </div>
                        <div class="item">
                            <img
src="https://www.dolby.com/globalassets/consumer/the-weeknd/the_weeknd.
jpg" alt="Singer 2">
                            <div class="item-info">
                                <h3></h3>
                            </div>
                        </div>
                    </div>


                </section>


                <!-- Popular Albums Section (Unmodified) -->
                <section id="popular-albums" class="section">
                    <h2>Popular Albums</h2>
                    <div class="albums-list">
                        <!-- Add popular album items here -->
                        <!-- ... -->
                        <div class="item">
                            <img
src="https://www.heroesandvillains.app/images/cover.jpg" alt="Heroes &
Villains">
```

```
                                <div class="item-info">
                                    <h3>Heroes & Villains</h3>
                                    <p>Created by: Metro boomin</p>
                                </div>
                            </div>
                            <div class="item">
                                <img
src="https://t2.genius.com/unsafe/300x300/https%3A%2F%2Fimages.genius.c
om%2F960edcb36156c3aed9cb70ede250780a.1000x1000x1.jpg" alt="Lover ">
                                <div class="item-info">
                                    <h3>Lover</h3>
                                    <p>Created by: Taylor Swift</p>
                                </div>
                            </div>
                        </div>
                    </section>


                    <!-- Your Playlists Section -->
                    <section id="your-playlists" class="section">
                        <h2> Playlists Management</h2>
                        <div class="playlist-list">
                            <!-- Add user's playlists here -->
                            <!-- Example playlist item -->
                            <div class="item">
                                <a href="view-song.html">My Playlist
1</a>
                            </div>

                    </section>


                </main>
                <footer>
                    <p>&copy; 2023 Music Database</p>
                </footer>
                <script src="playlist.js"></script>
            </body>
            </html>
```

CSS:

FILENAME: styles.css

CODE:

```
/* Updated CSS for the entire page */

body {
    font-family: 'Arial', sans-serif;
    background-color: #f4ecec;
    margin: 0;
    padding: 0;
}

/* Header styles */
header {
    background-color: #020101;
    color: #f2f6f2;
    text-align: center;
    padding: 20px 0;
}

h1 {
    font-size: 36px;
    margin: 0;
}

/* Search input styles */
#search {
    width: 100%;
    max-width: 400px;
    padding: 10px;
    font-size: 16px;
    border: none;
    border-radius: 25px;
    margin-top: 20px;
}

/* Main content styles */
main {
    padding: 20px;
}

/* Section headers */
.section h2 {
```

```
    font-size: 28px;
    color: #000000;
    margin-bottom: 20px;
}

/* Item styles */
.item {
    display: flex;
    align-items: center;
    justify-content: space-between;
    margin-bottom: 20px;
    padding: 20px;
    background-color: #15cbb3;
    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
    transition: transform 0.2s ease-in-out;
    border-radius: 8px;
    position: relative;
}

.item:before {
    content: "\f005"; /* FontAwesome star icon for playlist items */
    font-family: FontAwesome;
    position: absolute;
    top: 10px;
    right: 10px;
    font-size: 24px;
    color: #ffcc00;
}

.item:hover {
    transform: scale(1.02);
    box-shadow: 0 4px 8px rgba(4, 4, 4, 0.2);
}

.item img {
    max-width: 80px;
    max-height: 80px;
    margin-right: 20px;
    border-radius: 50%;
    box-shadow: 0 0 4px rgba(0, 0, 0, 0.3);
    transition: transform 0.2s ease-in-out;
}
```

```css
.item img:hover {
    transform: scale(1.1);
}

.item-info {
    flex: 1;
}

/* Your Playlists Section Styles */
#your-playlists {
    background-color: #ffcc00;
    padding: 20px;
    border-radius: 8px;
}

.playlist-list .item {
    background-color: #ffebcc;
    position: relative;
}


.playlist-list .item .edit-button {
    background-color: #e2e0e0;
    border: none;
    font-size: 14px;
    cursor: pointer;
    position: initial;
}
.edit-button:hover {
    text-decoration: underline;
}

/* Playlist Creation Section Styles */
#create-playlist-form {
    background-color:rgb(9, 9, 8);
    padding: 32px;
    border-radius: 8px;
    margin-top: 29px;
}

#create-playlist-form h3 {
    font-size: 24px;
    padding-left: 190px;
```

```
    color: #f9f8f8;
    margin-bottom: 10px;

    /* display: inline; */
}

#create-playlist-form form {
    display: inline-block;
     padding-left: 20px;

    flex-direction: column;
}

#create-playlist-form label {
    font-size: 18px;
    color: #fffdfd;
    margin-bottom: 10px;
}

#create-playlist-form input[type="text"] {
    padding: 10px;
    font-size: 15   px;
    border: none;
    border-radius: 3px;
    margin-bottom: 31px;
    display: inline;
    width: 200px;
}

#create-playlist-form button[type="submit"] {
    background-color: #333;
    color: rgb(0, 0, 0);
    border: none;
    padding: 10px 20px;
    border-radius: 8px;
    cursor: pointer;
    font-size: 18px;
    transition: background-color 0.3s ease-in-out;
}

#create-playlist-form button[type="submit"]:hover {
    background-color:rgb(220, 22, 22);
}
```

```
.Inline{
    display: inline-block;
}
```

Javascript:

```
const searchButtons = {
  1: document.getElementById("searchButton1"),
  2: document.getElementById("searchButton2"),
  3: document.getElementById("searchButton3"),
  4: document.getElementById("searchButton4"),
};

// Add an event listener to each search button
for (let value in searchButtons) {
  if (searchButtons.hasOwnProperty(value)) {
    searchButtons[value].addEventListener("click", function () {
      // Perform the search based on the clicked button
      // You can add your search logic here
      console.log(`Performing search for option ${value}`);
    });
  }
}
```

FILE NAME : Login_signup.html (updated)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <link rel="stylesheet" href="Login_signup2.css">
    <title>Login & Sign Up</title>
</head>
<body>
    <div class="container">
        <div class="form-container sign-in-container">
            <form action="#">
                    <h2>Login</h2>
                <input type="email" required placeholder="Email">
                <input type="password" required placeholder="Password">

                <button>Sign In</button>
```

```
            </form>

        </div>

        <div class="overlay-container">
            <div class="overlay">
                <div class="overlay-panel overlay-left">
                    <h1>Welcome Back!</h1>
                    <p>To keep connected with us, please login with
your personal info</p>
                    <button class="ghost" id="signIn">Sign In</button>
                </div>
                <div class="overlay-panel overlay-right">
                    <h1>Hello, Friend!</h1>
                    <p>Enter your personal details and start your
journey with us</p>
                    <button class="ghost" id="signUp"
style="background-color: black;"> <a href="signup.html"
style="text-decoration: none;-webkit-text-fill-color: white;">Sign
Up</a></button>
                </div>
            </div>
        </div>
    </div>
</body>
</html>
```

CSS :

FILE NAME :Login_signup2.css

```
* {
    box-sizing: border-box;
}

body {
    background: #f6f5f7;
    display: flex;
    align-items: center;
    justify-content: center;
```

```
    height: 100vh;
    margin: 0;
    font-family: 'Roboto', sans-serif;
}

.container {
    background: #fff;
    border-radius: 10px;
    box-shadow: 0 0 10px 0 rgba(0, 0, 0, 0.1);
    overflow: hidden;
    width: 900px;
    max-width: 100%;
    position: relative;
    display: grid;
    grid-template-columns: 1fr 1fr;
}

.form-container {
    position: relative;
    text-align: center;
    max-width: 90%;
    padding: 83px;
    transform: translateY(10%);
}

.sign-in-container {
    transform: translateY(-10%);
}

.form-container form {
    background: #fff;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    height: 100%;
    text-align: center;
    display: grid;
}

h1 {
    margin: 0;
}
```

```css
input {
    background: #f0f0f0;
    border: none;
    padding: 12px 15px;
    margin: 8px 0;
    width: 100%;
    border: 1px solid #ccc;
    border-radius: 5px;
}

button {
    background: #333;
    color: #fff;
    border: none;
    padding: 10px 15px;
    border-radius: 5px;
    cursor: pointer;

}

.overlay-container {
    position: absolute;
    top: 0;
    left: 50%;
    width: 50%;
    height: 100%;
    overflow: hidden;
    transition: transform 0.6s ease-in-out;
    z-index: 100;
}

.overlay {
    background: #333;
    background: linear-gradient(to right, #333 0%, #e1e1e1 100%);
    background-repeat: no-repeat;
    background-size: cover;
    background-position: 0 0;
    color: #fff;
    position: relative;
    left: -100%;
    height: 100%;
    width: 200%;
```

```
    transform: translateX(0);
    transition: transform 0.6s ease-in-out;
}

.overlay-panel {
    position: absolute;
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
    padding: 0 40px;
    text-align: center;
    top: 0;
    height: 100%;
    width: 50%;
    transform: translateX(0);
    transition: transform 0.6s ease-in-out;
}

.overlay-panel.overlay-left {
    transform: translateX(-20%);
}

.overlay-panel.overlay-right {
    right: 0;
    transform: translateX(0);
}

.overlay-panel h1 {
    font-size: 1.2rem;
}

.overlay-panel p {
    font-size: 14px;
    padding: 0 10px;
}

.ghost {
    background: transparent;
    border: 1px solid #000000;
    color: #050000;
}
```

```css
/* Animation */

.container.sign-up-mode .overlay {
    transform: translateX(-100%);
}

.container.sign-up-mode .overlay-panel.overlay-left {
    transform: translateX(0);
}

.container.sign-up-mode .overlay-panel.overlay-right {
    transform: translateX(20%);
}

.container.sign-up-mode .sign-in-container {
    transform: translateY(20%);
}

.container.sign-up-mode .sign-up-container {
    transform: translateY(0);
}

/* Responsive */

@media (max-width: 768px) {
    .container {
        grid-template-columns: 1fr;
        max-width: 100%;
    }

    .overlay-container {
        display: none;
    }

    .form-container {
        max-width: 100%;
    }
}
```

FILE NAME : signup.html(edited)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Sign-up - Music Database</title>
    <link rel="stylesheet" href="signupcss.css">
</head>
<body>
    <div class="container">
        <h1><p style ="font-family:system-ui, -apple-system,
BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell,
'Open Sans', 'Helvetica Neue', sans-serif">Sign-up </p> </h1>
        <form id="signup-form" action = "http://localhost:8000/form"
method = "POST">
            <div class="form-group">
                <label for="new-email">E-Mail:</label>
                <input type="email" id="new-email" name="email"
required placeholder = "xyz@mail.com">
            </div>
            <div class="form-group">
                <label for="new-password">Password:</label>
                <input type="password" id="new-password"
name="password" required >
            </div>
            <div class="form-group">
                <label for="new-username">Username:</label>
                <input type="text" id="new-username" name="username"
required placeholder = "ABC_10">
            </div>
            <button type="submit" id = "new-submit"
name="submit">Submit</button>
        </form>
        <div class="switch-form">
           <p style ="font-family:system-ui, -apple-system,
BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell,
'Open Sans', 'Helvetica Neue', sans-serif" >
            Already have an account?
            <!-- <button class="login-button" ><a
href="Login_signup.html" >Login</a></button>  -->
```

```
            <a href="Login_signup.html" class="login-button">Login</a>
        </p>
        </div>
    </div>
    <script src = "server.js"> </script>
</body>
</html>
```

FILE NAME signupcss.css

```
body {
    font-family: Arial, sans-serif;
    background-color: #f2f2f2;
    margin: 68px;
    padding: 47px;
}



.container {
    max-width: 342px;
    margin: 0 auto;
    padding: 130px;
    background-color: #fff;
    border-radius: 6px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 1);
}
```

```css
.login-button {
      background-color: #007bff;
      color: #fff;
      border: none;
      border-radius: 4px;
      padding: 8px 16px;
      cursor: pointer;
    }







h1 {
    text-align: center;
    margin-bottom: 20px;
}




.form-group {
    margin-bottom: 20px;
}




label {
    display: block;
    font-weight: bold;
}




input[type="email"],
input[type="password"],
input[type="text"],
input[type="phone"] {
```

```
    font-family: system-ui, -apple-system, BlinkMacSystemFont, 'Segoe
UI', Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue',
sans-serif;
    width: 100%;
    padding: 10px;
    border: 1px solid #ccc;
    border-radius: 5px;
}



button {
    width: 100%;
    padding: 10px;
    background-color: #333333;
    color: #fff;
    border: none;
    border-radius: 5px;
    cursor: pointer;
}



button:hover {
    background-color: #253657;
}



.switch-form {
    text-align: center;
}


.login-button {
    display: inline-block;
    padding: 10px 20px;
    background-color: #3498db;
    color: #ffffff;
    text-decoration: none;
```

```
    border: none;
    border-radius: 5px;
    cursor: pointer;
}

.switch-form a {
    text-decoration: none;
    color: #101011;
}
```

Backend For signup page :

FILE name : server.js

const express = require ('express');

```
const app = express();
const port = 8000;
const bodyParser = require('body-parser');
app.use ( express.static(__dirname));

app.get("/form", (req,res)  => {
    res.sendFile( __dirname + "/signup.html");
});


app.use(bodyParser.urlencoded({extended:false}))

app.get('/submit' , function(req, res) {
    console.log("Data Saved!");
})

const {Pool , Client} = require('pg');

const connectString =
'postgressql://music_db:password@localhost:5432/music-db'

const client = new Client ( {
    connectionString:connectString
})

app.post('/form' , (req, res) => {
```

```
    const { email, password, username} = req.body
    client.connect()
    client.query( 'INSERT INTO users (email, password, username) VALUES
($1, $2, $3);', [ email, password, username] , ( err,res ) => {

        console.log(err,res);
        client.end()
    })
    res.sendFile(__dirname + "/index.html");
})

app.listen(port, () => {
    console.log(`App listening on port ${port}!`)
})
```

File_Name:playlist.html

```
// Sample playlist data
let playlists = [];

// Index of the playlist being edited
let editingIndex = -1;

// Function to open the create playlist modal
function openCreatePlaylistModal() {
    document.getElementById("create-playlist-modal").style.display =
"block";
}

// Function to close the create playlist modal
function closeCreatePlaylistModal() {
    document.getElementById("create-playlist-modal").style.display =
"none";
}

// Function to open the edit playlist modal
function openEditPlaylistModal(index) {
    editingIndex = index;
    document.getElementById("edit-playlist-modal").style.display =
"block";
}

// Function to close the edit playlist modal
function closeEditPlaylistModal() {
    editingIndex = -1;
    document.getElementById("edit-playlist-modal").style.display =
"none";
}

// Function to open the add song modal
function openAddSongModal(index) {
    editingIndex = index;
    document.getElementById("add-song-modal").style.display = "block";
}

// Function to close the add song modal
function closeAddSongModal() {
    editingIndex = -1;
    document.getElementById("add-song-modal").style.display = "none";
```

```
}

// Function to add a new playlist
function addPlaylist() {
    const playlistName =
document.getElementById("playlist-name").value;

    // Validate playlist name
    if (playlistName.trim() === "") {
        alert("Please enter a valid playlist name.");
        return;
    }

    // Create a new playlist object
    const newPlaylist = {
        name: playlistName,
        songs: [] // You can add more details about the songs if needed
    };

    // Add the new playlist to the playlists array
    playlists.push(newPlaylist);

    // Clear input field
    document.getElementById("playlist-name").value = "";

    // Close the modal
    closeCreatePlaylistModal();

    // Update the playlist display
    displayPlaylists();
}

// Function to edit a playlist
function editPlaylist() {
    const newPlaylistName =
document.getElementById("edit-playlist-name").value;

    // Validate playlist name
    if (newPlaylistName.trim() === "") {
        alert("Please enter a valid playlist name.");
        return;
    }
```

```
    // Update the playlist name
    playlists[editingIndex].name = newPlaylistName;

    // Clear input field
    document.getElementById("edit-playlist-name").value = "";

    // Close the modal
    closeEditPlaylistModal();

    // Update the playlist display
    displayPlaylists();
}

// Function to delete a playlist
function deletePlaylist(index) {
    const confirmDelete = confirm("Are you sure you want to delete this
playlist?");

    if (confirmDelete) {
        playlists.splice(index, 1);
        displayPlaylists();
    }
}

// Function to search and add a song to a playlist
function searchAndAddSong() {
    const searchQuery = document.getElementById("search-song").value;

    // Simulate a search operation (replace with actual API call in a
real app)
    // For simplicity, assume we find a song with the given name
    const foundSong = { name: searchQuery, artist: "Unknown" };

    // Add the found song to the playlist
    playlists[editingIndex].songs.push(foundSong);

    // Clear input field
    document.getElementById("search-song").value = "";

    // Close the modal
    closeAddSongModal();

    // Update the playlist display
```

```
    displayPlaylists();
}

function ViewingSong(index)
{

}

// Function to display playlists on the home page
function displayPlaylists() {
    const playlistContainer =
document.getElementById("playlist-container");
    playlistContainer.innerHTML = ""; // Clear previous content

    // Display each playlist
    playlists.forEach((playlist, index) => {
        const playlistDiv = document.createElement("div");
        playlistDiv.className = "playlist-item";
        playlistDiv.textContent = playlist.name;

        // Add buttons for edit, delete, and add song
        const editBtn = document.createElement("button");
        editBtn.textContent = "Edit";
        editBtn.addEventListener("click", () =>
openEditPlaylistModal(index));

        const deleteBtn = document.createElement("button");
        deleteBtn.textContent = "Delete";
        deleteBtn.addEventListener("click", () =>
deletePlaylist(index));

        const addSongBtn = document.createElement("button");
        addSongBtn.textContent = "Add Song";
        addSongBtn.addEventListener("click", () =>
openAddSongModal(index));

        const viewsongBtn = document.createElement("button");
        viewsongBtn.textContent = "View Song";
        viewsongBtn.addEventListener("click",() => ViewingSong(index));

        // Append buttons to the playlist item
        playlistDiv.appendChild(editBtn);
        playlistDiv.appendChild(deleteBtn);
```

```
        playlistDiv.appendChild(addSongBtn);

        // Add playlist item to the container
        playlistContainer.appendChild(playlistDiv);
    });
}

// Initial display of playlists
displayPlaylists();
```

PLAYLIST MANAGEMENT

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Playlist Management</title>
    <style>
            body {
                font-family: Arial, sans-serif;
                background-color: #f4f4f4;
                margin: 0;
                padding: 0;
            }

            input[type="text"] {
                padding: 8px;
                margin: 5px;
            }

            button {
                padding: 10px;
                margin: 5px;
                cursor: pointer;
                background-color: #4CAF50;
                color: white;
                border: none;
                border-radius: 5px;
            }
```

```css
button:hover {
    background-color: #45a049;
}

#playlists {
    display: flex;
    flex-wrap: wrap;
    gap: 20px;
    margin-top: 20px;
}

.playlist-container {
    background-color: #fff;
    padding: 10px;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

.playlist-container strong {
    font-size: 18px;
}

ul {
    list-style-type: none;
    padding: 0;
}

li {
    margin-bottom: 5px;
}

.modal {
    display: none;
    position: fixed;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    background-color: #fefefe;
    padding: 20px;
    border: 1px solid #888;
    width: 300px;
}
```

```
            .modal-content {
                position: relative;
            }

            .close {
                color: #aaa;
                float: right;
                font-size: 28px;
                font-weight: bold;
                cursor: pointer;
            }

            .close:hover {
                color: black;
            }
        </style>

</head>
<body>
<h1 style="color:black">Your PlayList Management</h1>
<!-- Playlist Name Input -->
<input type="text" id="playlist-name" placeholder="Enter Playlist
Name">
<button onclick="addPlaylist()">Add Playlist</button>

<!-- Song Name Input -->
<input type="text" id="song-name" placeholder="Enter Song Name">
<!-- Add a dropdown to select the playlist -->
<select id="playlist-dropdown">
    <!-- Playlist options will be dynamically added here -->
</select>
<button onclick="addSong()">Add Song</button>

<!-- Display Playlists and Songs -->
<div id="playlists"></div>

<!-- Add Song Modal -->
<div id="add-song-modal" class="modal">
    <div class="modal-content">
        <span class="close"
onclick="closeAddSongModal()">&times;</span>
        <h2>Add Song to Playlist</h2>
```

```
        <!-- Update the label and add a dropdown to select the playlist
-->
        <label for="song-name-modal">Song Name:</label>
        <input type="text" id="song-name-modal" placeholder="Enter Song
Name">
        <label for="playlist-dropdown-modal">Select Playlist:</label>
        <select id="playlist-dropdown-modal">
            <!-- Playlist options will be dynamically added here -->
        </select>
        <button onclick="confirmAddSong()">Add Song</button>
    </div>
</div>

<!-- View Song Modal -->
<div id="view-song-modal" class="modal">
    <div class="modal-content">
        <span class="close"
onclick="closeViewSongModal()">&times;</span>
        <h2>View Song of Playlist</h2>
        <label for="view-song">Songs:</label>
        <ul id="view-song-list"></ul>
    </div>
</div>

<script>
const playlists = [];

function addPlaylist() {
    const playlistName =
document.getElementById("playlist-name").value;

    // Validate playlist name
    if (playlistName.trim() === "") {
        alert("Please enter a valid playlist name.");
        return;
    }

    // Create a new playlist object
    const newPlaylist = {
        name: playlistName,
        songs: [] // You can add more details about the songs if needed
    };
```

```
    // Add the new playlist to the playlists array
    playlists.push(newPlaylist);

    // Clear input field
    document.getElementById("playlist-name").value = "";

    // Update the playlist display
    updatePlaylistDropdowns();
    displayPlaylists();
}

function addSong() {
    const songName = document.getElementById("song-name").value;

    // Validate song name
    if (songName.trim() === "") {
        alert("Please enter a valid song name.");
        return;
    }

    // Find the currently selected playlist (you can enhance this logic
based on your UI)
    const selectedPlaylistIndex =
document.getElementById("playlist-dropdown").selectedIndex;

    // Add the new song to the selected playlist
    playlists[selectedPlaylistIndex].songs.push(songName);

    // Clear input field
    document.getElementById("song-name").value = "";

    // Update the playlist display
    displayPlaylists();
}

function displayPlaylists() {
    const playlistsContainer = document.getElementById("playlists");

    // Clear existing content
    playlistsContainer.innerHTML = "";

    // Display each playlist
    playlists.forEach((playlist, index) => {
```

```
        const playlistDiv = document.createElement("div");
        playlistDiv.innerHTML = `<strong>${playlist.name}</strong>`;

        // Display each song in the playlist
        if (playlist.songs.length > 0) {
            const songsList = document.createElement("ul");
            playlist.songs.forEach((song, songIndex) => {
                const songItem = document.createElement("li");
                songItem.textContent = song;
                songsList.appendChild(songItem);

                // Add a delete button for each song
                const deleteSongButton =
document.createElement("button");
                deleteSongButton.textContent = "Delete Song";
                deleteSongButton.onclick = () => deleteSong(index,
songIndex);
                songsList.appendChild(deleteSongButton);
            });
            playlistDiv.appendChild(songsList);
        }

        // Add a "View Songs" button for each playlist
        const viewSongsButton = document.createElement("button");
        viewSongsButton.textContent = "View Songs";
        viewSongsButton.onclick = () =>
openViewSongModal(playlist.songs);
        playlistDiv.appendChild(viewSongsButton);

        // Add an "Add Song" button for each playlist
        const addSongButton = document.createElement("button");
        addSongButton.textContent = "Add Song";
        addSongButton.onclick = () => openAddSongModal(index);
        playlistDiv.appendChild(addSongButton);

        // Add a "Delete Playlist" button
        const deletePlaylistButton = document.createElement("button");
        deletePlaylistButton.textContent = "Delete Playlist";
        deletePlaylistButton.onclick = () => deletePlaylist(index);
        playlistDiv.appendChild(deletePlaylistButton);

        playlistsContainer.appendChild(playlistDiv);
    });
```

```
    // Update the playlist dropdowns in the modals
    updatePlaylistDropdowns();
}

function updatePlaylistDropdowns() {
    const playlistDropdown =
document.getElementById("playlist-dropdown");
    const playlistDropdownModal =
document.getElementById("playlist-dropdown-modal");

    // Clear existing content in the dropdowns
    playlistDropdown.innerHTML = "";
    playlistDropdownModal.innerHTML = "";

    // Populate the dropdowns with playlist options
    playlists.forEach((playlist, index) => {
        const option = document.createElement("option");
        option.value = index;
        option.textContent = playlist.name;

        playlistDropdown.appendChild(option.cloneNode(true));
        playlistDropdownModal.appendChild(option);
    });
}

function openAddSongModal(selectedPlaylistIndex) {
    const addSongModal = document.getElementById("add-song-modal");

    // Set the selected playlist in the modal dropdown
    const playlistDropdownModal =
document.getElementById("playlist-dropdown-modal");
    playlistDropdownModal.value = selectedPlaylistIndex;

    // Show the modal
    addSongModal.style.display = "block";
}

function closeAddSongModal() {
    const addSongModal = document.getElementById("add-song-modal");

    // Hide the modal
    addSongModal.style.display = "none";
```

```
}

function confirmAddSong() {
    const songName = document.getElementById("song-name-modal").value;

    // Validate song name
    if (songName.trim() === "") {
        alert("Please enter a valid song name.");
        return;
    }

    // Get the selected playlist index from the modal dropdown
    const selectedPlaylistIndex =
document.getElementById("playlist-dropdown-modal").value;

    // Add the new song to the selected playlist
    playlists[selectedPlaylistIndex].songs.push(songName);

    // Clear input field
    document.getElementById("song-name-modal").value = "";

    // Close the modal
    closeAddSongModal();

    // Update the playlist display
    displayPlaylists();
}

function openViewSongModal(songs) {
    const viewSongModal = document.getElementById("view-song-modal");
    const viewSongList = document.getElementById("view-song-list");

    // Clear existing content
    viewSongList.innerHTML = "";

    // Display each song in the modal
    songs.forEach(song => {
        const songItem = document.createElement("li");
        songItem.textContent = song;
        viewSongList.appendChild(songItem);
    });

    // Show the modal
```

```
    viewSongModal.style.display = "block";
}

function closeViewSongModal() {
    const viewSongModal = document.getElementById("view-song-modal");

    // Hide the modal
    viewSongModal.style.display = "none";
}

function deletePlaylist(index) {
    // Confirm deletion
    const confirmDelete = confirm("Are you sure you want to delete this
playlist?");

    if (confirmDelete) {
        // Remove the playlist at the specified index
        playlists.splice(index, 1);

        // Update the playlist display
        displayPlaylists();
    }
}

function deleteSong(playlistIndex, songIndex) {
    // Confirm deletion
    const confirmDelete = confirm("Are you sure you want to delete this
song?");

    if (confirmDelete) {
        // Remove the song at the specified index in the specified
playlist
        playlists[playlistIndex].songs.splice(songIndex, 1);

        // Update the playlist display
        displayPlaylists();
    }
}
</script>

</body>
</html>
```