

**PROJECT REPORT**

**Rhythmic Tunes-Your Melodic Companion**

|  |  |
| --- | --- |
| YEAR | : **2024 – 2025** |
| COLLEGE NAME | : **K.C.S KASI NADAR COLLEGE OF ARTS & SCIENCE** |
| CODE | : UNM203 |
| DEPARTMENT | : **COMPUTER SCIENCE** |
| PROGRAM | : **B.Sc Computer Science** |
| SEMESTER | : **VI** |

PROJECT SUBMITTED TO: UNIVERSITY OF MADRAS / NAAN MUDALVAN

Course Name : **Front End Development and Database Administration**

**TEAM LEADER:**

HARIHARAN R

**MEMBERS:**

1. BALAJI M

2.YOGESHWARAN E

3. KARNESH V

**GUIDED BY: Dr. S.KALAISELVI**

**SPOC NAME: Dr.K. LALITHAKAMESWARI**

**Source Code:**

**Index.html:**

<!doctype html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<link rel="icon" type="image/svg+xml" href="/vite.svg" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.16/dist/tailwind.min.css" rel="stylesheet">

<title>Music-Player</title>

</head>

<body>

<div id="root"></div>

<script type="module" src="/src/main.jsx"></script>

</body>

</html>

**App.css:** \*{ padding: 0; margin: 0;

}

Body { background-image: url("assets/background.jpg"); background-size: cover;

}

**Index.css:** :root { font-family: Inter, system-ui, Avenir, Helvetica, Arial, sans-serif; line-height: 1.5; font-weight: 400;

color-scheme: light dark; color: rgba(255, 255, 255, 0.87); background-color: #242424;

font-synthesis: none; text-rendering: optimizeLegibility;

-webkit-font-smoothing: antialiased;

-moz-osx-font-smoothing: grayscale;

}

a { font-weight: 500; color: #646cff; text-decoration: inherit;

} a:hover { color: #535bf2;

}

body { margin: 0; display: flex; place-items: center; min-width: 320px; min-height: 100vh;

}

h1 { font-size: 3.2em; line-height: 1.1;

}

button { border-radius: 8px; border: 1px solid transparent; padding: 0.6em 1.2em; font-size: 1em; font-weight: 500; font-family: inherit; background-color: #1a1a1a; cursor: pointer; transition: border-color 0.25s;

} button:hover { border-color: #646cff; } button:focus, button:focus-visible { outline: 4px auto -webkit-focus-ring-color;

}

@media (prefers-color-scheme: light) {

:root { color: #213547; background-color: #ffffff;

} a:hover { color: #747bff; } button { background-color: #f9f9f9;

}

}

**Sidebar.css:**

.sidebar {

color: white; width: 200px; height: 100%; position: fixed; padding-top: 20px;

}

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

}

.sidebar li { margin-bottom: 10px;

}

.sidebar a { text-decoration: none; color: white; font-size: 18px; display: flex; align-items: center;

}

.content { margin-left: 200px; padding: 20px;

}

.vl { border-left: 3px solid rgb(74, 73, 73); height: 700px; position: absolute; left: 106%;

top: 0;

}

.sidebar { color: white; width: 200px; height: 100%; position: fixed; padding-top: 20px;

}

.sidebar ul { padding: 0; list-style: none; }

.sidebar li { margin-bottom: 10px;

}

.sidebar a { text-decoration: none; color: white; font-size: 18px; display: flex; align-items: center;

}

.content { margin-left: 200px; padding: 20px;

}

.vl { border-left: 3px solid rgb(74, 73, 73); height: 900px; position: absolute; left: 106%; top: 0; }

@media (max-width: 768px) {

.sidebar { width: 100%; height: auto; position: relative; padding-top: 20px;

}

.content { margin-left: 0;

}

.vl { display: none;

}

}

**eslint.config.js:**

import js from '@eslint/js'

import globals from 'globals' import reactHooks from 'eslint-plugin-react-hooks' import reactRefresh from 'eslint-plugin-react-refresh'

export default [

{ ignores: ['dist'] },

{

files: ['\*\*/\*.{js,jsx}'], languageOptions: { ecmaVersion: 2020, globals: globals.browser, parserOptions: { ecmaVersion: 'latest', ecmaFeatures: { jsx: true }, sourceType: 'module',

},

}, plugins: {

'react-hooks': reactHooks,

'react-refresh': reactRefresh,

}, rules: {

...js.configs.recommended.rules,

...reactHooks.configs.recommended.rules,

'no-unused-vars': ['error', { varsIgnorePattern: '^[A-Z\_]' }],

'react-refresh/only-export-components': [

'warn',

{ allowConstantExport: true },

],

},

},

]

**Main.jsx:**

import React from 'react' import ReactDOM from 'react-dom/client' import App from './App.jsx'

// import './index.css'

ReactDOM.createRoot(document.getElementById('root')).render(

<React.StrictMode>

<App />

</React.StrictMode>,

)

**App.jsx:**

import 'bootstrap/dist/css/bootstrap.min.css'; import './App.css' import { BrowserRouter,Routes,Route } from 'react-router-dom' import Songs from './Components/Songs' import Sidebar from './Components/Sidebar' import Favorities from './Components/Favorities' import Playlist from './Components/Playlist'

function App() { return (

<div >

<BrowserRouter>

<div>

<Sidebar/>

</div>

<div>

<Routes>

<Route path='/' element={<Songs/>} />

<Route path='/songs' element={<Songs/>} />

<Route path='/favorities' element={<Favorities/>} />

<Route path='/playlist' element={<Playlist/>} />

</Routes>

</div>

</BrowserRouter>

</div>

)

} export default App **Favorities.jsx:** import { useState, useEffect } from 'react'; import axios from 'axios'; import { Button, Table } from 'react-bootstrap'; import { FaHeart,} from 'react-icons/fa';

function Favorities() { const [playlist, setPlaylist] = useState([]);

const [currentlyPlaying, setCurrentlyPlaying] = useState(null);

useEffect(() => {

axios

.get(`http://localhost:3000/favorities`)

.then((response) => {

const playlistData = response.data; setPlaylist(playlistData);

})

.catch((error) => {

console.error('Error fetching playlist items: ', error);

});

const handleAudioPlay = (itemId, audioElement) => { if (currentlyPlaying && currentlyPlaying !== audioElement) { currentlyPlaying.pause(); // Pause the currently playing audio

}

setCurrentlyPlaying(audioElement); // Update the currently playing audio

};

const handlePlay = (itemId, audioElement) => { audioElement.addEventListener('play', () => { handleAudioPlay(itemId, audioElement);

});

};

playlist.forEach((item) => { const audioElement = document.getElementById(`audio-${item.id}`); if (audioElement) { handlePlay(item.id, audioElement);

}

}); return () => {

playlist.forEach((item) => { const audioElement = document.getElementById(`audio-${item.id}`); if (audioElement) {

audioElement.removeEventListener('play', () => handleAudioPlay(item.id, audioElement));

}

});

};

}, [playlist, currentlyPlaying]);

const removeFromFavorites = async (itemId) => { try {

const selectedItem = playlist.find((item) => item.itemId === itemId); if (!selectedItem) {

throw new Error('Selected item not found in wishlist');

}

await axios.delete(`http://localhost:3000/favorities/${selectedItem.id}`); const response = await axios.get('http://localhost:3000/favorities'); setPlaylist(response.data);

} catch (error) { console.error('Error removing item from wishlist: ', error);

} }; return ( <div >

<div style={{marginLeft:"240px"}}>

<div className="container mx-auto p-8">

<h2 className="text-3xl font-semibold mb-4 text-center">Favorites</h2>

<Table responsive style={{width:"1150px"}} >

<thead>

<tr>

<th>#</th>

<th>Title</th>

<th>Genre</th>

<th></th>

<th>Actions</th>

</tr>

</thead>

<tbody>

{playlist.map((item, index) => (

<tr key={item.\_id}>

<td>{index + 1}</td>

<td>

<div style={{ display: 'flex' }}>

<img

src={item.imgUrl} alt="Item Image" className="rounded"

style={{ height: '50px', width: '50px' }}

/>

<div style={{ paddingLeft: '20px' }}>

<strong> {item.title}</strong>

<p><td>{item.singer}</td></p>

</div>

</div>

</td>

<td>{item.genre}</td>

<td>

<Button

style={{ backgroundColor: 'white', border: 'none' }} onClick={() => removeFromFavorites(item.itemId)}

>

<FaHeart color="red" />

</Button>

</td>

<td>

<audio controls id={`audio-${item.id}`} style={{ width: '250px' }}>

<source src={item.songUrl} />

</audio>

</td>

</tr>

))}

</tbody>

</Table>

</div>

</div>

</div>

);

}

export default Favorities; **Playlist.jsx:**

import{ useState, useEffect } from 'react'; import axios from 'axios'; import { Button, Table } from 'react-bootstrap'; import {FaPauseCircle, FaPlayCircle } from 'react-icons/fa';

function Playlist() {

const [playlist, setPlaylist] = useState([]);

const [currentlyPlaying, setCurrentlyPlaying] = useState(null); const [isPlaying, setIsPlaying] = useState(false);

useEffect(() => { axios

.get('http://localhost:3000/playlist')

.then((response) => { const playlistData = response.data; setPlaylist(playlistData);

})

.catch((error) => {

console.error('Error fetching playlist items: ', error);

});

const handleAudioPlay = (itemId, audioElement) => { if (currentlyPlaying && currentlyPlaying !== audioElement) { currentlyPlaying.pause();

}

setCurrentlyPlaying(audioElement);

};

const handlePlay = (itemId, audioElement) => { audioElement.addEventListener('play', () => { handleAudioPlay(itemId, audioElement);

});

};

playlist.forEach((item) => { const audioElement = document.getElementById(`audio-${item.id}`); if (audioElement) {

handlePlay(item.id, audioElement);

}

});

return () => {

playlist.forEach((item) => { const audioElement = document.getElementById(`audio-${item.id}`); if (audioElement) {

audioElement.removeEventListener('play', () => handleAudioPlay(item.id, audioElement));

}

});

};

}, [playlist, currentlyPlaying]);

const removeFromPlaylist = async (itemId) => { try {

const selectedItem = playlist.find((item) => item.itemId === itemId);

if (!selectedItem) {

throw new Error('Selected item not found in playlist');

}

await axios.delete(`http://localhost:3000/playlist/${selectedItem.id}`);

const response = await axios.get('http://localhost:3000/playlist'); setPlaylist(response.data);

} catch (error) { console.error('Error removing item from playlist: ', error);

}

};

const playAllSongs = () => { const currentSongAudio = document.getElementById(`audio-${playlist[0].id}`);

if (currentSongAudio.paused) {

currentSongAudio.play();

currentSongAudio.addEventListener('ended', () => {

const nextIndex = (playlist.indexOf(playlist[0]) + 1) % playlist.length; const nextSongAudio = document.getElementById(`audio-${playlist[nextIndex].id}`);

if (nextSongAudio) {

nextSongAudio.play();

}

}); setIsPlaying(true);

} else {

currentSongAudio.pause(); setIsPlaying(false);

}

};

return (

<div>

<div style={{ marginLeft: '230px' }}>

<div className="container mx-auto p-8">

<h2 className="text-3xl font-semibold mb-4 text-center">Playlist</h2>

<Table responsive style={{ width: '1150px' }}>

<thead>

<tr>

<th>#</th>

<th>Title</th>

<th>Genre</th>

<th>Actions</th>

<th>

<Button

style={{ backgroundColor: 'blue', border: 'none', width: '50px', display: 'flex', justifyContent: 'center' }} onClick={playAllSongs}

>

{isPlaying ? <FaPauseCircle style={{ width: '40px', height: '25px', display: 'flex', justifyContent: 'center' }} /> : <FaPlayCircle style={{ width: '40px', height: '25px', display: 'flex', justifyContent: 'center' }} />}

</Button>

</th>

</tr>

</thead>

<tbody>

{playlist.map((item, index) => (

<tr key={item.id}>

<td>{index + 1}</td>

<td>

<div style={{ display: 'flex' }}>

<img

src={item.imgUrl} alt="Item Image" className="rounded"

style={{ height: '50px', width: '50px' }}

/>

<div style={{ paddingLeft: '20px' }}>

<strong>{item.title}</strong>

<p>

<td>{item.singer}</td>

</p>

</div>

</div>

</td>

<td>{item.genre}</td>

<td>

<audio controls id={`audio-${item.id}`} style={{ width: '280px' }}>

<source src={item.songUrl} />

</audio>

</td>

<td>

<Button

style={{ backgroundColor: 'red', border: 'none' }} onClick={() => removeFromPlaylist(item.itemId)}

>

Remove

</Button>

</td>

</tr>

))}

</tbody>

</Table>

</div>

</div>

</div>

);

}

export default Playlist;

**Search.jsx:**

import{ useState } from 'react'; import { Form, InputGroup } from 'react-bootstrap'; import { FaSearch } from 'react-icons/fa';

function Search() {

const [searchTerm, setSearchTerm] = useState('');

const [searchResults] = useState([]);

return (

<div>

<h2 className="text-3xl font-semibold mb-4 text-center">Search</h2>

<InputGroup className="mb-3">

<InputGroup.Text id="search-icon">

<FaSearch />

</InputGroup.Text> <Form.Control type="search"

placeholder="Search by singer, genre, or song name" value={searchTerm}

onChange={(e) => setSearchTerm(e.target.value)}

style={{ outline: 'none', boxShadow: 'none', border: '1px solid #ced4da', borderRadius:

'0.25rem' }} className="search-input"

/>

</InputGroup>

<br />

<div style={{ width: '960px', display: 'grid', textalign: 'left',marginleft: 0,gridTemplateColumns: 'auto auto auto auto', gap: '30px' }}>

{searchResults.map((result) => (

<div key={result.\_id} className="bg-white p-4 rounded shadow">

{ }

</div>

))}

</div>

</div>

); }

export default Search;

**Sidebar.jsx:**

import { Link } from 'react-router-dom'; import { FaHome, FaHeart, FaList,} from 'react-icons/fa'; import './sidebar.css'

const Sidebar = () => {

return (

<nav className="sidebar">

<ul className="list-unstyled">

<strong style={{display:"flex",justifyContent:"center",fontSize:"30px"}}>MusicPlayer</strong>

<div style={{marginTop:"35px"}}>

<li>

<Link to="/songs">

<p style={{paddingLeft:"10px"}}> <FaHome /> </p> <p style={{paddingLeft:"10px"}}>Home</p>

</Link>

</li>

</div>

<hr/>

<div>

<p style={{display:"flex",justifyContent:"center"}}> Your Library</p>

<li>

<Link to="/favorities">

<p style={{paddingLeft:"10px"}}> <FaHeart /> </p> <p style={{paddingLeft:"10px"}}>Favorites</p>

</Link>

</li>

<li>

<Link to="/playlist">

<p style={{paddingLeft:"10px"}}> <FaList /> </p> <p style={{paddingLeft:"10px"}}>PlayList</p>

</Link>

</li>

<div class="vl">

</div>

</div>

</ul>

{}

</nav>

);

};

export default Sidebar; **Song.jsx:**

import { useState, useEffect } from 'react'; import { Button, Form, InputGroup } from 'react-bootstrap'; import { FaHeart, FaRegHeart, FaSearch } from 'react-icons/fa'; import axios from 'axios'; import './sidebar.css'

function Songs() {

const [items, setItems] = useState([]); const [wishlist, setWishlist] = useState([]); const [playlist, setPlaylist] = useState([]); const [currentlyPlaying, setCurrentlyPlaying] = useState(null); const [searchTerm, setSearchTerm] = useState('');

useEffect(() => {

axios.get('http://localhost:3000/items')

.then(response => setItems(response.data))

.catch(error => console.error('Error fetching items: ', error));

axios.get('http://localhost:3000/favorities')

.then(response => setWishlist(response.data))

.catch(error => {

console.error('Error fetching Favvorities:', error);

setWishlist([]);

});

axios.get('http://localhost:3000/playlist')

.then(response => setPlaylist(response.data))

.catch(error => {

console.error('Error fetching playlist: ', error);

setPlaylist([]);

});

const handleAudioPlay = (itemId, audioElement) => {

if (currentlyPlaying && currentlyPlaying !== audioElement) { currentlyPlaying.pause();

}

setCurrentlyPlaying(audioElement);

};

const handlePlay = (itemId, audioElement) => { audioElement.addEventListener('play', () => { handleAudioPlay(itemId, audioElement);

});

};

items.forEach((item) => { const audioElement = document.getElementById(`audio-${item.id}`); if (audioElement) {

handlePlay(item.id, audioElement);

}

});

return () => { items.forEach((item) => { const audioElement = document.getElementById(`audio-${item.id}`); if (audioElement) {

audioElement.removeEventListener('play', () => handleAudioPlay(item.id,

audioElement));

}

});

};

}, [items,currentlyPlaying, searchTerm]);

const addToWishlist = async (itemId) => { try {

const selectedItem = items.find((item) => item.id === itemId); if (!selectedItem) {

throw new Error('Selected item not found');

}

const { title, imgUrl, genre, songUrl, singer, id: itemId2 } = selectedItem;

await axios.post('http://localhost:3000/favorities', { itemId: itemId2, title, imgUrl, genre, songUrl, singer });

const response = await axios.get('http://localhost:3000/favorities'); setWishlist(response.data);

} catch (error) { console.error('Error adding item to wishlist: ', error);

}

};

const removeFromWishlist = async (itemId) => { try {

const selectedItem = wishlist.find((item) => item.itemId === itemId); if (!selectedItem) {

throw new Error('Selected item not found in wishlist');

}

await axios.delete(`http://localhost:3000/favorities/${selectedItem.id}`);

const response = await axios.get('http://localhost:3000/favorities'); setWishlist(response.data);

} catch (error) { console.error('Error removing item from wishlist: ', error);

}

};

const isItemInWishlist = (itemId) => {

return wishlist.some((item) => item.itemId === itemId);

};

const addToPlaylist = async (itemId) => { try {

const selectedItem = items.find((item) => item.id === itemId); if (!selectedItem) {

throw new Error('Selected item not found');

}

const { title, imgUrl, genre, songUrl, singer, id: itemId2 } = selectedItem;

await axios.post('http://localhost:3000/playlist', { itemId: itemId2, title, imgUrl, genre, songUrl, singer });

const response = await axios.get('http://localhost:3000/playlist'); setPlaylist(response.data);

} catch (error) {

console.error('Error adding item to wishlist: ', error); }

};

const removeFromPlaylist = async (itemId) => { try {

const selectedItem = playlist.find((item) => item.itemId === itemId); if (!selectedItem) {

throw new Error('Selected item not found in wishlist');

}

await axios.delete(`http://localhost:3000/playlist/${selectedItem.id}`);

const response = await axios.get('http://localhost:3000/playlist'); setPlaylist(response.data);

} catch (error) { console.error('Error removing item from wishlist: ', error);

}

};

const isItemInPlaylist = (itemId) => {

return playlist.some((item) => item.itemId === itemId);

};

const filteredItems = items.filter((item) => {

const lowerCaseQuery = searchTerm.toLowerCase(); return (

item.title.toLowerCase().includes(lowerCaseQuery) || item.singer.toLowerCase().includes(lowerCaseQuery) || item.genre.toLowerCase().includes(lowerCaseQuery)

);

});

return (

<div style={{display:"flex", justifyContent:"flex-end"}}>

<div className="songs-container" style={{width:"1300px"}}>

<div className="container mx-auto p-3">

<h2 className="text-3xl font-semibold mb-4 text-center">Songs List</h2>

<InputGroup className="mb-3">

<InputGroup.Text id="search-icon">

<FaSearch />

</InputGroup.Text> <Form.Control type="search"

placeholder="Search by singer, genre, or song name" value={searchTerm}

onChange={(e) => setSearchTerm(e.target.value)} className="search-input"

/>

</InputGroup>

<br />

<div className="row row-cols-1 row-cols-md-2 row-cols-lg-3 row-cols-xl-4 g-4">

{filteredItems.map((item) => (

<div key={item.id} className="col">

<div className="card h-100">

<img src={item.imgUrl} alt="Item Image" className="card-img-top rounded-top" style={{ height: '200px', width: '100%' }}

/>

<div className="card-body">

<div className="d-flex justify-content-between align-items-left mb-2">

<h5 className="card-title">{item.title}</h5>

{isItemInWishlist(item.id) ? (

<Button variant="light"

onClick={() => removeFromWishlist(item.id)}

>

<FaHeart color="red" />

</Button>

) : ( <Button variant="light"

onClick={() => addToWishlist(item.id)}

>

<FaRegHeart color="black" />

</Button>

)}

</div>

<p className="card-text">Genre: {item.genre}</p>

<p className="card-text">Singer: {item.singer}</p>

<audio controls className="w-100" id={`audio-${item.id}`} >

<source src={item.songUrl} />

</audio>

</div>

<div className="card-footer d-flex justify-content-center">

{isItemInPlaylist(item.id) ? (

<Button variant="outline-secondary" onClick={() => removeFromPlaylist(item.id)}

>

Remove From Playlist

</Button>

) : (

<Button variant="outline-primary" onClick={() => addToPlaylist(item.id)}

>

Add to Playlist

</Button>

)}

</div>

</div>

</div>

))}

</div>

</div>

</div>

</div>

);

}

export default Songs; **Vite.config.js:**

import { defineConfig } from 'vite'

import react from '@vitejs/plugin-react' export default defineConfig({ plugins: [react()],

})

**Package.json:**

{

"name": "music-player-frontend-",

"private": true,

"version": "0.0.0",

"type": "module",

"scripts": {

"dev": "vite",

"build": "vite build",

"lint": "eslint . --ext js,jsx --report-unused-disable-directives --max-warnings 0", "preview": "vite preview"

},

"dependencies": {

"axios": "^1.6.2",

"bootstrap": "^5.3.2",

"json-server": "^0.17.4",

"react": "^18.2.0",

"react-bootstrap": "^2.9.1",

"react-dom": "^18.2.0",

"react-icons": "^4.12.0",

"react-router-dom": "^6.20.1",

"tailwindcss": "^3.3.6"

},

"devDependencies": {

"@types/react": "^18.2.43",

"@types/react-dom": "^18.2.17",

"@vitejs/plugin-react": "^4.2.1",

"eslint": "^8.55.0",

"eslint-plugin-react": "^7.33.2",

"eslint-plugin-react-hooks": "^4.6.0",

"eslint-plugin-react-refresh": "^0.4.5",

"vite": "^5.0.8"

}

}

**Db.json:**

{

"items": [ {

"id": 1,

"title": "Tum Hi Ho",

"singer": "Arijit Singh",

"genre": "Romantic",

"imgUrl": "src/assets/tumhiho.jpg",

"songUrl": "Songs/tumheho.mp3"

},

{

"id": 2,

"title": "Dandelions",

"singer": "Ruth B",

"genre": "Pop",

"imgUrl": "src/assets/dandelion.jpeg",

"songUrl": "Songs/dandelions.mp4"

},

{

"id": 3,

"title": "Vaa Kannamma",

"singer": "Sean Roldan", "genre": "Tamil folk-fusionr",

"imgUrl": "src/assets/vaa.jpeg",

"songUrl": "Songs/vaakannamma.mp4"

},

{

"id": 4,

"title": "Perfect",

"singer": "Ed Sheeran",

"genre": "POP",

"imgUrl": "src/assets/perfect.jpg",

"songUrl": "Songs/perfect.mp4"

},

{

"id": 5,

"title": "Chaleya",

"singer": "Arijit Singh",

"genre": "Romantic",

"imgUrl": "https://akm-img-ain.tosshub.com/indiatoday/images/media\_bank/202309/chaleya-song-1416181931x1.jpg?VersionId=4H9DEXybp0zu2rGuku7n\_C2z8stxuBJZ",

"songUrl": "Songs/chaleya.mp3"

},

{

"id": 6,

"title": "Look Me",

"singer": "Jonas Brothers",

"genre": "POP Rock",

"imgUrl": "src/assets/lookme.jpg",

"songUrl": "Songs/lookme.mp4"

},

{

"id": 7,

"title": "Love Is Gone",

"singer": "Slander & Dylan Matthew",

"genre": "Emotional Ballad", "imgUrl": "src/assets/love.jpeg",

"songUrl": "Songs/iamsorry.mp3"

},

{

"id": 8,

"title": "Dheema Dheema",

"singer": "Anirudh Ravichander",

"genre": "Tamil Flim Soundtrack",

"imgUrl": "src/assets/dheema.jpeg",

"songUrl": "Songs/dheema.mp3"

},

{

"id": 9,

"title": "Love me Like you do",

"singer": "Ellie Goulding",

"genre": "Electropop",

"imgUrl": "src/assets/loveme.jpg",

"songUrl": "Songs/loveme.mp4"

}

],

"favorities": [

{

"itemId": 6,

"title": "Look Me",

"imgUrl": "src/assets/lookme.jpg",

"genre": "POP Rock",

"songUrl": "Songs/lookme.mp4",

"singer": "Jonas Brothers",

"id": 2

},

{

"itemId": 1,

"title": "Tum Hi Ho",

"imgUrl": "src/assets/tumhiho.jpg",

"genre": "Romantic",

"songUrl": "Songs/tumheho.mp3",

"singer": "Arijit Singh",

"id": 3

}

],

"playlist": [

{

"itemId": 9,

"title": "Love me Like you do",

"imgUrl": "src/assets/loveme.jpg",

"genre": "Electropop",

"songUrl": "Songs/loveme.mp4",

"singer": "Ellie Goulding",

"id": 1

},

{

"itemId": 1,

"title": "Tum Hi Ho",

"imgUrl": "src/assets/tumhiho.jpg",

"genre": "Romantic",

"songUrl": "Songs/tumheho.mp3",

"singer": "Arijit Singh",

"id": 2

}

]

}