/src

├── App.jsx

├── components/

│ ├── SongForm.jsx

│ ├── SongList.jsx

│ └── SongItem.jsx

└── main.jsx

Main.jsx

import React from 'react';

import ReactDOM from 'react-dom/client';

import App from './App';

import './index.css';

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

<React.StrictMode>

<App />

</React.StrictMode>

);

App.jsx

import React, { useState } from 'react';

import SongForm from './components/SongForm';

import SongList from './components/SongList';

const App = () => {

const [songs, setSongs] = useState([]);

const addSong = (song) => {

setSongs([...songs, song]);

};

const updateSong = (index, updatedSong) => {

const updatedSongs = [...songs];

updatedSongs[index] = updatedSong;

setSongs(updatedSongs);

};

const deleteSong = (index) => {

const updatedSongs = songs.filter((\_, i) => i !== index);

setSongs(updatedSongs);

};

return (

<div className="min-h-screen bg-gray-100 p-6">

<h1 className="text-3xl font-bold mb-6 text-center text-purple-700">🎵 Music Playlist Manager</h1>

<SongForm addSong={addSong} />

<SongList songs={songs} updateSong={updateSong} deleteSong={deleteSong} />

</div>

);

};

export default App;

songform.jsx

import React, { useState } from 'react';

const SongForm = ({ addSong }) => {

const [title, setTitle] = useState('');

const [artist, setArtist] = useState('');

const handleSubmit = (e) => {

e.preventDefault();

if (!title || !artist) return;

addSong({ title, artist });

setTitle('');

setArtist('');

};

return (

<form onSubmit={handleSubmit} className="mb-6 max-w-xl mx-auto bg-white p-4 rounded shadow">

<h2 className="text-xl font-semibold mb-2">Add New Song</h2>

<input

type="text"

placeholder="Song Title"

value={title}

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

className="w-full p-2 mb-2 border rounded"

required

/>

<input

type="text"

placeholder="Artist"

value={artist}

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

className="w-full p-2 mb-2 border rounded"

required

/>

<button type="submit" className="w-full bg-purple-600 text-white py-2 rounded hover:bg-purple-700">

Add Song

</button>

</form>

);

};

export default SongForm;

import React from 'react';

import SongItem from './SongItem';

const SongList = ({ songs, updateSong, deleteSong }) => {

return (

<div className="max-w-xl mx-auto">

{songs.length === 0 ? (

<p className="text-center text-gray-500">No songs in your playlist.</p>

) : (

<ul className="space-y-4">

{songs.map((song, index) => (

<SongItem

key={index}

index={index}

song={song}

updateSong={updateSong}

deleteSong={deleteSong}

/>

))}

</ul>

)}

</div>

);

};

export default SongList;

Component SongItem.jsx

import React, { useState } from 'react';

const SongItem = ({ index, song, updateSong, deleteSong }) => {

const [isEditing, setIsEditing] = useState(false);

const [editedTitle, setEditedTitle] = useState(song.title);

const [editedArtist, setEditedArtist] = useState(song.artist);

const handleUpdate = () => {

updateSong(index, { title: editedTitle, artist: editedArtist });

setIsEditing(false);

};

return (

<li className="bg-white p-4 rounded shadow">

{isEditing ? (

<>

<input

type="text"

value={editedTitle}

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

className="w-full mb-2 border p-1 rounded"

/>

<input

type="text"

value={editedArtist}

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

className="w-full mb-2 border p-1 rounded"

/>

<div className="flex justify-between">

<button onClick={handleUpdate} className="bg-green-500 text-white px-3 py-1 rounded">

Save

</button>

<button onClick={() => setIsEditing(false)} className="text-gray-500 px-3 py-1">

Cancel

</button>

</div>

</>

) : (

<>

<div className="flex justify-between items-center">

<span>

<strong>{index + 1}. </strong>{song.title} — <em>{song.artist}</em>

</span>

<div className="space-x-2">

<button

onClick={() => setIsEditing(true)}

className="text-blue-500 hover:underline"

>

Edit

</button>

<button

onClick={() => deleteSong(index)}

className="text-red-500 hover:underline"

>

Delete

</button>

</div>

</div>

</>

)}

</li>

);

};

export default SongItem;

index.css

@tailwind base;

@tailwind components;

@tailwind utilities;

Particular Playlist deletion code

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Music Playlist</title>

<style>

body {

font-family: Arial, sans-serif;

background: #f5f5f5;

padding: 20px;

}

h1 {

text-align: center;

color: #663399;

}

.playlist {

max-width: 500px;

margin: 20px auto;

background: white;

padding: 20px;

border-radius: 8px;

box-shadow: 0 0 5px rgba(0,0,0,0.1);

}

.song {

display: flex;

justify-content: space-between;

padding: 10px 0;

border-bottom: 1px solid #ddd;

}

.song:last-child {

border-bottom: none;

}

.song button {

background-color: crimson;

color: white;

border: none;

padding: 5px 10px;

border-radius: 4px;

cursor: pointer;

}

.song button:hover {

background-color: darkred;

}

</style>

</head>

<body>

<h1>🎵 My Playlist</h1>

<div class="playlist" id="playlist"></div>

<script>

let playlist = [

{ title: "Blinding Lights", artist: "The Weeknd" },

{ title: "Levitating", artist: "Dua Lipa" },

{ title: "Shape of You", artist: "Ed Sheeran" },

{ title: "Peaches", artist: "Justin Bieber" },

];

function renderPlaylist() {

const playlistDiv = document.getElementById('playlist');

playlistDiv.innerHTML = '';

playlist.forEach((song, index) => {

const songDiv = document.createElement('div');

songDiv.className = 'song';

songDiv.innerHTML = `

<span><strong>${index + 1}.</strong> ${song.title} — <em>${song.artist}</em></span>

<button onclick="deleteSong(${index})">Delete</button>

`;

playlistDiv.appendChild(songDiv);

});

if (playlist.length === 0) {

playlistDiv.innerHTML = '<p style="text-align:center; color:gray;">Playlist is empty.</p>';

}

}

function deleteSong(index) {

if (confirm(`Are you sure you want to delete "${playlist[index].title}"?`)) {

playlist.splice(index, 1);

renderPlaylist();

}

}

renderPlaylist();

</script>

</body>

</html>