

Data Structures using JavaScript Guided Project

Introduction

This week, we will be building a **browser-based music player application** that enables users to manage and play a collection of songs. The application should provide functionalities for playing, pausing, shuffling, and searching for songs within the provided collection.

While building this application, we will be using various concepts like Array manipulation and Shuffling an array, for this we will be using various functions like `Math.floor()`, `Math.random()`, `indexOf()`, `length()` etc.

Housekeeping points

- This is a minimal example and may not follow some standard practices.
- We focus on the main flow, and not much error handling.

Problem Statement

In this project, we will build a **browser-based music player application** that helps users to manage and play a collection of songs. The application should have the following features :

1. **Playback Control:** Allow users to play, pause, move to the next, and previous tracks in the playlist using dedicated buttons. Users can click on a song in the playlist to start playback. The music player should display song information (title and artist) when a song is played.
2. **Shuffle Functionality:** Enable users to shuffle the playlist for a randomized listening experience. The shuffle button should toggle between shuffling and restoring the original playlist order. On shuffle, the order of the displayed playlist should change, reflecting the new order of songs.
3. **Search Capability:** Implement a search feature to filter songs based on the user's input in real-time. Searching for a song title should dynamically update the displayed playlist to show only matching songs.

Functional Requirements:

- The application should be built using HTML, CSS, and JavaScript.
- Songs data, including details and url, should be stored in a JS file in JSON format.
- The user interface should be clean, intuitive, and responsive.