Software Project

- QIAN VICTOR
- RUBEUS LOUIS
- LAFAYE KILLIAN
- JOULIN SWAN

Project description

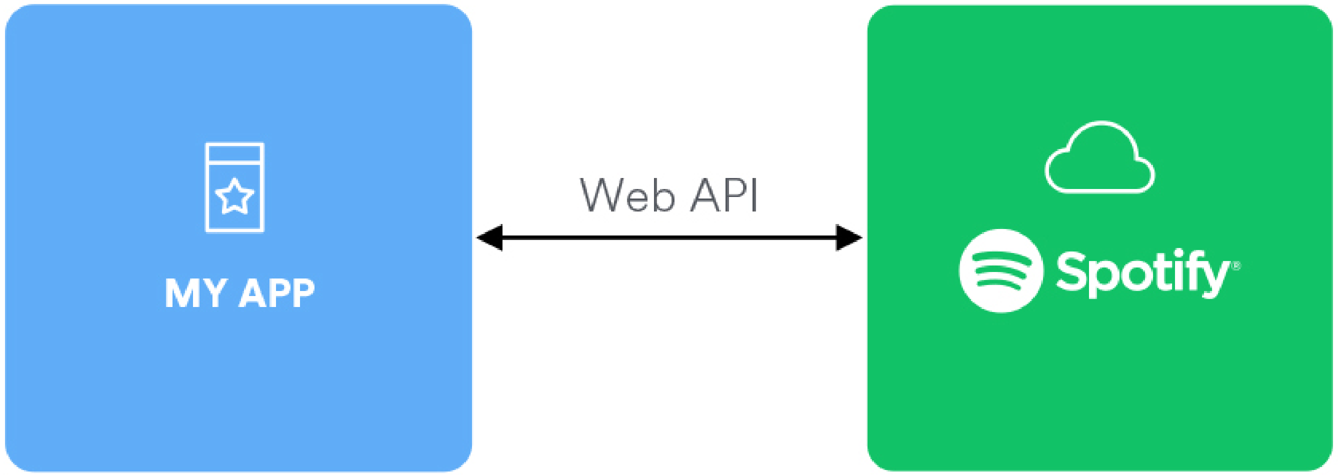
- Website called Moodify

- Suggests music to the user based on a mood and genre

Methods

Language

Library/API



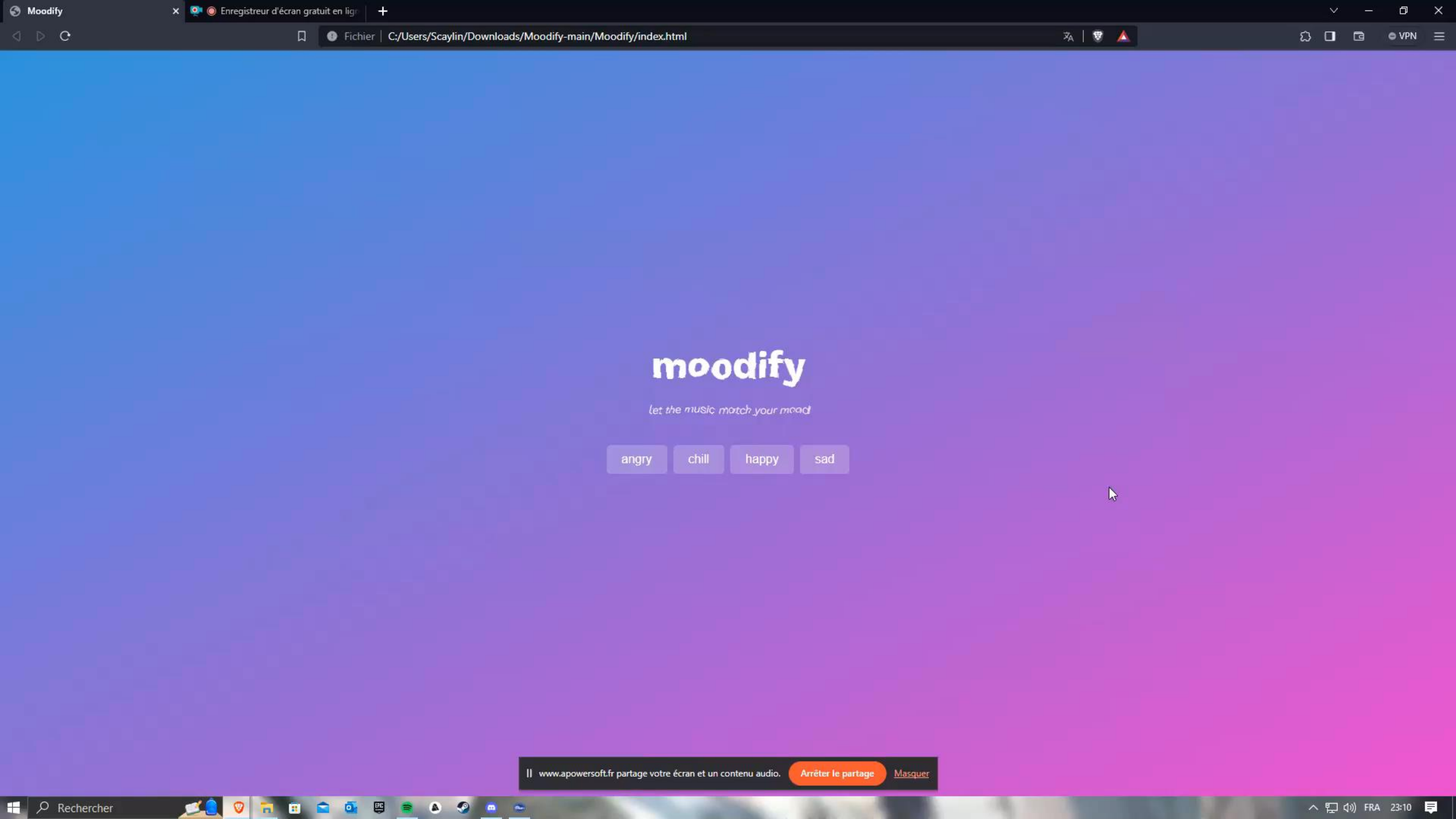
MY APP

Web API

Spotify

Tool used

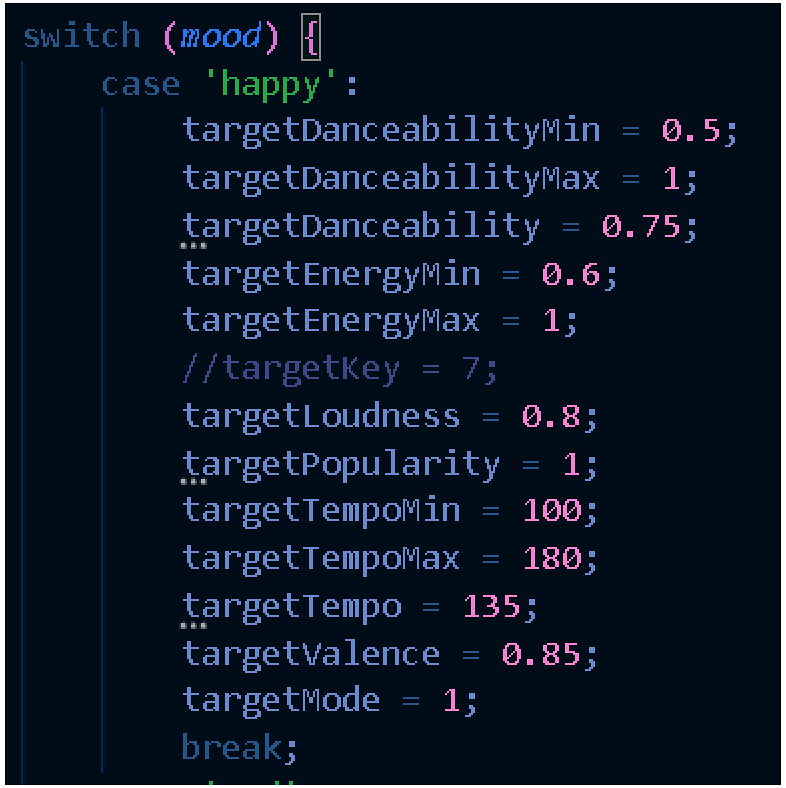
Design choice


Moodify
Enregistreur d'écran gratuit en lign
Fichier
C:/Users/Scaylin/Downloads/Moodify-main/Moodify/index.html
VPN
moodify
let the music match your mood
angry
chill
happy
sad
Arrêter le partage
Masquer
Rechercher
FRA
23:10

www.apowersoft.fr partage votre écran et un contenu audio.

Code insight (1)

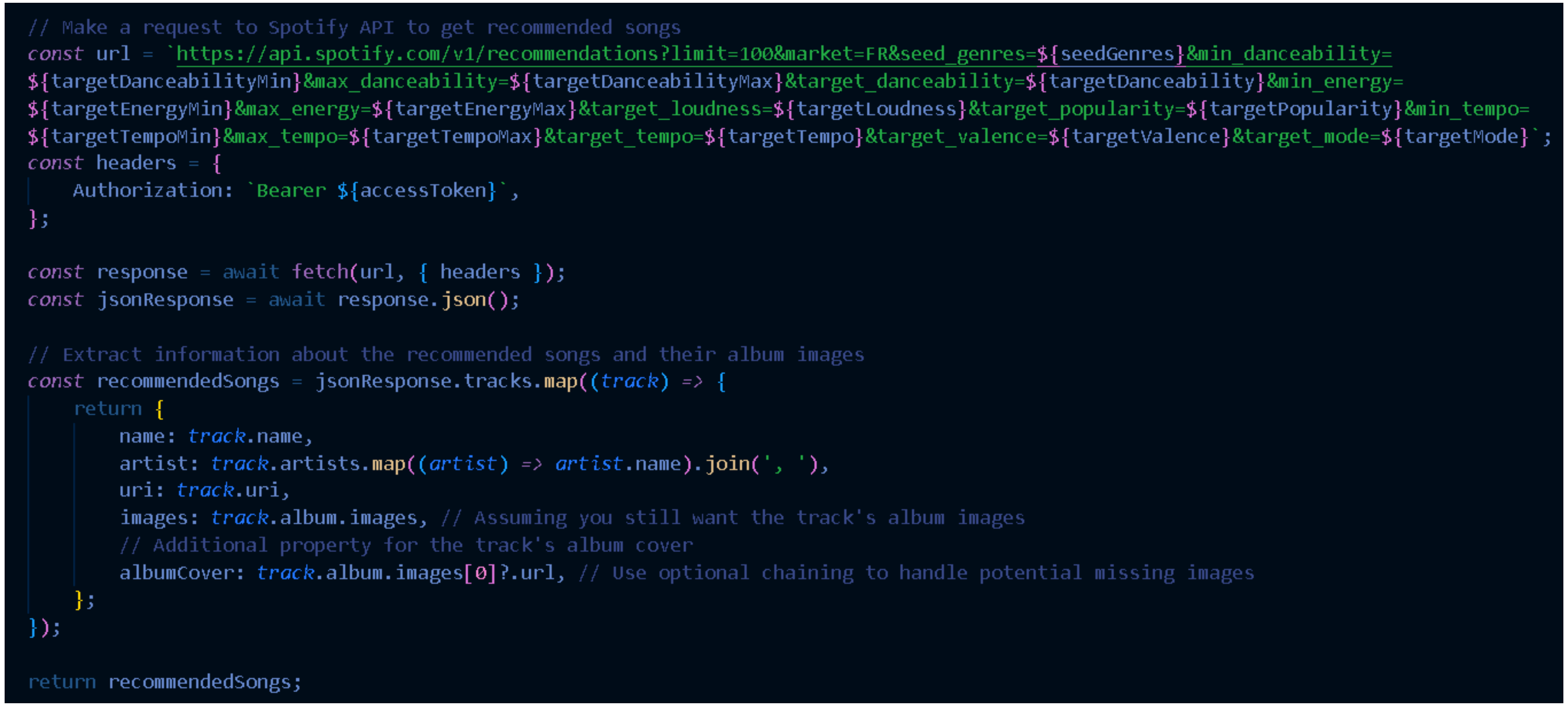
- Variables input



```
switch (mood) {
    case 'happy':
        targetDanceabilityMin = 0.5;
        targetDanceabilityMax = 1;
        targetDanceability = 0.75;
        targetEnergyMin = 0.6;
        targetEnergyMax = 1;
        //targetKey = 7;
        targetLoudness = 0.8;
        targetPopularity = 1;
        targetTempoMin = 100;
        targetTempoMax = 180;
        targetTempo = 135;
        targetValence = 0.85;
        targetMode = 1;
        break;
```

Code insight (2)

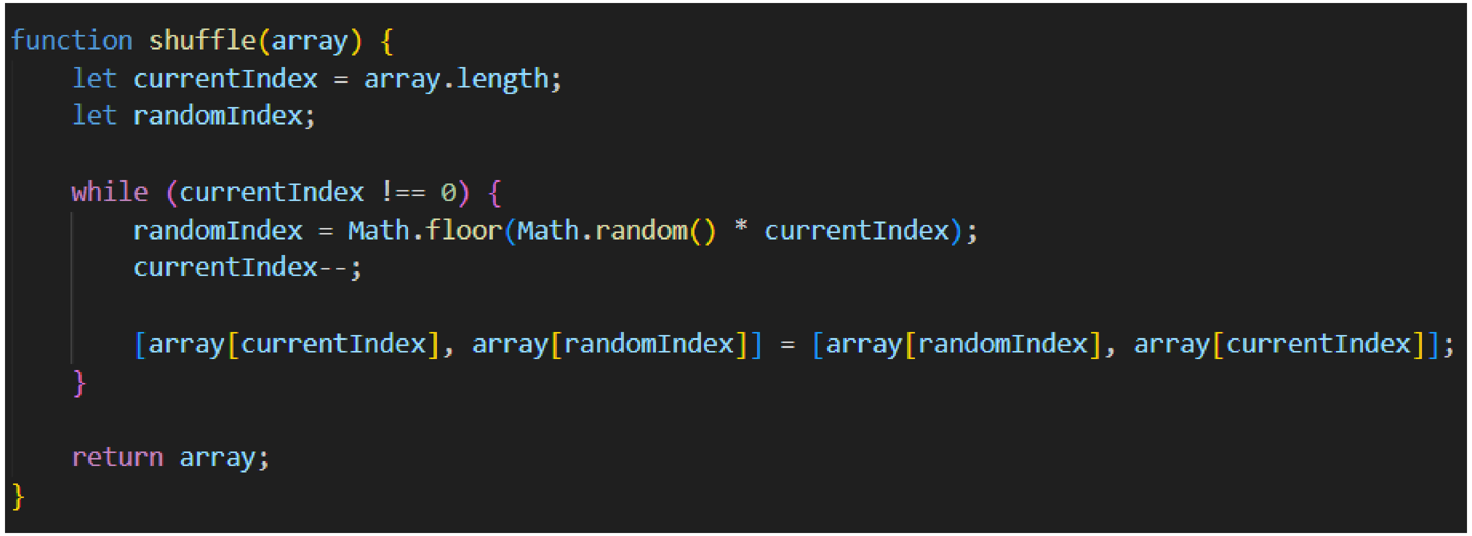
- Link to Spotify



```javascript
// Make a request to Spotify API to get recommended songs
const url = `https://api.spotify.com/v1/recommendations?limit=100&market=FR&seed_genres=${seedGenres}&min_danceability=
${targetDanceabilityMin}&max_danceability=${targetDanceabilityMax}&target_danceability=${targetDanceability}&min_energy=
${targetEnergyMin}&max_energy=${targetEnergyMax}&target_loudness=${targetLoudness}&target_popularity=${targetPopularity}&min_tempo=
${targetTempoMin}&max_tempo=${targetTempoMax}&target_tempo=${targetTempo}&target_valence=${targetValence}&target_mode=${targetMode}`;
const headers = {
    Authorization: `Bearer ${accessToken}`,
};

const response = await fetch(url, { headers });
const jsonResponse = await response.json();

// Extract information about the recommended songs and their album images
const recommendedSongs = jsonResponse.tracks.map((track) => {
    return {
        name: track.name,
        artist: track.artists.map((artist) => artist.name).join(', '),
        uri: track.uri,
        images: track.album.images, // Assuming you still want the track's album images
        // Additional property for the track's album cover
        albumCover: track.album.images[0]?.url, // Use optional chaining to handle potential missing images
    };
});

return recommendedSongs;
```

Code insight (3)

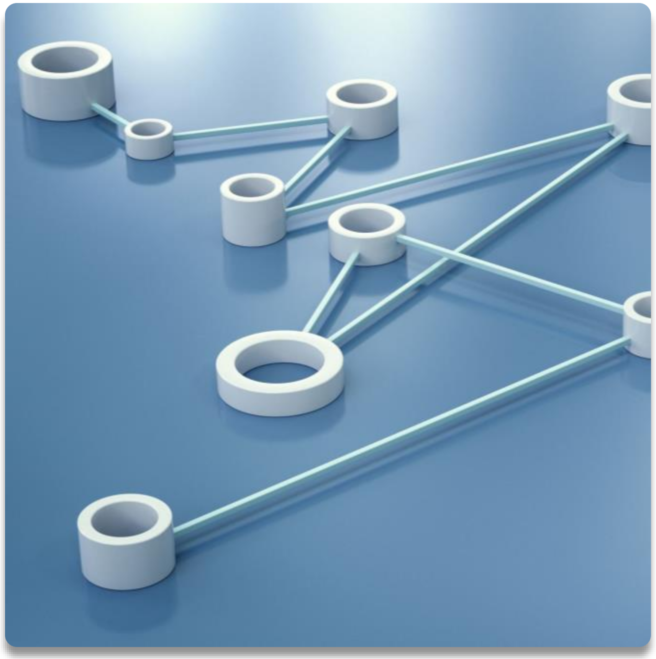
- Shuffle function



```
function shuffle(array) {
    let currentIndex = array.length;
    let randomIndex;

    while (currentIndex !== 0) {
        randomIndex = Math.floor(Math.random() * currentIndex);
        currentIndex--;

        [array[currentIndex], array[randomIndex]] = [array[randomIndex], array[currentIndex]];
    }

    return array;
}
```

Retrospective

- Recommendation algorithm
- API Spotify
- Application --> site
- Too much scattering



Conclusion

Happy with our prototype

Our next steps :

- Expansion (not only music)
- Optimise the recommendation algorithm
- Improve the design of the site
- New functionnalities
- additional API (Deezer)