



ÉCOLE CENTRALE LYON

L^AT_EX APPROFONDI
RAPPORT

Projet Application WEB

Élèves :

AL Mahdi LAHRACHE
Oussama HADDA

Enseignant :

2 avril 2023

Table des matières

1	Introduction	2
2	Cahier des charges	2
3	Javascript coté serveur	2
4	Implémentation de l'application	3
5	Pistes de développement supplémentaires	4
6	Conclusion	4

1 Introduction

Nous avons choisi de concevoir une application web multimedia pour exploiter pleinement les fonctionnalités offertes par le langage JavaScript et la plateforme Node.js, en association avec les langages HTML et CSS. Notre projet vise à permettre à l'utilisateur de profiter d'une expérience musicale interactive en ligne. En explorant les fonctionnalités de ces langages et plateformes, nous avons travaillé à la conception de l'interface utilisateur, à la gestion des fichiers audio et à la mise en place de fonctionnalités telles que la recherche de musique et l'ajout de fichiers audio à la bibliothèque musicale. Nous avons ainsi pu approfondir notre compréhension des fonctionnalités de JavaScript et de Node.js, et des capacités de développement web qu'ils offrent.

2 Cahier des charges

- Permettre à l'utilisateur la possibilité de choisir la musique à jouer.
- Permettre à l'utilisateur de voir si un audio est disponible en tapant le titre de l'audio sur la barre de recherche.
- Permettre à l'utilisateur la possibilité de chercher si un audio existe en tapant le titre sur une barre de recherche.
- Permettre à l'utilisateur la possibilité de télécharger la musique.
- Permettre à l'utilisateur de sélectionner ou glisser les fichiers (mp3) pour les ajouter à la playlist.
- Permettre à l'utilisateur le contrôle de la lecture (pause, lecture, avance rapide, retour rapide)
- Affichage des métadonnées du fichier audio en cours de lecture (titre, artiste, album,...)
- Permettre à l'utilisateur la possibilité d'aimer une musique tout en enregistrant les musiques aimées dans un répertoire ou historique.
- Permettre à l'utilisateur de voir son historique.

3 Javascript coté serveur

Nous avons utilisé principalement le framework Express de Node.js pour créer notre serveur, ainsi que plusieurs modules pour faciliter la gestion des fichiers mp3. Nous avons commencé par créer la fonction `gettags`, qui utilise `jsmediatags.read()` pour extraire les tags des fichiers mp3 tels que le nom de l'artiste, le titre, la durée et l'image. Ensuite, nous avons utilisé la fonction `Remplirdb` pour remplir un objet db avec les tags des fichiers mp3 disponibles dans le dossier `public/songs/` et pour stocker ces informations dans un fichier JSON nommé `playlist.json`.

```
// fs sera utilisé pour lire / écrire .. et manipuler des fichiers
var fs = require('fs');

// Path sera utilisé pour manipuler les chemins des fichiers manipulés
var path = require('path');

// Récupérer la durée d'un fichier .mp3
var getMP3Duration = require('get-mp3-duration') // npm install --save get-mp3-duration

// jsmediatags.read() est la fonction utilisée dans la suite, c'est pour récupérer les données ( tags ) d'un mp3 file
var jsmediatags = require('jsmediatags')
```

FIGURE 1 – Modules utilisés et utilité dans la suite

Nous avons également créé la fonction `extraire()` pour extraire les fichiers audio présents sur le serveur et stocker leurs noms dans un fichier JSON. Enfin, nous avons créé une requête GET pour afficher le contenu de la page `index.html` et avons mis le serveur à l'écoute du port 8080.

4 Implémentation de l'application

Le lecteur de musique est organisé de manière ergonomique pour l'utilisateur. La liste numérotée des audios disponibles est facilement accessible en bas de la barre de recherche. L'audio qui est en lecture est mis en évidence en haut de la liste, accompagné du cover de l'audio. La barre de lecture en bas permet à l'utilisateur de naviguer facilement dans l'audio en cours de lecture.

Pour la partie serveur de l'application, l'équipe a utilisé le Framework Express, qui a permis de faciliter la tâche en permettant d'envoyer toutes les requêtes nécessaires, notamment pour la création de la page web.

La première fonction, appelée "extraire path", permet de renvoyer le chemin d'un dossier ou d'un fichier audio. Ensuite, la fonction "crawl" permet de récupérer tous les chemins des fichiers audio présents sur l'ordinateur, et d'écrire ces chemins dans un fichier JSON sous forme de paires clé-valeur, avec le numéro de chemin comme clé et le chemin comme valeur.

Dans la partie HTML de l'application, une fonction appelée "get data" est utilisée pour envoyer une requête AJAX afin de récupérer le fichier JSON contenant les chemins de tous les fichiers audio. Cette fonction inclut également d'autres fonctions, telles que des événements de clic pour ajouter des fichiers audio au tableau de lecture.

En ajoutant des fonctionnalités telles que la possibilité de créer des playlists ou de partager des musiques avec des amis, l'application pourrait être encore améliorée pour offrir une expérience de musique en ligne plus complète et conviviale pour l'utilisateur.

5 Pistes de développement supplémentaires

Le premier point de développement supplémentaire pour un lecteur de médias pourrait être l'ajout de fonctionnalités d'interaction sociale. Par exemple, une fonctionnalité de partage sur les réseaux sociaux qui permettrait aux utilisateurs de partager leur musique préférée avec leurs amis. Cette fonctionnalité pourrait être mise en œuvre en utilisant des API de réseaux sociaux telles que Twitter ou Facebook. De plus, une autre fonctionnalité

intéressante pourrait être la possibilité de créer des listes de lecture collaboratives. Cela permettrait à un groupe d'utilisateurs de collaborer sur une liste de lecture, ajoutant ou supprimant des chansons selon leurs préférences. Les utilisateurs pourraient également voter pour les chansons qu'ils préfèrent, créant ainsi une liste de lecture dynamique et interactive. De plus, une autre piste de développement pourrait être l'ajout d'une fonc-

tionnalité de reconnaissance vocale pour le contrôle du lecteur de musique. Cette fonctionnalité permettrait aux utilisateurs de contrôler la lecture de leur musique à l'aide de commandes vocales telles que "lecture", "pause" ou "suivant". Cette fonctionnalité pourrait être mise en œuvre en utilisant des API de reconnaissance vocale telles que la SpeechRecognition API de Google.

Enfin un autre point intéressant à développer est de pouvoir développer un modèle de suggestion de musique basé sur les anciennes musiques aimées et des fréquences des musiques entendues.

6 Conclusion

Ce projet Web nous a permis d'appliquer les compétences que nous avons acquises au cours du semestre en électronique d'application web, y compris en HTML, CSS et JavaScript, ainsi que dans la construction de serveurs en utilisant Node.js. Nous avons mis en pratique nos connaissances en utilisant ces technologies pour créer une application Web fonctionnelle qui répond aux besoins de l'utilisateur. Ce projet nous a également permis de découvrir de nouveaux concepts et de les intégrer dans notre travail, tout en nous offrant une expérience pratique précieuse pour notre avenir professionnel.