



ÉCOLE  
**CENTRALE**LYON

# ÉCOLE CENTRALE LYON

UE ELECTIFS  
APPLICATION WEB  
RAPPORT

## Projet Media Player

*Élèves :*  
Alric JAMES  
Damien CARRASQUET

*Enseignants :*  
Daniel MULLER  
Rene CHALON

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Fonctionnalités</b>	<b>2</b>
<b>3</b>	<b>Implémentation technique</b>	<b>4</b>
<b>4</b>	<b>Conclusion</b>	<b>5</b>

## 1 Introduction

Pour ce travail en autonomie, nous avons décidé de créer un lecteur de musique. Ce travail fut réalisé grâce à NodeJs et Express. Le produit final est donc un site internet proposant une interface de lecture de musiques en local. Ce rapport présente les différentes fonctionnalités de cette application ainsi que son implémentation technique, tout en montrant plusieurs visuels illustrant les explications.

## 2 Fonctionnalités

Pour commencer, énonçons les différentes fonctionnalités de l'application. Son rôle principal est donc la lecture de musiques présentes sur l'ordinateur de l'utilisateur.

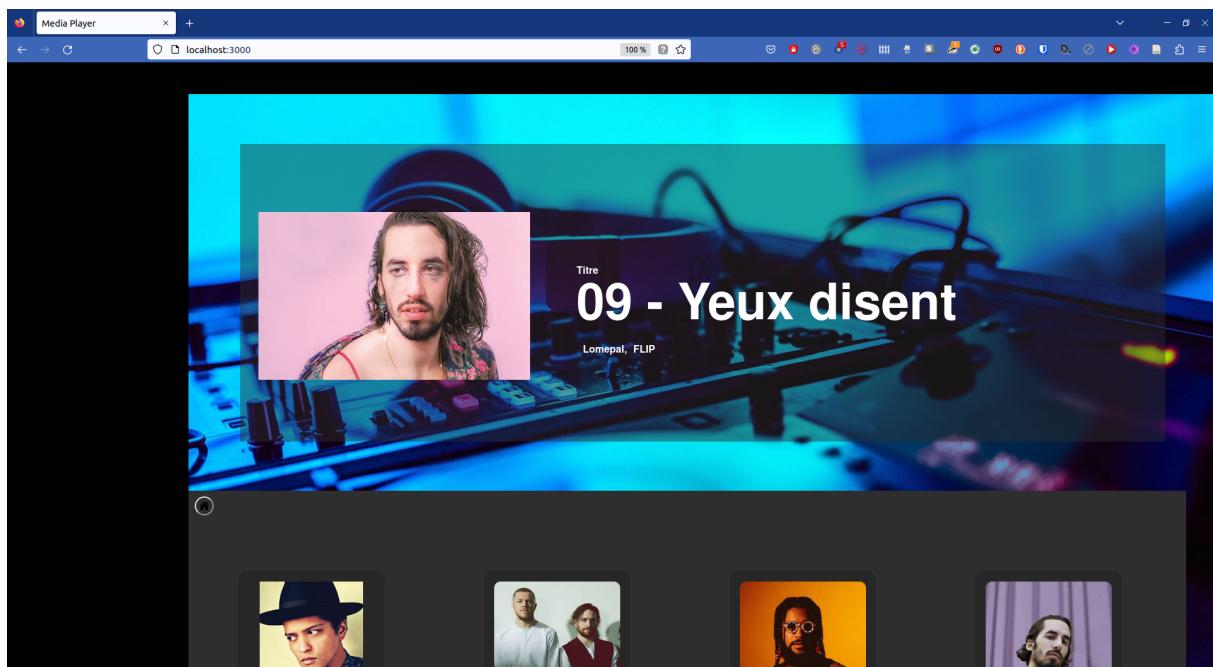


FIGURE 1 – Accueil du site avec une musique aléatoire mise en avant

L'utilisateur peut donc naviguer à travers le site dans ses dossiers sous réserve d'une organisation précise des dossiers et fichiers musicaux de son ordinateur comme suit : un dossier pour chaque artiste, qui contiennent chacun un dossier pour chaque album de l'artiste, qui eux-mêmes contiennent les musiques de l'album. L'utilisateur peut également bénéficier d'un plus visuel lors de sa recherche s'il possède un dossier contenant des images pour les albums et artistes de son répertoire. Lors de sa navigation, l'utilisateur peut revenir à tout moment à la liste des artistes.

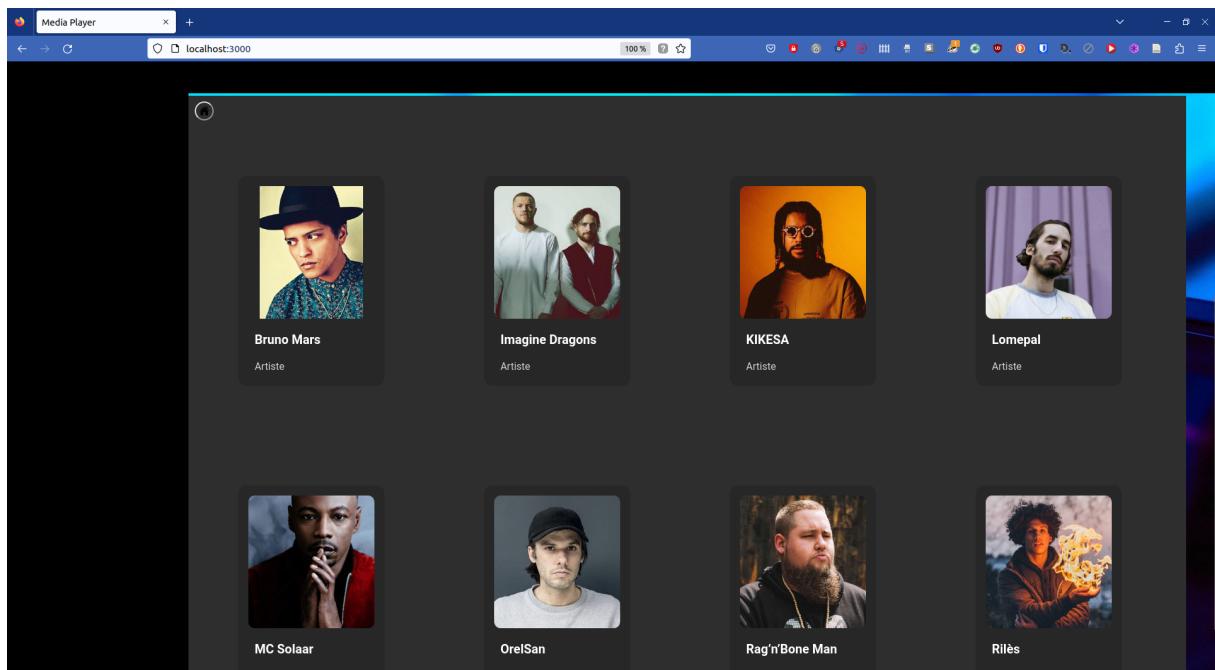


FIGURE 2 – Affichage de tous les artistes contenus dans le dossier Musique

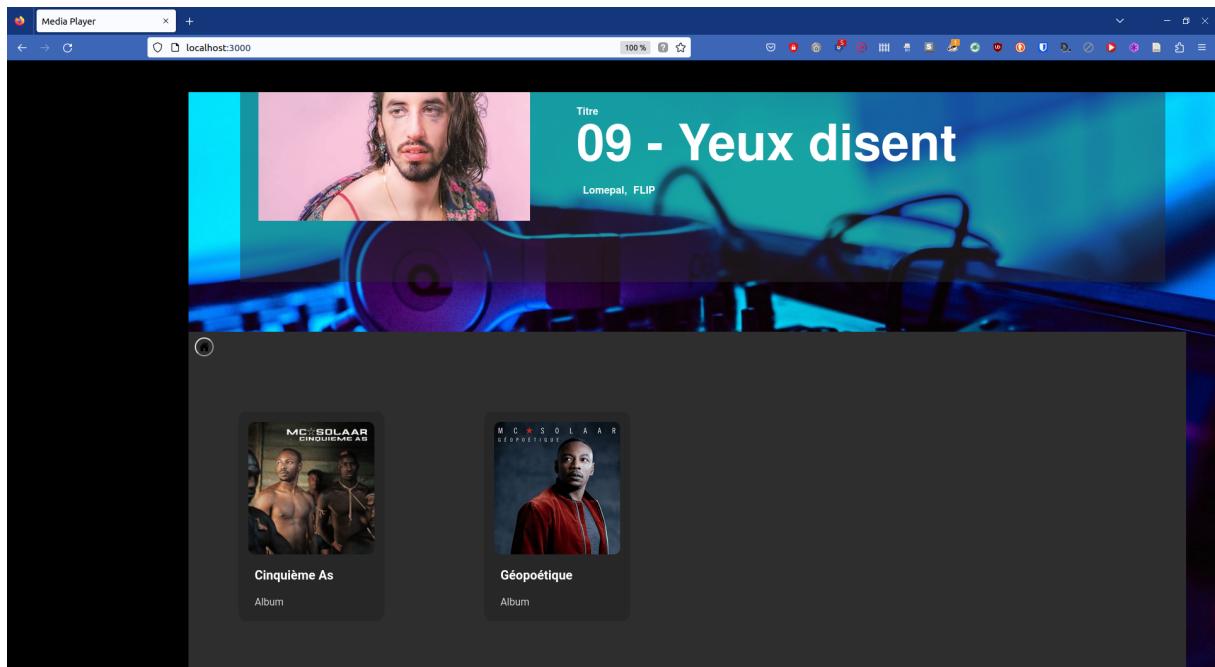


FIGURE 3 – Affichage des albums d'un artiste

Lorsqu'il se décide de cliquer sur une musique, il est redirigé vers une route où il peut lire sa musique grâce au lecteur intégré. Ce lecteur permet la lecture et pause de la musique, ainsi que l'accès à n'importe quel moment de la musique via la barre d'avancement, ainsi que la possibilité de passer à la prochaine ou précédente musique de l'album. L'interface de musique est toujours présente, l'utilisateur peut donc choisir une autre album ou artiste à tout moment et continuer son écoute.

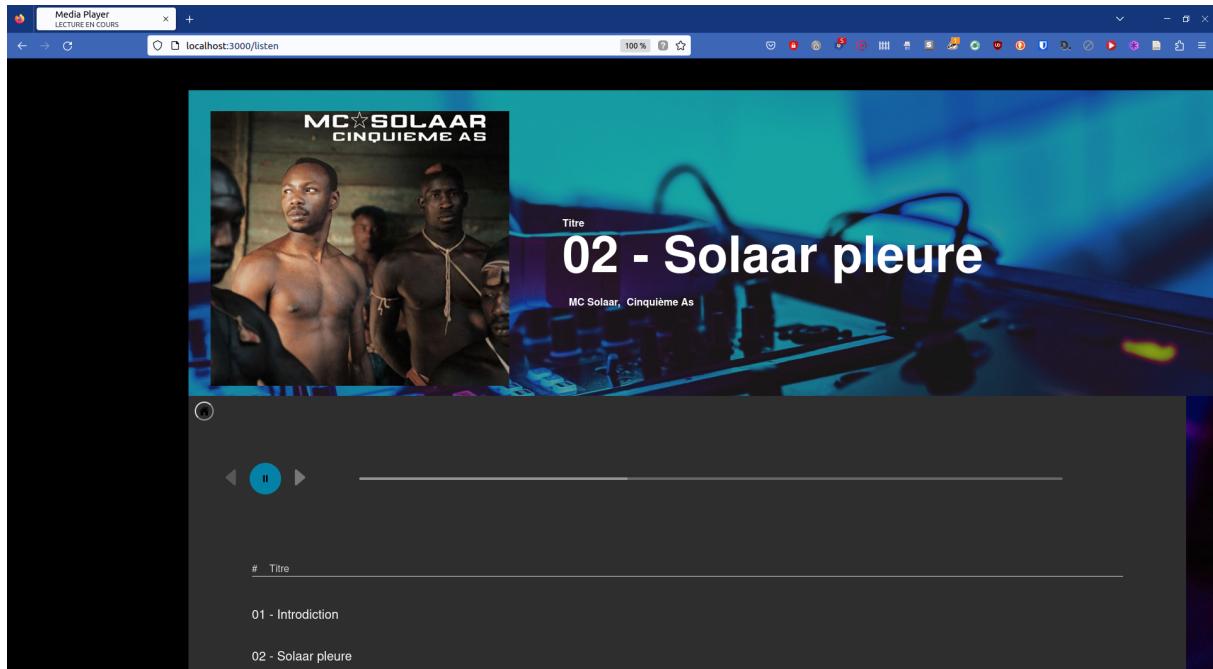


FIGURE 4 – Page de lecture d'une musique avec la liste des musiques de l'album correspondant

Une autre fonctionnalité présente également est la proposition aléatoire de musique. Pour faire varier les plaisirs de l'utilisateur, une musique aléatoire de son répertoire est mise en avant sur la page d'accueil du site à chaque fois qu'il l'ouvre. Il peut évidemment la lire en cliquant dessus, ouvrant la page de lecture de musique accompagné de l'album auquel la musique appartient.

### 3 Implémentation technique

L'application est implémentée en Express et NodeJs. Elle constituée de deux routes (fichiers .ejs), 2 fichiers .js associés, ainsi que le fichier get.js qui prend en compte les requêtes au serveur et le fichier qui contient le squelette de l'application. Les deux routes, index.ejs et listen.ejs correspondent respectivement à la page d'accueil et la page du lecteur de musique de l'application. Au delà de créer les routes, le fichier App contient les variables globales de la musique en cours de lecture, ainsi que les dossiers dans l'ordinateur de l'utilisateur dans lesquels le site doit chercher les musiques ainsi que les images. Dans les deux pages, on retrouve deux grandes div principales : la div frontMusic qui contient l'affichage de la musique proposée en page d'accueil et en cours de lecture pour la page listen, ainsi que la div folders contenant l'interface de recherche de musique. Au lancement de la page d'accueil, deux choses se produisent :

- La musique mise en avant est tirée aléatoirement en parcourant le dossier racine contenu dans la variable globale, et en sélectionnant aléatoirement un artiste, puis un de ses album, et enfin une de ses musiques. Les informations sont envoyées au client qui les affiche sur la page d'accueil.
- La fonction de display-artiste est lancée. Elle permet de parcourir le dossier racine et d'afficher tout les artistes contenus dedans, afin de permettre à l'utilisateur de débuter sa recherche. Chaque div (ou card) correspondant à un artiste appelle une

fonction display-album, qui vide la div "folders" lorsqu'elle est appelée pour laisser place aux albums de l'artiste qui sont obtenus en parcourant le dossier de l'artiste.

L'affichage des musiques par sélection d'un album fonctionne de la même manière.

Lors d'un clic sur une musique, que ce soit la musique recommandée ou une musique recherchée, l'utilisateur est redirigé vers la route listen. La variable globale de la musique actuelle est actualisée, et permet l'utilisation du lecteur du musique ainsi que l'affichage de l'album dans la div folder de la nouvelle route afin que l'utilisation puisse continuer sa recherche s'il le souhaite. Le lecteur audio est reliée à un bouton qui lance Play ou Pause, des boutons de prochaine ou dernière musique, ainsi qu'à une barre de progression créée par des div (et non une progressBar de hmtl), qui est reliée par onClick à des fonctions faisant appel et modifiant les informations temporelles de la musique.

## 4 Conclusion

Ce travail en autonomie nous a permis de prendre en main Javascript et d'appréhender les possibilités offertes par NodeJS et Express pour concevoir un site Web interactif. Un point intéressant à explorer aurait été de mettre en lien notre application avec une base de données SQL mais nous n'avons pas trouvé l'utilité dans notre cas.

D'autre part, cette application a de nombreuses possibilités d'améliorations. En effet, elle ne prend en compte pour l'instant que le format .ogg pour les musiques et .jpeg pour les images des artistes et des albums. Pouvoir mixer les différents formats de fichier serait moins restrictif pour l'utilisateur. La création et la lecture de playlists seraient une fonctionnalité intéressante mais nous n'avons pas eu le temps de l'implémenter.

Pour finir, si vous souhaitez tester l'application, il vous suffit de cloner le répertoire git, de modifier dans le fichier app.js l'emplacement des dossiers de musique et d'images, puis exécuter la commande suivante dans un terminal en étant dans le projet git :

```
node bin/www
```

N'hésitez pas à nous contacter en cas de problème avec le git. Si vous n'avez pas de dossiers d'images et de musiques compatibles avec notre application, vous pouvez télécharger un fichier .zip pouvant être utilisé pour notre application à l'adresse suivante :

<https://nextcloud.equoprivacy.com/s/5L33s3n9wQFdKzL>.