



Applications WEB

Rapport

---

# Lecteur Multimédia Groovy

---

<https://github.com/IlyasBBB/Lecteur-Multim-dia.git>

*Auteurs :*

Ayoub ASSEFFAR

Ilyas BENALLA

Version du  
3 avril 2025

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Choix techniques</b>	<b>2</b>
2.1	HTML, CSS et JavaScript : Structure et Dynamisme . . . . .	2
2.2	Node.js et Backend . . . . .	2
2.3	Intégration d'API Externes . . . . .	3
<b>3</b>	<b>Présentation de l'application</b>	<b>3</b>
3.1	Lecture de médias locaux . . . . .	3
3.2	Importation de fichiers . . . . .	3
3.3	Gestion de bibliothèque . . . . .	4
3.4	Playlists personnalisables . . . . .	4
3.5	Historique de lecture . . . . .	4
3.6	Intégration YouTube . . . . .	4
3.7	Intégration Spotify . . . . .	4
3.8	Fonctionnalités avancées . . . . .	4
3.9	Responsive design . . . . .	5
3.10	Sécurité et performances . . . . .	5
<b>4</b>	<b>L'application en images</b>	<b>5</b>
4.1	Page d'accueil . . . . .	5
4.2	Barre du menu . . . . .	6
4.3	Lecteur audio . . . . .	6
4.4	Création de playlists . . . . .	7
4.5	Lecteur video . . . . .	7
4.6	Lecteur images . . . . .	8
4.7	Page Youtube . . . . .	8
4.8	Page Spotify . . . . .	9
<b>5</b>	<b>Conclusion &amp; Perspectives</b>	<b>9</b>
5.1	Conclusion . . . . .	9
5.2	Perspectives . . . . .	9

# 1 Introduction

Dans le cadre de notre projet, nous avons développé une application de lecteur multimédia afin de mettre en pratique nos acquis du cours. Ce projet s'inscrit dans une démarche d'apprentissage alliant théorie et pratique.

L'objectif principal était de concevoir une plateforme intuitive et fonctionnelle permettant de lire, organiser et gérer divers types de contenus multimédias. En intégrant des technologies modernes du web, nous avons cherché à optimiser l'expérience utilisateur tout en garantissant une bonne performance et une accessibilité sur différents supports.

Ce rapport détaille les différentes étapes de conception et de réalisation du projet, en mettant en avant les choix techniques effectués ainsi que les défis rencontrés lors du développement.

## 2 Choix techniques

Pour le développement de notre lecteur multimédia, nous avons opté pour un ensemble de technologies modernes du web afin d'assurer une application fluide, interactive et extensible. Nos choix se sont portés sur **HTML**, **CSS**, **JavaScript**, **Node.js**, ainsi que l'intégration d'**API externes** comme YouTube et Spotify.

### 2.1 *HTML, CSS et JavaScript : Structure et Dynamisme*

- **HTML5** : Utilisé pour la structure du site, permettant d'intégrer le lecteur multimédia avec les balises `<audio>`, `<video>` et `<canvas>` pour l'affichage des images et l'édition basique.
- **CSS3** : Permet de styliser l'interface utilisateur, avec un design responsive et des modes clair/sombre. Des animations et transitions ont également été utilisées pour améliorer l'expérience utilisateur.
- **JavaScript (ES6+)** : Gère l'interactivité du site, comme la lecture des fichiers multimédias, la gestion des playlists et l'historique. Nous avons exploité les fonctionnalités de manipulation du DOM et des événements pour une navigation fluide.

### 2.2 *Node.js et Backend*

- **Node.js** : Utilisé pour la gestion du backend, permettant d'héberger l'application et de gérer les requêtes des utilisateurs.

- **Express.js** : Framework minimaliste pour gérer les routes et les API nécessaires à l'interaction avec les fichiers locaux et les services externes.
- **Stockage en Base64 / LocalStorage** : Permet de sauvegarder temporairement les fichiers multimédias sans base de données lourde, optimisant ainsi la rapidité d'accès.

### 2.3 Intégration d'API Externes

- **API YouTube IFrame** : Utilisée pour rechercher et lire des vidéos YouTube directement dans notre application. Elle permet également de charger des playlists et d'interagir avec les commandes de lecture.
- **Spotify API** : Permet d'accéder aux titres musicaux et aux playlists Spotify. L'authentification OAuth est utilisée pour récupérer les informations de l'utilisateur et gérer les restrictions entre comptes premium et gratuits.

## 3 Présentation de l'application

L'application se distingue par une interface moderne et intuitive, permettant aux utilisateurs d'importer leurs médias, de créer des playlists personnalisées et d'accéder rapidement à leur historique de lecture. En intégrant des API externes telles que YouTube et Spotify, elle offre également la possibilité de rechercher et lire du contenu en ligne directement depuis la plateforme.

Notre objectif principal était de concevoir un lecteur multimédia complet, combinant les fonctionnalités d'un lecteur local avec l'accès à des services de streaming, tout en assurant une navigation ergonomique et une compatibilité multiplateforme.

Les principales caractéristiques de l'application incluent :

### 3.1 Lecture de médias locaux

- Prise en charge des fichiers audio, vidéo et images.
- Lecture via le lecteur HTML5 intégré.
- Gestion des erreurs et formats non supportés.

### 3.2 Importation de fichiers

- Glisser-déposer (drag & drop) des fichiers.
- Sélection via l'explorateur de fichiers.

- Conversion et stockage en base64.

### **3.3 *Gestion de bibliothèque***

- Stockage local des médias importés.
- Affichage organisé par type (audio, vidéo, image).
- Statistiques d'utilisation (nombre de fichiers, taille totale).
- Suppression individuelle ou totale des médias.

### **3.4 *Playlists personnalisables***

- Création de playlists nommées.
- Ajout/suppression de médias dans les playlists.
- Lecture séquentielle des playlists.

### **3.5 *Historique de lecture***

- Suivi des derniers médias joués.
- Accès rapide aux contenus récents.
- Possibilité de vider l'historique.

### **3.6 *Intégration YouTube***

- Recherche de vidéos et musiques.
- Lecture via l'API YouTube IFrame.
- Mode musique avec contrôles simplifiés.
- Cache des résultats de recherche.

### **3.7 *Intégration Spotify***

- Connexion via OAuth.
- Deux modes de lecture :
  - Comptes premium : lecture complète.
  - Comptes gratuits : prévisualisations de 30 secondes.
- Affichage des covers d'albums.

### **3.8 *Fonctionnalités avancées***

- Édition d'images (rotation, zoom, filtres).

- Raccourcis clavier (Ctrl+Shift+R pour réinitialiser).
- Mode sombre/clair (selon le code CSS associé).
- Gestion des erreurs et messages utilisateur.

### 3.9 *Responsive design*

- Adapté aux différents écrans.
- Barres de progression interactives.
- Notifications contextuelles.

### 3.10 *Sécurité et performances*

- Limitation de la taille du cache.
- Gestion des quotas de stockage.
- Optimisation des requêtes API.

Cette application combine ainsi les fonctionnalités d'un lecteur multimédia local avec l'accès à des plateformes externes comme YouTube et Spotify, le tout dans une interface unifiée.

## 4 L'application en images

### 4.1 *Page d'accueil*

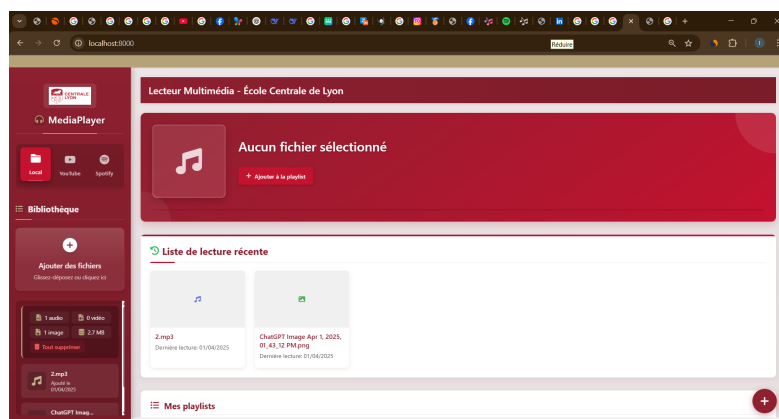


FIGURE 1 – Première page

## 4.2 Barre du menu

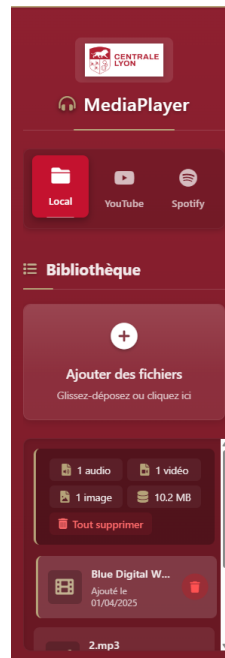


FIGURE 2 – Barre du menu

## 4.3 Lecteur audio

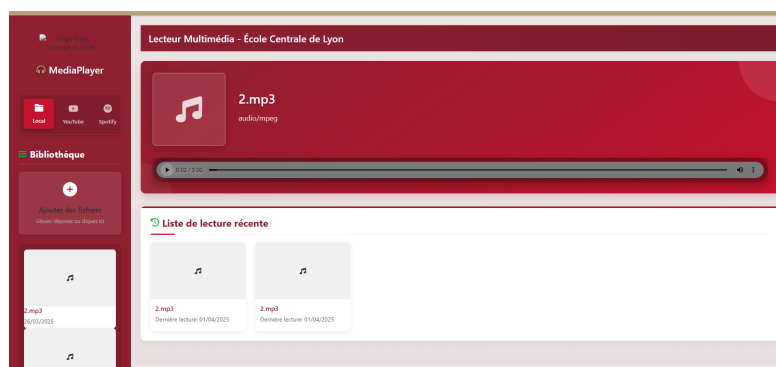


FIGURE 3 – Lecteur Audio

## 4.4 Création de playlists

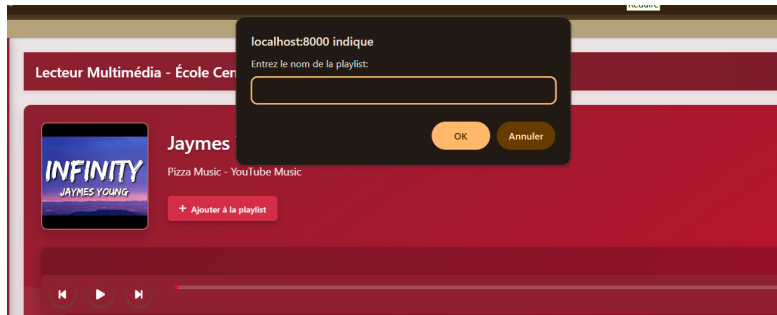


FIGURE 4 – Création de playlists

## 4.5 Lecteur video

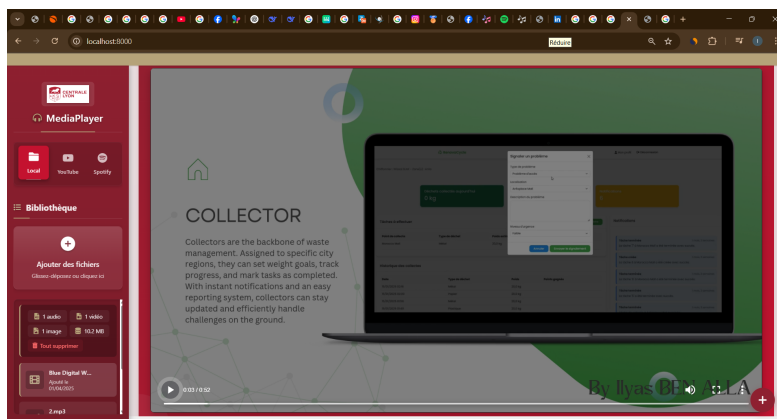


FIGURE 5 – Lecteur Video



## 4.6 Lecteur images

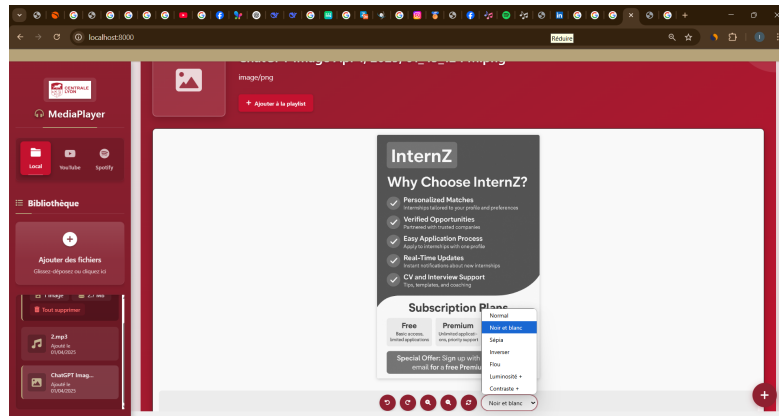


FIGURE 6 – Lecteur Images

## 4.7 Page Youtube

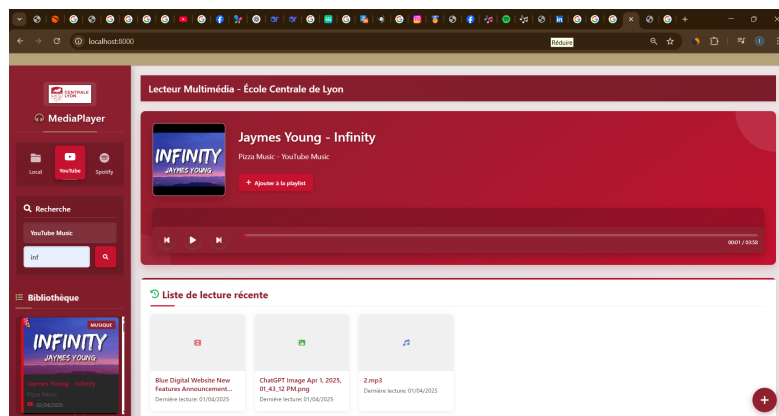


FIGURE 7 – Page YouTube

## 4.8 Page Spotify

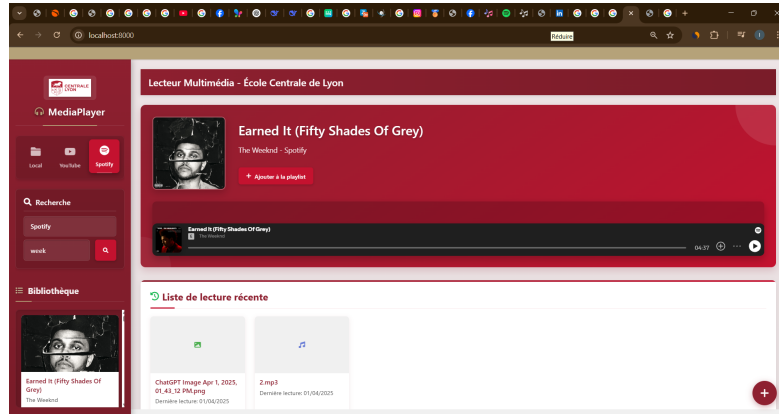


FIGURE 8 – Page Spotify

## 5 Conclusion & Perspectives

### 5.1 Conclusion

Ce projet a permis de développer une application multimédia complète et interactive, intégrant à la fois un lecteur de fichiers locaux et des fonctionnalités avancées telles que l'accès à YouTube et Spotify. Grâce à l'utilisation des technologies modernes comme HTML5, CSS3, JavaScript et Node.js, l'application offre une interface fluide et adaptable à différents types de médias et appareils. Les fonctionnalités telles que l'importation de fichiers, la gestion des playlists, et l'édition d'images viennent enrichir l'expérience utilisateur et permettent de gérer efficacement les contenus multimédias.

Le design responsive et l'intégration des API externes contribuent à rendre l'application pertinente et dynamique, répondant ainsi aux besoins des utilisateurs recherchant une solution centralisée pour leurs médias.

### 5.2 Perspectives

Malgré les fonctionnalités riches déjà implémentées, plusieurs pistes d'amélioration et d'extension sont envisageables pour rendre l'application encore plus complète et performante :

- **Amélioration de la gestion des formats multimédias** : Actuellement, le support des formats multimédias est limité. L'ajout de la prise en charge

d'autres formats (par exemple, des fichiers audio FLAC ou des vidéos en 4K) permettrait d'étendre la compatibilité de l'application.

- **Partage de médias** : Ajouter la possibilité de partager des médias entre utilisateurs, via des liens ou des réseaux sociaux, renforcerait l'aspect collaboratif de l'application.
- **Amélioration de la sécurité** : Intégrer une gestion des utilisateurs avec authentification et autorisation (par exemple, via un système de comptes et de sessions sécurisées) garantirait la protection des données personnelles.
- **Fonctionnalités de recommandation** : En intégrant des algorithmes de recommandation basés sur l'historique de lecture et les préférences des utilisateurs, l'application pourrait suggérer des contenus audio, vidéo ou image adaptés à chaque utilisateur.
- **Optimisation des performances** : Une meilleure gestion du cache et des requêtes API permettrait d'améliorer encore la réactivité de l'application, notamment pour les utilisateurs avec des connexions internet lentes.

Ces améliorations permettraient de répondre aux besoins des utilisateurs de manière encore plus fluide et d'étendre les fonctionnalités offertes par l'application pour en faire un outil encore plus puissant et polyvalent.