

projet de rattrapage M1

École-IT 2024/2025

→ Réalisateur : Lyna AIT BRAHAM

→ ID: 105366

Présentation Globale du Projet Spotify

1. Introduction

Ce projet a été réalisé dans le cadre du rattrapage du Master 1 en Data Science et IA. Il consiste à créer une application de gestion de chansons complète, incluant un frontend en Angular, un backend Node.js/Express, une base de données MongoDB, des tests API via Postman, une visualisation BI avec Power BI, ainsi qu'un plan de continuité et un plan marketing.

2. Technologies utilisées

- **Frontend** : Angular 19
- **Backend** : Node.js + Express.js
- **Base de données** : MongoDB avec Mongoose
- **API** : architecture RESTful
- **BI** : Power BI Desktop
- **Tests** : Postman
- **Versionnage** : GitHub (local)


3. Fonctionnalités développées

Gestion des chansons (Frontend)

L'application propose une interface utilisateur intuitive avec des formulaires dynamiques et un affichage réactif des chansons ajoutées.

1-Page d'accueil Angular et listes des chansons

localhost:4200

 Spotify Frontend

[Voir les chansons](#) | [Ajouter une chanson](#)

Liste des chansons

Titre	Artiste	Genre	Durée (min)	Action	
je t'aime	lara fabian	pop	208	Supprimer	Modifier
adagio	lara fabian	pop	254	Supprimer	Modifier
tout	lara fabian	pop	231	Supprimer	Modifier
humana	lara fabian	pop	225	Supprimer	Modifier
Je suis malade	Lara Fabian	Vérité française	300	Supprimer	Modifier
Il y a trop de gens qui t'aiment	Hélène Ségara	Variété française	240	Supprimer	Modifier
Elle, tu l'aimes	Hélène Ségara	Variété française	230	Supprimer	Modifier
L'amour est un soleil	Hélène Ségara	Variété française	240	Supprimer	Modifier
Une voix dans la nuit	Hélène Ségara	Variété française	260	Supprimer	Modifier
Tu vas me quitter	Hélène Ségara	Variété française	215	Supprimer	Modifier
On n'oublie jamais rien, on vit avec	Hélène Ségara & Laura Pausini	Variété française	275	Supprimer	Modifier
Il faut que tu t'en ailles	Hélène Ségara	Variété française	225	Supprimer	Modifier
Zora sourit	Céline Dion	Variété française	250	Supprimer	Modifier
Pour que tu m'aimes encore	Céline Dion	Variété française	270	Supprimer	Modifier
Je sais pas	Céline Dion	Pop	265	Supprimer	Modifier
My Heart Will Go On	Céline Dion	Pop	280	Supprimer	Modifier
Sous le vent	Garou & Céline Dion	Pop	260	Supprimer	Modifier
Et si tu n'existais pas	Joe Dassin	Variété française	230	Supprimer	Modifier
L'été indien	Joe Dassin	Variété française	260	Supprimer	Modifier
Les Champs-Élysées	Joe Dassin	Variété française	195	Supprimer	Modifier
À toi	Joe Dassin	Variété française	250	Supprimer	Modifier
Salut	Joe Dassin	Variété française	240	Supprimer	Modifier
Whenever, Wherever	Shakira	Pop	210	Supprimer	Modifier
Hips Don't Lie	Shakira	Pop	220	Supprimer	Modifier
Waka Waka (This Time for Africa)	Shakira	Pop	210	Supprimer	Modifier
Beautiful Liar	Shakira & Beyoncé	Pop	230	Supprimer	Modifier
Underneath Your Clothes	Shakira	Pop	235	Supprimer	Modifier
Chasing the Sun	Beyoncé	Pop	225	Supprimer	Modifier

2 – Formulaire d'ajout



The screenshot shows a web browser window with the address bar displaying 'localhost:4200/add'. The page title is 'Spotify Frontend'. Below the title, there are two links: 'Voir les chansons' and 'Ajouter une chanson'. The main heading is 'Ajouter une chanson'. The form contains four input fields: 'Titre', 'Artiste', 'Genre', and 'Durée (en secondes)'. The 'Durée' field has a value of '0'. There is an 'Ajouter' button at the bottom.

← → ↻ ⓘ localhost:4200/add

Spotify Frontend

[Voir les chansons](#) | [Ajouter une chanson](#)

Ajouter une chanson

Titre :

Artiste :

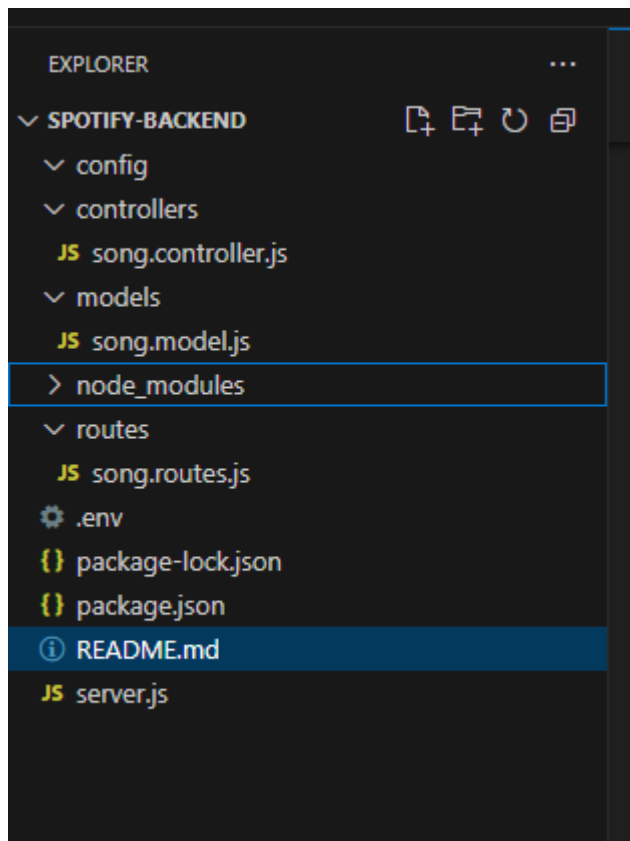
Genre :

Durée (en secondes) :

API Backend (Node.js + Express)

Le backend est conçu en architecture RESTful avec Express.js. Il contient des routes pour gérer les chansons (CRUD), et utilise des middlewares comme [body-parser](#) et [cors](#).

4 – Vue backend dans VS Code



Tests API avec Postman

Tous les endpoints ont été testés via Postman pour garantir leur bon fonctionnement (POST, GET, PUT, DELETE).

Capture 6 – Test GET (récupération chansons)

The screenshot shows the Postman interface with a GET request to `http://localhost:3000/api/songs`. The response is a JSON array of two song objects, displayed in the 'Body' tab. The status is 200 OK, with a response time of 37 ms and a size of 8.85 KB.

Request: `GET http://localhost:3000/api/songs`

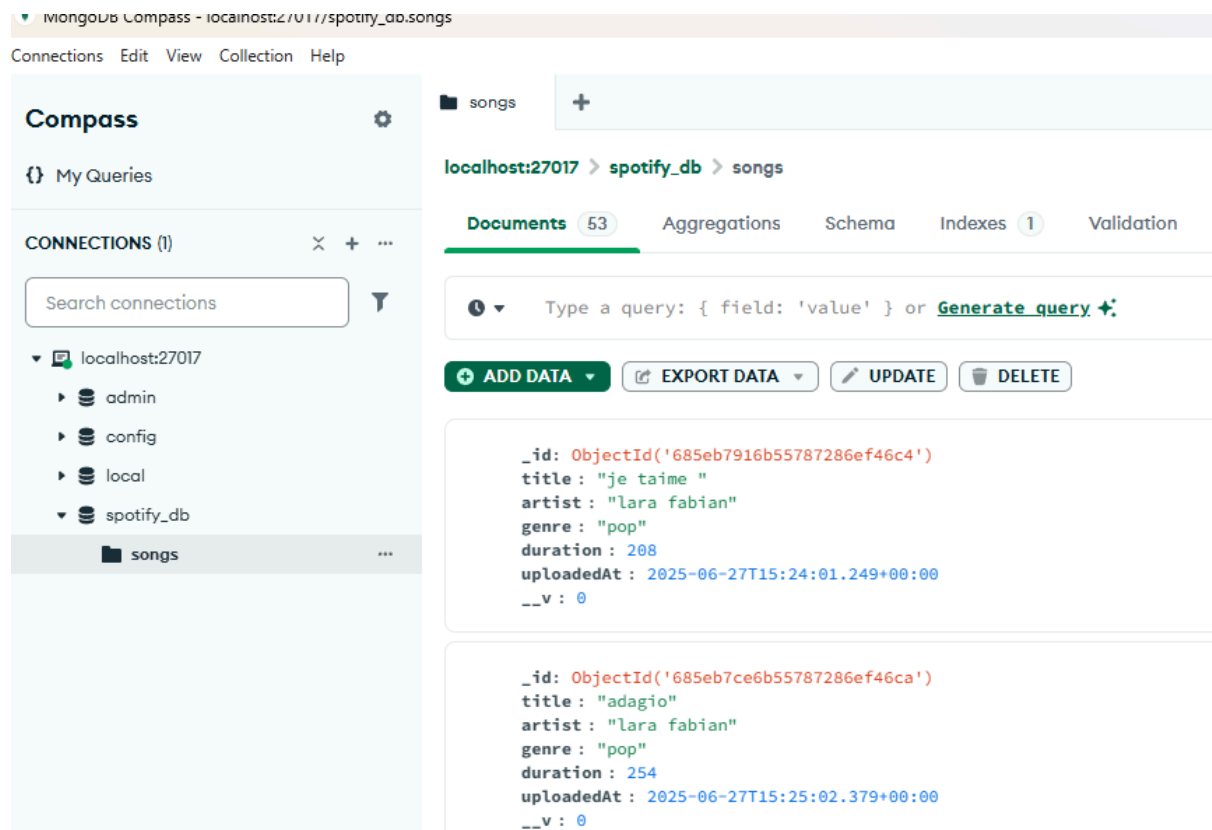
Response Body (JSON):

```
1 {
2   {
3     "_id": "685eb7916b55787286ef46c4",
4     "title": "je t'aim",
5     "artist": "lara fabian",
6     "genre": "pop",
7     "duration": 208,
8     "uploadedAt": "2025-06-27T15:24:01.249Z",
9     "__v": 0
10  },
11  {
12    "_id": "685eb7ce6b55787286ef46ca",
13    "title": "adagio",
14    "artist": "lara fabian",
15    "genre": "pop",
16    "duration": 254,
17    "uploadedAt": "2025-06-27T15:25:02.379Z",
```

Base de données MongoDB

Les données sont stockées dans MongoDB via Mongoose. Une collection est dédiée aux chansons avec des champs personnalisés.

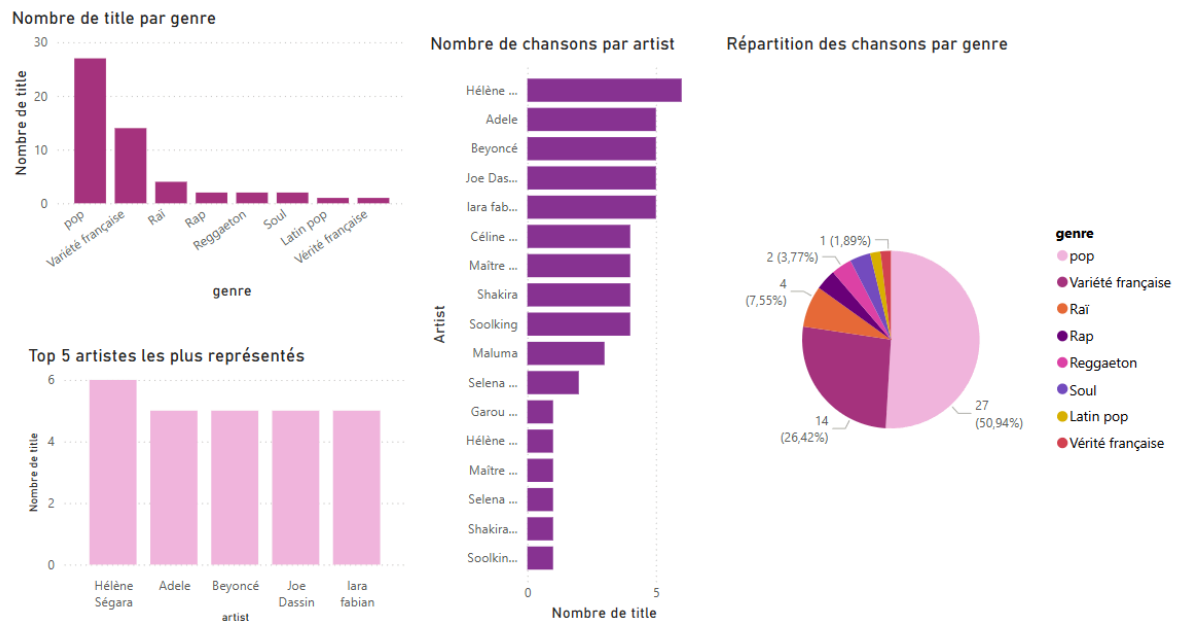
7 – MongoDB Compass



Visualisation avec Power BI

Un tableau de bord Power BI a été conçu en important les données JSON depuis l'API. Il affiche des indicateurs clés comme le nombre de chansons par artiste.

8 – Rapport Power BI



4. Conclusion

Ce projet m'a permis de mobiliser mes compétences en développement web fullstack, en gestion de données, en visualisation BI et en organisation de projet.

Chaque étape a été réalisée avec rigueur, depuis le backend jusqu'à la mise en page finale et l'optimisation de l'interface.