

Projet Application de Musique - Musique & Chill.

Présentation:

Une application sur laquelle on pourra écouter de nouvelles musiques en fonction de nos goûts, et qui se rendra adaptative en fonction de nos préférences.

Le projet contient quatres composantes, une API, la partie recommandation qui gérera les musiques à découvrir, une ux adaptée et intuitive, et enfin la BDD.

Rôles de l'équipe:

Corentin FAGET à majoritairement contribué au côté front de l'application, c'est pourquoi il hérita de la casquette de Lead Dev Front. Il a pu apporter son expertise pour la conceptualisation de l'ui et pour gérer les échanges de données avec la partie API.

Edouard LAFFONT était quant à lui Lead Dev Back, puisqu'il est majoritairement intervenu sur la partie API et BDD de l'application bien qu'il ait également pu intervenir sur le front également.

Quentin GUIHENEUC enfin a endossé un rôle approchant celui de Scrum Master. En effet, en plus d'intervenir sur le code côté back et front, il a coordonné les efforts de l'équipe et organisé le développement des fonctionnalités du projet au cours des différents sprints.

Technologies utilisées :

Front: React, axios et typescript notamment ont permis de construire une interface stylisé, stable et permettant l'échange de données avec l'AP

Back: Node, Postgresql, Sequelize(ORM). Une API se basant sur une BDD Postgresql soutenue par un ORM, Sequelize.

Structure algorithmique :

L'équipe a estimé que le modèle MVC était le plus adapté à l'application. Principalement du fait que c'est une structure très efficace et surtout, la mieux appréhendée par l'équipe. Cela permettait donc de se focaliser sur d'autres enjeux techniques, notamment l'utilisation d'un ORM qui fut une nouveauté pour tous les membres de l'équipe.

Liste des fonctionnalités validées :

- Création et connection d'utilisateurs.
- Afficher une liste de musique disponible à l'écoute.
- Écouter (pause, avancer dans le morceau, revenir au morceau précédent, aller au morceau suivant) de la musique.
- Permettre une recherche de musique via une barre de recherche dédiée.
- Ajouter un "like" à un morceau écouté.
- Accéder à la liste des "likes" de l'utilisateur via une page Favoris.
- Accéder à une écoute de morceaux personnalisés basée sur les préférences de l'utilisateur

Explications des fonctionnalités principales :

- Gestion du player: Nous avons adapté le player react-player en utilisant ses principales fonctions pour qu'elles répondent au mieux à nos besoins. De même en utilisant Redux on peut gérer les différentes musiques et les données qu'elles renvoient par utilisateur

- Gestion des recommandations: l'équipe a fait le choix de baser son système de recommandation sur le type de la musique en argumentant que cette solution permettait une plus grande versatilité de recommandation. Grâce à Sequelize on peut récupérer les types de musiques les plus appréciés de l'utilisateur via les "like" attribués et proposer des recommandations en suivant. Pour éviter que l'utilisateur soit enfermé dans un seul type de musique, on renvoie toujours des recommandations basées sur les trois types les plus appréciés actuellement par l'utilisateur.