

Spotifiie

Groupe 26: Rachid Krita, Yasmine Ouyahya, Roxane Douc et Marie Hyvernaud

12 Mai 2020

1 Introduction

Le but du projet est de mettre en place un site web qui servirait de plateforme musicale pour les élèves de l'école. En effet, chaque année, certaines associations organisent des événements et pourraient ainsi animer leurs activités avec des playlists sur mesures. De plus, l'association Muzzik organise régulièrement des concerts avec les groupes de l'école. Le site permettrait alors aux utilisateurs disposant d'un compte d'accéder aux créations musicales enregistrées des promos actuelles et précédentes.

La répartition des tâches s'est effectuée de la manière suivante: Marie a créé la base de données relationnelle PostgreSQL tandis que Rachid s'est chargé de sa gestion afin de satisfaire aux opérations CRUD (Create Read Update Delete). Yasmine s'est occupée de la programmation en JavaScript et celle du PHP associé et Roxane de l'implémentation des animations CSS et, avec Marie du code PHP. Nous avons créé un git afin de partager les fichiers et que chacun puisse suivre l'avancé du projet.

2 Description de la base de données

La base de données nommée "init.sql" contient tout d'abord une table "utilisateurs". Celle-ci contient toutes les informations personnelles, ainsi que le mot de passe chiffré à l'aide du protocole de hachage sha1. Les informations relatives aux musiques sont stockées dans la table "musiques". En particulier, une clé primaire permet d'identifier de manière unique chacune des chansons. Toutefois, les fichiers musicaux sont stockés à l'extérieur de la base. Avec ce système, nous pouvons accéder de manière unique aux fichiers mp3, stockés dans le répertoire "musiques". Pour permettre aux utilisateurs de créer leurs propres playlists, nous avons créé une table "playlist" associée à une table d'association "musiquesdansplaylist" qui permet de stocker les musiques présentes dans chaque playlist. Enfin, nous avons voulu donner aux utilisateurs la possibilité d'écrire un commentaire, qui est stocké dans la table "commentaires".

3 Fonctionnalités

3.1 Gestion d'utilisateurs/inscription:

Lors de l'ouverture du site, l'utilisateur est deconnecté. Il a alors accès à une interface basique lui permettant de s'identifier ou de s'inscrire via un formulaire contenant des champs obligatoires. Une fois loggué, l'utilisateur a accès au coeur du site web. Son login est stocké dans une session, qui permettra aussi de stocker d'autres variables comme nous le verrons par la suite.

3.2 Écoute de musique pendant navigation

La principale difficulté technique de notre site réside dans le fait qu'il nous est impossible de recharger totalement la page, au risque d'arreter la musique en cours d'écoute et de la faire reprendre du debut. De ce fait, nous avons ete contraints d'utiliser javascript, jQuery et surtout Ajax qui nous permet de modifier une seule partie du site web sans recharger la page complète.

Ainsi, au lieu d'utiliser des boutons de type "submit" dans les menus, nous avons systématiquement géré manuellement les clics en javascript dans le fichier "js/code.js". En outre, la methode ".load" de Ajax nous a permis de modifier les parties souhaitées de la page. Il faut cependant noter que ".load" fait toujours appel à index.php , car c'est le seul fichier dans lequel toutes les variables definies jusqu'a present sont reconnues. Aussi, nous faisons appel à index avec un paramètre "todo" passé en méthode GET, ainsi que d'éventuels paramètres additionnels. Cela mène donc a une structure légèrement complexe au sein de la page "index.php" qui teste toutes les valeurs possibles pour "todo". Dans le cas où on ne recharge qu'une partie de la page, l'appel à index se termine toujours par "exit(0);" afin de ne recharger que la section voulue.

3.3 Recherche et affichage des musiques

Il est possible de rechercher une musique via la zone de texte située en haut à droite de la page. La recherche de musique s'effectue avec le mot clé "LIKE" au sein d'une requête SQL. Cette requête est effectuée au sein d'une méthode de la classe "Playlist" du fichier "adoDB/playlist.php". Nous obtenons en retour toutes les musiques dont le titre ou l'asso contient le mot clé demandé. Bien entendu, tout ceci se fait toujours sans recharger la page complète, et donc à l'aide de javascript et Ajax. L'affichage des musiques est pris en charge par le module Datatable, un plugin de jQuery. Cela permet d'obtenir un affichage agréable des musiques dans un tableau permettant de trier suivant différentes colonnes, en plusieurs pages, etc...

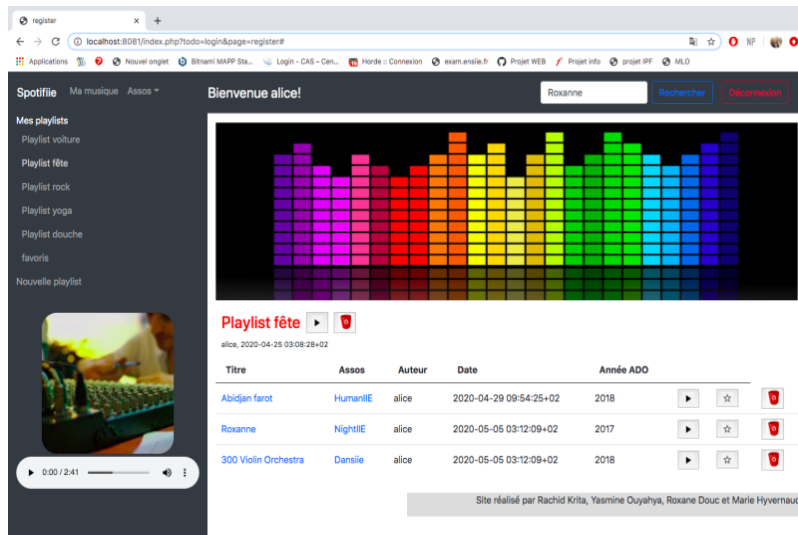


Figure 1: Recherche de musique

3.4 Affichage des musiques par assos

L'utilisateur peut également demander l'affichage de toutes les musiques d'une assos. Pour cela, il peut tout d'abord faire appel au menu déroulant "Assos". Il peut également cliquer sur le lien d'une asso correspondant à une musique lorsque celle-ci est affichée à l'écran. Une fois la page de l'asso affichée, l'utilisateur peut lancer la playlist de toutes les musiques de l'asso qui vont vous être jouées à la suite automatiquement. Le fonctionnement des playlists est détaillé dans la partie "Écouter une playlist".

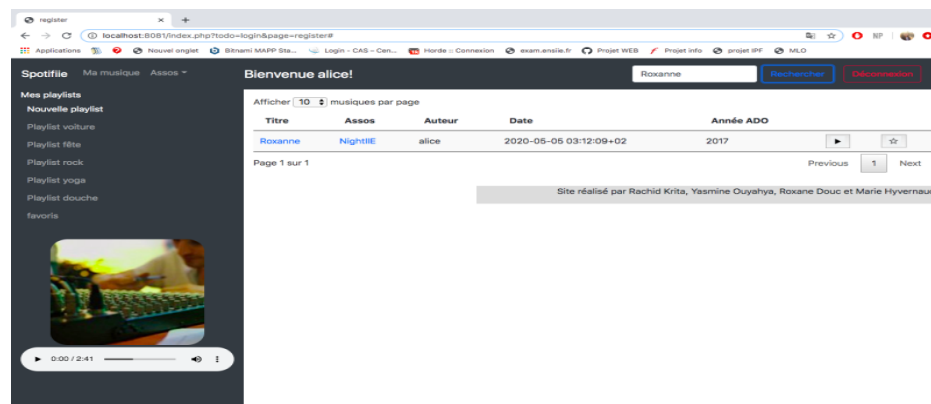


Figure 2: Affichage des musiques par assos

3.5 Commentaires

L'utilisateur a la possibilité de laisser un commentaire sur une musique. Pour cela, il doit se rendre sur la page de la musique en cliquant sur le titre de cette dernière, puis poster un commentaire dans le cadre prévu à cet effet. Les commentaires s'actualisent automatiquement pour tous les utilisateurs dès qu'ils sont postés.

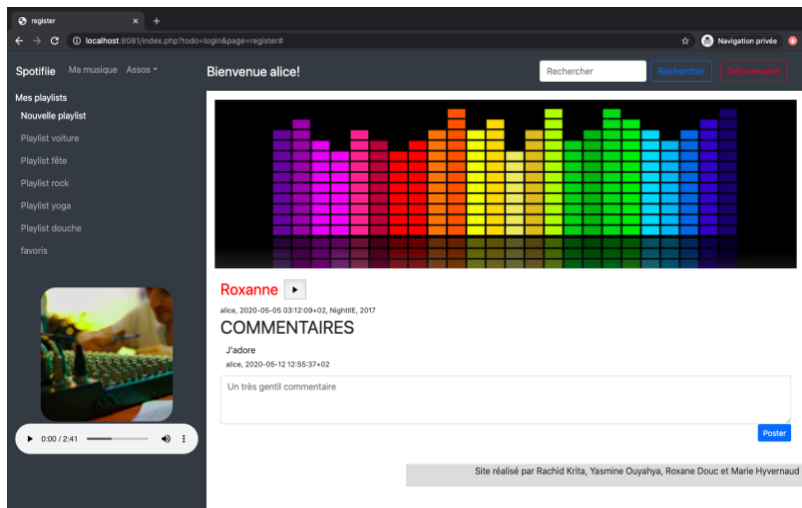


Figure 3: Ecrire un commentaire

3.6 Création/suppression de playlists perso

Lors de la création de son compte, le site crée automatiquement une playlist “favoris”, également accessible via l’onglet “Ma musique”. Pour créer ses propres playlists, l'utilisateur doit cliquer sur “Nouvelle playlist”. Un popup s’affiche alors, lui permettant de choisir un nom. Une fois créée, la liste des playlists est automatiquement mise à jour à gauche de l’écran. Pour supprimer une playlist, il suffit de cliquer dessus, puis de cliquer sur l’icône corbeille qui supprimera instantanément la playlist.

3.7 Ajout/suppression de musiques a une playlist

Pour ajouter une musique à une playlist, il faut cliquer sur l’icône en forme d’étoile associé à la musique choisie. Une fenêtre popup permet alors de sélectionner la playlist à laquelle on souhaite ajouter la musique. Pour supprimer une musique d’une playlist, il suffit d’aller sur la page de la playlist, puis de cliquer sur l’icône corbeille associée à la musique à retirer.

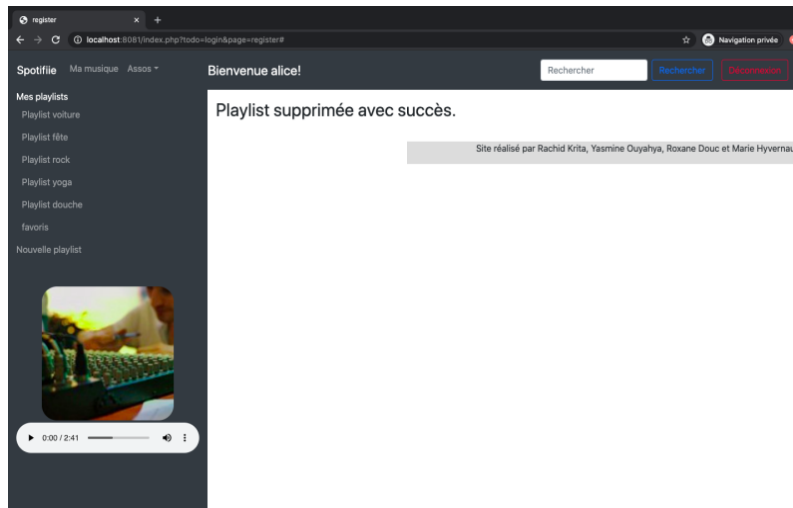


Figure 4: Supprimer une playlist

3.8 Écouter une playlist

Nous avons défini deux modes d’écoute. Le premier est le mode “single”, a savoir que lorsque l’utilisateur clique sur une musique, uniquement celle-ci est jouée. Au contraire, le mode playlist lance l’enchaînement automatique de toutes les musiques d’une playlist. Pour écouter une playlist, il faut se rendre sur la page de la playlist et appuyer sur le bouton Play. Les musiques s’enchaînent alors, et la playlist recommence une fois terminée. Pour obtenir cette fonctionnalité, nous avons eu besoin de stocker dans des variables de session la liste des musiques de la playlist, ainsi que la musique en cours. Dès que la première musique de la playlist se termine, un événement javascript se déclenche ce qui a pour effet de faire un appel à index à l’aide d’Ajax. C’est dans index que l’on fait appel à la musique suivante et qu’on la lance.

A tout moment, si l’utilisateur lance une autre musique, le mode “playlist” s’arrête pour repasser en mode “single”.

4 Description de l’interface visuelle

Le site s’appuie essentiellement sur la bibliothèque Bootstrap pour la mise en page, et le plugin jQuery Datatable pour gérer l’affichage et l’exploration des musiques de la base de données, le reste étant écrit en css à la main. L’interface principale (lorsque un utilisateur est connecté), présente une barre de menu horizontale fixe en haut de la page, permettant la recherche de musique et la déconnexion. Un volet de navigation vertical fixe à gauche permet de naviguer parmi les différentes listes de lecture de l’utilisateur, et contient également le lecteur audio et son illustration. Ces deux éléments fixes constituent le squelette

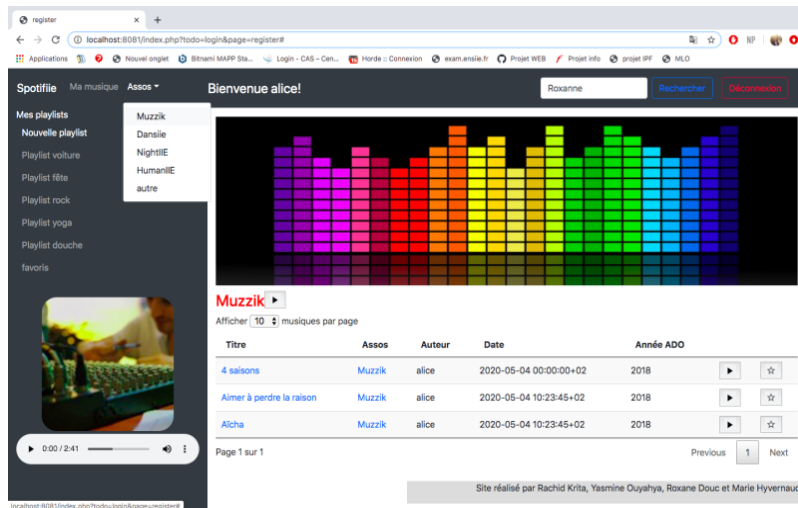


Figure 5: Ecouter une playlist

de la page, et sont constamment accessibles à l'utilisateur au cours de sa navigation. Le contenu à afficher est quant à lui chargé selon la commande dans l'espace désigné par `< main >`.

`< header >` de la classe navbar servant de volet supérieur

`< verticalFrame >` cadre du volet de navigation vertical, ayant une position fixe à l'écran

`< verticalMenu >` affichant les playlists, contenu lui-même dans un bloc de taille fixe, permettant de défiler au sein du bloc si la liste devient trop longue

`< container_player >` bloc de taille fixe contenant le lecteur principal, zone d'affichage de contenu, rechargée avec Ajax la plupart du temps

`< table >` affichage de la liste des musiques recherchées à l'aide de DataTable

5 Conclusion

Afin d'améliorer l'expérience Spotify, nous pouvons améliorer le player afin de pouvoir avancer ou reculer la lecture. De plus nous n'avons pas eu le temps de permettre aux utilisateurs de pouvoir importer de nouvelles musiques. Il serait également intéressant de pouvoir visualiser les clips associés aux musiques et d'offrir un système de notation (5 étoiles) des musiques pour ainsi créer des playlists "les plus populaires" en fonction des promos.