Nathan REYNAUD

Compte Rendu Projet PHP - Potify

Sommaire

Intro

(Rappel des enjeux du projet, conditions, etc...)

Présentation

(Présentation des dossiers et fichiers)

Travaux réalisés, problèmes et solutions

(Présentation des travaux réalisés, fonctionnement du site, etc...)

Conclusion

(Ce que m'a apporté le projet)

Annexe

(Recherches effectuées)

Introduction

Ce Projet PHP était à réaliser pendant les vacances et a consisté en la création d'un site web en PHP/HTML/JavaScript avec une partie front office et back office.

Notre site devait présenter une multitude de produits sur le thème de notre choix dont les informations étaient stockées en base de données Sqlite3.

Les produits présentés devaient avoir pour thème un sujet qui nous passionne.

Ce DM à pour enjeux de mettre en œuvre les TP PHP réalisés jusqu'à présent afin d'élaborer un site complet. Il permet de mettre en œuvre notre culture du langage PHP et demande de l'enrichir de nous même de part nos idées pour notre site nécessitant plusieurs recherches sur internet afin d'être abouties.

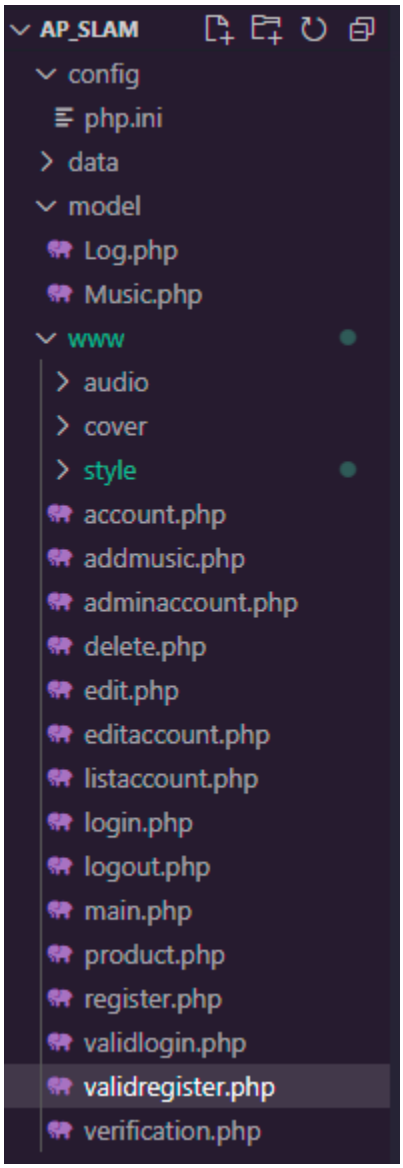
Présentation

Ayant grandi avec la musique et étant un passionné de cette dernière je décide d'en faire mon thème.

Je décide donc de réaliser un site web similaire à une plateforme de streaming ou les produits seront les musiques que l'on pourra écouter et voir les caractéristiques.

Organisation des fichiers

Voici l’organisation des fichiers de mon projet :



Dossier Config

Le dossier config est composé d'un seul fichier : php.ini qui correspond aux paramètres du langage PHP.

./model/php.ini

```
error_reporting=E_All
display_errors= On
```

```
upload_max_filesize = 20M
```

Dossier Data et Model

Les dossiers Data et Model sont utilisés pour stocker la base de données et les fichiers représentant les classes de mon projet.

Contenu de la base de données

Dès les premières minutes de travail, je commence à réaliser ma base de données.

Dans un premier temps je décide de faire deux base de données distinctes :

- L'une pour les comptes
- L'autre pour les produits, donc les musiques

J'avais décidé de faire ce choix afin de ne pas regrouper mes informations de connexion avec les produits pour des raisons de sécurité au cas où ma base serait récupérée par une personne malveillante.

Cependant le professeur nous oblige à réaliser une seule et même base de données. Donc mon choix ne marche pas je décide donc de regrouper les deux bases de données dans deux tables différentes :

- La table " log" pour tous les login
- La table "music" pour toutes les musiques

Il me faudra trouver une solution pour pouvoir sécuriser ma table log plus tard.

Table "music"

Ma table music est composée de quelques champs représentant les caractéristiques des musiques, elles étaient composées de :

- id, permettant de donner un identifiant unique à chaque musique.

- title, représentant le titre de la musique.
- time, son temps .
- artist, l'artiste de la musique.
- nol, (abréviation de number of listening) le nombre d'écoute de la musique (sur Spotify).
- genre, le genre de la musique.
- album, le nom de l'album.
- year, son année de parution.

Cependant je trouvais que le nombre de caractéristiques n'était pas suffisant, je décide d'ajouter les champs suivants dans ma table :

- nom, (abréviation de number of music) représentant le nombre de musiques présentes sur l'album.
- albumtime, représentant le temps de l'album.
- BPM, représentant le BPM de la musique.
- key, la note de la musique.
- interpreter, l'interprète de la musique.
- compositor, le compositeur de la musique.
- productor, le producteur de la musique.

Voici le code de création de ma base :

```
CREATE TABLE "music" (
	"id"	INTEGER NOT NULL,
	"title"	VARCHAR(255) NOT NULL,
	"time"	VARCHAR(10) NOT NULL,
	"nol" TEXT NOT NULL,
	"artist"	VARCHAR(255) NOT NULL,
	"genre"	VARCHAR(255) NOT NULL,
	"album"	VARCHAR(255) NOT NULL,
	"albumtime" VARCHAR(10) NOT NULL,
	"year"	INTEGER NOT NULL,
	"nom" INTEGER NOT NULL,
	"BPM" INTEGER NOT NULL,
	"key" VARCHAR(5) NOT NULL,
```

```
      "interpreter"     VARCHAR(255) NOT NULL,
      "compositor"      VARCHAR(255) NOT NULL,
      "productor" VARCHAR(255) NOT NULL,
      PRIMARY KEY("id" AUTOINCREMENT)
);
```

Table “log”

Ma table log est composé de 6 champs :

- id, un id unique pour chaque compte.
- username, le nom d’utilisateur du compte.
- mail, l’adresse mail du compte, obligatoirement unique.
- password, le mot de passe du compte.
- right, le droit du compte.
- registration_date, la date d’ajout du compte.

Chaque table possède un fichier classe pour passer les données en objets.

- Music.php
- Log.php

Dossier WWW

Le dossier www stocke tous les fichiers php avec lesquels l’utilisateur va interagir, un dossier style contenant les fichiers bootstrap (css et js) et mes fichiers css/js ainsi que le logo du site puis deux dossiers contenant les images de mes musiques et leurs fichiers audios.

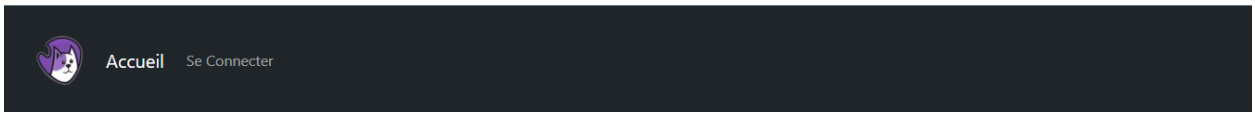
Travaux réalisés, problèmes et solutions

Lors de la réalisation du site j'ai repris comme exemples les fichiers des différents TP effectués en classe.

Visuel du site

Navbar

Pour donner un style visuel à mon site j'ai utilisé l'outil bootstrap qui m'a permis d'ajouter cette navbar ci-dessous à mon site :



dont voici le code :

```
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
    <div class="container-fluid">
      <a href="./main.php"><img src="./style/logo.png" alt="Logo du site"></a>
      <?php if (isset($_SESSION["username"])) { ?>
        <a class="navbar-brand" href="./main.php">Bonjour,
          <?php echo $_SESSION["username"] ?>
        </a>
      <?php } else {
        ?>
        <a class="navbar-brand" href="./main.php">Accueil</a>
      <?php } ?>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarSupportedContent"
        aria-controls="navbarSupportedContent" aria-expanded="false"
aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav me-auto mb-2 mb-lg-0">
          <li class="nav-item">
```

```
            <?php if ($_SESSION == null) { ?>
              <a class="nav-link" href="./login.php">Se Connecter</a>
            <?php } else { ?>
              <a class="nav-link" href="./account.php">Mon Compte</a>
            <?php } ?>
          </li>
          <?php if (@$_SESSION["right"] == "A") { ?>
            <li class="nav-item dropdown">
              <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown"
role="button" data-bs-toggle="dropdown"
                aria-expanded="false">
                Panneau d'administration
              </a>
              <ul class="dropdown-menu " aria-labelledby="navbarDropdown">
                <li class="list-group-item list-group-item-dark"><a
class="dropdown-item" href="./main.php">Editer
                    Musique</a></li>
                <li class="list-group-item list-group-item-dark"><a
class="dropdown-item" href="./adminaccount.php">Editer
                    Compte</a></li>
                <li class="list-group-item list-group-item-dark"><a
class="dropdown-item" href="./addmusic.php">Ajouter
                    Musique</a></li>
              </ul>
            </li>
          <?php } ?>
        </ul>
      </div>
    </div>
  </nav>
```

Cette navbar permet d'afficher un lien pour se connecter si la superglobal $_SESSION est vide, lorsque l'utilisateur se connecte sur la page prévue. $_SESSION récupérera le nom d'utilisateur, le droit du compte ainsi que l'id du compte récupéré depuis la table log. Ainsi lorsque l'utilisateur retourne sur la page principale, la page affiche son nom d'utilisateur en

le récupérant dans la session et en fonction de son droit d'utilisateur, la navbar affiche les options d'admins si l'utilisateur est admin sinon elle affiche un onglet pour aller sur la page du compte de l'utilisateur.

Les options d'admins sont :

- Editer une musique
- Ajouter une musique
- Editer les comptes

Background

M'inspirant du site Spotify pour mon site je voulais utiliser les mêmes couleurs c'est-à-dire principalement du noir et du vert, Téo m'avais proposé crée une image mais après reflexion, cela rendait mon site moins original.

C'est à ce moment que Robin me fit découvrir le site CSS Gradient qui permet de réaliser un gradient en css tout en donnant le code css nécessaire pour celui choisi. Je décide donc d'utiliser ce site tout en changeant de couleur, étant fan de la couleur violette je décide de basé mon gradient sur le noir et le violet.

Après avoir créé le gradient approprié pour mon site je décide de créer un nouveau fichier css plutôt que d'utiliser le css de Bootstrap et je défini le background des balises html et body pour mon site tout en définissant le background en no-repeat, la taille de la balise html ainsi que celle de la balise body pour que mon fond soit disposé sur toute la page.

Bouton

J'ai modifié le code css de Bootstrap afin d'ajouter des boutons de couleur violette avec la classe btn-purple. Cela permet de rendre certains boutons de mon site assortis au visuel de mon site.

Page d'accueil

Ma page d'accueil est le fichier main.php, cette page contient la liste des musiques contenues dans un tableau comme demandé dans les consignes du projet. Afin de pouvoir récupérer le contenu de ma table music je m'inspire des TP PHP, je récupère donc le contenu de ma table en effectuant une connexion à ma base pour effectuer une requête sql dont je stockerai le résultat dans une liste pour pouvoir l'afficher avec la classe Music contenant un constructeur des variables privées de la classe, leurs accesseurs et mutateurs ainsi que des fonctions statique que l'on utilisera pour la suppression, ajout et mise à jour d'une musique.

Afin de rendre mon tableau plus esthétique j'utilise Bootstrap afin de rendre ma table noire, pour qu'elle soit cohérente avec mon fond de page. L'utilisation d'une boucle foreach à permis d'ajouter des lignes à la table pour chaque élément présent dans la liste stockant le résultat de la requête sqlite.

Je décide de n'afficher que le titre, la durée, le nombre d'écoute, l'artiste, le nom de l'album, le genre et l'année sur mon tableau afin de garder des caractéristiques moins importantes mais précises à afficher sur la page du produit.

J'ai décidé de rendre chaque ligne de mon tableau cliquable afin de renvoyer vers la page product.php contenant toutes les caractéristiques de la musique. Afin de pouvoir réaliser cette étape j'ai réalisé un nouveau fichier javascript contenant un code permettant de renvoyer l'utilisateur vers la page product.php passant en GET l'id de la musique grâce à l'accesseur getId() du fichier Music.php.

modification apporté à la page main et aperçu du fichier js :

./www/main.php

```
      <tbody>
        <?php foreach ($musicList as $music): ?>
          <tr data-href="<?php echo "product.php?id=" . $music->getId() ?>">
```

./www/style/script.js

```
document.addEventListener("DOMContentLoaded", () => {
    const rows = document.querySelectorAll("tr[data-href]");

    rows.forEach(row => {
      row.addEventListener("click", () => {
        window.location.href = row.dataset.href;
      });
    });
  });
```

Si l'utilisateur connecté est un administrateur(ce que l'on vérifie en lisant la valeur de $_SESSION['right']) deux colonnes de plus apparaissent sur le tableau, chacune des colonnes propose une option, l'une permet de renvoyer vers la page de modification de la musique l'autre permet de supprimer la musique en revoyant vers la page nécessaire.

Un bouton apparaît aussi au-dessus du tableau permettant de rediriger vers la page d'ajout de musique.

Page de connexion/inscription

Les fichiers login.php et register.php représentent les pages de connexion et d'inscription.

login.php contient un formulaire afin de renseigner ses identifiants, l'autre permet de créer un compte. Afin que les saisies de l'utilisateur soit vérifiées dans la base de données ou ajoutées dans la base de données, j'envoie les formulaires sur les fichiers validlogin.php et validregister.php permettant de vérifier qu'un champs n'est pas vide, que les saisies soient bien valide en utilisant les fonctions php preg_match qui, dans mon code, permettent de vérifier à l'aide d'une expression régulière si chaque caractères de la saisie sont valides.

Si c'est deux conditions sont validées alors:

- le fichier login effectue une requête vers la base de données, vers la table log avec un "select * from log where username=? and password=?" Chaque point

d'interrogation est remplacé par les saisies du formulaire. Si le résultat de la requête n'est pas vide alors la connexion est autorisée et l'utilisateur est renvoyé vers la page d'accueil après un message confirmant la réussite de sa connexion. Dans le cas contraire, il est renvoyé vers le formulaire de connexion pour effectuer une nouvelle tentative de connexion.

- le fichier register.php effectue une connexion à la base de données et insère les valeurs renseignées par l'utilisateur (nom d'utilisateur, mail, password) dans la table log avec la requête sql suivante : "insert into log(username,mail, password, right,) values (?,?,?,?) où chaque point d'interrogation est remplacé par les saisies du formulaire. Le dernier ? prend la valeur de "U" afin de définir le droit du compte, il est donc Utilisateur. L'utilisateur à confirmation de la réussite de son inscription et est renvoyé vers la page d'accueil.

Chaque formulaire est protégé par les attaques par injection SQL, pour cela j'ai repris le TP vu en cours permettant d'éviter les injections, chaque " ' " est enlevé de la saisie.

Page de produit

Le fichier product.php contient à nouveau un tableau similaire à celui de la page d'accueil mais dont toutes les informations de la musique sont affichées. De plus, l'image de l'album et le fichier audio sont sur la gauche de la page et le tableau est sur la droite. Chaque image et fichier audio est nommée par l'id de la musique correspondante. Afin d'afficher la bonne image et le bon audio pour la musique, le fichier appelle l'image en utilisant l'id de la musique renseignée dans l'URL de la page, donc en GET.

Page de Compte

Le fichier account.php représente la page où, de la même manière que les tableaux précédant, les informations du compte de l'utilisateur sont affichées excepté le mot de

passe pour des questions de sécurité. Un bouton “Se Déconnecter” est aussi présent qui, lorsqu'il est cliqué permet de déconnecter l’utilisateur en supprimant le contenu de la session. Lorsque l’utilisateur clique dessus il est renvoyé vers la page de déconnexion (étant le fichier logout.php) qui affiche un message confirmant sa déconnexion et que l’utilisateur est redirigé vers la page d’accueil. C’est dans le fichier logout.php que les sessions sont supprimées et que l’utilisateur à confirmation de sa déconnexion.

Page de modification de musique

Le fichier edit.php représente la page de modification de musique uniquement accessible lorsque l’utilisateur est administrateur. J’ai récupéré le TP PHP 3 qui m’avait donné du fil à retordre afin de pouvoir afficher les valeurs dans les champs du formulaire permettant de modifier les caractéristiques d’un produit.

Ici la page est agencée de la même manière que la page produit mais le tableau est remplacé par un formulaire permettant de modifier les caractéristiques de la musique choisie. Sur la gauche de la page il est possible de modifier l’image et l’audio de la musique en autorisant l'utilisateur à faire deux input de type file. L’un accepte seulement les fichiers en .jpg et .jpeg, l’autre uniquement en .mp3.

Chaque input du formulaire possède un nom précis qui me permettra de récupérer les saisies dans des variables dans le fichier verification.php, pour ce formulaire toutes les saisies sont stockées dans une liste appelée values. Vu que le fichier verification.php me sert pour plusieurs formulaires, le fait de stocker les saisies dans une liste me permet de choisir quelle action faire selon la liste renseignée.

Lorsque l’administrateur envoie le formulaire, les saisies sont vérifiées et validées dans le fichier validation.php qui sert de vérification pour les ajouts de musique, les modifications de musique et les mises à jour de compte., puis une mise à jour de la musique est effectuée dans la table music en utilisant le fonction statique présente dans le fichier Music.php permettant d’effectuer une requête UPDATE sur la table music.

Le fichier validation.php vérifie si une image a été envoyé, si oui il supprime l'image de la musique à mettre à jour pour pouvoir ajouter l'image renseignée dans sur la page edit.php. De même pour le fichier audio.

Page d'ajout de musique

Le fichier addmusic.php permet à l'administrateur d'ajouter une musique à l'aide d'un formulaire. Ici, contrairement à la page edit.php, aucun id n'est renseigné en GET. Lors de l'envoi du formulaire le fichier verification.php vérifie si un id est renseigné via la méthode GET, si il n'est pas renseigné alors le fichier effectue un INSERT dans la table qui est dans la fonction statique AddMusic() du fichier Music.php puis fait une requête de l'id maximum de la table music afin de pouvoir stocker le résultat dans une variable qui sera utilisé pour renommer les fichiers par l'id de la musique qui vient d'être rajoutée avant de les stocker dans les dossiers cover et audio. La fonction LastInsertRowId() était une solution mais sachant que la fonction renvoie le dernier id renseigné dans la base et qu'il n'est pas possible de préciser une table précise j'ai préféré opter pour la solution utilisée dans le fichier.

Page de suppression de musique

Le fichier delete.php permet de supprimer la musique choisie depuis la page d'accueil lorsque l'utilisateur est un administrateur. Lorsque l'administrateur clique sur le lien supprimer de la page d'accueil, l'id de la musique est passé en GET vers la page delete.php qui vérifie si l'utilisateur est un administrateur, si oui le fichier appelle la fonction statique DeleteMusic() où une requête DELETE est effectuée en prenant l'id de la superglobale $_GET['id]

Gestion des comptes

Un administrateur peut gérer les comptes présents dans la table log. Le fichier adminaccount.php permet de lister tous les comptes présents dans un tableau avec la même méthode que les tableaux sur la page d'accueil et sur la page produit.

L'administrateur peut supprimer où modifier un compte. Il est le seul à pouvoir modifier le droit d'un compte.

Le fichier editaccount.php agit de la même manière que le fichier edit.php mais sur la table log, il vérifie, valide, appelle la fonction statique UpdateAccount() du fichier Log.php et met à jour la base de données.

Le fichier delete.php vérifie si $_GET['id'] est définie alors il supprime une musique, si $_GET['userid'] il supprime un compte.

Pareil pour la mise à jour, le fichier verification.php vérifie si la variable $values, étant un tableau récupérant les valeurs du fichier edit.php, n'est pas null alors il effectue les vérifications pour les formulaires des musiques. Si la variable $accountvalues n'est pas null alors il effectue les vérifications pour la modification de compte.

Sur la page adminaccount.php, le tableau agit de la même manière que le tableau de la page d'accueil. Lorsque l'administrateur clique une ligne du tableau il est renvoyé vers la page listaccount qui prend en paramètre l'userid stocké en session. La page permet d'afficher les informations du compte sélectionné.

Problèmes rencontrées

Lors du début du projet j'avais pour idée de récupérer le contenu d'un fichier csv de ma playlist spotify personnelle mais les valeurs du fichier csv étaient cryptées, j'ai donc décidé de renseigner les musiques à la main.

Impossible d’afficher la valeur du nombre d'écoutes avec un input type number car la valeur contient des espaces. J’ai dû passer l’input en text pour pouvoir afficher le nombre d'écoutes sur le formulaire de modification.

Idem pour l’input time sur la durée de l’album, les valeurs sont trop grandes j’ai donc dû passer l’input en text.

Pour la page d’accueil je voulais effectuer une pagination pour le tableau mais après plusieurs recherches et essaie j’ai décidé d'abandonner l’idée de pagination.

Il m’était impossible d’envoyer un fichier audio via un formulaire. Le $_FILES récupère uniquement le nom du fichier mais aucune autre info. Après avoir fait de nombreuses recherche je me suis demander si le fichier php.ini était pris en compte avec WAMP Server, après vérification WAMP Server possède un fichier php.ini, il m’a suffit de le feuilleter pour remarquer que la taille maximum des fichiers pouvant être envoyé était trop petite pour envoyer un fichier audio. Après modification de la taille, le problème a été résolu, mon site a été sauvé.

J’avais pour idée de créer un système de playlist pour chaque utilisateur mais, par manque de temps, je n'ai pas pu développer l’idée pour la date limite du projet. Je la rajouterai si le prochain projet de professionnalisation nous permet de continuer à travailler sur notre site.

Conclusion

Ce projet à permis d’utiliser les connaissances obtenues lors des TP PHP pour effectuer un site de A à Z. Personnellement, ce projet à permis de nettement améliorer mon aisance en PHP/HTML. Le fait de nous avoir laissé choisir le thème du site m’a permis de réellement

apprécier ce sur quoi je travaillais, le fait de travailler sur un sujet qui nous passionne donne envie de faire des recherches afin de réaliser le plus possible sur notre site.

En conclusion, n'aimant absolument pas le langage PHP au début des TP j'ai réussi à apprécier le langage et m'améliorer tout en développant mes connaissances sur les problèmes facilement rencontrés lors du développement d'un site utilisant une base SQLite tout en possédant un accès front et back office. Je ne regrette aucun instant passé sur la documentation sur php.net, sur stackoverflow ou encore à regarder des vidéos explicatives jusqu'à des heures improbables afin de chercher les solutions à mes problèmes ou les méthodes pour réaliser les idées que j'avais pour mon site. J'en ressors plus instruit, plus à l'aise sur le PHP/HTML et encore plus passionné par la programmation. Je ne doute pas que mes recherches me serviront lors de futur projet perso où même professionnel.

Annexes

Voici les recherches effectuées qui m'ont permis de faire avancer le développement de mon site :

PHP: foreach - Manual

<tr>: The Table Row element - HTML: HyperText Markup Language | MDN

Apical | Modifier la durée de vie d'une session

[Résolu] Supprimer SESSION après un certain temps par PtitSniper - page 1 - OpenClassrooms

supprimer dans session php - Recherche Google

oop - Store Object in PHP Session - Stack Overflow

HTML Color Codes - What's your color

Les sessions

9.3. Utiliser l'attribut scope pour associer les cellules aux entêtes des tableaux de données simples - AcceDe Web

Html how to make H1, H2, etc as links? - Stack Overflow

Bootstrap Input Sizing. Form controls · Bootstrap v5.0

Create a Data Table in Bootstrap 5 (2023)

Align H1 Header and Normal Text in Same Line - html

How to Force HTML Elements to Stay on the Same Line? - Designcise

HTML td headers Attribute

Center AND Right Align - Stack Overflow

https://www.youtube.com/results?search_query=table+row+clickable

Rename Image while uploading using phpinfo in PHP | Move uploaded File Function

Columns · Bootstrap v5.0

How can I prevent background position from moving/collapsing? - Stack Overflow

background-position - CSS: Cascading Style Sheets | MDN

Song BPM

Quelques autres sites dont j'ai oublié de sauvegarder le lien.