

Rendu final du Projet VidéoFlex de BDD

Par Agonsé Césaire et Bourennane Yacine

Voici notre version finale de notre projet finale. Il s'agit du site de streaming VidéoFlex qui permet de regarder tout un catalogue de film et séries avec des fonctionnalités croustillantes pour les différents utilisateurs avec un système de recommandation et de visionnage intelligent. Un site qui interagit avec une base de données et qui gère différents comptes. Le site est accessible juste à cette adresse : <https://etudiant.u-pem.fr/~cagonse/2020/site/index.html>

Toute ressemblance avec Amazon Prime serait purement fortuite.

Une copie des fichiers du site est disponible en fichier ci joint, ainsi que le dump de la bdd correspondante.

Pensez à personnaliser le fichier connexion.php pour vous connecter avec succès à la bdd.

Pour rappel voici nos anciens schémas qui nous ont servis pour notre projet:

Schéma relationnel :

CLIENT (idClient,mdp ,nomC, prenomC, dateFin, adresse, courriel,

Premium,Fils1,Fils2,Fils3,Fils4)

UTILISATEUR (idUtilisateur, pseudo, ageU, photoprofileU)

REGARDE (idUtilisateur, idOeuvre, dateVisionnage, dateDerVisionnage, timeCode)

NOTE (idUtilisateur, idOeuvre, noteUtilisateur)

SUGGESTION (idUtilisateur, idOeuvre, idLabel)

ARTISTE (idArtiste, nomA, prenomA, ageA, photoprofileA)

JOUE (idArtiste, idOeuvre, role)

OEUVRE (idOeuvre, nomO, estSerie, realisateur, date, durée, noteGlobale, idSaga, numOpus, synopsis)

APPARTIENT (idOeuvre, idLabel)

LABEL (idLabel, designation)

SAGA (idSaga, nomS, opus)

FK: Fils1,Fils2,Fils3 et Fils4 dans CLIENT font référence à idUtilisateur dans UTILISATEUR

FK : réalisateur dans ŒUVRE fait référence à idArtiste dans ARTISTE

FK : idSaga dans ŒUVRE fait référence à idSaga dans SAGA

PK : il y a une unique combinaison(idSaga,numOpus) dans SAGA

FK : idUtilisateur dans REGARDER fait référence à idUtilisateur dans UTILISATEUR

FK : idOeuvre dans REGARDER fait référence à idOeuvre dans ŒUVRE

FK : idUtilisateur dans NOTE fait référence à idUtilisateur dans UTILISATEUR

FK : idOeuvre dans NOTE fait référence à idOeuvre dans ŒUVRE

FK : idUtilisateur dans SUGGESTION fait référence à idUtilisateur dans UTILISATEUR

FK : idOeuvre dans SUGGESTION fait référence à idOeuvre dans ŒUVRE

FK : idLabel dans SUGGESTION fait référence à idLabel dans LABEL

PK : il y a une unique combinaison (idUtilisateur,idOeuvre,idLabel) dans SUGGESTION

FK : idArtiste dans JOUE fait référence à idArtiste dans ARTISTE

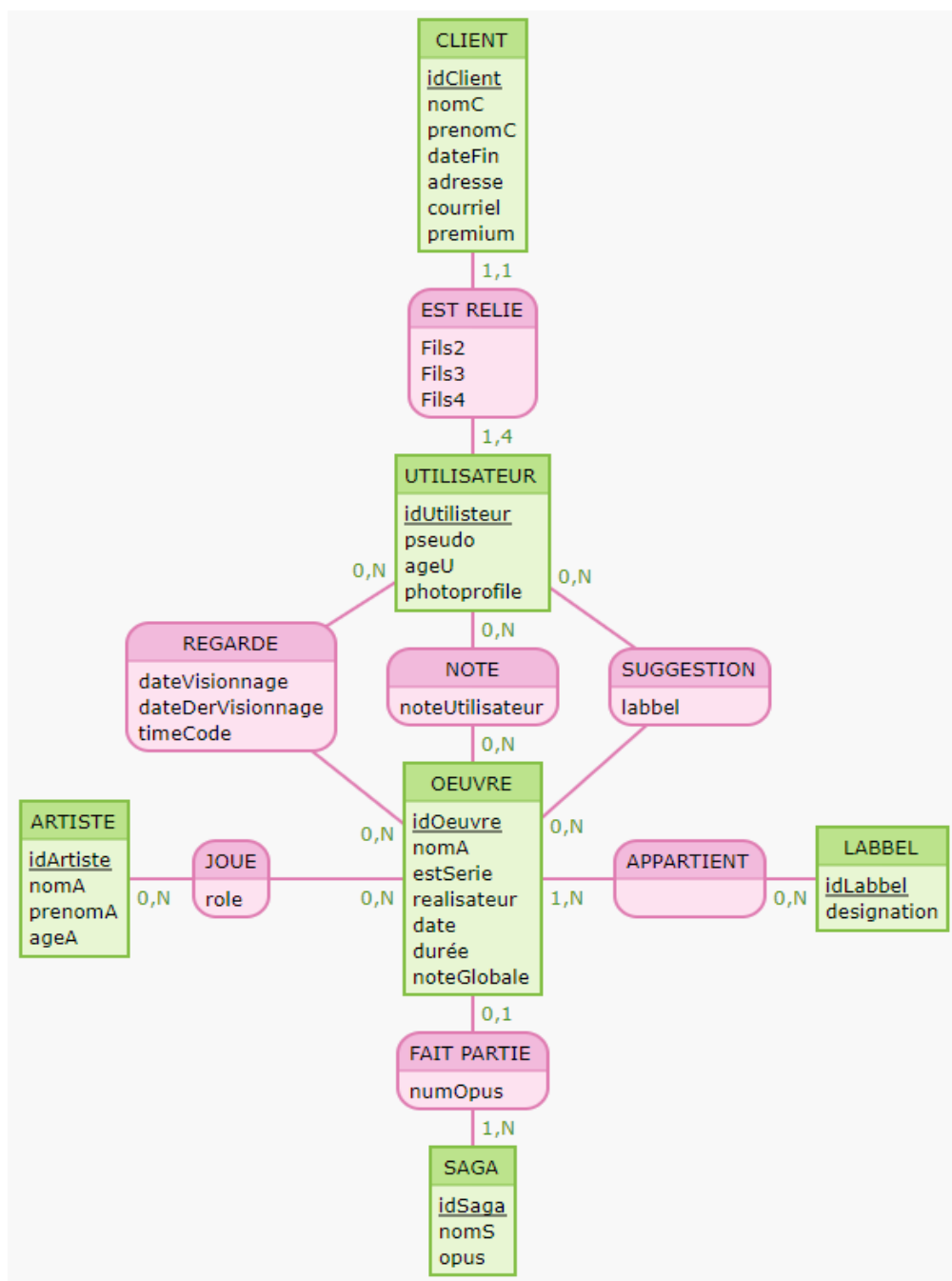
FK : idOeuvre dans JOUE fait référence à idOeuvre dans ŒUVRE

PK : il y a une unique combinaison (idArtiste,idOeuvre,roleArtiste) dans JOUE

FK : idLabel dans APPARTIENT fait référence à idLabel dans LABEL

FK : idOeuvre dans APPARTIENT fait référence à idOeuvre dans ŒUVRE

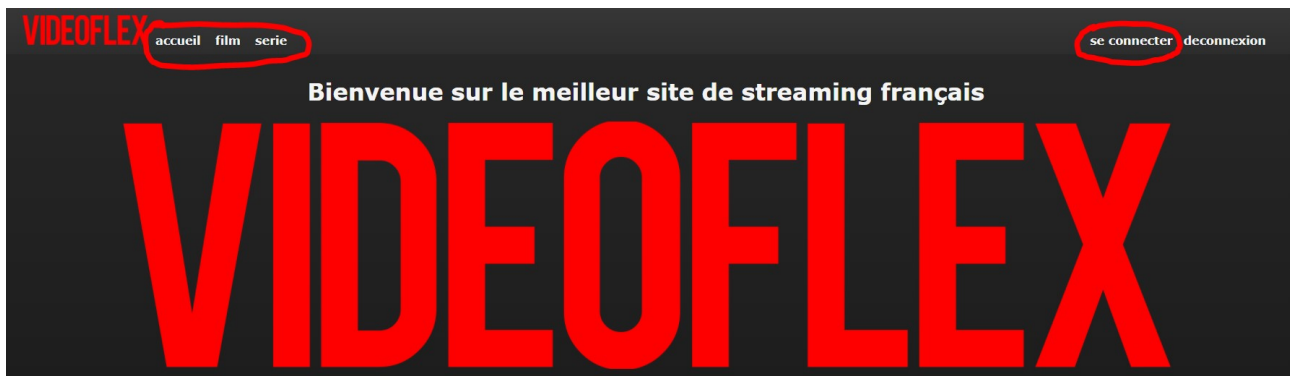
Schéma entité-association :



Manuel d'utilisateur :

Dans toute la suite du manuel, si l'action que vous avez faite n'a aucune incidence il faut actualiser la page pour que ça aie un effet.

Une fois sur le site internet (<https://etudiant.u-pem.fr/~cagonse/2020/site/>), la page d'accueil du site s'affiche. C'est seulement une page de bienvenue et le header du site s'affiche donc vous pouvez cliquer sur les différentes pages à afficher :

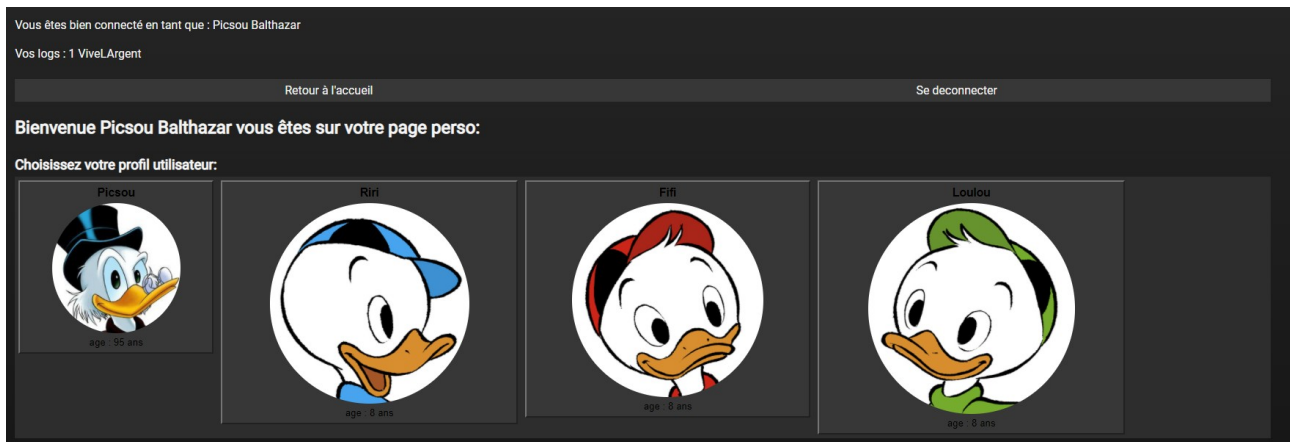


Sans compte connecté, le site n'affichera ni de films ni de séries (mais vous mettra un lien pour vous connecter), donc cliquons d'abord sur « se connecter ».

Sur la page suivante on doit donner un identifiant et un mot de passe. On peut aussi s'enregistrer si on ne possède pas de compte en cliquant sur lien correspondant. Si vous voulez, il y a deux comptes exemples qui sont présents : le premier compte étant un compte premium (composé de 4 sous-comptes maximum) et le deuxième non.

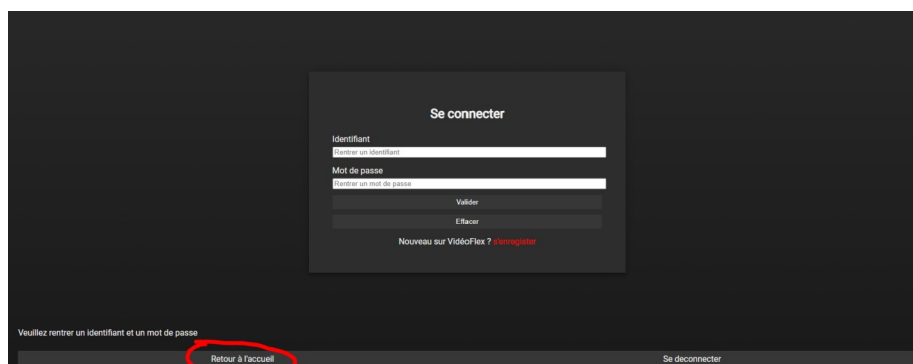
Première compte : id : 1 mdp : ViveLArgent
Deuxième compte : id : 2 mdp : RSA

Après avoir correctement rentré les id et mdp, vous pouvez confirmer la sélection de votre compte en cliquant sur le compte souhaité comme ci-dessous : (actualisez pour que la sélection prenne effet).

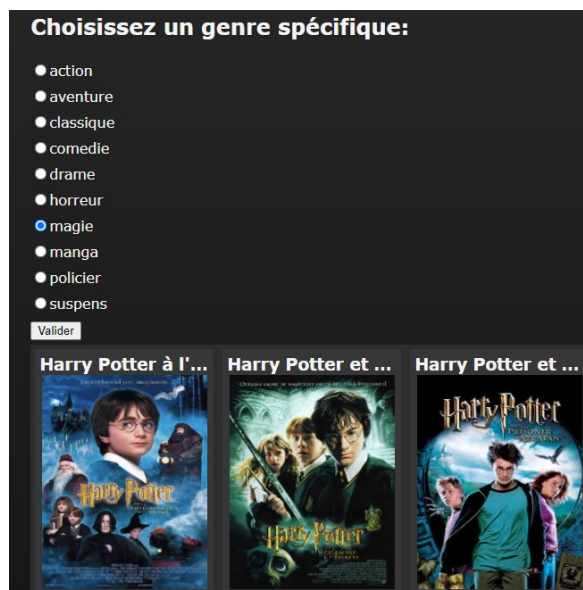
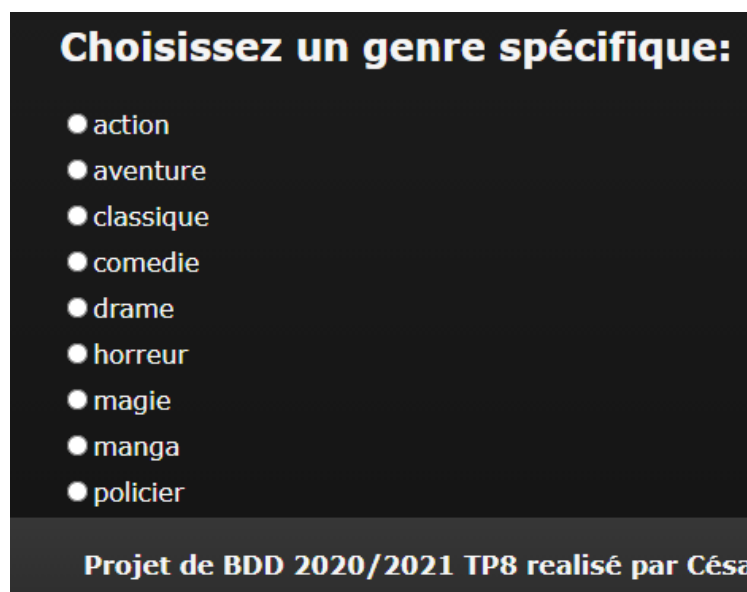


Une fois le compte sélectionné vous êtes sur la page perso, vous pouvez modifier vos données personnelles, vos id et mdp, renouveler un abonnement et regarder votre historique (seul le deuxième compte, donc celui qui n'est pas premium, possède un historique). Vous pourrez vous déconnecter en cliquant sur « se déconnecter » ou « déconnexion » sur le header et vous reconnecter sur la page qui suivra en allant en bas de la page et en cliquant sur « se connecter ».

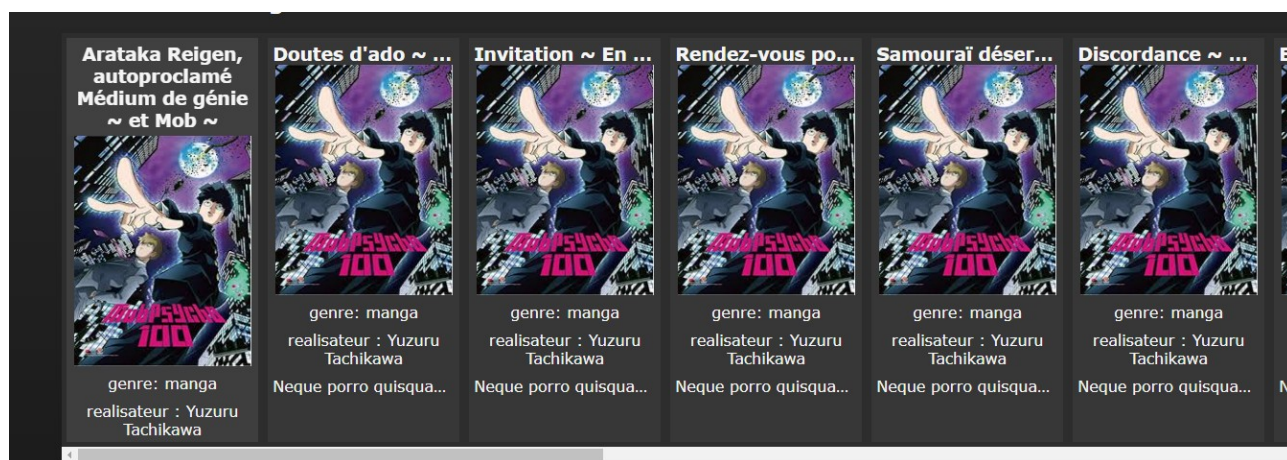
Vous pouvez donc aller sur les pages « film » et « série » en retournant d'abord à l'accueil :



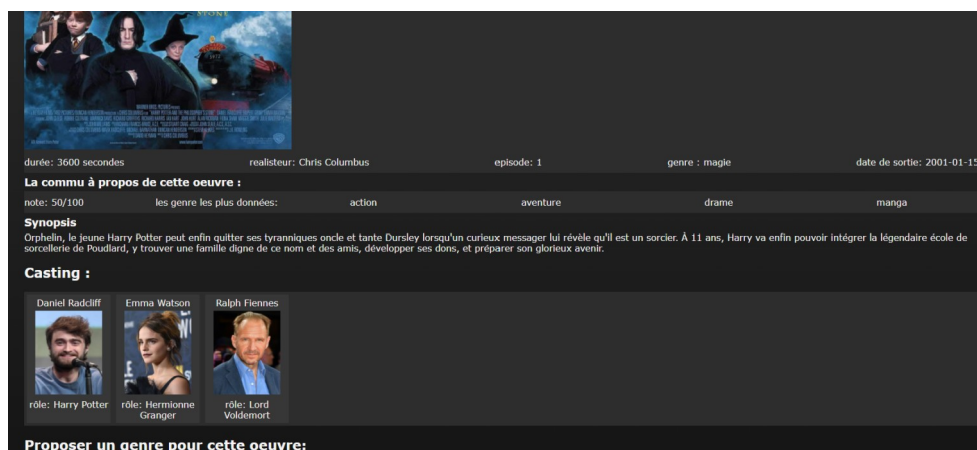
Si vous choisissez « film », vous allez voir tous les films disponibles et les films qui vous sont suggérés en fonction de vos précédents visionnages. Vous pouvez aussi sélectionner un genre pour voir les films qui correspondent au genre sélectionné (appuyez sur Entrée pour valider votre choix de genre) :



Pour une série c'est pareil, sauf que les séries sont classés par saison dans l'ordre d'épisode. D'ailleurs comme avec les films, passer le curseur de la souris sur le titre permet de le voir en entier :




En cliquant sur une œuvre, nous avons toutes les infos la concernant, dont son synopsis et son casting avec la possibilité de rajouter des genres en tant qu'utilisateur (les 5 genres les plus sélectionnés apparaîtront). Plus bas se trouve une liste de suggestions se trouvant dans la même saga et une autre liste possédant le genre de l'œuvre sélectionnée.



En cliquant sur un membre du casting, on a son âge et on peut voir quels films/séries il a réalisé et/ou dans quels films/séries il a joué :









Fiche de Daniel Radcliff :



age: 18 ans

A réalisé :

Joue dans :

Harry Potter à l'...	Harry Potter et ...	Harry Potter et ...	Harry Potter et ...	Harry Potter et ...	Harry Potter et ...	Harry Potter et ...	Harry Potter et ...
							
genre : magie realisateur : Chris Columbus	genre : magie realisateur : Chris Columbus	genre : magie realisateur : Alfonso Cuaron	genre : magie realisateur : Mike Newell	genre : magie realisateur : David Yates	genre : magie realisateur : David Yates	genre : magie realisateur : David Yates	genre : magie realisateur : David Yates

Manuel du programmeur :

Notre site est divisé en plusieurs grosses pages php. La page index.html les relie tous par le header, puis le header et le footer sont présent à chaque pages du site pour un confort d'utilisation avec une mise en page administrée par les différents fichiers css images présent dans le dossier public. Ainsi nous avons :

- login.php, logout.php, logininfo.php

Le fichier login.php est la page contenant « le plus de code ». C'est à dire que par ses nombreux formulaire d'inscriptions et d'authentifications, sa gestion des variables cookies, post et sessions il y a beaucoup de lignes en php et au final peu de balises html. La page cherche donc à savoir si les variables post sont présent, si oui alors en fonction des inscriptions ou de l'authentification il y a une mise à jour ou une insertion dans la bdd via une requête puis des définitions de cookies pour la session. Une fois tout initialisé et connecté le fichier inclus le logininfo qui fait globalement la même chose que login mais avec les utilisateurs. Le client peut également choisir de renouveler son abonnement pendant un an et de même en passant premium. Par manque de temps ceci marche mais le fait d'être premium ne donne aucun avantage à part perdre de l'argent en plus.

La page pense à gérer les différents cas d'erreur mais n'est pas parfaite. C'est pour ça qu'il y a de nombreuses lignes en commentaire pour déboguer.

Le client peut à tout moment se déconnecter via le logout.php, page qui nettoie tout les cookies de la session.

(Pour une raison inconnue le header n'est pas inclus sinon les cookies ne veulent pas s'initialiser heureusement il y a quand même un bouton pour revenir à l'accueil.

Pour que tout s'affiche correctement il est conseillé d'actualiser la page après chaque formulaire rempli).

- film.php, serie.php

Une fois connecté, l'utilisateur peut visionner le catalogue entier de film sur film.php, peut avoir une liste de suggestion via son historique avec les genres les plus appréciés. Et faire une recherche de film par genre. La page serie.php fonctionne exactement pareil mais n'affiche

uniquement les séries. Par manque de temps l'affichage n'est pas très optimisé pour afficher tous les épisodes d'une saison de manière groupée. Chaque œuvre affichée envoie au clic un lien vers œuvre.php avec l'id de l'œuvre grâce à la méthode \$GET et dans l'url.

- artiste.php, oeuvre.php, regarde.php

Dans œuvre.php, selon l'id qu'il y a dans l'url, toutes les infos relatives à l'œuvre en question sont affichées via une requête grâce à l'id en variable. Il y a ensuite le casting du film qui s'affiche. Et l'utilisateur peut choisir de proposer un genre à l'œuvre (Par manque de temps il ne peut pas proposer de note). Enfin il y a une liste de film du même genre qui s'affiche de la même manière que dans film.php et si jamais un film fait partie d'une saga, les autres films de celle-ci s'afficheront dans une liste ; de même si une saison contient plusieurs épisodes.

Si jamais l'utilisateur clique sur l'image du film dans œuvre.php, il se retrouvera redirigé vers regarde.php et pourra reprendre là où il en était si jamais il avait déjà regardé une partie via un timecode donné dans un formulaire. Commencera une nouvelle œuvre si l'utilisateur n'a jamais regardé et actualisera l'historique de l'utilisateur dans la bdd via une requête (ne marche pas pour l'insertion pour une raison qui nous est obscure).

La liste d'acteur dans le casting ainsi que le bouton réalisateur redirigent vers artiste.php avec une fois de plus l'id dans l'url comme pour œuvre.php avec film.php. Ainsi s'affiche une fiche pour l'artiste avec une liste d'œuvre réalisée et une autre avec toutes ses apparitions dans une œuvre.

Pour chaque requêtes sql nous utilisons connexion.php qui permet une authentification à la base de données si les bons identifiants sont déjà présents aux préalables. Ainsi une requête suivi de son affichage ressemble à ceci dans la plus part des cas. Pour plus de détails veuillez vous diriger vers les fichiers donnés ci dessus.

```
<?php
include "connexion.php";

$results = $cnx->query("SELECT idoeuvre,nomo,affiche,designation,prenom,noma,synopsis
FROM oeuvre
JOIN artiste ON oeuvre.realisateur=artiste.idartiste
NATURAL JOIN appartient
JOIN labbel ON labbel.idlabbel=appartient.idlabbel
WHERE estSerie IS FALSE;");

foreach ($results as $line) {
    ?>
    <a class="lst_film_element" href=<?php
        printf("oeuvre.php?idoeuvre=%s",$line['idoeuvre']);
        ?>>
        <h3 class="lst_film_element_titre"><?php
            printf("%s",$line['nomo']);
            ?></h3>
        <img class="lst_film_element_affiche" title="affiche" src=<?php
            printf("%s",$line['affiche']);
            ?>><img/>
        <p>genre:<?php
            printf(" %s",$line['designation']);
            ?></p>
        <p>realisateur : <?php
            printf("%s %s",$line['prenom'],$line['noma']);
            ?></p>
        <p class="lst_film_element_synopsis"><?php
            printf("%s",$line['synopsis']);
            ?></p>
    </a>
<?php
```

Difficultés rencontrées et répartition des tâches :

Notre projet n'est malheureusement pas optimal car pendant les vacances à cause d'un manque d'organisation couplé à un crash de serveur pendant les vacances ou nous ne pouvions pas accéder pendant plus d'une semaine aux fichiers. Mais à part ceci il n'y a pas trop eu de dysfonctionnement même si la logique du langage php et de sa mise en page assez brouillon dans le code nous échappait un peu par moment. Césaire a travaillé sur le php et les requêtes sql du site et Yacine sur la mise en page html ainsi que son css. Même si en général nous nous sommes entr aidé sur toutes les tâches et travaillons en même temps sur le même contenu. Nous avons fait l'erreur d'essayer de condenser « le plus de code » sur la page login.php ce qui provoque un code brouillon et l'obligation d'actualiser pour chaque formulaire. Malheureusement il nous manque des fonctionnalités implémentées. Nous trouvons également que nous avons pas mis assez d'exemple dans la bdd. Mais malgré tout, cela nous a permis de nous améliorer sur le domaine de la bdd et de la prog web et sommes fiers de notre rendu final.