

VIDÉOFLEX



Kishor Thiruvanandarajah
Prashath Sivayanama
TP:7

Mathématiques – Informatiques Année
Universitaire : 2020– 2021

TABLE DES MATIÈRES:

I – Présentation du site web et Documentation Utilisateur	page 1
II – Section Technique - structure & schéma faite à l'étape 1 et 2	page 2-3
III – Description Technique et général du site	page 4-6
IV – Organisation et répartition des tâche au sein du binôme	page 7

I – Présentation du site web et Documentation Utilisateur

Le lien qui permet d'accéder a la page : <https://etudiant.u-pem.fr/~prashath.sivayanama/>

Vidéoflex est un site web conçu en php et html grâce à une base de données postgresql, il s'agit d'un service de film en streaming qui exige une authentification.

Le site permet également de vous renseigner sur une recherche du film , la label , le nom acteur ou même le réalisateur.

Il vous sera possible de créer des profil en fonction de votre choix d'abonnement ,....

Lors du lancement du site "Vidéoflex, vous aurez le choix entre vous connecter si vous détenez déjà un compte ou vous inscrire.

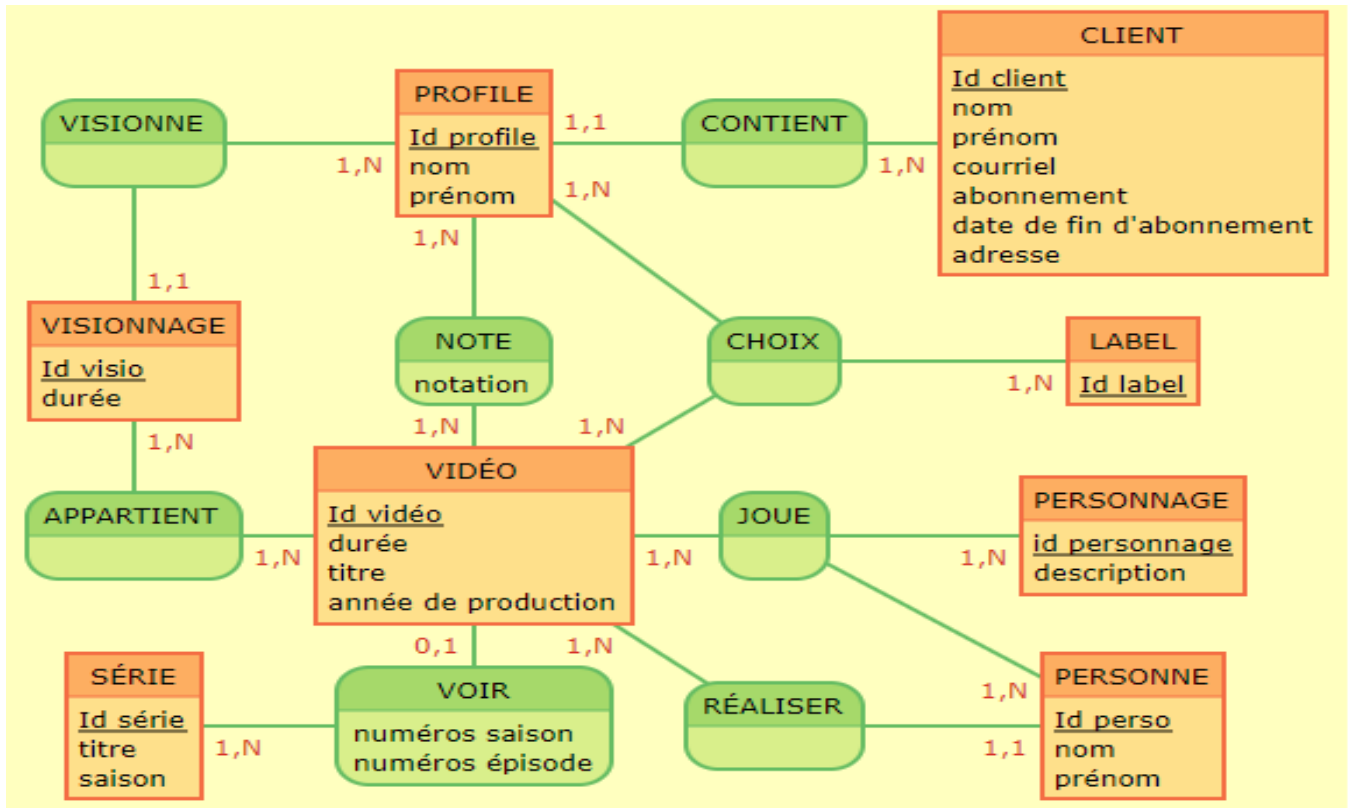
Une fois que vous êtes authentifiée, vous pourrez :

- Commencer à créer des profils ou de même supprimer des profils.
- Lancer une recherche concernant un label , le nom de l' acteur ou du réalisateur.
- Trouver des informations sur les films quand vous cliquerez dessus tel la description , la date de sorti ou de la durée du film .
- Regarder les films disponible à voir.
- Noter les film de 0 à 5 .

- Et reprendre le visionnage d'où vous êtes arrêtés.

II – Section Technique - structure & schéma faite à l'étape 1 et 2

Schéma entité association de Vidéoflex:



Les informations présentes dans le cahier des charges, nous avons établi deux versions de schémas l'entité-association dont voici la plus récente et le schéma relationnel qui sera la page d'après, ces schémas nous a servi à concevoir notre base de données.

Toute les entités sont reconnu par un identifiant unique c'est à dire une clé primaire qui est pour :

- Chaque client à un id_client, on connaît également son nom, son prénom, courriel, son abonnement, adresse, date de fin d'abonnement.
- Chaque vidéo à un id_vidéo, durée, titre, année de production.
- Chaque profil à son id_profil, son nom, prénom.
- Chaque personne à son id_perso, son nom et prénom.
- Chaque personnage à son id_personnage, description.
- Chaque série à son id_série, titre, saison.
- Chaque label à son id_label.
- Chaque visionnage à son id_visio, durée.

Le schéma relationnel ci-dessous a été conçu à partir du schéma entité-association vu précédemment.

Schéma relationnelle :

PROFILE (Id profile, nom, prénom, Id client)
CLIENT (Id client , nom, prénom, courriel, abonnement, date de fin d'abonnement, adresse)
VISIONNAGE (Id visio, durée, Id profile)
NOTE (Id vidéo, Id profile, notation)
CHOIX (Id label, Id vidéo, Id profile)
APPARTIENT (Id vidéo, Id visio)
VIDÉO (Id vidéo, durée, titre, année de production, Id série, numéros saison, numéros épisode)
JOUE (id personnage, Id perso, Id vidéo)
PERSONNAGE (id personnage, description)
SÉRIE (Id série, titre, saison)
PERSONNE (Id perso, nom, prénom, Id vidéo)
LABEL(Id label)

FK : **Id client** dans **PROFILE** fait référence à **Id client** dans **CLIENT** ;

FK : **Id profile** dans **VISIONNAGE** et dans **NOTE** et dans **CHOIX** fait référence à **Id profile** dans **PROFILE** ;

FK : **Id vidéo** dans **NOTE** et dans **CHOIX** et dans **APPARTIENT** et dans **JOUE** et dans **PERSONNE** fait référence à **Id vidéo** dans **VIDÉO** ;

FK : **Id label** dans **CHOIX** fait référence à **Id label** dans **LABEL** ;

FK : **Id visio** dans **APPARTIENT** fait référence à **Id visio** dans **VISIONNAGE** ;

FK : **Id série** dans **VIDÉO** fait référence à **Id série** dans **SÉRIE** ;

FK : **id personnage** dans **JOUE** fait référence à **Id personnage** dans **PERSONNAGE** ;

FK : **id perso** dans **JOUE** fait référence à **Id perso** dans **PERSONNE** ;

Nous avons ensuite conçu notre base de données SQL sur le schéma relationnel comme cela on pouvait savoir la clé primaire et étrangère .

Notre fichier .SQL contient dans ses première lignes afin d'éviter tout problème entre les bases de données la commande : «**DROP TABLE IF EXISTS nom_table CASCADE;** »

Cela revient à réinitialiser notre base de données si celle-ci existé déjà.

Ensuite dans le but de créer les différentes tables de notre base de données , pour cela nous avons utilisé la commande :« **CREATE TABLE nom_table (nom_col, type_donnee contrainte_col, ...);**»

nom_table: étant le nom de la table.

nom_co : nom de la colonne.

type_donnee : type de donnée.

contrainte_col : contrainte sur la colonne.

Pour chaque table, nous avons choisi le type de donnée pour chaque attribut qui correspondait à la fonction de l'attribut.

Par exemple : nom VARCHAR(25),

- Le nom est une chaîne de caractère, donc nous avons besoin d'un type de donné qui comprend les chaînes de caractère.

De plus nous avons ajouté la contrainte **NOT NULL** pour la colonne « nom , prénom » de la table client car des informations qui sont obligatoires ou même **UNIQUE** comme pour le mail car c'est unique à chacun .

Le **CHECK** qu'on a utilisé dans la table note qui permet de vérifier si la noté entre tel à tel note et pas d'autre .Pour toutes les tables, nous avons préciser les contraintes de clés primaires et étrangères.

III – Description Technique et général du site

Quand vous allez accéder à notre site ,vous allez apercevoir notre page d'accueil qui est une page d'inscription :

Inscription

Nom

Prenom

adresse électronique

adresse

abonné

Expiration

Si vous avez pas de compte vous remplissez le formulaire d'inscription et vous cliqué sur le bouton « Enregistrer » cela va vous dirigez sur une page nommée « insertion.php » c'est grâce à ce fichier php qu'on va utiliser pour insérer les données émis dans la base de donnée .

Pour cela on inclus dans ce fichier, un fichier.php nommée « connexion.inc.php » dans ce fichier j'utilise la fonction « PDO » qui permet de me connecter a ma base de donnée puis grâce la requête sql « **INSERT** » et la méthode « **Bindvalue** » en php (dans fichier « insertion.php ») permet d'insérer ses donnée à ma base

id_client	nom	prenom	courriel	adresse	abonnement	date_de_fin_abonnement
0	Chevalier	Robert	robert@hotmail.fr	28 avenue François mauriac ,75001 Paris	premium	2025-01-03
1	Dupont	Charle	Charle@hotmail.fr	15 rue de jean mermoz ,75000 Paris	normale	2025-01-17
2	Bureau	Martin	Martin@hotmail.fr	67 bis villebois mareuil ,75001 Paris	premium	2025-11-07
3	kishor	thiruvandarajah	kishor78@outlook.fr	28 avenue François Mauriac	p	2021-12-25

(4 lignes)

(vous la voyez dans la dernière ligne).

Sinon vous cliqué sur le bouton « Connecter » qui va diriger sur une page de connexion ce fichier ce nomme « connect.php » d'où vous mettez votre nom et votre mail :

Connexion

NOM

Adresse électronique

si vous avez le bon nom et le bon mail quand vous êtes connecté ça va diriger vers une page nommée « compte.php »sur cette page là vous allez créer vos profil entre 1 à 4 en fonction si vous avez choisis entre normale ou premium , sinon si vous rentrez le mauvais mail ou le mauvais nom , ou même avoir rien remplie dans le formulaire de connexion cela

vous dirigera sur la page nommée « connexion.php » qui vous affichera un message d'erreur .

Donc si vous êtes connecté ,vous accédez à la page « compte.php » c'est dessus que vous allez créer vos profil , dans ce fichier vous pouvez créer d'autre profil (si vous avez un abonnement premium)

Voici vos profils :

dans ce fichier on utilisera des requête tel que **SELECT** sur tout la table client et profil pour que « id_client » de profil et client soit égaux alors le nom est affichera dans la page « page_d_accueil.php » le site qui affichera les films, avec le nom en question au haut de la page et puis grâce au bouton « créer profil » qui va se diriger dans « inscription_profil.php » que vous remplirez dans le formulaire

Inscription profil

Nom

Prenom

et quand vous le validez ça se dirigera dans une page nommée « insertion_profil.php » qui va être insérer dans la base de donnée (comme dans «insertion.php»)

id_profil	nom	prenom	id_client
0	joseph	kévin	0
1	joe	zaz	1
2	kishor	prashath	2
3	cd	cdc	3
4	pedro	josé	4


(on le voit dans la dernière ligne)

Voici vos profils :

(voici le bouton ajouté)

Sur la page compte.php. il y aussi une bouton supprimer qui est un fichier «supprimer_un_profil_index.php , ce fichier permet de savoir quelle est id_client qui correspond à la table profil et à la table client

suppression de profil



si c'est le cas alors quand on clique sur le bouton supprimer elle va se diriger vers un fichier nommée «supprimer_un_profil »

profil supprimé

dans ce fichier on va prendre l'information que vous avez cliquer sur le bouton grâce \$_POST puis on utilise pour savoir si elle correspond a des prénoms de la table profil grâce un "SELECT *" sur notre table profil puis on compare et si c'est le cas alors on va créer une nouvelle requête sql qui va permettre de supprimer le prénom correspondant qui est : `UPDATE profil SET prenom = NULL WHERE prenom = :1` , cela va permettre de d'afficher NULL dans la colonne du prénom dans notre base donnée.

Donc dans le fichier « compte.php » vous cliquez sur un des profil créé et vous accéderez comme je l'avais dit précédemment dans « page_d_accueil.php » avec tout les film affiché

Bienvenue pedro voici les films disponible pour le moment :

Dans ce fichier on va reprendre les titre qui dans notre base de donnée grâce un "SELECT* " de la table vidéo et un foreach pour créer des bouton pour chaque film dans une même page , quand vous cliquez sur un des titre du film vous allez être dirigé dans la page vidéo.php d'où il y a la vidéo correspondant avec son description , durée et date de sortie du film et enfin une note attribué par le client

film



Titre : Morbius Durée : 00:03:19 Année de production : 2021-11-02
Leto Jared joue Morbius qui est un personnage de marvel et qui est un vampire et qui est le personnage centrale du film.

Note

et dans ce fichier il fait la différence entre l'acteur et le producteur avec id perso de la table personne , id video de vidéo, id personnage de personnage , et grâce a table joue qui groupe c'est 3 id on pourrait comparer les id pour différencier les réalisateurs et les acteurs et aussi les informations qui vont avec pour chaque film.

Quand vous êtes sur une vidéo vous pouvez reprendre d'où vous êtes arrêté mais il faut indiquer à quelle temps vous êtes arrêté (en seconde) puis ceci sera enregistré dans le fichier «sauvegarde_visionnage.php » et dans ce fichier on sélectionne la table appartient et visionnage et on utilise une commande sql: `UPDATE visionnage SET durée =:mp WHERE id_visio = :l AND id_profil=:ui;` ceci permet à l'utilisateur de voir à la durée qu'il veut.

IV – Organisation et répartition des tâches au sein du binôme

À propos du découpage du travail du projet, nous l'avons divisé en deux parties : • pour le programme, Prashath a fait la majeure partie du programme (60%) et Kishor (40%) ;

À propos des fréquences de rencontres, il était assez difficile de trouver des créneaux horaires entre nous deux du à nos emplois du temps et au dernier examen qui a été lieu cette semaine , de plus Kishor à un job étudiant donc les week end on pouvait pas faire ensemble le projet pour pouvoir se concrétiser, nous nous envoyons le programme par e-mail et se parler sur Zoom, quand on pouvait se voir entre temps à la fac on expliquait le code à l'un et à l'autre donc petit à petit le programme se faisait au fur et à mesure, ainsi que le rapport.

Dans le programme les parties où on a eu le plus de difficulté et prit plus de temps à comprendre , par exemple on avait du mal à insérer une vidéo le seul moyen qu'on a pensé c'était d'ajouter un attribut à notre table de base de données on a mis "lien_vidéo"

pour que ça fonctionne , puis on avait du mal à supprimer une donnée dans la table comme enlever le prénom dans la table profil avec quelque recherche et plusieurs essais sur le terminale de la base de donnée on a utiliser une requête qui permettait d'utiliser dans la suite du programme , il y avait la partie on a eu du mal car on pouvait arrêter un visionnage sur youtube donc le seul moyen qu'on a put faire c'est de demander à l'utilisateur d'indiquer à quelle temps elle s'est arrêter pour prendre cette information et comme dans notre base donnée la durée c'était un type TIME on l'a changé en type INT pour que vous indiquez uniquement en seconde pour que ça soit plus simple. .