

Movies ! | WIS3

Présentation

Movies! est un site de recommandation de films. Les utilisateurs n'y trouvent que des films de qualité, consultent des informations sur ceux-ci (dont les bandes annonces), laissent des critiques publiques, et peuvent ajouter des films à leur "liste de films à voir".

Un back-office permet aux administrateurs de gérer la base de données de films, de critiques et d'utilisateurs du site.

Matériel fourni

La base de données de films est fourni dans un fichier .sql, ainsi que les affiches (posters) des films.

Technos utilisées

- Serveur : Apache
- Base de données : MySQL
- Backend : PHP procédural ou PHP OO MVC ou Symfony2 , 3 ou 4 ou autres
- Framework CSS : as you wish
- Front-end : as you wish
- Gestion de versions : git

Infos sur la base de données

La base de données vous est fournie en 2 versions, contenant autant d'informations sur les films.

1. Une version bien structurée, suivant les bonnes pratiques, etc. mais plus difficile à utiliser. Des jointures sont rapidement requises.
2. Une version qui tient dans une seule table (movie_simple) : tout y est !

À vous de choisir la version qui vous convient, et de supprimer les tables inutiles.

Si vous choisissez Symfony et que vous avez peu d'expérience avec ce framework, utilisez la table movie_simple !

Conseils pour la mise en place du projet en PHP procédural :

1. Importer la base de données movie.sql par PHPMyAdmin
2. Supprimer les tables inutiles
3. Créer un nouveau dossier dans votre dossier web (www/ ou htdocs/)
4. Créer un nouveau projet dans votre éditeur de texte / ide
 - a. Créer un fichier index.php pour la page d'accueil
 - b. Créer un fichier de connexion à la base de donnée (db.php)
 - c. Créer un dossier css/ et y placer sa feuille de style
 - d. Créer un dossier img/ pour les images. Y placer le dossier posters/
 - e. Créer un dossier js/ pour les éventuels fichiers .js
5. Dans index.php
 - a. Créer une structure HTML complète
 - b. Relier la feuille de style

Étapes de réalisation

1. ACCUEIL

La page d'accueil présente une liste d'affiches de film. Il n'y a que les affiches qui sont affichées, pour un look bien graphique.

Chacune est cliquable, et mène vers la page de détails du film.

Par défaut, 50 affiches au hasard sont affichées.

Au bas de la liste des posters, un lien permet d'afficher 50 autres films au hasard. Il recharge simplement la page.

L'entête du site (commune à toutes les pages) affiche un logo (à vous de choisir sur le web) et le nom du site (Movies !). Ceux-ci sont cliquables et mènent vers l'accueil. Prévoir un emplacement pour un menu horizontal.

Le pied de page (commun à toutes les pages) du site affiche "©2017 Movies !"

Conseils

- Au haut du fichier index.php, connectez-vous à la base de données
- Réalisez une requête SELECT pour récupérer 50 films
- Un petit "ORDER BY RAND()" en SQL fait bien le travail pour récupérer les films
- Dans le <body>, faire une boucle foreach sur le résultat pour afficher les posters
- Le nom du fichier image est le champ 'imdbld' dans la table 'movie'.
- Prenez le temps de faire la base du css avant de passer à la page suivante

2. PAGE DE DÉTAILS

La page de détail d'un film affiche le poster, le titre, l'année, les acteurs, réalisateurs et scénaristes et toutes les autres informations publiques que nous possédons sur le film.

La bande-annonce (trailer) est à intégrer directement dans la page.

Vous pouvez aussi afficher un lien vers la page IMDB de ce film (nous avons leur id, à utiliser dans l'URL).

Vous pouvez aussi créer un lien vers la page de recherche d'un site de téléchargement ou de streaming... légal.

Conseils

- Créer un nouveau fichier detail.php à la racine de votre projet
- Vous avez besoin de l'id du film dans l'URL ! Voir ci-dessous.
- Récupérer ce film depuis la bdd avec une requête SELECT
- Pour récupérer les acteurs, vous devez réaliser une jointure avec entre les tables 'movie_actor' et 'people'. Même principe pour les réalisateurs et les scénaristes.
- Vous avez l'identifiant YouTube du trailer de chaque film. Pour savoir comment intégrer la bande-annonce sur votre site, analyser les codes fournis dans la section "Partage" d'une vidéo de YouTube !

Comment passer l'id du film dans l'URL ?

1. Sur index.php, ajouter l'id de chaque film dans l'attribut href de vos liens. Ceux-ci doivent prendre la forme suivante : ``
2. Rechargez bien la page index.php
3. Quand vous cliquez sur vos liens, vérifiez dans la barre d'adresse de votre navigateur que vous avez bien l'id dans l'URL
4. Sur detail.php, vous pouvez récupérer la valeur de l'id présente dans l'URL en utilisant : `$id = $_GET["id"];`

3. CRITIQUES DE FILMS

Sur la page de détail des films, afficher un formulaire. Celui-ci permet aux utilisateurs du site de publier des critiques sur le film.

Les critiques sont toutes affichées sous ce formulaire.

Le formulaire doit permettre de renseigner :

- Un titre pour la critique
- Un pseudo
- La critique elle-même

Tous les champs sont requis.

Conseils

- Créer une nouvelle table de bdd : review
- Traiter le formulaire directement au haut du fichier detail.php
- Si des champs ne sont pas remplis, afficher un message d'erreur dans le formulaire
- Sinon, sauvegarder la critique avec une requête INSERT
- Pour afficher la liste des critiques, faites une requête SELECT dans la table 'review', puis faites une boucle foreach sous le formulaire

4. FILTRES ET RECHERCHE

Un formulaire devrait être présent sur la page d'accueil (ou dans l'entête de toutes les pages) pour permettre à l'utilisateur de filtrer les films.

Ce formulaire devrait permettre les 2 filtres suivants :

- Catégorie de film (liste déroulante)
- Année min/max (2 listes déroulantes)

Un champ devrait également permettre de réaliser une recherche par mot-clef. La recherche devrait s'effectuer sur le titre, les acteurs et les réalisateurs.

Les résultats de recherche s'affiche exactement comme la liste des films de la page d'accueil.

Les options sélectionnées dans le formulaire devraient être présélectionnées lors de l'affichage des résultats.

Conseils

- Les filtres et les mot-clefs ne doivent pas exécuter une requête SQL complètement différente : c'est la requête SQL de l'accueil qui doit être altérée en fonction des critères envoyés
- Utiliser la page d'accueil pour l'affichage des résultats, c'est pareil.
- Pour la recherche par mot-clef, entourer le mot-clef par des pourcentages dans la requête SQL (joker).

5. INSCRIPTION DES UTILISATEURS

Cette page permet aux internautes de se créer un compte sur le site. Elle affiche un formulaire comportant les champs suivants :

- Email
- Pseudo
- Mot de passe
- Confirmation du mot de passe

Lors de la soumission du formulaire, plusieurs validations sont réalisées :

- Tous les champs doivent être renseignés
- L'email doit être valide et pas déjà inscrit
- Le pseudo doit être unique
- Le mot de passe doit avoir au moins 6 caractères
- Le mot de passe et sa confirmation doivent être identiques

Un lien doit être affiché dans le menu du haut pour mener à cette page d'inscription.

Conseils

- Créer une nouvelle table de bdd : users (ne pas utiliser 'user' au singulier comme nom de table)
- Créer une nouvelle page : register.php
- Traiter le formulaire au haut de cette page
- Pour vérifier si l'email et le pseudo sont déjà utilisés, faites 2 requêtes SELECT
- Pour vérifier si l'email est valide, utilisez la fonction filter_var()
- Si le formulaire est valide, ajouter l'utilisateur à la bdd avec une requête INSERT

6. CONNEXION DES UTILISATEURS

Une fois inscrits, les utilisateurs doivent pouvoir se connecter sur le site. Afficher une nouvelle page "connexion" permettant de renseigner dans un formulaire :

- Son pseudo ou son email
- Son mot de passe

Si les pseudo et mot de passe existent et concordent, connecter l'utilisateur.

Une fois connecté, afficher un indicateur clair montrant à l'utilisateur qu'il est connecté.

Créer également un bouton de déconnexion, permettant de... se déconnecter.

Conseils

- Pour connecter un utilisateur, il suffit de stocker une information quelconque dans la variable de session `$_SESSION`
- On y stocke souvent toutes les informations de l'utilisateur sous la clef "user"
- Pour le déconnecter, il suffit de supprimer ces informations de la session

7. LISTE DES FILMS À VOIR

Lorsqu'un utilisateur est connecté, il devrait avoir la possibilité d'ajouter des films à sa "watchlist".

Un bouton doit être affiché sur la page de détail d'un film. Ce bouton permet d'ajouter un film à sa watchlist OU de le retirer si le film est déjà présent.

Un nouvel item de navigation devrait mener à la page de watchlist. Visuellement, cette page affiche les films de la watchlist de la même manière que sur la page d'accueil.

Il est aussi possible d'afficher le bouton "ajouter/retirer de la watchlist" sur chaque poster dans la liste des résultats en page d'accueil (un peu difficile à faire) et de la watchlist.

La page "watchlist" ne devrait être accessible que par les utilisateurs connectés. De même, les boutons ne doivent pas apparaître si l'on est anonyme.

Conseils

- Avant de se lancer, il faut bien réfléchir à la structure de base de données nécessaire pour stocker cette relation de watchlist.
- Pour pouvoir stocker le fait que tel utilisateur veut voir tel film, vous avez besoin de l'identifiant du film et de l'identifiant de l'utilisateur connecté.
- Pour savoir si un utilisateur est connecté, il suffit ensuite de regarder si des informations sont présentes dans la \$_SESSION

8. UN PEU D'AJAX DANS TOUT ÇA (très optionnel)

Il y a 2 facettes du site qui serait nettement améliorées avec de l'AJAX, sans que ce soit une tâche trop difficile à mettre en place ou à maintenir.

1. Au lieu du bouton en page d'accueil qui permet de recharger de nouveaux films, il serait intéressant d'implémenter un "infinite scroll", comme dans Google Images ou sur Facebook.
2. Les boutons d'ajout ou de retrait à la watchlist pourrait également faire leur requête en AJAX.

9. BACK-OFFICE

- Gestion des films
- Gestion des utilisateurs
- Gestion des reviews
- Statistiques

