

Margaux BRULIARD - Nahida BENHAFFAF

DODOCINÉ

PROJET PROGRAMMATION WEB



20 Mai 2021

TABLE DES MATIÈRES

Table des matières	1
Introduction	2
Présentation du projet	2
1 Architecture du site web	3
1.1 Mise en place de la base de données	3
2 Accueil et Navigation	4
2.1 Les composants de la page d'accueil	4
2.2 La barre de navigation	4
2.2.1 mise en place d'une recherche auto-complétée	4
3 Accès utilisateurs	5
3.1 Inscription ou Connexion sur DodoCiné	5
3.2 L'espace membre associé à un compte	6
3.2.1 Rappels des notes et des messages postés	7
3.2.2 Paramètres du compte	7
4 Présentation des films	9
4.1 Présentation des films disponibles sur le site	9
4.2 Page de présentation d'un film	9
4.2.1 Construction du système de notation des films	9
4.2.1.1 Mise en place du système de notation pour l'utilisateur	9
4.2.2 Fonctions supplémentaires :	9
4.2.3 Gestion du forum relatif au film	10
4.2.4 Mise en place des messages dans le forum	10
4.2.4.1 Répondre à un message ou le supprimer	10
Conclusion	11

INTRODUCTION

Présentation du projet

De nos jours, l'arrivée en masse des plateformes de "streaming" sur Internet a permis de populariser l'accès aux films de tous types et de toutes générations. Afin d'aiguiller et d'aider les cinéphiles en herbes que nous sommes, nombre de sites nous aident aujourd'hui à découvrir des films qui correspondent à nos préférences. Parmi ces sites, nous retrouvons notamment [le site Allociné]<https://allocine.fr> qui présente des films ainsi que les acteurs et les réalisateurs y ayant participé.

Dans le cadre du module **Programmation Web** de la licence d'Informatique de l'Université Sorbonne Paris Nord, nous nous sommes penchés sur le fonctionnement de ce type de site web et avons choisi de construire un "mini-clone" du site d'Allociné, que nous appellerons en cette période de pandémie **DodoCiné** afin de parfaire nos connaissances sur la programmation web avec les langages de programmation PHP¹ et Ajax côté serveur et les langages HTML5/CSS3/Javascript pour le côté client.

Mise en production

*Il est possible de visualiser notre site **DodoCiné** directement en cliquant sur ce lien : <http://dodocine.infinityfreeapp.com/>.*

Cependant, notre hébergement étant gratuit, il est à noter que la protection de la base de données n'est pas assurée. En conséquence, nous ne vous conseillons pas d'utiliser votre véritable adresse email si vous souhaitez créer un compte

accès au github

Vous pouvez retrouver l'ensemble de nos fichiers sources dans notre repository Git : <https://github.com/MBruliard/dodocine>

1. nous utiliserons ici la version de PHP7

1. ARCHITECTURE DU SITE WEB

Pour ce projet, nous avons essayé de tirer parti du dynamisme de PHP en créant de multiples fichiers annexes qui seront appelés de multiples fois dans différentes pages. Ces fichiers sont regroupés dans le dossier *elements*. On y retrouve notamment des parties de codes réutilisées pour chaque page de notre site

- *elements/head.php* contient la balise 'head' que nous rappelons pour chaque page de notre site. Nous lui avons ajouté deux variables, *\$title_page* qui permet de changer le titre de la page au besoin ainsi que *\$css_addon*, qui prend une chaîne de caractères pour ajouter des liens vers des fichiers css supplémentaires.
- *elements/footer.php* contient le footer, répété dans chacune de nos pages.
- *elements/js_files.php* appelle l'ensemble des scripts Javascript/JQuery dont nous avons besoin. De la même façon que pour *elements/head.php*, nous avons construit une variable *\$js_addon* pour ajouter des scripts supplémentaires dans certaines pages.
- *elements/navigation.php* la barre de navigation

1.1 Mise en place de la base de données

Nous avons fait le choix de créer des scripts PHP contenant l'ensemble des fonctions qui interagiront avec la base de données via le système PDO de PHP. Ces scripts sont regroupés dans le dossier *controller* :

- *controller/authentication.php* pour les fonctions relatives aux utilisateurs
- *controller/datafilms.php* pour les fonctions relatives aux films et à la notation de ceux-ci
- *controller/dataindividus.php* pour les fonctions permettant d'obtenir des informations sur les acteurs/réalisateurs contenus de la base de données.
- *controller/dataforum.php* pour la gestion du forum.

Pour ce projet, nous avons fait le choix d'utiliser le système de gestion de base de données nommé SQLite et non pas MySQL, principalement afin de pouvoir avoir un hébergement gratuit de notre site pour sa présentation. Cependant, si SQLite est adaptée pour un projet universitaire, il engendre des ralentissements de navigation lorsque la base de données est imposante et lorsque le nombre de visiteurs sur le site est grand.

On remarque, que dans l'écriture de requêtes SQL avec PDO, SQLite ferme automatiquement la connexion à la base de données contrairement à MySQL.

2. ACCUEIL ET NAVIGATION

Commençons par vous présenter la page centrale de notre site : la page d'accueil. Il s'agit de la page que les visiteurs voient en premier c'est pourquoi nous l'avons construite de manière épurée et fonctionnelle.

TODO: image de la page d'accueil

2.1 Les composants de la page d'accueil

On souhaite que, dès son arrivée, le visiteur puisse avoir un aperçu des derniers films présentés sur le site. Pour ce faire, nous utilisons un "carousel", c'est-à-dire une bannière animée, pour les mettre en avant.

Afin de simplifier son inclusion et son interaction avec l'ensemble de la page, nous avons fait le choix d'utiliser l'élément "carousel" intégré dans le framework Bootstrap, et qui gère de façon automatique les actions dynamiques autour du "carousel".

2.2 La barre de navigation

2.2.1 mise en place d'une recherche auto-complétée

3. ACCÈS UTILISATEURS

Afin de filtrer et maîtriser les informations circulant sur le site, ainsi que fidéliser les visiteurs les plus assidus, nous construisons un espace membre permettant aux visiteurs de créer et/ou de se connecter pour avoir accès à des fonctionnalités supplémentaires du site.

3.1 Inscription ou Connexion sur DodoCiné

Les visiteurs du site peuvent se connecter à tout moment, en utilisant le bouton **Connexion** situé à droite dans la barre de navigation.

TODO: image page connexion/inscription

Si le visiteur ne possède pas encore son propre compte, il peut s'inscrire en se rendant sur l'onglet **S'inscrire** et remplir un formulaire d'inscription. Pour plus de confidentialité, on demande au futur membre de choisir un pseudonyme : ce pseudonyme servira à l'identifier sur le site tout au long de sa visite auprès des autres membres du site.

Fichiers utilisées pour cette partie du site

- *login.php* qui est la page appelée lorsque l'on clique sur le bouton "Connexion"
- *elements/login-form.php* et *elements/signup-form.php* contiennent les formulaires de connexion et d'inscription. Nous avons fait le choix de les écrire dans des fichiers séparés avant d'améliorer la visibilité de notre fichier *login.php*.
- *static/css/login.css*
- *static/js/login.js* est le script JQuery/Javascript permettant d'ajouter du dynamisme à la page.
- *login-response.php* contient le script PHP "pur" qui répond aux demandes Ajax du fichier *static/js/login.js*

Présentation de la page

En nous inspirant des dernières tendances, nous avons construit une unique page dans laquelle il est possible au choix de se connecter ou de s'inscrire. Le passage d'un formulaire à l'autre se fait via une fonction JQuery qui, en fonction du bouton sélectionné cache un formulaire et fait apparaître le second.

Vérification des informations renseignées dans les formulaires

Afin d'offrir une meilleure expérience aux utilisateurs du site, nous avons fait le choix d'effectuer toutes les vérifications d'informations à l'aide de Ajax et d'un script PHP. Lorsque l'utilisateur appuie sur le bouton **S'enregistrer** ou **S'inscrire**, nous effectuons un appel Ajax qui appelle à son tour notre script PHP *login-response.php* par la méthode POST. A l'intérieur de ce script, nous vérifions tout d'abord si tous les champs sont bien renseignés puis si les informations entrées sont correctes vis-à-vis du contenu de la base de données (en particulier la table **users** ici). Si des erreurs surviennent, celles-ci sont renvoyées afin que l'on puisse afficher une indication à l'utilisateur. A l'inverse, en l'absence d'erreur l'utilisateur est connecté et est renvoyé sur la page d'accueil du site.

TODO: images des formulaires avec et sans erreur

Une fois connecté, le visiteur peut remarquer que son pseudonyme remplace le bouton de connexion dans la barre de navigation. En cliquant dessus, il peut alors accéder à son espace personnel¹ ou tout simplement se déconnecter.

Problèmes rencontrés lors de la programmation :

- Suite à l'utilisation d'AJAX pour passer nos requêtes, nous constatons que l'utilisation de la fonction "header('Location : index.php')" ne fonctionne plus. Après recherches dans la documentation de PHP, nous avons découvert que cela était dûe au fait que cette fonction ne fonctionne que si aucun code html n'est encore affiché sur la page. Pour effectuer cette redirection, nous choisissons dorénavant d'utiliser la méthode Javascript "window.location.replace(url_redirection) ;"
- Lors de l'écriture d'un script JQuery, l'utilisation d'un argument *event* dans la fonction `$(document).ready(function(event) {})` ; nécessitait l'appel de la méthode *event.preventDefault()* ; afin d'empêcher le rechargement automatique de la page. Cependant, si *function* ne prend aucun argument, alors l'utilisation de la méthode n'est plus indispensable et la page ne se recharge pas.

3.2 L'espace membre associé à un compte

TODO: ajouter une image de la page

Dans cet espace, l'utilisateur peut modifier les paramètres de son compte, tel que changer son mot de passe, son email, ou même se désinscrire (entraînant la suppression de son compte), ou bien accéder plus rapidement aux messages et aux notes qu'il a posté sur le site. Il peut également modifier l'apparence du site pour son utilisation personnel, en activant le "mode Nuit".

1. voir plus loin

Fichiers utilisés pour la partie 'espace membre'

- *parameters.php* qui est la page principale.
- *architecture des onglets de la page* :
 - *parameters-config.php* pour modifier ses informations personnelles
 - *parameters-looking.php* l'onglet Apparence
 - *parameters-rating.php* est l'onglet rappelant à l'utilisateur les notes qu'il a données à différents films ainsi que ses derniers messages dans différents forum.

Chacun des onglets est caché/affiché à l'aide de fonctions définies dans le script JQuery/Javascript.

- *static/css/parameters.css*
- *static/js/parameters.js* est le script JQuery/Javascript permettant d'ajouter du dynamisme à la page.
- *parameters-response.php* contient le script PHP "pur" qui répond aux demandes Ajax du fichier *static/js/login.js*

3.2.1 Rappels des notes et des messages postés

Nous proposons à l'utilisateur, dans l'onglet "mes notes et messages", de retrouver quelques statistiques sur ces interventions sur le site :

- le nombre de messages postés
- le nombre de films notés
- la moyenne des notes données

De plus, il peut voir les derniers messages qu'il a posté et avoir un lien direct pour rejoindre le fil de discussion et continuer à échanger avec d'autres utilisateurs.

Les messages sont présentés dans un tableau, avec le début du message ainsi que le film auquel se rapporte le message. En cliquant sur le nom du film, l'utilisateur retourne directement sur sa fiche et a accès au forum. Nous faisons le choix de n'afficher que les 5 derniers messages et ce pour éviter, le cas échéant, l'affichage d'un nombre trop important de messages.

3.2.2 Paramètres du compte

Dans cette section, nous avons créé deux formulaires distincts, l'un permettant de modifier le mot de passe du compte et le second, l'email. Nous avons construit ces formulaires et les appels associés de la même façon que pour la connexion/inscription d'un utilisateur : à l'aide d'Ajax, nous vérifions les erreurs potentielles entrées par l'utilisateur et les affichons le cas échéant. En cas de succès, une fenêtre modale apparaît pour indiquer que les informations ont bien été modifiées.

Cette fenêtre modale est un composant intégré du framework Bootstrap, ce qui nous permet de l'utiliser de façon simplifiée sans s'inquiéter de la partie Javascript qui est pré-programmée. De plus, ces "modals" sont très facilement modulables ce qui les rends très simples d'utilisation.

Enfin, le bouton **Suppression** permet à l'utilisateur, s'il le souhaite, de supprimer de façon définitive son compte. En cas de sélection, le compte est supprimé, et les informations concer-

nant cet utilisateur seront supprimées. Néanmoins, les notes données, ainsi que les messages postés sont conservés car devenus publics.

4. PRÉSENTATION DES FILMS

4.1 Présentation des films disponibles sur le site

On propose sur cette page de retrouver tous les films disponibles sur la plateforme. L'utilisateur peut ainsi rechercher le film qu'il désire, en tapant directement le nom du film dans la barre de recherche, ou plus simplement en parcourant les différentes listes présentées.

Chaque liste de films (présentée sous la forme d'un "slider carousel"¹) correspond à une catégorie de films.

Ou alors là je viens de trouver un framework qui s'appelle <https://flickity.metafizzy.co> qui à l'air vraiment top alors on va essayer. Ce framework est en "open-source" (licence GPLv3) pour les projets personnels et universitaires ce qui nous convient parfaitement.

4.2 Page de présentation d'un film

4.2.1 Construction du système de notation des films

2 fonctionnalités : Afficher la note du film donné par les utilisateurs ET faire noter le film par l'utilisateur

4.2.1.1 Mise en place du système de notation pour l'utilisateur

On utilisera un système de notation allant de 1 à 5 pour noter les films et sera visualiser par un système d'étoiles.

Les étoiles ici proviennent du framework "Font Awesome".

Mise en place de AJAX : deux formats de retour possibles : XML ou JSON. Comme le XML est généralement plus lourd et plus compliqué à manier, on préfère utiliser ici le JSON.

Ici on a simplement, une méthode POST qui renvoie 3 choses : l'ID de l'utilisateur, l'ID du film à noter et la note. On pourrait presque simplement renvoyer du HTML ou passer directement les arguments dans la fenêtre

4.2.2 Fonctions supplémentaires :

On propose à l'utilisateur des films en lien avec le film sélectionné :

- un film de même catégorie
- un film fait par le même réalisateur
- un autre film dans lequel a joué un des acteurs

Pour cela on va utiliser des requêtes SQL (via PDO de PHP) pour récupérer les résultats puis nous tirerons au hasard un des ses résultats pour l'afficher à l'aide d'une fonction incluse dans PHP : "array_rand"

1. Pour la mise en place du "slider carousel", nous nous sommes largement inspirées du code source présenté par **WebShalla** dans [cette vidéo](#)

4.2.3 Gestion du forum relatif au film

Dans ce projet, on considère que les utilisateurs connectés peuvent écrire dans le forum du film pour discuter entre eux. Si un visiteur du site n'a pas de compte, il peut lire les messages postés mais ne peut pas écrire ou répondre à des messages.

Un utilisateur connecté peut :

- écrire un message dans le forum du film
- répondre à un message déjà posté (via le petit bouton) **TODO: faire une capture d'écran pour montrer**
- supprimer un message qu'il a posté auparavant

Idées à mettre en place si on a le temps :

- créer des filtres pour éviter les insultes dans les messages
- créer des tables en cas de trop nombreux messages pour ne pas avoir des pages trop longues

Fichiers :

- "elements/forum-message.php" : pour écrire un message
- "elements/forum".php pour la gestion du forum. Il sera appelé pour chaque film
- "static/css/forum.css" pour le style
- "static/js/forum.js" pour la gestion des événements
- "controller/dataforum.php" pour les fonctions faisant appel à la base de données
- "forum-response.php" pour les appels ajax

Finalement le fichier "elements/forum-message.php" n'est pas utilisé dans la construction du forum... A la place on passe directement cette partie du code dans le fichier "forum-response.php" pour renvoyer du code html. Ce n'est pas très joli au niveau du code mais ça marche et c'est la seule méthode que j'ai trouvée...

4.2.4 Mise en place des messages dans le forum

Lors de l'appel Ajax, nous utilisons le script *forum-response.php* qui va écrire automatiquement le code HTML correspondant à chaque message relatif au film dont parle le forum.

En plus d'afficher le message, son auteur et la date à laquelle le message a été posté, nous créons deux boutons :

- le bouton de réponse, visible pour tous les utilisateurs connectés
- le bouton de suppression du message, visible uniquement par l'auteur du dit message

Nous construisons deux classes associées *forum-reply-btn* et *forum-delete-btn* ce qui nous permettra de les appeler plus simplement dans le script javascript/Jquery associé *static/js/forum.js*.

4.2.4.1 Répondre à un message ou le supprimer

Un utilisateur peut supprimer son message à tout moment.

CONCLUSION

TODO: Créer une annexe avec l'arborescence des fichiers du projet