Présentation de MovieShelter

****

Ce document est rédigé à la suite du déploiement du site Movie Shelter et de son application mobile, réalisé par :

-Jimmy ANDRE

-Eliott CHEMINAL

-Gaëlle REMINIAC

-Maël BELHACENE

Il a pour but de présenter l’architecture de l’application web, des technologies utilisées, ainsi que de fournir des instructions faisant office de référence pour les utilisateurs et aux administrateurs.

1) Architecture et technologies

1. Technologies :

* Laravel V11.22
* Composer
* PHP V8.2
* MySQL/MariaDB
* Apache
* SMTP Google
* React Native
* Git

1. Une image contenant texte, capture d’écran, diagramme, nombre

   Description générée automatiquementMCD

2) Présentation de l’application

Movie Shelter est une vidéothèque permettant à ses utilisateurs de noter, lister et commenter des films permettant de réunir la communauté cinéphile afin d’échanger et de proposer des films.

L’application se compose de plusieurs pages :

* La page d’Accueil

Permet de visualiser les films les plus populaires ou encore ceux actuellement à l’affiche ou qui sortent prochainement

Une image contenant texte, capture d’écran, collage, graphisme

Description générée automatiquement

* La page de Recherche



Une image contenant texte, capture d’écran, logiciel, Système d’exploitation

Description générée automatiquement

* Page de présentation d’un film

Lorsqu’un utilisateur est connecté, et n’a pas encore ajouté le film à sa liste, il peut l’ajouter et sera considéré par défaut comme non-vu. Il peut également laisser des commentaires et les éditer.

Une image contenant texte, capture d’écran, personne, habits

Description générée automatiquement

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

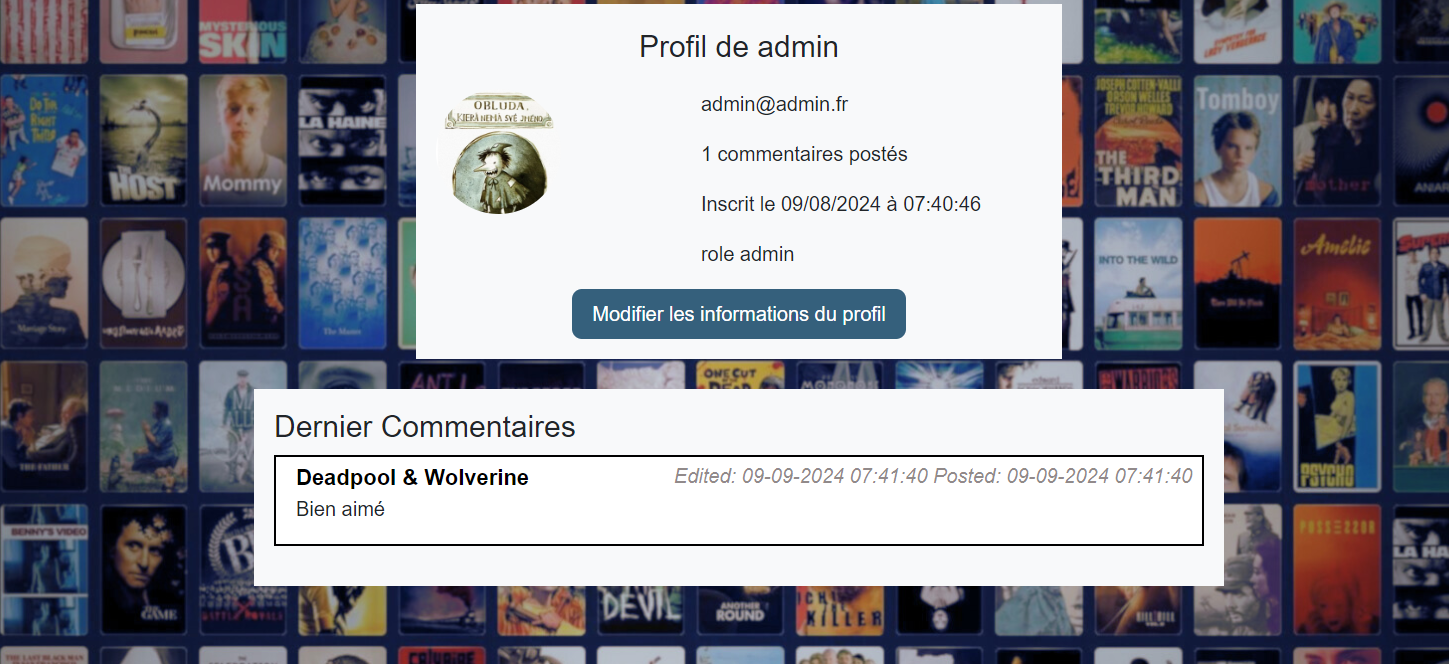
* Ma liste

Une image contenant texte, capture d’écran, logiciel, affichage

Description générée automatiquementPermet de lister les films qu’on souhaite voir où qu’on a vu et de les noter.

* Profil

Le profil d’un utilisateur est accessible par tous. Lorsque l’utilisateur est connecté il peut modifier son profil.



* Connexion

Une image contenant texte, capture d’écran, Rectangle, logiciel

Description générée automatiquement

* Inscription



* Interface administrateur

Un utilisateur dont le rôle est ‘admin’ possède l’accès au panneau d’administration. De là, il peut modérer les commentaires, et bannir ou dé-bannir des utilisateurs.

Une image contenant texte, capture d’écran, affichage, logiciel

Description générée automatiquement

2) Retour D’expérience

* Jimmy

Il est possible que la décision d’utiliser un framework comme Laravel plutôt que du PHP pur nous a apporté des difficultés que nous aurions pu éviter en étant moins ambitieux. Le nombre élevé de modules et le manque de familiarité avec le MVC a rendu la démarche plus hasardeuse, pour ma part, j’ai découvert certaines fonctionnalités et méthode d’organisation après avoir écrit le code qui en aurait bénéficié.

Malheureusement, modifier du code fonctionnel au potentiel détriment du projet est un risque que je n’ai pas souhaité prendre compte tenu de la date de rendu proche assez proche (dans le sens ou tout développements sur ce projet cesseront à la fin du bloc actuel).

Un exemple du manque de compréhension du MVC est l’une de mes idées fausses au début de ce projet. Je pensais (à tort) qu’un Controller correspondait uniquement à une page web, visible par l’utilisateur.

C’est ce jugement erroné qui m’a poussé à écrire des méthodes comme « WriteComment » et « UpdateComment » dans le Controller « MovieDetails », car ce dernier renvoyait une vue censée afficher les détails d’un film ainsi que les commentaires s’y rapportant.

Il aurait été bien plus judicieux de créer un « CommentController » pour gérer l’ajout, la mise à jour et la suppression de commentaire.

Quant à Laravel, il m’est arrivé d’écrire des méthodes de Modèles qu’Eloquent proposait déjà ! Ou encore de passer à côté des Components qui auraient grandement facilité l’écriture de templates réutilisables.

Je regrette ne pas avoir utilisé Bootstrap plus tôt, surtout pour créer des éléments pouvant être « collapse » sans écrire de JavaScript et la facilité de faire du responsive.

Pour le positif, j’ai trouvé le sujet très intéressant. Notamment la création de scripts pour scraper l’API TheMovieDB et convertir le .json en expressions SQL dans le but de peupler la base de données.

Un effort a été fourni pour maintenir une communication quasi-hebdomadaire entre les membres du groupe, ce qui a facilité la productivité.

Selon moi, la somme des difficultés rencontrées et des progrès réalisés m’ont mené à une meilleure compréhension du développement web et du MVC de manière générale.

* Eliott

Cette première approche du langage web/serveur en MVC m’a permis d’approfondir mes connaissances globales sur le développement full stack, néanmoins je rejoins Jimmy sur le fait qu’utiliser un Framework tel que Laravel pour un premier projet fut particulièrement difficile au fur et à mesure que le projet avançait.

Nous avons notamment découvert trop tard que l’utilisation de Laravel Ui nous aurait grandement aidés pour la génération des tables, des droits ou encore des scaffolding.

Un grand effort de recherche a été nécessaire, surtout pour la configuration du SMTP et la gestion de l’inscription et de la connexion par Google par exemple.

La partie CSS/HTML m’a permis de mettre en application ce que nous avons appris en cours, avec la satisfaction de voir notre application web évoluer en temps réel. Je note néanmoins la difficulté de créer un menu burger sans Bootstrap.

Pour la création du MCD, la création des tables et l’insertion des données, un travail de recherche a été mené dès la création de l’application, afin d’éviter de futurs problèmes de contrainte qui peuvent mener à une refonte de l’application, accompagné par des tests à l’aide de seed avant d’importer les datas de l’API. La création et la gestion du MVC User furent pour moi le premier défi d’une implémentation sous modèle MVC.

L’aspect sécurité a aussi été pris en compte et m’a permis de comprendre l’importance de se protéger contre les attaques XSS ou encore les injections SQL.

La fréquence hebdomadaire de nos réunions dès le début nous a permis de vite nous organiser et de nous entraider sur nos sujets respectifs. De plus, l’application de règles strictes via Git que nous nous sommes imposés dès le début du projet a grandement aidé à l’organisation et à la répartition des tâches au sein du groupe.

En conclusion, je suis satisfait du travail produit qui m’a beaucoup apporté, mais je reste convaincu que nous aurions gagné à mieux nous renseigner sur Laravel avant de nous lancer dans du développement.

* Gaëlle

Je rejoins mes collègues concernant l’utilisation de Laravel. Malgré le cours qui nous a permis une découverte de la technologie, nous n’avions pas réalisé l’approfondissement nécessaire pour développer un projet complet comme celui-ci.

Devoir avancer à tâtons en comprenant au fur et à mesure les fonctionnements a causé quelques obstacles à corriger plus tard, et ainsi du temps perdu.

Cependant, l’exercice était très enrichissant. J’avais personnellement déjà commencé à mettre en place un pattern MVC dans le cadre de mon travail et ce projet m’a permis d’encore mieux apprivoiser ce modèle.

Pour ma part, j’ai également passé beaucoup de temps pour développer l’application mobile. J’ai énormément apprécié l’aisance de prise en main de react native (avec Snack Expo) avec lequel on peut avoir de jolis résultats assez rapidement. Aller plus loin avec les Modals, les Contexts et les appels API n’ont pas été simples à mettre en place en premier abord mais une fois la logique acquise, tout devient plus fluide.

L’organisation de l’équipe a été très appréciable. Les retours hebdomadaires ont permis de garder un œil sur l’avancement général du projet et de prendre conscience des éventuels oublis en prenant du recul.

Pour finir, je me réjouis fortement de ce projet dont la double casquette développement web et application mobile a été très instructif.

* Maël

Pour ma part, l'expérience avec Laravel a été à la fois stimulante et formatrice. Le choix du framework, bien que ambitieux pour un second projet, m'a permis de plonger profondément dans le modèle MVC et de comprendre ses nuances. Au début, j'ai eu des difficultés à saisir comment structurer le code de manière optimale, ce qui a entraîné des ajustements tardifs et des révisions. Cependant, cette immersion m'a permis de découvrir la richesse de Laravel, comme les outils de migration et les fonctionnalités d'authentification intégrées, que nous avons réalisées plus tard dans le projet. Même si j'ai parfois eu des difficultés à éviter les redondances dans le code. La partie la plus enrichissante a été de voir le projet prendre forme et d'appliquer les concepts appris en cours à des cas concrets. je suis satisfait des compétences acquises et de la manière dont nous avons surmonté les défis ensemble.