**Présentation de MovieShelter**

****

Ce document est rédigé suite au déploiement du site MovieShelter et de son application mobile, réalisé par :

-Jimmy ANDRE

-Eliott CHEMINAL

-Gaelle 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
* Git

1. MCD

//MCD À INSÉRER ICI

1. Architecture Globale

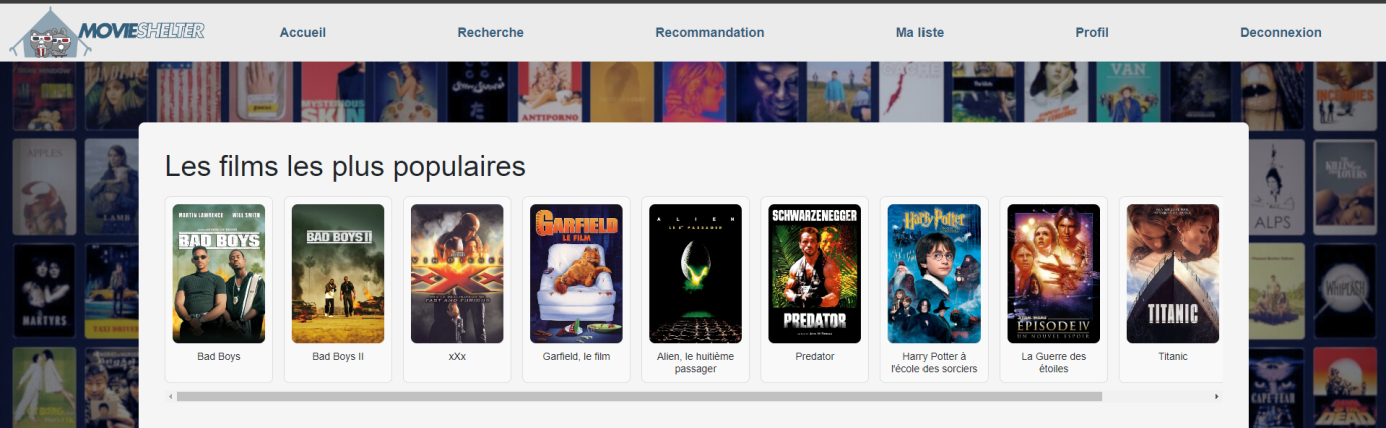
2) Présentation de l’application

Movie Shelter est une vidéothèque permettant à ces 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



* La page de RechercheUne image contenant texte, capture d’écran, logiciel, ordinateur

  Description générée automatiquement

Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement

* Ma liste

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

Description générée automatiquement

* Profil

Une image contenant texte, capture d’écran

Description générée automatiquement

* Connexion

Une image contenant texte, capture d’écran

Description générée automatiquement

* Inscription

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

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’organisations 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 sensée 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 que qu’Eloquent proposait déjà ! Ou encore de passer à côté des Components qui auraient grandement facilités l’écriture de templates réutilisable.

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 scrapper 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 difficulté rencontrées et des progrès réalisés m’ont mené à une meilleure compréhension du développement web et du MVC de manière générale.