

## Laravel Séries

---

Compte rendu de travail



**Etudiant : Ruffieux Mathis**

# **Sommaire :**

## **I) Introduction**

## **II) Guide d'installation du projet**

## **III) Exercices du TP2**

- 1) Page d'accueil
- 2) Page des séries
- 3) Formulaire de contact
- 4) CRUD des séries

## **IV) Exercices supplémentaires**

- 1) La gestion des commentaires
- 2) La gestion des acteurs
- 3) Le système de notation
- 4) Identification et rôles d'administration
- 5) Uploader des fichiers médias
- 6) Gestion des fichiers multimédia

## **V) Surprise**

- 1) Affichage des profils utilisateurs
- 2) Barre de recherche
- 3) Design CSS soigné

## **VI) Conclusion**

## **VII) Crédits**

## **VIII) Annexes**

## I) Introduction

Tout d'abord, j'ai travaillé seul sur ce projet car je développe sur un PC fixe depuis chez moi et je souhaitais apprendre par moi même à faire toutes les fonctionnalités.

Ce projet m'a permis de découvrir le framework Laravel ainsi que la notion d'architecture MVC en PHP.

Le temps passé à l'apprentissage de Laravel et à la réalisation de ce projet est d'environ 50 heures.

Toutes les routes se trouvent dans le fichier web.php.

*(Je n'ai pas créé de seeder qui génère automatiquement des séries aléatoires, car il est nécessaire d'ajouter au moins une image de l'affiche d'une série lors de la création des séries. Il est aussi mieux d'ajouter un lien youtube valide pour voir le trailer depuis la page d'une série plutôt qu'une chaîne de caractères aléatoire.)*

Vous trouverez des exemples de descriptions et d'images de films dans le répertoire movies\_assets se trouvant dans le même répertoire que ce rapport sur le git.

Pour créer des séries, il vous suffira donc d'utiliser ces images et de copier coller le titre, les tags, la description, les acteurs et l'url du trailer de chaque film se trouvant dans ce dossier (fichier movies\_data.txt).

Vous pouvez aussi créer des films avec les images et la description de votre choix.

## II) Guide d'installation du projet

Tout d'abord, télécharger le projet via git.

Ensuite, ouvrez la cmd et taper les commandes suivantes :

- composer install
- composer update

Créer un fichier .env en copiant le fichier .env.example

Dans ce fichier, changer toutes les lignes commençant par "DB\_" en la ligne suivante :

"DB\_CONNECTION=sqlite"

Créer un fichier nommé database.sqlite dans le répertoire database puis effectuer la commande :

- php artisan migrate

ou

- php artisan migrate fresh

Ensuite, pour ouvrir le serveur il faut générer une clef :

- php artisan key:generate
- php artisan serve

Aller à l'adresse indiquée (exemple : <http://127.0.0.1:8000> )

Pour donner l'accès au storage, exécuter la commande suivante :

- php artisan storage:link

Le projet est prêt à être utilisé.

### **III) Exercices du TP2**

#### **1) Page d'accueil**

La vue de la page d'accueil est le fichier menu.blade.php.

Toutes les vues du projet étendent la vue main.blade.php, dans laquelle se trouve la navigation bar (en haut) et le footer (en bas).

Le fichier MenuController.php comporte une fonction index() qui renvoie cette vue avec la liste des séries.

Sur la page d'accueil sont affichées les affiches des 8 premières séries disponibles sur le site. [Voir annexe 1](#)

- Si il n'y a aucune série alors un message “*Il n'y a encore aucune série !*” apparaît.
- Si il y a moins de 8 séries, certaines séries sont répétées jusqu'à qu'il y ait exactement 8 affiches pour compléter le cercle.

#### **2) Page des séries**

Le fichier series.blade.php permet l'affichage de la page avec l'ensemble des films ou séries du site. [Voir annexe 2](#)

Sur cette page se trouve la liste des séries avec leur affiche respective. Lorsqu'on passe la souris sur une série, son nom apparaît avec un hover. En cliquant sur l'affiche, on arrive sur leur page dédiée à cette série avec toutes ses informations.

Par ailleurs, il y a une barre de recherche pour chercher une série par titre, acteurs ou tags que l'on verra dans la partie V (2) du rapport.

Sur la page d'une série [Voir annexe 3](#), on y trouve son titre, ses acteurs, son propriétaire (celui qui l'a posté), un trailer (youtube), des icônes vers les réseaux sociaux (si on veut en ajouter), ses notes (étoiles) et les commentaires en dessous.

### 3) Formulaire de contact

Pour remplir le formulaire de contact [Voir annexe 4](#), il n'y a pas besoin d'être connecté. En effet, on doit spécifier son nom et son email avec le message. En dessous du formulaire se trouve la liste des profils administrateurs du site qui pourront voir nos messages.

Lorsque l'on envoie un message avec la méthode post, on appelle la fonction store du fichier ContactController.php qui va enregistrer le message et ses informations dans la base Contat.

Ces messages sont visibles dans cette même vue en dessous du formulaire lorsqu'on est connecté en tant qu'administrateur [Voir annexe 5](#).

### 4) CRUD des séries

Lorsqu'on est connecté, de nouveaux boutons apparaissent dans la navigation bar [Voir annexe 6](#) :

Celui pour ajouter une série et celui pour voir son profil.

Toutes les fonctions du CRUD pour les séries sont dans le fichier SeriesController.php

On accède donc au formulaire de création d'une série en cliquant sur ce dernier [Voir annexe 7](#).

Dans ce formulaire, on nous demande toutes les informations d'une série et y compris 3 images.

L'image de l'affiche du film est obligatoire ainsi que tous les champs de texte. Les images représentant le fond d'écran et d'icône du films sont facultatives, car elles sont remplacées par des images par défaut si ces champs sont vides. (Nous reviendrons dessus dans la partie IV - 5 avec le multi-upload)

Pour modifier ou supprimer une série, il faut être connecté et que la page de série sur laquelle on se trouve nous appartienne [Voir annexe 8](#). Dans ce cas, les boutons de modification et de suppression de la série apparaissent.

Lorsqu'on clique sur supprimer, la fonction remove du SerieController est appelée. Tous les commentaires sont supprimés ainsi que toutes les images uploadées avant la suppression effective de la série, pour ne pas laisser de données zombies dans les bases.

Lorsqu'on clique sur modifier, on est redirigé vers le formulaire de modification de cette série sur laquelle les champs de texte sont pré-remplis par les données actuelles de la série et ses images sont visibles. [Voir annexe 9](#).

Nous pouvons ainsi changer par exemple l'image de fond si elle ne nous convient pas (il est recommandé d'utiliser des images plutôt "sombres" pour aller avec le design du site) ou pour ajouter des tags et des acteurs.

Si on essaye tout de même d'accéder au lien de la modification d'une série sans être connecté ou en étant connecté avec un compte qui ne possède pas cette série, un message dédié apparaît et la modification de la série n'est pas autorisée [Voir annexe 10](#).

## **IV) Exercices supplémentaires**

### **1) La gestion des commentaires**

Nous pouvons voir les commentaires (Triés par récents d'abord) postés sous les séries [Voir annexe 11](#).

Les commentaires disposent d'un CRUD comme pour les séries, dans le fichier CommentController.php

Pour poster, modifier ou supprimer un commentaire, nous devons être connectés et le commentaire doit nous appartenir.

En effet, un commentaire appartient à la fois à une série et à un User.

Lorsqu'une série est supprimée, tous les commentaires qui lui sont associés le sont aussi.

### **2) La gestion des acteurs**

Il est facile de créer un nouveau modèle en Laravel (comme pour les commentaires par exemple), mais il n'y a pas de Modèle dédiés aux acteurs car il est suffisant de les écrire sous la forme d'une chaîne de caractère dans le modèle Série pour effectuer une recherche par acteurs.

Par exemple, si je tape Emma Watson dans la barre de recherche [Voir annexe 12](#), je trouve toutes les séries dans laquelle son nom apparaît.

### 3) Le système de notation

Pour les notes, j'ai créé un nouveau modèle Rate.php qui appartient à la fois à une série et à un user. La valeur d'une note est comprise entre 1 et 5 et sa création / modification est gérée dans le fichier RatesController.php

Pour noter une série, il faut être connecté. On peut ensuite cliquer sur les étoiles de la série [Voir annexe 13, 14](#) et la moyenne des notes de cette série sera calculée puis stockée dans un paramètre "note" du modèle Serie, qui peut être nullable (lorsqu'il n'y a pas encore de notes).

On verra donc sur la vue série le nombre d'étoiles remplis en fonction de la note, arrondie au demi point au dessus (par exemple si la note moyenne est 2.5, on affiche 3 étoiles mais si la note moyenne est 2.4, on affiche 2 étoiles).

Lorsqu'il n'y a aucune note, aucune étoile n'est remplie et un message "Aucun vote" est affiché. On peut voir le nombre de votes sur chaque série à côté des étoiles.

Lorsque l'on vote, si l'User connecté n'a encore jamais voté pour cette série, une nouvelle data Rate est créée.

Si l'utilisateur revote pour la même série, cette note sera modifiée et les notes moyennes sont calculées. Ainsi un utilisateur ne peut pas créer 2 notes différentes pour une même série.

### 4) Identification et rôles d'administration

Il est possible de se connecter ou de se créer un compte via les boutons "Log in" et "Register" situés en haut à droite de chaque page. Un modèle User.php a été créé pour les utilisateurs.

Les utilisateurs ont : Un nom, un email, un username et un mot de passe. Ils possèdent aussi des séries, des commentaires et des notes.

Les utilisateurs peuvent modifier et supprimer les séries qui leurs appartiennent, ainsi que modifier et supprimer les commentaires qui leurs appartiennent uniquement. [Voir annexe 15](#).

Par ailleurs, lorsqu'on est connecté, deux nouveaux boutons apparaissent dans la navigation bar. Le bouton pour ajouter une série qui nous redirige vers le formulaire de création d'une série et le bouton pour voir notre profil (expliqué en partie V (1) ) [Voir annexe 16](#).

Pour se déconnecter, il suffit de cliquer sur le bouton "Logout" et on sera redirigé vers la page d'accueil du site.

Pour différencier un utilisateur classique d'un administrateur, un booléen isAdmin est implémenté dans la table de migration des Users. Lors de la création d'un compte, ce booléen est *null*. Dans les vues, lorsqu'un utilisateur est un admin (booléen à *true*), il dispose de fonctionnalités supplémentaires.

Un administrateur pourra modifier et supprimer toutes les séries et tous les commentaires même ceux qui ne leur appartiennent pas.

Un administrateur pourra voir les messages postés dans le formulaire de contact [Voir annexe 5](#).

Un administrateur pourra, en allant sur la page profil d'un utilisateur, le passer en tant qu'administrateur [Voir annexe 17](#).

De la même manière, il pourra lui retirer ses droits [Voir annexe 18](#).

Lorsqu'un profil est passé en Administrateur, il aura les mêmes droits que l'administrateur d'origine, c'est-à-dire qu'il pourra par exemple lui-même se retirer ses droits.

**Procédure à suivre pour créer le premier administrateur du site** (après avoir créé au moins un compte avec le bouton Register) :

- php artisan tinker
- \$user = App\Models\User::find(1)
- \$user->isAdmin = true
- \$user->save()
- \$user->push()

Ainsi, le premier utilisateur (ici d'id 1) sera un admin et pourra passer d'autres utilisateurs en tant qu'admin directement via le site en cliquant sur leur profil.

## 5) Uploader des fichiers médias

Nous pouvons uploader des images dans les formulaires de création et de modification des séries, avec les vues createSerie.blade.php et updateSerie.blade.php [Voir annexe 7](#).

Dans les fichiers de migration des Séries se trouvent des chaînes de caractères représentant les paths vers les images miniatures (pour les affiches de films), background et logo des séries.

Ces paths sont gérés par les fonctions de store, remove et update dans le fichier SeriesController.php.

L'image de miniature est requise lors de la création d'une série, mais les images de background et de logo sont facultatives. En effet, le contrôleur remplace leur paths par des chemins par défaut si aucune image n'a été uploader.

Lors de l'upload d'une image (en utilisant un multi-upload), elles sont stockées dans le répertoire storage du projet.

Pour donner l'accès aux images dans les vues et qu'elles s'affichent correctement dans les vues, nous avons lié ce répertoire avec répertoire storage publique à l'initialisation du projet (commande "php artisan storage:link").

Lors de la suppression d'une série, nous allons aussi supprimer les images se trouvant dans les variables paths pour ne pas laisser des images inutiles.

(Fonction remove() du controller des séries)

Nous ne supprimons évidemment pas les chemins des images par défaut si aucune image n'a été uploadé pour le background ou le logo.

De la même façon nous pourrions uploader des images de profils utilisateurs et en mettre une par défaut (le champ string "image\_profile" du modèle User à été ajouté).

## 6) Gestion des fichiers multimédia

En plus de l'authentification, du multi-upload, du CRUD des images et link du répertoire storage, nous pouvons voir des vidéos de trailer sur youtube en utilisant une API qui permet de lire les liens youtubes sur notre site.

Source de l'API utilisée :

<https://github.com/thelevicole/youtube-to-html5-loader>

Dans la vue, lorsque nous cliquons sur le bouton "Voir trailer" [Voir annexe 3](#), un champ s'ouvre avec la vidéo et une croix pour la fermer en haut à droite.

Bien sur celà fonctionne que si on met un lien de vidéo valide (des exemples d'url à copier coller sont écrits dans le répertoire movies\_assets du git).

## **V) Surprise**

### **1) Affichage des profils utilisateurs**

Une vue dédiée aux profils utilisateurs a été implémentée dans le fichier profile.blade.php [Voir annexe 17](#).

Lorsque nous sommes connectés, nous pouvons accéder à notre profil via un bouton [Voir annexe 16](#). Il est aussi possible d'y accéder en cliquant sur notre pseudo qui s'affiche en haut à droite de l'écran.

Sur une page de profil, se trouvent nos informations : Le pseudo, le nom, la date de création du profil ainsi que les séries postées par ce profil. Nous pouvons accéder à ces séries en cliquant dessus. Nous pouvons aussi accéder au profil qui a posté une série spécifique en cliquant dessus (le bouton “Série postée par [pseudo]” sur [l'annexe 3](#) ).

Nous pouvons voir la liste des profils des administrateurs en dessous du formulaire de contact [Voir annexe 4](#) et cliquer dessus pour accéder à chacun de leur profil.

### **2) Barre de recherche**

Comme vous avez pu le voir sur [l'annexe 2](#), une barre de recherche a été implémentée sur la vue series.blade.php.

Nous pouvons grâce à cette barre, chercher des séries par titre, tags ou nom d'acteurs.

Cette barre de recherche est un formulaire en méthode post qui appelle la fonction searchByTags() dans le controller des séries. Cette fonction renvoie à la vue avec la chaîne de caractère recherchée et n'affichera que les séries contenant cette chaîne (dans son titre, tags ou acteurs)

Par exemple, en cherchant les mots clés “Super héros” [Voir annexe 19](#), nous verrons uniquement les séries avec de super héros.

Nous pouvons aussi bien chercher des mots en majuscules qu'en minuscules.

Pour annuler la recherche, il suffit de cliquer sur le bouton “Voir tout” en rouge qui affiche toutes les séries.

### **3) Design CSS soigné**

En bonus j'ai porté une attention particulière au design de mon site, surtout pour le menu principal, la page de recherche des séries et la page de description d'une série.

Le thème est sombre pour être plus agréable à l'œil de nuit et je souhaitais ajouter les affiches des films pour faire comme un vrai site de cinéma, pas seulement une liste de noms de séries générées aléatoirement.

## **VI) Conclusion**

En conclusion, j'ai apprécié découvrir ce nouveau framework.

Pour le design, je me suis contenté d'utiliser du css classique sans utiliser saas ou d'autres outils car ce projet était plutôt centré sur le backend que le frontend et je préférais passer du temps sur l'implémentation des fonctionnalités.

En remarque, j'ai eu un peu de mal à faire un affichage correct et cliquable des petites étoiles pour le système de notation.

Merci d'avoir pris le temps de lire mon rapport.

## **VII) Crédits :**

Vidéo qui m'a aidé à réaliser le css des images qui tournent sur la page d'accueil :  
<https://youtu.be/A6gk6h1MjS4>

(Je me sers notamment de cette chaîne youtube pour apprendre le css.)

Vidéo qui m'a aidé à apprendre le MVC sur Laravel :

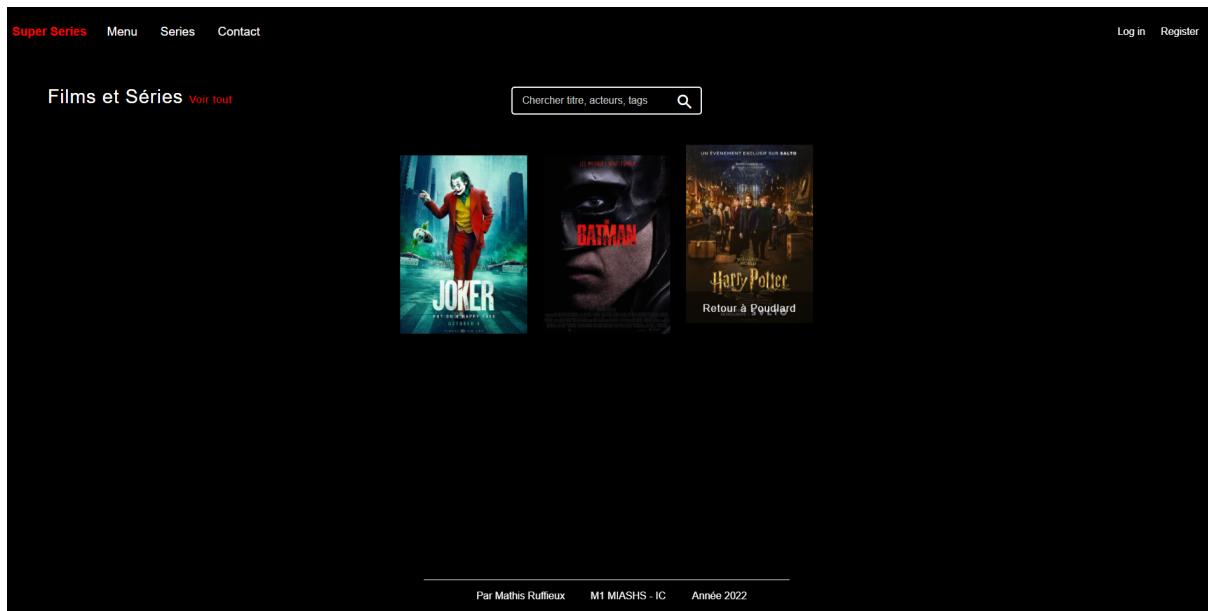
<https://youtu.be/lmtZ5yENzgE>

## **VIII) Annexes**

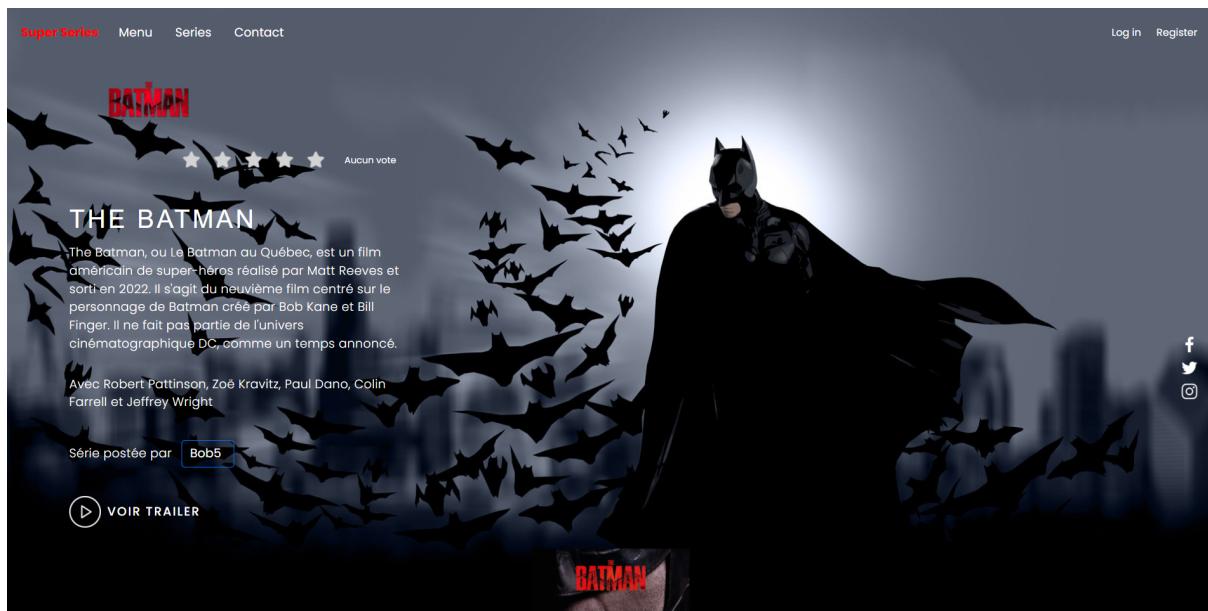
## Annexe 1 : Menu principal



## Annexe 2 : Page des séries



## Annexe 3 : Page d'une série



#### Annexe 4 : Formulaire de contact

A screenshot of a contact form page. The header includes "Super Series", "Menu", "Series", "Contact", "Log in", and "Register". The main section is titled "Formulaire de contacts". It contains three input fields: "Nom" (Name), "Email", and "Message". Below these is a blue "Envoyer" (Send) button. Underneath the message field, there's a section for "Administrateurs du site" (Site administrators) with a single entry "Admin". At the bottom of the page, there's a footer with credits: "Par Mathis Ruffieux", "M1 MASHS - IC", and "Année 2022".

#### Annexe 5 : Messages postés avec le formulaire de contact (Si admin)

Message :

Envoyer

Administrateurs du site :

Admin

**Vous êtes administrateur**

Liste des messages :

Par Bob le bricoleur le 2022-03-16 18:08:16  
Email : test@test.com  
Bonjour ceci est un message

Par John McCain le 2022-03-16 18:08:41  
Email : test@test.com  
Ceci est un autre message

Par Mathis Ruffieux M1 MIASHS - IC Année 2022

This screenshot shows a dark-themed web application interface. At the top, there's a message input field with a blue 'Envoyer' button. Below it, a section for 'Administrateurs du site' shows a single user 'Admin'. A prominent message from 'Bob le bricoleur' is displayed, followed by another from 'John McCain'. At the bottom, the footer includes the developer's name 'Par Mathis Ruffieux' and project details 'M1 MIASHS - IC Année 2022'.

## Annexe 6 : Bouton ajout de série

The screenshot shows the homepage of 'Laravel Series'. It features a dark background with movie posters for 'RENOV', 'Harry Potter', and 'Joker'. The central text reads 'Bienvenue sur **Laravel** Series'. At the top, there's a navigation bar with links: 'Super Series', 'Menu', 'Series', 'Ajout serie' (which is highlighted with a red arrow), 'Mon profil', and 'Contact'. In the top right corner, there's a user profile with the name 'Bob5' and a 'Logout' link, also highlighted with a red arrow. The footer contains the same developer information as the previous screenshot.

## Annexe 7 : formulaire de création de série

Super Series   Menu   Séries   Ajout série   Mon profil   Contact   Bob5   Logout

### Ajouter une série

Titre :

Description :

Acteurs :

Tags :

URI de la bande annonce :

Image de l'affiche du film : (Requis)

No file chosen

Image de fond d'écran : (Facultatif)

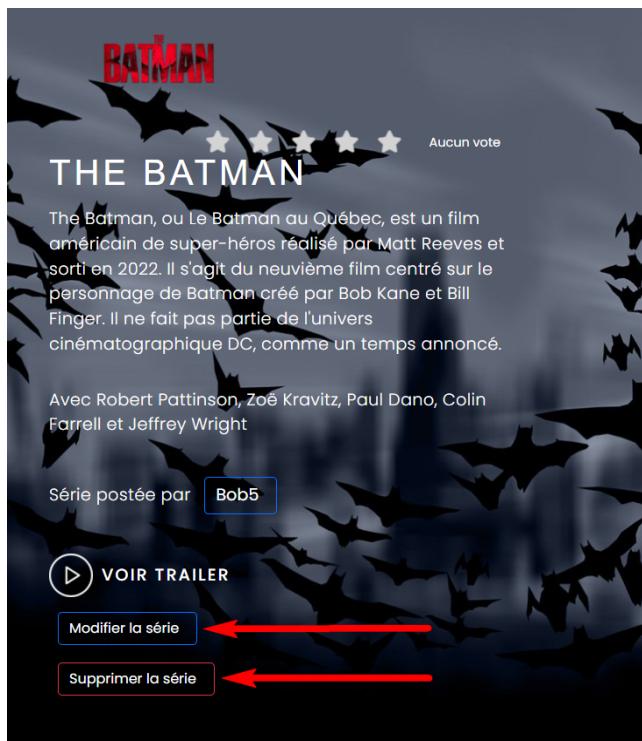
No file chosen

Image de l'icône du film : (Facultatif)

No file chosen

Par Mathis Ruffieux   M1 MIASHS - IC   Année 2022

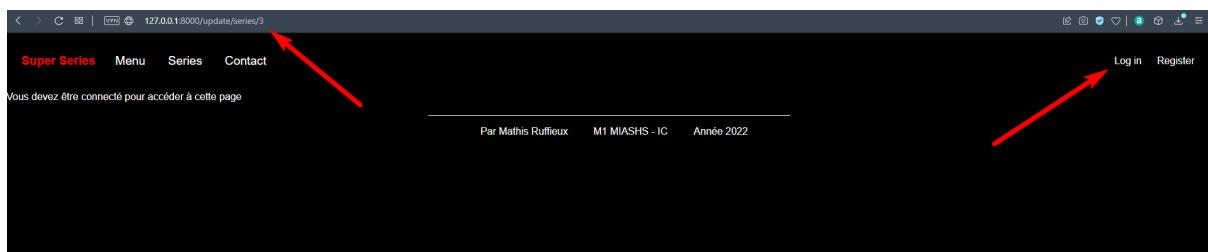
### Annexe 8 : Modifier / Supprimer une série



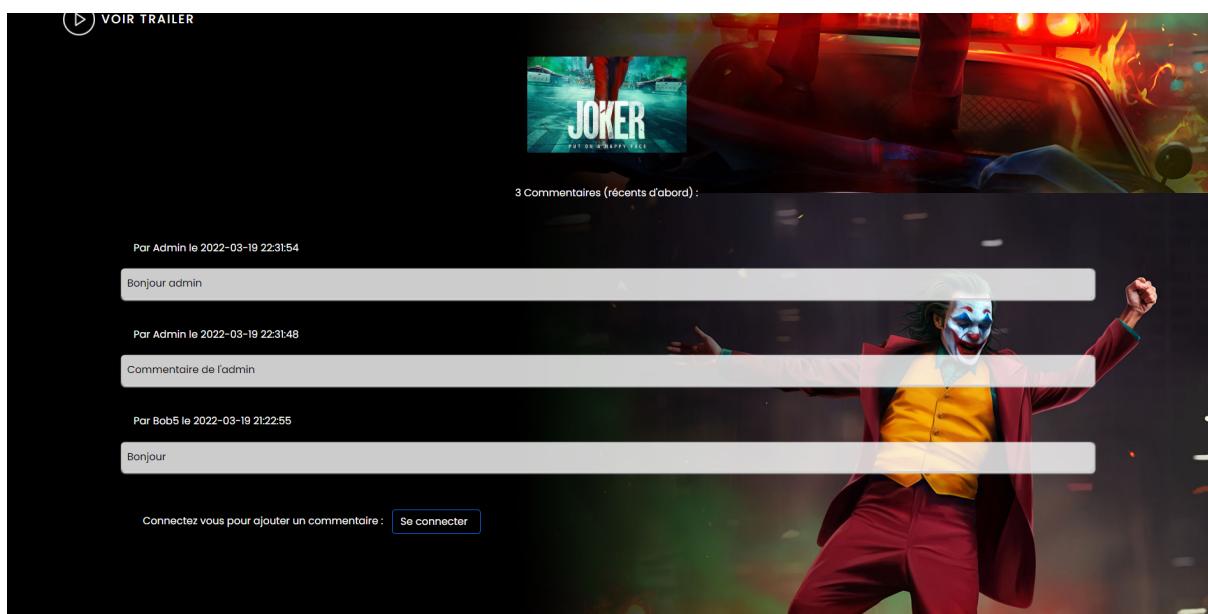
### Annexe 9 : Formulaire de modification d'une série



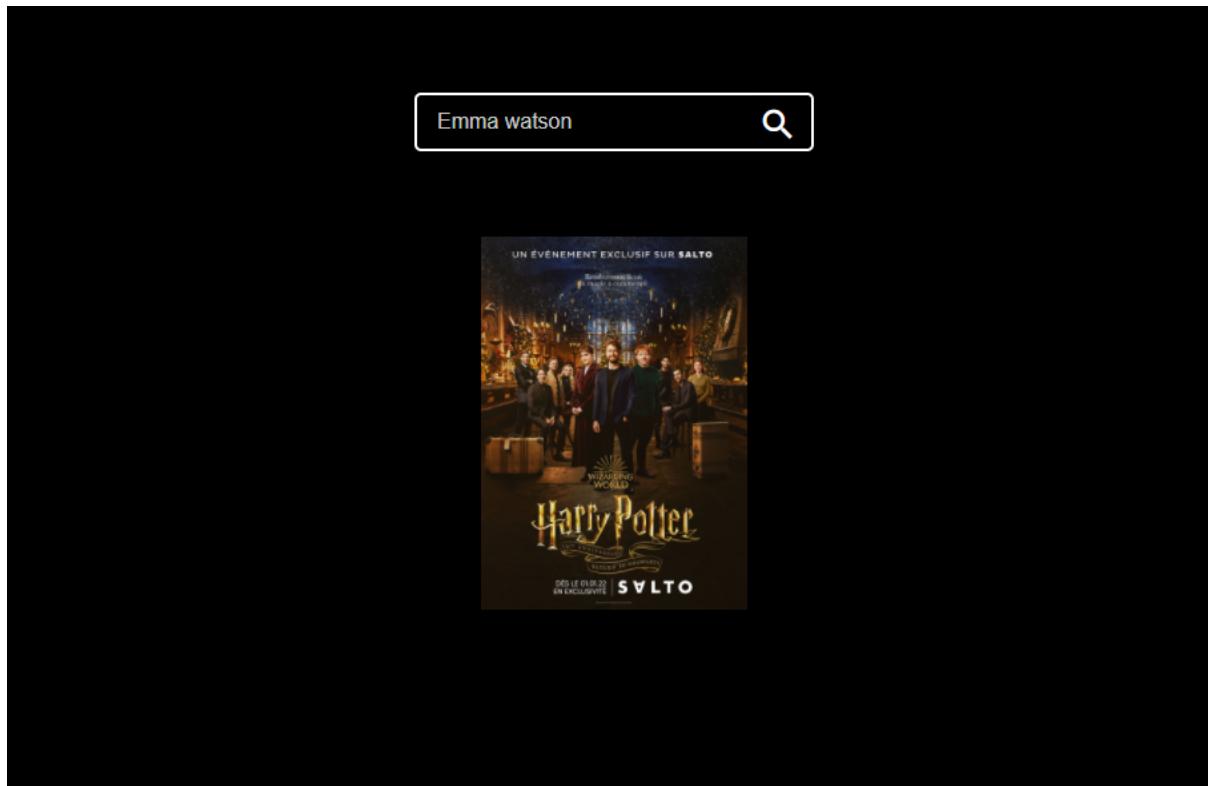
#### Annexe 10 : Tentative de modification d'une série sans être connecté



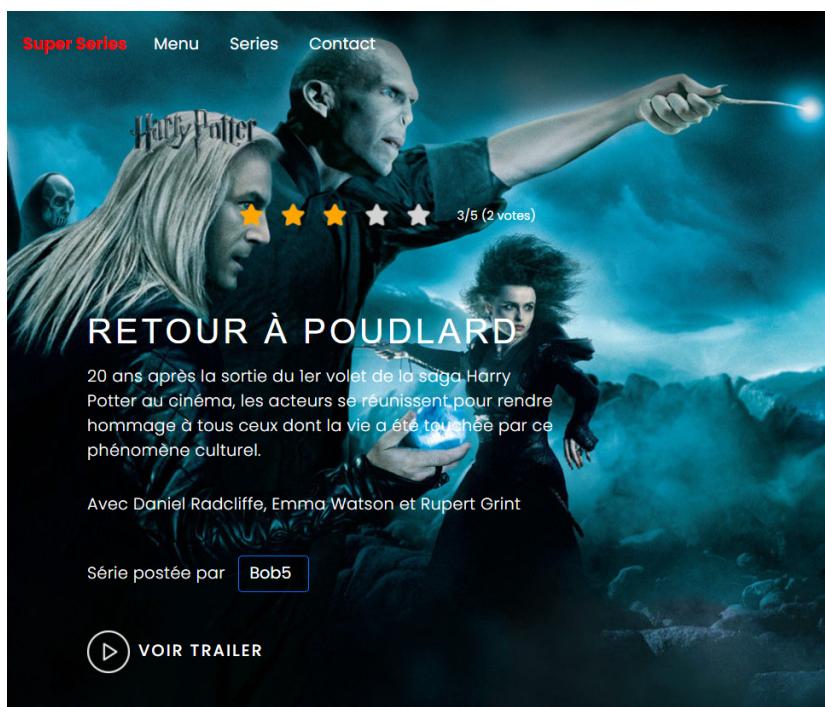
#### Annexe 11 : Commentaires des séries



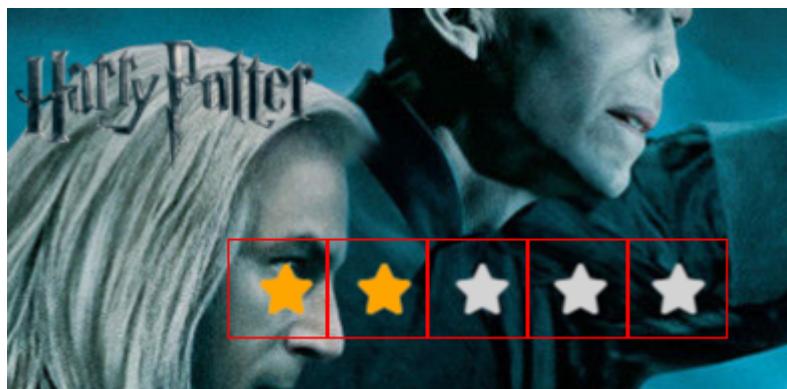
## Annexe 12 : Gestion des acteurs



## Annexe 13 : Le système de notations



## Annexe 14 : Boutons étoiles



#### Annexe 15 : Modifier / supprimer des séries et commentaires qui nous appartiennent

A screenshot of a movie series page. At the top, it says "Avec Robert Pattinson, Zoë Kravitz, Paul Dano, Colin Farrell et Jeffrey Wright". Below that, it says "Série postée par Bob5". There is a "VOIR TRAILER" button with a play icon. Two red arrows point to two buttons: "Modifier la série" and "Supprimer la série". Further down, there is a section titled "1 Commentaires (récents d'abord) :". It shows a comment by "Bob5" from "2022-03-21 14:28:17" with the text "First". Two red arrows point to the "Supprimer" and "Modifier" buttons next to this comment. At the bottom, there is a text input field "Ecrire un nouveau commentaire :" and a "Poster" button.

#### Annexe 16 : Bouton pour ajouter une série lorsqu'on est connecté

A screenshot of a navigation bar. The items listed are "Super Series", "Menu", "Series", "Ajout serie" (which is enclosed in a red box), "Mon profil", and "Contact".

#### Annexe 17 : Vue d'un profil / Bouton pour lui donner les droits d'admin

[Super Series](#) [Menu](#) [Series](#) [Ajout série](#) [Mon profil](#) [Contact](#)

[Admin](#) [Logout](#)

## Profile

Pseudo : Bob5

Nom : Bob le bricoleur

Membre depuis le 16-03-2022

Mes séries :



Ajouter Bob le bricoleur en tant qu'administrateur :

[Ajouter Admin](#)

---

Par Mathis Ruffieux M1 MIASHS - IC Année 2022

### Annexe 18 : Retirer a un profil ses droits d'administration

[Super Series](#) [Menu](#) [Series](#) [Ajout série](#) [Mon profil](#) [Contact](#)

[Admin](#) [Logout](#)

## Profile Administrateur

Pseudo : Bob5

Nom : Bob le bricoleur

Membre depuis le 16-03-2022

Mes séries :



Retirer à Bob le bricoleur ses droits d'administrateur :

[Retirer Admin](#)

---

Par Mathis Ruffieux M1 MIASHS - IC Année 2022

### Annexe 19 : Barre de recherche (Super héros)

Films et Séries [Voir tout](#)





Par Mathis Ruffieux      M1 MIASHS - IC      Année 2022