

---

# Dossier de projet

---

## **Projet *MovieLovers***

<https://www.cefii-developpements.fr/arnaud1450/projetCinema/public/>

*Du 31 mars au 30 mai 2025*

**Arnaud BERTHOLLET**

**Titre Professionnel Développeur Web et Web Mobile**

# Sommaire

|   |        |
|---|--------|
| • Introduction .....                          | page 3 |
| • Contexte du projet .....                    | page 4 |
| • Eléments significatifs coté Front-End ..... | page 5 |
| • Eléments significatifs coté Back-End .....  | page 5 |
| • Eléments de sécurité de l'application ..... | page 5 |
| • Jeu d'essai de tests fonctionnels .....     | page 5 |
| • Déploiement .....                           | page 5 |
| • Description de la veille effectuée .....    | page 5 |
| • Références .....                            | page 5 |
| • Conclusion .....                            | page 5 |
| • Annexes .....                               | page 5 |

# Introduction

Dans le cadre de la formation **Développeur Web et Web Mobile**, j'ai eu l'opportunité de réaliser un projet personnel de fin de parcours visant à mettre en pratique les compétences techniques et méthodologiques acquises tout au long de la formation. Ce projet m'a permis de concevoir, développer et structurer un site web dynamique autour d'une thématique qui me tient à cœur : le cinéma.

Inspiré de plateformes telles que SensCritique et AlloCiné, ce site propose aux utilisateurs de **découvrir des films**, consulter des fiches détaillées sur ces films et les personnes y ayant participé (acteurs et réalisateurs). Le site propose également de lire et publier des critiques.

Le site comporte **2 types d'utilisateurs** : les utilisateurs "simples" et les administrateurs (ayant accès à des fonctionnalités spécifiques).

Il s'agit d'un projet complet qui m'a permis de mobiliser les technologies du **développement web full stack**, avec une **architecture MVC en PHP**, une base de données, du **JavaScript** pour les fonctionnalités dynamiques, **Bootstrap** pour le positionnement, ainsi qu'un **CSS personnalisé** pour l'identité visuelle, en veillant à respecter le **Responsive Web Design** et à corriger les principales **failles de sécurité** vues au cours de la formation : XSS (Cross-site scripting, injections SQL, détournement de session et faille CSRF (Cross Site Request Forgery)).

Au-delà des aspects techniques, ce projet m'a également permis de développer des compétences en modélisation de base de données, en gestion de projet, en réflexion UX/UI, et en **autonomie** dans la recherche de solutions.

## Liste des compétences du référentiel couvertes par le projet :

| Activités types  | Compétences professionnelles   |
|--|--|
| Développer la partie front-end d'une application web ou web mobile sécurisée | Installer et configurer son environnement de travail en fonction du projet web ou web mobile |
|  | Maquetter des interfaces utilisateur web ou web mobile                                       |
|  | Réaliser des interfaces utilisateur statiques web ou web mobile                              |
|  | Développer la partie dynamique des interfaces utilisateur web ou web mobile                  |
| Développer la partie back-end d'une application web ou web mobile sécurisée  | Mettre en place une base de données relationnelle  |
|  | Développer des composants d'accès aux données SQL et NoSQL                                   |
|  | Développer des composants métier coté serveur  |
|  | Documenter le déploiement d'une application dynamique web ou web mobile                      |

# Contexte du projet

N'étant pas rattaché à une entreprise, **j'ai défini moi-même les besoins du projet et les contraintes techniques** en fonction de mes connaissances, de mes capacités et de mon envie de m'entraîner à pratiquer certaines notions vues pendant la formation.

J'ai endossé le rôle complet du développeur, de la phase de conception à la mise en œuvre technique : développement et déploiement de l'application.

## Expression des besoins / fonctionnalité principales

### Expression des besoins:

Mon objectif principal a été de concevoir une **plateforme web et web mobile** dédiée aux passionnés de cinéma, et contenant 3 types d'utilisateurs :

- les **visiteurs** n'ayant de compte sur le site : ils ont seulement la possibilité de consulter les différentes pages du site.
- les **simples utilisateurs** qui se sont créés un compte : ils peuvent publier des critiques sur les films une fois qu'ils sont connectés à leur compte.
- les **administrateurs**, disposant de fonctionnalités de CRUD.

L'application doit pouvoir permettre, aux utilisateurs de consulter des fiches détaillées de films, acteurs et réalisateurs ; de consulter des critiques de films laissés par d'autres utilisateurs et d'en publier s'ils sont connectés à leur compte après s'être inscrit.

Elle doit permettre aux administrateurs du site de gérer le CRUD (lire, créer, modifier et supprimer) des 3 principales entités de l'application (films, acteurs et réalisateurs), et aussi de gérer la modération des critiques.

L'application doit être **dynamique**, notamment avec des requêtes **AJAX**, et respecter le principe du **Responsive Web Design**, afin de rendre agréable la navigation de pages en pages et ainsi améliorer **l'expérience utilisateur**.

Le développement de l'application doit également se faire dans un souci d'empêcher les **principales failles de sécurité** d'être exploitées.

## Fonctionnalités principales :

- Utilisateurs simples :
  - Créer un compte et se connecter sur le site
  - Consulter des listes de films regroupés par genres.
  - Consulter des fiches détaillées sur les films comprenant une partie "présentation", une partie "casting", et une partie "critiques".
  - Consulter des fiches détaillées sur les acteurs et réalisateurs comprenant une partie "présentation", une partie "biographie", et une partie "filmographie".
  - Consulter les critiques d'un film (contenant une note et un avis) publiés par d'autres utilisateurs.
  - Consulter les critiques d'un film (contenant une note et un avis) publiés par d'autres utilisateurs.
- fonctionnalités spécifiques aux administrateurs :
  - Créer d'autres comptes de type administrateur.
  - Afficher, ajouter, modifier et supprimer des films, acteurs et réalisateurs.
  - Supprimer les critiques de film ne respectant pas les règles de bienséance.

# Contraintes techniques / livrables attendus

- Structure de l'application :

L'application est réalisée en **PHP** avec une structure **MVC**.

**WampServer** permet de l'interpréter coté navigateur et de stocker la base de données via phpMyAdmin.

L'application contient des **composants d'accès aux données** (exemple : UserModel pour accéder aux données utilisateur de la base données).

L'application contient des **composants métiers** (exemple : UserController pour manipuler les données utilisateur depuis une vue en utilisant le UserModel).

L'application contient des **vues** (exemple : formLogin pour permettre à un utilisateur de se connecter en utilisant le composant métier UserController).

- Style de l'interface :

- Utilisation de **Bootstrap** pour le positionnement, et pour gérer le responsive lorsque cela est possible.
- **CSS personnalisé** pour donner une entité visuelle à l'application.

- Langages et technologies utilisées :

- Front-end : HTML, CSS, JavaScript
- Back-end : PHP, MySQL et Bootstrap

- Versioning :

Utilisation de **Git** et **GitHub** pour garder un **historique** des différentes versions de l'application durant la conception et le développement et ainsi pouvoir revenir en arrière lorsque des erreurs importantes auraient été commises et d'assurer le maintien d'un code fonctionnel.

Etant seul à travailler sur le projet et dans un souci de gain de temps, je n'ai pas jugé nécessaire de créer des branches sur mon repository et j'ai donc fait tous les commit directement sur la branche "main".

- Plannification des tâches et gestion du projet :

Utilisation de Clickup\* pour, premièrement, décomposer le travail en fonctionnalités, puis en tâches et sous-tâches.

Ensuite, Clickup permet de planifier le travail et enfin de suivre son avancement.

exemple : fonctionnalité = Ajouter un film, une des tâche = formulaire d'ajout, une des sous-tâche = vérifier les champs du formulaire

\*Clickup est un équivalent de Trello que j'ai trouvé plus pratique à utiliser

- Livrables attendus du projet

- Les maquettes de l'interface utilisateur
- Le modèle conceptuel de données (MCD)
- Le modèle logique de données (MLD)
- Le modèle physique de données (MPD)

- Environnement humain et technique

N'étant pas rattaché à une entreprise, j'ai utilisé des outils personnels ainsi que les ressources fournies par le CEEFii :

- Un PC windows 11 :
  - Editeur de code : Visual Studio Code
  - Figma pour les maquettes et schémas
  - Looping pour les modèles de données
  - FileZilla (pour le déploiement)
  - WampServer et ses outils
  - ChatGpt pour créer les données initiales (films, acteurs, réalisateurs, critiques)
- Les ressources fournies par le CEFii :
  - Un serveur pour exécuter nos applications web et héberger nos bases de données

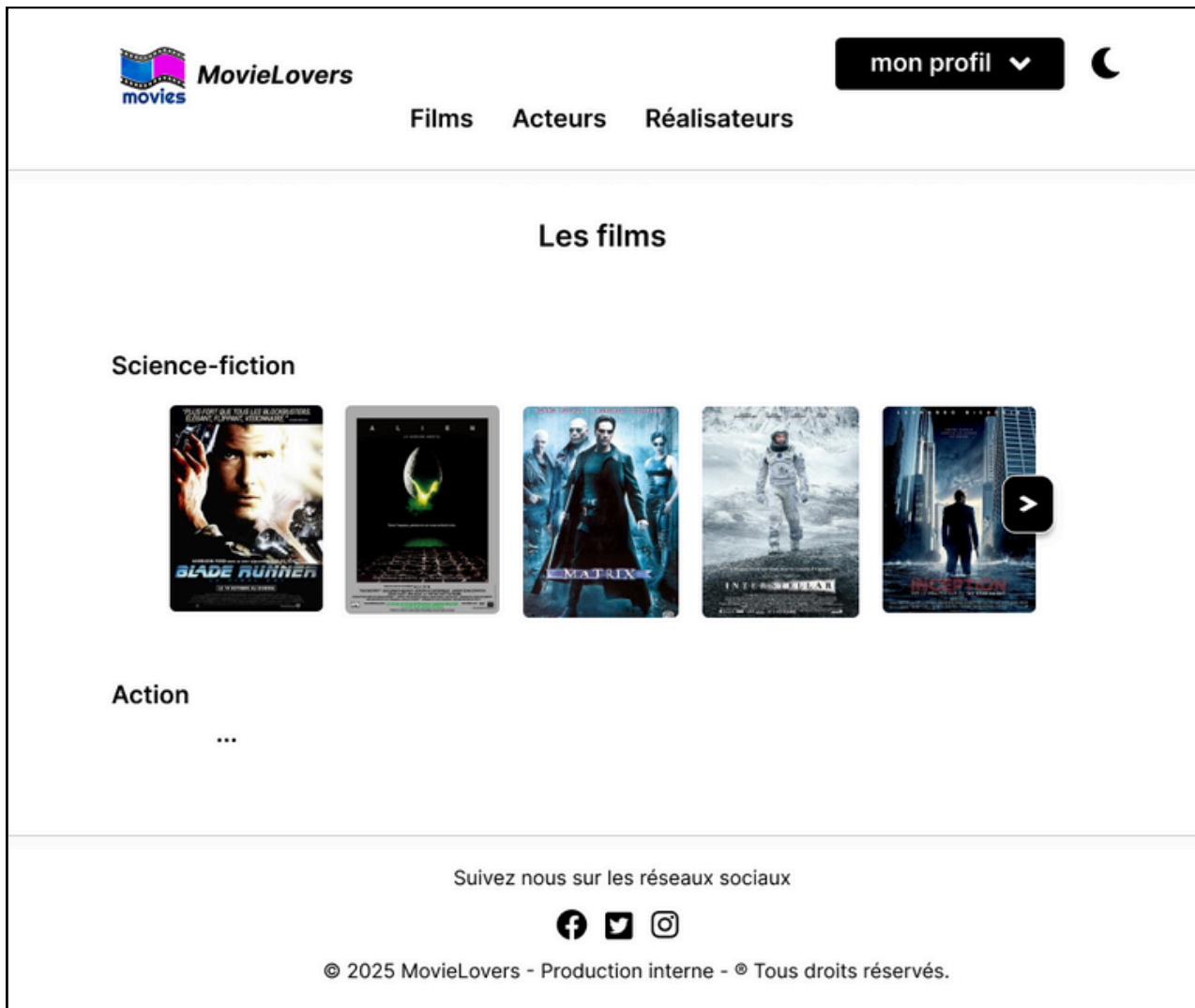
Environnement humain : J'ai travaillé seul sur ce projet depuis chez moi et depuis les locaux du CEFii.



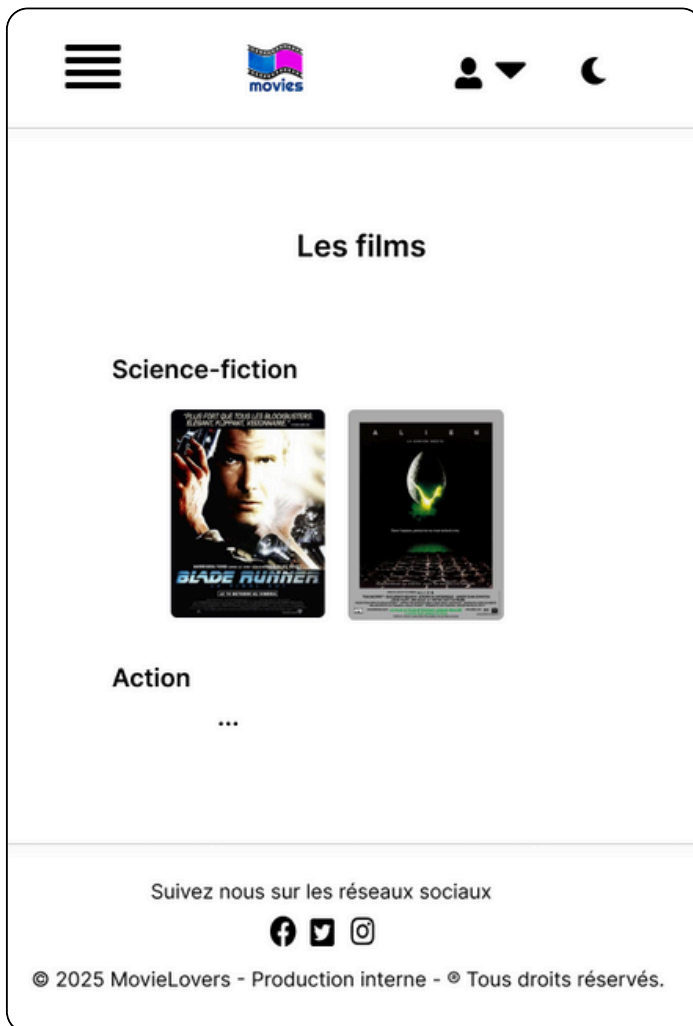
# Coté front-end

## Les maquettes de l'interface graphique, réalisées avec Figma

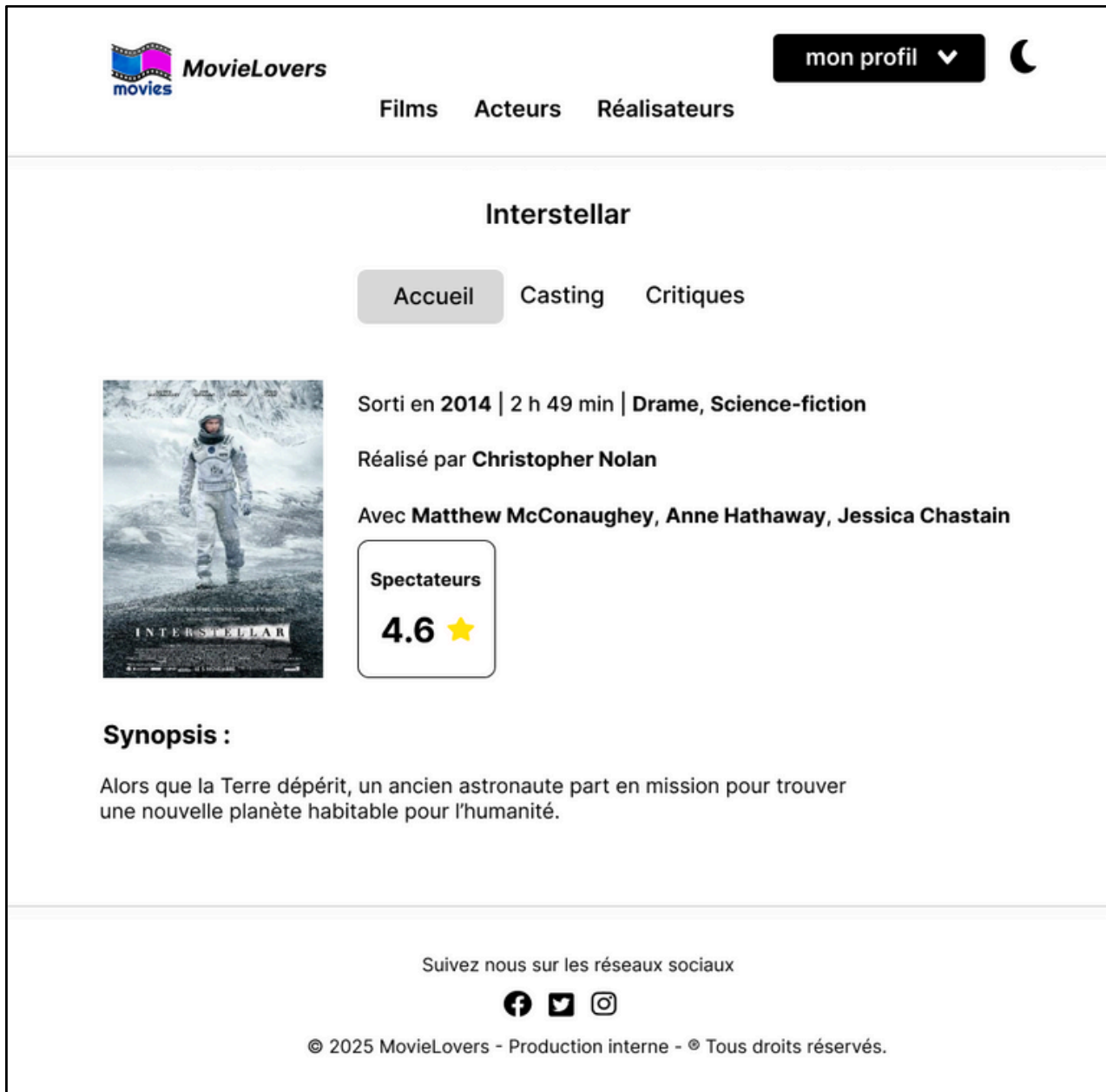
- Maquette de la page d'accueil version desktop :



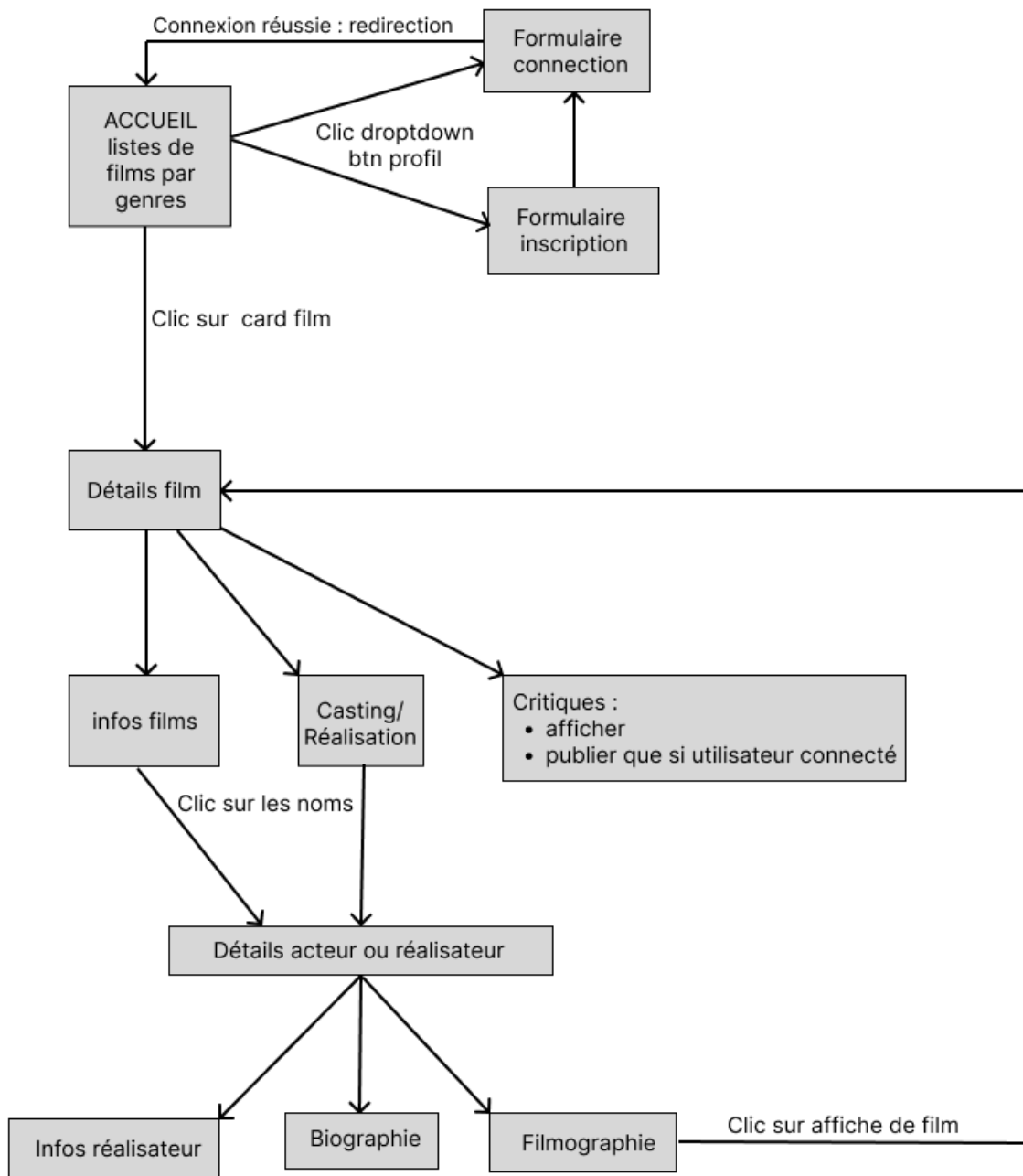
- Maquette de la Page d'accueil version mobile :



- Maquette de la page des détails d'un film :



## Schéma du parcours utilisateur

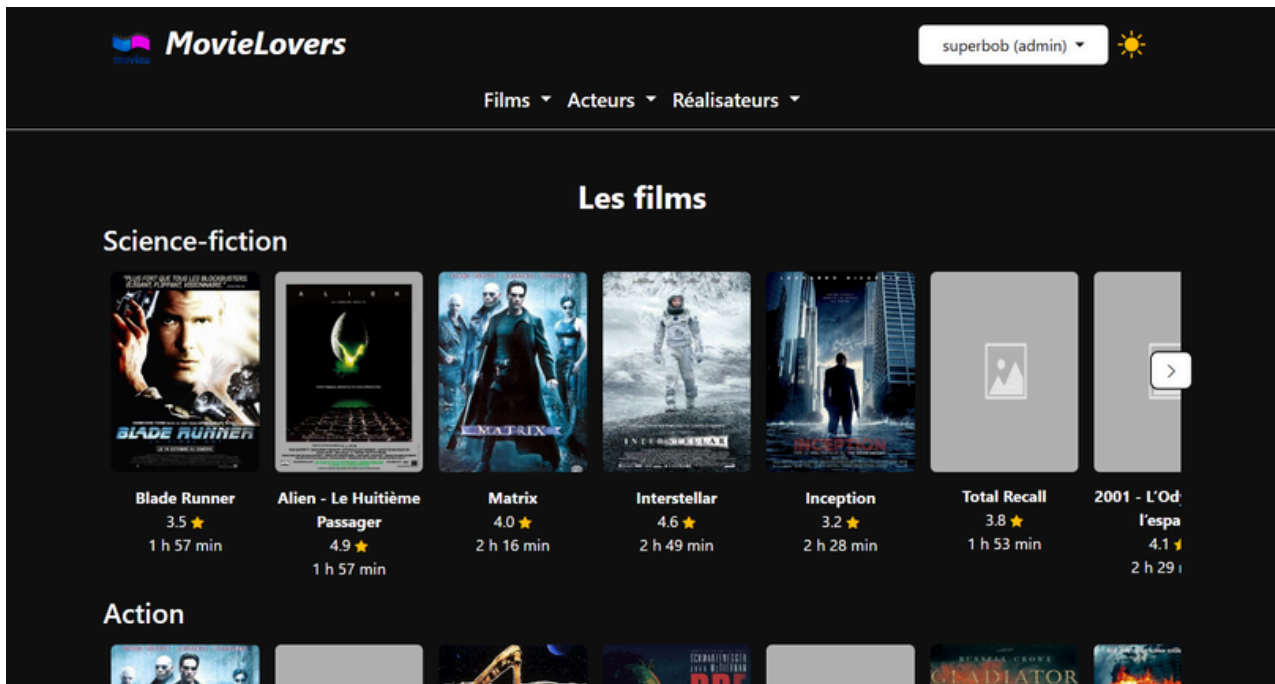


Note : Ce schéma illustre le parcours d'un simple utilisateur/visiteur du site.

Il ne comprend pas toutes les fonctionnalités comme par exemple les fonctionnalités spécifiques à l'administrateur.

# Réalisations front-end significatives

- Interface statique : la page d'accueil



Pour réaliser cette **page statique**, j'ai placé premièrement une balise HTML `<div>` pour englober le contenu à afficher dans la page et à laquelle j'ai attribué la classe bootstrap "**container**" qui a permis ici de **centrer le contenu** de la page.

Les données envoyés par le back-end sur cette page sont contenues dans une variable se présentant sous la forme d'un **tableau à 2 dimensions** contenant :

- Dimension 1 (D1) :
  - key = nom du genre ;
  - value = tableau de films (cf. dimension 2)
- Dimension 2 (D2) :
  - key = index du tableau de cette dimension (pas utilisé) ;
  - value = objet film

## Exemple :

D1['sci-fi'] = D2[ ] (key= genre "sci-fi", value= D2[genre 0])

D2[0] = film 0 du genre 0      objet film

D2[1] = film 1 du genre 0      objet film

D2[2] = film 2 du genre 0      objet film

D1['action'] = D2[ ] (key= action, value= D2[genre 1])

D2[0] = film 0 du genre 1      objet film

D2[1] = film 1 du genre 1      objet film

Pour pouvoir afficher ces données sous la forme désirée, une succession de listes horizontales (scrollables horizontalement avec du JS), j'utilise **2 boucles foreach imbriquées** pour parcourir tous les genres de films, et pour chaque genre, parcourir tous les films qu'il contient.

Les **flexbox** de bootstrap sont utilisées (la classe "d-flex") pour pouvoir afficher les listes horizontales.

Pour garantir la **sécurité** de l'application dans cette page (une des View de mon MVC), j'utilise la fonction PHP **htmlspecialchars( )** qui convertit certains caractères prédéfinis (notamment les "<" et ">" chevrons ouvrants et fermants) en entités HTML. Cela évite que des scripts soient lancés en lisant ces variables, dans l'éventualité où des scripts malveillants seraient envoyées en base de données via les formulaires. Cela constitue la pratique recommandée pour se prémunir de l'exploitation de la faille **XSS** (qui redirige via ces scripts les utilisateurs vers des pages "miroirs" pour subtiliser des informations personnelles par exemple).

L'extrait de code :

```
<div class="container position-relative">
    <!-- Titre principal de la page -->
    <h2 class="text-center fw-bolder">Les films</h2>

    <?php
    // Parcours dimension 1 : les genres et leurs films
    // -----
    <?foreach ($filmsByGenres as $genre => $films) { ?>

        <!-- Titre du "carroussel netflix" -->
        <h3 class="mb-0"><?= htmlspecialchars($genre, ENT_QUOTES, "UTF-8") ?></h3>

        <?php
        // Parcours dimension 2 : les films du genre courant
        // -----
        <?>
        <!-- Carroussel container -->
        <div class="filmScroll d-flex overflow-auto py-3 px-2 gap-3">

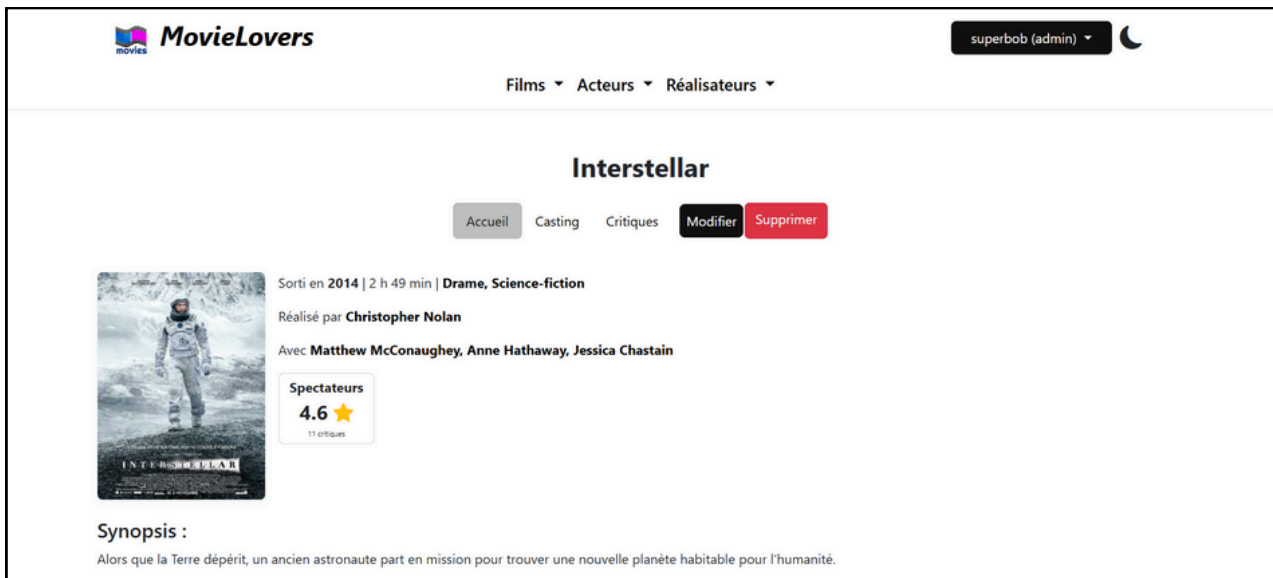
            <!-- Flèche gauche (masquée sur petit écran) -->
            <button class="scrollLeft btn darkBtn btnWithBorders position-absolute start-0 translate-middle-y d-none d-md-block z-3" style="z-index:
                <i class="bi bi-chevron-left"></i>
            </button>

            <?php
            // Liste des films en scroll horizontal
            <?foreach ($films as $film) { ?>
                <a href="index.php?controller=Film&action=details&id_film=<?= htmlspecialchars($film->id_film, ENT_QUOTES, "UTF-8") ?>" class="film
                <div class="flex-shrink-0" style="width: 150px;">
                    <object data="img/img_films/<?= htmlspecialchars($film->picture, ENT_QUOTES, "UTF-8") ?>" class="img-fluid rounded shadow-
                    
                    </object>

                    <p class="text-center fw-bold mt-2 mb-0"><?= htmlspecialchars($film->title, ENT_QUOTES, "UTF-8") ?></p>
                    <?php if (isset($film->average_rating)) { ?>
                        <p class="text-center my-0">
                            <?= htmlspecialchars($film->average_rating, ENT_QUOTES, "UTF-8") ?>
                            <i class="bi bi-star-fill text-warning" style="font-size: small;"></i>
                        </p>
                    <?php
                    } ?>
                    <p class="text-center my-0"><?= htmlspecialchars($film->duration, ENT_QUOTES, "UTF-8") ?></p>
                </div>
            </a>
            <?php
            } ?>
            <?>

            <!-- Flèche droite (masquée sur petit écran) -->
            <button class="scrollRight btn darkBtn btnWithBorders position-absolute end-0 translate-middle-y d-none d-md-block z-3" style="z-index:
                <i class="bi bi-chevron-right"></i>
            </button>
        </div>
    <?php
    } ?>
</div>
```

- **Interface statique : la page des détails d'un film**



Pour la partie "Accueil", qui correspond à la présentation générale du film j'utilise à nouveau les flexbox de bootstrap : une flexbox contenant 2 items : l'affiche du film positionnée sur la gauche et les infos du film sur la droite.

J'utilise la fonction "displayNames" pour formater l'affichage des genres et des noms d'acteurs/réalisateurs ; et ce afin qu'ils soient séparés par des virgules et qu'il n'y ait pas de virgule après le dernier élément.

J'englobe ces éléments dans des balises <a> afin de pouvoir naviguer vers les pages correspondantes.

Pour le sous-menu menant vers le casting et les critiques, j'utilise la classe "nav-pills" de bootstrap en plus de la classe "nav", ce qui permet d'afficher seulement le contenu dont l'utilisateur clique sur la "pills" correspondante dans le menu et ainsi de simuler de l'asynchrone ; bootstrap utilisant parfois du javascript.

Des boutons réservés aux administrateurs (modification et suppression) sont intégrés à la page si la variable php super-globale \$\_SESSION contient un utilisateur connecté et si la valeur de la clé "type" (\$\_SESSION["user"]["type"]) est "admin".

Pour la partie "Casting", j'utilise les flexbox pour afficher 2 listes horizontales : une pour les réalisateurs et une pour les d'acteurs du film.

Pour la partie "Critiques", j'utilise de nouveau les flexbox pour afficher toutes les critiques sous la forme d'une liste verticale, avec comme dernier item le formulaire pour publier des critiques.

J'ai mis en place une structure HTML similaire pour les pages de détails des acteurs et réalisateurs

Comme pour la page "Home", cette page "filmDetails" reçoit la variable "\$film", qui est de type tableau à 1 dimension. Cette variable provient du back-end.

Cette variable contient un tableau de 6 éléments: "détails", "genres", "acteurs", "réalisateurs", "reviews" et "average\_rating".

Ces 6 éléments sont constitués de:

- "détails" : Un objet qui contient des champs "id\_film", "title", ...
- "genres" : Un tableau d'objets dont chaque élément contient les champs "id\_genre", "name", ...
- "acteurs" : Un tableau d'objets dont chaque élément contient les champs "id\_actor", "name", ...
- "réalisateurs" : Un tableau d'objets dont chaque élément contient les champs "id\_director", "name", ...
- "reviews" : Un tableau d'objets dont chaque élément contient les champs "id\_review", "content", "rating", ...
- "average\_rating" : Un float qui représente la moyenne des critiques du film courant.



Les extraits de code pour cette page :

```
<?php
use App\Core\CSRFTokenManager as CSRFTokenManager;

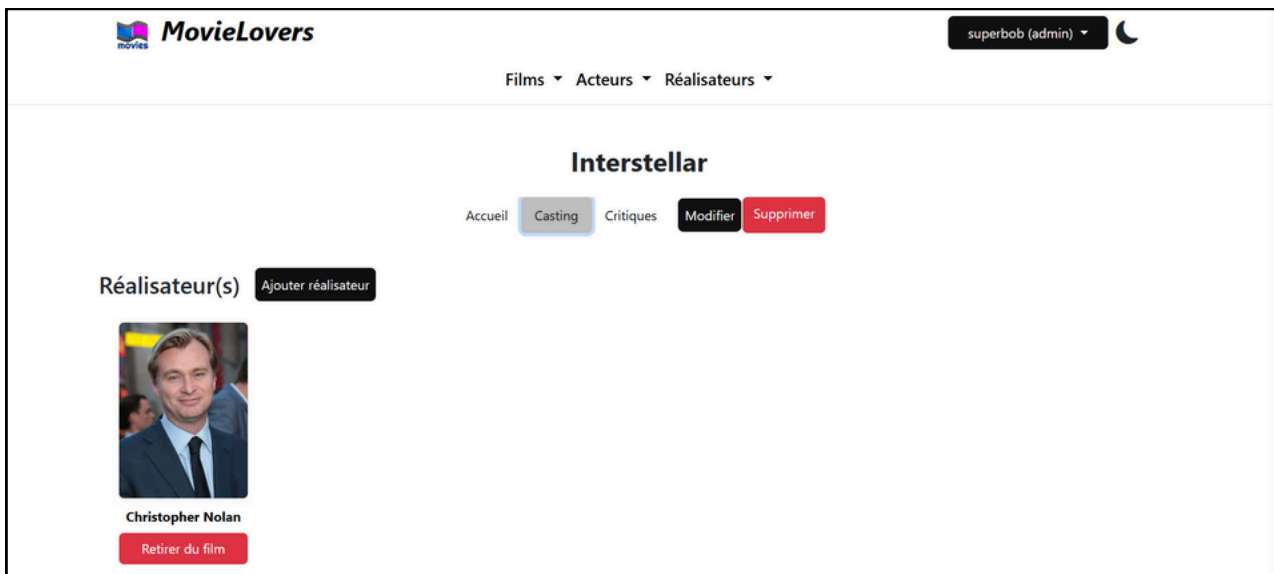
function displayNames($items, $max = null, $controller = null)
{
    $count = count($items);
    $max = $max == null ? $count : $max;

    for ($i = 0; $i < $count; $i++) {
        if ($controller == "Genre") {
            $key = "id_genre";
            $id = $items[$i]→id_genre;
        } elseif ($controller == "Actor") {
            $key = "id_actor";
            $id = $items[$i]→id_actor;
        } elseif ($controller == "Director") {
            $key = "id_director";
            $id = $items[$i]→id_director;
        }

        if ($i == $max - 1) {
            // affichage du dernier élément
            $name = $items[$i]→name; ?>
            <a href="index.php?controller=<?> $controller ?>&action=details&<?> $key ?><?> htmlspecialchars($id, ENT_QUOTES, "UTF-8") ?>"
              class="darkType menuLinks linksOnHover">
                <b><?> htmlspecialchars($name, ENT_QUOTES, "UTF-8") ?></b>
            </a>
        }
        <?php
        break;
    } else {
        // affichage du 1er à l'avant dernier élément
        $name = $items[$i]→name . ", "; ?>
        <a href="index.php?controller=<?> $controller ?>&action=details&<?> $key ?><?> htmlspecialchars($id, ENT_QUOTES, "UTF-8") ?>"
          class="darkType menuLinks linksOnHover">
            <b><?> htmlspecialchars($name, ENT_QUOTES, "UTF-8") ?></b>
        </a>
    }
}
<?php
}
```

La partie casting dans les détails d'un film avec les boutons spécifiques aux administrateurs :

*note: les acteurs sont affichés sous les réalisateurs et ne sont pas visibles ici...*



La page qui affiche la liste de tous les réalisateurs :



La version mobile de cette page avec la barre de navigation adaptée web mobile :

