**Cahier des charges : Application "Wiki Films"**

**Contexte et objectifs du projet**

L’objectif principal de l’application "Wiki Films" est de fournir une plateforme interactive et collaborative dédiée au cinéma, permettant aux utilisateurs de rechercher des informations détaillées sur des acteurs et leurs films, tout en offrant la possibilité aux utilisateurs inscrits d’enrichir la base de données avec des informations nouvelles.

**1. Fonctionnalités générales**

**1.1. Consultation des données**

* Accès à la liste des acteurs.
* Recherche d’acteurs par nom.
* Affichage du profil d’un acteur, comprenant :
* Informations personnelles (nom, date de naissance, biographie, etc.).
* Liste complète des films dans lesquels il a joué.
* Détails de chaque film (titre, synopsis, date de sortie, genre, réalisateur, etc.).

**1.2. Gestion des utilisateurs**

Différenciation des droits selon le type d’utilisateur :

* Visiteur : Peut consulter les données, mais ne peut pas les modifier.
* Utilisateur enregistré : Peut ajouter de nouveaux acteurs, biographie et filmographie.
* Administrateur : Dispose de tous les droits pour gérer les utilisateurs et les contenus.

**1.3. Sécurité**

Authentification obligatoire pour ajouter ou modifier des données.

Gestion des rôles (visiteur, utilisateur enregistré, administrateur).

Validation et modération des contenus ajoutés par les utilisateurs.

**2. Fonctionnalités détaillées par rôle**

**2.1. Visiteur (non connecté)**

* Accès public :
* Rechercher un acteur.
* Consulter les informations d’un acteur.
* Explorer les détails d’un film.
* Restriction :
* Ne peut pas modifier ni ajouter de contenu.
* Ne peut pas gérer les utilisateurs.

**2.2. Utilisateur enregistré (connecté)**

Fonctionnalités disponibles :

* Rechercher et consulter des acteurs et des films.
* Ajouter un nouvel acteur avec :
* Informations personnelles.
* Biographie.
* Liste de films (avec titre, synopsis, réalisateur, etc.).
* Ajouter un film à la filmographie d’un acteur existant.
* Restriction :
* Ne peut pas supprimer ou modifier des données existantes (uniquement des ajouts).
* Ne peut pas gérer les autres utilisateurs.

**2.3. Administrateur**

Fonctionnalités de gestion :

* Créer, modifier ou supprimer des acteurs et leurs films.
* Valider ou rejeter les ajouts effectués par les utilisateurs enregistrés.
* Gérer les comptes utilisateurs (ajouter, modifier, bloquer ou supprimer des utilisateurs).
* Surveillance :
* Accéder à un tableau de bord listant les derniers ajouts/modifications.
* Recevoir des alertes en cas de contenu signalé par d’autres utilisateurs.

**3. Architecture fonctionnelle**

**3.1. Interfaces principales**

Page d'accueil : Présentation des fonctionnalités principales avec barre de recherche.

Page acteur :

* Photo et informations personnelles.
* Biographie.
* Liste des films.

Page film :

* Affichage des détails du film.
* Liste des acteurs du film.

Page d’administration (réservée à l’administrateur) :

* Gestion des utilisateurs.
* Tableau de bord des validations de contenu.

**3.2. Système d’inscription et connexion**

Inscription : Nom, courriel, mot de passe, rôle par défaut "Utilisateur".

Connexion : Courriel et mot de passe.

Gestion des sessions : Déconnexion, récupération de mot de passe.

4. **Contraintes techniques**

**4.1. Frontend**

Technologies proposées : Larravel.

Interface utilisateur responsive, compatible mobile et desktop.

4.2. Backend

Framework : Larravel.

Base de données : MySQL

4.2 Sécurité

Gestion des rôles et permissions.

Validation des formulaires.

5. Livrables attendus

Code source versionné sur une plateforme (GitHub, GitLab).

Note : Ce cahier des charges peut être affiné selon les retours du client ou des parties prenantes du projet.

**Tables principales**

**Table users**

Représente les utilisateurs de la plateforme.

* id (PK) : Identifiant unique de l'utilisateur.
* name : Nom de l'utilisateur.
* email : Adresse email unique (pour l'authentification).
* password : Mot de passe hashé.
* role : Rôle de l'utilisateur (visitor, registered\_user, admin).
* created\_at : Date de création du compte.
* updated\_at : Date de dernière mise à jour du compte.

**Table actors**

Représente les acteurs.

* id (PK) : Identifiant unique de l'acteur.
* name : Nom complet de l'acteur.
* date\_of\_birth : Date de naissance.
* biography : Biographie de l'acteur.
* profile\_image : URL ou chemin vers la photo de profil.
* created\_by (FK vers users.id) : Utilisateur ayant ajouté cet acteur.
* validated : Booléen indiquant si l'acteur a été validé par un administrateur.
* created\_at : Date de création de l'entrée.
* updated\_at : Date de dernière mise à jour de l'entrée.

**Table movies**

Représente les films.

* id (PK) : Identifiant unique du film.
* title : Titre du film.
* synopsis : Résumé du film.
* release\_date : Date de sortie.
* genre : Genre du film (action, comédie, etc.).
* director : Nom du réalisateur.
* created\_by (FK vers users.id) : Utilisateur ayant ajouté ce film.
* validated : Booléen indiquant si le film a été validé par un administrateur.
* created\_at : Date de création de l'entrée.
* updated\_at : Date de dernière mise à jour de l'entrée.

**Table actor\_movie**

Table associative entre les acteurs et les films.

* id (PK) : Identifiant unique.
* actor\_id (FK vers actors.id) : Identifiant de l'acteur.
* movie\_id (FK vers movies.id) : Identifiant du film.

**Table user\_roles**

Gestion des permissions des utilisateurs.

* id (PK) : Identifiant unique.
* role : Nom du rôle (visitor, registered\_user, admin).
* description : Description du rôle.

**Tables secondaires**

**Table content\_reports**

Pour signaler des contenus problématiques.

* id (PK) : Identifiant unique.
* content\_type : Type de contenu (actor, movie).
* content\_id : Identifiant du contenu signalé (références actors.id ou movies.id).
* reported\_by (FK vers users.id) : Utilisateur ayant signalé.
* reason : Raison du signalement.
* created\_at : Date de signalement.

**Contraintes et relations**

* **users.role** : Référence les rôles définis dans user\_roles.
* **actors.created\_by et movies.created\_by** : Liaison avec la table users pour attribuer la création à un utilisateur.
* **actor\_movie** : Gère la relation "plusieurs à plusieurs" entre acteurs et films.
* **Validation des données** : Les champs validated dans les tables actors et movies assurent que seuls les contenus validés sont visibles publiquement.
* **Signalement** : La table content\_reports permet de surveiller les contenus signalés.

On peut retrouver la filmographie de l’acteur il suffit pour cela de sélectionner dans la base de donné la liste des films ou actor\_id es present