



MY CINEMA

< DÉVELOPPEMENT WEB AVEC PHP ET MYSQL
/ >



MY CINEMA

Présentation du projet



Un **gérant de cinéma** vient tout juste de faire l'acquisition d'un nouveau cinéma. Le **gérant** a besoin d'une application web pour gérer le fonctionnement du nouveau cinéma, il souhaite pouvoir **créer, modifier, supprimer et afficher** : les films à l'affiche, les salles de projection et le planning des séances.

Dans ce projet, vous devez créer un **back-office** pour le gérant de cinéma afin qu'il puisse gérer le mieux possible son cinéma.

L'objectif principal est de se concentrer sur la **logique back-end**, en mettant en œuvre :

- ✓ La programmation orientée objet (**POO**) en **PHP**
- ✓ Un **CRUD complet en SQL**
- ✓ Une **architecture MVC claire et maintenable**
- ✓ Une **base de données relationnelle** à partir d'un modèle fourni
- ✓ Une **API**

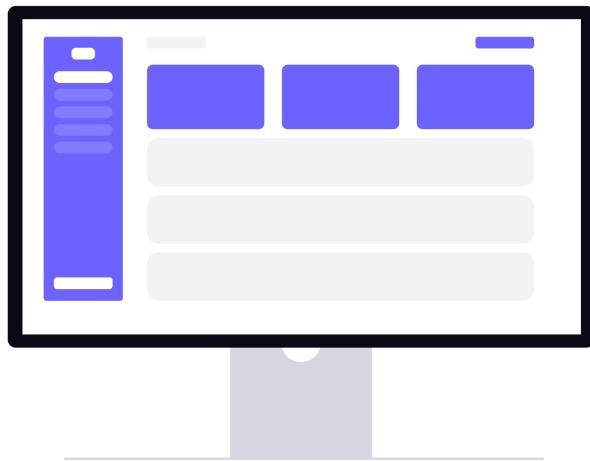
Aucun framework n'est autorisé.

Aucune maquette n'est fournie pour l'interface frontend, vous êtes libre de la mise en page souhaitée. Restez simple, mais continuez à respecter les bonnes pratiques d'accessibilité.



- ✓ Le projet doit être réalisé **sans framework PHP**.
- ✓ Toute la logique métier doit être organisée en **classes PHP** au sein d'une **architecture MVC**.
- ✓ L'ensemble du projet doit être sécurisé (requêtes préparées, vérification des saisies utilisateurs).

Fonctionnalités attendues



Vous allez développer l'application web **My Cinema**, qui est l'**interface d'administration** permettant au gérant de mettre à jour :

- ✓ Les **films**
- ✓ Les **salles**
- ✓ Le **planning des séances**

Il n'est pas attendu de faire une plateforme web à destination des clients.

Gestion des films

Le gérant souhaite disposer d'une page permettant de gérer tous les films qui ont été, sont ou seront diffusés au sein de son cinéma.

Il doit pouvoir accéder à la liste paginée des films enregistrés, ajouter un nouveau film, modifier et supprimer un film existant. La suppression ne doit être possible que pour les films sans séance associée.

Gestion des salles

Le gérant du cinéma doit également disposer d'une page permettant de gérer les salles de son cinéma. Il doit pouvoir afficher la liste de ses salles (nom/numéro, capacité, type de salle - standard, 3D, IMAX, ...), en créer, et modifier et supprimer les salles existantes.



En cas de suppression de salle, il faut penser à gérer les séances en lien avec cette salle.
Renseignez-vous sur la notion de **soft delete**.

Gestion des séances

Enfin, le gérant doit avoir une page permettant l'affichage, la modification, la suppression et la création de séance.

Une séance correspond à la diffusion d'**un film**, dans **une salle** à une **date** précise.

Contraintes importantes :

- ✓ Le gérant ne doit pas pouvoir créer **deux séances au même horaire dans une même salle**.
- ✓ La durée du film doit être prise en compte pour éviter les chevauchements.



Toute la logique de vérification des conflits doit être gérée côté back-end.

Spécifications techniques

Technologies

Le projet doit être réalisé avec **HTML, CSS et JavaScript** pour le frontend (les frameworks CSS sont autorisés). Le site doit tourner sous **PHP 8.3 minimum** (sans framework) et **MySQL. PDO** sera utilisé pour l'accès à la base de données.

Base de données

La base de données doit respecter ce modèle de données :

Le diagramme est également accessible sur DB Diagram : <https://dbdiagram.io/d/My-Cinema-69694924d6e030a0242f4404>



Attention à bien créer les **clés primaires** et les **clés étrangères** au sein de votre base de données.



Au besoin, vous pouvez ajouter des données supplémentaires, mais les tables et colonnes présentes sur le diagramme doivent être créées sans modifier la conception.

Architecture

Le projet doit être séparé en deux dossiers distincts :

- ✓ un dossier **backend** qui contiendra le code PHP et qui a pour but de faire tourner l'API développée permettant de gérer les films, les séances et les salles.
- ✓ un dossier **frontend** qui contiendra le code HTML/CSS et JavaScript et qui consommera l'API développée.

Backend :

Le backend devra contenir :

- ✓ des entités, sous forme de **classes**, représentant les données issues de la base de données
- ✓ un point d'entrée unique `index.php`
- ✓ des contrôleurs, en charge de réaliser les actions attendues (par exemple : récupération des films, création de séances, etc.).

Frontend :

Les technologies frontend sont libres. `fetch` sera utilisé pour réaliser les appels API.

Bonus (facultatif)

Fonctionnalités supplémentaires possibles :

- ✓ Filtrer les films par année ou par genre
- ✓ Afficher le planning par salle
- ✓ Système d'authentification & contrôle des accès

Livrables

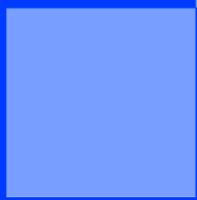
À la fin du projet, les éléments suivants devront être fournis au sein de votre dépôt GitHub :

- ✓ Le code source complet du projet
- ✓ Le script SQL `script.sql` de création de la base de données
- ✓ Un fichier `README.md` expliquant :
 - L'installation
 - La configuration de la base de données
 - Le lancement du projet

Ressources utiles

- ✓ **PHP (POO)** – <https://www.php.net/manual/fr/language.oop5.php>
- ✓ **PDO** – <https://www.php.net/manual/fr/book pdo.php>
- ✓ **SQL** – <https://sql.sh/>
- ✓ **MDN Web Docs** – <https://developer.mozilla.org/>

v 1.0.0



{EPITECH}