

Projet : Xpress-movie

Énoncé

Dans ce travail pratique, vous allez développer une application web, **Xpress-movie**, de gestion de films en utilisant le framework **Express.js**. Cette application permettra aux administrateurs d'ajouter et de gérer des films, tandis que les abonnés pourront les consulter, les ajouter à leurs favoris et aussi gérer leur compte. Ce projet vous permettra d'appliquer les concepts fondamentaux du développement web en respectant l'architecture **MVC**, en manipulant une base de données avec le package **mysql2**, et en créant des interfaces utilisateurs ergonomiques avec **EJS** et **Bootstrap**.

Objectifs pédagogiques

- Élaborer des diagrammes **UML**, notamment des diagrammes de cas d'utilisation et de classes.
- Comprendre et appliquer le modèle **MVC** avec **Express.js**.
- Gérer la persistance des données.
- Créer des interfaces utilisateurs avec **EJS** et **Bootstrap**.

Contexte

Vous devez développer une application de gestion de films destinée à deux types d'utilisateurs :

- **Administrateurs** : ajout, modification et suppression de films.
- **Abonnés** : création de compte, ajout et suppression de films favoris, gestion de compte.

Chaque utilisateur est défini par les attributs suivants :

- un **id** (de type **int**),
- un **nom** (de type **string**),
- un **prenom** (de type **string**),
- un **email** unique (de type **string**),
- un **password** (de type **string**) et
- un **role** (de type **string**) pouvant être **ADMIN** ou **ABONNE**.

Pour chaque film, nous enregistrons :

- un **id** (de type **int**),
- un **titre** unique (de type **string**),
- une **image** (de type **string**) correspondant au chemin local de l'affiche du film,
- une **description** (de type **string**),
- une **dateSortie** (de type **datetime**) et
- un **genre** (de type **string**).

Pour chaque film ajouté aux favoris, nous souhaiterions enregistrer la date et l'heure d'ajout (un attribut de type `datetime`).

L'accès à toutes les pages de l'application, hors page de connexion ou page de création de compte, nécessite une authentification. Pour s'authentifier, il faut saisir un `email` et un `password` valides. Les mots de passe doivent être hachés avant d'être stockés en base de données (<https://www.npmjs.com/package/bcrypt>).

Une fois authentifié, l'abonné est redirigé vers la page d'accueil où la liste des films disponibles sera affichée. Il pourra ainsi rechercher un film selon le titre et/ou ajouter un film à sa liste de favoris.

Un menu est affiché sur toutes les pages consultées par l'abonné, proposant les liens suivants :

- Lien vers la page d'accueil.
- Lien vers la page des films favoris (pour consulter ses films favoris ou supprimer un film de sa liste des films favoris). Lors de la suppression d'un film des favoris, une confirmation via un message d'alerte est requise.
- Lien vers la page de gestion de compte pour la modification ou la suppression du compte.
- Lien de déconnexion impliquant une redirection immédiate vers la page de connexion.

La navigation doit être intuitive et responsive grâce à **Bootstrap** (ou autre framework **CSS**). Le nom et le prénom de l'utilisateur connecté doivent être affichés sur toutes les pages de l'application au même niveau que le menu.

L'administrateur ne peut accéder qu'aux pages qui lui permettent de consulter la liste des films, ajouter un nouveau film et modifier ou supprimer un film existant. Seuls les abonnés peuvent gérer leur liste de films favoris.

Les comptes administrateurs seront insérés directement en base de données, sans interface de création/gestion prévue.

Questions

1. Élaborez un diagramme de classes et un diagramme de cas d'utilisation global.
2. Réalisez une maquette de l'application [optionnelle (CCP 2)].
3. Créez un projet **Express.js** et associez-le à un dépôt (repository) **GitHub** : chaque étape de ce projet doit être marquée par un `commit`. (CCP 1)
4. Ajoutez les dépendances nécessaires (CCP 1).
5. Transformez le diagramme de classes en MCD puis MLD [optionnelle (CCP 5)].
6. Créez puis exécutez le Script SQL de création de la base et des tables (CCP 5).
7. Configurez la base de données dans le fichier `.env` (CCP 1).
8. Préparez la couche DAL : Data Access Layer (CCP 6).
9. Ajoutez quelques validateurs (CCP 7).

10. Développez les contrôleurs (CCP 7) et les vues **EJS** nécessaires pour les fonctionnalités suivantes (CCP 3 et 4) :

- Inscription
- Connexion/déconnexion
- Modification et suppression de compte
- Affichage de tous les films disponibles dans la page d'accueil
- Ajouter un film aux favoris
- Afficher les favoris de l'utilisateur
- Supprimer un film des favoris
- Liste, ajout, modification et suppression de films pour l'administrateur

Liste des compétences professionnelles couvertes par le projet

- 1 - Installer et configurer son environnement de travail en fonction du projet web ou web mobile
- 2 - Maquetter des interfaces utilisateur web ou web mobile [question optionnelle]
- 3 - Réaliser des interfaces utilisateur statiques web ou web mobile
- 4 - Développer la partie dynamique des interfaces utilisateur web ou web mobile
- 5 - Mettre en place une base de données relationnelle
- 6 - Développer des composants d'accès aux données SQL ~~et/NoSQL~~
- 7 - Développer des composants métier coté serveur
- 8 - ~~Documenter le déploiement d'une application dynamique web ou web/mobile~~