Rapport de projet : Application de gestion de médiathèque avec Django

Etude et correctifs du code fourni :

Le code fourni était une ébauche pour une application en mode console. Plusieurs problèmes ont été identifiés :

- Redondance dans les modèles de médias :

Le code initial définissait les attributs nom et disponible dans chaque classe enfant Livre, CD et DVD.

Une classe mère Media a été ajoutée pour centraliser ces attributs communs. Les classes Livre, CD et DVD héritent désormais de cette classe Media et ne conservent que leurs attributs spécifiques auteur, realisateur ou artiste.

- Absence de modèle Emprunt :

Le code ne disposait pas de modèle Emprunt, et les attributs emprunteur et dateEmprunt se trouvaient dans les modèles Livre, CD et DVD.

Un modèle Emprunt a été ajouté avec les attributs emprunteur, media, date_emprunt et date_retour.

- Gestion du statut des membres :

L'attribut bloque prévu pour gérer les restrictions des membres a été remplacé par deux attributs distincts en_retard et emprunts_actifs, permettant d'indiquer au membre pourquoi il est bloqué et des méthodes ont été créées pour pouvoir mettre à jour ces attributs à chaque emprunt ou retour de média.

Mise en place des fonctionnalités demandées :

- Système de connexion :

Un système de connexion a été mis en place en utilisant le module django.contrib.auth. Il vérifie si l'utilisateur appartient au groupe membres ou bibliothecaires et le redirige automatiquement vers l'application correspondante, garantissant que chaque utilisateur ne peut accéder qu'aux fonctionnalités adaptées à son rôle.

Pour automatiser la gestion des utilisateurs liés aux membres, des signaux Django post_save et post_delete ont été mis en place. Ils créent automatiquement un compte utilisateur lorsqu'un membre est ajouté et le suppriment lorsque le membre est supprimé.

- Fonctionnalités accessibles aux bibliothécaires :

Gestion des médias : Possibilité d'ajouter, de modifier et de supprimer des médias ou jeux de plateau.

Gestion des membres : Possibilité d'ajouter, de modifier et de supprimer des membres.

Lors de la création d'un emprunt, la méthode peut_emprunter est utilisée pour s'assurer qu'un membre en retard ou ayant déjà 3 emprunts actifs ne peut pas emprunter.

Gestion des emprunts : Possibilité de créer un emprunt ou de le supprimer en retournant le média.

Les méthodes save et delete mettent automatiquement à jour la disponibilité du média, le nombre d'emprunts actifs du membre, et vérifie son retard éventuel. La date de retour est automatiquement fixée à 7 jours.

- Fonctionnalités accessibles aux membres emprunteurs :

Un membre emprunteur ne peut que consulter la liste des médias et ses emprunts actifs.

Stratégie de tests :

Des tests ont été mis en place avec pytest pour vérifier le bon fonctionnement des fonctionnalités demandées et un fichier avec des données test est inclus dans le projet.

Instructions d'installation et d'execution :

Cloner le projet :

git clone https://github.com/eihtac/mediatheque

cd mediatheque

Créer et activer un environnement virtuel :

python –m venv venv

venv\Scripts\activate

Installer les dépendances :

pip install –r requirements.txt

Effectuer les migrations :

python manage.py migrate

Charger les données de test :

python manage.py loaddata bibliothecaires/fixtures/donnees_test.json

Créer un superutilisateur :

python manage.py createsuperuser