

Rapport de Projet – Médiathèque Django

Le projet consistait à s'appuyer sur un code fourni pour créer une application web de médiathèque avec Django. L'objectif était de permettre la consultation et l'emprunt de livres, CDs, DVDs et jeux de plateau (uniquement en consultation). Deux interfaces ont été créées : l'espace bibliothécaire pour gérer les médias et les membres, et l'espace membre pour consulter les médias.

1. L'étude et les correctifs du code fourni

Le code initial fourni était très basique et destiné à une application en mode console. Il comportait plusieurs classes représentant les différents médias (livre, dvd, cd, jeuDePlateau) et les emprunteurs (Emprunteur). Ces classes utilisaient des attributs simples comme name, auteur, disponible, emprunteur ou dateEmprunt. Des fonctions de menu étaient également présentes (menu, menuBibliotheque, menuMembre).

Cependant, ce code présentait plusieurs limites et problèmes :

- 1. Absence de structure Django :** Le code n'était pas organisé selon les standards d'un projet Django. Il n'y avait ni modèle, ni vues, ni templates, ni gestion de la base de données.
- 2. Mauvaise utilisation des classes :** Les classes étaient définies sans constructeurs et les attributs étaient initialisés avec des chaînes vides, ce qui ne permettait pas de créer facilement des instances dynamiques.
- 3. Absence de base de données :** Il n'y avait aucun stockage dans une base de données, ce qui empêchait la réutilisation ou la consultation des informations après l'exécution du programme.
- 4. Pas de gestion des règles :** Les contraintes comme "3 emprunts maximum par membre" ou "jeux de plateau non emprantables" n'étaient pas implémentées.
- 5. Lisibilité et maintenabilité :** Le code était difficile à maintenir, avec des erreurs de syntaxe et une mauvaise indentation.

Correctifs apportés

Pour transformer ce code en une application web fonctionnelle avec Django, plusieurs changements ont été réalisés :

- **Création de modèles Django (models.py)** : Les classes Livre, DVD, CD, JeuDePlateau et Membre ont été refaites comme modèles Django avec des champs typés (CharField, BooleanField, DateTimeField). Les données sont conservées de façon permanente dans une base SQLite.
- **Création de formulaires Django (forms.py)** : Pour gérer la création et la modification des médias et des membres, des formulaires Django ont été créés (ModelForm).
- **Vues Django (views.py)** : Les fonctions menu, menuBibliotheque, menuMembre ont été remplacées par des vues web sécurisées par login, avec des fonctionnalités CRUD et d'emprunt intégrant la logique.
- **Gestion des emprunts et des contraintes métier** : La logique pour vérifier si un média est disponible, limiter les emprunts à 3 par membre, vérifier les retards, et empêcher l'emprunt des jeux de plateau a été intégrée dans les vues.
- **Templates HTML et interface utilisateur** : Les menus console ont été remplacés par des pages HTML interactives avec base.html pour la structure, et des templates spécifiques pour l'affichage des médias, des membres et des formulaires.
- **Logging** : Un système de log a été ajouté pour enregistrer toutes les opérations importantes (ajout, modification, suppression, emprunt, retour) avec date et heure.

En résumé, le code initial a été corrigé et restructuré afin de respecter les standards de Django, de permettre le stockage durable des données dans la base et d'intégrer toutes les fonctionnalités attendues pour les applications de médiathèque.

2. Mise en place des fonctionnalités demandées

L'application de la médiathèque a été développée en respectant les spécifications fournies. Deux applications distinctes ont été mises en place :

une application principale pour les bibliothécaires et une application de consultation pour les emprunteurs.

2.1. L'application bibliothécaire

L'application bibliothécaire permet de gérer les membres et les médias ainsi que les emprunts. Elle offre la possibilité de créer un membre-emprunteur, d'afficher la liste complète des membres et de mettre à jour ou supprimer un membre existant. La liste des médias est également accessible, avec toutes les informations pertinentes telles que le nom, l'auteur ou l'artiste, et leur disponibilité.

Pour la gestion des emprunts, l'application permet de créer un emprunt pour un média disponible et de le rendre lorsqu'il est retourné. Les règles ont été appliquées : un membre ne peut pas avoir plus de trois emprunts actifs à la fois, et il ne peut pas emprunter si un de ses emprunts est en retard. Les jeux de plateau, ne sont pas empruntables et restent uniquement disponibles pour la consultation.

L'application permet aussi d'ajouter, modifier et supprimer des médias, avec un CRUD pour tous les types de médias (livres, CDs, DVDs, jeux de plateau). Pour améliorer la traçabilité et faciliter la maintenance, un système de logs a été intégré : chaque opération sur un média ou un membre est enregistrée dans un fichier de log avec la date et l'heure. De plus, des messages clairs de confirmation ou d'erreur sont affichés à l'utilisateur après chaque action.

2.2. L'application de consultation

L'application de consultation est accessible à tous les utilisateurs et permet uniquement de visualiser la liste de tous les médias. Chaque média indique sa disponibilité : les livres, CDs et DVDs peuvent être empruntés, tandis que les jeux de plateau sont signalés comme uniquement consultables. Cette distinction garantit que les règles sont respectées dans l'interface publique.

3. Stratégie de tests

Une stratégie de tests a été définie afin de vérifier le bon fonctionnement des fonctionnalités. Des tests unitaires ont été réalisés pour contrôler que les jeux de plateau ne pouvaient pas être empruntés, que les membres avec un emprunt en retard ne pouvaient pas effectuer de nouvel emprunt et que la limite de trois emprunts par membre était respectée. D'autres tests ont vérifié que les retours mettaient correctement à jour la disponibilité des médias. Ces

tests ont été exécutés avec la commande **python3 manage.py test** et ont permis de vérifier automatiquement que toutes les fonctionnalités du site fonctionnent.

4. Une base de données avec des données test

Une base de données SQLite a été créée pour faciliter l'installation et les tests. Elle contient la structure nécessaire pour stocker les médias, les membres et les emprunts. Il est possible d'ajouter des données de test pour simuler différents scénarios et vérifier le fonctionnement de toutes les fonctionnalités. La base de données est sécurisée via un mot de passe stocké dans une variable d'environnement .env.

5. Instructions pour exécuter le programme

1. Cloner le projet

```
git clone https://github.com/Rromanee/Me-diathe-que-Django  
cd mediatheque
```

2. Installer Python 3 et Django si nécessaire :

```
pip install django
```

3. Créer la base de données et appliquer les migrations

```
python manage.py makemigrations  
python manage.py migrate
```

4. Lancer le serveur Django :

```
python manage.py runserver
```

5. Accéder à l'application

- Espace membre : <http://127.0.0.1:8000/medias/>
- Espace bibliothécaire : <http://127.0.0.1:8000/bibliothecaire/> (login requis)
- Espace Django administration : <http://127.0.0.1:8000/admin/> (login requis)

Login:

Nom d'utilisateur : notrelivrenotremedia

Mot de passe : notremedianotrelivre