



Créer un logiciel de gestion de médiathèque

Avec Django

Sommaire

- 1.Introduction
- 2.Étude et Correctifs du Code Fourni
- 3.Mise en Place des Fonctionnalités Demandées
- 4.Stratégie de Tests
- 5.Instructions pour Exécuter le Programme

1. Introduction

La médiathèque propose une variété de supports à la consultation et à l'emprunt, notamment des livres, des CDs, des jeux de plateau et des DVDs. Tous ces supports sont disponibles à la consultation et à l'emprunt, à l'exception des jeux de plateau qui sont réservés à la consultation uniquement.

Le projet comprend le déploiement de deux applications distinctes :

Une application principale réservée aux bibliothécaires, offrant des fonctionnalités telles que la gestion des membres, la gestion des médias, la gestion des emprunts, et plus encore.

Une application de consultation destinée aux emprunteurs, permettant uniquement d'afficher la liste de tous les médias disponibles.

Dans le cadre de ce projet, il est impératif de respecter certaines contraintes métiers :

Limiter à trois le nombre maximum d'emprunts par membre.

Exiger le retour des emprunts dans un délai d'une semaine.

Empêcher les membres ayant des emprunts en retard d'emprunter de nouveaux médias.

Exclure les jeux de plateau du processus d'emprunt.

Ces directives définissent les bases sur lesquelles je travaillerai pour répondre aux besoins spécifiques de la médiathèque "Notre livre, notre média" et pour garantir le succès de ce projet de modernisation.

2. Étude et Correctifs du Code Fourni

Analyse du code fourni

Un développeur a commencé à réfléchir sur les classes model.
Nous allons reprendre ce
qui a été fait et l'améliorer et l'adapter pour pouvoir l'utiliser
avec Django.

Analyse du code

```
def menu():
    print("menu")

if __name__ == '__main__':
    menu()

class livre():
    name = ""
    auteur = ""
    dateEmprunt = ""
    disponible = ""
    emprunteur = ""

class dvd():
    name = ""
    realisateur = ""
    dateEmprunt = ""
    disponible = ""
    emprunteur = ""

class cd():
    name = ""
    artiste = ""
    dateEmprunt = ""
    disponible = ""
    emprunteur = ""

class jeuDePlateau :
    name = ""
    createur = ""

class Emprunteur():
    name = ""
    bloque = ""

def menuBibliotheque() :
    print("c'est le menu de l'application des bibliothécaire")

def menuMembre():
    print("c'est le menu de l'application des membres")
    print("affiche tout")
```

Les fonctions de menu (menu, menuBibliothèque, menuMembre) seront gérées par les views au sein de Django, nous n'allons donc pas les conserver.

– Le script : `if __name__ == '__main__':`

`menu()`

sera intégré au dossier `manage.py` du projet Django et sera adapté :

`if __name__ == '__main__':`

`main()`

– Concernant les classes, plusieurs défauts sont à corriger :

→ Les noms des classes doivent commencer par une majuscule, or seule la classe `Emprunteur` répond à cette contrainte.

→ pour déclarer une classe, les parenthèses ne sont nécessaire que dans les cas d'héritage.

→ En langage Python, l'indentation a un rôle important pour l'exécution du code, or les attributs « `realisateur` », « `dateEmprunt` », « `disponible` » de la classe « `Dvd` » et l'attribut « `bloque` » de la classe « `Emprunteur` » présentent des erreurs d'indentation (4 espaces en trop).

→ Enfin, afin de ne pas répéter des éléments de code, nous pourrons utiliser l'héritage lors de la création des classes ; en effet, les classes « `Livre` », « `Dvd` » et « `Cd` » ont en commun les attributs « `name` », « `dateEmprunt` », « `disponible` » et « `emprunteur` ». Nous allons donc créer une classe « `Média` » qui aura ces attributs, puis nous utiliserons l'héritage pour créer les classes « `Livre` », « `Dvd` » et « `Cd` », en ajoutant dans chaque classe les attributs qui lui sont propres.

3. Mise en Place des Fonctionnalités Demandées

J'ai défini des modèles Django pour représenter les membres, les médias (livres, DVD, CD), les emprunteurs, etc.

J'ai utilisé l'héritage pour organiser les différents types de médias.

Mis en place des vues pour gérer les fonctionnalités de l'application : ajout de membres, affichage des médias, gestion des emprunts, etc.

Géré l'authentification des utilisateurs et la session.

Fonctionnalités implémentées :

J'ai implémenté la création de membres, l'affichage des membres, la création d'emprunts, l'affichage des médias disponibles, j'ai aussi créer un tableau à la page d'accueil pour voir les emprunts ,une mise à jour des membres, la suppression des membres etc.

Géré les limitations d'emprunts par membre et la vérification des emprunts en retard.

4. Stratégie de Tests

Afin de tester les différentes fonctionnalités du logiciel, j'ai utilisé pytest et j'ai mis mes fichiers de test dans mon dossier `gestion_bibliothecaire` mes trois fichiers sont: le fichier `test_models.py`, le fichier `test_urls.py` et le fichier `test_views.py`. À l'intérieur de ces fichiers, les fonctions de test ont toutes un nom commençant par `test_(...)` afin d'être bien détectées par pytest et j'ai exécuter cette commande dans la console pour le vérifier : `pytest --ds=gestion_mediatheque.settings`.

```
PS C:\Users\kanth\gestion_mediatheque> pytest --ds=gestion_mediatheque.settings
===== 30 passed in 12.90s =====
===== test session starts =====
platform win32 -- Python 3.12.2, pytest-8.0.2, pluggy-1.4.0
django: version: 5.0.2, settings: gestion_mediatheque.settings (from option)
rootdir: C:\Users\kanth\gestion_mediatheque
plugins: django-4.8.0
collected 44 items

gestion_bibliothecaire\test_models.py ..... [ 13%]
gestion_bibliothecaire\test_urls.py ..... [ 59%]
gestion_bibliothecaire\test_views.py ..... [100%]

===== 44 passed in 13.31s =====
PS C:\Users\kanth\gestion_mediatheque> |
```

5. Instructions pour Exécuter le Programme

Instructions concernant l'exécution du programme

Afin de pouvoir utiliser ce logiciel, vous devez exécuter les étapes suivantes :

- Installation de Python sur votre ordinateur.

Le logiciel a été créé avec Python, il faut donc installer cette version de Python. Vous pouvez télécharger Python à partir du site officiel : <https://www.python.org/downloads/>

- Téléchargement du logiciel :

Le logiciel se trouve sur ce site : [DKanthio/gestion_mediatheque \(github.com\)](https://github.com/DKanthio/gestion_mediatheque)

Pour le récupérer, vous devez cliquer sur Download zip

Il faut ensuite ouvrir le dossier avec un IDE (PyCharm, VSCode...)

- Afin d'isoler les dépendances du projet, vous devez créer un environnement virtuel : dans le terminal de votre IDE, tapez la ligne de commande :

« python -m venv env »

puis pour activer l'environnement virtuel :

« source env/bin/activate »

- Allez ensuite dans le dossier « mediatheque » :

Cd gestion_mediatheque

Puis lancez le serveur de développement :

python manage.py runserver

et cliquez sur le lien pour ouvrir le site :

Vous pouvez vous connecter à l'interface d'administration du site en utilisant ce lien :

<http://127.0.0.1:8000/admin/> et les identifiants :

Nom d'utilisateur : Kanthio

Mot de passe : Maman51936757