Projet :

Créer un logiciel

de gestion

de médiathèque

Python

DjangoDB

Table des matières

Le projet

La médiathèque « Notre livre, notre média » nous a contacté afin de moderniser son système de gestion interne qui est pour l'instant géré à l'ancienne par des fiches papiers.
L'objectif est d'avoir un système fonctionnel, sécurisé, utilisable par les bibliothécaires et également par le public.

La médiathèque propose à la consultation et à l'emprunt des livres, des CDs, des jeux de plateaux et des DVD.
Tous ces supports sont ouverts à la consultation et à l'emprunt, sauf les jeux de plateau qui sont uniquement disponibles en consultation.

Il y aura deux applications à déployer :
! - Une application principale qui ne sera accessible qu'aux bibliothécaires.
! Elle devra permettre de :
! Créer un membre-emprunteur.
! Afficher la liste des membres.
! Mettre à jour un membre.
! Afficher la liste des médias.
! Créer un emprunt pour un média disponible.
! Ajouter un media.
! Rentrer un emprunt.

! - Une application de consultation accessible aux emprunteurs.

! Elle doit permettre uniquement d'afficher la liste de tous les médias.

Voici les contraintes à respecter :
! Un membre ne peut pas avoir plus de 3 emprunts à la fois.
! Un emprunt doit être retourné au bout d'1 semaine.
! Un membre ayant un emprunt en retard ne peut plus emprunter.
! Les jeux de plateaux ne sont pas concernés par les emprunts.

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.

```
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 bibliothéquaire")




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 parenthèse 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.

Mise en place des fonctionnalités

Afin de mettre en place le projet Médiathèque, après la création du projet Django, j'ai créé plusieurs apps permettant la gestion des différentes fonctions :

- Une app « authentication » pour gérer la connection au site.
- Une app « users » qui contient le model User, le form associé et les views permettant la création, la mise à jour et la suppression des utilisateurs, ainsi que les urls permettant d'afficher les templates contenant les fonctionnalités.
- Une app « media » qui contient les models Media, Book, Dvd et Cd, les forms associés et les views permettant la création, la mise à jour et la suppression des media, ainsi que les urls permettant d'afficher les templates contenant les fonctionnalités.
- Une app « loans » qui contient le model Loan, le form associé et les views permettant l'enregistrement des emprunts dans le respect des contraintes métiers (pas plus de 3 emprunts par utilisateur, emprunt bloqué en cas de retard de plus de 7 jours).

Stratégie de tests

Afin de tester les différentes fonctionnalités du logiciel, j'ai utilisé pytest et créé un dossier « test » regroupant les fichiers de test. J'ai organisé ce dossier en trois fichiers : 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.

Instructions concernant l'exécution du programme

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

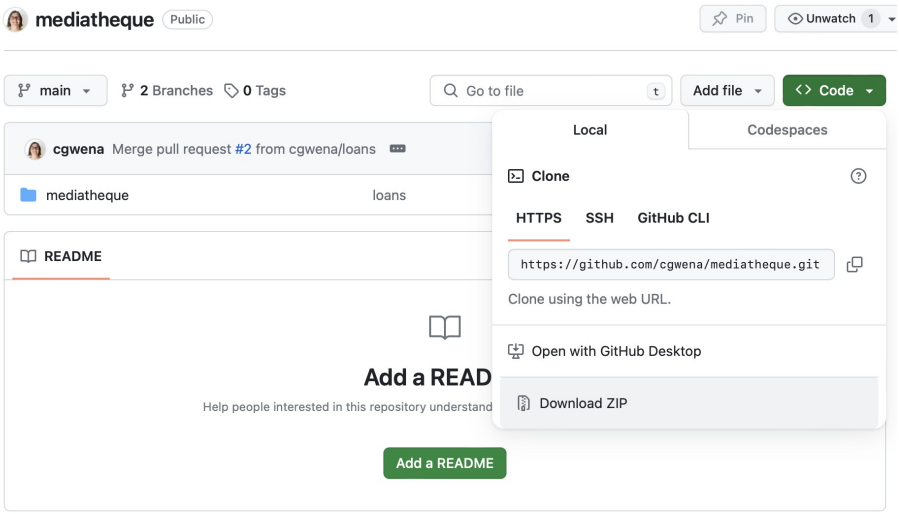
– Installation de Python sur votre ordinateur.

Le logiciel a été créé avec Python3, 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 : https://github.com/cgwena/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 :

« python3 -m venv env »

puis pour activer l'environnement virtuel :

« source env/bin/activate »

– Allez ensuite dans le dossier « mediatheque » :

cd mediatheque

Puis lancez le serveur de développement :

python3 manage.py runserver

et cliquez sur le lien pour ouvrir le site :

```
System check identified no issues (0 silenced).
February 09, 2024 - 16:29:53
Django version 5.0.2, using settings 'mediatheque.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

– 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 : Admin

Mot de passe : Adminmediatheque123