

Projet CEF:

Création d'une app de gestion de médiathèque à l'aide du framework Django Langage Python Réalisation: MASCLET Clément

Projet

Le projet médiathèque doit pouvoir proposer sur l'application bibliothécaire:

- Création d'un média
- Affichage de la liste des médias
- Emprunt d'un média
- Retour d'un média
- Création d'un membre emprunteur
- Consultation des membres
- Ajout et suppression d'un membre

Et une application consultation permettant:

- La consultation d'un média

Les contraintes:

- Un membre ne peut emprunter que 3 médias à la fois
- Un emprunt doit être retourné sous 7 jours
- En cas de retard, blocage des emprunts pour le membre
- Les jeux de plateau ne peuvent pas être empruntés

Code à reprendre

```
def menu():  
    print("menu")  
  
if __name__ == '__main__':  
    menu()
```

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

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

Erreur ici, trop d'espace dans l'indentation

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

```
class jeuDePlateau :  
    name = ""  
    createur = ""
```

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

Erreur ici, trop d'espace dans l'indentation

```
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 suivantes ne seront pas conservés

```
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")
```

Pour le reste,

Les classes cd, livres et Dvd hériteront d'une classe Media

```
class Media(models.Model):
    name = models.CharField(max_length=155)
    disponible = models.BooleanField(default=True)
    date_emprunt = models.DateField(null=True, blank=True)
    date_retour = models.DateField(null=True, blank=True)
    emprunteur = models.ForeignKey(Emprunteur, on_delete=models.SET_NULL, null=True, blank=True, related_name='media_emprunt')
```

Toutes les classes seront réécrites car doivent commencer par une Majuscule

L'héritage se fera par les majuscules à la déclaration de la classe héritant de la classe Media

```
class Dvd(Media):
    realisateur = models.CharField(max_length=155)
```

Mise en place des fonctionnalités

Après la création du projet Django, création de deux app distinctes:

- App bibliothécaire
 - Permettant création, modification et suppression d'un membre
 - création et retour d'un emprunt
 - création et consultation d'un média

Celle ci contient les Models suivant:

- Emprunteur
- Media et les models héritant de cette classe:
 - Livre
 - Cd
 - Dvd
- JeuDePlateau

Les forms ont été écrits en conséquences

Elle contient aussi les views permettant les fonctionnalités listées.

- App Consultation
 - Ne permet que la consultation des médias créés dans l'app bibliothécaire

Stratégie des tests

Les tests unitaires sont utilisé pour tester le comportement des modèles (Media, Livre, Cd, Dvd, Emprunteur).

Chaque modèle est testé pour s'assurer que les méthodes de gestion des emprunts (emprunter, rendre, etc.) fonctionnent correctement et que les contraintes de modèle sont respectées.

Chaque méthode de test utilise la méthode 'setUp' pour initialiser un environnement de test propre avant l'exécution afin de garantir l'isolement d'un test des autres et ne dépend pas de l'état des autres.

Lancement du projet

1. Installer Python

Installer Python, téléchargeable depuis ce lien: <https://www.python.org/downloads/>

2. Cloner le repository

Ouvrir un terminal et utiliser la commande suivante :






```
git clone https://github.com/Frostbowl/projet_django.git
```

Depuis le terminal, créer un environnement virtuel (Venv) à l'aide des commandes suivantes:

```
projet_django\projet_django>cd Scripts  
projet_django\projet_django\Scripts>activate
```

3. Ouvrir le projet sur un IDE

Ouvrir le fichier "mediatheque" sous un IDE prenant en charge le langage Python

 Include	04/07/2024 20:38	Dossier de fichiers	
 Lib	04/07/2024 20:38	Dossier de fichiers	
 mediatheque	08/07/2024 22:38	Dossier de fichiers	
 Scripts	04/07/2024 20:39	Dossier de fichiers	
 pyvenv	04/07/2024 20:38	Fichier source Co...	1 Ko

4. Lancer le serveur de développement

Démarrer le serveur avec la commande suivant

```
projet_django\mediatheque> python manage.py runserver
```

Puis ouvrir le liens donné dans le terminal

```
Django version 5.0.6, using settings 'mediatheque.settings'  
Starting development server at http://127.0.0.1:8000/  
Quit the server with CTRL-BREAK.
```

Pour accéder à l'app de consultation, écrire l'URL suivante:
<http://127.0.0.1:8000/consultation/medias/>