Notre livre, notre média

RAPPORT

SOMMAIRE

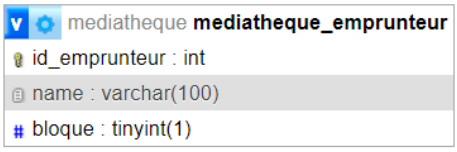
1. Introduction

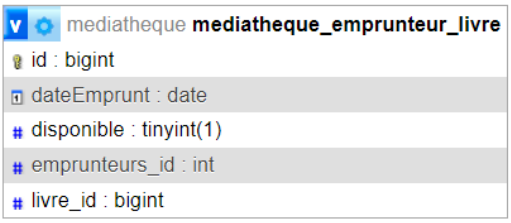
Ce projet consiste à développer un système de gestion de médiathèque utilisant le framework Django en Python. L'objectif principal est de permettre la gestion des emprunteurs ainsi que des différents types de médias disponibles dans la médiathèque tels que les livres, les DVD, les CD et les jeux de plateau. Dans ce rapport, je vais détailler la démarche de travail, les choix techniques effectués et la mise en place du projet.

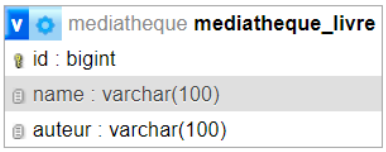
2. Base de données

J'ai dû créer des tables intermédiaires afin de réaliser les liens entre l'emprunteur et le médias.

Ex :







mediatheque_emprunteur sera utiliser pour enregistrer les noms des emprunteurs
mediatheque_livre sera utiliser pour enregistrer les livre à emprunter
mediatheque_emprunteur_livre permettra de stocker les livres emprunté par l'emprunteur

3. Code utilisé

3.1. Application médiathèque

Dans le fichier models.py, je crée les class qui seront générer dans mon SGBD via une commande django

python manage.py makemigrations
python manage.py migrate

ex :

```
class Emprunteur(models.Model):
    id_emprunteur = models.AutoField(primary_key=True)
    name = models.CharField(max_length=100)
    bloque = models.BooleanField(null=True)
```

Dans la table Emprunteur, il y aura 2 éléments indispensable

- Id emprunteur, est la clé primaire, auto incrémentée
- Name, de type chaine de caractère d'une longueur de 100 caractères

```
class Livre(models.Model):
    name = models.CharField(max_length=100, null=True)
    auteur = models.CharField(max_length=100, null=True)

    def __str__(self):
        return self.name
```

Dans la table Livre, il y a 2 éléments

- Name et auteur, de type chaine de caractère d'une longueur de 100 caractères
- J'utilise également une méthode **def __str__(self)** qui renvoi le nom du livre lorsque celle si est converti en chaine de caractère.

```
class Emprunteur_Livre(models.Model):
    dateEmprunt = models.DateField(default=timezone.now)
    disponible = models.BooleanField(null=True)
    emprunteurs = models.ForeignKey(Emprunteur, on_delete=models.CASCADE)
    livre = models.ForeignKey(Livre, on_delete=models.CASCADE, default=None)
```

Dans la table intermédiaire Emprunteur_Livre, je crée différents éléments, dont les plus importantes, emprunteurs et livre qui sont tout 2 des clés étrangères. Elles permettront de réaliser le lien entre la table Emprunteur et Livre.

Il y a quelque contrainte pour le retour des médias comme empêcher l'emprunt de plus de 3 médias.
Au début, je pensais faire un calcul simple, mais à la validation du formulaire, la variable total_medias était égale à 0.

Je procède donc autrement :

```
livres_selectionnes = form.cleaned_data['livres']
dvds_selectionnes = form.cleaned_data['dvds']
cds_selectionnes = form.cleaned_data['cds']
jeux_selectionnes = form.cleaned_data['jeux']

total_medias = (len(livres_selectionnes)+len(dvds_selectionnes)+len(cds_selectionnes))
```

La fonction **len()** me retourne la nombre de médias sélectionné, ce qui me permet d'effectuer une somme. Et par la suite de réaliser une condition pour afficher un message.

4. Eléments utiles

Lien du projet
https://github.com/tealc94/python/

Pour la récupération du dépôt
git clone https://github.com/tealc94/python.git

Url pour le bibliothécaire
http://127.0.0.1:8000/

Url pour les membres
http://127.0.0.1:8000/membre/

Un backup de la sauvegarde avec des données de test se trouve dans le dossier **bdd** de github.

Les identifiants de connexion sont :
id : django
mdp : zqg5RHY_dhw1npw7fjg