

1/ Présentation du 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 « 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 d'avoir un système fonctionnel, sécurisé, utilisable par les bibliothécaires et également par le public.

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 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 plateau 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

.Une application de consultation accessible aux emprunteurs

2/ Etudes et correctif du code fourni

Dans ce projet un développeur avait déjà commencé un code. Afin d'avoir un code propre j'ai analysé le code et corrigé ce qui devait être corrigé.

Le script

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

```
    menu()
```

n'a pas besoin d'être déclaré car il est déjà déclaré dans le fichier manage.py lors de la création du projet. Je l'ai donc pas réutilisé.

les fonctions "def menu()", "def menuBibliotheque()" et "def menuMembre()". Ne sont pas à conserver car ils vont être gérés par des views

Les noms des classes doivent toutes commencer par une majuscule

Dans le langage python, l'indentation joue un rôle important mais dans la classe "dvd", les attributs "réalisateurs", "dateEmprunt" et "disponible" ainsi que dans la classe "Emprunteur" pour l'attribut "bloque", ont une indentation de trop.

afin de raccourcir le code et de ne pas se répéter, j'ai regroupé tout les médias dans une liste media et modifier "auteur", "réalisateur", "artiste" et "createur" par "creator"

3/ Mise en place des fonctionnalités demandées

j'ai commencé par la création du projet "mediatheque". ensuite j'ai installé django et je l'ai importé dans le projet. j'ai ensuite créé une application "bibliotheque". j'ai utilisé la partie "mediatheque" pour la partie membre afin d'afficher uniquement et la liste de tous les médias et la page de connexion. Et la partie "bibliotheque" pour les afficher les autres pages demandées

4/ Stratégie de tests

pour les tests, j'ai utilisé pytest afin de vérifier mes fonctions; j'ai donc créé des fichiers test_models.py et test_views.py. Dans ces fichiers toutes les fonctions test commencent par "test_" afin que pytest fonctionne correctement. j'ai aussi créé un fichier "pytest.ini", contenant:

```
[pytest]
```

```
DJANGO_SETTINGS_MODULE= mediatheque.settings
```

afin de spécifier l'app à utiliser.

5/ Base de données avec des données test

pour la base de données j'ai créé plusieurs membres ainsi que plusieurs médias afin de tester réellement mon site

6/ Instruction pour exécuter le programme

Afin de pouvoir exécuter le programme ? vous devez suivre les étapes suivantes:

- .installez python sur votre ordinateur
- .télécharger le programme dur github
- .une fois le fichier télécharger extraire le fichier du fichier .zip
- .ouvrir le fichier avec IDE(Environnement de développement)
- .creer un environnement virtuel:python -m venv env . puis activer l'environnement virtuel tapez:env/scripts/activate
- .ouvrir un nouveau terminal aller dans le dossier mediatheque et ensuite tapez python manage.py runserver
- .cliquer sur le lien
- .indendifiant:bibliothecaire
- .mot de passe:mediatheque