

---

# Créer un logiciel de gestion de médiathèque

13 novembre 2024

---

## Stacks

Python / DjangoDB

## Récap

1. Le projet
  2. Analyse du code fourni
  3. Mise en place des fonctionnalités
  4. Stratégie de test
  5. installations
-

## 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 dvds. Tous ces supports sont ouverts à la consultation et à l'emprunt, sauf les jeux de plateau qui sont uniquement disponibles en consultation.

## ANALYSE DU CODE FOURNI

Voici la listes des points que nous avons notifié:

- Plusieurs normes de PEP8 ne sont pas respectées (nomination des classes devant être écrite en PascalCase et ne devrait avoir aucune parenthèse car aucun héritage/ indentation de quatre espace non respecté dans la classe dvd et emprunteur / organisation général du fichier / redondance et pertinence des données etc..).
- Les méthodes concernant les menus (menu / menuBibliothèque / menuMembre) seront gérées par le système de vues(views) de Django.
- Le script de lancement sera intégré et remanié par le fichier manage.py de l'application générale.

## MISE EN PLACE DES FONCTIONNALITÉS

Plusieurs application sont disponibles:

- une app "authent" pour gérer la connexion du site.

- une app "borrowing" qui gère les emprunts dans le respect des contraintes imposées. Contient le model Borrowing, le form associé et les views pour l'enregistrement des emprunts.
- une app "media" pour gérer les différents médias. Contient les models Media, Book, Cd et Dvd, les forms associés et les views pour la création, suppression et m à j des dit medias.
- une app "users" qui gère les utilisateurs. Contient le model User, les forms associés et les views pour la création, suppression et m à j des dit medias.

## STRATÉGIE DE TEST


J'ai utilisé le package [pytest-django](#). Un fichier d'initialisation à été créé, j'utilise le décorateur `@pytest.fixture` pour simuler des scénarios réalistes. Les tests sont concluants.

```
plugins: django-4.9.0
collected 52 items

tests\test_models.py ..... [ 11%]
tests\test_urls.py ..... [ 23%]
tests\test_views.py ..... [ 59%]
tests\test_urls.py ..... [100%]
```

## Instruction

- Installation de python 3.13 sur votre ordinateur (<https://www.python.org/downloads/>)
- Téléchargement du logiciel sur: [https://github.com/NikoNiiOS/mediatheque\\_poo](https://github.com/NikoNiiOS/mediatheque_poo)

- 
- Il faut isoler les dépendances du projet, pour cela, il faut créer un environnement virtuel. dans le terminal de votre IDE utilisez la ligne de commande suivante: `python3 -m venv env`
  - activé ensuite l'environnement: `source env/bin/activate`
  - allez dans le dossier mediatheque : `cd mediatheque`
  - lancé le le serveur de dev: `python manage.py runserver`
  - cliquez sur le lien affiché dans la console : <http://127.0.0.1:8000/>