



« Notre livre, notre média »

SOMMAIRE

- 1. INTRODUCTION
- 2. REPRISE DU CODE EXISTANT
- 3. CHOIX TECHNIQUES
- 4. MISE EN PLACE
- 5. **DEVELOPPEMENT**
- 6. CONTRAINTE
- 7. VUE DU SITE
- 8. CONCLUSION
- 9. ANNEXES

Rapport de Projet : Gestion de Médiathèque

1.Introduction

Contexte du projet : Le projet consiste à développer un logiciel de gestion de médiathèque pour permettre la consultation et l'emprunt de divers médias tels que des livres, des CDs, des jeux de plateau et des DVDs. Deux applications seront déployées : une pour les bibliothécaires et une pour les emprunteurs.

Objectifs:

- Développer dans un langage objet
- ☑ Développer la partie dynamique de l'application avec des composants servers, dans un style défensif, et éventuellement en asynchrone
- Appeler des Web Services dans un composant serveur
- ☑ Utiliser des composant d'accès aux données
- Réaliser un jeu de tests de l'application web

P1

Analyse du code existant: Le code existant présente plusieurs problèmes, tant au niveau du style que de la logique, surtout si l'on considère qu'il est destiné à être utilisé dans une application Django.

```
def menu():
    print("menu")

if __name__ == '__main__':
    menu()

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

class cd():
        name = ""
        artiste = ""
        dateEmprunt = ""
        def menufeur = ""

class jeuDePlateau :
        name = ""
        createur = ""

class Emprunteur():
        name = ""
        def menuMembre():
        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 méthodes menu permet juste d'afficher du texte dans la console et donc n'as aucune interaction avec le reste du code ni pour l'affichage sur un site l'utilisation d'un template en html est bien meilleur.
- ☑ Absence des models pour les class des medias et de l'emprunteur
- ☑ La méthode « if __name__ == '__main__' » et utiliser en python pour des scripts autonomes

Pour résumer le code nous a servie a reprendre les différent class et les modifier pour rendre la class compatible a django pour les models

Technologies utilisées:

- Python : Langage principal pour le développement.
- Django : Framework web pour structurer l'application.
- ☑ Git : Pour le contrôle de version et la collaboration.

Architecture du projet :

- Dossiers: mediatheque/ contient les fichiers principaux, bibliothecaire et membre pour l'utilisation des 2 application. Fait avec la commande: django-admin startapp
- Fichiers: manage.py pour les commandes de gestion, requirements.txt pour les dépendances.

Dépendances :

 Les dépendances sont listées dans le fichier requirements.txt et incluent des bibliothèques comme Django.

Installation et configuration :

1. Cloner le dépôt :

git clone https://github.com/greggit49/my-mediath-python.git cd mediatheque git remote remove origin

2. Installer les dépendances :

pip3 install -r requirements.txt - Unix/macOS pip install -r requirements.txt - Windows

3. Démarrer le serveur de développement :

cd mediatheque
python3 manage.py runserver - Unix/macOS
python manage.py runserver - Windows

5. Développement

Fonctionnalités développées :

- Application bibliothécaire : Création et gestion des membres, gestion des emprunts et des retours, ajout de nouveaux médias.
- Application membre : Consultation de la liste des médias disponibles.

Tests:

- Tests unitaires pour vérifier le bon fonctionnement des fonctionnalités.
- Tests d'intégration pour s'assurer que les différentes parties de l'application fonctionnent bien ensemble.
- Commande : python manage.py test

Ρ4

6. Contraintes

Contraintes métiers respecter :

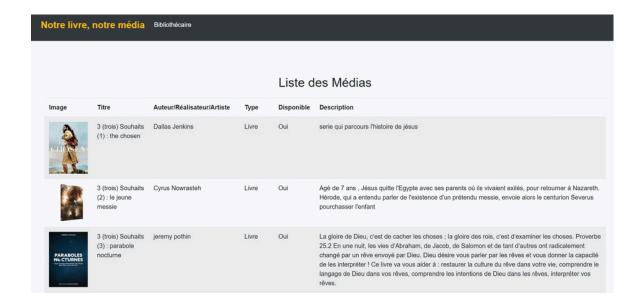
- Limite de 3 emprunts par membre.
- Durée d'emprunt d'une semaine.

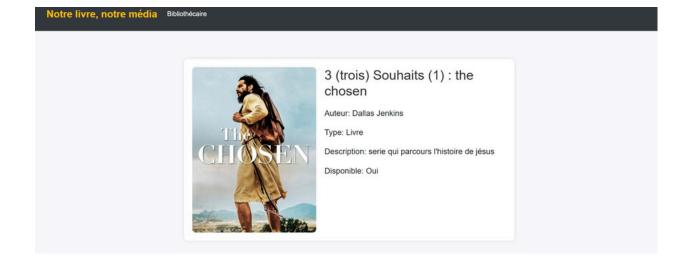
- Interdiction d'emprunter pour les membres ayant des retards.
- Les jeux de plateau ne peuvent pas être empruntés.

7. Vue du site

Application Membre

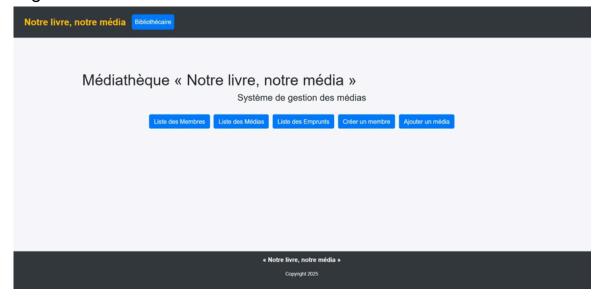
Page d'accueil:



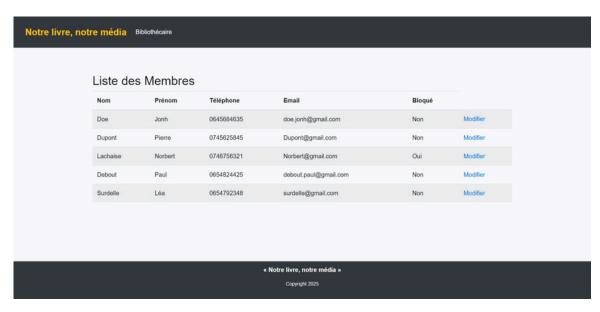


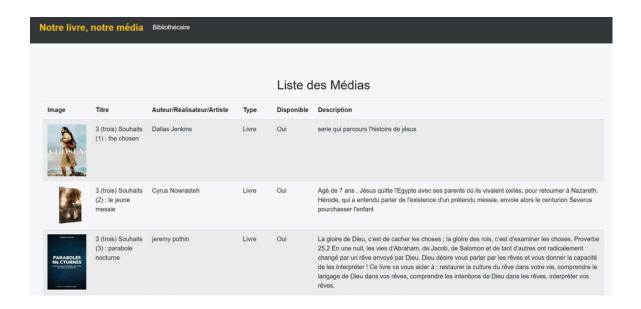
Application Bibliothécaire

Page d'accueil:

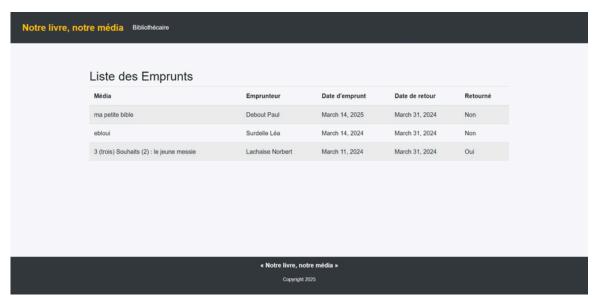


Page de la liste des membres :

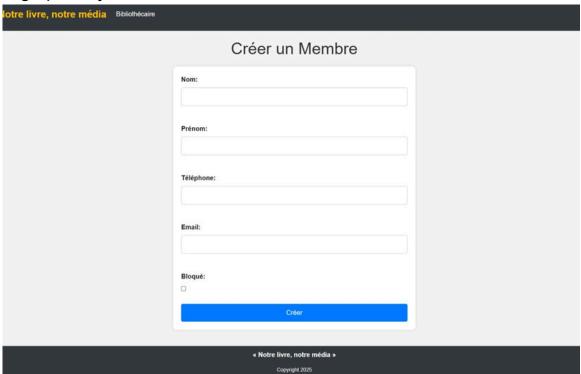


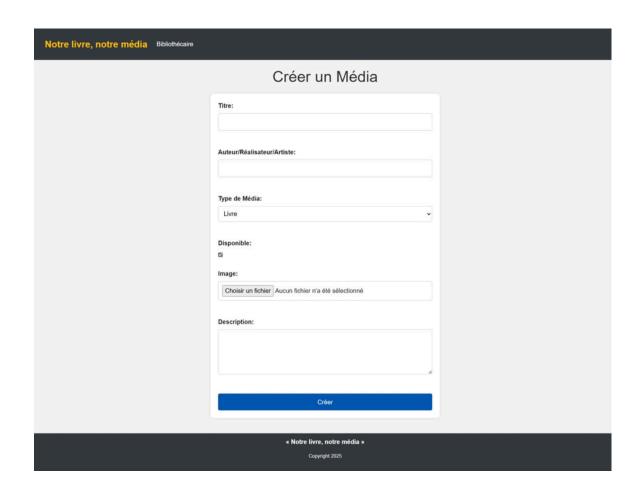


Page de la liste des emprunts



Page pour ajouter un membre





Résumé : Le projet a permis de développer une application de gestion de médiathèque répondant aux besoins spécifiques des bibliothécaires et des emprunteurs.

Perspectives:

- ☑ Amélioration de l'interface utilisateur.
- ☑ Ajout de nouvelles fonctionnalités comme la réservation de médias.

9. Annexes

Code source : utilisation du code fourni

Documentation : image et documentation des médias