**Rapport de Travail sur le Projet de Médiathèque**

**1. Étude et Correctifs du Code Fourni**

**Analyse Initiale :**

* Le code initial comportait des erreurs dans la configuration des URL et des vues.
* Les fonctionnalités de connexion et de déconnexion n'étaient pas correctement configurées, ce qui empêchait l'accès aux pages d'ajout de médias (livres, DVD, CD, jeux de plateau).

**Correctifs Apportés :**

* **Configuration des URLs :** Inclusion des URL d'authentification de Django dans media\_project/urls.py et media/urls.py.
* **Ajout des Templates de Connexion et Déconnexion :** Création des fichiers login.html et logout.html dans media/templates/registration/.
* **Paramètres de Redirection :** Ajout des paramètres LOGIN\_REDIRECT\_URL et LOGOUT\_REDIRECT\_URL dans settings.py.
* **Création d'un Super Utilisateur :** Utilisation de la commande createsuperuser pour créer un compte administrateur permettant l'accès aux pages restreintes.

**2. Mise en Place des Fonctionnalités Demandées**

**Fonctionnalités Ajoutées :**

* **Ajout de Livres :** Page permettant l'ajout de livres à la médiathèque.
* **Ajout de DVD :** Page permettant l'ajout de DVD à la médiathèque.
* **Ajout de CD :** Page permettant l'ajout de CD à la médiathèque.
* **Ajout de Jeux de Plateau :** Page permettant l'ajout de jeux de plateau à la médiathèque.

**Étapes :**

* **Création des Vues :** Définition des vues correspondantes dans media/views.py.
* **Configuration des URLs :** Ajout des chemins dans media/urls.py.
* **Création des Templates :** Création des fichiers HTML pour les pages d'ajout dans media/templates/media/.

**3. Stratégie de Tests**

**Tests Manuels :**

* **Vérification de la Connexion :** Tester l'accès aux pages de connexion et de déconnexion.
* **Ajout de Médias :** Ajouter différents types de médias et vérifier leur enregistrement dans la base de données.
* **Accès Restreint :** Vérifier que les pages d'ajout de médias sont accessibles uniquement aux utilisateurs authentifiés.

**Tests Automatisés :**

* **Tests de Vue :** Utilisation de django.test.TestCase pour tester les vues et les redirections.
* **Tests de Modèle :** Vérification de l'intégrité des données dans la base de données.

**4. Base de Données avec des Données de Test**

**Mise en Place de la Base de Données :**

* **Migration des Modèles :** Utilisation des commandes makemigrations et migrate pour créer les tables nécessaires.
* **Population des Données :** Création de scripts de fixtures pour charger des données de test.

**Exemple de Fixture :** (fichier JSON pour la commande loaddata)

json

Copier le code

[

{

"model": "media.book",

"pk": 1,

"fields": {

"title": "Test Book",

"author": "Author Name",

"isbn": "1234567890",

"publication\_date": "2023-01-01"

}

}

]

**5. Instructions pour Exécuter le Programme**

**Prérequis :**

* **Python 3.x**
* **Django (version spécifiée dans requirements.txt)**

**Étapes :**

1. **Configurer l'Environnement Virtuel :**

bash

Copier le code

python -m venv venv

source venv/bin/activate # Sur Windows: venv\Scripts\activate

1. **Installer les Dépendances :**

bash

Copier le code

pip install -r requirements.txt

1. **Configurer la Base de Données :**

bash

Copier le code

python manage.py makemigrations

python manage.py migrate

1. **Créer un Super Utilisateur :**

bash

Copier le code

python manage.py createsuperuser

1. **Charger les Données de Test (Optionnel) :**

bash

Copier le code

python manage.py loaddata fixtures.json

1. **Démarrer le Serveur de Développement :**

bash

Copier le code

python manage.py runserver

1. **Accéder à l'Application :** Ouvrir un navigateur et naviguer vers http://127.0.0.1:8000.

**Note :**

* Pour exécuter ce programme sur n'importe quelle machine, assurez-vous d'avoir Python et les packages requis installés. Vous pouvez également utiliser Docker pour encapsuler l'application si nécessaire.