

L'étude et le correctif du code fournis :

Le code qui nous a été fournis est le suivant :

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

if __name__ == '__main__':
    menu()

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

class dvd():
    name = ""
    realisateur = ""
    dateEmprunt = ""
    disponible = ""
    emprunteur = ""

class cd():
    name = ""
    artiste = ""
    dateEmprunt = ""
    disponible = ""
    emprunteur = ""

class jeuDePlateau :
    name = ""
    createur = ""

class Emprunteur():
    name = ""
    bloque = ""

def menuBibliotheque() :
    print("c'est le menu de l'application des bibliothécaire")

def menuMembre():
    print("c'est le menu de l'application des membres")
    print("affiche tout")
```

Le script

```
if __name__ == '__main__':  
    menu()
```

Il est directement disponible dans le fichier manage.py fournis par django

```
if __name__ == '__main__':  
    main()
```

Les fonctions menu, menuBibliotheque et menuMembre sont directement dans le fichier views.

Les classes doivent commencer par des majuscules et les parenthèses sont utiles qu'en cas d'héritage d'une classe parent.

Dans certaines classes comme la classe dvd ou emprunteur il y a des erreurs d'indentation, en effet il y a des espaces en trop, il faut 4 espaces.

J'ai créé une classe Media générale et les classes DVD, CD, Livre hériteront de la classe Media pour éviter les répétitions.

Mise en place des fonctionnalités :

J'ai mis en place 2 applications, une consultation et une gestion_bibliotheque.

L'application consultation permet uniquement d'afficher la liste de tous les médias.

L'application gestion_bibliotheque donne accès aux fonctionnalités :

- Créer un membre-emprunteur.
- Afficher la liste des membres.
- Mettre à jour un membre.
- Afficher la liste des médias.
- Créer un emprunt pour un média disponible.
- Ajouter un media.
- Rentrer un emprunt.

Cette dernière gère aussi la fonctionnalité d'authentification afin de pouvoir y accéder grâce à un identifiant et un mot de passe.

Stratégie de tests :

Pour tester mon projet j'ai utilisé pytest. Un dossier 'test' regroupe donc tous les tests nécessaires pour vérifier que toutes les fonctionnalités fonctionnent bien.

La base de données contient des données de test, notamment un emprunt non rendu de plus de 7 jours.

Informations pour le projet :

Identifiants pour l'application gestion_bibliotheque :

Id : admin

Mdp : admin