



Mini-projet final

Introduction

Le projet consiste à réaliser une application de gestion de bibliothèque en Python, utilisant SQLAlchemy pour la gestion de la base de données PostgreSQL. L'objectif est de permettre la gestion des étudiants, des livres et des emprunts via un menu interactif, avec des fonctionnalités CRUD complètes.

Technologies et dépendances

- Python 3.13.9
 - PostgreSQL
 - SQLAlchemy (`pip install sqlalchemy`)
 - psycopg (`pip install psycopg`)
 - python-dotenv (`pip install dotenv`)
 - Faker (`pip install faker`)
1. Organisation du projet
 - `main.ts` : point d'entrée et menu principal de l'application
 - `database.py` : fichier de connexion à la base de données
 - `models/` : définitions des tables via SQLAlchemy
 - `repositories/` : fonctions CRUD par table
 - `seeder.py` : script de remplissage de la base pour tests
 - `create_tables.sql` : script SQL de création des tables

Tables principales :

- **Etudiant**
 - id_etud, nom, prenom, email, date_inscription, solde_amende
- **Livre**
 - isbn, titre, editeur, annee, exemplaires_dispo

- Emprunt
 - id_emprunt, id_etud, isbn, date_emprunt, date_retour, amende

Contraintes et règles

- Les clés primaires et étrangères assurent l'intégrité référentielle.
- Les contraintes CHECK assurent la validité des données (ex : année > 1900, solde_amende >= 0).
- Les emails étudiants sont uniques.

Fonctionnalités principales

- CRUD complet pour les étudiants, les livres et les emprunts.
- Menu interactif pour naviguer facilement entre les options.
- Validation des données et gestion des erreurs.

Sécurité et robustesse

- Validation minimale des entrées utilisateur.
- Gestion des exceptions SQL.
- Utilisation de variables d'environnement pour la connexion à la base.

Lancement du projet

Avant de lancer l'application, il est nécessaire de compléter le fichier `.env` avec les informations de connexion à la base de données

Python ▾

```
1 DB_HOST=
2 DB_PORT=
3 DB_NAME=
4 DB_USER=
5 DB_PASSWORD=
```

Ensuite, il est recommandé d'exécuter `seeder.py` en premier pour générer automatiquement les tables et peupler les données si nécessaire. Une fois cette étape réalisée, l'application peut être lancée via `main.py` pour accéder au menu interactif.

Conclusion

Le projet permet de gérer efficacement les étudiants, les livres et leurs emprunts, tout en respectant l'intégrité de la base et en offrant une interface console simple et fonctionnelle. Des améliorations futures pourraient inclure une interface graphique ou des notifications pour les retards d'emprunt.

Exemple de sortie

Page d'accueil

Bienvenue

Total Étudiants	Total Livres	Total Emprunts
61	101	104

À propos de cette application

Cette application gère :

- Étudiants : Ajouter, modifier, supprimer des étudiants
- Livres : Gérer le catalogue de la bibliothèque
- Emprunts : Enregistrer et gérer les emprunts/retours
- Amendes : Suivi automatique des amendes de retard

Page Étudiants

Gestion des Étudiants

Ajouter Afficher Modifier Supprimer

+ Ajouter un nouvel étudiant

Nom	Dupont
Prénom	Jean
Email	jean.dupont@example.com
Ajouter l'étudiant	

Page Livres

Gestion de Bibliothèque

Gestion des Livres

Ajouter Afficher Modifier Supprimer

+ Ajouter un nouveau livre

ISBN
978-2-07-036822-8

Titre
Le Seigneur des Anneaux

Éditeur
Pocket

Année de publication
2024

Exemplaires disponibles
1

Ajouter le livre

Page Emprunts

Gestion de Bibliothèque

Gestion des Emprunts

Enregistrer Retourner Tous les emprunts Par étudiant

+ Enregistrer un nouvel emprunt

Sélectionnez un étudiant
5 - Éditifefsef Boulay

Sélectionnez un livre
9784438209028 - Accrocher bras témoin terme lire

Enregistrer l'emprunt

Page Statistiques

Navigation

- Sélectionnez une section :
-  Accueil
 -  Etudiants
 -  Livres
 -  Emprunts
 -  Statistiques

Gestion de Bibliothèque

Statistiques

