
Projet 2 : Gestion d'une Bibliothèque - MyLib

- **Respectez strictement les instructions mentionnées dans cet énoncé.**
- À rendre au plus tard le Samedi 08 Février 2025 à 23h59, Aucun retard accepté.
- Ce projet vaut 40% de la moyenne du cours.
- **Fichier à remettre :**
 - Vous devez remettre sur **Teams** uniquement un fichier **Jupyter Notebook** par le chef de groupe seulement.
 - Nom du fichier : `projet<avec numéro de projet et numéro du groupe>.ipynb`.
 - **Ne soumettez pas** de fichier PDF, Word ou tout autre format.
- Dans le fichier `.ipynb`, on trouve le **Code** et le **Rapport** :
 - Remplissez une cellule **Markdown** au début pour introduire votre projet, incluant :
 - * Nom du projet.
 - * Description du projet.
 - * Votre nom et prénom.
 - * Votre groupe.
 - Avant chaque cellule de code, expliquez la partie de ce code là dans une cellule **Markdown**.
 - Commentez chaque section de votre code pour expliquer son rôle (dans les blocs de code).
- **Structure du fichier :**
 - Le fichier doit être organisé, avec des blocs de code et de texte clairement séparés.
 - Assurez-vous que votre code s'exécute correctement dans l'ordre des cellules.
 - Aucun Template n'est donné donc soyez organiser et créative.

Objectif du Projet :

Dans ce projet, Vous devez construire un système de gestion de bibliothèque appelé **MyLib**, entièrement écrit en Python. Le programme permettra de gérer les livres, les utilisateurs et les emprunts. Toutes les informations seront stockées dans des structures Python (dictionnaires, listes) pendant l'exécution, et sauvegardées dans un fichier CSV à la fin.

Fonctionnalités Demandées :

Partie 1 : Structure des Dictionnaires (12%)

Pour organiser les données, les informations sur les livres, les utilisateurs et les emprunts seront stockées dans des dictionnaires structurés comme suit:

1.) Dictionnaire des livres :

- Chaque livre est identifié par son titre.
- Les informations stockées pour chaque livre:
 - **Auteur** : Nom de l'auteur.
 - **Genre** : Genre littéraire (ex. : Roman, Programmation, Science-fiction, etc.).
 - **Exemplaires** : Nombre d'exemplaires disponibles dans la bibliothèque.
 - **Emprunts** : Nombre total d'emprunts effectués pour ce livre.
- Exemple :

```
books = {  
  "Python Programming": {  
    "Auteur": "John Doe",  
    "Genre": "Programmation",  
    "Exemplaires": 5,  
    "Emprunts": 10  
  },  
  "The Great Gatsby": {  
    "Auteur": "F. Scott Fitzgerald",  
    "Genre": "Roman",  
    "Exemplaires": 2,  
    "Emprunts": 7  
  }  
}
```

2.) Dictionnaire des utilisateurs :

- Chaque utilisateur est identifié par un ID unique.
- Les informations stockées pour chaque utilisateur:
 - **Nom** : Nom de l'utilisateur.
 - **Prénom** : Prénom de l'utilisateur.
 - **Email** : Adresse email de l'utilisateur.
 - **Téléphone** : Numéro de téléphone de l'utilisateur.
 - **Emprunts** : Nombre total de livres empruntés.
- Exemple :

```
users = {  
  1: {  
    "Nom": "Smith",  
    "Prénom": "Alice",  
    "Email": "alice@gmail.com",  
    "Téléphone": "514-888-9696",  
    "Emprunts": 5,  
  }  
}
```

```

        "ListeLivreLu":["Python Programming","The Great Gatsby","Marx's Inferno",
                        "Atomic Habits"]
    },
    2: {
        "Nom": "Brown",
        "Prénom": "Bob",
        "Email": "bob@gmail.com",
        "Téléphone": "430-568-8985",
        "Emprunts": 2,
        "ListeLivreLu":["Python Programming","The Great Gatsby"]
    }
}

```

3.) Liste des emprunts :

- Chaque emprunt est représenté comme un dictionnaire dans une liste.
- Les informations stockées pour chaque emprunt:
 - Utilisateur_ID : ID de l'utilisateur ayant emprunté le livre.
 - Livre : Titre du livre emprunté.
 - Date_Emprunt : Date à laquelle le livre a été emprunté.
 - Date_Retour : Date à laquelle le livre a été retourné (ou None si le livre est encore emprunté).
- Exemple :

```

    loans = [
        {
            "Utilisateur_ID": 1,
            "Livre": "Python Programming",
            "Date_Emprunt": "2024-12-01",
            "Date_Retour": "2024-12-10"
        },
        {
            "Utilisateur_ID": 2,
            "Livre": "The Great Gatsby",
            "Date_Emprunt": "2024-11-25",
            "Date_Retour": None
        }
    ]

```

4.) Liens entre les dictionnaires

- Livres et emprunts :
 - Les emprunts utilisent les titres des livres pour les associer à leur entrée dans le dictionnaire des livres.
- Utilisateurs et emprunts :
 - Les emprunts utilisent l'Utilisateur_ID pour les associer à leur entrée dans le dictionnaire des utilisateurs.

- À chaque fois qu'un utilisateur emprunte un livre, le nom du livre est ajouté à la liste `ListeLivreLu`. Si l'utilisateur emprunte le même livre plusieurs fois, celui-ci n'est pas ajouté une seconde fois à cette liste (pas de répétition).
- Mise à jour dynamique :
 - Lors d'un emprunt, le nombre d'exemplaires disponibles dans `books` est réduit et un nouvel emprunt est ajouté dans `loans`.
 - Lors d'un retour, la date de retour est mise à jour dans `loans` et le stock dans `books` est augmenté.

Partie 2 : Requêtes et Analyses 55%

Vous devez implémenter les 9 requêtes avancées suivantes :

1.) (5%) Ajouter ou supprimer un livre dans la bibliothèque.

- Ajouter ou supprimer un livre dans la bibliothèque.
- Enregistrer des informations comme le titre, l'auteur, le genre, le nombre d'exemplaires disponibles, et le nombre total d'emprunts.

2.) (5%) Ajouter ou supprimer un utilisateur.

- Ajouter ou supprimer un utilisateur.
- Enregistrer les informations comme le nom, le prénom, l'email (format obligatoire : `email@gmail.com`), le numéro de téléphone (format obligatoire : `xxx-xxx-xxxx`) et le nombre total d'emprunts effectués.

3.) (5%) Enregistrer un emprunt ou un retour.

- Enregistrer un emprunt (vérifier la disponibilité du livre avant validation).
- Enregistrer un retour (mettre à jour la durée d'emprunt et appliquer une pénalité en cas de retard).

4.) (5%) Lister les livres les plus empruntés.

- Afficher les 5 livres ayant le plus grand nombre d'emprunts.

5.) (5%) Calculer la durée moyenne des emprunts par genre.

- Analyser les emprunts pour chaque genre et calculer la durée moyenne.

6.) (5%) Identifier les utilisateurs les plus actifs.

- Trier les utilisateurs par nombre total d'emprunts effectués et afficher les 3 plus actifs.

7.) (5%) Afficher le statut de la bibliothèque sous forme de statistiques.

- Nombre total de livres.
- Nombre total d'exemplaires empruntés.
- Pourcentage d'exemplaires actuellement disponibles.
- Nombre moyen de livres empruntés par utilisateur.

8.) (10%) (**Documentez vous sur la librairie matplotlib**) Visualisation: Diagramme circulaire des emprunts par genre.

- Afficher un graphique en cercle montrant la répartition des emprunts par genre.

9.) (10%) (**Documentez vous sur la librairie matplotlib**) Visualisation: Évolution mensuelle des emprunts.

- Tracer un graphique montrant le nombre d'emprunts effectués chaque mois pour l'année en cours.

Partie 3 : Visualisation (08%)

- Créer un diagramme en barres qui montre les 10 utilisateurs les plus actifs en termes de nombre total de livres empruntés.
- Trier les utilisateurs par leur nombre d'emprunts (Emprunts) en ordre décroissant.
- Extraire les 10 premiers utilisateurs avec leurs noms et le nombre d'emprunts.
- Tracer un graphique en barres avec:
 - Axe X : Les noms des utilisateurs (format "Prénom Nom").
 - Axe Y : Le nombre de livres empruntés.

Partie 4 : Menu Interactif (15%)

Menu Interactif (5%)

Créer un menu interactif qui permet à l'utilisateur de choisir les requêtes qu'il souhaite exécuter.
Exemple de menu:

--- Menu Principal ---

1. Ajouter ou supprimer un livre dans la bibliothèque.
2. Ajouter ou supprimer un utilisateur.
3. Enregistrer un emprunt ou un retour.
4. Lister les livres les plus empruntés.
5. Calculer la durée moyenne des emprunts par genre.
6. Identifier les utilisateurs les plus actifs.
7. Afficher le statut de la bibliothèque sous forme de statistiques.
8. Visualisation: Diagramme circulaire des emprunts par genre.
9. Visualisation: Évolution mensuelle des emprunts.
10. Quitter.

Sauvegarde des Données (10%)

À la fin de l'exécution, toutes les données doivent être sauvegardées dans des fichiers CSV:

- (3.33%) livres.csv : Colonnes : Titre, Auteur, Genre, Exemplaires, Emprunts.
- (3.33%) utilisateurs.csv : Colonnes : ID, Nom, Prénom, Email, Téléphone, Emprunts.
- (3.33%) emprunts.csv : Colonnes : Utilisateur_ID, Livre, Date_Emprunt, Date_Retour.

Partie 4.beta : Interface Graphique - Bonus (30%)

Vous pouvez développer une interface graphique pour votre application (**Ce n'est pas obligatoire**) .
Par exemple :

- Utiliser les librairies `tkinter`, `PyQt`, ou `Kivy`.
- Fournir des boutons pour chaque requête.
- Afficher les graphiques et résultats dans des fenêtres interactives.

Partie 5 : Présentation du projet (10%)

À la fin du projet, chaque groupe d'étudiants devra réaliser une présentation orale obligatoire devant l'ensemble de la classe. Cette présentation représente une étape clé pour :

- Évaluer votre compréhension du projet.
- Valoriser votre travail collectif.
- Juger votre capacité à communiquer et expliquer des concepts techniques.

Durée

- 15 à 20 minutes pour présenter le projet.
- 5 minutes pour une session de questions-réponses.

Contenu de la Présentation :

1.) Introduction (2-3 minutes) :

- Présenter brièvement le projet et son objectif.
- Expliquer pourquoi ce sujet est pertinent et intéressant.

2.) Fonctionnalités principales (5-7 minutes) :

- Décrire les principales fonctionnalités développées.
- Montrer comment l'utilisateur interagit avec le programme (menu, options, etc.).

3.) Aspect technique (5-7 minutes) :

- Expliquer les bibliothèques utilisées.
- Décrire la structure du code.
- Présenter les défis techniques rencontrés et comment ils ont été résolus.

4.) Démonstration (5 minutes) :

- Exécuter le programme en direct pour montrer son fonctionnement.
- Exemple d'utilisation : effectuer une ou deux requêtes interactives.
- Afficher au moins une visualisation graphique (diagramme en cercle ou graphique en barres).

5.) Session Questions-Réponses (5 minutes) :

- Répondre aux questions du prof ou des autres étudiants.

- Montrer votre capacité à défendre vos choix et expliquer votre raisonnement.

Cette présentation permet non seulement de valider le travail réalisé, mais aussi de renforcer les compétences en communication technique, en gestion de projet, et en esprit critique. Juste amusez vous a la faire - avec un peu de stress c'est cool. ;)

Rapport Final

Le rapport final doit être rédigé dans un fichier Jupyter Notebook (.ipynb), structuré en sections claires et lisibles. Ce format permet de mélanger du texte explicatif (en **Markdown**) et du code Python, tout en incluant les résultats directement dans le document. Cela facilite la démonstration des concepts, du fonctionnement du code, et des visualisations.

Documentation et librairie d'aide

Bibliothèque	Usage principal	Documentation
csv	Lire et écrire des fichiers CSV pour enregistrer les données des livres, utilisateurs et emprunts.	Lien
json	Manipuler et exporter les données en JSON.	Lien
datetime	Gérer les dates et effectuer des calculs temporels.	Lien
matplotlib	Créer des visualisations, comme des graphiques en cercle et en barres.	Lien
numpy	Effectuer des calculs mathématiques complexes, comme la moyenne ou la médiane (optionnel).	Lien
tkinter	Créer une interface graphique avec des boutons et des champs interactifs (optionnel).	Lien

Autre aide et recommandations Générales :

- ChatGPT : Utilisez-le intelligemment.
- W3School : Pour toutes les fonctionnalités de Python (Dictionnaire / fonctions / ...etc).

Outils collaboration et travail en groupe :

- Gestion de temps et de tâche: Trello
- Coder en temps réel en groupe dans un seul .ipynb : Google Colab

Message de soutien

- Si vous ressentez une baisse de motivation, rencontrez des difficultés ou faites face à des imprévus, sachez que c'est tout à fait normal dans le cadre d'un projet qui représente 40% de la moyenne du cours. Travaillez en équipe, communiquez avec vos camarades, organisez votre temps efficacement et répartissez les tâches de manière équilibrée selon vos compétences et disponibilités.
- En cas de conflit ou de situation délicate, je reste disponible et à votre écoute. N'hésitez pas à me contacter si vous avez besoin de conseils ou d'assistance.

Bon courage à tous !