

## **E23 – POO III - Travail Pratique 4 (20%)**

### **I - Objectifs de ce TP**

Ce travail pratique (TP) vise à évaluer votre compréhension des notions vues en cours à savoir :

- Création du modèle de classes;
- Création des contrôleurs;
- Création des vues ASP.NET Core;
- Validation de cohérence des données;
- Accès à une base de données avec Entity Framework Core;
- Persistance des données;
- Utilisation de LINQ;
- Respect des principes SOLID;
- Utilisation de l'architecture propre et du pattern repository;
- Création des APIs Web.

### **II - Contexte**

Ce travail doit être effectué par le même groupe que celui du TP 3. Vous devez utiliser le même dépôt que celui du TP3 ou créer un nouveau dépôt privé sur GitHub et m'envoyer une invitation en tant que collaborateur.

Mon compte GitHub est : hinault.

Vous devez déposer à partir de Lea un fichier contenant les informations suivantes :

- Le rapport PDF
- Un zip du code de votre application

### **III - Date de remise**

Votre travail doit être remis au plus tard le dimanche 24 septembre 2023 à 23h59.

### **IV - Critères d'évaluation**

Votre travail doit respecter l'ensemble des critères suivants :

- Le code doit compiler
- Le code ne doit pas comporter d'erreurs d'exécution
- Le code doit réaliser toutes les fonctionnalités demandées
- Le code doit respecter les bonnes pratiques et l'architecture propre
- L'application doit être en français

- Qualité du rapport
- -10% par jour de retard. 0 en cas de remise du correctif par l'enseignant.

## **V - Miniprojet bibliothèque**

Vous avez été engagé par la bibliothèque LIPAJOLI qui souhaite digitaliser ses activités. L'entreprise aimerait mettre en place un site Web pour la gestion et la location des livres.

Votre enseignant fait office de représentant de l'entreprise. Vous pouvez lui poser toute question en rapport avec les besoins du client.

L'application doit permettre de fournir les fonctionnalités suivantes :

- Gestion des livres :
  - Affichage de la liste complète des livres
  - Recherche d'un livre
  - Ajout/suppression/modification/consultation d'un livre
- Gestion des usagers :
  - Affichage de la liste complète des usagers
  - Recherche d'un usager
  - Ajout/suppression/modification/consultation d'un usager
- Gestion des emprunts :
  - Faire un emprunt
  - Rendre un livre
  - Consulter les emprunts en cours
  - Consulter l'historique des emprunts
  - Consulter la fiche d'un emprunt : elle doit contenir le nom de l'emprunteur et le nom du livre

### **Spécifications sur la gestion des emprunts**

La gestion des emprunts doit permettre de savoir à quelle date l'exemplaire d'un livre a été emprunté par un usager.

Un historique doit être conservé et contenir également la date de retour de l'exemplaire.

Un usager peut emprunter trois livres au maximum et un seul exemplaire d'un livre.

Un emprunt décrémente la quantité d'exemplaires du livre et un retour incrémente cette quantité.

Lors du retour du livre, une défaillance est automatiquement portée au dossier de l'utilisateur si la date de retour est supérieure à la date limite de retour (cette date est obtenue en additionnant la date d'emprunt à un nombre de jours obtenu à partir des fichiers de configuration de l'application MVC). Ce nombre doit être défini à 10 jours dans le fichier AppSettings.json.

A trois défaillances, l'utilisateur ne doit plus être capable d'emprunter.

## VI - Travail à faire

- Dans votre solution du TP 3, ajoutez les projets Bibliotheques.API, Bibliotheques.ApplicationCore et Bibliotheques.Infrastructure
- Implémentez l'API en tenant compte des spécifications sur la gestion des emprunts et en respectant le patron repository et l'architecture propre. Vous devez également documenter de façon exhaustive l'API en utilisant Swagger. L'API doit exposer les méthodes suivantes :
  - Inscription d'un nouvel emprunt
  - Obtention de tous les emprunts : l'emprunt doit inclure l'utilisateur et le livre
  - Obtention d'un emprunt : l'emprunt doit inclure l'utilisateur et le livre
  - Retour d'un emprunt
  - Suppression d'un emprunt : un emprunt ne peut être supprimé si le livre a été retourné
  - Obtention de tous les utilisateurs
  - Obtention des livres
- Implémentez la gestion des emprunts dans l'application MVC. Vous devez utiliser l'API pour toutes les opérations.

### Notes :

Vous ne devez apporter aucune modification à la gestion des utilisateurs et des livres dans l'application MVC. Mais, pour que l'API et l'application MVC utilisent la même base de données, vous devez déplacer votre base de données dans le répertoire racine de votre solution. La chaîne de connexion à ajouter dans l'application MVC et l'API Web devrait ressembler à ce qui suit :

```
"ConnectionStrings": {  
  "DefaultConnection": "Data Source=..\lipajoli.db"  
}
```

## VII – Rapport PDF

Vous devez rédiger un rapport PDF à remettre contenant :

- Un miniguide d'utilisation de l'application. Documentez uniquement la gestion des emprunts
- La liste des difficultés que vous avez rencontrées