

Sujet : Mini-application de gestion de bibliothèque en PHP (MVC + POO)

Objectif

Créer une petite application web permettant de gérer une bibliothèque :

- Gestion des **livres** (ajout, suppression, édition, affichage)
- Gestion des **utilisateurs** (simple CRUD)
- Possibilité pour un utilisateur d'**emprunter un livre**

Le tout en **PHP POO** avec une structure **MVC maison** (pas de framework type Laravel ou Symfony).

Contraintes techniques

1. Organisation MVC :

- /public/index.php → point d'entrée unique (Front Controller).
- /app/Controllers/ → les contrôleurs.
- /app/Models/ → les modèles (classes qui représentent les entités et communiquent avec la BDD).
- /app/Views/ → les vues (HTML + un peu de PHP).
- /core/ → classes de base (Router, Database, Controller, Model...).

2. Base de données MySQL (minimum 2 tables) :

- books (id, titre, auteur, disponible [bool])
- users (id, nom, email)
- borrows (id, user_id, book_id, date_emprunt, date_retour [nullable])

3. Classes à créer :

- Core\Database : singleton pour gérer PDO.
- Core\Controller : classe parent des contrôleurs.
- Core\Model : classe parent des modèles.
- App\Models\Book, App\Models\User, App\Models\Borrow.
- App\Controllers\BookController, App\Controllers\UserController, App\Controllers\BorrowController.

- Core\Router : dispatcher pour charger le bon contrôleur/action selon l'URL.

4. Fonctionnalités attendues :

- Lister les livres disponibles.
- Ajouter/éditer/supprimer un livre.
- Lister les utilisateurs.
- Enregistrer un nouvel utilisateur.
- Emprunter un livre (choix user + book).
- Retourner un livre.

5. Bonus (si temps) :

- Ajouter une validation simple (ex : email valide).
- Pagination des listes.
- Gérer un design simple avec Bootstrap.

Étapes suggérées (≈5h de travail)

1. Mise en place de la structure MVC (1h30)

- Créer l'arborescence des dossiers.
- Implémenter Router, Controller, Database.
- Configurer index.php pour router les requêtes.

2. Création des modèles et de la BDD (45 min)

- Créer la base library.
- Écrire les migrations SQL (tables books, users, borrows).
- Implémenter Book, User, Borrow.

3. Création des contrôleurs (1h)

- BookController : CRUD complet.
- UserController : CRUD basique.
- BorrowController : emprunt/retour.

4. Création des vues (1h)

- views/books/index.php : liste des livres.
- views/books/add.php : formulaire d'ajout.
- views/users/index.php etc.

5. Test et finalisation (45 min)

- Tester chaque fonctionnalité.
- Corriger les bugs.
- Ajouter un peu de style.