Projet PHP POO – Gestion de Bibliothèque

# 1. Présentation du projet

Ce projet est une application PHP orientée objet permettant de gérer une petite bibliothèque. Il inclut la gestion des livres et des catégories à travers un CRUD complet, en utilisant PDO pour la connexion à la base de données, et une architecture MVC légère pour structurer le code proprement.

# 2. Objectifs pédagogiques

Ce projet permettra de :

- Comprendre les principes de la POO (classes, objets, encapsulation, héritage, polymorphisme).

- Créer un CRUD connecté à une base de données MySQL avec PDO.

- Structurer un projet en utilisant une architecture MVC simplifiée.

- Implémenter un système d’autoloading des classes.

# 3. Structure du projet

L'arborescence du projet est la suivante :

bibliotheque/  
│  
├── config/  
│ └── database.php → Connexion PDO  
│  
├── core/  
│ ├── Controller.php → Contrôleur de base  
│ └── Model.php → Modèle de base (gestion PDO)  
│  
├── models/  
│ ├── Livre.php → Classe Livre (hérite de Model)  
│ └── Categorie.php → Classe Categorie (hérite de Model)  
│  
├── controllers/  
│ ├── LivreController.php → Gère les actions sur les livres  
│ └── CategorieController.php  
│  
├── views/  
│ └── livre/  
│ ├── index.php → Liste des livres  
│ ├── create.php → Formulaire ajout  
│ ├── edit.php → Formulaire modif  
│  
├── public/  
│ └── index.php → Point d’entrée (front controller)  
│  
├── assets/  
│ └── css/ etc.  
│  
└── autoloader.php → Chargement automatique des classes

# 4. Structure de la base de données

Voici le script SQL de base pour créer les tables nécessaires :

CREATE TABLE categories (  
 id INT AUTO\_INCREMENT PRIMARY KEY,  
 nom VARCHAR(100) NOT NULL  
);  
  
CREATE TABLE livres (  
 id INT AUTO\_INCREMENT PRIMARY KEY,  
 titre VARCHAR(255) NOT NULL,  
 auteur VARCHAR(255),  
 id\_categorie INT,  
 FOREIGN KEY (id\_categorie) REFERENCES categories(id)  
);

# 5. Description des fichiers clés

- config/database.php : gère la connexion à MySQL via PDO.

- core/Model.php : modèle de base utilisé par Livre et Categorie.

- core/Controller.php : contrôleur parent.

- autoloader.php : gère le chargement automatique des classes.

- public/index.php : point d’entrée du site, traite les routes.

# 6. Concepts POO couverts

- Classe et objet : Livre, Categorie, etc.

- Encapsulation : propriétés privées avec getters/setters.

- Héritage : Livre hérite de Model.

- Polymorphisme : méthodes redéfinies dans les classes enfants.

# 7. Étapes de réalisation du projet

1. Créer la base de données avec le script SQL.

2. Configurer le fichier database.php pour la connexion PDO.

3. Créer les classes modèles (Livre, Categorie).

4. Créer les contrôleurs pour gérer les actions.

5. Créer les vues HTML correspondantes.

6. Implémenter l’autoloader et tester chaque fonctionnalité.

# 8. Bonus possible

- Ajouter une gestion des emprunts (livres empruntés, date retour, etc.).

- Ajouter une recherche ou un filtre par catégorie.

- Ajouter une pagination pour la liste des livres.