

## Documentation Projet Bibliothèque PHP :

### 1) Présentation rapide

- Nom du projet : Book Search App
- Objectif : Développer une application web permettant de rechercher des livres via une API, de gérer des comptes utilisateurs et de sauvegarder des favoris de manière persistante.
- Contexte : Projet full-stack (Front-end, Back-end, BDD).

### 2) Fonctionnalités

- Recherche Hybride : Système de recherche par titre et genre utilisant l'API Google Books (via PHP cURL et JavaScript Fetch).
- Gestion Utilisateur : Système complet d'inscription et de connexion (Sign-up/Login) avec sessions sécurisées.
- Espace Personnel : Profil utilisateur modifiable (avatar, email, nom) et gestion d'une liste de favoris personnalisée.
- Persistance des données : Ajout et suppression de livres en favoris stockés directement en base de données MySQL.
- Sécurité : Utilisation de requêtes préparées pour prévenir les injections SQL.

### 3) Partie technique simplifiée

- Langages : PHP (Logique serveur), JavaScript (Interactivité asynchrone), SQL (Gestion BDD), CSS (Grille de résultats).
- Architecture : Approche orientée objet avec l'utilisation de classes PHP (User, Favorite..) pour une meilleure structure du code.
- API & cURL : Consommation de l'API Google Books via PHP (cURL) pour les recherches initiales et JavaScript (Fetch) pour la mise à jour dynamique de l'interface.
- Communication AJAX : Envoi de données JSON entre le front-end et le back-office (profil.php) pour ajouter/supprimer des favoris sans recharger la page.
- Base de données : Utilisation de l'extension PDO pour une connexion robuste et sécurisée à MySQL.

#### 4) Difficultés / Ce que j'ai appris

Ce projet a été le plus exigeant techniquement, car il combine plusieurs flux de données simultanés :

- Synchronisation Front et Back : J'ai dû apprendre à faire communiquer JavaScript et PHP de manière asynchrone via des requêtes fetch. Le défi était de s'assurer que lorsqu'un utilisateur clique sur "Favori", la base de données soit mise à jour instantanément et que l'interface reflète ce changement sans latence.
- Gestion des sessions et sécurité : J'ai appris à sécuriser les accès aux pages via `session_start()` et à protéger les données sensibles en utilisant systématiquement des requêtes préparées PDO. Cela m'a sensibilisé aux failles de sécurité courantes (Injections SQL).
- Manipulation d'API complexes : Travailler avec Google Books m'a appris à traiter des objets JSON complexes et à gérer les cas d'erreurs (images manquantes, auteurs inconnus) pour éviter que l'affichage ne "casse".
- Architecture Orientée Objet : En utilisant des fichiers séparés (`config.php`, `user.php`, `favori.php..`), j'ai appris à organiser mon code de façon modulaire, ce qui rend l'application beaucoup plus facile à maintenir et à faire évoluer.