



REPUBLIQUE DU SENEGAL

Un Peuple – Un But – Une Foi

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE

UNIVERSITE GASTON BERGER DE SAINT-LOUIS

U.F.R DE SCIENCES APPLIQUEES ET DE TECHNOLOGIE

MASTER 2 GDIL

**Conception et Implémentation d'un Système de
Gestion de Bibliothèque Distribuée : Cas des
Universités UGB et UAD**

Presenté par :

MAME ANTA BATHILY

DAOUDA FICKOU

MOUHAMADOU GUEYE

Année académique 2024-2025

Table des matières

I.	Introduction	3
II.	Contexte du Projet	3
III.	Problématique	3
IV.	Schéma global	4
V.	Schémas de placement.....	5
VI.	Le coordonnateur (micro-service).....	8
VII.	Présentation de l'Application	9
VIII.	Conclusion et Perspectives	15

I. Introduction

Ce rapport présente le travail réalisé dans le cadre du module Base de données distribuées. L'objectif est de concevoir et de développer une solution logicielle permettant de gérer le partage de ressources documentaires entre deux grandes institutions du Sénégal : l'Université Gaston Berger (UGB) et l'Université Alioune Diop (UAD). À travers ce projet, nous mettons en pratique les concepts de fragmentation, de placement et de gestion de la cohérence des données au sein d'un environnement réparti. L'application finale (**BiblioUniv**) permet non seulement de gérer les tâches administratives classiques (CRUD), mais aussi d'assurer une expérience fluide pour les bibliothécaires des deux sites.

II. Contexte du Projet

Le projet s'inscrit dans une démarche de mutualisation des savoirs. Historiquement, chaque université gérait sa bibliothèque de manière isolée, ce qui limitait l'accès aux ouvrages pour les étudiants en déplacement ou inscrits sur un autre site. Pour remédier à cela, l'UGB et l'UAD ont décidé d'unifier leurs services de prêts. Les règles du projet sont strictes : chaque université reste responsable de ses propres étudiants et de ses employés, mais le catalogue de livres et le suivi des prêts deviennent globaux. Technique, cela signifie que les données ne sont pas stockées à un seul endroit, mais réparties sur les deux sites, tout en restant accessibles comme s'il s'agissait d'une seule et même base de données.

III. Problématique

La question centrale de ce projet est la suivante : Comment assurer une gestion unifiée et sans erreur alors que les données et les utilisateurs sont physiquement situés sur des sites différents ?

Plusieurs défis majeurs découlent de cette situation :

La transparence d'accès : Un étudiant de l'UGB doit pouvoir consulter le stock de l'UAD et effectuer un emprunt sans ressentir la complexité technique liée à la distance entre les serveurs.

La cohérence des compteurs globaux : C'est le point le plus critique. Étant donné qu'un étudiant est limité à un certain nombre d'emprunts sur l'ensemble du réseau, comment garantir que son compteur est mis à jour instantanément, qu'il emprunte à Saint-Louis ou à Bambeay ?

La localisation des données (Placement) : Où doit-on stocker chaque information pour optimiser les performances ? Il faut décider si les données d'un étudiant doivent le suivre ou rester sur son site d'origine, tout en permettant aux requêtes globales de s'exécuter rapidement.

L'intégrité des règles métier : Par exemple, forcer le retour d'un ouvrage sur son site de prêt initial tout en mettant à jour le dossier de l'étudiant géré par son université d'origine.

Répondre à ces problématiques nous a obligés à réfléchir à la conception du schéma de placement et à la manière dont les sites communiquent pour maintenir la base de données toujours cohérente.

IV. Schéma global

Le schéma global représente l'ensemble des données du système de manière cohérente, indépendamment de leur localisation physique. Il constitue une vue logique unifiée de la base de données avant toute fragmentation ou distribution.

Il comprend cinq relations principales, décrites ci-dessous :

Relation	Attributs principaux	Description
EMPLOYÉ	IdEm, nom, adr, statut, bibliothèque	Identifie le personnel et leur bibliothèque d'affectation
ETUDIANT	Id_Etud, nom, adr, université_Origine, spec, nbre_Emprunts	Contient les informations personnelles et le nombre d'emprunts
OUVRAGE	IdOuv, titre, #IdAut, éditeur, an, domaine, stock, site	Détaille les ouvrages et leur site physique

AUTEUR	IdAut, nom_auteur	Référence des auteurs des ouvrages
PRÊT	#IdOuv, #IdEtud, dateEmp, dateRetour	Suivi des emprunts avec dates de sortie et retour

Les **clés étrangères** assurent la cohérence entre les relations :

- *IdAut* dans **OUVRAGE** référence **AUTEUR(IdAut)** ;
- *IdOuv* dans **PRET** référence **OUVRAGE(IdOuv)** ;
- *IdEtud* dans **PRET** référence **ETUDIANT(IdEtud)**.

Le champ **nbreEmprunts** de la relation **ETUDIANT** permet de contrôler le nombre maximal d'ouvrages qu'un étudiant peut emprunter simultanément sur l'ensemble des bibliothèques.

V. Schémas de placement

5.1 Stratégie de Fragmentation

Le système utilise trois stratégies de fragmentation complémentaires :

A. Fragmentation Horizontale (Primaire)

- Tables concernées : ETUDIANT, EMPLOYE, OUVRAGE
- Critère de fragmentation : Attribut de localisation (université/Bibliothèque/site)
- Justification :
 - Autonomie locale : Chaque site gère ses propres ressources
 - Réduction du trafic réseau : Les requêtes locales n'accèdent qu'à la base locale
 - Performance : Pas de jointures distribuées pour les opérations courantes

Prédicats de fragmentation :

- ETUDIANT :
 - $F_1(ETUDIANT) : \sigma(universite = 'UGB')(ETUDIANT) \rightarrow ETUDIANT_ugb$
 - $F_2(ETUDIANT) : \sigma(universite = 'UAD')(ETUDIANT) \rightarrow ETUDIANT_uad$
- EMPLOYE :

- $F_1(\text{EMPLOYE}) : \sigma(\text{Bibliotheque} = \text{'UGB'})(\text{EMPLOYE}) \rightarrow \text{EMPLOYE_ugb}$
- $F_2(\text{EMPLOYE}) : \sigma(\text{Bibliotheque} = \text{'UAD'})(\text{EMPLOYE}) \rightarrow \text{EMPLOYE_uad}$
- OUVRAGE :
- $F_1(\text{OUVRAGE}) : \sigma(\text{site} = \text{'UGB'})(\text{OUVRAGE}) \rightarrow \text{OUVRAGE_ugb}$
- $F_2(\text{OUVRAGE}) : \sigma(\text{site} = \text{'UAD'})(\text{OUVRAGE}) \rightarrow \text{OUVRAGE_uad}$

B. Fragmentation Dérivée

- Table concernée : PRET
- Critère de dérivation : Basée sur la fragmentation de ETUDIANT (clé étrangère IdEtud)
- Règle : Un prêt est stocké dans la base de données de l'université de l'étudiant emprunteur.
- Prédicats de fragmentation :
 - $F_1(\text{PRET}) : \text{PRET} \bowtie \text{ETUDIANT_ugb} \rightarrow \text{PRET_ugb}$ (sur site UGB)
 - $F_2(\text{PRET}) : \text{PRET} \bowtie \text{ETUDIANT_uad} \rightarrow \text{PRET_uad}$ (sur site UAD)

Justification :

- Cohérence avec la règle métier : "Les emprunts sont gérés par l'université d'origine de l'étudiant"
- Optimisation des jointures : PRET et ETUDIANT sont co-localisés
- Traçabilité : Chaque site conserve l'historique complet de ses étudiants

Attribut site_ouvrage : Permet de tracer l'origine du livre emprunté: Essentiel pour les transactions de retour (mise à jour du stock sur le bon site)

C. RéPLICATION Totale

- Table concernée : AUTEUR
- Stratégie : Copie complète sur les deux sites (UGB et UAD)

Justification :

- Table de référence de petite taille
- Lecture fréquente, modification rare
- Évite les requêtes distribuées lors de l'affichage des ouvrages
- Améliore la disponibilité (pas de point de défaillance unique)

Fragment	Site UGB	Site UAD	Type de fragmentation
ETUDIANT_ugb	✓		Horizontale
ETUDIANT_uad		✓	Horizontale
EMPLOYEE_ugb	✓		Horizontale
EMPLOYEE_uad		✓	Horizontale
OUVRAGE_ugb	✓		Horizontale
OUVRAGE_uad		✓	Horizontale
PRET (UGB)	✓		Dérivée
PRET (UAD)		✓	Dérivée
AUTEUR	✓	✓	RéPLICATION totale

VI. Le coordonnateur (micro-service)

1. Définition et Rôle

Le **Coordonateur** est un micro-service qui agit comme une façade unifiée pour l'application cliente. Il masque la complexité de la distribution des données et assure la cohérence du système.

Ses trois responsabilités majeures sont :

- **Transparence de Localisation** : Le client ne sait pas où sont stockées les données.
- **Agrégation de Données** : Il interroge plusieurs sources pour fournir une réponse unique.
- **Gestion de Transactions Distribuées** : Il garantit l'atomicité des opérations impliquant plusieurs sites (**Protocole 2PC** simulé).

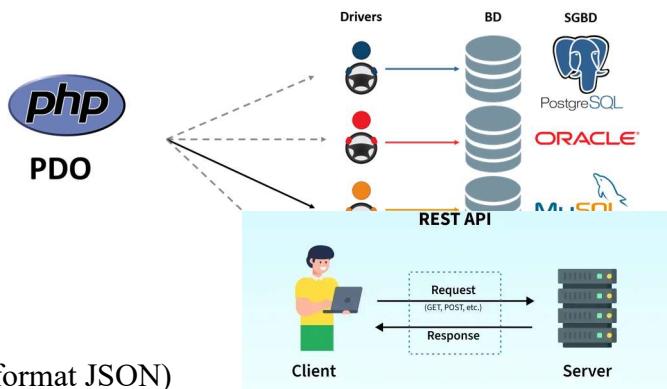
2. Implémentation Technique

Stack Technologique :

Langage : PHP (Vanilla)

Interface SGBD : PDO (PHP Data Objects)

Protocole : HTTP/REST (Communication format JSON)



Pourquoi ce choix pour le Coordonateur ? :

L'utilisation de PDO est critique ici. Contrairement à des drivers spécifiques (mysqli ou pgsql), PDO permet d'instancier plusieurs connexions de types différents (pgsql: et mysql:) au sein de l'objet PHP. Cela permet au Coordonateur de "parler" aux deux bases simultanément dans le même **thread** d'exécution.

3. Fonctionnement Détaillé

A. Routage et Identification

Lorsqu'une requête arrive (ex: GET /emprunts_etudiant), le Coordonateur doit savoir quelle base interroger ou bien interroger toutes les bases à la fois .

Processus : Il accepte un paramètre site ou tente une lecture séquentielle (UGB puis UAD).

Exemple : Pour l'authentification, il scanne la table EMPLOYE de l'UGB. Si échoue, il scanne celle de l'UAD.

B. Gestion des Transactions Distribuées (2PC)

C'est la fonction la plus complexe. Prenons l'exemple d'un Retour d'Ouvrage où l'étudiant est à l'UGB mais le livre appartient à l'UAD.

Phase de Préparation (Begin Transaction) :

Le Coordinateur ouvre une transaction SQL sur UGB (`$ugb->beginTransaction()`). Il ouvre une transaction SQL sur UAD (`$uad->beginTransaction()`).

Phase d'Exécution :

- Il met à jour la table **PRET** sur UGB (Date de retour).
- Il met à jour le compteur de l'étudiant sur UGB.
- Il met à jour le stock du livre sur UAD.

Phase de Validation (Commit/Rollback) :

- Si toutes les requêtes SQL réussissent sans erreur :
 - Il valide UGB (`$ugb->commit()`).
 - Il valide UAD (`$uad->commit()`).
- Si une seule erreur survient (ex: contrainte violée, panne réseau) :
 - Il annule tout sur UGB (`$ugb->rollBack()`).
 - Il annule tout sur UAD (`$uad->rollBack()`).

Cela garantit que le système ne se retrouve jamais dans un état **incohérent** (ex: Livre rendu mais stock **non incrémenté**).

Le Coordinateur est l'intelligence du système. Il transforme deux(ou plusieurs) bases de données isolées et hétérogènes en un système d'information cohérent et unifié pour l'utilisateur final.

VII. Présentation de l'Application

BIBLIOUNIV est une application web de gestion de bibliothèque distribuée permettant la gestion unifiée des ressources documentaires de deux sites universitaires : l'Université Gaston Berger (UGB) de Saint-Louis et l'Université Alioune Diop (UAD) de Bambe.

L'application offre une interface unique pour gérer les étudiants, les ouvrages, les employés et les opérations de prêt/retour, tout en respectant l'autonomie de chaque site et les contraintes de la distribution des données.

Architecture Technique

Stack Technologique

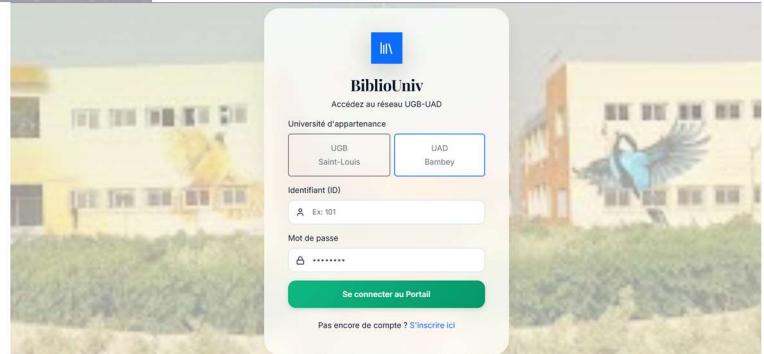
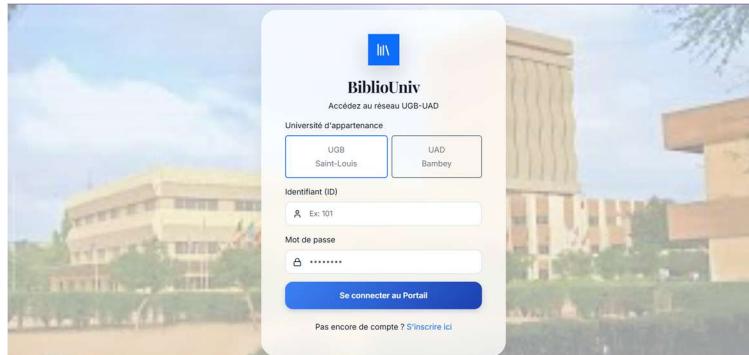
Frontend : HTML5, CSS3 (Bootstrap 5), JavaScript (Vanilla JS), Lucide Icons pour l'iconographie, Architecture SPA-like avec pages HTML séparées

Backend : PHP 7+ (API REST), Middleware d'agrégation (index.php), PostgreSQL (UGB) et MySQL (UAD)

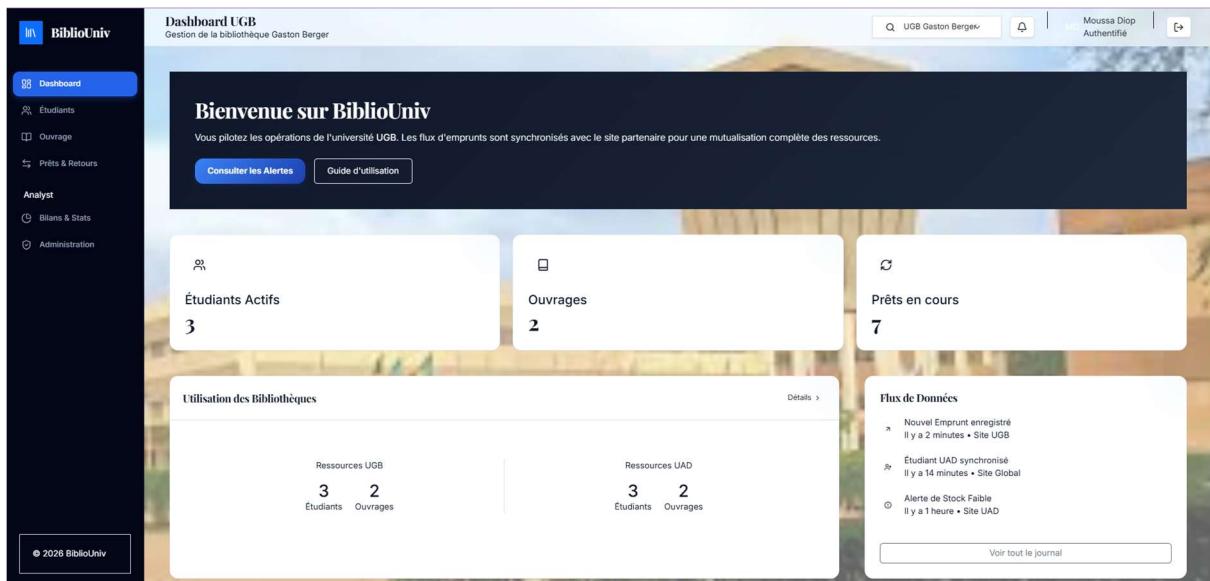
Communication : API REST JSON, Service centralisé [data-service.js](#) pour toutes les requêtes

Composants Principaux

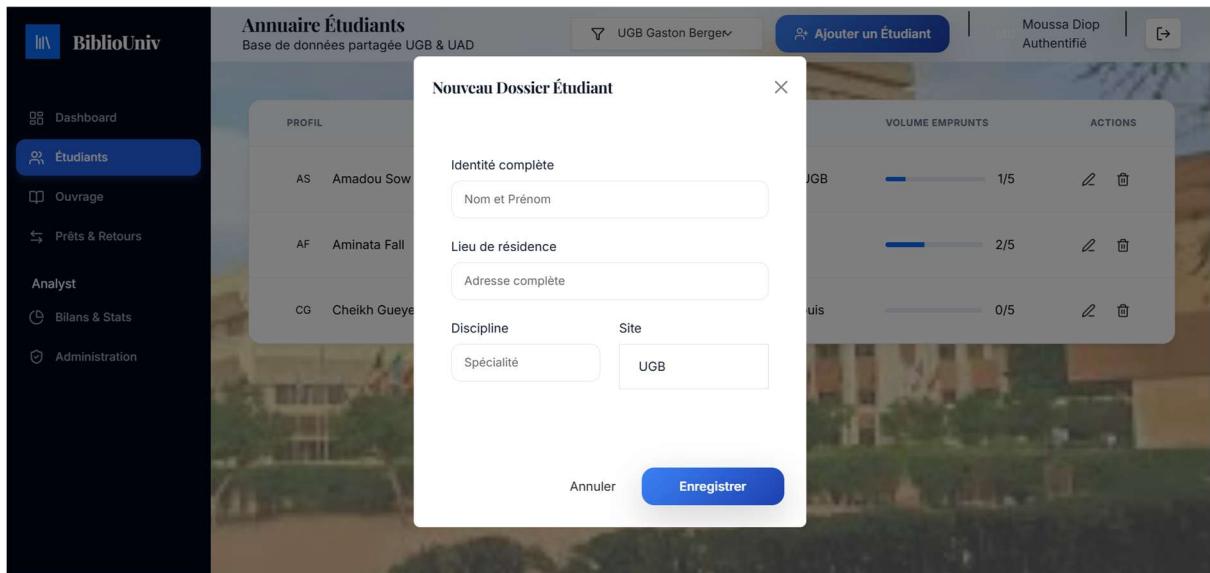
1. **Module d'Authentification** (login.html) : Connexion par identifiant employé et sélection d'université



2. **Tableau de Bord** (dashboard.html)



3. Gestión des Étudiants (etudiants.html) : CRUD complet : Ajout, modification, suppression d'étudiants



4. Gestión des Ouvrages (ouvrages.html) : CRUD complet : Ajout, modification, suppression de livres

OUVRAGE	ÉDITEUR / ANNÉE	DOMAINE	ÉTAT STOCK	LOCALISATION	ACTIONS
Les Misérables Gallimard	Gallimard (1862)	Littérature	2 dispo.	UGB	
Une si longue lettre NEA	NEA (1979)	Littérature	2 dispo.	UGB	

5. Prêts & Retours (emprunts.html) Enregistrement de prêts : Sélection étudiant/ouvrage avec vérification de disponibilité

BÉNÉFICIAIRE	OUVRAGE & SITE SOURCE	DATE D'EMPRUNT	TRAJET DE PRÊT	STATUT ACTUEL	GESTION
Amadou Sow	Things Fall Apart SITE : UAD	2026-01-17	UGB → UAD	En circulation	
Aminata Fall	Things Fall Apart SITE : UAD	2026-01-17	UGB → UAD	En circulation	
Amadou Sow	Les Misérables SITE : UGB	2026-01-17	UGB → UGB	Rendu le 2026-01-18	
Cheikh Gueye	Les Misérables SITE : UGB	2026-01-17	UGB → UGB	Rendu le 2026-01-18	
Cheikh Gueye	Things Fall Apart SITE : UAD	2026-01-18	UGB → UAD	Rendu le 2026-01-18	
Aminata Fall	Things Fall Apart SITE : UAD	2026-01-18	UGB → UAD	En circulation	
Aminata Fall	Les Misérables SITE : UGB	2026-01-18	UGB → UGB	Rendu le 2026-01-19	

6. Bilans & Statistiques (requetes.html) : Vue globale partagée : Agrégation des données UGB + UAD

Fonctionnalités Clés

1. Transparence de la Localisation

L'utilisateur interagit avec une interface unique sans se soucier de la localisation physique des données. Le middleware index.php :

- Route automatiquement les requêtes vers la bonne base de données
- Agrège les résultats des deux sites
- Gère les transactions distribuées (prêts inter-sites)

2. Gestion des Prêts Inter-Sites

Un étudiant de l'UGB peut emprunter un livre de l'UAD et vice-versa :

- Transaction 2PC : Mise à jour du stock sur le site du livre ET du compteur d'emprunts sur le site de l'étudiant
- Stockage dérivé : Le prêt est enregistré dans la base de l'étudiant avec référence au site_ouvrage
- Contrainte de retour : Un livre doit être retourné dans sa bibliothèque d'origine

3. Normalisation des Données

Le frontend normalise les différences de casse entre PostgreSQL et MySQL :

- IdEtud (PostgreSQL) vs idetud (MySQL) → idEtud

- nbreEmprunts vs nbreemprunts → nbreEmprunts
- Garantit la cohérence d'affichage

4. Gestion des Collisions d'Identifiants

Les deux sites utilisent des séquences d'ID indépendantes (ID 1 existe à UGB et UAD) :

- Solution : Transmission explicite de l'université d'origine (universite_etudiant) dans les requêtes
- Lookup composite : Recherche par (ID, Université) pour éviter les ambiguïtés

5. Filtrage Contextuel

- Pages locales (emprunts.html) : Affichent uniquement les données gérées par le site connecté
- Pages globales (requestes.html, dashboard.html) : Agrègent les données des deux sites

Flux Utilisateur Typique

Scénario : Prêt Inter-Sites

- Connexion : Employé UGB (ID: 1) se connecter
- Navigation : Accès à "Prêts & Retours"
- Sélection :
 - Étudiant : "Amadou Sow (UGB)"
 - Livre : "Things Fall Apart (UAD)"
- Validation :
 - Frontend envoie {id_etudiant: 1, id_ouvrage: 1, universite_etudiant: "UGB"}
 - Backend vérifie stock UAD et limite d'emprunts UGB
- Transaction :
 - OUVRAGE_uad.stock : 1 → 0
 - ETUDIANT_ugb.nbreEmprunts : 0 → 1
 - PRET (UGB) : Nouvelle ligne avec site_ouvrage = 'UAD'
- Affichage : Trajet "UGB → UAD" visible dans la liste
- Scénario : Consultation Globale
 - Connexion : Employé UAD
 - Navigation : "Bilans & Stats"

- Requête : "Top Emprunteurs"
- Agrégation :
 - Backend fusionne ETUDIANT_ugb et ETUDIANT_uad
 - Trié par nbreEmprunts décroissant
- Affichage : Liste unifiée avec badges UGB/UAD

VIII. Conclusion et Perspectives

Conclusion

Le projet BIBLIOUNIV a permis de concevoir et d'implémenter un système complet de gestion de bibliothèque distribuée pour deux sites universitaires (UGB et UAD), démontrant la faisabilité d'une architecture offrant autonomie locale et coopération inter-sites.

Réalisations

L'application développée répond intégralement aux exigences :

- Fragmentation des données : Mise en œuvre de la fragmentation horizontale (étudiants, employés, ouvrages par site), dérivée (prêts stockés avec l'étudiant) et réPLICATION totale (table AUTEUR)
- Transparence de la localisation : Le middleware PHP masque la complexité de la distribution via une interface unique
- Transactions distribuées : Implémentation 2PC pour les prêts/retours inter-sites garantissant la cohérence ACID
- Interface complète : Application web moderne couvrant CRUD, gestion des prêts et statistiques globales

Défis Résolus

- Collision d'identifiants : Transmission explicite de l'université d'origine et lookup composite (ID, Université)
- Hétérogénéité SGBD : Normalisation des différences PostgreSQL/MySQL (casse des colonnes)

Perspectives

- Authentification JWT avec sessions sécurisées
- Notifications email pour retours en retard
- Indexation des colonnes fréquentes
- Recommandations basées sur l'historique

- Export PDF/Excel des rapports
- Migration React/Vue.js (frontend) + Node.js (backend)
- Recommandations personnalisées par profil étudiant