

Rapport de Projet : Système de Gestion de Bibliothèque avec Laravel 10

1. Présentation du Projet

Ce projet a pour objectif la **conception et le développement d'un système complet de gestion de bibliothèque universitaire** en utilisant le framework **Laravel 10**.

L'application permet une gestion efficace des aspects clés d'une bibliothèque, incluant :

- La gestion des **abonnés**.
- La tenue du **catalogue des livres**.
- Le suivi des **prêts et retours de livres**.
- L'automatisation du calcul et de la gestion des **pénalités de retard**.

2. Environnement Technique

Le tableau ci-dessous détaille l'environnement technique utilisé pour le développement de ce système :

Technologie	Utilisée
Framework	Laravel 10
Langage	PHP 8.x
Base de données	MySQL (via phpMyAdmin - XAMPP)
Frontend	Blade (Laravel), Tailwind CSS
Authentification	Laravel Breeze
Serveur de dev.	Artisan Serve

3. Modèle de Données

Entités Principales

Le système repose sur les entités suivantes, modélisant les données essentielles de la bibliothèque :

- **Abonné** : nom, prénom, classe, email (unique), période d'abonnement (date de début et de fin).
- **Livre** : titre, auteur, sujet, stock (nombre d'exemplaires disponibles), identifiant unique.
- **Prêt** : abonné, livre, date d'emprunt, date retour prévue, date retour effective (peut être nulle).
- **Pénalité** : liée à un prêt, montant, statut (réglé ou non réglé).

4. Fonctionnalités Développées

Le système intègre un ensemble de fonctionnalités robustes pour la gestion de la bibliothèque :

Gestion des Abonnés

- **CRUD complet** (Création, Lecture, Mise à jour, Suppression) des informations des abonnés.
- **Recherche dynamique** par nom, prénom et classe.

Gestion des Livres

- **CRUD complet** des livres.
- **Gestion automatique du stock** disponible en fonction des prêts et retours.
- **Recherche multicritères** par titre, auteur et sujet.

Gestion des Prêts

- **Limitation à 3 prêts simultanés** par abonné.
- **Vérification systématique du stock disponible** avant tout enregistrement de prêt.
- **Prévention du double emprunt** du même livre par un abonné avant retour.
- **Historique complet des prêts** pour chaque abonné.

Gestion des Retours et Pénalités

- **Calcul automatique des pénalités** de retard (100 FCFA par jour de retard).
- **Blocage automatique de l'abonné** en cas de pénalité non réglée, l'empêchant de faire de nouveaux emprunts.

5. Difficultés Rencontrées et Solutions Apportées

Le développement a présenté certains défis techniques, résolus comme suit :

- **Gestion du stock en temps réel** : Résolu grâce à l'utilisation judicieuse des **jointures Eloquent** pour maintenir la cohérence des données.
- **Calcul dynamique des pénalités** : Implémentation de la bibliothèque **Carbon** pour une manipulation précise et efficace des dates, facilitant le calcul des retards.
- **Validation métier robuste** : Mise en place de **multiples vérifications côté serveur** pour garantir l'intégrité et la cohérence des données, évitant les erreurs d'entrée et les comportements inattendus.

6. Évolutions Futures Possibles

Pour enrichir davantage le système, les améliorations suivantes pourraient être envisagées :

- **Intégration d'un système d'envoi d'e-mails automatique** pour notifier les abonnés des pénalités ou des rappels de retour.
- **Génération de statistiques d'utilisation de la bibliothèque** (livres les plus empruntés,

- périodes d'affluence, etc.).
- **Fonctionnalité d'exportation PDF des rapports de pénalités.**
- **Gestion multi-utilisateurs** avec différents rôles (ex: plusieurs bibliothécaires avec des droits distincts).
- **Listes déroulantes dynamiques** (autocomplétion) lors de la création de prêts pour la sélection des abonnés et des livres, améliorant l'ergonomie.

7. Conclusion

Ce projet a été une opportunité d'appliquer et de consolider de nombreuses compétences techniques liées au framework Laravel. J'ai pu maîtriser l'**ORM Eloquent**, la gestion des **relations entre modèles**, les **validations**, les **middlewares**, la mise en place d'une **authentification sécurisée avec Laravel Breeze**, ainsi que l'intégration du **frontend avec Blade et Tailwind CSS**.

L'application est désormais **pleinement opérationnelle et prête à être déployée en production**, offrant une solution complète et fiable pour la gestion d'une bibliothèque.

8. Auteur

AHOUGNON Yannis

Version 1.0

Projet personnel complet de gestion de bibliothèque développé avec Laravel 10.

Guide Utilisateur Fonctionnel du Système de Gestion de Bibliothèque

Ce guide décrit les étapes essentielles pour utiliser le système de gestion de bibliothèque.

1. Connexion au Système

Lorsque vous accédez à l'application, la première page est celle de **connexion**.

- Seul le **bibliothécaire** est autorisé à se connecter (les comptes utilisateurs sont créés directement dans la base de données).
- Une fois connecté, vous accédez au **Dashboard principal**, qui inclut une barre latérale de navigation.

2. Menu Principal (Barre Latérale)

La barre latérale offre un accès rapide aux différentes sections de l'application :

- **Dashboard** : Page d'accueil du système.
- **Abonnés** : Pour gérer les membres inscrits à la bibliothèque.
- **Livres** : Pour gérer le catalogue et le stock de livres.
- **Prêts** : Pour suivre et gérer les emprunts en cours.
- **Pénalités** : Pour administrer les retards de retour et les frais associés.

3. Gestion des Abonnés

A. Ajouter un Nouvel Abonné

1. Naviguez vers **Abonnés** → **Ajouter un abonné**.
2. Remplissez les champs suivants :
 - **Nom**
 - **Prénom**
 - **Classe**
 - **Email** (doit être unique)
 - **Dates d'abonnement** (début et fin).
3. Cliquez sur **Enregistrer**.

B. Rechercher un Abonné

- Un champ de recherche est disponible sur la page de liste des abonnés.
- Saisissez un **nom**, **prénom** ou **classe**, puis appuyez sur **Rechercher**. La liste s'actualisera automatiquement.
- Cliquez sur **Réinitialiser** pour afficher la liste complète des abonnés.

C. Modifier ou Supprimer un Abonné

Depuis la liste des abonnés, vous pouvez interagir avec chaque enregistrement :

- Cliquez sur le bouton **Modifier** (vert) pour corriger les informations d'un abonné.
- Cliquez sur le bouton **Supprimer** (rouge) pour supprimer définitivement un abonné.
- Cliquez sur le bouton **Voir** (bleu) pour afficher les détails complets d'un abonné sur une page dédiée.

4. Gestion des Livres

A. Ajouter un Livre

1. Accédez à **Livres** → **Ajouter un livre**.
2. Remplissez les champs requis :
 - **Titre**
 - **Auteur**
 - **Sujet**
 - **Identifiant unique** (doit être unique)
 - **Stock total disponible**
3. Cliquez sur **Enregistrer**.

B. Rechercher un Livre

Trois champs de filtre sont disponibles sur la page des livres :

- **Par titre**
- **Par auteur**

- **Par sujet** La liste se mettra à jour lorsque vous cliquerez sur **Rechercher** si au moins un champ de recherche est renseigné. Si aucun champ n'est spécifié, le bouton de recherche affichera la liste complète des livres.

C. Modifier ou Supprimer un Livre

Comme pour les abonnés, les boutons **Voir**, **Modifier** (vert) et **Supprimer** (rouge) sont présents sur chaque ligne de la liste des livres.

5. Gestion des Prêts

Création d'un Prêt

A. Accès à la page de création

1. Depuis le menu **Prêts** → **Nouveau prêt**.
2. La page affiche deux listes déroulantes :
 - Sélection de l'**abonné**.
 - Sélection du **livre** (avec le stock restant affiché).
3. Remplissez :
 - **Date d'emprunt** (par défaut, la date du jour).
 - **Date de retour prévue**.

B. Contrôles automatiques (logique métier Laravel)

Lors de l'enregistrement du prêt, le système effectue des vérifications invisibles pour garantir l'intégrité des données :

- ☒ Vérifie si l'abonné n'a pas dépassé la limite de **3 prêts actifs**.
- ☒ S'assure que l'abonné n'a **pas déjà emprunté le même livre** avant son retour.
- ☒ Contrôle si le **stock du livre est suffisant** (au moins 1 exemplaire disponible).

C. Après validation

- Le prêt est enregistré dans le système.
- Le **stock disponible du livre diminue automatiquement**, grâce au comptage dynamique des prêts en cours.

Affichage de la Liste des Prêts

Sur la page **Prêts**, chaque ligne présente les informations clés d'un emprunt :





Abonné	Livre	Dates	Retour effectif	Pénalité	Actions
--------	-------	-------	-----------------	----------	---------

- Si le prêt est **en cours** (livre non retourné) :
 - ☒ Le bouton **Retour** (bleu) apparaît.
- Si une pénalité est associée à ce prêt :
 - ☒ Le **montant** et le **statut** (réglé / non réglé) sont affichés.
 - ☒ Si la pénalité est non réglée, le bouton **Régler** (violet) apparaît.

Fonctionnement du Bouton "Retour"

Quand vous cliquez sur le bouton **Retour** pour un prêt en cours :

Le système :

-  Enregistre la **date de retour effective** (date du jour).
-  **Compare la date de retour prévue à la date effective.**
-  Si le livre est en retard, une **pénalité est automatiquement créée.**
-  Si le livre est rendu à temps, **aucune pénalité n'est générée.**




Simultanément :

- Le **stock du livre augmente automatiquement** d'un exemplaire.
- La ligne du prêt bascule : soit vers l'historique, soit reste visible sur la liste des prêts courants si une pénalité non réglée est présente.

Fonctionnement du Bouton "Régler la Pénalité"

Lorsque vous cliquez sur le bouton **Régler** :

Le système :

-  Change le statut de la pénalité à « **Réglée** ».
-  **Vérifie si l'abonné est désormais débloqué** pour de nouveaux prêts.
-  Retire cette pénalité des pénalités actives.

Simultanément :

- Sur la liste des prêts, le prêt disparaît de la liste principale (s'il était visible uniquement à cause de la pénalité).
- Sur la liste des pénalités, la pénalité apparaît désormais comme « **Réglée** ».

6. Gestion des Pénalités

A. Calcul Automatique

- Lors du retour d'un livre :
 - Si le livre est rendu en retard, le système calcule automatiquement la pénalité (fixée à **100 FCFA par jour de retard**).
 - Une pénalité est alors automatiquement enregistrée.

B. Régler une Pénalité

1. Accédez à la section **Pénalités**.
2. Pour chaque pénalité non réglée, cliquez sur le bouton **Régler** (bleu).
3. Une fois réglée, l'abonné redevient éligible à de nouveaux prêts.

C. Supprimer ou Modifier une Pénalité

Les boutons classiques **Modifier** (violet) et **Supprimer** (rouge) sont disponibles pour corriger ou supprimer manuellement une pénalité si nécessaire.

7. Sécurité

- Toutes les pages de l'application sont sécurisées via le système d'**authentification de Laravel Breeze**.
- Des **validations côté serveur robustes** sont implémentées pour prévenir toute manipulation abusive ou incohérence des données.

8. Cas Pratiques Importants

Ce tableau résume les comportements clés du système face à différentes situations :

Cas	Comportement du système
L'abonné a atteint 3 prêts actifs	Refus de créer un nouveau prêt
Pas d'exemplaire de livre disponible	Refus de créer un nouveau prêt
Retour d'un livre en retard	Calcul automatique d'une pénalité
Pénalité non réglée	Abonné bloqué pour tout nouvel emprunt

9. Remarque Technique

Toutes les modifications de stock, le calcul des pénalités et les validations sont gérés **automatiquement par le système**, sans nécessiter d'intervention manuelle du bibliothécaire.

10. Conclusion

L'application a été conçue avec pour objectifs principaux :

- Assurer une **gestion fluide et sécurisée** de la bibliothèque.
- Offrir une **interface simple et intuitive**, même pour un bibliothécaire n'ayant pas de compétences techniques avancées.
- Faciliter la **maintenance** du système grâce à l'utilisation des bonnes pratiques de Laravel et MySQL.



FIN DU GUIDE

Comment obtenir votre PDF :