

**Licence 3 parcours MIAGE
Introduction aux systèmes d'information**

**Cahier de charges et rapport du projet
SYSTÈME DE GESTION DE BIBLIOTHÈQUE**

Quynh NGUYEN - 44010253
Thi Huyen NGO - 44003743

2025 - 2026

Cahier des charges du projet

1. Introduction et contexte

Dans un contexte de **transformation numérique accélérée**, les bibliothèques universitaires se trouvent confrontées à la nécessité d'optimiser leurs méthodes de gestion et d'accès à l'information. Les systèmes traditionnels basés sur la gestion manuelle (fiches papier, registres physiques, suivi non automatisé) entraînent plusieurs difficultés :

- une **perte de temps considérable** dans le traitement des emprunts et retours ;
- une **augmentation des erreurs humaines** lors de la saisie ou du suivi des livres ;
- un **accès limité** aux données en dehors des horaires d'ouverture ;
- et une **absence de statistiques fiables** permettant d'analyser la fréquentation et les emprunts.

Afin de répondre à ces besoins de modernisation, notre projet vise à **concevoir un système d'information web complet** pour la gestion d'une bibliothèque universitaire. Ce projet s'inscrit dans le cadre du module *Introduction aux Systèmes d'Information* de la Licence 3 parcours MIAGE, et permet de mettre en pratique les compétences en **analyse, conception, développement et intégration d'un système d'information moderne**.

Le public cible comprend :

- les **étudiants**, utilisateurs finaux du système, qui souhaitent emprunter, réserver ou consulter les ouvrages ;
- les **bibliothécaires**, responsables de la gestion des stocks, des demandes et du suivi des retours ;
- et l'**administration universitaire**, qui supervise l'activité et analyse les données d'utilisation.

En somme, le projet s'inscrit dans une démarche d'innovation et de digitalisation des services documentaires universitaires, tout en favorisant l'efficacité, la fiabilité et l'accessibilité des ressources.

2. Objectifs du projet

L'objectif principal du projet est de **concevoir et développer une application web de gestion de bibliothèque universitaire** permettant d'automatiser et de centraliser les processus de prêt, de retour et de suivi des ouvrages.

2.1 Objectif général

Développer un **système d'information complet, ergonomique et sécurisé** qui facilite la gestion quotidienne d'une bibliothèque universitaire, tout en améliorant l'expérience utilisateur.

2.2 Objectifs spécifiques

- **Simplification des processus** : automatiser les tâches manuelles telles que les emprunts, retours et demandes de livres.
- **Accessibilité** : offrir une plateforme web disponible 24h/24, compatible avec ordinateurs, tablettes et mobiles.
- **Fiabilité** : assurer la cohérence des données grâce à une base MySQL robuste.
- **Sécurité** : protéger les comptes utilisateurs par authentification et hachage des mots de passe.
- **Performance** : permettre la consultation rapide des statistiques (livres empruntés, retards, utilisateurs actifs).
- **Évolutivité** : concevoir une architecture modulaire, capable d'intégrer ultérieurement des fonctions d'IA ou de recommandations personnalisées.

2.3 Objectifs pédagogiques

Ce projet a également une dimension pédagogique : il vise à renforcer les compétences acquises en MIASH-S/MIAGE dans les domaines suivants :

- **Analyse et modélisation** d'un besoin organisationnel réel ;
- **Conception et développement** d'une application web complète ;
- **Utilisation d'outils modernes** (VS Code, phpMyAdmin, GitHub, Apache) dans un cadre professionnel ;
- **Travail en binôme**, planification et répartition des tâches selon une démarche projet structurée.

3. Analyse fonctionnelle

Le système de gestion de bibliothèque repose sur deux profils principaux d'utilisateurs : les **étudiants**, qui utilisent le service pour consulter et emprunter des ouvrages, et les **bibliothécaires**, qui assurent la gestion administrative et technique du système.

3.1 Fonctionnalités côté étudiant

- Création d'un compte et connexion sécurisée ;
- Recherche de livres par titre, auteur ou catégorie ;
- Consultation de la disponibilité des ouvrages ;
- Demande d'emprunt en ligne ;
- Suivi de l'état des demandes et de l'historique des emprunts ;
- Modification du profil personnel.

3.2 Fonctionnalités côté bibliothécaire

- Authentification et accès à un tableau de bord sécurisé ;
- Gestion des livres (ajout, suppression, modification, consultation) ;
- Gestion des catégories et des comptes étudiants ;
- Validation ou refus des demandes d'emprunt ;
- Suivi des retours et gestion des retards ;
- Visualisation des statistiques globales.

3.3 Fonctionnalités globales du système

- Interface claire et responsive pour tous les appareils ;
- Gestion de sessions et différenciation des rôles ;
- Formulaire de contact et envoi d'emails via SMTP ;
- Sécurité applicative (hachage, filtrage SQL, gestion des erreurs).

En résumé, cette analyse fonctionnelle définit les besoins et interactions essentiels entre les différents acteurs du système, garantissant une organisation fluide et un service performant.

4. Spécifications techniques

Le système de gestion de bibliothèque repose sur une **architecture web dynamique** développée à l'aide de technologies courantes et accessibles, garantissant à la fois performance, sécurité et évolutivité.

4.1 Environnement technique

- **Langages utilisés :**
 - PHP 8.x — pour la logique serveur et la communication avec la base de données ;
 - HTML5 / CSS3 — pour la structure et la mise en forme des pages ;
 - JavaScript (ES6) — pour les interactions dynamiques et les animations ;
 - SQL (MySQL) — pour le stockage et la gestion des données.
- **Serveur web :** Apache (intégré à XAMPP ou WAMP).
- **Système de gestion de base de données :** MySQL 5.7+.
- **Outils de développement :** Visual Studio Code, phpMyAdmin, GitHub pour le suivi de version.

4.2 Sécurité et fiabilité

- **Authentification sécurisée** : gestion de sessions PHP et redirection automatique en cas de non-connexion ;
- **Protection des mots de passe** : hachage via l'algorithme `password_hash()` ;
- **Validation des entrées** : filtrage des champs contre les injections SQL et failles XSS ;
- **Gestion des erreurs** : affichage contrôlé pour éviter la divulgation d'informations sensibles.

4.3 Performances et accessibilité

- Interface **responsive** et adaptée à tous les écrans (ordinateurs, tablettes, smartphones) ;
- Temps de chargement optimisé grâce au **loader GIF** et au découpage modulaire des pages ;
- Compatibilité multi-navigateurs (Chrome, Edge, Firefox) ;
- Respect des normes **W3C** et de l'accessibilité **WCAG** niveau A.

4.4 Architecture logicielle

Le projet adopte une structure en couches inspirée du modèle MVC simplifié :

- **Présentation (Front-end)** — pages web et feuilles de style.
- **Logique métier (Back-end)** — traitement des requêtes et vérifications des données.
- **Données (Base MySQL)** — tables et relations pour gérer livres, utilisateurs et emprunts.

L'ensemble garantit une application fiable, maintenable et extensible pour de futures évolutions technologiques.

5. Architecture et structure du projet

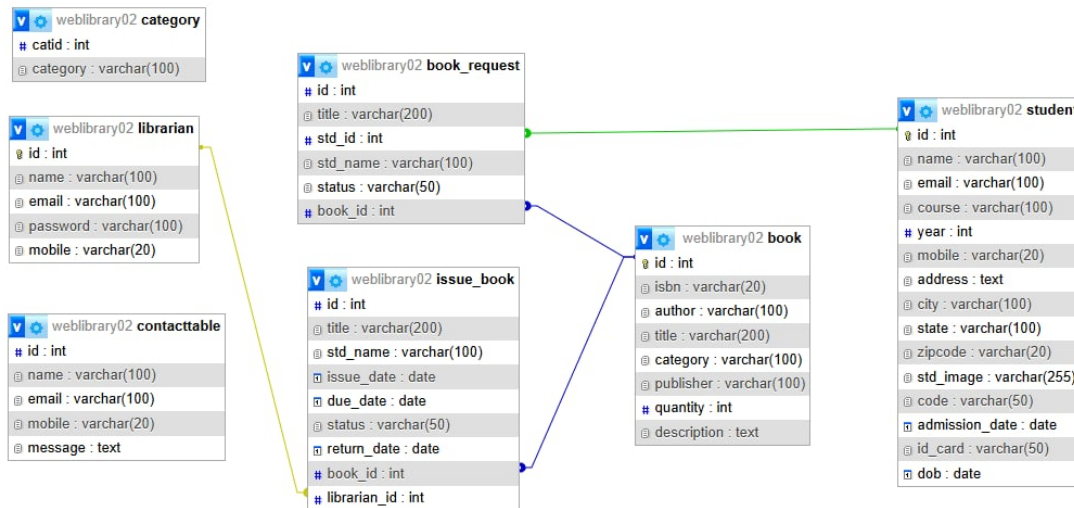
L'application est divisée en plusieurs modules interdépendants, organisés selon une logique fonctionnelle claire.

5.1 Organisation générale

- **Module principal** – `index.php` : point d'entrée du site, centralise la navigation et le routage.
- **Module d'administration** – `Panel/` :
 - Gestion des livres : ajout, suppression, modification ;
 - Gestion des utilisateurs : étudiants et bibliothécaires ;
 - Suivi des emprunts et retours.
- **Module étudiant** – `Student/` :
 - Tableau de bord individuel ;

- Historique des emprunts ;
- Mise à jour du profil personnel.
- **Module contact** – assets/webpages/contact.php :
 - Formulaire d’envoi de message ;
 - Enregistrement des requêtes dans la table contacttable.
- **Base de données** – librarydb.sql : contient les tables book, student, librarian, category, book_request, issue_book et contacttable.

5.2 Schéma conceptuel de la base de données



Le modèle relationnel relie les livres, les utilisateurs et les transactions à travers des clés primaires et étrangères. Cette organisation permet une **cohérence forte des données** et un suivi précis de chaque opération.

5.3 Interaction entre les modules

- Les bibliothécaires interagissent avec les étudiants à travers les requêtes de prêt ;
- Les modules Panel et Student partagent la même base de données ;
- Le module index.php agit comme intermédiaire entre les pages publiques et les modules internes.

Cette architecture modulaire rend le projet évolutif, facile à maintenir et réutilisable dans d'autres contextes institutionnels.

6. Maquettes et ergonomie

L'interface utilisateur a été pensée pour être **simple, intuitive et cohérente**. Chaque page du site respecte les principes de design ergonomique afin d'assurer une expérience fluide.

6.1 Charte graphique et design

- Palette de couleurs dominée par le **bleu universitaire** (#6c5dd4) et le blanc ;
- Police Poppins, moderne et lisible ;
- Icônes vectorielles issues de Boxicons et FontAwesome ;
- Structure CSS basée sur Flexbox et Grid pour l’adaptabilité.

6.2 Maquettes principales

Page d’accueil : Présente le logo, une brève description du service, et des boutons d’accès rapide (« Connexion », « Catalogue », « Contact »). Un visuel de fond est ajouté pour une ambiance immersive.

Tableau de bord étudiant : Permet de visualiser les livres empruntés, les dates de retour et les notifications de retard. Les actions principales sont accessibles en un clic (demande de livre, modification du profil).

Tableau de bord bibliothécaire : Affiche les statistiques clés : nombre de livres disponibles, livres empruntés, demandes en attente. Inclut des raccourcis pour la gestion des catégories et des utilisateurs.

Page contact : Intègre un formulaire d’envoi de message et une carte Google Maps localisant la bibliothèque.

6.3 Principes d’ergonomie appliqués

- **Cohérence visuelle** entre les pages pour éviter la désorientation ;
- **Navigation simple** avec barre fixe et liens ancrés ;
- **Accessibilité** : contraste suffisant et boutons larges pour le mobile ;
- **Feedback utilisateur** : messages de confirmation, pop-ups et survols colorés.

Ces choix d’ergonomie et de design visent à renforcer la convivialité du site, tout en valorisant l’image moderne de la bibliothèque universitaire.

7. Planning prévisionnel

Afin de structurer le développement du projet et de respecter les délais imposés, un **planning prévisionnel** a été établi. Celui-ci couvre l’ensemble des phases — de l’analyse des besoins à la mise en ligne — en intégrant les étapes de validation et de test.

7.1 Découpage des phases du projet

Étape	Description	Durée estimée
1. Analyse et conception	Étude du besoin, rédaction du cahier des charges, conception du MCD/MLD et architecture du site.	1 semaine
2. Développement front-end	Création des pages HTML, intégration CSS et test d’affichage sur différents supports.	2 semaines
3. Développement back-end	Programmation PHP/MySQL, configuration de la base de données et gestion des sessions.	3 semaines
4. Phase de test et validation	Vérification des fonctionnalités, correction d’erreurs, renforcement de la sécurité.	1 semaine
5. Livraison et rapport final	Déploiement local du projet, rédaction du rapport et soutenance orale.	1 jour

7.2 Répartition des rôles (binôme)

- **NGO Thi Huyen** : conception de la base de données, développement back-end (PHP/MySQL), validation fonctionnelle.
- **NGUYEN Quynh** : conception front-end (HTML/CSS/JS), rédaction du cahier des charges, intégration visuelle et tests d’ergonomie.

7.3 Gestion du temps et outils de suivi

- Outil de suivi : GitHub Projects et historique de commits.
- Communication : Discord et Google Docs.
- Sauvegardes régulières sur GitHub.

Ce planning a permis de maintenir une progression équilibrée, une communication efficace entre les membres, et une livraison finale conforme aux attentes.

8. Conclusion et perspectives

Ce projet de **système de gestion de bibliothèque universitaire** constitue une étape importante dans notre apprentissage du développement web et de la conception de systèmes d’information. Il a permis de passer de la théorie à la pratique en mobilisant plusieurs compétences transversales : analyse fonctionnelle, modélisation de base de données, développement, ergonomie et documentation technique.

8.1 Apports du projet

- Acquisition d’une méthodologie de travail rigoureuse (analyse, conception, implémentation, test).
- Mise en œuvre des langages et outils étudiés au cours du module (PHP, SQL, CSS, GitHub).
- Développement de compétences en collaboration, autonomie et résolution de problèmes.

8.2 Limites rencontrées

- Quelques difficultés initiales liées à la configuration du serveur local (Apache/MySQL).
- Gestion complexe des sessions multiples (étudiant/bibliothécaire).
- Manque de temps pour intégrer certaines fonctionnalités avancées (statistiques dynamiques, export PDF).

8.3 Perspectives d'évolution

Afin de poursuivre l'amélioration du projet, plusieurs axes de développement sont envisagés :

- Intégration d'un **module de recommandation de livres** basé sur les préférences des utilisateurs ;
- Mise en place d'un **système de notifications automatiques** (email ou SMS) pour les retards ;
- Génération de **rapports statistiques** automatiques pour les bibliothécaires ;
- Hébergement du projet sur un **serveur distant** (AWS ou Hostinger) pour un accès universel.

Ce projet constitue une base solide pour de futures applications universitaires et professionnelles, et ouvre la voie à une gestion numérique plus efficace des ressources documentaires.

9. Rapport du projet (Récapitulatif de la réalisation)

Le développement du site s'est déroulé selon une approche incrémentale, favorisant la vérification progressive de chaque module avant intégration.

9.1 Étapes de réalisation

- **Phase d'analyse** : identification des besoins et élaboration du schéma conceptuel de la base.
- **Phase de conception** : création du prototype de l'interface et planification du back-end.
- **Phase de développement** : écriture du code, tests unitaires, gestion des sessions et validation des formulaires.
- **Phase de test et correction** : simulation des scénarios d'emprunt et d'administration.

9.2 Difficultés rencontrées et solutions

- **Problème** : incohérence des données lors des suppressions multiples. **Solution** : mise en place de contraintes de clé étrangère (ON DELETE CASCADE).
- **Problème** : lenteur du chargement sur mobile. **Solution** : optimisation des images et compression CSS.
- **Problème** : validation incomplète des formulaires. **Solution** : ajout de filtres `htmlspecialchars()` et contrôle côté serveur.

9.3 Résultats obtenus

Le résultat final est une **application web complète et fonctionnelle**, comprenant :

- Une interface publique pour les étudiants (consultation, emprunt, profil).
- Une interface d'administration pour les bibliothécaires (gestion, statistiques).
- Une base de données relationnelle robuste et normalisée.
- Un design moderne, responsive et cohérent.

Le projet a été mené à bien dans les délais impartis, répondant aux exigences techniques et pédagogiques fixées par l'encadrant.

Annexes :

- Lien GitHub du projet : <https://github.com/nthuyen1901/systeme-de-gestion-de-bibliotheque>
- Captures d'écran des principales pages de l'application.
- Schéma complet de la base de données.

Rapport du projet

1. Lien vers le projet

Le schéma détaillé ainsi que le code complet du projet sont accessibles via ce dépôt GitHub : <https://github.com/nthuyen1901/systeme-de-gestion-de-bibliotheque>

2. Objectifs du projet

L'objectif principal de ce projet est de concevoir un site web de gestion de bibliothèque avec les fonctionnalités suivantes :

- Gestion des informations sur les livres (ajout, modification, suppression, recherche de livres).
- Gestion des catégories de livres selon le genre, l'auteur et l'éditeur.
- Gestion des lecteurs (inscription, mise à jour des informations, historique des emprunts).
- Suivi du processus d'emprunt et de retour des livres.
- Statistiques sur le nombre de livres par catégorie et état d'emprunt-retour.

3. Périmètre de l'étude

Ce système est conçu pour être utilisé dans une bibliothèque de [école / entreprise / bibliothèque publique...]. Le site web est développé en PHP associé à MySQL pour le stockage des données, et utilise des technologies telles que HTML, CSS, JavaScript pour concevoir une interface utilisateur intuitive et facile à utiliser.

4. Méthodologie de recherche

Pour atteindre cet objectif, nous avons structuré notre projet autour d'un fichier central, `index.php`, qui joue le rôle de point d'entrée principal du site. Il orchestre l'affichage des différentes pages et la communication avec la base de données.

Le projet est organisé en plusieurs composants techniques qui interagissent avec `index.php`:

- Les assets : regroupent les feuilles de style (CSS), les images et les scripts pour l'affichage et l'interactivité du site.
- La base de données : stocke les informations sur les utilisateurs et les ouvrages disponibles.
- Le loader : optimise le chargement des pages pour une navigation fluide.
- Le panel d'administration : permet de gérer les contenus et d'assurer la maintenance du site.
- Les différentes pages web : composent l'interface utilisateur et sont dynamiquement générées par `index.php`.

Nous avons mobilisé toutes les connaissances acquises à l'université, mais cela ne suffisait pas. Nous avons donc approfondi nos compétences en explorant des forums technologiques et en nous auto-formant à travers des ressources en ligne. L'intelligence artificielle nous a également été d'une grande aide, nous permettant d'identifier et de corriger les erreurs de nos fichiers PHP. Ce fut un parcours semé d'embûches, mais nous sommes fiers du résultat obtenu.

Nous espérons que vous prendrez plaisir à découvrir notre projet et que vous l'apprécierez avec un regard bienveillant. Nous vous remercions sincèrement pour votre attention et votre évaluation.

INTRODUCTION AU CSS

Le fichier `index.css` définit l'apparence du site et garantit une expérience utilisateur cohérente et responsive. Il couvre :

- **Mise en page** : Flexbox et Grid.
- **Charte UI** : couleurs, police Poppins, transitions.
- **Réactivité** : media queries pour l'adaptation multi-écrans.

Synthèse de l'analyse de `index.css`

1) Réinitialisation & global : Remise à zéro des marges et espacements (`margin:0; padding:0;`) pour homogénéiser le rendu, et activation du défilement fluide (`scroll-behavior:smooth;`).

2) Barre de navigation (nav) : Disposition en flex (`justify-content:space-between;`) pour séparer logo et menu, avec une ombre portée (`box-shadow`) pour la profondeur.

- *Logo* : largeur $\approx 60\text{px}$, couleur `#131428`, espacement des lettres.
- *Menu* : éléments en flex avec gap; survol en violet `#6c5dd4`.

3) Section d'accueil (home) : Image de fond avec linear-gradient sombre pour la lisibilité, centrage du contenu avec `display:flex; justify-content:center; align-items:center;`, et hauteur fixée à 80vh. Boutons : `border-radius:6px`, texte 15px, couleur violette `#6c5dd4` sur fond blanc.

4) Vitrine des livres (books-showcase) : Grille à quatre colonnes (`grid-template-columns:repeat(4,1fr);`) avec espacement vertical de 30px. Chaque carte livre possède une bordure légère (`1px solid #f0f0f0`) et des coins arrondis (`border-radius:6px`); les images sont également arrondies pour un rendu homogène.

5) Section contact : Disposition centrée (`align-items:center; justify-content:center;`); les champs du formulaire ont des coins arrondis (`border-radius:10px`) et une bordure fine.

6) Responsive (media queries) : Pour les écrans de moins de 860px, les sections passent en affichage vertical (`flex-direction:column;`) et le menu hamburger devient visible (`display:flex;`).

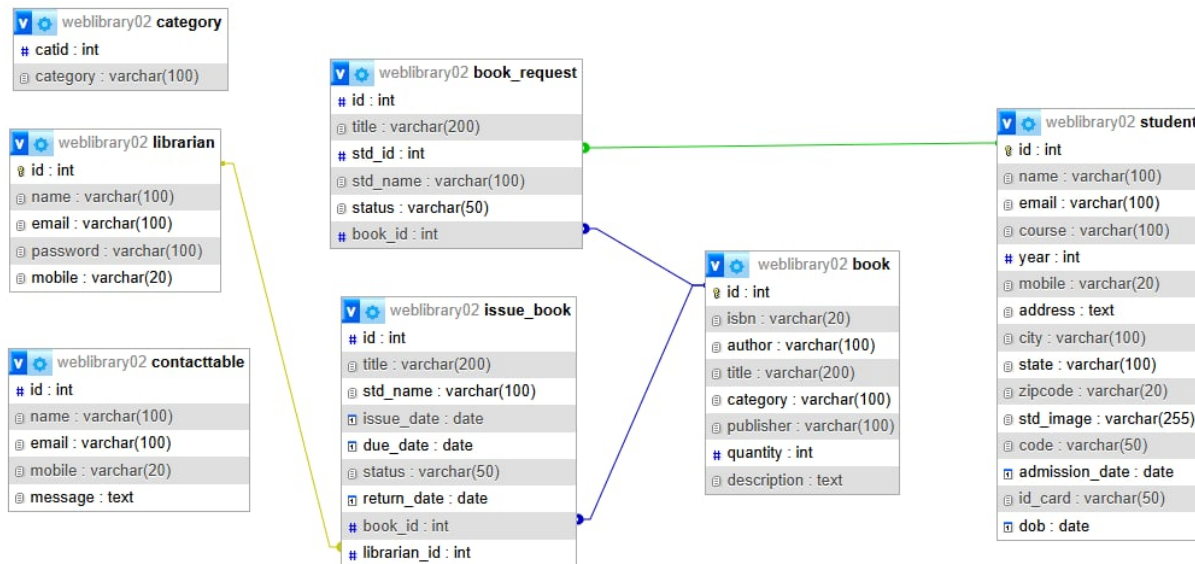
7) Fenêtre modale : Initialement cachée (`display:none;`), affichée via JavaScript; effet d'apparition douce (`transition:.3s ease-in-out;`) et fond semi-transparent (`background-color:rgba(0,0,0,.4);`) pour focaliser l'attention de l'utilisateur.

En bref : La feuille `index.css` fournit une base UI moderne, lisible et adaptée à tous les appareils, en harmonisant navigation, accueil, vitrines de livres, formulaires et éléments interactifs.

INTRODUCTION AUX DATABASE

Ce fichier SQL Dump est une sauvegarde (*backup*) de la base de données librarydb, créée à l'aide de l'outil phpMyAdmin. Il contient l'intégralité de la structure et des données, y compris les tables, les données d'exemple et les configurations associées. Il peut être utilisé pour restaurer ou recréer la base de données sur un autre serveur ou dans un environnement de développement.

SCHEMA :



Le fichier librarydb.sql est un **dump MySQL** généré via phpMyAdmin, contenant la structure complète et les données du système de gestion de bibliothèque universitaire. Il sert à la fois de sauvegarde, de base de migration et de support pour le développement et les tests.

Fonctionnalités principales du fichier SQL Dump

1. **Sauvegarde et restauration des données** : permet d'enregistrer l'ensemble de la base (structure + contenu).
2. **Partage et migration** : facilite le transfert du projet entre environnements ou développeurs.
3. **Création d'un environnement de développement** : utile pour répliquer la base et exécuter des tests fonctionnels.
4. **Analyse de la structure** : offre une vue d'ensemble des tables, clés primaires/étrangères et contraintes.

Structure du schéma relationnel

Ce schéma représente la structure de la base de données utilisée pour gérer les opérations du site web. Il se compose de plusieurs tables interconnectées garantissant l'intégrité et la cohérence des données.

1. Tables principales

- **book (Livres)** : stocke les informations sur les livres (ISBN, auteur, titre, catégorie, éditeur, quantité, description).
Clé primaire : id.
- **student (Étudiants)** : contient les informations des étudiants (nom, email, cours, année, téléphone, adresse, etc.).
Clé primaire : id.
- **librarian (Bibliothécaires)** : enregistre les responsables du système (nom, email, mot de passe, téléphone).
Clé primaire : id.
- **category (Catégories)** : regroupe les livres par catégorie.
Clé primaire : catid.
- **book_request (Demandes de prêt)** : gère les requêtes d'emprunt et leur statut (en attente, accepté, refusé).
Clé primaire : id.
- **issue_book (Emprunts)** : suit les prêts et retours (dates, statut, références du livre et du bibliothécaire).
Clé primaire : id.
- **contactable (Messages)** : conserve les messages envoyés depuis le formulaire de contact.
Clé primaire : id.

2. Relations entre les tables

- **Livre – Catégorie** : relation *un-à-plusieurs* — chaque livre appartient à une catégorie (*category_id*).
- **Étudiant – Demande de prêt** : un étudiant peut émettre plusieurs demandes (*std_id*).
- **Livre – Demande de prêt** : un livre peut être demandé plusieurs fois (*book_id*).
- **Livre – Emprunt** : un livre peut être emprunté à plusieurs reprises (*book_id*).
- **Bibliothécaire – Emprunt** : un bibliothécaire gère plusieurs prêts (*librarian_id*).

Cette structure relationnelle assure la fiabilité des opérations (emprunt, retour, contact) et permet une gestion centralisée et cohérente des ressources de la bibliothèque.

* Utilité du schéma

Ce modèle relationnel facilite une gestion optimale des livres, des emprunts et des utilisateurs dans une bibliothèque universitaire, permettant ainsi une gestion efficace et organisée des ressources et des opérations administratives quotidiennes.

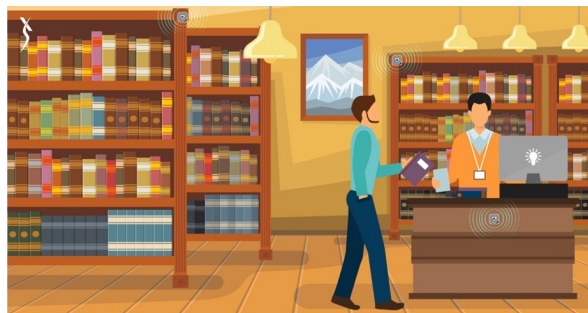
INTRODUCTION AU IMAGE

Cette section regroupe les fichiers d'image nécessaires au bon fonctionnement et à la présentation visuelle du système de gestion de bibliothèque. Les fichiers sont les suivants :

1. **loader.gif** : Un fichier GIF utilisé pour indiquer le chargement en cours. Il est affiché lors des processus nécessitant une attente utilisateur, comme le chargement de données ou l'actualisation des pages, afin d'améliorer l'expérience utilisateur en fournissant un retour visuel.



2. **012.jpg** : Une image utilisée comme couverture d'exemple pour représenter un livre dans le catalogue. Cela aide à illustrer le design du système et montre comment les informations visuelles des livres peuvent être affichées.



3. **chevron-down.svg** : Une icône en format SVG représentant une flèche orientée vers le bas. Elle est utilisée dans l'interface pour indiquer des menus déroulants ou d'autres interactions où l'utilisateur peut afficher plus d'options.



4. **college.jpg** : Le logo de l'établissement, ajouté pour identifier visuellement l'institution associée à la bibliothèque. Cela contribue à renforcer l'identité de la plateforme.



5. **lib.png** : Le logo spécifique de la bibliothèque, utilisé dans l'interface pour personnaliser davantage le système et le rendre distinctif par rapport aux autres services.



Ces éléments visuels jouent un rôle essentiel dans la création d'une interface utilisateur intuitive et esthétiquement plaisante, tout en renforçant l'identité visuelle de l'application.

INTRODUCTION LOADER

Ce fichier inclut une combinaison de CSS, HTML, et JavaScript pour la gestion d'un élément de chargement (loader) dans le système de gestion de bibliothèque. Il est conçu pour améliorer l'expérience utilisateur en affichant une animation pendant le chargement des ressources ou des pages.

```
1 <style>
2   #loader {
3     position: fixed;
4     width: 100%;
5     height: 100vh;
6     background-color: #fff;
7     z-index: 99999;
8     display: flex;
9     align-items: center;
10    justify-content: center;
11  }
12
13  #loader img {
14    width: 20%;
15  }
16 </style>
17 <div id="loader">
18   
19 </div>
20
21 <script>
22   let loader = document.getElementById("loader");
23
24   function preloader() {
25     loader.style.display = 'none';
26   }
27 </script>
```

Structure du fichier:

1. Section CSS :

- L'élément #loader est configuré pour couvrir tout l'écran pendant le chargement, avec un style de position fixe et une couleur de fond blanche.
- Les propriétés display: flex, align-items: center et justify-content: center permettent de centrer verticalement et horizontalement l'image d'animation.
- L'image de chargement est dimensionnée à 20% de la largeur de l'écran pour garantir une présentation claire et non intrusive.

2. Section HTML :

- L'élément <div> avec l'identifiant loader est utilisé comme conteneur pour l'animation de chargement.
- L'image animée (loader.gif) est intégrée via une balise et sert à indiquer que le chargement est en cours.

3. Section JavaScript :

- Une fonction preloader() est définie pour masquer l'élément de chargement (#loader) une fois que les ressources ou les actions nécessaires ont été complètement chargées.
- Cette fonction peut être appelée dans d'autres scripts ou associée à des événements comme window.onload pour déclencher automatiquement le masquage du loader.

=> Utilité dans le projet

Ce fichier joue un rôle crucial dans la fluidité de l'application en informant les utilisateurs que le système est en cours de traitement, réduisant ainsi l'impression de latence. L'animation visuelle améliore la perception globale de performance et la convivialité de l'application.

INTRODUCTION AU PANEL

Organisation

Le répertoire Panel regroupe les fonctionnalités d'administration, l'espace étudiant et les ressources partagées (CSS, images, utilitaires).

Volet Administration (admin-pages/)

- **Livres** : ajout, édition, suppression, listing paginé.
- **Utilisateurs** : création, mise à jour, suppression, recherche.
- **Prêts/Retours** : émission, suivi, rappels e-mail.
- **Demandes** : approbation/refus, création de prêt depuis une demande.
- **Catégories** : gestion du référentiel et filtrage.

Tableau de bord (dashboard.php)

- **Accès** : authentification par session (lib-name).
- **Indicateurs** : emprunts en cours/retours, total livres, total étudiants.
- **Navigation** : menu latéral vers les sous-modules (livres, catégories, prêts, utilisateurs).

Espace Étudiant (student/)

- **Profil** : consultation des informations et historique.
- **Tableau de bord** : échéances de retour, nouveautés, demandes en attente.
- **Déconnexion** : destruction de session et redirection sûre.

Ressources partagées

- **CSS principal** (main.css) : styles et responsive unifiés.
- **Stockage** : img-store (couvertures) et book-pdf (documents).
- **Utilitaires** : pagination SQL (LIMIT/OFFSET), envoi e-mails (PHPMailer), logout, sidebar.

Sécurité & qualité (synthèse)

- Contrôle d'accès par session, validation/échappement des entrées.
- Pagination des grandes listes; possibilité d'AJAX pour mises à jour partielles.
- Améliorations conseillées : CSRF, hachage mots de passe, logs serveurs.

En bref : un back-office modulaire pour gérer livres, utilisateurs et prêts, avec un tableau de bord clair et des utilitaires communs.

INTRODUCTION `index.php`

Rôle du fichier

`index.php` est le point d'entrée de l'application : il initialise la session, se connecte à la base de données, orchestre l'affichage des sections publiques et traite le formulaire de contact.

Traitements serveur (PHP)

- **Session & erreurs** : `session_start()`, configuration minimale d'erreurs.
- **Connexion BDD** : inclusion de `db-connect.php`.
- **Données affichées** : sélection des derniers livres pour la vitrine d'accueil.
- **Contact** : validation basique, insertion en table `contacttable`, envoi d'email de confirmation, affichage d'une modale de succès/erreur.

Structure HTML & assets

- **Head** : métas responsive/SEO, feuilles de style (CSS principal, icônes).
- **Loader** : inclusion du préchargeur pour améliorer la perception de performance.
- **Scripts** : interactions essentielles (menu mobile, fermeture de modales, préchargement).

Sections UI principales

- **Navigation** : menu desktop + hamburger, liens conditionnels selon l'authentification.
- **Accueil** : titre, texte d'intro, CTA qui s'adapte (connexion/tableau de bord).
- **Livres** : grille responsive alimentée par MySQL, lien vers la fiche détaillée.
- **À propos** : description courte du système.
- **Contact** : Google Maps + formulaire validé côté serveur (messages contextuels).
- **Footer** : colonnes (catégories, coordonnées, réseaux), carte embarquée.

Sécurité & bonnes pratiques (synthèse)

- Régénération d'ID de session après login, hachage des mots de passe (côté auth).
- Validation/échappement systématique des entrées.
- Journalisation d'erreurs côté serveur (à la place de `error_reporting(0)`).

En bref : `index.php` connecte les données (BDD) et l'interface (UI) de manière simple et modulaire pour l'expérience publique du site.

RÉSULTATS DU PROCESSUS DE CRÉATION DU SITE WEB

Voici les captures d'écrans des résultats obtenu après l'exécution du programme.

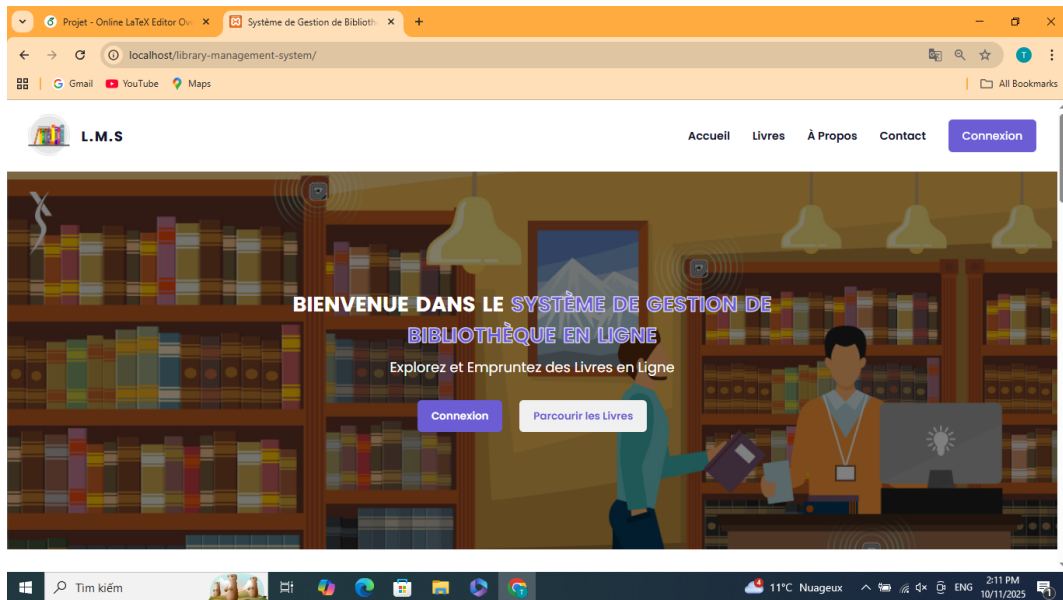


Figure 1: Page d'accueil du système

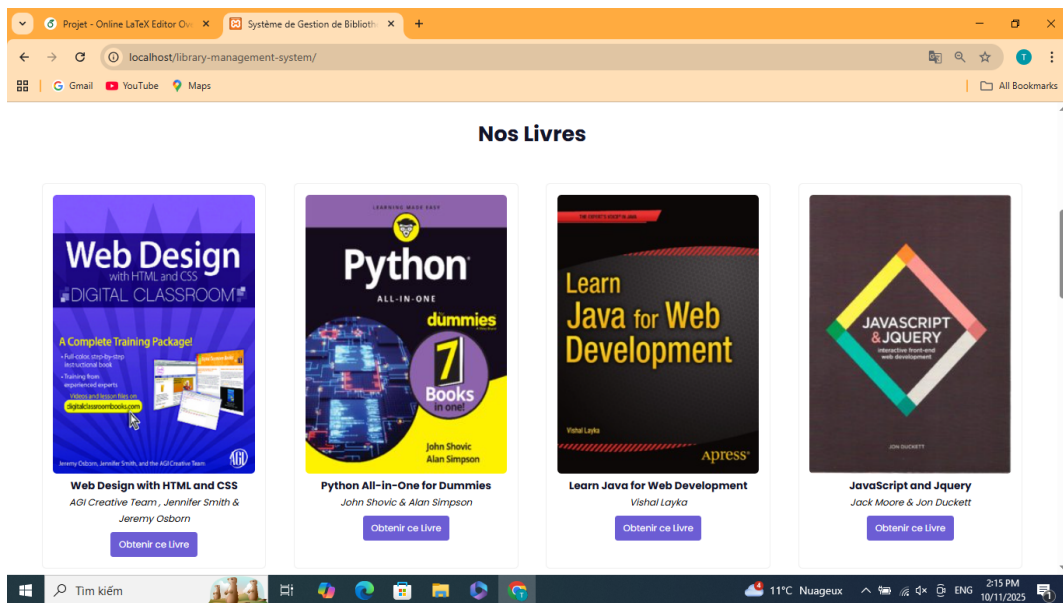


Figure 2: Affichage des livres disponibles

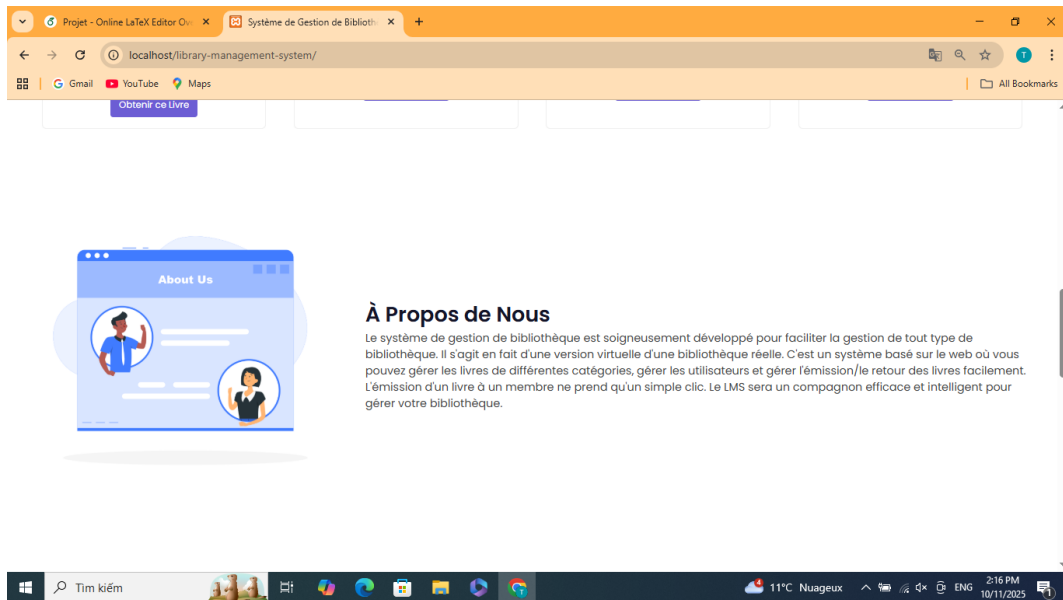


Figure 3: À propos de nous

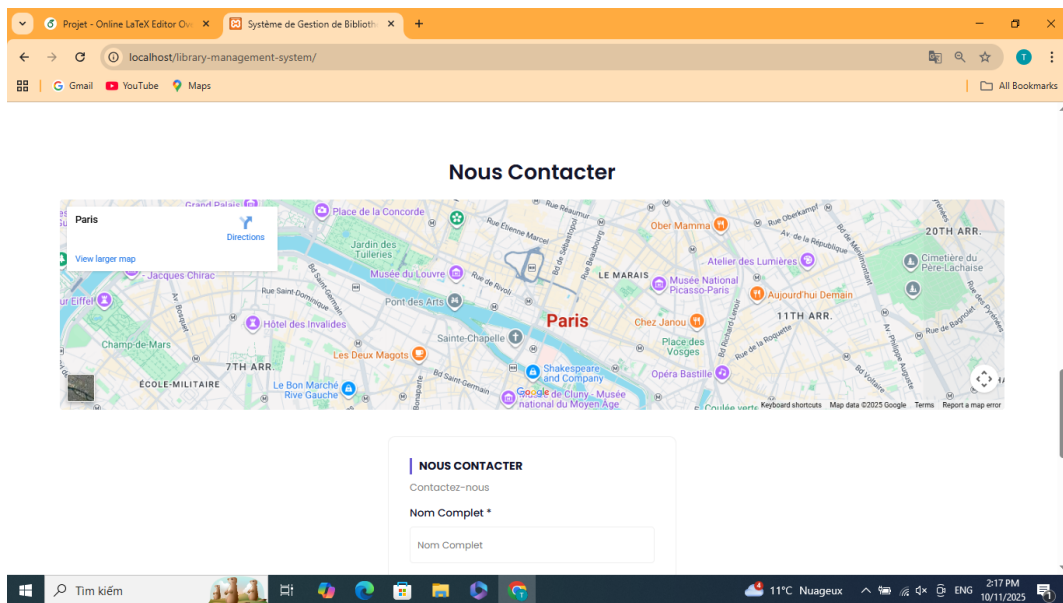


Figure 4: Notre région

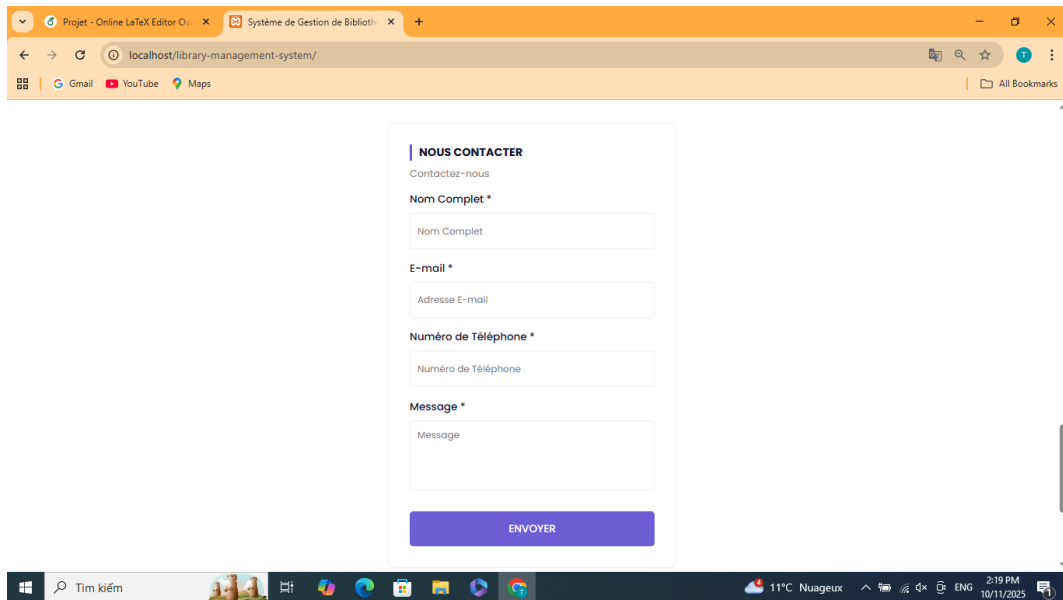


Figure 5: Page de contact

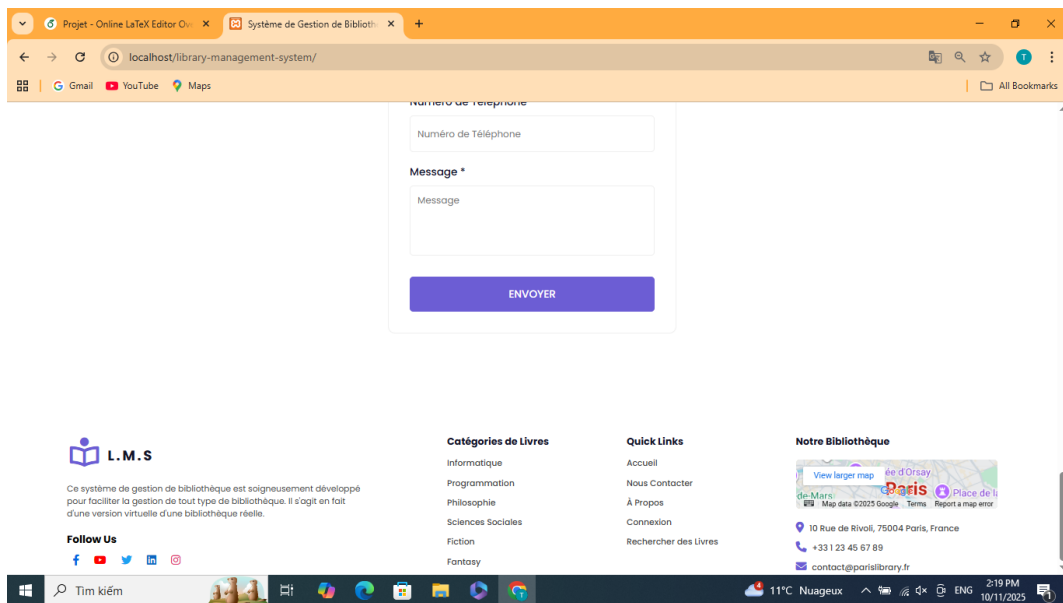


Figure 6: Footer du site

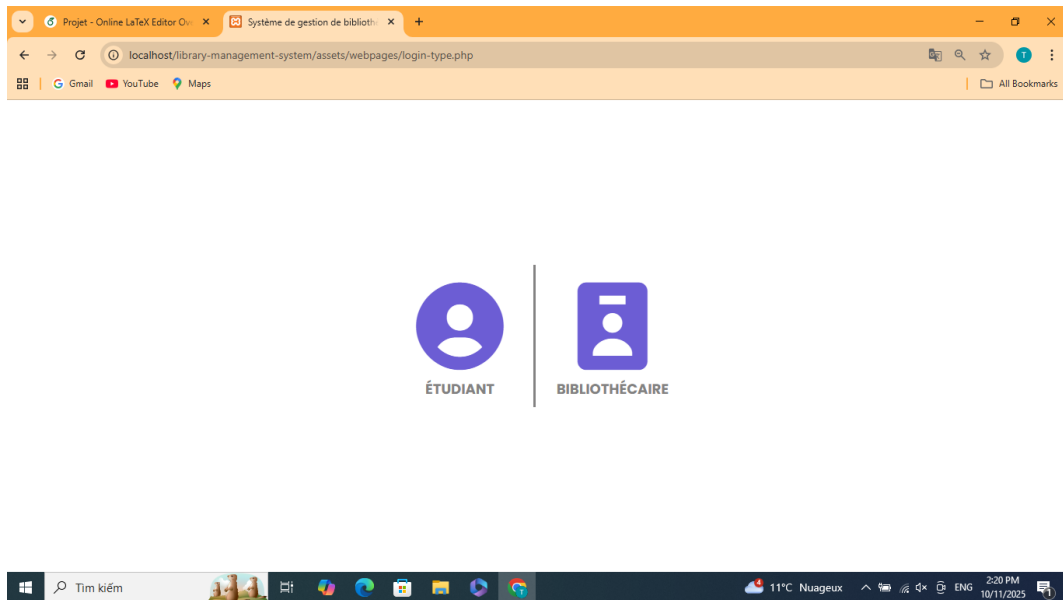


Figure 7: Formulaire de connexion étudiant

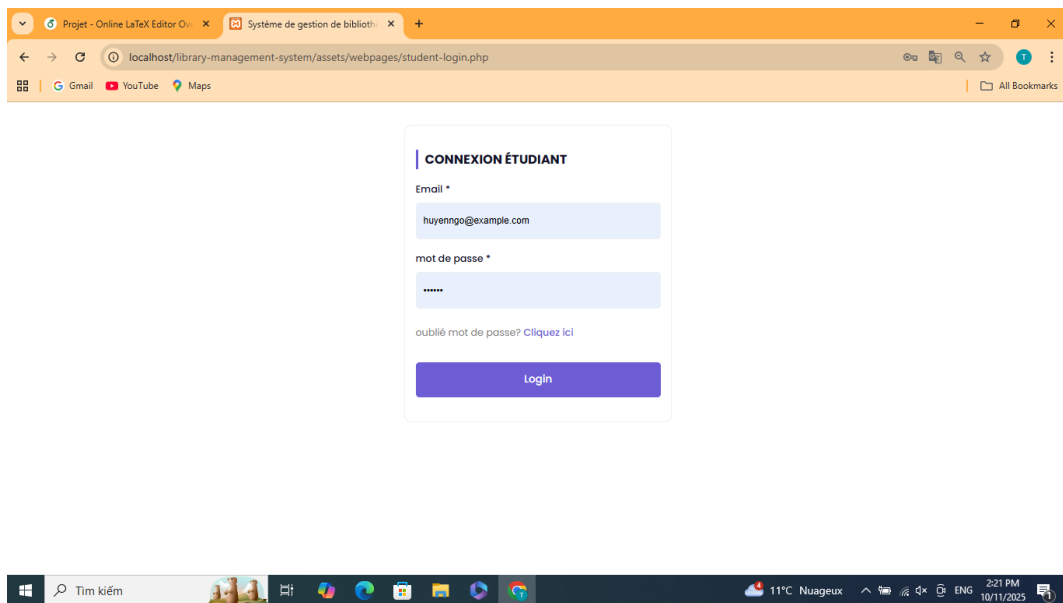


Figure 8: Page de connexion pour étudiant

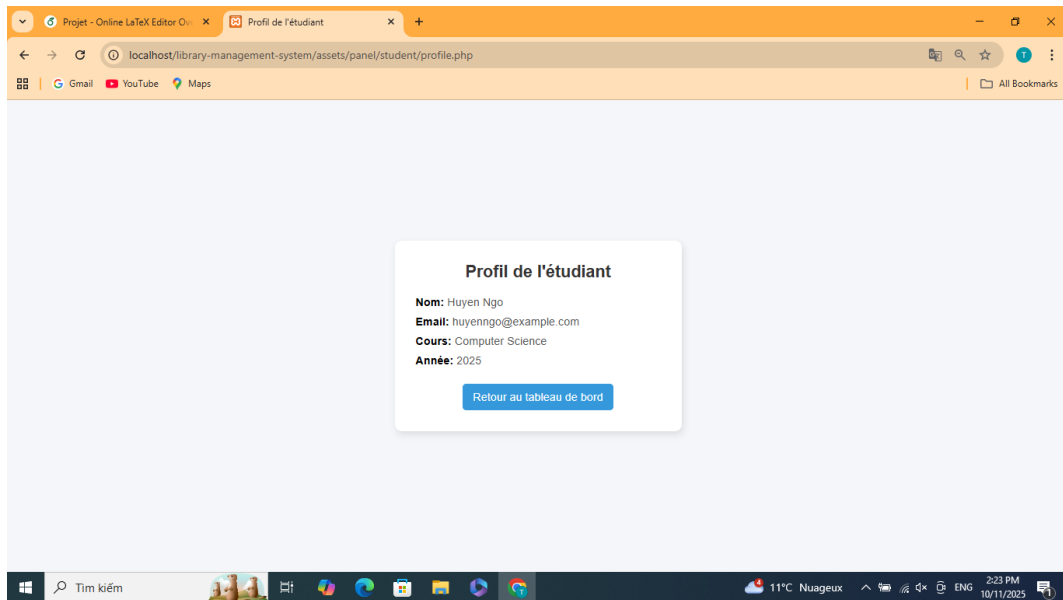


Figure 9: Profil de l'étudiant

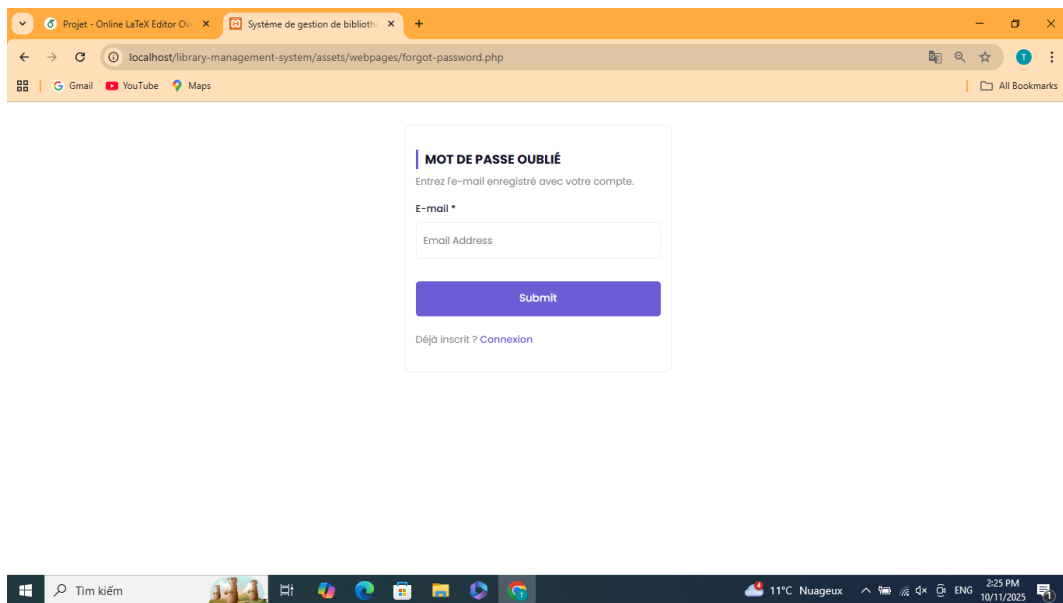


Figure 10: Réinitialiser le mot de passe

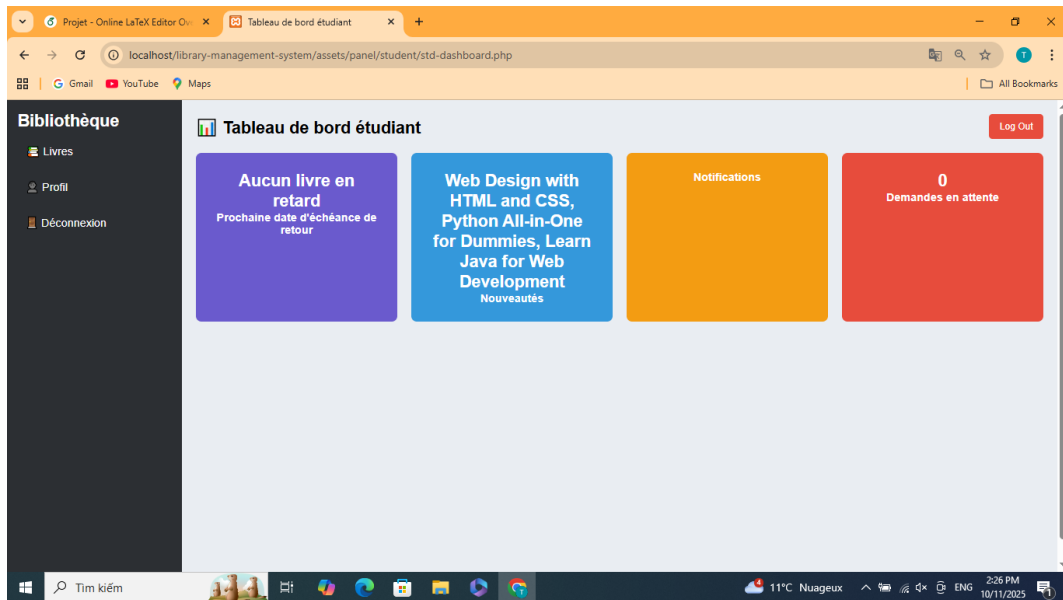


Figure 11: Tableau de bord étudiant

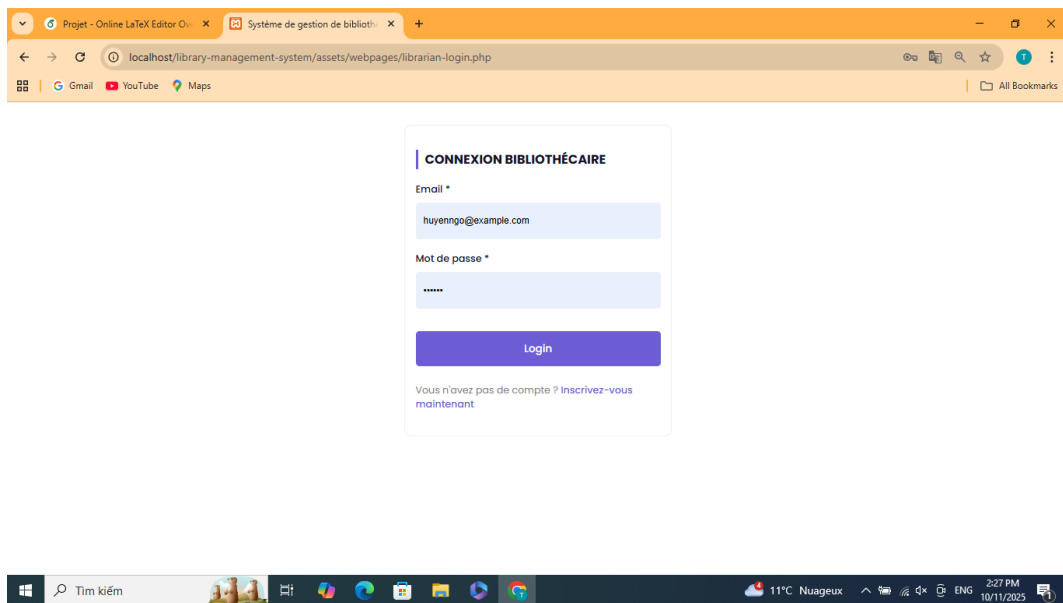


Figure 12: Connexion bibliothécaire

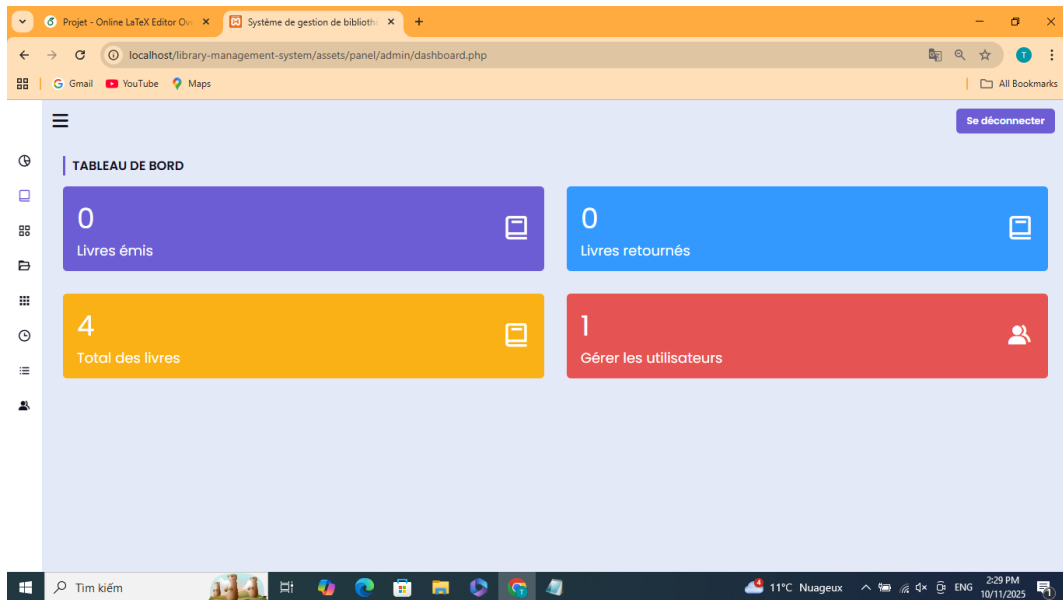


Figure 13: Tableau de bord bibliothécaire

Figure 14: Formulaire d'ajout de livre

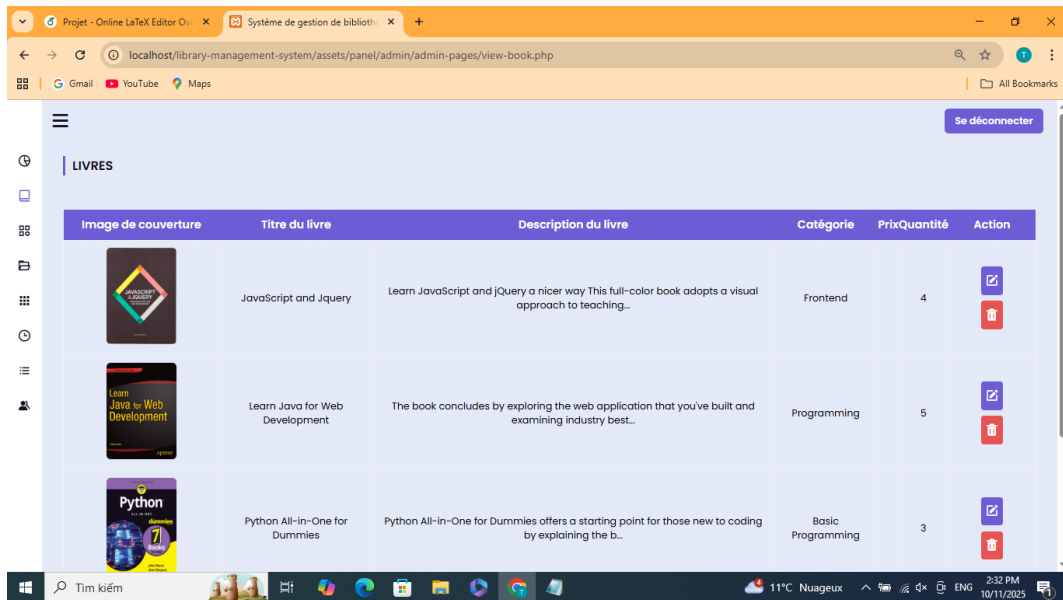


Figure 15: Liste complète des livres

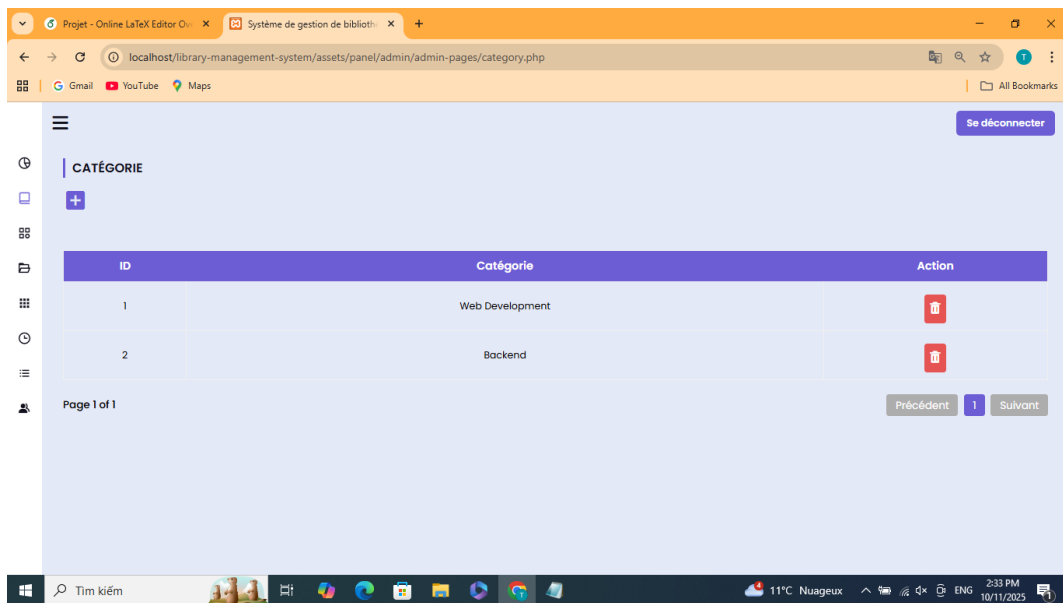


Figure 16: Gestion des catégories

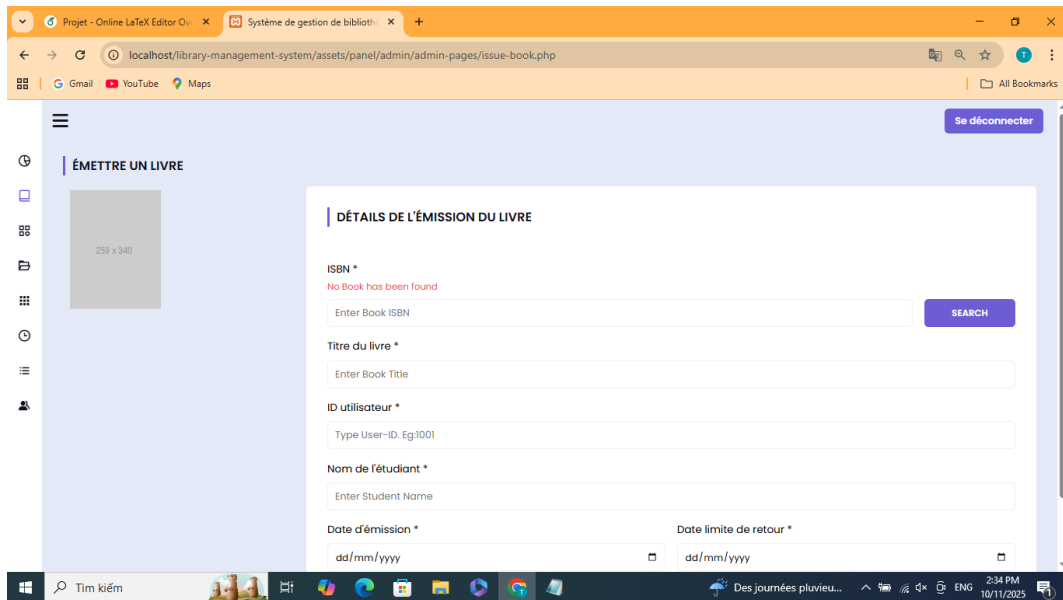


Figure 17: Ajout d'une catégorie

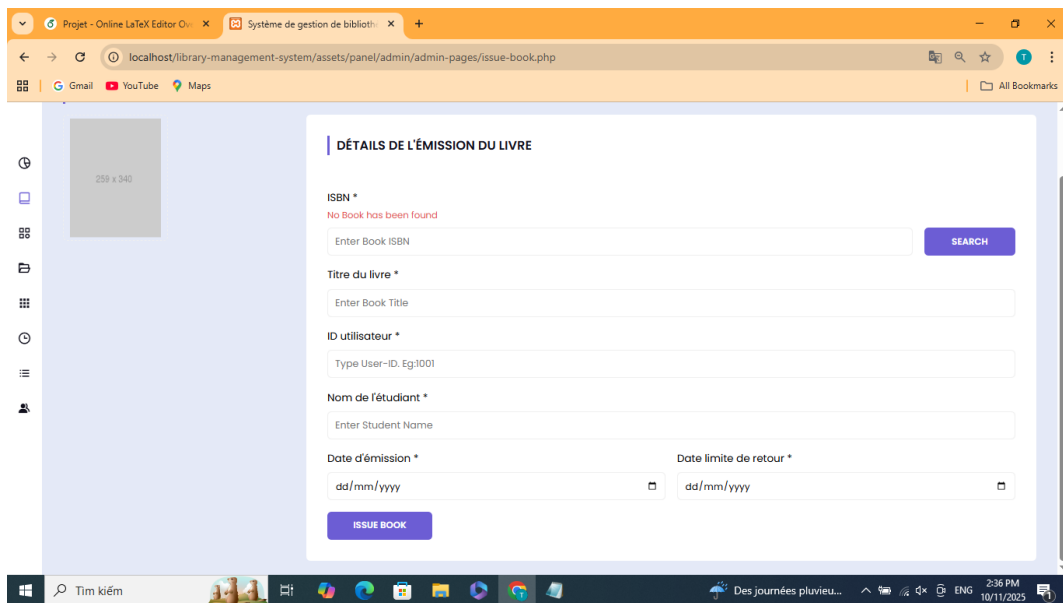


Figure 18: Modification de catégorie

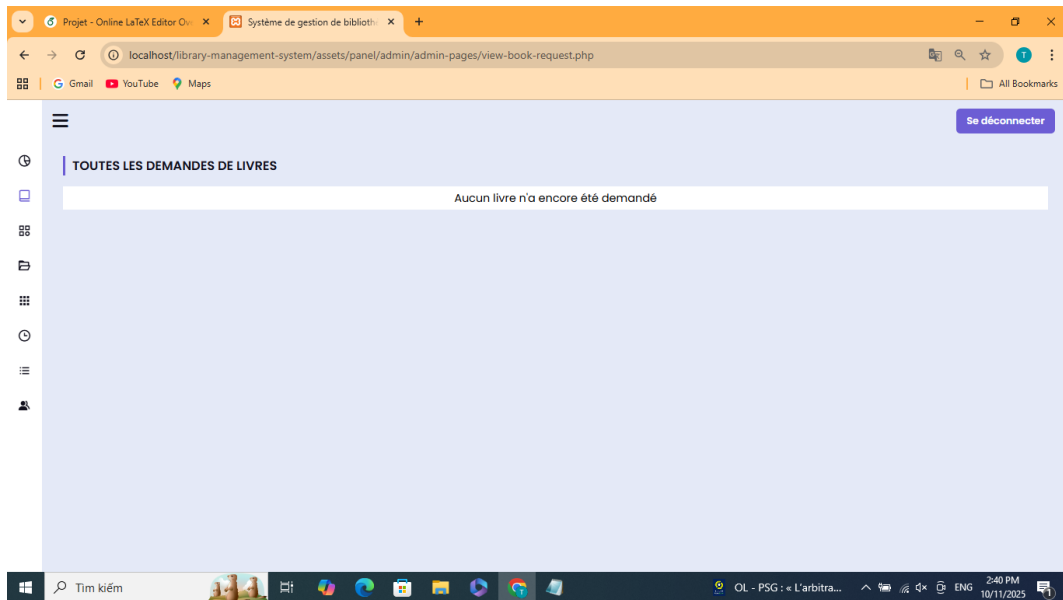


Figure 19: Filtrage par catégorie

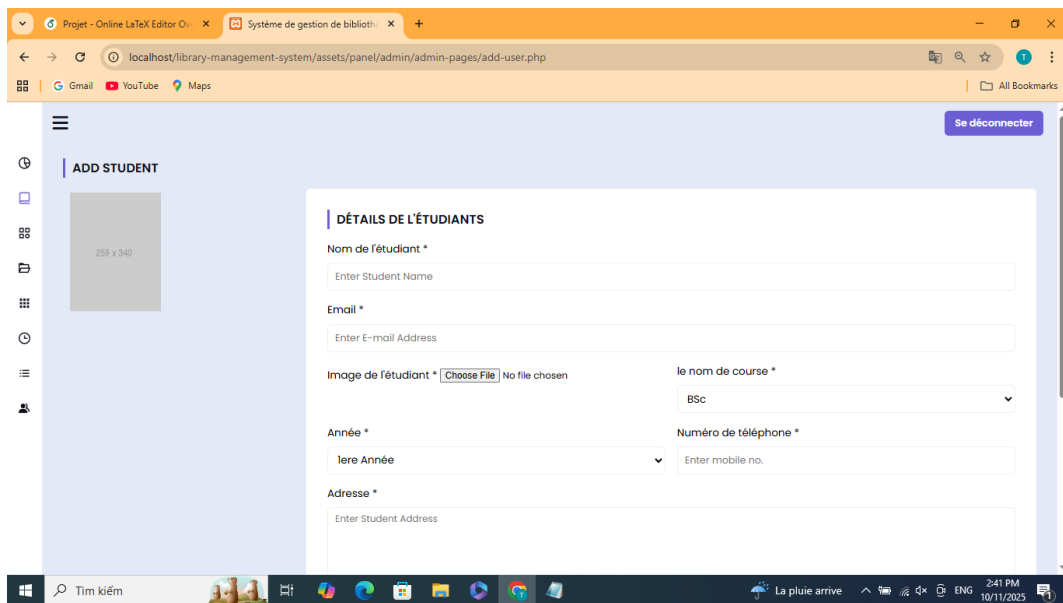


Figure 20: Formulaire d'inscription étudiant

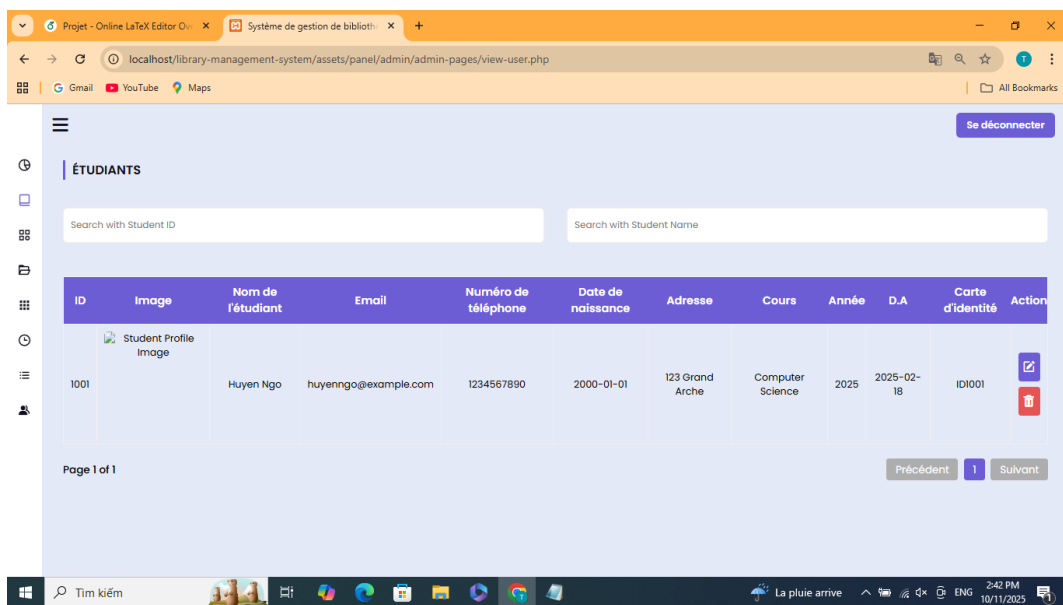


Figure 21: Liste des étudiants

Remerciements et Conclusion

Après des mois de recherche et de développement, nous avons finalisé le projet intitulé « **Création d'un site web de gestion de bibliothèque** », conçu pour répondre aux besoins essentiels d'un système moderne de gestion de bibliothèque. Ce site propose des fonctionnalités pratiques telles que :

- La gestion des livres, des catégories et des utilisateurs ;
- Le suivi des emprunts et des retours ;
- Des outils statistiques facilitant une administration simplifiée et efficace.

Ce projet nous a permis d'appliquer concrètement nos compétences en développement web, en exploitant des technologies telles que **PHP, MySQL, HTML, CSS et JavaScript**. Il a également contribué à renforcer nos capacités d'analyse et de conception de systèmes adaptés aux besoins réels.

Remerciements : Nous tenons à exprimer notre profonde gratitude à **Monsieur CHERGUI Anis**, dont les conseils et l'accompagnement tout au long de ce travail ont été d'une aide précieuse. Ses remarques et suggestions nous ont permis de progresser et d'enrichir nos connaissances.

« Travailler sur ce projet a été une expérience enrichissante et formatrice, que nous espérons poursuivre à travers de nouveaux défis. »