

Rapport d'Application Web: Gestion de Bibliothèque

Hasna Daoui & Yassine Elghazi

January 2025

1 Introduction du Projet : Gestion de Bibliothèque

Dans un monde de plus en plus numérique, les bibliothèques restent des piliers essentiels pour l'accès à la connaissance. Cependant, la gestion efficace des ressources d'une bibliothèque, qu'il s'agisse des livres, des emprunts ou des retours, peut devenir une tâche complexe. Le projet de développement d'une application de gestion web vise à moderniser et à simplifier ces processus en offrant une solution numérique efficace et intuitive.

1.1 Objectif principal

Concevoir et développer une application web qui permet une gestion centralisée et optimisée des opérations courantes d'une bibliothèque, tout en assurant une expérience utilisateur fluide et agréable.

1.2 Description du système

L'application de gestion de bibliothèque sera basée sur les technologies **PHP**, **HTML**, **CSS** et **MySQL**. Elle fournira des outils performants pour :

- **Gérer les livres et utilisateurs** : Ajout, modification, suppression et recherche des ouvrages disponibles.
- **Offrir des statistiques** : Tableaux de bord affichant le nombre total de livres et de membres.
- **Assurer une gestion des utilisateurs** : Mise en place d'un système d'authentification pour les administrateurs et les membres, avec des rôles définis pour chaque type d'utilisateur.

1.3 Pourquoi ce projet ?

La gestion traditionnelle des bibliothèques peut être fastidieuse, nécessitant de multiples registres ou fichiers. Une application web centralisée simplifie ces

tâches en réduisant les erreurs humaines, en accélérant les recherches, et en offrant un meilleur suivi des ressources. Elle répond également aux besoins modernes des bibliothèques en facilitant l'accès aux informations en temps réel et en optimisant la gestion globale.

Ce projet s'inscrit dans une démarche d'innovation et de simplification des pratiques, en permettant aux bibliothèques de se concentrer sur leur mission principale : rendre la lecture et la connaissance accessibles à tous.

2 Analyse des Besoins

2.1 Identification des Utilisateurs

- **Administrateur :**

- Gestionnaire global avec un accès complet
- Peut gérer les utilisateurs (ajout, modification, suppression)
- Peut gérer les livres (ajout, modification, suppression)
- Accède aux tableaux de bord affichant :
 - * Le nombre total de livres disponibles
 - * Le nombre d'utilisateurs enregistrés

- **Membre (Utilisateur) :**

- Accès à la page d'accueil pour consulter les livres disponibles
- Recherche avancée des livres
- Connexion ou inscription requise pour réserver un livre
- Gestion de ses propres réservations (ajout ou suppression)
- Modification de ses informations personnelles

2.2 Besoins Fonctionnels

- **Administrateur :**

- Système d'authentification sécurisé
- Gestion des livres : ajout, modification, suppression
- Gestion des utilisateurs : ajout, modification, suppression
- Accès à des tableaux de bord pour des statistiques sur les livres et les utilisateurs

- **Membre :**

- Consultation et recherche des livres disponibles
- Réservation des livres après authentification
- Gestion des réservations (ajout ou suppression)
- Modification des informations personnelles

2.3 Besoins Non Fonctionnels

- Sécurité :

- Protection des mots de passe avec des algorithmes de hachage sécurisés (MD5)
- Gestion des permissions selon les rôles

- Performance :

- Temps de réponse rapide pour les recherches et les opérations CRUD

- Ergonomie :

- Interface intuitive et facile à naviguer
- Compatibilité avec les navigateurs modernes

3 Conception du Système

3.1 Architecture de l'Application

L'application est basée sur une architecture **client-serveur**, où :

- **Client** : L'interface utilisateur développée en HTML et CSS, avec l'intégration de JavaScript pour les interactions dynamiques
- **Serveur** : Gestion des requêtes et traitement des données via PHP
- **Base de données** : Stockage des données sur MySQL

3.2 Modèle de la Base de Données

3.2.1 Structure des Tables

- Table Users :

```
CREATE TABLE Users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(255) UNIQUE NOT NULL,
    password VARCHAR(255) NOT NULL,
    role ENUM('member', 'administrator') NOT NULL DEFAULT 'member'
);

INSERT INTO users(username, password, role)
VALUES('admin', MD5('password'), 'administrator');
```

- Table Books :

```

CREATE TABLE Books (
    id INT AUTO_INCREMENT PRIMARY KEY,
    title VARCHAR(255) UNIQUE NOT NULL,
    author VARCHAR(255) NOT NULL,
    genre VARCHAR(100),
    publication_date DATE,
    number_of_books INT NOT NULL
);

```

- **Table Reserved_books :**

```

CREATE TABLE Reserved_books (
    id INT AUTO_INCREMENT PRIMARY KEY,
    book_id INT NOT NULL,
    user_id INT NOT NULL,
    final_return_date DATE NOT NULL,
    FOREIGN KEY (book_id) REFERENCES Books(id) ON DELETE CASCADE,
    FOREIGN KEY (user_id) REFERENCES Users(id) ON DELETE CASCADE
);

```

3.3 Schéma Relationnel

- **Users** : Stocke les informations des utilisateurs avec rôles distincts.
- **Books** : Gère les livres avec des détails tels que le titre, l'auteur, le genre et la quantité disponible.
- **Reserved.books** : Relie les utilisateurs aux livres réservés avec une date limite de retour.

4 Fonctionnalités de l'Interface Administrateur

- **Gestion des livres**
 - Ajouter
 - Modifier
 - Supprimer
- **Gestion des utilisateurs**
 - Ajouter
 - Modifier
 - Supprimer
- **Voir des statistiques à propos des livres et des membres**

5 Fonctionnalités de l'Interface Membre

- Gestion des livres
 - Réserver
 - Rechercher
 - Consulter les livres disponibles
- Gestion des données personnelles
 - Modifier les informations personnelles

6 Database Connection (Connexion à la Base de Données)

Le processus de connexion à la base de données dans le système comprend les étapes suivantes :

- Définition des paramètres de connexion à la base de données : hôte, nom de la base de données, nom d'utilisateur et mot de passe.
- Tentative de connexion en utilisant PDO (PHP Data Objects).
- Si la connexion est réussie, l'attribut ATTR_ERRMODE est défini sur PDO::ERRMODE_EXCEPTION pour gérer les erreurs.
- Si la connexion échoue, une exception est lancée et un message d'erreur est affiché.

7 Système de Login et Sign-Up

7.1 Login (Se connecter)

Le processus de connexion dans le système comprend les étapes suivantes :

- Vérification des informations d'identification fournies par l'utilisateur.
- Comparaison des données avec celles enregistrées dans la base de données.
- Connexion réussie avec redirection vers la page principale appropriée (membre ou administrateur).
- Affichage d'un message d'erreur en cas d'informations incorrectes.

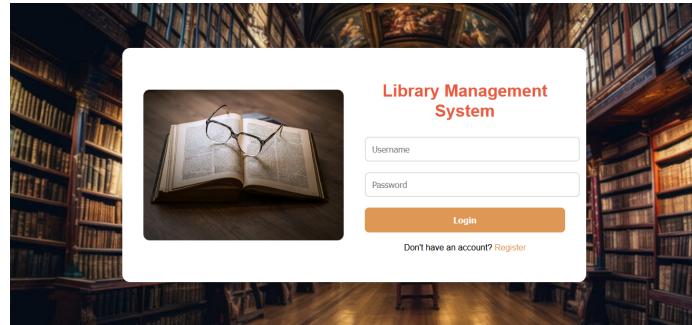


Figure 1: Login interface

7.2 Sign-Up (Créer un compte)

Le processus de création de compte dans le système comprend les étapes suivantes :

- Vérification que tous les champs requis sont remplis.
- Vérification que le mot de passe et sa confirmation correspondent.
- Validation si le nom d'utilisateur est déjà utilisé.
- Enregistrement de l'utilisateur avec un mot de passe sécurisé.
- Redirection vers la page de connexion en cas de succès.

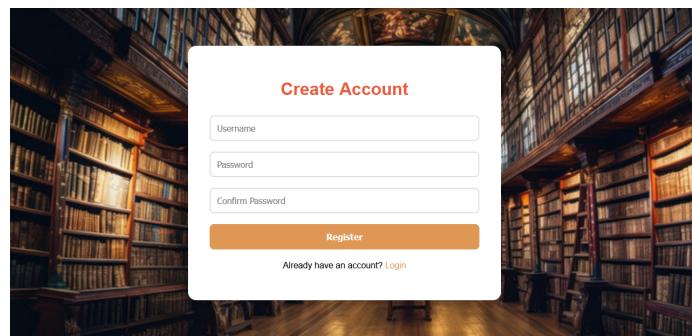


Figure 2: Sign up interface

8 l'Interface Administrateur

8.1 Admin Dashboard (Tableau de Bord Administrateur)

Le processus d'affichage de l'interface administrateur après la connexion comprend les étapes suivantes :

- Vérification de la connexion de l'utilisateur.
- Si l'utilisateur n'est pas connecté ou n'a pas les droits administratifs, redirection vers la page de connexion.
- Affichage du tableau de bord avec les fonctionnalités principales :
 - Gestion des livres : Ajouter, modifier ou supprimer des livres.
 - Gestion des utilisateurs : Gérer les comptes, permissions et rôles.
- Affichage d'un bouton de passer au interface utilisateur.
- Application d'effets visuels pour améliorer l'expérience utilisateur.

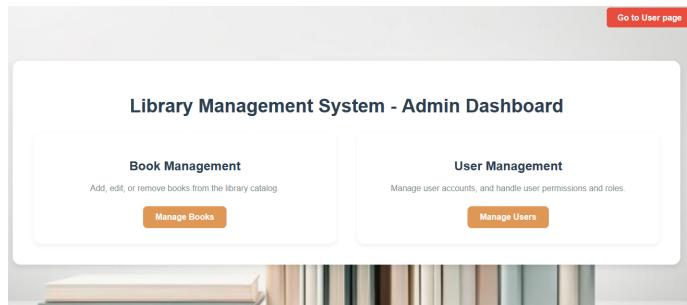


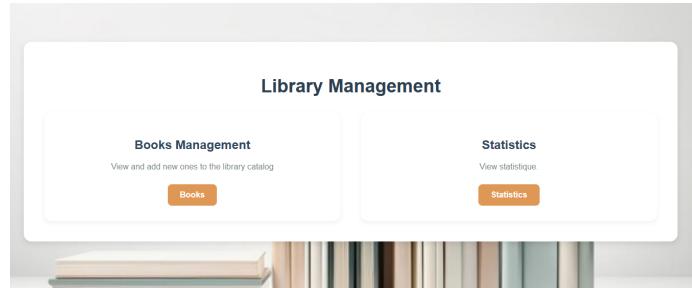
Figure 3: Admin Dashboard

8.2 Book Management Interface (Interface de Gestion des Livres)

Le processus d'affichage de l'interface de gestion des livres après que l'administrateur ait cliqué sur "Manage Books" comprend les étapes suivantes :

- Vérification de la connexion de l'administrateur à la page.
- Si l'administrateur n'est pas connecté, redirection vers la page de connexion.
- Chargement de la page "Books Management" avec les éléments suivants :

- Affichage d'un titre indiquant "Library Management" en haut de la page.
- Affichage de deux sections principales dans la grille :
 - * **Books Management** : Permet d'ajouter et d'afficher les livres dans le catalogue de la bibliothèque.
 - * **Statistics** : Permet à l'administrateur de consulter des statistiques sur les livres.
- Chaque section contient un bouton qui permet à l'administrateur de naviguer vers les pages respectives (affichage des livres et consultation des statistiques).
- Application d'un design visuel soigné avec des effets d'hover pour les éléments interactifs.
- Affichage d'un bouton de déconnexion permettant de quitter la session administrateur.



8.3 barre de navigation

- **Interface de navigation :**
 - Affiche une barre de navigation fixe en haut de la page avec des boutons pour accéder aux pages de gestion des livres (`book_management.php`) et des utilisateurs (`user_dis.php`).
 - Chaque bouton redirige l'utilisateur vers la page correspondante lorsqu'il est cliqué.
- **Gestion de la session :**
 - Un bouton *Go to user Page* permet de rediriger l'utilisateur vers la page d'accueil ou la page utilisateur en cliquant sur le bouton de déconnexion.

- **Design et navigation :**

- Le design est simple et épuré, avec une barre de navigation fixe en haut et une interface de gestion placée au centre.
- Utilisation de boutons stylés pour une expérience utilisateur agréable, avec des effets visuels (changement de couleur au survol).
- La page se met à jour de manière dynamique avec l'ajout ou la suppression de livres et la gestion des utilisateurs.

8.4 Gerer les livres

- **Connexion à la base de données :**

- Inclut un fichier de connexion à la base de données (`db.php`).
- Effectue une requête pour récupérer tous les livres dans la table `Books`, triés par titre.

- **Gestion des livres :**

- Affiche la liste de tous les livres sous forme de tableau avec des informations telles que le titre, l'auteur, le genre, la date de publication et le nombre de copies disponibles.
- Pour chaque livre, des boutons permettent de modifier ou de supprimer le livre.

- **Ajout de nouveaux livres :**

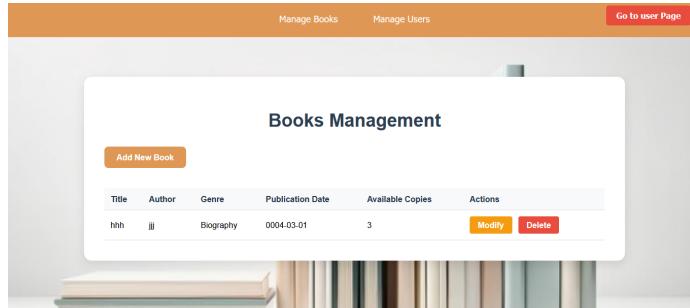
- Un bouton permet d'ajouter un nouveau livre en redirigeant vers une page `addbooks.php`.

- **Messages de succès :**

- En cas d'ajout, modification ou suppression réussie d'un livre, un message de succès est affiché en haut de la page.

- **Effet visuel et design :**

- Utilisation de styles CSS pour une présentation soignée avec des couleurs attrayantes et des effets au survol des éléments (ex. : boutons, lignes du tableau).



8.5 Ajouter un nouveau livre

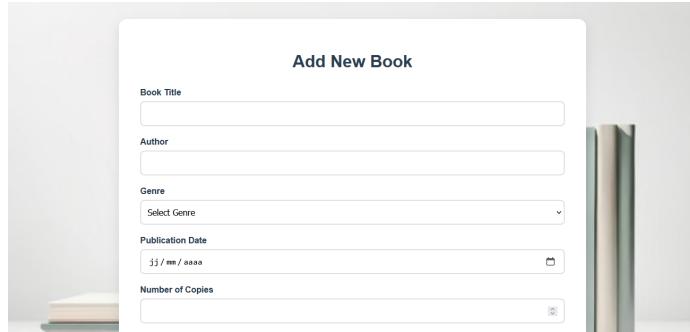
- Vérification de la session utilisateur :
 - Le script commence par vérifier si l'utilisateur est connecté en vérifiant la session (`$_SESSION["logged_in"]`).
 - Si l'utilisateur n'est pas connecté, il est redirigé vers la page `index.php`.
- Connexion à la base de données :
 - Une connexion à la base de données `LibraryManagement` est établie à l'aide de PDO (PHP Data Objects).
 - Les erreurs de connexion sont gérées et un message d'erreur est affiché si la connexion échoue.
- Traitement du formulaire :
 - Lorsque le formulaire est soumis en POST, les données du livre (titre, auteur, genre, date de publication, nombre de copies) sont récupérées.
 - Une requête SQL `INSERT` est préparée pour ajouter ces informations dans la table `Books`.
 - Si l'ajout est réussi, l'utilisateur est redirigé vers `displaybooks.php` avec un message de succès.
- Gestion des erreurs :
 - Si une erreur survient lors de l'ajout du livre, un message d'erreur est affiché à l'utilisateur.
- Structure du formulaire HTML :
 - Le formulaire utilise des champs de saisie pour le titre, l'auteur, le genre, la date de publication et le nombre de copies disponibles.
 - Des éléments `select` et `input` sont utilisés pour capturer les informations.
 - Les champs sont validés en JavaScript et avec l'attribut `required` pour garantir que toutes les informations nécessaires sont fournies.

- **Design et style CSS :**

- Le formulaire est stylisé avec des couleurs de fond douces, des bordures et des ombres pour créer une interface élégante.
- Les boutons et champs de saisie utilisent des transitions pour des effets visuels agréables.

- **Navigation et retour :**

- Un lien `Back to Books List` permet à l'utilisateur de revenir à la liste des livres.



8.6 Modifier les livres

- **Connexion à la base de données :**

- Inclut un fichier de connexion à la base de données (`db.php`).
- Effectue une requête pour récupérer les détails d'un livre spécifique à partir de la table `Books`, en utilisant un identifiant (`id`).

- **Modifier des livres :**

- Affiche un formulaire permettant de modifier les informations du livre (titre, auteur, genre, date de publication et nombre de copies disponibles).
- Si la mise à jour est réussie, l'utilisateur est redirigé vers la page `displaybooks.php` avec un message de succès.

- **Vérification et validation des données :**

- Les champs sont vérifiés pour s'assurer qu'aucun champ obligatoire (titre, auteur, genre, etc.) n'est laissé vide.
- Si des erreurs sont détectées, un message d'erreur est affiché pour informer l'utilisateur.

- **Gestion des erreurs :**

- Si des erreurs surviennent lors de l'accès à la base de données ou lors de la mise à jour, un message d'erreur est affiché à l'utilisateur.

- **Effet visuel et design :**

- Le formulaire et la page utilisent des styles CSS pour rendre l'interface attrayante, avec des boutons colorés et une présentation soignée.
- Le formulaire est conçu pour être convivial, avec des champs de saisie facilement identifiables.

The screenshot shows a 'Modify Book' form. The form has a title 'Modify Book' at the top. It contains several input fields: 'Book Title' with the value 'hhh', 'Author' with the value 'jjj', 'Genre' with the value 'Biography', 'Publication Date' with the value '01 / 03 / 0004', and 'Number of Copies' with the value '3'. Below the form is a small thumbnail image of a book. At the bottom right of the form is a red 'Update Book' button.

9 Suppression d'un livre

- **Vérification de l'ID du livre :**

- Le script vérifie si un identifiant de livre (`delete_id`) est passé dans l'URL via la méthode GET.

- **Vérification de l'existence du livre :**

- Une requête SQL SELECT est préparée pour vérifier si un livre avec l'ID spécifié existe dans la table `books`.
- Si aucun livre n'est trouvé avec l'ID donné, un message d'erreur est affiché.

- **Suppression du livre :**

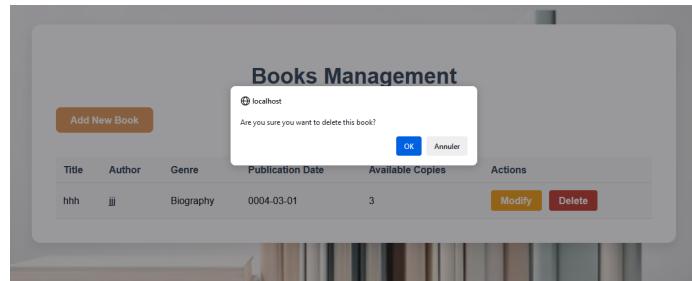
- Si le livre existe, une requête SQL DELETE est préparée pour supprimer le livre de la base de données.
- L'exécution de la requête de suppression est suivie par un contrôle pour s'assurer que l'opération a été réussie.

- **Redirection après suppression :**

- Si la suppression est réussie, l'utilisateur est redirigé vers la page `displaybooks.php` avec un paramètre `status=deleted` dans l'URL pour indiquer que le livre a été supprimé.

- **Gestion des erreurs :**

- Si une erreur survient lors de la suppression, un message d'erreur est affiché.
- Si l'ID du livre n'est pas valide ou s'il ne correspond à aucun livre existant, un message d'erreur est également affiché.



9.1 Affichage du nombre d'utilisateurs et de livres

- **Connexion à la base de données :**

- Le script utilise PDO pour se connecter à la base de données `LibraryManagement` avec les paramètres de connexion : `localhost`, `root` comme utilisateur, et une mot de passe vide.
- Si la connexion échoue, une exception est levée et un message d'erreur est affiché.

- **Requêtes SQL :**

- `userQuery` récupère le nombre total d'utilisateurs dans la table `Users` via `SELECT COUNT(*)`.
- `bookQuery` calcule le nombre total de livres disponibles en additionnant la colonne `number_of_books` de la table `Books`.

- **Exécution des requêtes :**

- Les requêtes sont préparées et exécutées à l'aide de la méthode `execute()` sur les objets `userStmt` et `bookStmt`.
- Les résultats sont récupérés à l'aide de `fetch(PDO::FETCH_ASSOC)` pour obtenir respectivement le nombre d'utilisateurs et le total des livres.

- **Affichage du graphique :**

- Un graphique à barres est généré à l'aide de la bibliothèque `Chart.js`.
- Le graphique affiche deux barres : l'une pour le nombre d'utilisateurs et l'autre pour le nombre total de livres.
- Les valeurs des barres sont remplies par les variables PHP `$userCount` et `$totalBooks` passées à JavaScript.
- Les couleurs des barres sont définies avec `backgroundColor` et `borderColor`.

- **Style et présentation :**

- Le design de la page utilise un fond clair avec une section de contenu centrée et une boîte ombrée pour le graphique.
- Le titre est centré et de couleur orange pour correspondre à la palette de couleurs du projet.
- Le graphique est responsive et s'adapte à différentes tailles d'écran.



9.2 Affichage de liste de utilisateurs

The screenshot shows a "Users Management" interface. At the top, there are navigation links: "Manage Books", "Manage Users", and "Go to user Page". Below this is a table with the following data:

Name	Password (Hidden)	Role	Actions
admin	21232f297a57a5a743894a0e4a801fc3	administrator	Modify Delete
hello	5d41402abc4b2a79b9719d911017c592	member	Modify Delete

An "Add New User" button is located at the bottom left of the table area. The background features a decorative pattern of colored squares.

9.3 Ajout d'un utilisateur

- Lorsqu'un formulaire est soumis en méthode POST avec le bouton `add_user`, le système récupère les données du nom d'utilisateur, du mot de passe (crypté en MD5 pour la sécurité), et du rôle (utilisateur ou administrateur).
- Une requête `INSERT` est exécutée pour insérer ces données dans la table `Users`.
- Si l'ajout réussit, l'utilisateur est redirigé vers la page de gestion des utilisateurs avec un paramètre de succès. Sinon, un message d'erreur est affiché.

- **Structure du formulaire d'ajout :**

- Le formulaire d'ajout comporte trois champs :
 - * `username` : Champ pour le nom d'utilisateur.
 - * `password` : Champ pour le mot de passe, qui est crypté avant l'insertion.
 - * `role` : Liste déroulante permettant de sélectionner le rôle de l'utilisateur (membre ou administrateur).
- Les données sont envoyées au script PHP pour traitement et insertion dans la base de données.

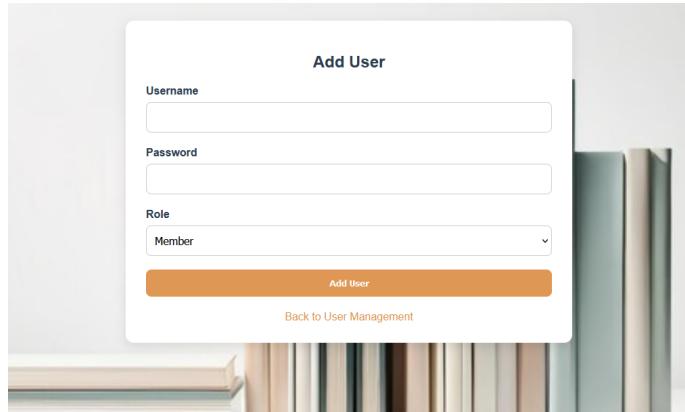
- **Interface utilisateur :**

- L'interface utilise un design moderne avec des champs de saisie élégants et un bouton de soumission en couleur orange pour attirer l'attention.
- Un message d'erreur s'affiche si l'ajout échoue, et un lien permet de revenir à la page de gestion des utilisateurs.
- Le formulaire est également conçu pour être réactif et s'adapte à différentes tailles d'écran.

9.4 Suppression d'un utilisateur

- Lorsqu'une requête GET est effectuée avec l'ID de l'utilisateur à supprimer, le système vérifie si l'ID est valide (c'est-à-dire un entier supérieur à zéro).
- Si l'ID est valide, une requête `DELETE` est exécutée sur la table `Users` pour supprimer l'utilisateur correspondant.
- En cas de succès, l'utilisateur est redirigé vers la page de gestion des utilisateurs avec un message de succès. Sinon, un message d'erreur est affiché.

- **Sécurité :**



- L'ID de l'utilisateur est converti en entier et vérifié pour éviter toute injection SQL.
- La requête `DELETE` utilise des paramètres liés de manière sécurisée pour garantir l'intégrité des données.

- **Gestion des erreurs :**

- Si l'ID est manquant ou invalide, l'utilisateur est redirigé vers la page de gestion des utilisateurs avec un message d'erreur.
- En cas d'échec de la suppression, un message d'erreur est affiché à l'utilisateur.

9.5 Modification d'un utilisateur

- **Processus de modification :**

- Lorsqu'un utilisateur est sélectionné à partir de la liste des utilisateurs (via l'ID), les informations de cet utilisateur sont récupérées depuis la base de données.
- Un formulaire est présenté avec les champs pré-remplis : nom d'utilisateur, mot de passe et rôle (membre ou administrateur).
- Lorsqu'un utilisateur soumet le formulaire de modification, les nouvelles informations sont envoyées à la base de données pour être mises à jour.

- **Sécurité :**

- Le mot de passe est haché à l'aide de la fonction MD5 avant d'être enregistré, garantissant une protection de base des informations sensibles.
- L'ID de l'utilisateur est récupéré via une requête sécurisée et utilisé pour identifier l'utilisateur à modifier.

- **Gestion des erreurs :**

- En cas d'erreur lors de la récupération des informations ou de la mise à jour de l'utilisateur, un message d'erreur est affiché.
- Si l'ID de l'utilisateur est invalide ou non fourni, l'utilisateur est redirigé vers la liste des utilisateurs.

The screenshot shows a 'Modify User' form within a web application. The top navigation bar includes 'Manage Books', 'Manage Users', and a 'Go to user Page' button. The main form has fields for 'Username' (admin), 'Password' (empty), and 'Role' (Administrator). It features an 'Update user' button and a 'Back to Users List' link at the bottom. The background of the page features a stack of books.

10 Interface Membre (Member Interface)

10.1 États de l'Interface (Interface States)

L'interface membre présente deux états distincts basés sur l'authentification de l'utilisateur :

- **État Non Authentifié :**

- Affichage d'une barre de navigation avec options de connexion et d'inscription
- Possibilité de consulter la liste des livres disponibles
- Fonctionnalités de recherche de livres par titre, auteur, genre, ou année
- Impossibilité de réserver des livres

- **État Authentifié :**

- Accès à toutes les fonctionnalités de l'état non authentifié
- Possibilité de réserver des livres disponibles
- Accès à "Mes Réservations" pour voir les livres réservés
- Option de modification du profil (nom d'utilisateur et mot de passe)
- Bouton de déconnexion

Figure 4: Interface utilisateur - État non authentifié

Figure 5: Interface utilisateur - État authentifié

10.2 Fonctionnalités Principales

10.2.1 Recherche de Livres

Le système offre plusieurs options de recherche :

- Recherche par titre
- Recherche par auteur
- Recherche par genre

- Recherche par année de publication

10.2.2 Système de Réservation

Pour les utilisateurs authentifiés, le processus de réservation comprend :

- Vérification de la disponibilité du livre
- Vérification que l'utilisateur n'a pas déjà réservé ce livre
- Attribution automatique d'une date de retour (14 jours après la réservation)
- Mise à jour du nombre de copies disponibles

10.2.3 Gestion du Profil

Les utilisateurs connectés peuvent :

- Modifier leur nom d'utilisateur
- Changer leur mot de passe
- Consulter leurs réservations actuelles

10.3 Sécurité et Gestion des Sessions

Le système implémente plusieurs mesures de sécurité :

- Utilisation de sessions PHP pour maintenir l'état de connexion
- Protection contre les injections SQL via l'utilisation de requêtes préparées
- Hachage des mots de passe avec MD5
- Validation des données utilisateur côté serveur

10.4 Interface Responsive

L'interface utilisateur est conçue pour être responsive avec :

- Une mise en page fluide s'adaptant à différentes tailles d'écran
- Des contrôles de navigation clairement disposés
- Des messages de feedback pour les actions utilisateur
- Un design moderne utilisant des ombrages et des transitions

11 Conclusion

L'interface membre du système de gestion de bibliothèque offre une expérience utilisateur intuitive avec des fonctionnalités adaptées aux besoins des utilisateurs authentifiés et non authentifiés. Le système maintient un équilibre entre facilité d'utilisation et sécurité, tout en offrant les fonctionnalités essentielles de recherche et de réservation de livres.