

Rapport du projet : Gestion des bibliothèques

Jawhara khachachab et Asma Daoui

Web Programming, Janvier 2025

Introduction

Contexte du projet

Ce projet a pour objectif de développer un système de gestion pour une bibliothèque permettant de gérer les livres, les emprunts et les retours. Il vise à faciliter l'administration des bibliothèques tout en offrant une interface intuitive pour les utilisateurs.

Objectifs du projet

Fournir une application web qui permet aux utilisateurs d'emprunter et de retourner des livres, ainsi que d'ajouter de nouveaux livres à la base de données.

Technologies utilisées

PHP pour le backend (avec PDO ou MySQLi), MySQL pour la base de données, HTML et CSS pour le frontend.

Organisation du rapport

Ce rapport détaille les différentes étapes du développement, les problèmes rencontrés et les solutions mises en place.

1 Étapes de Développement

1.1 Analyse des besoins

- Identification des fonctionnalités principales :
 - Gestion des livres (ajout, modification, suppression).
 - Gestion des emprunts (suivi des emprunts et retours).
 - Gestion des utilisateurs (administrateurs et clients).
 - Génération de statistiques et rapports.

1.2 Création de la base de données

Dans cette étape, nous avons créé la base de données et les tables nécessaires pour gérer les utilisateurs, les catégories de livres, les emprunts et les livres eux-mêmes. La base de données utilisée est MySQL et nous avons structuré les tables de la manière suivante :

- **Table des utilisateurs :** Cette table contient les informations des utilisateurs du système, incluant leur identifiant, nom d'utilisateur, mot de passe, et rôle (administrateur ou client). Le code SQL pour la création de cette table est le suivant :

```
CREATE TABLE users (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    username VARCHAR(50) NOT NULL UNIQUE,  
    password VARCHAR(255) NOT NULL,  
    role ENUM('admin', 'client') NOT NULL  
);
```

Ensuite, nous avons inséré des utilisateurs de test (un administrateur et un client) dans la table :

```
INSERT INTO users (username, password, role) VALUES  
( 'admin', '123456', 'admin'), -- Admin avec mot de passe '123456'  
( 'client', '12345', 'client'); -- Client avec mot de passe '12345'
```

- **Table des catégories :** Cette table permet de classer les livres par catégorie. Chaque catégorie a un identifiant unique et un nom. Voici la requête SQL pour créer cette table :

```
CREATE TABLE categories (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nom VARCHAR(100) NOT NULL UNIQUE  
);
```

- **Table des livres :** Cette table stocke les informations des livres, y compris leur titre, auteur, ISBN, et la catégorie à laquelle ils appartiennent. Chaque livre est lié à une catégorie via une clé étrangère. La création de cette table se fait avec la requête suivante :

```
CREATE TABLE livres (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    titre VARCHAR(255) NOT NULL,  
    auteur VARCHAR(255) NOT NULL,
```

```

        isbn VARCHAR(20) NOT NULL,
        categorie_id INT NOT NULL,
        FOREIGN KEY (categorie_id) REFERENCES categories(id) ON DELETE CASCADE
    );

```

- **Table des emprunts :** Cette table gère les emprunts de livres. Chaque emprunt est associé à un utilisateur et un livre, et contient les dates de l'emprunt et du retour. La relation entre les emprunts et les utilisateurs, ainsi que les livres, est définie par des clés étrangères. Voici la requête SQL pour la création de cette table :

```

CREATE TABLE loans (
    id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    livres_id INT NOT NULL,
    borrow_date DATE NOT NULL,
    return_date DATE DEFAULT NULL,
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE,
    FOREIGN KEY (livres_id) REFERENCES livres(id) ON DELETE CASCADE
);

```

Une modification a été apportée à la table des emprunts pour ajouter une colonne `loan_date` afin de stocker la date et l'heure de l'emprunt. La requête suivante ajoute cette colonne :

```

ALTER TABLE loans ADD COLUMN loan_date DATETIME DEFAULT CURRENT_TIMESTAMP;

```

Ces tables sont essentielles pour gérer les données relatives aux livres, aux utilisateurs, et aux emprunts dans le système de gestion de bibliothèque. Chaque table est reliée par des clés étrangères, ce qui permet d'assurer l'intégrité des données et de maintenir une structure cohérente dans la base de données.

1.3 Développement de l'application web (PHP, HTML, CSS)

Une fois la base de données créée, nous avons entamé le développement de l'application web, en utilisant PHP pour le backend, HTML et CSS pour le frontend, afin de fournir une interface utilisateur simple et fonctionnelle pour gérer les livres, les emprunts et les retours.

1.3.1 Page de connexion (Login) et gestion des utilisateurs

Pour la gestion de l'accès à l'application, nous avons créé une page de connexion distincte pour les utilisateurs. Cette page permet aux utilisateurs de se

connecter en tant que **client** ou **administrateur**. Les interfaces et les fonctionnalités varient en fonction du rôle de l'utilisateur.

- **Page de connexion** : La page de connexion comporte deux boutons distincts, un pour le **client** et un pour l'**administrateur**. Chaque bouton redirige l'utilisateur vers une interface différente en fonction de son rôle. Les utilisateurs peuvent se connecter avec leurs identifiants respectifs, et chaque rôle dispose de ses propres fonctionnalités.
- **Interface client** : Lorsque le client clique sur le bouton correspondant, il est redirigé vers une interface dédiée, où il peut voir les livres disponibles, effectuer des emprunts, et retourner des livres. L'interface du client est plus simple et offre uniquement des fonctionnalités liées à la gestion des emprunts. Par exemple, un client peut consulter la liste des livres et emprunter un livre. Le code PHP associé à la validation du login du client est le suivant :

```
if($user['role'] === 'client' && isset($_POST["client"])) {  
    header("location:dashboardclien.php");  
    exit();  
}
```

Cette redirection permet au client d'accéder à ses fonctionnalités spécifiques après une connexion réussie.

- **Interface administrateur** : En revanche, lorsque l'administrateur clique sur son bouton, il est redirigé vers une interface différente, offrant des fonctionnalités supplémentaires, telles que l'ajout de nouveaux livres, supprimer et modifier des livres, et la visualisation des emprunts en cours. l'administrateur après une connexion réussie :

```
if ($user['role'] === 'admin' && isset($_POST["valider"])) {  
    header("location:dashboardadmin.php");  
    exit();  
}
```

Cette interface permet à l'administrateur d'effectuer des actions de gestion de la bibliothèque.

1.3.2 Les fonctionnalités de l'administrateur

- **Ajouter des livres** : L'administrateur a la possibilité d'ajouter de nouveaux livres à la bibliothèque. Pour chaque livre, l'administrateur peut spécifier les informations suivantes : titre, auteur, catégorie et ISBN. Ces données sont ensuite enregistrées dans la base de données, permettant ainsi une gestion complète des livres disponibles dans la bibliothèque.

- **Modifier des livres :** L'administrateur peut modifier les informations relatives à un livre existant. Cela inclut la possibilité de mettre à jour le titre, l'auteur, la catégorie ou l'ISBN. Cette fonctionnalité permet de corriger ou de mettre à jour les informations d'un livre si nécessaire. Par exemple, l'administrateur peut modifier un titre ou changer la catégorie d'un livre pour refléter des informations plus précises ou actuelles.
- **Supprimer des livres :** L'administrateur peut également supprimer des livres de la base de données. Cette action est irréversible et supprime définitivement un livre, ainsi que toutes les informations associées, telles que les emprunts en cours pour ce livre. La suppression permet de maintenir une gestion propre et actualisée de la bibliothèque.
- **Ajouter une catégorie :** L'administrateur peut ajouter de nouvelles catégories pour organiser les livres de la bibliothèque. Pour chaque catégorie, l'administrateur spécifie un nom unique. Une fois ajoutée, cette catégorie devient disponible pour l'affectation aux livres. Cette fonctionnalité permet de mieux structurer les livres dans la bibliothèque et d'améliorer leur classification.
- **Consulter la liste des emprunts:**L'administrateur peut consulter une liste détaillée des livres empruntés par les clients. Cette vue inclut :
 - Le titre du livre emprunté,
 - Le nom du client (ou son identifiant),
 - La date d'emprunt,
 - L'état de retour (indiquant si le livre a été rendu ou est encore en cours d'emprunt). Cette fonctionnalité offre une meilleure traçabilité des emprunts, permettant à l'administrateur de surveiller les retards ou de relancer les clients si nécessaire.
- **Voir les statistiques des emprunts par catégorie :** L'administrateur peut consulter des statistiques sur les emprunts de livres en fonction de leurs catégories. Ces statistiques permettent d'afficher des informations telles que le nombre de livres empruntés dans chaque catégorie, la fréquence des emprunts et les livres les plus populaires. Cette fonctionnalité offre une vue d'ensemble des tendances d'emprunt, ce qui permet à l'administrateur de mieux gérer l'inventaire des livres et de prendre des décisions éclairées concernant l'acquisition de nouveaux livres.

1.3.3 Les fonctionnalités de client

- **Afficher les livres :** Le client peut consulter la liste complète des livres disponibles dans la bibliothèque. Cette fonctionnalité lui permet de voir tous les livres, leurs titres, auteurs et catégories, facilitant ainsi la recherche d'un livre à emprunter.

- **Rechercher un livre par nom, auteur, ou ISBN :** Le client peut rechercher un livre spécifique en utilisant différents critères, tels que le nom du livre, le nom de l’auteur ou le numéro ISBN. Cela permet au client de trouver rapidement un livre particulier parmi le catalogue de la bibliothèque. Une fois qu’un livre est trouvé, le client peut procéder à le emprunt.
- **Emprunter des livres:** Le client a la possibilité d’emprunter des livres disponibles en suivant ces étapes :
 - Sélection du livre souhaité : Le client peut choisir un livre parmi ceux marqués comme disponibles.
 - Confirmation de l’emprunt : Une fois validé, l’emprunt est enregistré dans le système avec les détails suivants : - Nom du client (ou identifiant), - Date de l’emprunt, - Échéance de retour (calculée selon la politique de la bibliothèque).
- **Retourner des livres:** Le client peut retourner un livre emprunté via l’interface dédiée :
 - Sélection du livre à retourner : Le client voit une liste des livres qu’il a empruntés et qui sont en cours.
 - Confirmation du retour : Après validation, le livre est enregistré comme retourné dans le système, et il redevient disponible pour les autres utilisateurs.
- **Afficher les livres réservés :** Le client peut consulter les livres qu’il a déjà réservés ou empruntés. Cette fonctionnalité lui permet de suivre l’état de ses emprunts et de voir les livres qu’il doit retourner ou qui sont encore réservés pour lui.

2 Difficultés Rencontrées et Solutions Apportées

2.1 Lien incorrect lors de l’emprunt des livres

- **Problème:**

Lorsque le client cliquait sur ”Emprunter”, le lien redirigeait vers la liste des livres de l’interface administrateur au lieu de l’interface client.

- **Solution:**

- Correction des routes et séparation claire des espaces administrateur et client.
- Ajout d’une logique pour vérifier le type d’utilisateur connecté (admin ou client).
- Modification des liens pour s’assurer qu’ils redirigent vers les pages appropriées selon les rôles des utilisateurs.

2.2 Gestion des permissions pour les administrateurs et clients

- **Problème :**

Il était nécessaire de limiter l'accès aux fonctionnalités en fonction des rôles des utilisateurs.

-Les administrateurs devaient avoir accès à des fonctionnalités avancées (ajout, suppression, statistiques).

-Les clients ne pouvaient accéder qu'aux fonctionnalités liées à la recherche, l'emprunt et le retour des livres.

- **Solution :**

-Mise en place d'un système de gestion des rôles et permissions basé sur des vérifications avant chaque action.

- EXEMPLE :**

```
// Vérifier si l'utilisateur est connecté
if (!isset($_SESSION['user_id']))
    echo "Erreur: Vous devez être connecté pour afficher vos livres réservés.";
exit();
```

3 Conclusion

Ce projet a permis de développer un système de gestion de bibliothèque efficace et sécurisé, répondant aux besoins des administrateurs et des clients. Malgré des défis tels que la gestion des permissions et des redirections, des solutions ont été mises en place pour offrir une interface intuitive et des fonctionnalités robustes : gestion des livres, emprunts, et statistiques.