

Projet Web



Création d'un site web de référencement, d'emprunt et
de réservation de livres

CHARLES Tanguy
GAUTHIER Louis
FLEURANCE Paul
FOURCADE Louis

Pour le 14 mai 2019

Table des matières

1	Introduction	3
2	Mise en place de la base de données	4
3	Représentation de la base de données	6
4	Mise en place du modèle PHP	7
4.1	Contrôle de la base de données	7
4.2	Gestion des sessions	7
4.3	Validation des informations envoyées par les utilisateurs	7
4.4	Affichages des informations de la base de données	8
4.5	Gestion des informations et protection des mots de passe	8
5	Design du site et utilisation de CSS	8
6	Rôle de chacun	9
7	Amélioration et évolution du site	9

1 Introduction

Le site que nous avons créé est destiné à la gestion des emprunts des livres proposés par l'association ScIIencE de l'école. Plusieurs contraintes devaient donc être respectées :

- La gestion de plusieurs types de comptes utilisateurs
- La gestion d'une large base de données (principalement pour les livres et les utilisateurs)
- La vérification de la validité des informations entrées

Nous avons également déterminer une liste de fonctionnalités nécessaires pour le site. Pour faciliter l'utilisation et l'accès aux livres, le site doit permettre :

- La consultation des livres disponibles par les utilisateurs
- La réservation de livres par les utilisateurs
- La notation des livres empruntés par un utilisateur
- La consultation des commentaires postés par les autres utilisateurs concernant un livre donné
- Pour chaque utilisateur, la possibilité d'éditer son propre profil
- L'emprunt de livres (validé par un administrateur)
- Le rendu de livres (validé par un administrateur)
- L'ajout de livres (par un administrateur)
- La consultation des livres non rendus (visible par les administrateurs)

2 Mise en place de la base de données

La base de données implantée tourne sous docker et utilise PostgreSQL. Nous avons mis en place 6 tables :

- La table **"User"** contenant :
 - `id_user` : l'id de l'utilisateur (clé primaire)
 - `prénom` : le prénom de l'utilisateur
 - `nom` : le nom de l'utilisateur
 - `"pseudo"` : le pseudo de l'utilisateur (unique)
 - `ddn` : la date de naissance de l'utilisateur
 - `mdp` : un hashcode du mot de passe de l'utilisateur
 - `mail` : l'adresse mail de l'utilisateur
 - `nb_livres_empruntés` : le nombre de livres empruntés par l'utilisateur
 - `nb_livres_rendus` : le nombre de livres rendus par l'utilisateur
 - `est_admin` : un booléen indiquant si l'utilisateur est administrateur
 - `(nom, prenom, pseudo)` est unique
- La table **"Livre"** contenant :
 - `id_livre` : l'id du livre (clé primaire)
 - `titre` : le titre du livre
 - `publication` : la date de publication du livre
 - `couverture` : un chemin vers l'image de couverture
 - `editeur` : l'éditeur du livre
 - `emprunteur` : l'utilisateur ayant emprunté ce livre (clé étrangère référence "User"(id_user))
 - `date_emprunt` : la date à laquelle ce livre a été emprunté pour la dernière fois
- La table **"Review"** contenant :
 - `id_review` : l'id de la review (clé primaire)
 - `id` : l'id du livre concerné (clé étrangère référence "Livre"(id_livre))
 - `personne` : l'id de l'utilisateur concerné (clé étrangère référence "User"(id_user))
 - `texte` : le texte descriptif correspondant à l'avis donné par l'utilisateur
 - `note` : la note donnée par l'utilisateur (entre 0 et 10)
- La table **"Historique"** contenant :
 - `id_hist` : l'id de l'élément (clé primaire)
 - `id_livre` : l'id du livre concerné (clé étrangère référence "Livre"(id_livre))

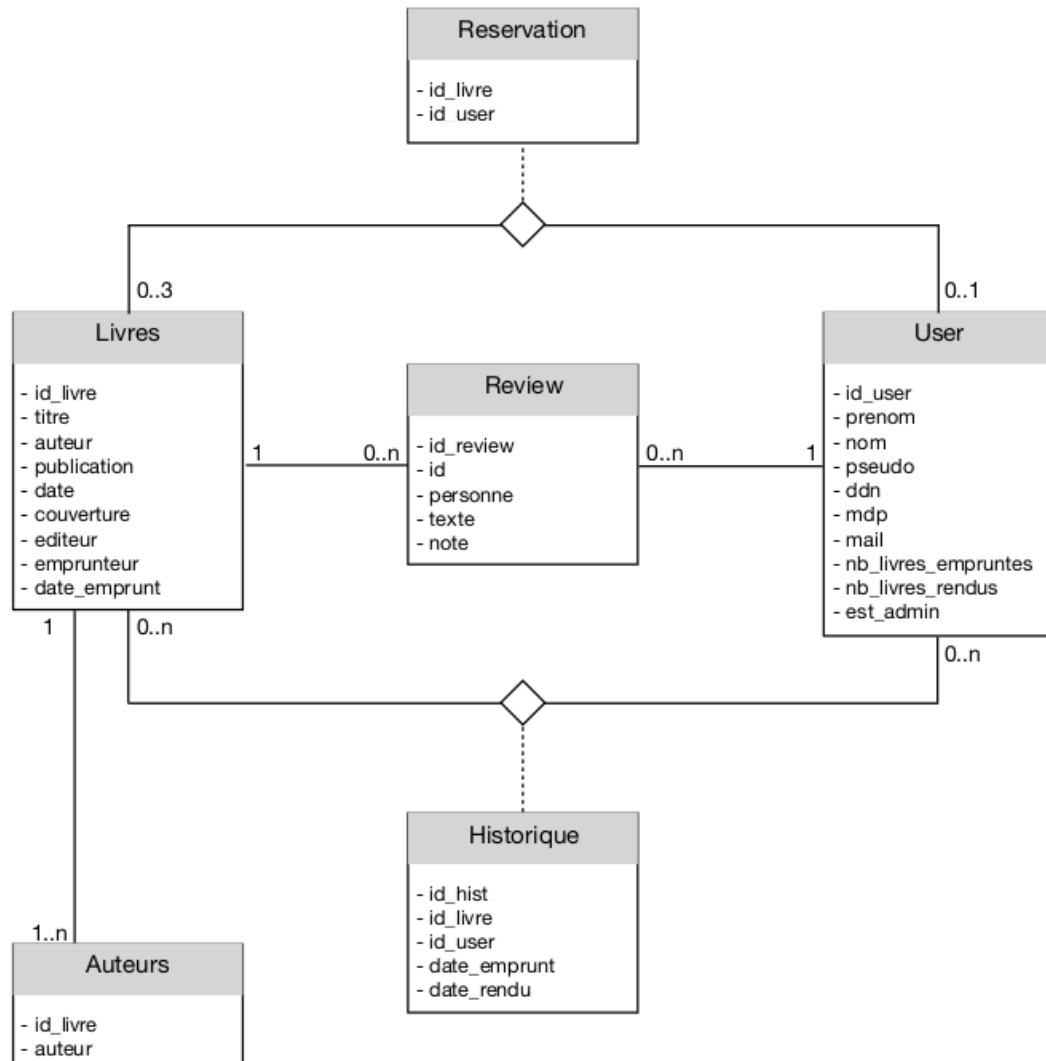
- `id_user` : l'id de l'utilisateur concerné (clé étrangère référence "User"(id_user))
- `date_emprunt` : la date à laquelle le livre a été emprunté
- `date_rendu` la date à laquelle le livre a été rendu
- La table "**Reservation**" contenant :
 - `id_livre` : l'id du livre concerné (clé étrangère référence "Livre"(id_livre))
 - `id_user` : l'id de l'utilisateur concerné (clé «étrangère référence "User"(id_user))
 - (`id_livre`, `id_user`) est clé primaire
- La table "**Auteur**" contenant :
 - `id_livre` : l'id du livre concerné (clé étrangère référence "Livre"(id_livre))
 - `auteur` : nom de l'auteur
 - (`id_livre`, `auteur`) est clé primaire

Nous avons également mis en place plusieurs fonctions et triggers sur cette base de données pour maintenir la cohérence des données lors de modifications qui auront lieu.

Nous avons donc deux triggers, le premier qui, lors de l'insertion d'un nouvel utilisateur, assure que les valeurs insérées sont bien cohérentes, et en particulier, initialise les valeurs de `nb_livres_empruntes` et de `nb_livres_rendus` à 0. Le second qui, lors de l'insertion d'un nouveau livre, assure que les valeurs sont cohérentes, et en particulier, initialise les valeurs de `emprunteur` et de `date_emprunt` à NULL.

Enfin, nous avons un trigger qui, lors de la mise à jour d'un élément de la table "Livre", augmente les valeurs des variables `nb_livres_empruntes` et `nb_livres_rendus` dans "User" pour l'utilisateur concerné si la mise à jour correspond à l'emprunt ou au rendu d'un livre.

3 Représentation de la base de données



4 Mise en place du modèle PHP

4.1 Contrôle de la base de données

Afin de permettre l'accès et la modification de la base de données par les fonctions PHP, nous avons mis en place une classe et un repository par table de la base de données.

Les pages PHP ne pourront interagir avec la base de données qu'au travers de ces fonctions.

Chaque classe contient une fonction pour mettre en place la valeur de chaque attribut qu'elle contient et une fonction pour obtenir la valeur de chaque attribut.

Chaque repository contient au moins une fonction `fetchAll` qui retourne une liste d'objets de la classe correspondante, représentant l'ensemble des données de la table adéquate.

De plus, selon les besoins d'accès à chaque table, certains repository contiennent des fonctions `fetch` avec paramètres, ce qui permet d'exécuter des requêtes SQL plus précises.

Exemple : `fetchId($id)` dans `LivreRepository.php` permet de récupérer les livres en fonction de leur `id`.

4.2 Gestion des sessions

Pour gérer les sessions utilisateurs, nous utilisons les variables de session disponibles en PHP grâce à la fonction `session_start()`, la seule variable de session mise en place est : `$_SESSION['id_user']` qui contient l'`id` de l'utilisateur courant. La différenciation entre utilisateur standard et administrateur est faite par le serveur à chaque entrée de page grâce à l'`ID` de l'utilisateur pour éviter des abus côté client.

À chaque chargement de page, on cherche dans la base de données si l'utilisateur est administrateur, et si c'est le cas, certaines fonctionnalités supplémentaires sont affichées dans la page.

Pour les pages exclusives aux administrateurs on ajoute une redirection automatique vers la page d'accueil (`index.php`) si l'utilisateur n'est pas connecté ou si l'utilisateur n'est pas administrateur pour éviter des accès intempestifs à ces pages.

4.3 Validation des informations envoyées par les utilisateurs

Pour les formulaires envoyés par les utilisateurs, (lors d'une inscription, de la modification du mot de passe, de la modification des informations personnelles, ou lors de l'ajout d'une review), nous avons décidé d'effectuer une vérification en deux temps.

Tout d'abord, nous effectuons une vérification pour savoir si l'ensemble des données sont entrées et également sous une forme valide grâce aux formulaires `HTML5` et de fonctions `javascript` de la page (vérification de la validité de la date, vérification de la bonne forme de l'adresse email, ...)

Ensuite, si la vérification côté client a été validée, une nouvelle validation est effectuée côté serveur, la validité des données est à nouveau vérifiée, puis, le serveur vérifie que les données sont compatibles et correspondent à la base de données (pseudo valide et pas déjà utilisé, ensemble (nom, prénom, pseudo) pas déjà utilisé, références correctes vers les auteurs dans le cas des livres, ...) si les deux tests sont passés, la modification de la base de données est effectuée.

4.4 Affichages des informations de la base de données

Pour afficher la liste des livres dans `bibliotheque.php` ou la liste des review dans `list_review.php` par exemple, on utilise des tableaux `HTML` dont on modifie dynamiquement certains attributs grâce à `javascript` (dans `list_review.php` par exemple la colonne pour écrire une review n'est affichée que lorsque l'utilisateur clique sur le bouton correspondant).

L'utilisation du `CSS` (avec des dimensionnements en pourcentage) permet également d'adapter plus facilement l'affichage des tableaux aux différentes tailles d'écrans, permettant un affichage adaptatif.

4.5 Gestion des informations et protection des mots de passe

Tous les formulaires du site sont envoyés en utilisant la méthode `POST` fournie par `PHP`. De plus, nous stockons les mots de passe dans la base de données en utilisant la fonction `password_hash(...)` de `PHP`. Ainsi, non seulement nous nous assurons que les informations transmises au serveur ne peuvent pas être récupérées par un tiers, mais en plus, notre base de données ne stocke pas les mots de passe des utilisateurs "en clair". Lors de la connexion d'un utilisateur sur le site, le serveur vérifie simplement la concordance entre le hash du mot de passe en base de données et celui rentré par l'utilisateur grâce à `verify_password(...)`.

5 Design du site et utilisation de `CSS`

Pour la présentation du site, nous avons mis en place plusieurs feuilles de style `CSS` pour répondre aux besoins de chaque page.

Ces feuilles gèrent entre autre l'affichage des titres, des bannières, la bonne intégration des images sur la page ainsi que le défilement du contenu par rapport à l'arrière plan. De plus, nous avons utilisé `CSS` directement dans notre code `HTML` sur certaines pages pour une meilleure expérience utilisateur.

Par exemple, sur les pages contenant un formulaire, le `CSS` intégré nous permet de mettre en

évidence les champs comportant une erreur ou non rempli.

6 Rôle de chacun

La liste des répartitions s’est faite de manière suivante :

- Louis Fourade : Gestion de la base de données
- Tanguy Charles : Création et gestion de la base de données et expérience utilisateur
- Louis Gauthier : Visuel du site
- Paul Fleurance : Manager, initiateur du projet et superviseur

7 Amélioration et évolution du site

Le site est entièrement fonctionnel et utilisable tel quel, cependant, dû à un manque de temps et d’expérience, certaines fonctionnalités pourraient être améliorées et d’autres n’ont pas été implémentées.

Tout d’abord, la structure du site vis-à-vis de la base de données peut être repensée afin de mettre en place un modèle MVC complet beaucoup plus efficace que ce qui est en place actuellement pour le projet (actuellement le Modèle et la Vue sont parfois sur les mêmes fichiers).

De plus, notre base de données étant relativement petite, l’affichage des livres disponibles dans la bibliothèque se fait d’un bloc. Il serait judicieux, lorsque le nombre de livres stockés augmentera de mettre en place un système d’affichage et d’accès à la base par tranche, permettant par exemple de lister les entrées par 50 ou par 100.

Notre site ne gère également pas les livres disponibles en plusieurs exemplaires, si un livre est en double par exemple, il faut créer une entrée de ce livre dans la bibliothèque par copie, ce qui peut être amélioré.

Enfin, il serait possible d’améliorer le système d’emprunt et de réservation pour permettre par exemple des réservations à des dates précises (permettant ainsi à plusieurs personnes de réserver le même livre) ou des emprunts sur des dates spécifiques.