

Bibliothèque

Ce site répond à la demande des clubs de prêt de livres : **Bakclub** et **BD-SF**. Leur système d'emprunt n'était pas très performant (feuille affichée dans une salle) alors nous avons souhaité créer un site web qui permette de centraliser tous les livres pour permettre aux étudiants de les emprunter plus facilement.

Nous avons choisi de mettre en place une page d'accueil où tous les livres seront visibles, et où nous pourrions faire une recherche de livre suivant son titre ou son auteur. Chaque utilisateur enregistré peut emprunter un livre, et uniquement les administrateurs peuvent ajouter un livre dans la base de données. Tous les utilisateurs peuvent s'enregistrer pour l'instant, mais avant de le rendre public, il faudra utiliser le système d'utilisateurs de Arise pour certifier que les étudiants sont bien de l'ENSIE.

Une fois les bases des pages définies, il a fallu redéfinir la structure du site car elle n'était pas encore adaptée au format MVC. La première fonction fut alors une fonction de gestion de View avec un `loadView` : cette fonction ne posa pas de problème étant donné son caractère récurrent dans les TP effectués tout au long de l'année.

Ensuite, le premier problème s'est posé lors de la création d'une barre de recherche, car l'utilisation d'un script javascript qui fait appel à une fonction php pour récupérer les œuvres qui correspondent à ce que l'utilisateur a entré fut complexe à gérer à cause des **LIKE** dans le sql (lorsqu'on utilise un **LIKE**, on ne peut pas faire un `bindParam` sur l'argument du **LIKE** qui est compris dans des %). De plus, l'affichage des aides à la recherche n'était pas user-friendly, il a fallu alors utiliser un script javascript plus performant et modifier des paramètres du CSS pour rendre tout ceci lisible par l'utilisateur. Suite à cela, il fallait donc charger la vue en fonction de livres différents ce qui fut simple avec l'utilisation du modèle MVC.

Nous nous sommes intéressés après la mise en place de la recherche, à la mise en forme du tableau des livres, où nous avons décidé de rajouter la couverture des livres, en rajoutant l'url de l'image dans la base de données et en la récupérant pour afficher une image dans le tableau.

Le problème le plus complexe vient ensuite, la gestion des utilisateurs, qui n'avait jamais été abordée encore en cours. Il a fallu créer une page de connexion, une page d'enregistrement d'utilisateur, un affichage différent suivant s'il on est connecté ou non, et en plus, une page pour que les administrateurs puissent ajouter des livres dans la base de données. Ainsi nous avons : une authentification, un compte administrateur avec fonctionnalités, une base de données relationnelle avec une table « book » pour stocker les livres, une table « users » pour stocker les utilisateurs et leurs mots de passes(hashés) et une table « summary » pour avoir les résumés et les likes et dislikes des utilisateurs (fonctionnalité à ajouter par la suite). De plus, le **INSERT** est utilisé lors de l'ajout d'un livre ou d'un utilisateur, le **DELETE** lors de la suppression d'un livre par un administrateur, et le **SELECT** lors de l'affichage des livres suite à une recherche.

Pour finir, le dernier problème s'est situé lors de l'affichage du détail d'une œuvre avec son résumé, car certaines œuvres n'ont pas forcément de résumé. Il a fallu donc ajouter ces œuvres dans la base de données « summary » pour que le **JOIN** fonctionne même sur les œuvres qui n'ont pas de résumé.

Dernièrement, un problème de SQL est survenu, lors d'un LEFT OUTER JOIN, si les id étaient présents dans la base de gauche mais pas dans la base de droite, ils étaient tout simplement éliminés du JOIN. C'est pour cela qu'il a fallu ajouter toutes les œuvres dans la base de données « summary ». De plus, il fallait rajouter la contrainte ON DELETE CASCADE pour que lors de la suppression d'un livre dans la base de données « book », il soit aussi supprimé dans « summary ».

Une fonctionnalité de likes a voulu être ajoutée, mais pour pouvoir stocker les likes, et surtout empêcher qu'un même utilisateur puisse voter plusieurs fois, il aurait fallu rajouter une nouvelle base de données relationnelle qui permettrait de savoir quel utilisateur a aimé quelle œuvre, et ainsi qu'il ne vote pas deux fois.

Le style n'était pas très important pour ce projet, mais nous avons tenté d'embellir le site avec des images, des décorations de texte, positionner les paragraphes et une page d'accueil.