

Bibliothèque

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

Nous avons choisi de mettre en place une page où tous les livres sont visibles, et où nous pouvons faire une recherche de livre suivant son titre ou son auteur. Chaque utilisateur enregistré peut emprunter un livre, mais uniquement les administrateurs peuvent ajouter, rendre ou supprimer 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 utilisateurs sont bien de l'ENSIIE.

Une fois les bases de données 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 : celle-ci 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. En effet 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. Tout ceci 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 a donc fallu charger la vue en fonction des différents livres, ce qui fut simple avec l'utilisation du modèle MVC.

Nous nous sommes ensuite intéressés à la mise en forme du tableau des livres. Nous avons décidé d'y inclure la couverture des livres, en rajoutant le chemin de l'image dans la base de données et en la récupérant pour afficher une image dans le tableau.

Le problème le plus complexe est venu ensuite car il n'avait pas été abordée en cours : la gestion des utilisateurs. Il a fallu créer une page de connexion, une page d'enregistrement d'utilisateur, un affichage différent suivant s'il est connecté ou non, et 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, les likes et les 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 la suppression d'un livre par un administrateur, et le **SELECT** lors de l'affichage des livres suite à une recherche.

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

Récemment, un problème de SQL est survenu, lors d'un LEFT OUTER JOIN. En effet, si les id étaient présents dans la base de gauche mais pas dans celle 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 a fallu 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. Celle-ci aurait permis de savoir quel utilisateur a aimé quelle œuvre, et ainsi qu'il ne puisse pas voter deux fois.

Enfin, nous avons embelli la structure du site grâce à du code CSS. Celui-ci nous a permis de positionner les paragraphes, de créer une page d'accueil, et d'ajouter des décorations de texte et des images.

La répartition des tâches s'est faite de la façon suivante :

- Aristide Auphan fut en charge de la gestion des bases de données SQL, de l'agencement de tout le site sous CSS, ainsi que de donner de l'attrait au site avec par exemple la page d'accueil ou bien les images de présentation des livres.
- Basile Bonicel et Nathan Jessen ont embelli le site via le CSS en ajoutant des images de fond et en modifiant les couleurs du site.
- Raphaël Carly s'est occupé du PHP, du JavaScript et de l'accès aux bases de données en créant les diverses pages utilisables. De plus, la remise en forme en structure MVC lui a été attribuée.