

Nom : OSHEI Andrew  
No Etud : 20001943

## TP 1 - Pagination

Pour ce TP, je n'ai pas eu besoin de rechercher de sources externes. L'année dernière en licence nous avons travaillé avec des bases de données MySQL et PHP. Bien que nous n'ayons pas implémenté la pagination l'année dernière, l'algorithme était assez simple pour être compris par moi-même.

Pour les données qui seront affichées par ma page, j'ai recyclé un script SQL que j'ai utilisé l'année dernière dans mon cours sur les bases des données, trouvé dans le fichier mock.sql. Ce script crée une base de données appelée « pagination\_mock », puis il crée une table « page\_mock » et enfin il remplit la table avec des données « mock » au format suivant :

ID – Titre – Categorie - Auteur

L'approche de base que j'ai utilisée était de fixer d'abord le nombre maximum de lignes à afficher par page, dans ce cas j'ai choisi 10 lignes par page. Je me connecte ensuite à la base de données et demande toutes les données de la table « page\_mock » en utilisant 'SELECT \* FROM page\_mock'. La taille des données à afficher est alors déterminée à l'aide de la commande mysqli\_num\_rows(). Cela renvoie le nombre total de lignes de données que je divise par 10 (le nombre maximum de lignes à afficher par page) afin d'obtenir le nombre de pages nécessaires pour afficher toutes les données. Je vérifie ensuite quelle est la page actuelle et détermine le décalage afin d'afficher les données pertinentes pour la page actuelle.

Une deuxième requête est alors faite à la base de données MySQL en utilisant la commande 'SELECT \* FROM page\_mock LIMIT offset, row\_limit'. Cela ne renvoie que les lignes qui doivent être affichées sur la page actuelle. À l'aide d'une boucle, les données sont ensuite écrites dans un tableau html ligne par ligne.

Pour la pagination, une simple boucle for est utilisée, de un au nombre maximum de pages. Chaque cycle de la boucle écrit un lien html vers la page correspondante. Des liens vers la page précédente et vers la page suivante sont également ajoutés, avec une vérification supplémentaire pour la page en cours qui désactive le lien pour le bouton de retour de page si nous sommes actuellement sur la première page et désactive le bouton de page suivante si nous sommes actuellement sur la dernière page. Chacun de ces éléments est regroupé dans un élément html div qui s'affiche centré sous le tableau html qui affiche les données.

Quelques notes supplémentaires, la page utilise une variable « page » envoyée via http afin de changer la page actuelle. Cette variable est vérifiée à chaque chargement de la page pour déterminer la page courante, si la variable n'est pas présente le programme suppose que nous sommes sur la première page. Les boutons de pagination utilisent ce même mécanisme pour changer de page. Enfin, j'ai également ajouté des styles CSS à la page afin de rendre la présentation des données claire et organisée.