Carnet Projet Web Mapping

06/12/2022 après-midi : Prise en main du sujet et relecture des autres sujets en lien avec le mien pour situer le contexte. Plusieurs questions posées au commanditaire et chargement de la base de données sur les PC de l'école pour voir la forme de cette dernière. Avec un problème de fichier corrompus au niveau du SQL résolu à l'aide d'un backup.

12/12/2022 matin : Installation de wamp et de PostgreSQL sur mon poste personnel et import de la base de données avec le backup. Visualisation de la vidéo sur formation Temp sur l'architecture client/serveur.

12/12/2022 après-midi : configuration de wamp à l'aide du mail envoyé avec le php.ini et l'activation avec un test.ws.php pour tester la connexion.

Problème : la base de données n'est pas détectée par pg_connect () malgré la configuration effectuée.

Problème résolu : la base de données était sur le port 5433 et non 5432 (vérification effectuée à l'aide de pgadmin)

13/12/2022 matin : Programmation du test unitaire en créant un formulaire à partir duquel on récupère les insertions de l'utilisateur pour pouvoir sur php récupérer. Le formulaire étant dynamique par le fait que l'on peut avoir plusieurs illustrations pour un exemple j'ai adapté le code php pour récupérer en mode POST les informations relatives aux différentes illustrations pour les insérer dans la base de données (fond cartographique, niveau de zoom, point centre).

L'aspect dynamique du formulaire est possible grâce au fichier javascript dans lequel on a un eventListener sur le champ « nb_illu » qui correspond au nombre d'illustration.

13/12/2022 après-midi : Le code est fonctionnel mais des problèmes sont notables par exemple le code php est susceptible aux injections SQL.

Solution :

Les requêtes préparées sont une technique de programmation qui permet de séparer la requête SQL des données d'entrée de l'utilisateur. Au lieu d'incorporer les données directement dans la requête SQL, les requêtes préparées utilisent des paramètres de requête pour définir les valeurs des données d'entrée. Les paramètres de requête sont marqués dans la requête SQL en utilisant un signe de dollar ($) suivi d'un numéro de paramètre (ex: $1, $2, etc.).

Dans le code présenté, la fonction pg_query_params() est utilisée pour exécuter des requêtes préparées. Cette fonction prend la requête SQL en premier argument, et un tableau de valeurs de paramètres en deuxième argument. Les valeurs de paramètres sont automatiquement échappées pour empêcher les attaques d'injection SQL.

Pour finir j'ai rajouté un CSS pour le formulaire du test unitaire pour rendre le formulaire plus lisible.