SOMMAIRE

- Rappel Fonction fléchées
- Rappel SELECT
- Requêtes préparées
- FETCH
- INSERT INTO
- Sécuriser les champs
- TP1 et TP2

FONCTIONS FLÉCHÉES

- Les fonctions fléchées sont une autre façon plus concise d'écrire des fonctions.
- La création d'une fonction fléchée ne crée pas de contexte et donc ne redéfini pas le this. elles n'ont pas leur propre liaison avec le this. À l'inverse des fonctions normales, les fonctions fléchées partagent le même this que leur scope parent. Du coup, le this que vous pourrez utiliser dans la fonction fléchée est celui du code parent. Ainsi on utilisera moins le bind().
- Une fonction fléchée sans paramètres nécessite des parenthèses,
- Une fonction avec 1 seul paramètre ne nécessite pas de parenthèse.
- Une fonction avec plusieurs paramètre nécessite des parenthèses.

REQUETES PREPAREES

Le principe des requêtes préparées est, de préparer les requêtes pour ensuite les utiliser. Elles permettent de se prémunir contre les injections SQL puisque seules les valeurs concrètes sont reprises. Elles sont aussi utiles car elles consomment beaucoup moins de ressources et sont plus rapides.

L'intégration et l'utilisation des requêtes préparées se fait en 3 temps:

1) Le première permet de créer des variables ou des caractères de remplissage qui sont uniquement remplacés par les valeurs réelles dans le système, en utilisant la fonction prepare(),
2) Le SGBD analyse les variables,
3) La requête préparée est exécutée.

FETCH VS FETCHALL

La fonction fetch () permet de récupérer les données de la requête une par une, elle est utilisé lorsqu'on s'attend à récupérer de nombreuses données.

fetchAll () récupérera toutes les lignes du résultat de la requête en une seule fois. Elle s'utilise lorsqu'on récupère peu de données .

RAPPEL SELECT

La requête SELECT permet de récupérer des éléments de la bdd. on pourra ajouter des conditions pour pouvoir filtrer ce qui sera retournés.

```
<?php
        $req = $bdd->prepare('SELECT age FROM use WHERE nom = ?');
        $req->execute(array($_GET['nom_recherché']));
?>
```

INSERT INTO

La requête INSERT INTO va permettre d'insérer des données dans la base de données.

```
<?php
$req = $bdd->prepare('INSERT INTO user (nom, prenom, age) VALUES (:nom, :prenom,  :age)');
$req->execute(array(
        'nom' => $nom,
        'prenom' => $prenom,
        'age' => $age,
        ));
?>
```

SECURISER LES CHAMPS

Pour s'assurer que les valeurs saisies dans les champs d'un formulaire ne contiennent pas de code nous pouvons utiliser la fonction htmlspecialchars() .

Cette fonction permet d'avoir un code qui ne soit pas interprété comme tel mais plutôt comme une chaîne de caractère.

Ce qui permet de se protéger un minimum des failles XSS.

TP 1

Reprendre le formulaire d'inscription du Tp correspondant.

Effectuer les vérification nécessaires sur les champs puis créer une base de

données ainsi qu'une table où seront stockés les données du formulaire.

Ensuite insérer les données saisies via le formulaire dans la base de données.

Attention: S'assurer que le code saisie dans les champs de soit pas interprété

(faille xss).

TP 2

Une fois inscrit, l'utilisateur peux choisir un login et un mot de passe, une vérification sera faite sur le mot de passe en demandant de le saisir une seconde fois pour s'assurer qu'il soit identique.

Les login et mot de passe seront stockés en bdd, ainsi lorsque l'utilisateur se connecte sa session est donc active.

Créer une page de profil sur laquelle accèdera l'utilisateur après s'être authentifié.

Afficher un bouton de déconnexion, lorsque ce dernier est cliqué l'utilisateur est dirigé vers la page d'accueil.