

## SITE DE COMMENTAIRE

Pour le BTS SIO SLAM je dois apprendre à coder en PHP, pour se faire j'ai développé mon premier CRUD en PHP, un site de commentaire qui permet d'ajouter / modifier / supprimer et même liker les commentaires.

Pseudo :

Message :

Envoyer

### Commentaires :

dddddddddd : Salut!

♥ 1 [\[Liker\]](#) [\[Supprimer\]](#) [\[Modifier\]](#)

regg : etrtztzte

♥ 4 [\[Liker\]](#) [\[Supprimer\]](#) [\[Modifier\]](#)

La première chose que j'ai fait c'est de créer les champs pseudo et commentaire ainsi que de créer un ID pour chaque commentaire (je savais que ça me serait utile plus tard)

```
if(isset($_POST['post'])){
    // 1. On récupère les données
    $pseudo = htmlspecialchars($_POST['pseudo']);
    $commentaire = htmlspecialchars($_POST['commentaire']);
    $id_update = $_POST['id_en_cours'];

    // 2. On vérifie si c'est rempli
    if(!empty($pseudo) && !empty($commentaire)) {

        if(!empty($id_update)){
            // MODE ÉDITION
            foreach($_SESSION['commentaires'] as $index => $com){
                if($com['id'] === $id_update){
                    @var mixed
                    $_SESSION['commentaires'][$index]['pseudo'] = $pseudo;
                    $_SESSION['commentaires'][$index]['commentaire'] = $commentaire;
                }
            }
        } else {
            // MODE CRÉATION
            $_SESSION['commentaires'][] = [
                'id' => uniqid(),
                'pseudo' => $pseudo,
                'commentaire' => $commentaire,
                'like' => 0
            ];
        }

        // On redirige direct après le succès : ça vide le formulaire et les erreurs
        header("Location: " . $_SERVER['PHP_SELF']);
        exit();
    } else {
        // Si un des champs est vide
        $erreur_commentaire = "Tous les champs doivent être remplis.";
    }
}
```

J'ai ensuite créé le formulaire html

```
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Espace commentaire</title>
  <link rel="stylesheet" href="style.css"> <!-- Pas encore créer -->
</head>
<body>
  <form action="" method="post">
    <input type="hidden" name="id_en_cours" value="<?php echo $id_a_modifier; ?>">
    <label>Pseudo :<br>
    <input type="text" name="pseudo" value="<?php echo $pseudo_edit; ?>"><br>
    </label>
    <label>Message :<br>
    <textarea name="commentaire"><?php echo $comm_edit; ?></textarea>
    </label><br>
    <input type="submit" name="post" value="<?php echo $mode_edition ? 'Enregistrer les modifications' : 'Envoyer'; ?>">
  </form>

  @var string|null $erreur_commentaire

  <?php if(isset($erreur_commentaire)) {
    echo "<p style='color: red;'>Erreur : " . $erreur_commentaire . "</p>";
  }?>

  <?php if(isset($message_succes)) {
    echo "<p style='color: green;'>" . $message_succes . "</p>";
  }?>

  <hr>
  <h2>Commentaires :</h2>
  <div class="commentaires">
    <?php
    if(!empty($_SESSION['commentaires'])){
      foreach($_SESSION['commentaires'] as $com){
        echo "<div style='border-bottom:1px solid #ccc; padding:10px;'>";
        echo "<strong>" . $com['pseudo'] . "</strong> : " . $com['commentaire'] . "<br>";
        // Affichage du compteur et du bouton Like
        echo "<span>❤️ " . $com['like'] . "</span> ";
      }
    }
  </div>
</body>
```

Evidemment au début, il n'y avait rien en lien avec le like la réponse et la suppression, j'ai du peu à peu modifier des éléments que j'avais déjà mis !

```

    echo "<span>❤️ " . $com['like'] . "</span> ";
    echo "<a href='?like_id=" . $com['id'] . "'>[Liker]</a> ";
    // Bouton Supprimer
    echo "<a href='?delete_id=" . $com['id'] . "' onclick='return confirm(\"Supprimer ?\")' style='color:red;'>[Supprimer]</a>";
    // Bouton Modifier
    echo "<a href='?edit_id=" . $com['id'] . "' style='color:green; margin-left:10px;'>[Modifier]</a>";
    echo "</div>";
  }
} else {
  echo "<p>Aucun commentaire pour le moment.</p>";
}
?>
</div>
</body>
```

Le premier gros défi était la suppression des commentaires

```
<!DOCTYPE HTML>
<?php
session_start(); // On démarre la session
$erreur_commentaire = null; // Sans ça le message erreur était tout le temps affichés.

if(!isset($_SESSION['commentaires'])){ //On créer le tableau des commentaires
  $_SESSION['commentaires'] = array();
}

if(isset($_GET['delete_id'])){ // On vérifie si le mot delete_id est présent dans l'URL . si oui, le bloc s'exécute.
  $id_a_supprimer = $_GET['delete_id']; // On stocke la valeur de l'ID reçue dans une variable pour l'utiliser plus facilement.

  // On parcourt le tableau pour trouver celui qui a cet ID
  foreach($_SESSION['commentaires'] as $index => $com){ // Pour chaque commentaire
    if($com['id'] == $id_a_supprimer){ // On regarde si l'ID (index) du commentaire qu'on regarde est celui qu'on veut delete.
      unset($_SESSION['commentaires'][$index]); // SI OUI on le retire du tableau
    }
  }
  // On réindexe pour éviter les trous (le com [2] devient le [1] si on supprime le 1 etc...)
  $_SESSION['commentaires'] = array_values($_SESSION['commentaires']);

  // On redirige vers la page sans le "?delete_id=..." dans l'URL
  header("Location: " . $_SERVER['PHP_SELF']);
  exit();
}
```

J'ai dû apprendre à utiliser l'ID d'un commentaire notamment en le récupérant avec \$\_GET, j'ai mis pas mal de temps à réussir cette fonction c'est pour ça que j'ai mis beaucoup de commentaire, j'ai du demandé de l'aide à mon professeur car je ne trouvais pas la solution (c'est là où j'ai pensé à GET)

```
//On va créer un système de like au commentaires
if(isset($_GET['like_id'])){ // Si est existant get like id (récupérer depuis URL l'id du com qu'on like)
    $id_a_liker = $_GET['like_id']; // $id_a_liker prends la valeur de $_GET['like_id']
    foreach($_SESSION['commentaires'] as $index => $com){
        if($com['id'] === $id_a_liker){ // On incrémente si le commentaire qu'on regarde est le meme que celui qu'on veut like
            // On incrémente la case 'like' du bon commentaire
            $_SESSION['commentaires'][$index]['like']++; // ++ ça revient à $var = $var + 1
        }
    }
    header("Location: " . $_SERVER['PHP_SELF']); // on redirige pour pas être bloqué
    exit();
}
```

La création de la fonction like n'était pas demandé, ayant eu besoin d'aide externe pour supprimer j'ai décidé d'implémenter un truc en plus et la logique était assez similaire, j'ai du simplement initialiser le compteur de like à 0 dès la création d'un commentaire, et créer l'incrément, le reste était assez semblable en structure à la suppression.

```
// On va créer le système de modification des commentaires
$mode_edition = false;
$id_a_modifier = '';
$pseudo_edit = '';
$comm_edit = '';

if(isset($_GET['edit_id'])){
    $mode_edition = true;
    $id_a_modifier = $_GET['edit_id'];

    // On cherche les infos actuelles pour les remettre dans le formulaire
    foreach($_SESSION['commentaires'] as $com){
        if($com['id'] === $id_a_modifier){
            $pseudo_edit = $com['pseudo'];
            $comm_edit = $com['commentaire'];
        }
    }
}
```

ENFIN j'ai créé la modification des commentaires, j'initialise sur false le mode édition je crée une nouvelle variable pour l'ID. Si l'ID à éditer est le même que celui du commentaire choisi alors le mode Edition devient vrai l'id à modifier obtiens la valeur de l'id edit\_id, on récupère les pseudo et

contenu du commentaire du commentaire à modifier ce qui permet de ne pas tout ré écrire (on peut donc changer le pseudo et le commentaire au complet ce qui serait très grave pour un vrai site, mais dans notre cas ça ira).

ET VOILA ! Mon premier CRUD est « terminé », je n'ai pas rencontré de difficulté particulière hormis pour la possibilité de supprimer un commentaire, une fois ceci fait tout est allé en avant très vite ! Prochainement je me lancerai sur Laravel pour mon apprentissage du BTS SIO SLAM.