Rapport sur le projet de Technologie - Web/Serveur

Wassim Saidane, Aurélien Authier

Table des matières

Introduction

1 Technologies utilisés

Pour ce projet, nous avons utilisé le language de programation PHP, nous avons choisis PHP car c'était le language le plus répendu, PHP est également facile à utilisé car on peut y inclure du HTML. On peut donc coder sans ce soucier d'aspect technique coté serveur. Un de nous deux connaissait déjà ce language.
Nous utilisé une base de donnée mySQL, le language nous avait été ensseigner en L2. Une autre raison est l'utilisation d'un serveur appache (plus simple pour nous qui codons sur Windows), pour le PHP, nous avons donc utilisé phpMyAdmin via WampServeur. Tous ces outils fonctionnent bien ensemble, et nous ont permis de gagner du temps sur des aspects techniques (serveur) que nous ne maitrisons pas encore bien.

2 Comment lancer notre code ?

Nous avons fait un mode d'emploi (... .pdf) explicant comment on lance le code.

3 La base de données

Avant de coder quoi que soit, nous nous sommes d'abord intéresser au contenu de la base de données. Nous avons ainsi créer quatre tables sur phpMyAdmin.

3.1 categories

Il s'agit d'un ajout (par rapport au sujet initial) de notre part. L'utilisateur choisis la catégorie en rapport avec son article.
La table est composé dans son état actuel de deux champs :
- id
- name

On a deux catégories [1]

1. On peut imaginer qu'un administrateur ou même un utilisateur créé des catégories selon son envie

3.2 membres

Ici sera stocké les données des utilisateurs qui ce seront inscrit. On a six champs :
- id
- pseudo
- email
- mdp
- isAdmin [2]
- isBanned [2]

3.3 sujet

Nous avons décider de marquer une petite différence avec l'énoncé (par facilité), au lieu d'avoir des articles et des commentaires classiques, nous avons des sujets et une discussion sous chaque sujets. D'un point de vu utilisateur celà ne change rien.
On a trois champs :
- id
- name
- categorie

3.4 postSujet

Il s'agit de la discussion dont nous avons parler précédement. On a cinq champs :
- id
- propri [3]
- contenu
- date
- sujet

2. binaire même si dans les faits il s'agit d'un entier
3. L'identifiant de celui qui à créer le commentaire

4 Le code

Nous avons divisé le code en trois, un repértoire css pour le style, un répertoire pour les fonctions et le reste.

4.1 L'inscription

Le fichier inscription.php va utiliser un formulaire HTML classique qui va appeler inscription.class.php dans le repértoire fonction.
Extrait de inscription.php :

```
        <form method="post" action="inscription.php">
            <p>
                <input name="pseudo" type="text" placeholder="
    Pseudo..." required /><br>
                <input name="email" type="text" placeholder="
    Adresse email..." required /><br>
                <input name="mdp" type="password" placeholder="Mot
    de passe..." required /><br>
                <input name="mdp2" type="password" placeholder="
    Confirmation..." required /><br>
                <input type="submit" value="S'inscrire!" />
                <?php
                    if(isset($erreur)){
                        echo $erreur;
                    }
                ?>
            </p>
        </form>

```

Ansi toutes les données entrées vont être récupéré dans le constructeur de la classe.
Extrait de inscription.class.php :

```
        class inscription{

            private $pseudo;
            private $email;
            private $mdp;
            private $mdp2;
            private $bdd;

        public function __construct($pseudo,$email,$mdp,$mdp2){

            $pseudo = htmlspecialchars($pseudo);
            $email = htmlspecialchars($email);

            $this->pseudo = $pseudo;
            $this->email = $email;
            $this->mdp = $mdp;
            $this->mdp2 = $mdp2;
            $this->bdd = bdd();

        }

```

Ensuite nous allons faire des requetes SQL pour entrer les données dans la BDD et faire une session active.
Extrait de inscription.class.php :

```
1        public function enregistrement(){
2
3        $requete = $this->bdd->prepare('INSERT INTO membres(pseudo,
    email,mdp) VALUES(:pseudo,:email,:mdp)');
4        $requete->execute(array(
5            'pseudo'=>  $this->pseudo,
6            'email' => $this->email,
7            'mdp' => $this->mdp
8        ));
9
10       return 1;
11
12   }
13
14   public function session(){
15       $requete = $this->bdd->prepare('SELECT id FROM membres
    WHERE pseudo = :pseudo ');
16       $requete->execute(array('pseudo'=>  $this->pseudo));
17       $requete = $requete->fetch();
18       $_SESSION['id'] = $requete['id'];
19       $_SESSION['pseudo'] = $this->pseudo;
20
21       return 1;
22   }
23
```

4.2 La connexion et la deconnexion

Pour la connexion le procéder est similaire, la seule différence réside dans le fait que nous entrons pas de valeur dans la base de données mais nous faisons qu'une vérification.
Extrait de connexion.class.php :

```
1        public function verif(){
2
3        $requete = $this->bdd->prepare('SELECT * FROM membres WHERE
     pseudo = :pseudo');
4           $requete->execute(array('pseudo'=> $this->pseudo));
5           $reponse = $requete->fetch();
6           if($reponse){
7
8               if($this->mdp == $reponse['mdp']){
9                   return 'ok';
10              }
11              else {
12                  $erreur = 'Le mot de passe est incorrect';
13                  return $erreur;
14              }
15
16
17          }
18          else {
19              $erreur = 'Le pseudo est inexistant';
20              return $erreur;
21           }
22
23
24      }
25
```

Pour la déconnexion on détruit juste la session active.

4.3 Ajouter un sujet et une discussion

De la même façon un formulaire est utilisé et les données sont transportées.
Extrait de addSujet.class.php :

```
1        public function insert(){
2        echo $this->categorie;
3        $requete = $this->bdd->prepare('INSERT INTO sujet(name,
    categorie) VALUES(:name,:categorie)');
4        $requete->execute(array(
5            'name'=> $this->name,
6            'categorie'=>  $this->categorie
7        ));
8
9        $requete2 = $this->bdd->prepare('INSERT INTO postSujet(
    propri,contenu,date,sujet) VALUES(:propri,:contenu,NOW(),:sujet
    )');
10       $requete2->execute(array('propri'=>$_SESSION['id'],'contenu
    '=>  $this->sujet,'sujet'=>  $this->name));
11
12       return 1;
13   }
14
```

Même procédé pour la discussion.

4.4 Modifier et supprimer

Pour supprimer on va utiliser un bouton et la requête tiens en quelques lignes.
delete.php : [4]

```
1        $requete = $bdd->prepare('DELETE FROM postSujet WHERE id =
    '.$id);
2
```

Pour modifier le procéder et le même, on utilise une textarea pour transporter la modification.

4.5 Administrateur et bannis

Ce point de code n'est malheureusement pas otpimale, nous avons juste mis une visibilité totale pour l'administrateur en jouant avec les if. [5]

4. On a pas utilisé de fonction/class par manque de temps
5. Il aurait fallu utiliser un modéle vue-controleur

Conclusion