Rapport / GALERIE VIRTUEL
Projet PROGRAMMATION WEB

JULLION Francois / 2018-2019

Sommaire

Objectif....4
Fonctionnalités....4
1-Système de compte....4
2-Menu de navigation....5
3-Œuvres....5
4-Artistes....5
5-Upload de photos....6
6-Système de Commentaires....6

Objectif

L'objectif de projet était de concevoir entièrement un site web dynamique, contenant une base de données, à l'aide de différents Langages de programmation (HTML, CSS et PHP principalement). J'ai donc choisis comme sujet : une galerie d'art virtuel, ou chaque personne peut mettre en ligne ses créations, voir celles des autres et rechercher des œuvres ou artistes en particulier. Ce principe s'apparente à un réseau social.

Fonctionnalités

1-Système de compte

Pour le système de compte, j'ai tout d'abord crée une table SQL d'utilisateurs avec toutes les informations nécessaires pour représenter un compte d'utilisateur :

```
CREATE TABLE IF NOT EXISTS `user`
(
  `Username` varchar(16) NOT NULL,
  `Password` varchar(16) NOT NULL,
  `email` varchar(50) NOT NULL,
  `Birth` date NOT NULL,
  PRIMARY KEY (`Username`)
)
```

On a donc bien un nom de compte qui sera aussi le nom d'utilisateur, un mot de passe , un email et une date de naissance.
Pour la connexion et l'inscription, on a chaque fois 2 pages php, une contenant le formulaire et l'autre utilisant les informations du formulaire.

- Connexion : Avant le formulaire, on a donc une vérification de variable de session, parce que si l'utilisateur est connecté, il ne peut pas se reconnecter. A la fin du formulaire, on a un script pour vérifier que les champs soit bien remplis, sinon un PopUp d'information s'ouvre en indiquant quels champs l'utilisateur doit remplir. Le script de connexion se connecte à la base de données, exécute 2 requêtes, contant le nombre de n-uplets contenant le mot de passe et le nom d'utilisateur. Si le nombre de n-uplets est différent de 0, donc qu'il y a une ligne dans la table qui a pour mot de passe et nom de compte les 2 champs mis dans le formulaire. Si la condition est valide, alors on crée une session, on sauvegarde des variables de sessions, et on redirige vers l'accueil. Sinon on redirige vers le formulaire avec un message d'erreur.
- Inscription : Le formulaire contient 4 champs qui correspondent aux 4 données souhaités à rentres dans notre table d'utilisateur. Il y a une vérification en JavaScript, de la validité des champs du formulaire. Le script d'inscription, vérifie donc que chaque champs (une requête par champ) est présent dans aucun n-uplets. Si c'est valide, alors on redirige vers la connexion ou l'utilisateur pourra se connecter, sinon on redirige vers le formulaire d'inscription en indiquant que les champs ne sont pas valide. Le service d'envoi de mail pour affirmer que l'inscription est mis en place, mais ne peut-être testé car j'utilise WAMP.

La page moncompte permet aussi de voir ses informations de compte. Une requete SQL est exécutée pour récupérer la ligne de la table ou le nom d'utilisateur est le même que la variable de session de l'utilisateur. On affiche ensuite toute la ligne de la table dans un tableau. Un point

important est la vérification que l'utilisateur est connecte, sinon il sera redirigé vers la page de connexion. Enfin, la page déconnexion permet de se connecter en détruisant la session actuelle et redirigeant vers l'accueil

2-Menu de navigation

Le site possède une barre de navigation simple mais efficace. Avec les balises header et nav, on construit notre barre. Sur chaque page, on vérifiera que les variables de session sont initialisées (donc que l'utilisateur soit connecté) : Si oui, on affichera (en plus des onglets classiques) Moncompte, MesOeuvres et Deconnexion. Sinon on affichera Inscription et Connexion.

3-Œuvres

Pour les œuvre, on a donc besoin de créer une nouvelle table avec les informations importantes : le nom, une description et le nom du fichier image.

```
CREATE TABLE IF NOT EXISTS `oeuvre` (
  `Nom` varchar(20) NOT NULL,
  `Description` varchar(200) NOT NULL,
  `Path` varchar(20) NOT NULL,
  `User` varchar(16) NOT NULL,
   KEY  `User` (`User`)
)
```

Voici la commande utilisée pour la création de table.

La page galerie.php permet donc d'afficher toutes les œuvres de la base de données sous forme de grille suite à un aménagement en CSS, en cliquant sur l'image de l'œuvre, nous sommes redirigés vers la page de l'œuvre ou apparaissent le titre, l'auteur, l'image et la description. En dessous, il y a l'espace de commentaires.

4-Artistes

Les artistes seront associés aux utilisateurs, donc pas besoin de créer une nouvelle table d'artistes. La page artistes.php permet d'obtenir la liste des utilisateurs dans l'ordre alphabétique avec une requete php. Une barre de recherche est disponible afin de chercher une utilisateur précis. Chaque utilisateur affiché possèdent un lien qui mène sur leur page personnel, ou on peut voir toutes les œuvres de l'artiste en question. En cliquant sur l'œuvre, on arrive sur la page de l'œuvre en question avec toutes les informations vues précédemment.

5-Upload de photos

Le système de mise en ligne de photos fut le compliqué à mettre en place. Le but est donc d'enregistrer un fichier dans un dossier du site, mais aussi de rajouter l'oeuvre dans la base de données SQL (pour ensuite manipuler les données). Lors de l'upload, on renseigne donc le nom, la description et l'image. On redirige vers le script upload.php. Il utilise les fonctions PHP pour le traitement de fichier, avec ceux-ci, on vérifie donc que les champs : tout d'abord que le titre de l'œuvre ne soit pas déjà pris. Ensuite il y a la vérification de l'extension et de la taille du fichier. Enfin on utilise la fonction move_upload_files pour bouger le dossier img. On retient le nouveau nom du fichier, la description et l'auteur pour ensuite les rentrer de notre table SQL œuvre.

6-Système de Commentaires

Pour les commentaires, j'ai donc crée une troisième table avec le contenu du commentaire, l'œuvre sur laquelle se situe le commentaire, et l'utilisateur qui l'a upload.

```
CREATE TABLE IF NOT EXISTS `comment` (
  `content` varchar(100) NOT NULL,
  `Username` varchar(16) NOT NULL,
  `Nom` varchar(20) NOT NULL,
  KEY `Username` (`Username`),
  KEY `Nom` (`Nom`)
)
```

On utilise ensuite un formulaire pour récupérer les 3 données, et on les insert dans la base de données.