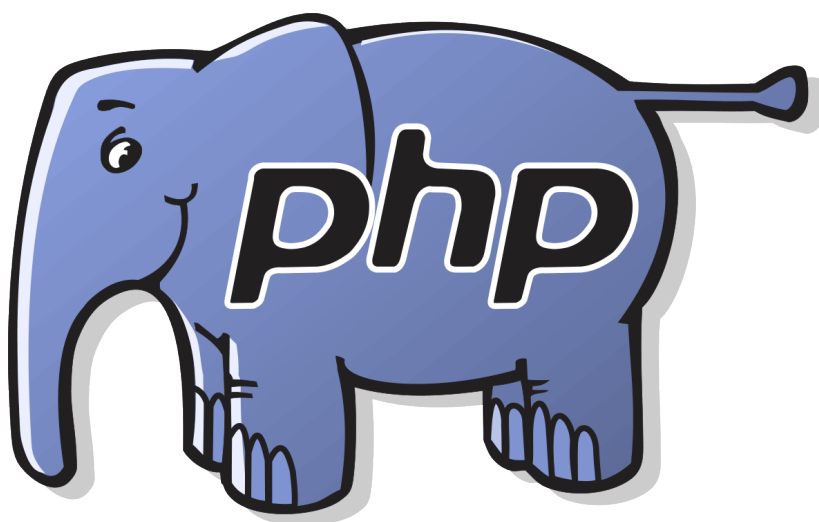


Rapport Programmation Web - PHP



Projet : réalisation d'un site Web de partage de fichiers

Cammarata Alessio
Bloc 2 (milieu de cycle) - Développement

Titulaires : monsieur Scopel & monsieur Malaise

Année académique : 2016 - 2017



Table des matières

INTRODUCTION	3
PRESENTATION DU PROJET	3
LA BASE DE DONNEES	4
PREMIERE TABLE - UTILISATEURS	4
DEUXIEME TABLE - FICHIERS	5
TROISIEME TABLE - MESSENGER	6
PLAN DE LA BASE DE DONNEES	7
LE SITE WEB	8
CONNEXION/INSCRIPTION	8
ACCUEIL	9
GESTION DES FICHIERS	9
PARAMETRES	10
NDU	10
LE MODE ADMINISTRATEUR	11
SCRIPTS	12
ARBRE DES SCRIPTS	12
EXPLICATION DES SCRIPTS	13
LES SCRIPTS PARTICULIERS	15
POINTS IMPORTANTS DU PROJET	17
POINTS FORTS	17
PROBLEMES RENCONTRES	17
CAHIER DE CHARGE DEMANDE	18
DEMANDES RESPECTEES	18
DEMANDES NON RESPECTEES	19
CONCLUSION	20

Introduction

Dans le cadre du cours de "Programmation Web - PHP", un projet a été demandé aux élèves. Ce projet est individuel et doit être fait avec les bonnes pratiques de programmation Web. L'utilisation de l'**HTML** et du **CSS** étant nécessaire afin d'avoir un site propre, le **JavaScript** peut également être utilisé pour rendre le site plus dynamique.

Dans mon cas, je me servirais de ces trois langages de programmation, mais également (et surtout) du **PHP**, nouveau pour moi.

Présentation du projet

Dans mon cas, le projet est simple : un site Web qui permet à un utilisateur de partager des fichiers (avec pour thème l'informatique en général) à d'autres utilisateurs du site, grâce à un système de connexion basé sur une adresse mail et un mot de passe.

Et donc, l'utilisateur lambda pourra :

- se créer un nouveau compte sur le site (obligatoire pour utiliser toutes les fonctionnalités);
- partager des fichiers dans certaines catégories afin d'être plus précis;
- télécharger les fichiers des autres utilisateurs;
- avertir l'administrateur d'un quelconque souci;
- profiter des autres fonctionnalités sur site.

Une partie pour l'**administrateur** a également été mise en place.

La base de données

La construction de la base de données ne s'est pas faite directement. J'avais d'abord ma base de données (appelée "**projetphp**") incluant 3 tables vides.

Première table - utilisateurs

La première table a été facile à compléter, c'est la table "**utilisateurs**". Facile à compléter car les informations qu'elle doit contenir sont logiques : elle va contenir les informations des utilisateurs.

Nom	Type
id (<i>clé primaire</i>)	int(11)
nom	varchar(255)
prenom	varchar(255)
photoprofil	varchar(255)
mail	varchar(255)
pass	varchar(255)

Tout d'abord le champ **id** de type **int**, qui sera auto-incrémenté. Un nouvel utilisateur, dès lors qu'il s'inscrit, aura un id unique.

Ensuite, les champs **nom**, **prenom**, et **mail** seront de type **varchar**. Purement du texte, ces informations s'enregistrent automatiquement lorsqu'un utilisateur s'inscrit ou modifie un de ces champs par la suite.

Le champ **photoprofil** est quant à lui de type **varchar**. En réalité, ce n'est pas la photo qui est enregistrée dans la base de données mais bien le **chemin vers cette photo** (par exemple : "ProfilPic/photo.jpg"). Cela permet de retrouver la photo facilement et également de pouvoir modifier ce chemin rapidement. Par défaut, la valeur de ce champ est "ProfilPic/vidé.png", chemin correspondant à une photo de profil par défaut.

Enfin, le champ **pass** qui est lui aussi de type **varchar**. Le mot de passe que l'utilisateur utilisera ne sera pas affiché en clair dans la base de données. Il sera **haché** par une fonction de PHP et sera ainsi remplacé par une chaîne de 60 caractères dans la base de données (cela signifie donc que l'administrateur ne pourra pas voir le mot de passe des utilisateurs en clair). Le but de cette manipulation est avant tout la protection des données.

Deuxième table - *fichiers*

La deuxième table "**fichiers**" a également été conçue logiquement : le principe du site Web étant le partage de fichiers, toutes les informations des fichiers devaient être présentes quelque part, et c'est désormais chose faite.

Nom	Type
id_fichier (<i>clé primaire</i>)	int(11)
id_user	varchar(255)
prenom_user	varchar(255)
nom_user	varchar(255)
photo_user	varchar(255)
nomfichier	varchar(255)
extension	varchar(255)
taille	int(20)
dateT	datetime
url	varchar(255)
commentaire	varchar(255)
categorie	varchar(255)

Le champ **id_fichier** qui est un nombre qui s'auto-incrémente lors du téléchargement d'un nouveau fichier.

Les champs **id_user**, **prenom_user**, **nom_user**, et **photo_user** reprennent les informations de l'utilisateur ayant téléchargé un fichier sur le site.

Les champs **nomfichier**, **extension**, et **taille** reprennent les informations du fichier téléchargé sur le site.

Le champ **dateT** reprend la date et l'heure auxquelles le fichier a été téléchargé sur le site, et **url** reprend le chemin menant au fichier.

Commentaire est le champ où sera placé le commentaire inscrit par l'utilisateur lors du téléchargement (optionnel).

Et enfin, **categorie** est la catégorie sélectionnée par l'utilisateur.

Troisième table - messenger

Cette dernière table s'est faite plus tard. Pas nécessaire au début, il s'avère qu'elle sera utilisée pour la gestion des messages adressés à l'administrateur par n'importe quel utilisateur.

Nom	Type
id_message (<i>clé primaire</i>)	int(11)
id_user	int(11)
prenom_user	varchar(255)
nom_user	varchar(255)
photo_user	varchar(255)
dateMessage	datetime
message	text

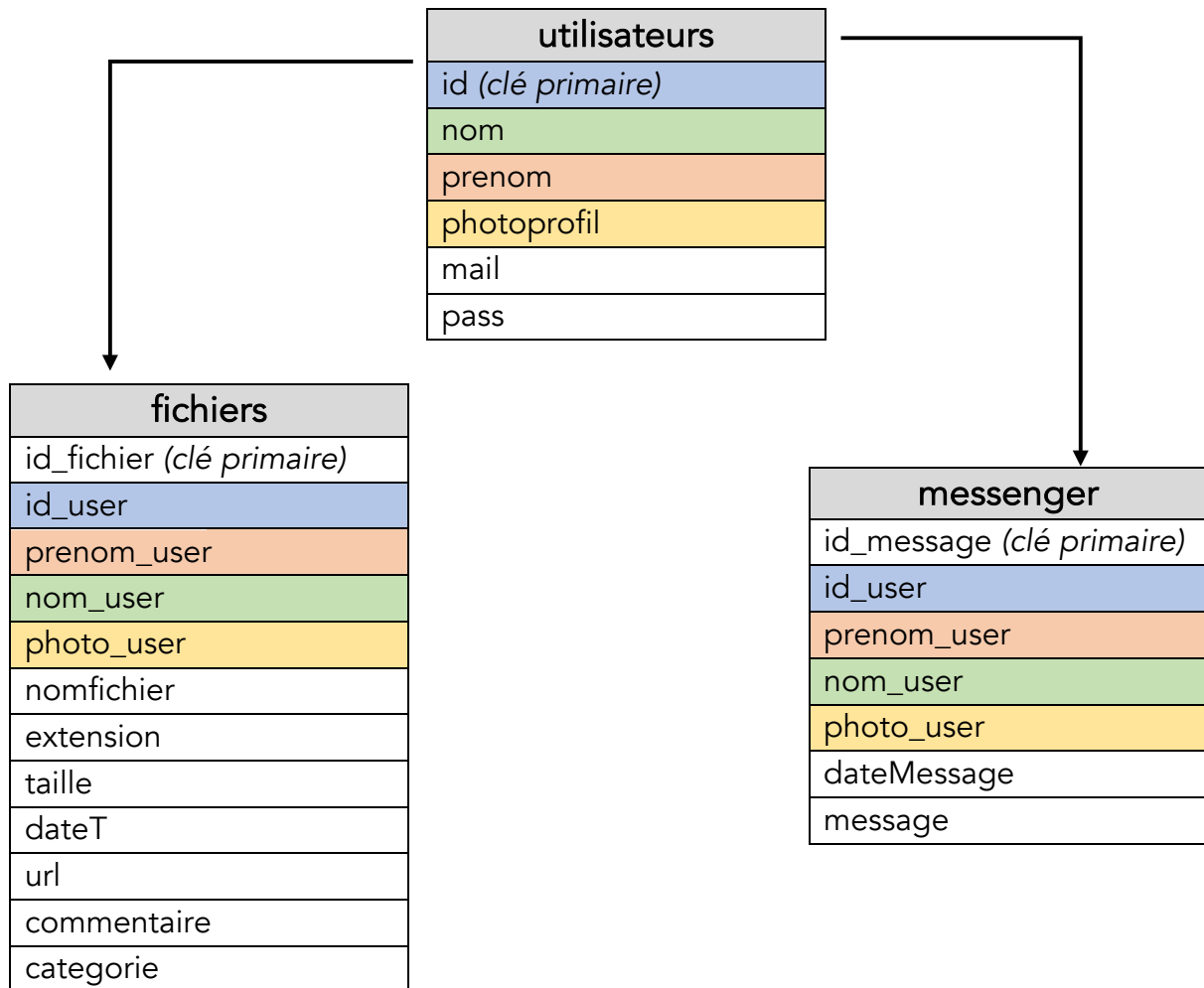
Le champ **id_message** est, comme pour les autres tables, le numéro spécifique à chaque nouvelle entrée. Il s'auto-incrémente.

Les **id_user**, **prenom_user**, **nom_user**, et **photo_user** reprennent les informations de l'utilisateur qui envoie le message à l'administrateur.

dateMessage est simplement la date complète à laquelle le message est envoyé.

Et enfin le champ **message** contient le contenu textuel du message.

Plan de la base de données



La table **utilisateurs** a été la première table créée dans la base de données. Elle est la table principale. Sans elle et son contenu, le site Web ne fonctionne simplement pas. En effet, elle contient les informations primordiales de chaque utilisateur. C'est pourquoi elle a été faite avant les deux autres.

La table **fichiers** nécessite des informations de la table **utilisateurs**. Elle stocke les informations de l'utilisateur qui télécharge un fichier. Elle a donc besoin de son nom et prénom, son id, et sa photo de profil. Le reste des champs est rempli en fonction du fichier téléchargé. Elle a été conçue après la table **utilisateurs**, étant donné qu'elle a besoin des informations de celle-ci pour être remplie parfaitement.


Enfin, la table **messenger** utilise des informations des utilisateurs également. Le reste des champs est rempli en fonction du message envoyé. Faite après les deux autres tables, car la fonction d'envoi de message est apparue tardivement.

Le site Web

Connexion/Inscription

S'inscrire | Se connecter

Site de partage/sauvegarde de fichiers informatiques en PHP



Connexion


Mail :

Mot de passe :

Pas de compte ? Pas de souci. [Inscrivez-vous ici !](#)

Sur ce site, vous pourrez partager (et accessoirement sauvegarder) des fichiers divers avec d'autres utilisateurs.
Attention de bien respecter les règles mentionnées lors de votre inscription.

Projet réalisé dans le cadre du cours de PHP, par Alessio Cammarata - HEH Campus Technique



Arrivé sur le site, on peut **se connecter** via le mail et le mot de passe. Interface simple, on peut facilement savoir quoi faire.

On peut également aller **s'inscrire** très simplement, à condition de respecter les règles imposées.

Formulaire d'inscription

Nous avons besoin de certaines coordonnées pour que votre inscription soit validée.

Nom

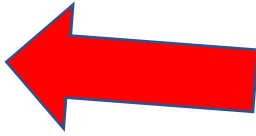
Prénom

Mail

Mot de passe

Confirmez votre mot de passe

Vous avez déjà un compte ? [Retour à l'accueil.](#)



Accueil



Dans l'accueil, on trouve 4 choses importantes :

- la liste des fichiers téléchargés via le site Web (avec la photo de profil, le nom de l'utilisateur ayant uploadé, la date ou temps écoulé, un commentaire éventuel, le nom du fichier, la taille en Mo et la catégorie sélectionnée);
- la **barre de menu**, présente sur chaque page du site, menant à l'accueil, à la page de gestion des fichiers, aux paramètres, à l'outil d'aide, et permet de se déconnecter;
- la **section de téléchargement**, un clic suffit pour faire apparaître un formulaire permettant de télécharger un fichier sur le site;
- la **liste des catégories**, menant vers les catégories spécifiques des fichiers.

Gestion des fichiers

Une fois sur la page de gestion des fichiers, on peut télécharger un fichier sur le site (de la même manière que dans la section de téléchargement ci-dessus). Il faut sélectionner un fichier en respectant les consignes mentionnées, choisir une catégorie, et mettre un commentaire optionnel.

Enfin, on peut avoir un aperçu des fichiers que nous avons téléchargé. On peut également les supprimer facilement grâce à la petite croix rouge sur le côté droit.

Sélectionnez un fichier à télécharger :

Choisir le fichier : aucun fichier sél.

Sélectionnez la catégorie de votre fichier ('Informatique' par défaut) :

Informatique

Laissez un commentaire (optionnel, 200 caractères max) :

Télécharger le fichier

...et le fichier fut !



Paramètres

Changer votre mot de passe
Confirmez votre nouveau mot de passe
Changer de mot de passe
Veillez à avoir un mot de passe de minimum 4 caractères, sans caractères spéciaux.

Entrez votre nouveau mail
Confirmez votre nouveau mail
Changer d'adresse mail
Veillez à avoir une adresse mail conforme (sans caractères spéciaux).

Entrez votre nouveau nom
Confirmez votre nouveau nom
Changer de nom
Veillez à avoir un nom sans caractères spéciaux.

Entrez votre nouveau prénom
Confirmez votre nouveau prénom
Changer de prénom
Veillez à avoir un prénom sans caractères spéciaux.

Sélectionnez une nouvelle photo de profil :
Choisir le fichier aucun fichier séli.
Changer de photo de profil
Veillez à sélectionner une photo de moins de 2 Mo.
L'image sera affichée en petite taille, prenez donc une photo adaptée.

Supprimer mon compte

Sur la page des paramètres, l'utilisateur peut modifier son **adresse mail**, son **nom**, son **prénom**, et son **mot de passe**. Par défaut, l'utilisateur n'a pas de **photo de profil** personnelle, il peut cependant la changer via cette page.

Il peut également supprimer son compte, il sera alors demandé d'inscrire son adresse mail et son mot de passe afin de vérifier ces informations, et être sûr qu'il veut supprimer son compte.

NDU

Alessio, faites donc un petit message à l'administrateur.

ATTENTION : veillez à écrire un message uniquement en cas de souci, de besoin, voir d'incompréhension. Il sera immédiatement envoyé à l'administrateur.
N'envoyez pas de message ridicule ou inutile, sous peine de sanction.
Votre message doit faire maximum 400 caractères, et ne pas contenir de caractères spéciaux.

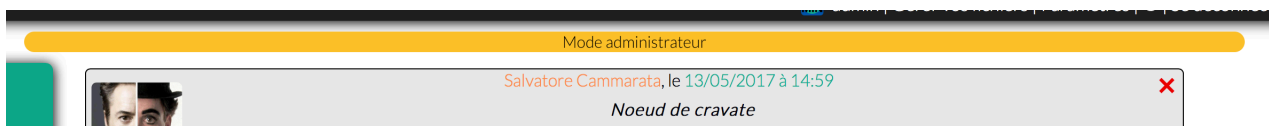
Laissez un message (400 caractères max) :

Envoyez votre message

La page NDU (Nouvelles Des Utilisateurs) est dédiée à un souci que l'utilisateur pourrait rencontrer.

Un utilisateur lambda pourrait avoir un souci quelconque avec le site, et pourra via cette page faire un message à l'**administrateur** du site.

Le mode administrateur







Une fois connecté avec le compte administrateur, une barre jaune se place au-dessus des autres éléments afin d'avertir que l'on est bien en **mode administrateur**.

De plus, une petite croix rouge apparaît sur le côté droit de chaque fichier. En effet, l'administrateur a la possibilité de **supprimer les fichiers de n'importe quel utilisateur**.

Sur la page des paramètres, le bouton "Supprimer le compte" a disparu. Je trouvais assez inutile de placer ce bouton en mode administrateur.

Un administrateur a la possibilité d'accéder à toutes les fonctionnalités du site Web, ainsi qu'aux bases de données. Donc, si un administrateur veut supprimer son compte, il peut le faire en supprimant des éléments directement dans la base de données.

Messages des utilisateurs	
	Salvatore Cammarata, le 2017-04-28 14:38:08 <i>Encore un test lol jpp</i>
	Alessio Cammarata, le 2017-04-28 14:37:08 <i>Ok dans ici c'est un nouveau test</i>
	Antonella Sanna, le 2017-04-17 22:19:50 <i>Ok un autre phrase !</i>
	Alessio Cammarata, le 2017-04-17 17:23:43 <i>Voila un troisieme message pour voir</i>
	Alessio Cammarata, le 2017-04-17 17:11:45 <i>Ok donc ici on retest hein juste ainsi tkt yolo</i>
	Alessio Cammarata, le 2017-04-17 17:11:14 <i>Petit message de test.</i>

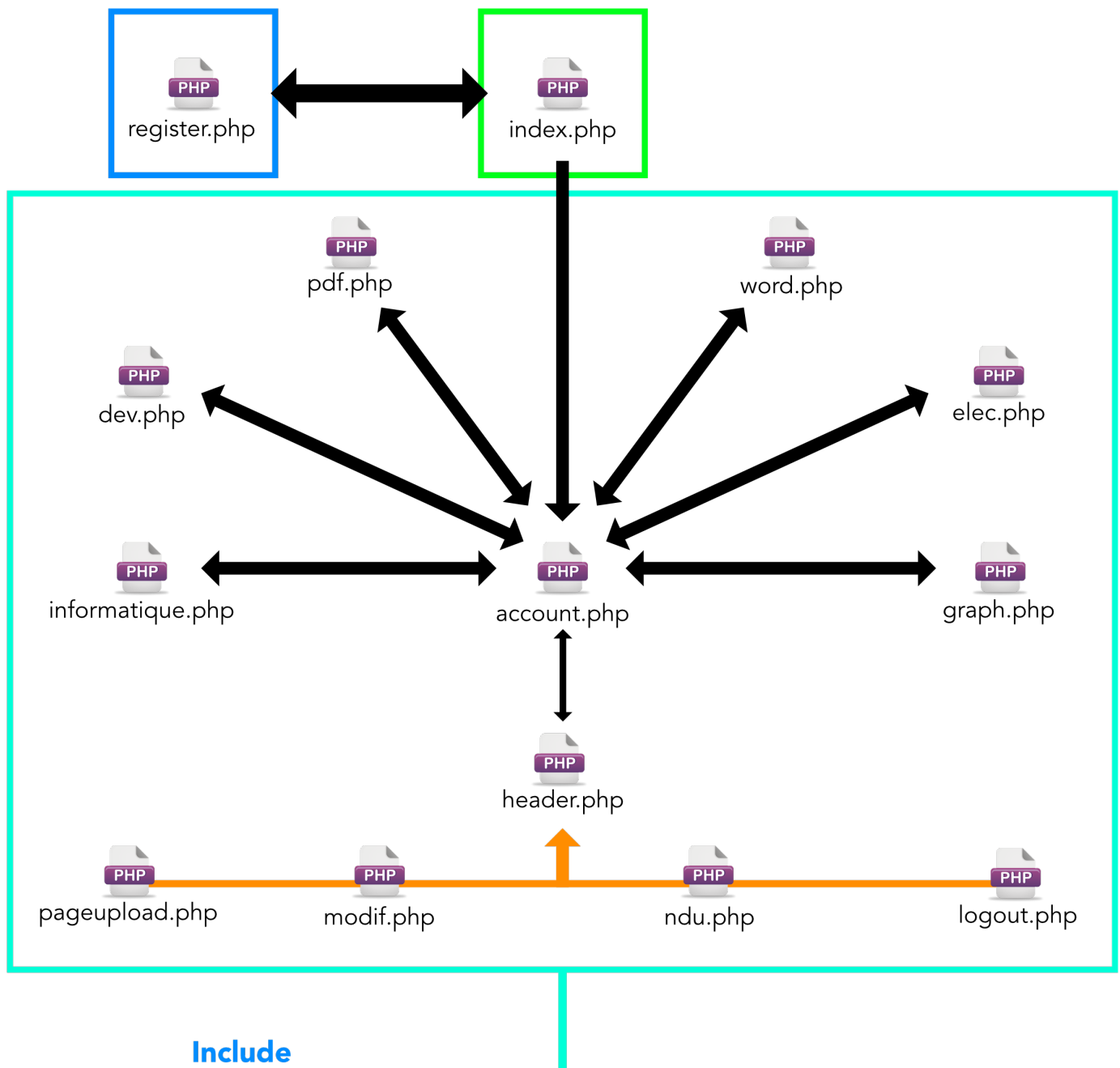
Infos des utilisateurs	
 Alessio Cammarata - ID n°16 - alessio@hotmail.com	×
 admin istrateur - ID n°19 - admin@admin.com	×
 Antonella Sanna - ID n°22 - lella92@hotmail.com	×
 Salvatore Cammarata - ID n°23 - salva.cammarata@hotmail.com	×

Sur la page NDU, l'administrateur n'envoie pas de message mais visualise ceux qu'il a reçu (avec le nom du destinataire et la date).

De plus, il a accès aux informations de tous les utilisateurs et peut supprimer qui il veut.

Scripts

Arbre des scripts



Explication des scripts

index.php : script de base sur lequel tous les utilisateurs arrivent, impossible de passer à côté. Sur cette page, on peut se connecter ou décider de s'inscrire en allant sur register.php.

Lors de la connexion, une vérification est faite dans la base de données pour être sûre que l'adresse mail existe et que le mot de passe inscrit est bien celui correspondant à cette dernière. Si cela n'est pas le cas, un message apparaît stipulant que la connexion est impossible. Mais si tout va bien, la connexion s'effectue grâce à ce morceau de code (entre autres) :

```
session_start();
$_SESSION['auth'] = $user;
header('Location: account.php');
```

La session démarre et est gardée en mémoire, et l'utilisateur est redirigé vers account.php (l'accueil).

Et, grâce à ce bout de code :

```
if(isset($_SESSION['auth'])){
    header('Location: account.php');
    exit();
}
```

Si l'utilisateur est déjà connecté mais entre manuellement dans la barre de recherche "index.php", il est redirigé vers account.php.

register.php : script d'inscription d'un nouvel utilisateur. Il vérifie que le nouveau mail entré n'est pas déjà existant dans la base de données et permet d'enregistrer les nouvelles informations de l'utilisateur via ce code :

```
$req = $pdo->prepare("INSERT INTO utilisateurs SET nom = ?, prenom = ?,
mail = ?, pass = ?");
$password = password_hash($_POST['pass'], PASSWORD_BCRYPT);
$req->execute([$_POST['nom'], $_POST['prenom'], $_POST['mail'],
$password]);
```

L'algorithme BCRYPT : il va créer une clé de hachage standard en utilisant l'identifiant "\$2y\$", et le résultat obtenu sera toujours une chaîne de 60 caractères.

Si l'inscription ne se fait pas correctement (donc si l'utilisateur ne respecte pas certaines règles), un message d'erreur apparaît stipulant l'erreur commise.

account.php : ce script est en réalité l'accueil sur lequel l'utilisateur va se retrouver une fois connecté. Il comprend tous les fichiers téléchargés sur le site avec les informations s'y rapportant, la barre de menu, la sélection des catégories et le système de téléchargement.

Les éléments sont affichés du plus récent au plus ancien grâce à une boucle dans le code :

```
<?php
$bdd = new PDO('mysql:host=localhost;dbname=projetphp;charset=utf8',
'root', 'root');
$reponse = $bdd->query('SELECT * FROM fichiers ORDER BY id_fichier DESC');

while($donnees = $reponse->fetch()){?>
<?php }
$reponse->closeCursor();
?>
```

Les scripts *dev.php*, *elec.php*, *graph.php*, *informatique.php*, *pdf.php*, et *word.php* contiennent un code très semblable à celui de *account.php*, ils sont simplement modifiés pour ne faire apparaître que certains fichiers en fonction de la catégorie souhaitée.

pageupload.php : ce script est utilisé pour télécharger des fichiers sur le site ou pour en supprimer. L'utilisateur doit sélectionner un fichier avec certaines convenances (taille, extension) et peut laisser un commentaire. Les fichiers sont placés dans le dossier "Uploads".

La suppression des fichiers est plus simple, on récupère l'id spécifique au fichier et on l'enlève de la base de données comme ceci :

```
$url = $_SERVER['REQUEST_URI'];
$parse = parse_url($url);
if(!empty($parse['query'])){
    $pdo->exec('DELETE FROM fichiers WHERE id_fichier =
    "'. $parse['query']. '");
    ?>
    <div class="uploadok">
        Fichier supprimé
    </div>
    <?php
}
```

modif.php : ce script permet de modifier les informations enregistrées dans la base de données, comme son nom, prénom, mail et mot de passe. La possibilité d'ajouter une photo de profil est également disponible.

ndu.php : script permettant à l'utilisateur d'envoyer un message à l'administrateur. L'utilisateur écrit une simple phrase en expliquant le souci et l'envoie directement. Cette phrase sera stockée dans une base de données.

L'administrateur, lui, pourra voir sur cette même page les messages des utilisateurs et même pouvoir supprimer ces utilisateurs.

Ce petit bout de code ci-dessous permet d'afficher des éléments seulement en fonction de l'id de l'utilisateur :

```
<?php if($_SESSION['auth']->id == '19'){ ... ?>
```

Si l'id de l'utilisateur est 19 (id de l'administrateur), alors il affichera un certain contenu invisible pour tous les autres.

Pourquoi utiliser l'id comme paramètre ?

Très simplement, imaginons que quelqu'un veuille se créer un nouveau compte avec l'adresse mail de l'administrateur. Il ne pourra pas étant donné qu'elle est déjà utilisée, mais s'il arrive à le supprimer et à vouloir le recréer, il ne sera pas pris en compte !

En effet, l'id s'incrémente tout seul, il aura donc une valeur supérieure à 19 et n'aura pas accès aux informations d'administrateur.

Les scripts particuliers

db.php : présent sur toutes les pages du site, il permet la connexion à la base de données.

```
$pdo = new PDO('mysql:host=localhost;dbname=projetphp', 'root', 'root');
```

fonctions.php : présent également sur toutes les pages du site, il accueille plusieurs fonctions utilisées constamment et sur chaque page (comme la fonction "logged_only").

deletefilesript.php : script permettant de supprimer les fichiers, présent sur les pages nécessaires.

uploadscript.php : permet de télécharger un fichier, présent sur les pages autorisant cette fonction.

Grâce au code suivant, l'utilisateur aura certaines restrictions quant au fichier à télécharger :

```
$date = date('Y/m/d H:i:s');  
$commentaire = $_POST['commentaire'];
```

```

$file_name = $_FILES['file']['name'];
$file_extension = strrchr($file_name, ".");
$extensions_autorisees = array('.docx', '.DOCX', '.docm', '.DOCM', '.dotx',
'.DOTX', '.doc', '.DOC', '.ppt', '.PPT', '.pptx', '.PPTX', '.pdf', '.PDF',
'.txt', '.TXT', '.html', '.HTML', '.css', '.CSS', '.js', '.JS',
'.afdesign', '.AFDESIGN', '.png', '.PNG', '.jpg', '.JPG', '.jpeg', '.JPEG',
'.gif', '.GIF', '.zip', '.ZIP', '.rar', '.RAR');
$file_taille = $_FILES['file']['size'];
$file_tmp_name = $_FILES['file']['tmp_name'];
$file_dest = 'Uploads/'.$file_name;

if(in_array($file_extension, $extensions_autorisees)){
    if($file_taille < 25000000){
        if(!preg_match('/[/#&$%ø_°|*$€%f+∞÷≠±•¿#%/' ,
$_POST['commentaire'])) {
            if (move_uploaded_file($file_tmp_name, $file_dest)) {
                $req = $pdo->prepare('INSERT INTO fichiers SET id_user = ?,
prenom_user = ?, nom_user = ?, photo_user = ?, nomfichier = ?, extension =
?, taille = ?, dateT = ?, url = ?, commentaire = ?, categorie = ?');

```

Le code va d'abord prendre en compte la date et le commentaire (s'il existe). Il va ensuite décomposer le nom complet du fichier de sorte à enregistrer le nom et séparer ce nom de l'extension.

Grâce à la liste des extensions autorisées, le code vérifiera que l'extension du fichier téléchargé fait bien partie de cette liste. Si c'est le cas, le code récupérera la taille du fichier (en octet), et si le fichier pèse moins de 25 Mo, il enregistrera ce dernier dans le dossier prévu à cet effet.

Ce code est également repris lors de l'upload de la photo de profil, le dossier final est simplement différent et la taille maximum est de 2 Mo.

barremenu.php : présent sur la plupart des pages, regroupe les différentes catégories.

header.php : offre les différentes fonctionnalités du site (comme la déconnexion).

footer.php : affiche un footer en bas de page avec certaines informations.

logout.php : déconnecte l'utilisateur.

Points importants du projet

Points forts

Faire ce projet m'a aidé. Il m'a apporté une nouvelle vision de la sécurité Web et des interactions possibles. J'ai constaté que PHP est un langage puissant et que les possibilités de projet sont nombreuses, on peut vraiment faire tout et n'importe quoi grâce à ça.

Etudier, comprendre, et utiliser le PHP m'a conforté dans l'idée que je souhaitais devenir développeur Web. Et je peux maintenant dire que je connais un minimum le PHP (bien que j'estime n'avoir gratter que la surface), et que je souhaite approfondir mes connaissances là-dedans.

Ce projet n'est pas parfait, mes connaissances n'étant pas absolues, mais je l'améliorerais encore de moi-même car le PHP et autres langages Web m'intéressent beaucoup.

Problèmes rencontrés

La gestion des bases de données a été revue, car c'est une matière que je n'utilise pas souvent. Mais le challenge n'a pas été difficile, une simple pique de rappel et j'ai compris le fonctionnement de MySQL.

Concernant la suppression d'un fichier/utilisateur via le site, cela m'a donné du fil à retordre. Je ne comprenais pas la logique de programmation derrière, ni comment aboutir au résultat escompté. Cependant, avec quelques recherches et l'aide du professeur, j'ai trouvé une solution qui sera finalement la solution finale, grâce à ce petit bout de code :

```
$url = $_SERVER['REQUEST_URI'];
$parse = parse_url($url);
if(!empty($parse['query'])){
    $pdo->exec('DELETE FROM fichiers WHERE id_fichier =
    "'. $parse['query']. '");
    ?>
    <?php
}
```

L'id du fichier est renvoyé dans l'URL, et la fonction "\$parse" se charge de récupérer uniquement ce qui m'intéresse. Une interaction avec la base de données permet de supprimer uniquement ce que je souhaite.

Enfin, l'affichage de la date ne s'est pas fait comme je le souhaitais.

Finalement :

```
$datetime = date_create($donnees['dateT']);  
$date = date_format($datetime, "d/m/Y");  
$time = date_format($datetime, "H:i");
```

La date enregistrée dans ma base de données est coupée en deux parties, et je peux enfin affichées les deux parties comme bon me semble.

Cahier de charge demandé

Lorsque le projet a été demandé, un cahier de charge nous a été imposé pour que le projet puisse effectuer certaines choses bien spécifiques.

Demandes respectées

- utilisation de l'HTML, CSS, JavaScript et PHP ;
- le site comprend une base de données comprenant 3 tables ;
- gestion des utilisateurs (CRUD) ;
- un utilisateur est représenté par un nom, prénom, adresse mail et mot de passe (j'ai même ajouté une photo de profil optionnelle) ;
- tout utilisateur peut s'enregistrer avec un profil de type "utilisateur" et a le droit d'exploiter les fonctionnalités du site ;
- l'utilisateur se connecte via son login (adresse mail et mot de passe de minimum 4 caractères) ;
- l'accès complet à la base de données n'est pas permis à tous les utilisateurs ;
- un administrateur a été créé par mes soins (mail : admin@admin.com + mdp : admin), il a accès à toute la base de données.

En tant qu'utilisateur :

- se dé/connecter (session) ;
- modifier les informations privées ;
- ajouter/modifier/effacer les informations dans les deux tables ;
- supprimer le compte ainsi que les fichiers uploadés ;
- communiquer avec l'administrateur.

En tant qu'administrateur :

- accès aux informations des clients ;
- supprimer un client ;
- ajouter du contenu au site ;
- éditer du contenu.

Sécurité :

- l'accès au site est sécurisé ;
- protection de la base de données ;
- les données entrées sont vérifiées ;
- un utilisateur non logué n'a pas accès aux données du site.

Demandes non respectées

- bannissement d'un client par l'administrateur.

Conclusion

Ce projet n'était pas chose simple. Je n'avais que très peu de notion de PHP et ce langage paraissait compliqué. La construction du site en HTML/CSS/JavaScript n'a pas été compliquée, le langage Web étant un langage que j'apprécie particulièrement, et j'aimerais me lancer dans le développement Web plus tard.

Cependant, l'enseignement que j'ai reçu pour PHP m'a été d'une grande aide. J'ai reçu les ressources nécessaires à la compréhension de ce langage et à son développement. Je dois en revanche reconnaître que ces ressources étaient *insuffisantes*.

Ce projet est un travail de recherche avant tout. Le PHP étant un langage, il faut d'abord le comprendre parfaitement avant de pouvoir l'utiliser. Et je dois dire que ce ne fut pas chose simple.

Mais, grâce aux nouvelles ressources trouvées sur Internet ou dans la documentation PHP, ou même grâce aux réponses des professeurs, le projet a été mené à bien.

Ce projet n'est pas parfait, et ma connaissance en PHP non plus. Cela dit, mon intérêt pour le PHP a été éveillé et je travaillerais très certainement encore dessus afin de le comprendre encore mieux.

PHP est un langage riche, qui permet de faire beaucoup de choses diverses, et ce projet m'a permis d'affirmer quelque chose dont j'étais déjà sûr :

le développement informatique est une chose complexe, mais grâce aux bonnes pratiques de programmation et à l'enseignement que j'ai reçu, je sais que le développement de nouveaux logiciels et des nouvelles technologies m'attire de plus en plus, et je veux en faire mon métier.

L'informatique et la technologie évoluent très vite, autant comprendre ces nouvelles choses qui arriveront dans le futur plutôt que d'étudier des langages qui ne sont plus utilisés, mais j'affirme que le PHP n'est pas prêt d'être abandonné et que ce langage persistera encore pendant longtemps.

Logo principal et autres images conçus par mes soins. Je me suis également aidé de deux images : l'une venant d'Apple et l'autre de Affinity Designer.