Développement web, projet PHP IRT2

**Forum de discussion**



2018-2019

Deryck Olivier

### Table des matières

[Table des matières 2](#_Toc11173041)

[1. Introduction 3](#_Toc11173042)

[2. Présentation générale du projet 3](#_Toc11173043)

[2.1. Rappel des fonctionnalités à implémenter 3](#_Toc11173044)

[2.2. Vue générale du site 4](#_Toc11173045)

[3. Base de données 6](#_Toc11173046)

[3.1. Conception de la base de données 6](#_Toc11173047)

[3.2. Les différents champs 7](#_Toc11173048)

[Table catégories : 7](#_Toc11173049)

[Table topics 7](#_Toc11173050)

[Table posts 7](#_Toc11173051)

[Table users 7](#_Toc11173052)

[Table messages 7](#_Toc11173053)

[4. Site 8](#_Toc11173054)

[4.1. Les scripts et pages 8](#_Toc11173055)

[4.2. Connexion à la base de données 10](#_Toc11173056)

[Utilisation de PDO 10](#_Toc11173057)

[4.3. Sécurités mises en place 10](#_Toc11173058)

[5. Problèmes rencontrés 11](#_Toc11173059)

[6. Améliorations et conclusions 12](#_Toc11173060)

[6.1. Possibilités d’amélioration 12](#_Toc11173061)

[6.2. Conclusion 12](#_Toc11173062)

[7. Bibliographie 12](#_Toc11173063)

[8. Annexes 12](#_Toc11173064)

[Identifiants administrateur 12](#_Toc11173065)

# Introduction

Dans le cadre du projet de PHP, j’ai opté pour un forum de discussion sans sujet précis, sur lequel plusieurs utilisateurs peuvent s’inscrire, se connecter, poster, s’échanger des messages, …

Il a donc fallu programmer toutes les fonctionnalités nécessaires au fonctionnement et à la sécurité de ce type de site.

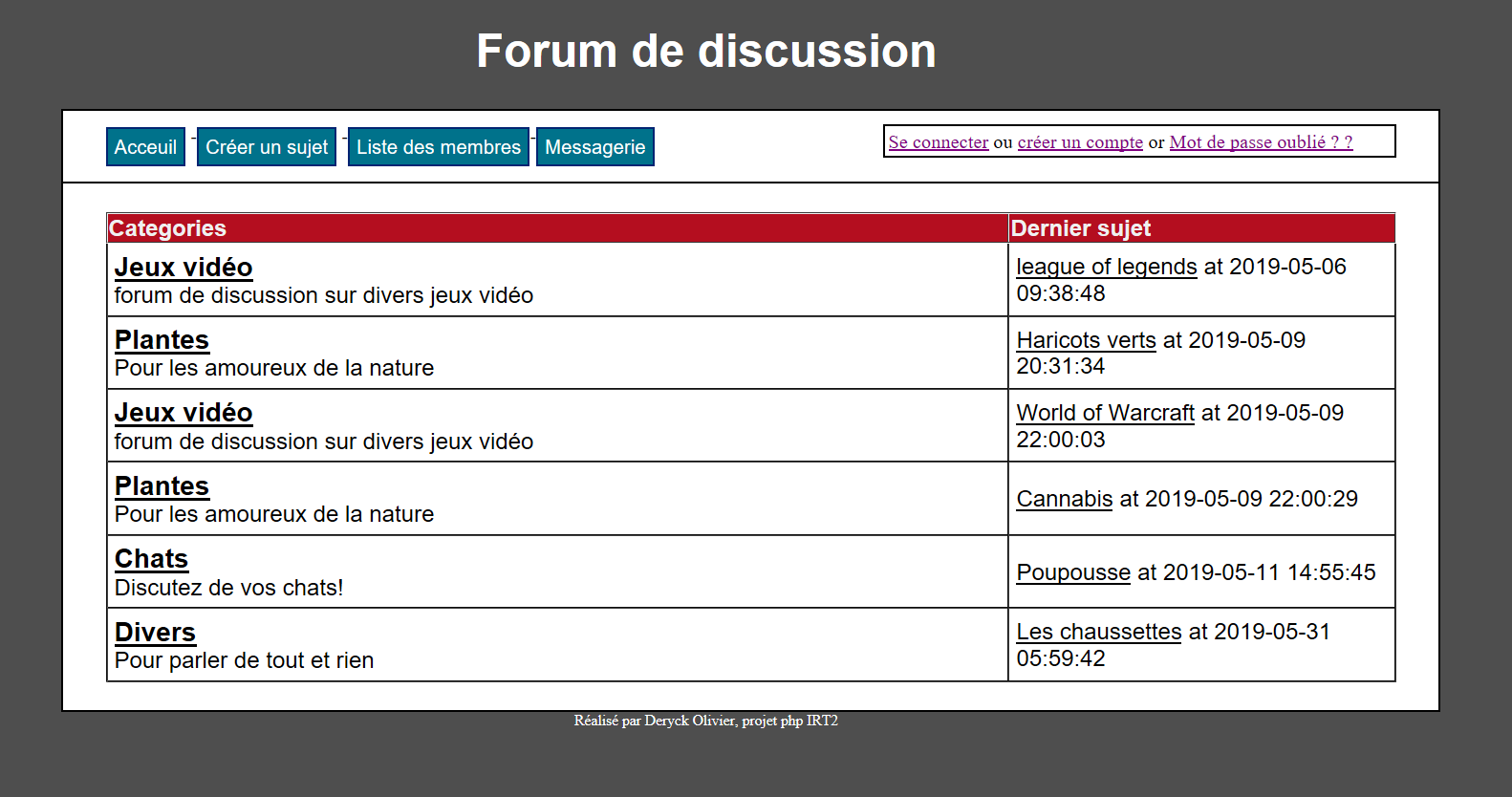
Dans une première partie, je vais brièvement présenter le projet en rappelant les fonctionnalités attendues et vais expliquer la conception et les champs de la base de données, après cela, présenter les différentes pages et scripts de mon projet, la connexion à la base de données et les sécurités mises en place. Et enfin aborder les problèmes rencontrés, avant de terminer sur les possibles améliorations.

# Présentation générale du projet

## Rappel des fonctionnalités à implémenter

|  |  |  |
| --- | --- | --- |
| En tant que… | Je veux… | Afin de… |
| Administrateur | Posséder une partie administrateur | Gérer les utilisateurs, et les différentes parties du site. |
| Administrateur | Accéder à des modifications sur chaque contenu | Modifier ou supprimer des postes ou réponses de n’importe quel utilisateur |
| Utilisateur | Pouvoir m’inscrire | Pouvoir être membre de la communauté |
| Utilisateur | Gérer mon profil | Modifier mes informations |
| Utilisateur | Pouvoir créer un sujet | Lancer une conversation sur un sujet/poser une question |
| Utilisateur | Répondre à un sujet déjà existant | Communiquer avec les autres utilisateurs |
| Utilisateur | Modifier mes contenus | Modifier ou supprimer les postes ou messages dont je suis l’auteur |
| Utilisateur | Avoir accès à ma messagerie | Envoyer et consulter ses messages privés. |
| Visiteur | Consulter certains sujets | S’informer sur les sujets du forum |

## Vue générale du site



Pour un utilisateur non connecté :

il peut consulter les sujets du forum, mais les onglets « créer un sujet », «  liste des membres », et « messagerie » l’enverront sur une page lui signalant qu’il n’est pas connecté, l’invitant à se créer un compte ou se connecter pour consulter l’entièreté du site.

Un utilisateur non connecté peut aussi récupérer son mot de passe via l’onglet « mot de passe oublié », qui le dirigera vers une fenêtre demander des informations afin de vérifier son identité.



Ensuite, quand un utilisateur est connecté, il peut :

-créer un sujet en choisissant la catégorie dans laquelle il sera publié

-consulter la liste des membres

-accéder à sa messagerie, dans laquelle il peut décider d’afficher ses conversations, ses messages non-lus, ses messages envoyés ainsi que la totalité de ses messages reçus

-ainsi que modifier ses postes / réponses et son profil (haut à droite).



L’administrateur a accès à un panneau dans lequel il peut :

-Créer un nouvelle catégorie (un utilisateur courant peut uniquement poster dans une catégorie existante, et pas en créer une)

-Supprimer une catégorie (et donc les postes qui en dépendent)

-Modifier les informations d’un utilisateur

-Supprimer définitivement un utilisateur

-Modifier les postes de n’importe quel utilisateur afin d’en modérer le contenu

# Base de données

## Conception de la base de données

Schéma MCD de la base de données :

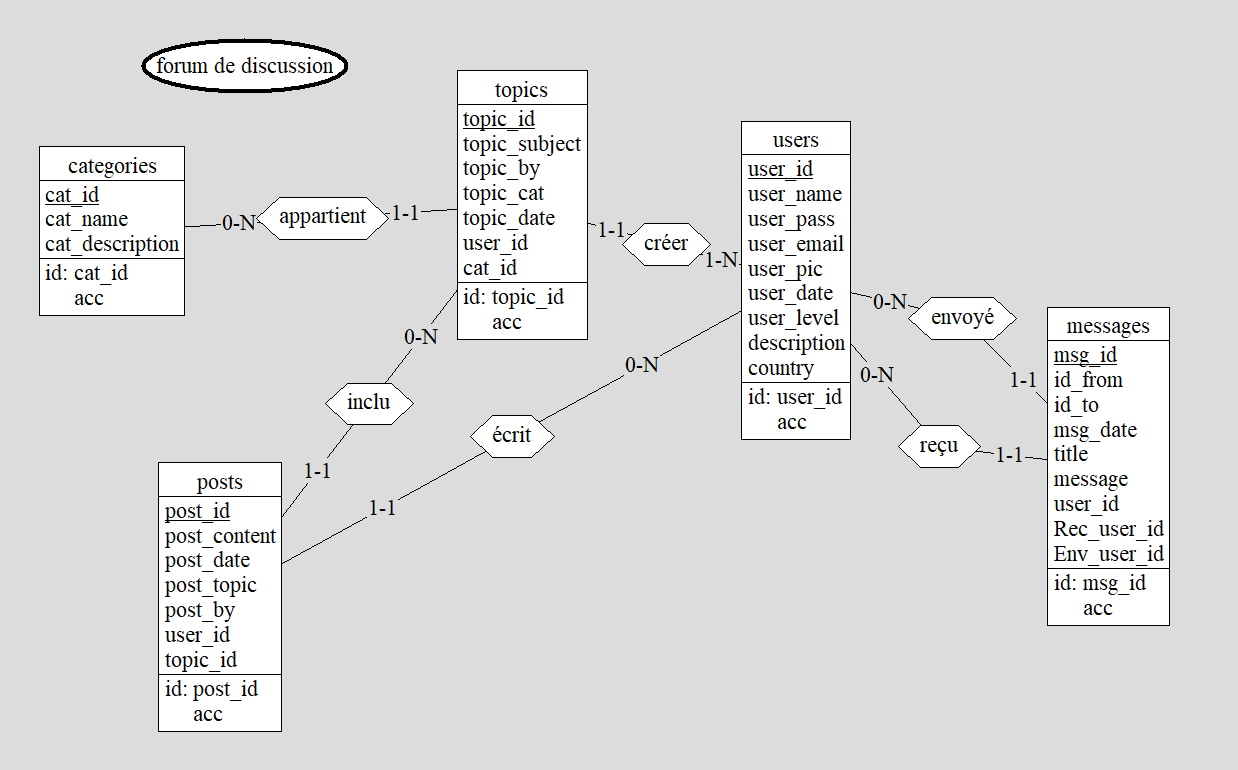
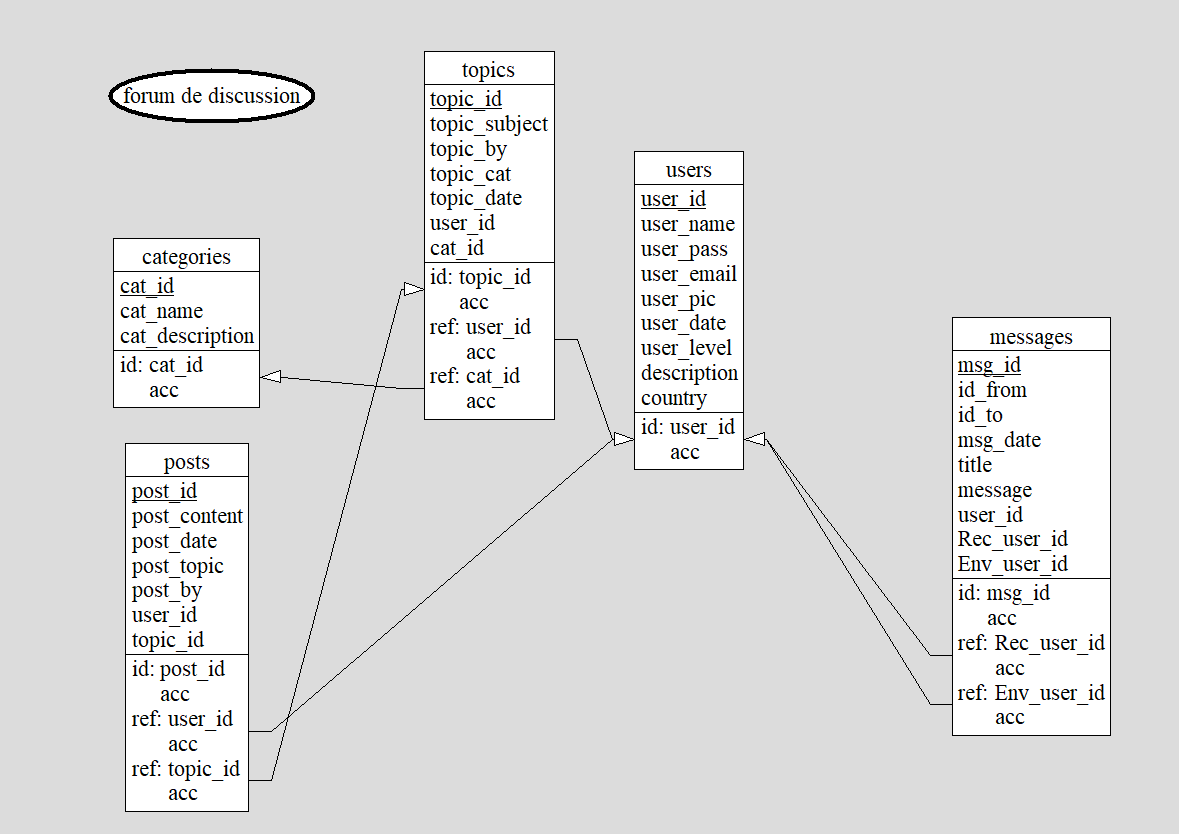


Schéma MLD de la base de données :



## Les différents champs

### Table catégories :

* **cat\_id** est l’identifiant des catégories, il est clé primaire et unique
* **cat\_name** est le nom de la catégorie
* **cat\_description** est la description de la catégorie

### Table topics

* **topic\_id** : identifiant du topic, clé primaire et unique
* **topic\_subject** : descriptif du topic
* **topic\_by** : prend la valeur de l’id de l’auteur du topic
* **topic\_cat** : prend la valeur de l’id de la catégorie du topic
* **topic\_date** : date de création du topic

### Table posts

* **post\_id** : identifiant du poste, clé primaire et unique
* **post\_content** : contenu du poste (commentaire utilisateur)
* **post\_date :** date du poste
* **post\_topic** : prend la valeur de l’id du topic auquel appartient le poste
* **post\_by** : prend la valeur de l’id de l’auteur du poste

### Table users

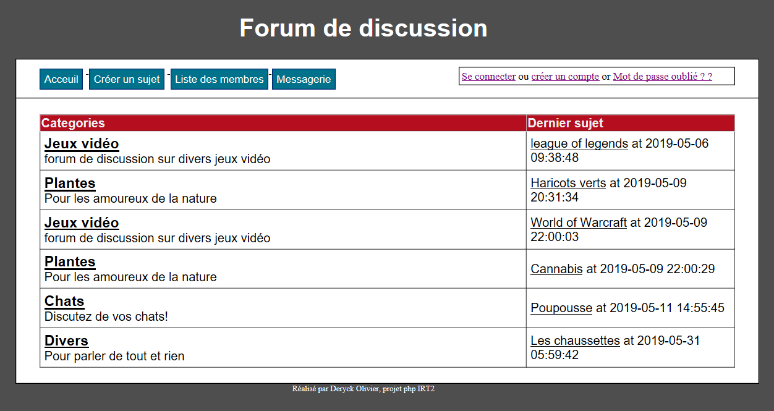
* **user\_id:** identifiant de l’utilisateur, clé primaire et unique
* **user\_name** : nom d’utilisateur , unique
* **user\_pass** : mot de passe, haché
* **user\_email** : email
* **profil\_pic** : photo de profile (non utilisée voir section « problèmes rencontrés)
* **user\_date** : date d’inscription de l’utilisateur
* **user\_level** : niveau d’autorisation de l’utilisateur, 0 pour les membres et 1 pour admin
* **description** : description de l’utilisateur
* **country**: pays de l’utilisateur

### Table messages

* **msg\_id:** identifiant du message, clé primaire et unique
* **id\_from:** id de l’utilisateur destinateur
* **id\_to**: id de l’utilisateur destinataire
* **msg\_date** : date d’envoi du message
* **title**: titre du message
* **message**: contenu du message
* **opened** : état d’ouverture du message, 0 = non ouvert et 1 = ouvert

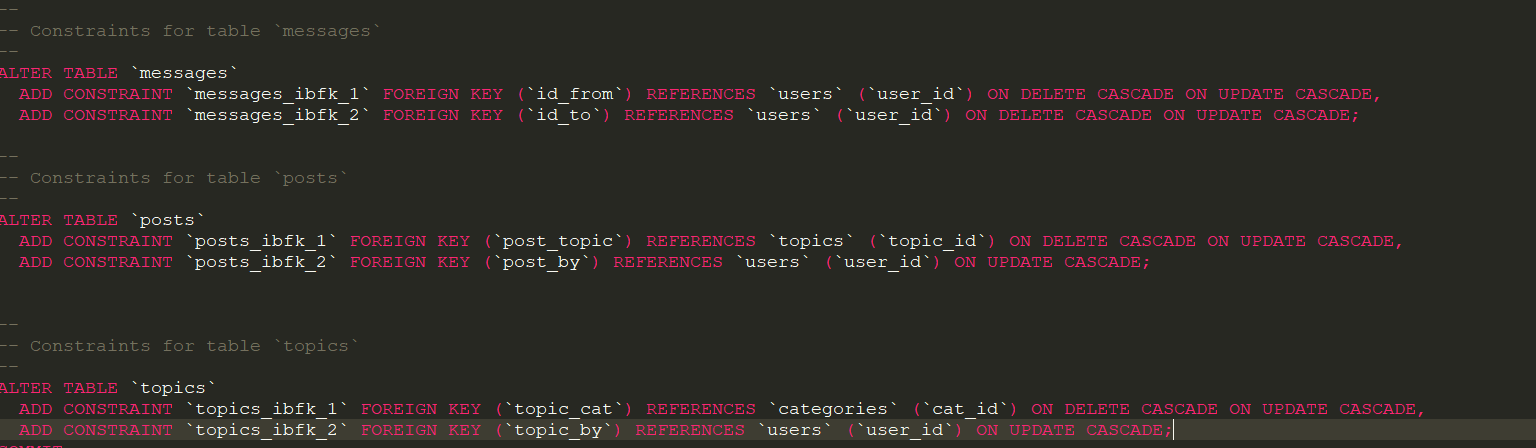
# Site

## Les scripts et pages

* **connect.php :** C’est le script qui établit la connexion à la base de données, il est à inclure au début des autres scripts nécessitant une connexion à la base de données.
* **header.php** : Script qui met l’ensemble du site en page, il comprend le titre du site, la barre de navigation, ainsi que le barre d’utilisateur, dont l’affichage varie selon qu’un utilisateur est connecté ou non. Il est à inclure pour les pages que l’on peut visiter, non nécessaire pour les scripts sans vue
* **footer.php** : Script postant juste le footer de la page. Il est à inclure en fin de script des pages visibles
* **index.php**

Vue d’accueil du site ; le script inclut le header.php et connect.php, ensuite le script se contente d’afficher dans un tableau les différentes catégories de la base de données

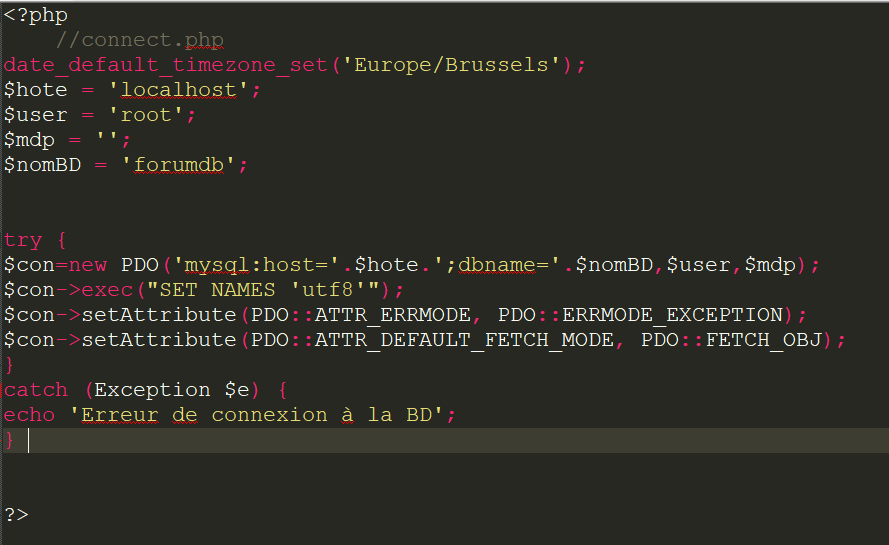
* **category.php** : ce script affiche les différents topics dans une catégorie
* **topic.php** : ce script affiche les différents messages sur un certain topic
* **create\_cat.php** : fenêtre qui permet de créer de nouvelles catégories. Disponible uniquement pour l’administrateur, via une vérification de variables de sessions (être connecté et avoir le niveau requis pour y accéder)
* **create\_topic.php** : fenêtre permettant de créer un nouveau topic dans une catégorie selectionnée. L’utilisateur doit être connecté, la vérification se fait à l’aide de la variable de session signed\_in
* **modif.php** : permet de modifier ou supprimer un commentaire posté ; l’utilisateur doit en être l’auteur, ou alors il doit être modérateur, les vérifications se font à l’aide des variables de session user\_id qui doit correspondre à l’id\_from du poste, ou si le niveau d’accès de l’utilisateur est supérieur à un (donc administrateur)
* **editprof.php** : script affichant la page de modification du profil personnel, vérification à laide de la variable de session user\_id, car un utilisateur ne doit pouvoir modifier que son profil
* **forget.php** : script offrant la possibilité aux utilisateurs de réinitialiser un mot de passe, le nom d’utilisateur et l’email sont requis, si les champs correspondent, une nouvelle fenêtre propose de modifier le mot de passe
* **function.php** : script sans visuel, sont définis une fonction d’affichage de pays (liste déroulante) et une vérification des champs pour les inscriptions et modifications. Ce script sert en cas de réutilisation des différentes fonctions définies
* **logout.php** : déconnecte l’utilisateur en détruisant sa session
* **member\_list.php** : affiche la liste des membres, l’utilisateur doit être connecté, la vérification se fait à l’aide de la variable de session « signed\_in »
* **messagerie.php** : affiche la messagerie, elle est uniquement accessible à un membre connecté
* **delete\_msg.php** : permet de supprimer un message de sa messagerie, requiert d’en être l’auteur ou le destinataire
* **myChat.php** : affiches les conversations d’un utilisateur avec les autres
* **read.php** : affiche le contenu d’un message, requiert d’être connecté, et d’en être soit l’auteur soit le destinataire
* **sendmsg.php** : envoi d’un message, l’utilisateur peut choisir un utilisateur (sauf lui-même) à qui envoyer le message, requiert d’être connecté
* **signin.php** : page de connexion, demande le nom d’utilisateur et le mot de passe, vérifie la correspondance lors de la validation
* **signup.php** : script d’inscription, affiche les champs à remplir, effectue les vérifications et enregistre les données du nouvel utilisateur dans la base de données



Clés étrangères entres les tables, quand un utilisateur est supprimé, ses contenus également.

## Connexion à la base de données

### Utilisation de PDO



Pourquoi PDO (Php Data Object)?

-Facilité des fonctions

-Encapsule de manière native les requêtes préparées, transactions, requêtes standards,…

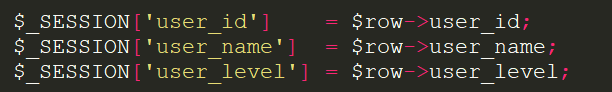
-Orienté objet

-Fonctions identiques peu importe le serveur de base de données (uniquement modifier la configuration de connexion)

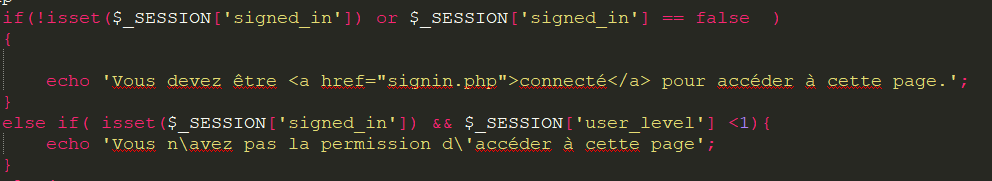
-utilisé avec les nouveaux outils

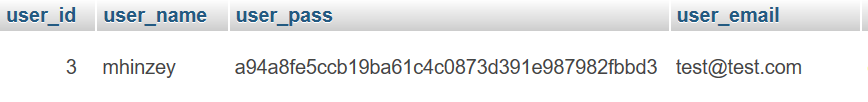
## Sécurités mises en place

Afin de sécuriser le site, plusieurs actions sont nécessaires :

* Variables de sessions :

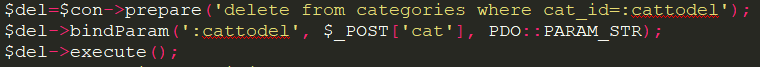
-Elles permettent de garder en mémoire des infos sur l’utilisateur de manière globale, afin de les réutiliser dans tous les autres scripts.

Permet aussi des vérifications avant d’exécuter un script, vérifier que l’id utilisateur correspond ou que le niveau de permission soit suffisant 



* Hachage de mot de passe (via sha1()) :

-Permet d’enregistrer des données de manières cryptées, de cette façon il est impossible pour un utilisateur de récupérer un mot de passe dans la base de donnée

* Requêtes préparées afin d’éviter les injections SQL 
* Failles xss, filtrer les symboles avec la fonction htmlentities lors de l’affichage d’un texte dans un echo pour éviter qu’un utilisateur puisse modifier l’affichage html des pages 
* Pour éviter les échecs d’insertions avec erreur, usage des fonctions BeginTransaction(), Rollback(), commit()
* Vérifications des champs lors de l’inscription ou des modifications des infos, un pseudo ne contient que des lettres et des chiffres, l’email un @, les mots de passe doivent correspondre, et un champ obligatoire doit être non nul
* Test des paramètres de l’URL, par exemple pour modifier un commentaire, l’id du commentaire est passé en paramètre dans l’URL , dans ce cas je teste la correspondance entre l’id de l’auteur du commentaire et l’id du membre connecté afin de savoir si il a le droit ou non de modifier le poste.

# Problèmes rencontrés

Au début du projet, je n’avais pas utilisé PDO pour la connexion et l’utilisation de la base de données, mes requêtes n’étaient pas préparées, et je n’avais pas protégé les echo d’attaques XSS et ai donc dû réécrire une bonne partie de mon projet en milieu de course, toutefois ce problème a été solutionné.

J’avais comme objectif de pouvoir afficher des images de profile pour les utilisateurs, que cela soit dans la liste des utilisateurs ou l’afficher à coté de leurs postes et commentaires. Je n’ai malheureusement pas réussi à gérer la sauvegarde et l’utilisation des photos depuis la base de données, j’ai donc abandonné l’idée pour me concentrer sur le reste.

Je n’ai pas exploité l’objet, ayant largement avancé dans mon projet sans utiliser des objets PHP, je n’ai plus trouvé cela nécessaire de tout modifier afin de les rajouter.

# Améliorations et conclusions

## Possibilités d’amélioration

* Le menu de changement des informations personnelles est perfectible, l’utilisateur doit rentrer un mot de passe pour valider les changements, ce n’est donc pas optimale pour l’utilisateur.
* La sécurité du reset de mot de passe peut laisser à désirer, il suffit de connaitre le pseudo et l’email de l’utilisateur (toutefois son adresse n’est affichée nulle part, il n’a donc qu’à ne pas la communiquer). L’idéal pour ce genre de cas serait d’envoyer une clé de reset par mail, ou un lien qui le redirige vers la page de remise à zéro du mot de passe, toutefois cela demande une configuration du serveur local qui ne pouvait se faire dans le cadre de ce projet.
* Ajout d’une fonction de recherche parmi les topics, les commentaires, ou les messages.
* Ajout de tris dans l’affichage des postes ou sujets (par date par exemple)
* Ajout de popup de confirmation lors de la sauvegarde, la suppression ou modifications de certaines informations

## Conclusion

Finalement, nous avons une version de base de l’application. La majorité des fonctionnalités de base ont été implémentées et fonctionnent correctement mais il reste quelques améliorations à faire pour aboutir véritablement à une version utilisable et qui pourrait être mise en ligne.

Ce premier projet en PHP m’aura permis de mieux comprendre et d’appliquer les cours de base de données et de programmation en PHP afin de produire un premier projet fonctionnel pour plusieurs utilisateurs.

# Bibliographie

<https://stackoverflow.com/>

Antoine Malaise, Notes théoriques de programmation PHP 2018-2019

<https://www.php.net/>

<http://php.net/manual/fr/book.mysql.php/>

<https://openclassrooms.com/fr>

[www.lephpfacile.com](http://www.lephpfacile.com)

fmaz.developpez.com

# Annexes

### Identifiants administrateur

Username : admin Password : admin