PROJET

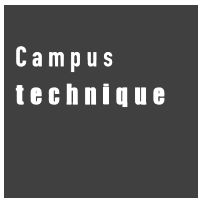
RÉALISATION D'UN SITE EN PHP

2ÈME BACHELIER EN INFORMATIQUE

Programmation web

Auteur :
Alexandre DUCOBU

Enseignants :
Antoine MALAISE
Fabrice SCOPEL



Année académique 2016 - 2017

PROJET

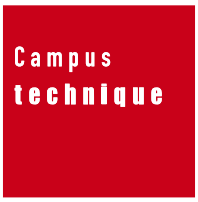
RÉALISATION D'UN SITE EN PHP

2ÈME BACHELIER EN INFORMATIQUE

Programmation web

Auteur :
Alexandre DUCOBU

Enseignants :
Antoine MALAISE
Fabrice SCOPEL



Année académique 2016 - 2017

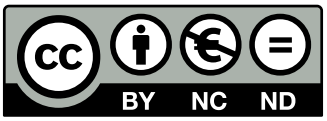


Table des matières

1 Présentation du projet

1.1 Introduction

Dans le cadre du cours de **Programmation web**, il nous a été demandé de réaliser un site dynamique en PHP.
Dans le cadre d'un site dynamique, les données disponibles sur le site proviennent d'une base de données, ici, composée de trois tables.

1.2 Énoncé

Le site doit comprendre une base de données composée de trois tables.

L'une d'elles est prévue pour la gestion des utilisateurs, représentés par, au minimum, un nom, un prénom, une adresse mail et un mot de passe.
L'adresse mail doit servir pour la connexion au site, et le mot de passe a une longueur minimale de 4 caractères.

Du côté des fonctionnalités, les utilisateurs doivent pouvoir se (dé)connecter, modifier leurs données, supprimer leur compte et contacter l'administrateur.
L'**administrateur**, utilisateur créé par nos soins, a accès à toutes les données de la base de données, peut ajouter et modifier du contenu, réinitialiser le mot de passe des utilisateurs et peut aussi les bannir.

1.3 Description

Ce site proposera aux visiteurs de s'inscrire afin de créer une liste privée de médicaments et ce, d'après une liste de médicaments *non-exhaustive*.
Cette liste sera établie par l'administrateur selon son envie et les demandes envoyées par les utilisateurs.

Chaque médicament comprendra : sa posologie, les effets néfastes, les contre-indications ainsi qu'un lien vers sa notice.

Les personnes n'ayant pas de compte auront accès à la liste des médicaments ne contenant que leur nom et leur posologie.

1.4 Langages

Les deux langages principaux sont donc, vu l'énoncé, le **PHP** et **MySQL**.
Ceux-ci sont, bien entendu, accompagnés du HTML, CSS ainsi que du JavaScript *(et jQuery)* pour la création du site.

FIGURE 1 – Logo de PHP

FIGURE 2 – Logo de MySQL

1.5 Outils

D'après le cours de bases de données du premier quadrimestre et le système d'exploitation utilisé, l'outil choisi pour le serveur web ainsi que la base de données est **MAMP**.
En ce qui concerne l'éditeur de texte, c'est **Atom** qui a été retenu pour sa simplicité ainsi que pour la familiarité que je ressent envers lui.

FIGURE 3 – Logo de MAMP

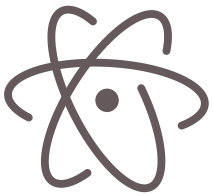


FIGURE 4 – Logo de Atom

2 Base de données

2.1 Création de la base de données

La base de données est composée de trois tables contenant toutes les données utiles au bon fonctionnement du site.
Il y a la table **Users** *pour tout ce qui concerne les utilisateurs*, la table **Medicines**, *qui concerne les médicaments* et le table **Reserves** *qui contient la réserve de chaque utilisateur*.

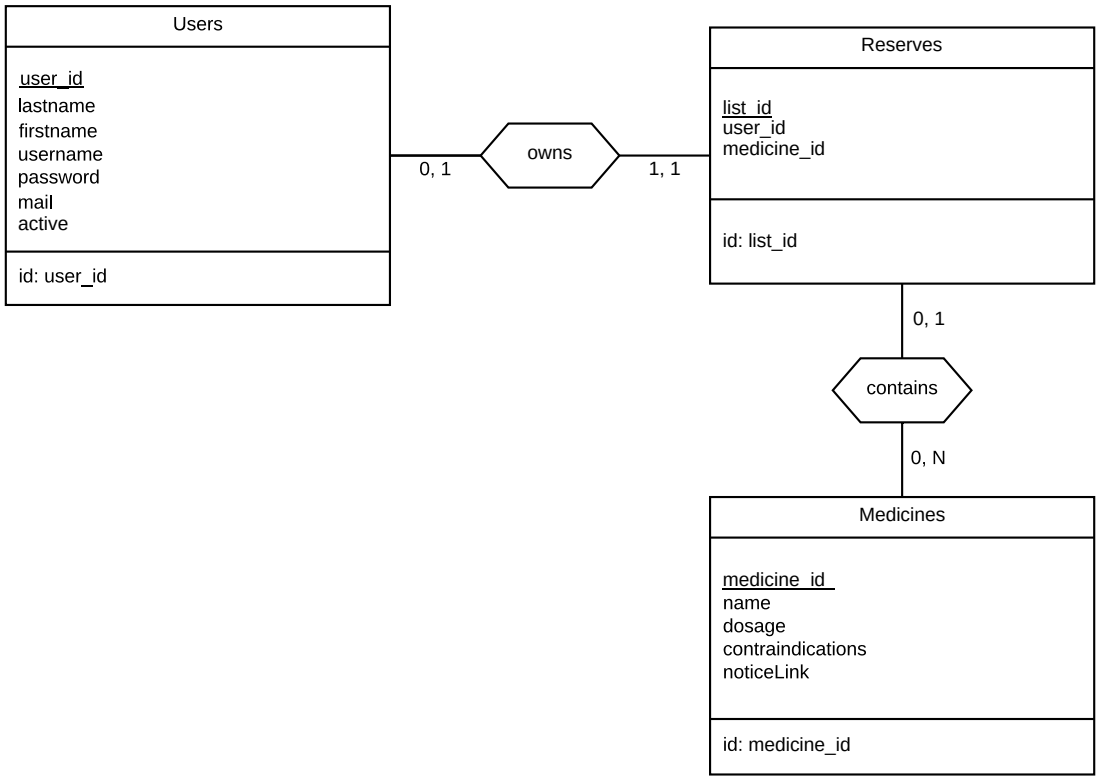


FIGURE 5 – Schéma conceptuel de la base de données

2.2 Vue détaillée

2.2.1 Table Medicines

Cette table contient les informations de chaque médicament.

- **medicine_id** est l'identifiant du médicament.
  Il sert de clé primaire de la table et est auto-incrémenté à partir de 1.

- **name** est le nom du médicament.
  Ce champ, en plus de contenir le médicament, identifie son type : comprimé, comprimé effervescent, gélule, etc.

- **dosage** contient le ou les différents dosages possibles *(en mg ou en g)*.

- **contraindications** comprend deux à trois lignes de contre-indications du médicament.

- **noticeLink** contient l'adresse vers la notice en ligne, à télécharger.

Remarque : l'image représentant le médicament est sauvegardée dans un dossier spécifique et est liée à son identifiant.
Elle ne se trouve donc pas dans la base de données.

2.2.2 Table Reserves

Cette table lie les deux autres : la réserve de médicaments lie chaque utilisateur à ses médicaments.

- **list_id** est l'identifiant unique de la réserve.
  Il est auto-incrémenté à partir de un et sert de clé primaire.

- **user_id** est l'identifiant de l'utilisateur.
  C'est une clé étrangère.

- **medicine_id** identifie les médicaments.
  C'est aussi une clé étrangère.

2.2.3 Table Users

C'est la table contenant les données de chaque utilisateur ainsi que l'état de leur compte.

- **user_id** est l'identifiant unique du médicament.
  Il sert de clé primaire de la table et est auto-incrémenté à partir de un.

- **lastname** est le nom de famille de l'utilisateur.

- **firstname** est le prénom de l'utilisateur.

- **username** est le nom d'utilisateur unique choisi lors de l'inscription.

- **password** est le mot de passe de l'utilisateur.
  Il a une taille minimale de 4 caractères et est haché[1] *(et salé)* à l'aide de SHA512.

- **mail** contient l'adresse mail de l'utilisateur.
  Ce champ est unique, vu qu'il sert à la connexion de l'utilisateur.

- **active** donne l'état du compte de l'utilisateur :
  - **0** indique que le compte est inactif.
    Cela signifie que le compte a été supprimé ou que l'administrateur a banni l'utilisateur.
  - **1** indique que le compte est actif *(par défaut)*.

1. `https://fr.wikipedia.org/wiki/Fonction_de_hachage_cryptographique`

3 Site

3.1 Arborescence

Le site a été divisé en neuf dossiers pour plus de clarté.

- Trois d'entre-eux ne sont pas liés au PHP :
  - **img :** contient toutes les images du site.
  - **js :** contient les différents scripts en JavaScript ainsi que jQuery.
  - **style :** contient les différents fichiers contenant le CSS du site.
- Ensuite, viennent les six autres qui contiennent le PHP :
  - **actions :** contient les scripts php liés aux actions telles que la (dés)activation d'un compte, l'ajout d'un médicament ainsi que la réinitialisation d'un mot de passe.
  - **connexion :** contient les scripts php liés à la connexion à la base de données ainsi que la (dé)connexion au site.
  - **DB :** contient les scripts MySQL qui créent et initialisent la base de données.
  - **forms :** contient les différents formulaires (avec leur version de vérification) tels que celui de l'ajout/modification d'un médicament, de connexion, etc.
  - **include :** contient les scripts php formant une partie des pages comme le pied de page et les différentes tables de la base de données.
  - **insidePages :** les pages internes du site comme la page principale, celle de changement de mot de passe, etc.
- Pour finir, l'index et la page d'erreur de connexion à la base de données se trouvent à la racine du site.

3.2 Arborescence détaillée

Voici maintenant chaque dossier détaillé.

3.2.1 Dossiers non liés au PHP

Le dossier *js* contient trois fichiers.

Le premier, ***jquery-3.1.1.min.js***, contient toutes les fonctions de jQuery.

Le second, ***jquery.cslide.js***, est une librairie permettant d'insérer facilement un slider [2].
Il est utilisé pour afficher le contenu des tables 5 par 5 grâce à l'utilisation de boutons « Suivant » et « Précédent ».

Le troisième et dernier, ***script.js***,contient trois fonctions :

1. hideUsers() : qui cache l'onglet *Utilisateurs* lorsqu'un utilisateur autre que l'administrateur est connecté.
2. setCurrentTab() : qui sélectionne l'onglet courant.
3. updateFooterBg() : qui change la couleur de fond du footer sur les autres navigateurs que Safari, qui permet d'avoir un fond flouté.

3.2.2 Dossiers PHP

Le dossier **actions** est constitué de quatre fichiers PHP.

— ***activateUser.php***, qui permet de désactiver le compte d'un utilisateur ou de le réactiver.
Tous les utilisateurs, à l'exception de l'administrateur, peuvent désactiver leur compte.
L'administrateur, lui, a le droit d'activer ou de bannir *(désactiver)* le compte de n'importe quel utilisateur sauf le sien.

— ***addMedToList.php*** : ce fichier reçoit donc les données d'un nouveau médicament qu'il ajoute à la base de données.

— ***changeDBData.php*** : il a un fonctionnement proche du précédent, sauf qu'il modifie les données de l'utilisateur (nom de famille, prénom et adresse mail).

— ***reinitPWD.php*** : appelé par l'administrateur, ce fichier réinitialise le mot de passe de l'utilisateur sélectionné en lui donnant une valeur par défaut *(EpcCM98)*.

2. `http://callmenick.com/post/responsive-content-slider`

Le dossier **connexion** en possède douze.

- ***connexion.php*** contient le script de connexion à la base de données.
- ***dbInfos.php*** contient les données de connexion à la base de données : nom d'hôte, nom de la base, nom d'utilisateur, mot de passe et numéro de port.
- ***login.php*** est le script appelé pour vérifier le login lors d'une tentative de connexion au site.
- ***loginPage.php*** : c'est la page de connexion au site. C'est elle qui appelle le script précédent.
- ***logout.php*** : comme son nom l'indique, il est utilisé pour déconnecter l'utilisateur du site.
- ***signUp.php*** : son fonctionnement proche de celui de login.php lui permet d'enregistrer un nouvel utilisateur.
- ***signuPage.php*** : c'est la page d'inscription au site. C'est elle qui appelle le script précédent.
- Les cinq suivant servent à vérifier les données des formulaires à l'aide de regex [3].
  Il est aussi parfois nécessaire de vérifier des données dans la base de données comme le nom d'un médicament lors de son ajout afin de ne pas avoir de doublons.

Le dossier **BD** est le plus simple, car il ne contient que deux fichiers SQL :

- ***CreateDB.sql*** crée la base de données ainsi que les trois tables la composant.
- ***InitDB.sql*** initialise la base de données en ajoutant trois utilisateurs dont l'administrateur ; ajoute sept médicaments différents ; et remplit la réserve des deux utilisateurs créés.

3. Expression régulière (en anglais: **Reg**ular **ex**pression)

Le dossier **forms** contient tous les formulaires du site.

En résumé, chaque formulaire existe en deux formes :

- Le formulaire vierge qui, une fois soumis, est vérifié par l'un des scripts de vérification du dossier connexion.

- Le formulaire de vérification qui, si les données sont acceptées, appelle le script adéquat.
  Dans le cas contraire, le formulaire est affiché avec les bonnes données et/ou les messages d'erreurs pour chaque champ.

Le dossier **include** contient 8 fichiers.

— ***footer.php*** contient le footer avec ou sans le moyen de contacter l'administrateur selon que l'on soit connecté ou non et que l'on soit l'administrateur ou non.

— Les deux fichiers de barre de navigation :
  — ***navBarConnect.php*** permet d'afficher la liste des médicaments et les liens de connexion et d'inscription au site.

  — ***navBarReserve.php*** : affichée lorsque l'utilisateur est connecté, elle permet de naviguer entre les différentes tables de la base.

— ***medsPopUp.php*** permet d'afficher les données des médicaments dans un popup.

— ***account.php*** affiche les données de l'utilisateur et lui permet de les changer ou de supprimer son compte.

— Les trois derniers permettent d'afficher les données contenues dans les différentes tables d'après l'utilisateur connecté.

Le dernier dossier est intitulé **insidePages**, car il est constitué des différentes pages disponibles lorsque l'utilisateur est connecté.

— ***chgMed.php*** est la page qui permet d'ajouter ou de modifier un médicament à la base.

— ***chgPwd.php*** permet de changer de mot de passe.

— ***chooseImage.php*** propose de choisir une image *(png)* pour son médicament.

— ***page.php*** est la page la plus utilisée du site : c'est elle qui permet de consulter les données de la base, de changer ses données personnelles et de supprimer son compte.

— ***productList.php*** est la page qui permet de consulter la liste des médicaments sans être connecté.

3.2.3 Dossier racine

Le fichier ***index.php*** affiche, comme on peut s'en douter, la page d'accueil du site qui permet de se connecter et de s'inscrire au site, et d'afficher la liste des médicaments.

La page d'erreur de connexion à la base de données, ***DBerror.php***, affiche un message d'erreur *(erreur 503)* lorsqu'il y a une erreur de connexion à la base de données.

3.3 Connexion à la base de données

En ce qui concerne la connexion à la base de données, aucun *design pattern* n'a été utilisé, car, lorsque cette matière a été abordée au cours, la majeure partie du projet était déjà achevée.

Le type de requêtes utilisé pour l'accès à la base de données est le type PDO.
De manière plus précise, ce sont des requêtes préparées qui ont été utilisées pour plus de sécurité *(voir la section 3.4* ***Sécurité****)*.
De plus, PDO permet plus de flexibilité, est recommandée, très utilisée *(et possède donc une importante communauté)* et est enseignée à l'école.

On pourrait alors ajouter, dans les évolutions possibles, le passage à la *programmation orientée objet*, couplée à l'utilisation du *singleton* et, comme structure de site, l'utilisation du *design pattern* ***MVC***.

3.4 Sécurité

Pour la sécurité du site tout accès à la base de données passe par des requêtes préparées *(PDO)*, car celles-ci protègent le site de toute injection SQL.

La connexion au site est surveillée grâce à la mise en place d'une session. Dès qu'un accès à une page se fait, la connexion est vérifiée. Si l'utilisateur n'est pas identifié, il est redirigé vers la page d'accueil.

En plus de cela, chaque champ de formulaire est protégé par des expressions régulières *(regex)*.
Celles-ci vérifient que les données entrées sont de la forme voulue : un nom ne doit contenir que des lettres et, selon une certaine forme, peut être divisé à l'aide d'un trait d'union ; une adresse mail ne contient que des lettres minuscules non-accentuées, des points et un seul arobase ; etc.

De plus, les mots de passe ont une longueur minimale de quatre caractères et doivent contenir, au moins, une majuscule, un minuscule et un chiffre.
Ils sont aussi hachés à l'aide de *SHA512* et d'un *salage dynamique* pour plus de sécurité.
En effet, le mot de passe est de la forme +%# **longueur mot_de_passe ¨*§**.
Par exemple, le mot de passe **Test1** devient +%#**5Test1¨*§** avant d'être haché.
Il est ainsi indéchiffrable, mais cela a aussi pour effet de sécuriser la base de données puisque le résultat ne contiendra que des caractères sûrs.

4 Conclusion

4.1 Résultat

Le site propose donc à chacun de créer un compte afin de posséder sa liste de médicaments.

Un utilisateur, identifié par son adresse mail et son nom d'utilisateur, peut modifier ses données depuis la page *Compte*.
Il peut ainsi modifier son nom de famille, son prénom, son adresse mail et son mot de passe. Il a aussi la possibilité de supprimer son compte.

L'administrateur peut être contacté par les utilisateurs du site que ce soit pour ajouter un médicament ou pour réinitialiser leur mot de passe.
Il peut aussi bannir des utilisateurs.
En plus d'ajouter des médicaments, il a bien sûr la possibilité de les modifier, mais pas de les supprimer. En effet, un utilisateur possédant un médicament sur le point d'être supprimé serait mécontent de le perdre.
Comme les autres utilisateurs, il peut changer ses données, mais il ne peut pas supprimer son compte, étant donné qu'il est l'administrateur.

Les visiteurs du site ne possédant pas de compte ont la possibilité de parcourir la liste des différents médicaments disponibles.

Il est à remarquer que, d'après les bonnes pratiques des bases de données, il ne faut pas supprimer les données mais les rendre invisibles aux utilisateurs.
Cette pratique est utilisée lors de la suppression d'un compte qui n'est pas supprimé, mais simplement désactivé.

4.2 Différences avec le cahier des charges

Lors de la création de la maquette du site, l'utilisation d'avatars pour identifier les utilisateurs s'est avérée moins intéressante et utile qu'escompté.
En effet, les utilisateurs n'ayant pas d'interactions entre-eux et ne communiquant avec l'administrateur que par mails, les avatars sont devenus superflus.

Lors de la conception de la version préliminaire, il est apparu que les effets néfastes et les contre-indications étaient souvent similaires.
J'ai alors pris la décision de les regrouper afin d'éviter les doublons.