

# Service d'authentification avec PHP et MySQL

## Introduction :

Lors de ce guide nous allons voir comment mettre en service un service d'authentification avec PHP et MySQL, ici pour accéder à un accès limité à une page web.

On va principalement se focaliser sur le code PHP et non sur le code HTML qui pourrait être modifié à votre guise pour donner un aperçu différent du miens.

## Quesque l'intranet ? :

L'intranet est un réseau informatique privé utilisé par une organisation pour permettre la communication, le partage d'informations et la collaboration entre ses membres autorisés. Il offre un accès restreint, facilite le partage de documents et de données, favorise la communication interne et peut héberger des applications spécifiques à l'organisation.

## Pré requis :

*Debian 12 (Graphique pour ma part)*

*Éditeur de langage tel que : VSC ou Notepad++*

*Connexion internet pour la première partie uniquement*

## 1 - Installation de XAMPP

Première étape, il faut se rendre à l'URL suivante dans le navigateur afin de récupérer la dernière version de xampp.

<https://sourceforge.net/projects/xampp/files/XAMPP%20Linux/>

Une fois l'installation terminée, celui-ci sera dans /home/user/Download donc on va faire la commande suivante pour ma part :

```
→ cd /home/lucas/Téléchargements
```

Ensuite on va pouvoir lancer l'installation de xampp avec les commandes suivantes :

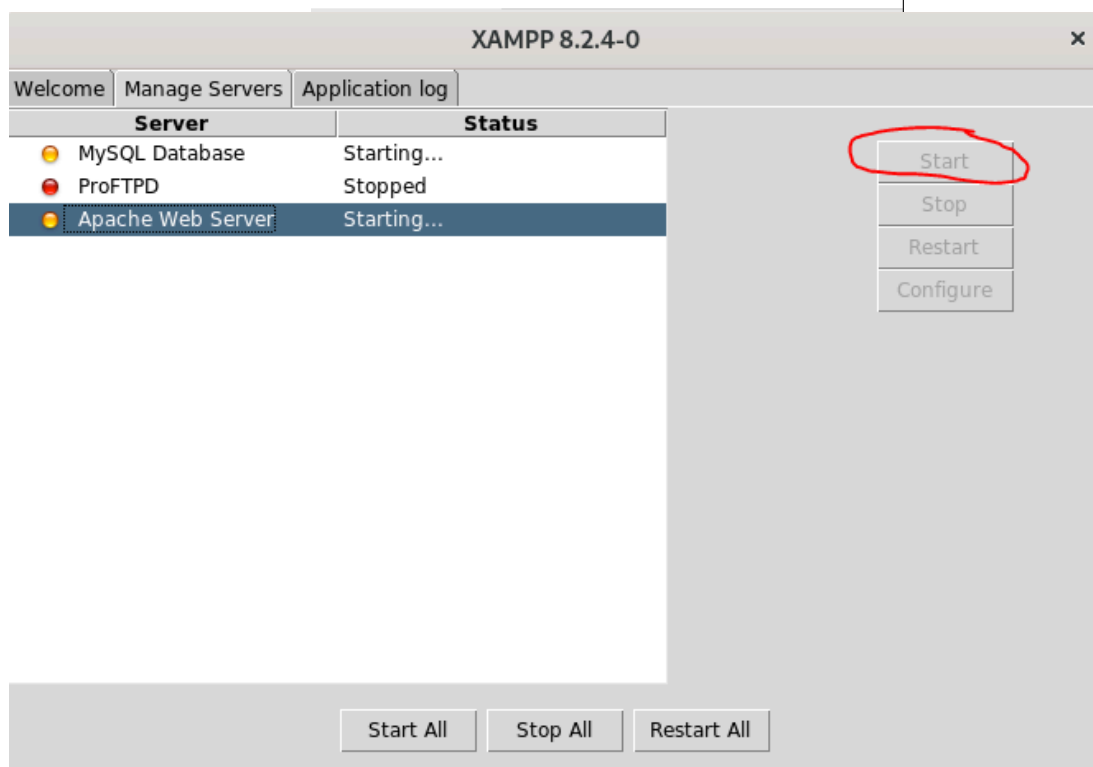
La première commande sert à mettre les droit d'exécution au fichier puis la seconde commande pour lancer l'installation.

```
lucas@srv-web-bdd: ~  
root@srv-web-bdd:/home/lucas/Téléchargements# chmod 777 xampp-linux-x64-8.2.4-0-installer.run  
root@srv-web-bdd:/home/lucas/Téléchargements# ./xampp-linux-x64-8.2.4-0-installer.run  
[]
```

Une fois l'installation de xampp terminée on va avoir cette onglet qui s'affiche :  
On clique donc sur finish en laissant la case "Launch XAMPP" de cocher.



Ensuite on va donc choisir les services qu'on souhaite démarrer et on fait start.



Ou sinon on peut le faire en commande avec les commandes suivantes :  
Si déjà un serveur apache2 alors faire un `systemctl stop apache2` et ensuite faire les commandes ci-dessous.

```

lucas@srv-web-bdd: ~
root@srv-web-bdd:/home/lucas/Téléchargements#
root@srv-web-bdd:/home/lucas/Téléchargements#
root@srv-web-bdd:/home/lucas/Téléchargements# /opt/lampp/lampp status
Version: XAMPP for Linux 8.2.4-0
Apache is running.
MySQL is running.
ProFTPD is running.
root@srv-web-bdd:/home/lucas/Téléchargements# /opt/lampp/lampp start

```

Enfin pour accéder à htdocs pour les pages web il faut faire :

```

root@srv-web-bdd:/home/lucas/Téléchargements# cd /opt/lampp/
root@srv-web-bdd:/opt/lampp# ls
apache2      error      info      logs      pear      README-wsrep  uninstall
bin          etc        lamp      man      php      RELEASNOTES  uninstall.dat
build        htdocs     lib       manager-linux-x64.run  phpmyadmin  sbin         var
cgi-bin      icons     lib64     manual    proftpd   share        xampp
ctlscrip.sh  img       libexec   modules   properties.ini  temp
docs         include   licenses  mysql     README.md  THIRDPARTY
root@srv-web-bdd:/opt/lampp# cd ../htdocs/
root@srv-web-bdd:/opt/lampp/htdocs# ls
applications.html bitnami.css dashboard favicon.ico img index.php webalizer
root@srv-web-bdd:/opt/lampp/htdocs#
root@srv-web-bdd:/opt/lampp# chmod -R 777 htdocs/

```

## Test de validation de la réussite de l'installation :



## Bienvenue dans XAMPP pour Linux 8.2.4

Vous avez installé avec succès XAMPP sur ce système ! Vous pouvez maintenant commencer à utiliser Apache, MariaDB, PHP et d'autres composants. Vous pouvez trouver plus d'informations dans la section FAQ ou consulter les guides HOW-TO pour démarrer avec les applications PHP.

XAMPP est destiné uniquement à des fins de développement. Il possède certains paramètres de configuration qui facilitent le développement local mais qui ne sont pas sécurisés si vous souhaitez que votre installation soit accessible à d'autres.

Démarrez le panneau de configuration XAMPP pour vérifier l'état du serveur.

### Communauté

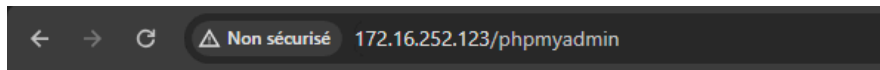
XAMPP existe depuis plus de 10 ans – il y a une énorme communauté derrière lui. Vous pouvez vous impliquer en rejoignant nos forums , en nous aimant sur Facebook ou en suivant nos exploits sur Twitter .

Maintenant nous allons aller dans notre base de données mySQL avec phpmyadmin

En local il faut mettre l'URL suivant : `localhost/phpmyadmin`

Pour y accéder sur le réseau comme je veux le faire il faut mettre l'URL suivant : `http://172.16.252.123/phpmyadmin`

on vas tomber sur cette page :



## Accès interdit!

### Nouveau concept de sécurité XAMPP:

L'accès à l'objet demandé est uniquement disponible sur le réseau local.

Ce paramètre peut être configuré dans le fichier "httpd-xampp.conf".

Si vous pensez qu'il s'agit d'une erreur du serveur, veuillez contacter le [webmestre](#).

### Error 403

Pour résoudre le problème on vas donc modifier le fichier de conf

`172.16.252.123`

Faire la commande suivante :

```
root@srv-web-bdd:/opt/lampp/htdocs# nano /opt/lampp/etc/extra/httpd-xampp.conf
```

Puis changer la ligne **Require local** en **Require all granted**

```
GNU nano 7.2 /opt/lampp/etc/extra/httpd-xampp.conf
# We will enable it by default
#<IfDefine PHP>
    LoadModule php_module          modules/libphp.so
#</IfDefine>

LoadModule perl_module             modules/mod_perl.so

Alias /phpmyadmin "/opt/lampp/phpmyadmin"

# since XAMPP 1.4.3
<Directory "/opt/lampp/phpmyadmin">
    AllowOverride AuthConfig Limit
    Require local
    ErrorDocument 403 /error/XAMPP_FORBIDDEN.html.var
</Directory>

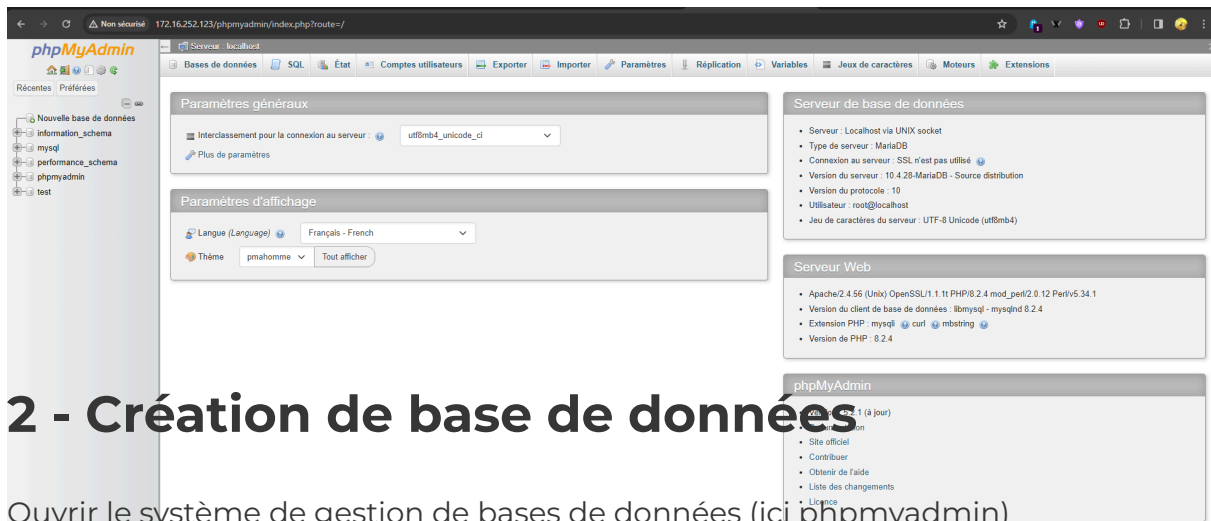
# since LAMP 1.0RC1
AddType application/x-httpd-php .php .php3 .php4
```

Maintenant redémarrer le services avec la commande suivante :

```
root@srv-web-bdd: /opt/lampp/htdocs# /opt/lampp/lampp restart
```

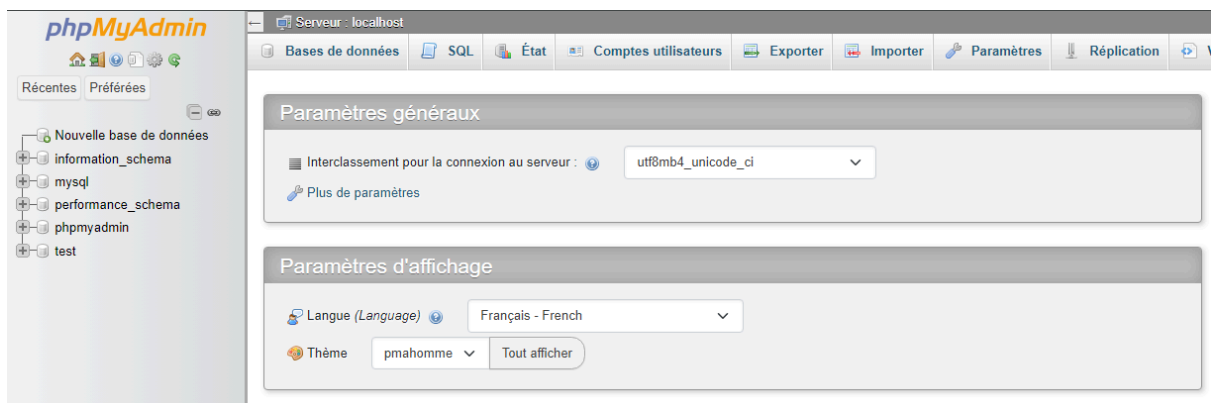
Une fois cela fait retourner dans le navigateur et faire F5 puis on as maintenant accès à phpmyadmin.

On obtient donc comme la capture suivante :

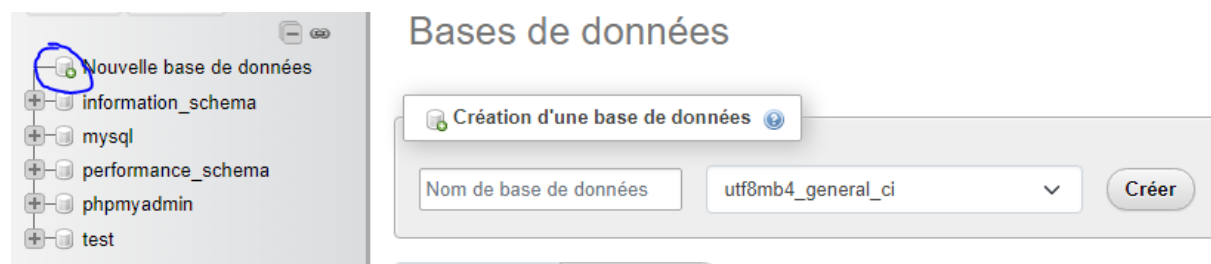


## 2 - Création de base de données

Ouvrir le système de gestion de bases de données (ici phpmyadmin)



Cliquer sur "new" ou "nouvelle base" et créer une nouvelle base de données appelée "auth\_system".



Ajoutez une table appelée "users" avec les colonnes suivantes :

*id (int, clé primaire, auto-increment)*  
*username (varchar 255)*  
*password (varchar 255)*  
*email (varchar 255)*

The screenshot shows a database management interface with a menu bar at the top including options like Structure, SQL, Rechercher, Requête, Exporter, Importer, Opérations, Privileges, Procédures stockées, Événements, Déclencheurs, and Suivi. Below the menu, the 'Structure' tab is active, displaying the structure of a table named 'users'. The table has four columns: 'id' (INT, PRIMARY, auto-increment), 'username' (VARCHAR, 255), 'email' (VARCHAR, 255), and 'password' (VARCHAR, 255). The 'id' column is marked as the primary key. The interface also includes fields for 'Nom de table', 'Ajouter', 'colonne(s)', and 'Exécuter'.

### 3. Création de dossiers et fichier nécessaires

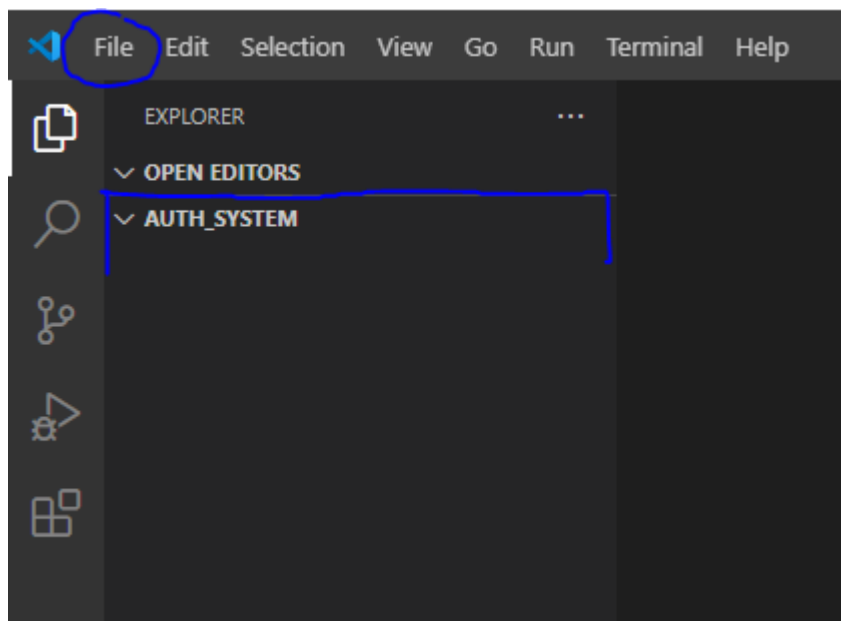
Rendez-vous sur la machine et créer un dossier nommé “auth\_system” avec un système unix, tapez: mkdir “auth\_system” Ouvrez le dossier avec VS code.

Pour ma part je vais créer sur mon pc hôte et j’enverrais les fichiers en SFTP avec Termius sur ma VM debian12

Donc je crée un dossier “auth\_system” avec un clique droit “nouveau dossier”

Puis j’ouvre VisualStudioCode et je fait en haut à gauche file puis open folder et sélectionne le dossier.

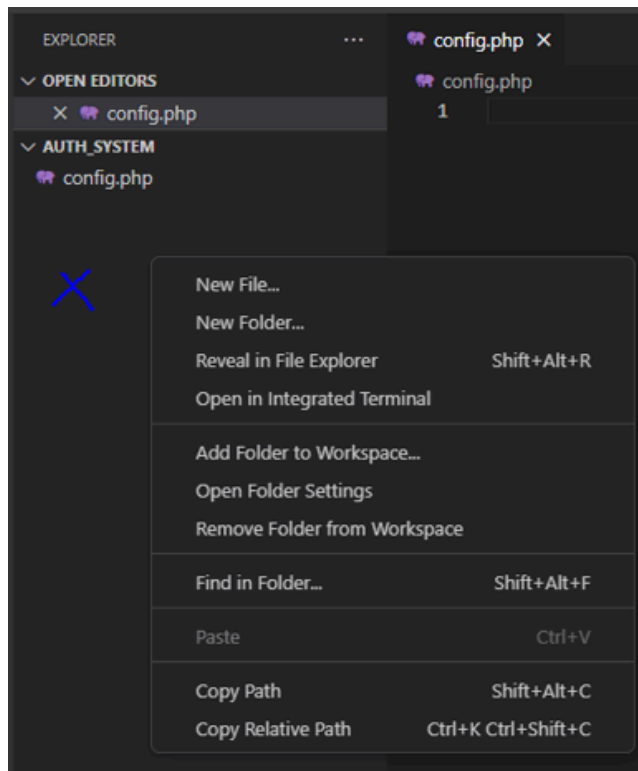
Ci la manipulation a été réussi, on obtient cela :  
(Dossier vide car aucun fichier présent)



## 4. Connection à la base de données :

Créez un fichier config.php:

Dans VisualStudioCode faire un clic droit puis “new file” et nommée le fichier.



Maintenant entrer le code suivant :

(A adapter en fonction des besoins)

```
config.php X
config.php
1  <?php
2  $host = 'http://172.16.252.123/';
3  $db   = 'auth_systeme';
4  $user = 'root';
5  $pass = '';
6  $port = '8889';
7  $charset = 'utf8mb4';
8
9  $dsn = "mysql:host=$host;port=$port;dbname=$db;charset=$charset";
10 $options = [
11     PDO::ATTR_ERRMODE            => PDO::ERRMODE_EXCEPTION,
12     PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
13     PDO::ATTR_EMULATE_PREPARES   => false,
14 ];
15
16 try {
17     $pdo = new PDO($dsn, $user, $pass, $options);
18 } catch (\PDOException $e) {
19     throw new \PDOException($e->getMessage(), (int)$e->getCode());
20 }
21 |
```

## 5. Page d'inscription :

Créez un formulaire d'inscription (register.php) qui prend le nom d'utilisateur, l'email et le mot de passe. Assurez-vous d'utiliser password\_hash() pour sécuriser le mot de passe avant de l'enregistrer.

(code html et css à ajouter pour de la mise en page)

```
register.php
1  <?php
2
3  include("config.php");
4
5  $message = '';
6
7  if (isset($_POST['username']) && isset($_POST['password'])) {
8      $username = $_POST['username'];
9      $password = password_hash($_POST['password'], PASSWORD_DEFAULT);
10     $email = $_POST['email'];
11
12     $sql = "INSERT INTO users (email, username, password) VALUES (:email, :username, :password)";
13     $stmt = $pdo->prepare($sql);
14     $result = $stmt->execute(['email' => $email, 'username' => $username, 'password' => $password]);
15
16     if ($result) {
17         $message = 'Inscription réussie!';
18         header('Location: login.php');
19     } else {
20         $message = 'Erreur lors de l\'inscription.';
21     }
22 }
23
```

## 6. Page de connexion :

Créez un formulaire de connexion (login.php). Lorsqu'un utilisateur essaie de se connecter, vérifiez le mot de passe avec password\_verify().

```
config.php  register.php  login.php X
login.php
1  <?php
2
3  include("config.php");
4
5  $message = '';
6
7  if (isset($_POST['username']) && isset($_POST['password'])) {
8      $username = $_POST['username'];
9      $password = $_POST['password'];
10
11     $sql = "SELECT * FROM users WHERE username = :username";
12     $stmt = $pdo->prepare($sql);
13     $stmt->execute(['username' => $username]);
14     $user = $stmt->fetch();
15
16     if ($user && password_verify($password, $user['password'])) {
17         session_start();
18         $_SESSION['user_id'] = $user['id'];
19         header('Location: dashboard.php');
20     } else {
21         $message = 'Mauvais identifiants';
22     }
23 }
24
```



## 7. Dashboard :

Créez un fichier (dashboard.php) où les utilisateurs connectés peuvent accéder à leurs informations.

```
dashboard.php
1  <?php
2  session_start();
3
4  // Vérifier si l'utilisateur est connecté
5  if (!isset($_SESSION['user_id'])) {
6      header('Location: login.php');
7      exit;
8  }
9
10 ?>
```

## 8. Déconnexion :

Créez un fichier (logout.php) pour supprimer la session:

```
<?php

session_start();

// Supprimer toutes les variables de session.
$_SESSION = array();

// Si vous voulez détruire complètement la session, supprimez également
// le cookie de session.
if (ini_get("session.use_cookies")) {
    $params = session_get_cookie_params();
    setcookie(session_name(), '', time() - 42000,
        $params["path"], $params["domain"],
        $params["secure"], $params["httponly"]
    );
}

// Finalement, détruire la session.
session_destroy();

// Rediriger vers la page de connexion.
header('Location: login.php');
exit;

?>
```

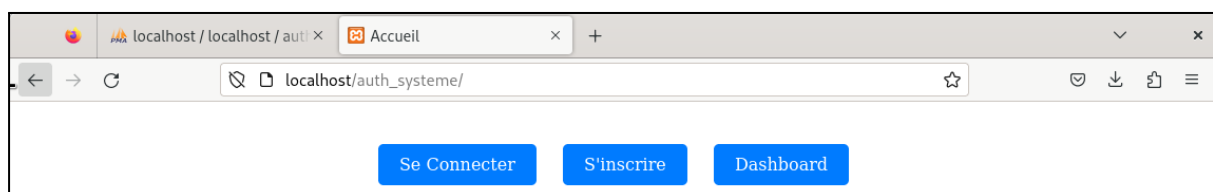
## 9. Finalisation et Test de l'application :

Ajoutez un fichier (index.php) et mettez le code suivant :

(Code pouvant être modifié pour un affichage différent du mien, mais voici tout de même un code simple permettant de tester la réalisation.

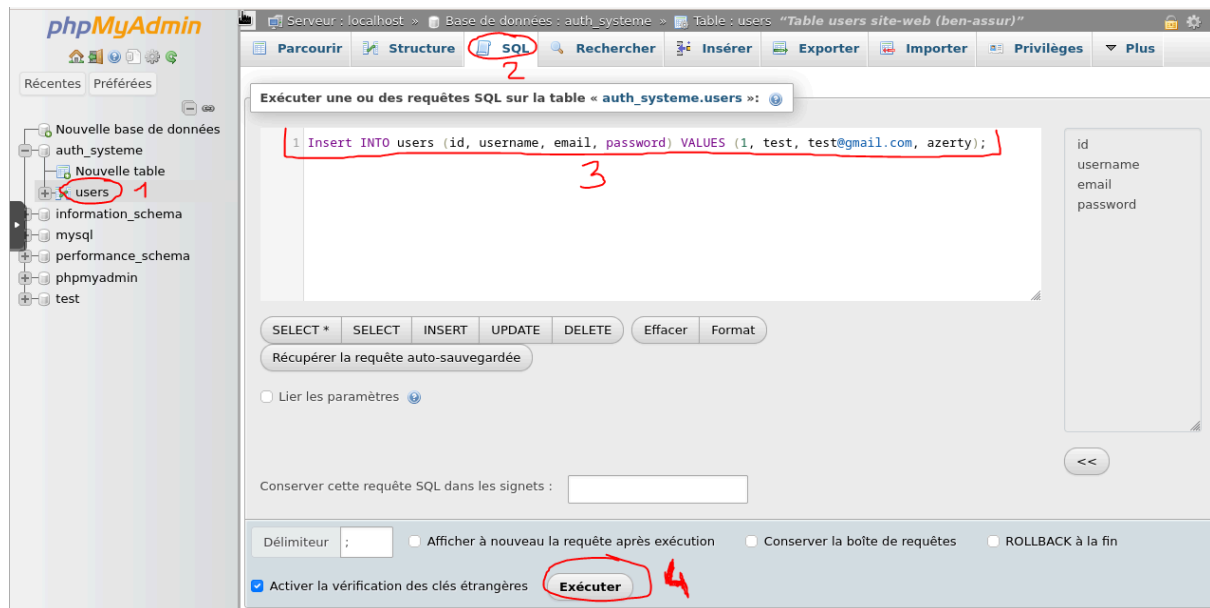
```
index.php
1  <!DOCTYPE html>
2  <html lang="fr">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Accueil</title>
7
8      <style>
9          .buttons-container {
10             text-align: center;
11             margin-top: 50px;
12         }
13         .btn {
14             margin: 10px;
15             padding: 10px 20px;
16             background-color: #007BFF;
17             color: #fff;
18             text-decoration: none;
19             border-radius: 5px;
20             transition: background-color 0.3s;
21         }
22         .btn:hover {
23             background-color: #0056b3;
24         }
25     </style>
26
27 </head>
28 <body>
29     <div class="buttons-container">
30         <a href="login.php" class="btn">Se connecter</a>
31         <a href="register.php" class="btn">S'inscrire</a>
32         <a href="dashboard.php" class="btn">Dashboard</a>
33     </div>
34
35 </body>
36 </html>
37
```

Pour valider, rendez-vous à l'URL suivante "[http://localhost:8888/auth\\_systeme/](http://localhost:8888/auth_systeme/)". Remplacer le port "8888" si vous n'avez pas mis le même que moi.



Testez le système en créant des comptes sur la page "s'inscrire", en vous connectant et en vérifiant le tableau de bord.

Pour entrer manuellement des comptes dans la base de données la commande suivante peut être utilisée dans phpmyadmin de la façon suivante (de 1 à 4) :



```
INSERT INTO users (id, username, email, password) VALUES (1, 1.menard, lucas@gmail.com, azerty);
```

Code source final : <https://github.com/luucasmnd/auth>