

PROJET MI-SEMESTRE: TCHAT

Programmation du web

Réalisée par: ABDI GHAVIDEL Azadeh

Master2 MIASHS (parcours: WEB ANALYSE)

Année: 2020-2021

Tuteur : Monsieur BIGO

Établissement: Université de LILLE



TABLE OF CONTENTS

CAHIER DES CHARGES	3
<i>LES CONTRAINTES.....</i>	<i>3</i>
<i>LES prestations attendues</i>	<i>3</i>
<i>LES besoins fonctionnels.....</i>	<i>3</i>
<i>Delai de Projet</i>	<i>3</i>
INTRODUCTION	4
Étapes du projet	4
<i>Préparation les machines virtuelles.....</i>	<i>4</i>
<i>Coté Client « Base de Données »</i>	<i>4</i>
<i>côté Serveur.....</i>	<i>5</i>
Partie codage.....	6
<i>Index.php (Figure 3).....</i>	<i>6</i>
<i>bd.php (Figure4)</i>	<i>7</i>
<i>LOGIN.PHP (Figure 5,6,7,8).....</i>	<i>7</i>
Partie PHP.....	8
Partie HTML.....	8
Partie AJAX	9
<i>Signup.php (Figure 9)</i>	<i>9</i>
<i>Chat.php (Figure 10, 11 et 12).....</i>	<i>10</i>
<i>SEND.PHP, RECIEVE.PHP (Figure 13 et 14)</i>	<i>12</i>
<i>Logout.php (Figure 15)</i>	<i>12</i>
Conclusion.....	13
Remerciement	13



CAHIER DES CHARGES

Projet Tchat est un site permettant à plusieurs utilisateurs de se connecter et de discuter.

Ce projet est Réalisé par Azadeh Abdi Ghavidel, étudiant en master MIASHS parcours web analyste deuxième année à l'université de Lille pour le cours de programmation du web et à l'aide de Monsieur Louise Bigo, tuteur et responsable de ce cours.

LES CONTRAINTES

- Le site doit être réalisé sur une machine virtuelle de l'université et versionné sous git.
- Ne pas utiliser de Framework.

LES PRESTATIONS ATTENDUES

- Concevoir une base de données qui correspondent à ce cahier des charges
- Écrire une application serveur en PHP qui permettent de poster et de récupérer les informations stockées dans la base de données
- Écrire une interface client agréable (HTML/CSS/JavaScript)
- (Optionnel) Pour chaque message, les utilisateurs peuvent voir par quels utilisateurs a été vu le message
- (Optionnel) Écrire un client Python3 et un petit robot simple de discussion

LES BESOINS FONCTIONNELS

- Création machine virtuelle pour le côté client dans laquelle il faut créer une base de données pour stocker les messages.
- Créer une machine virtuelle pour le côté serveur dans laquelle il faut mettre tous les pages html, php.
- Création d'une page Login
- Utilisation CSS et JavaScript
- Rédaction d'une page pour le chat
- Utiliser PDO pour connecter à la base de données.
- Utiliser ajax afin de poster et de récupérer les informations stockées dans la base de données.

DELAÏ DE PROJET

Tchat est le projet mi- semestre à finaliser pour la date prévisionnelle : le lundi 16 novembre 2020.



INTRODUCTION

Tchat est un site web permettant à plusieurs utilisateurs de se connecter et de discuter en utilisant les langages et les méthodes appliqués dans le cours de programmation web.

ÉTAPES DU PROJET

PREPARATION DES MACHINES VIRTUELLES

La première étape du travail a consisté à créer deux machines virtuelles sur le cloud d'université, une pour le côté client et l'autre pour le côté serveur et à installer les outils nécessaires comme apache2, MySQL, phpmyadmin.

L'IP de mes instances :

Serveur : 172.28.100.73/project/index.php?page=login

Client : 172.28.100.121

COTE CLIENT « BASE DE DONNEES »

Après la création des instances, sur le côté client j'ai créé une base de données qui s'appelle tchat, dans cette base de données j'ai créé deux tables, l'une « USERS » afin de stocker le nom d'utilisateurs et leur mot de passe (figure1), et l'autre « message » pour stocker les messages que les utilisateurs écrivent (figure2).


	#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra
<input type="checkbox"/>	1	id 	int			Non	Aucun(e)		AUTO_INCRE
<input type="checkbox"/>	2	username	varchar(15)	utf8mb4_0900_ai_ci		Non	Aucun(e)		
<input type="checkbox"/>	3	password	varchar(15)	utf8mb4_0900_ai_ci		Non	Aucun(e)		

Figure 1 : table USERS



COTE SERVEUR








#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/>	1	id 	int		Non	Aucun(e)		AUTO_INCREMENT	 Modifier  Supprimer ▼ F
<input type="checkbox"/>	2	name	varchar(15)	utf8mb4_0900_ai_ci	Non	Aucun(e)			 Modifier  Supprimer ▼ F
<input type="checkbox"/>	3	message	varchar(255)	utf8mb4_0900_ai_ci	Non	Aucun(e)			 Modifier  Supprimer ▼ F

Figure 2 : table message

Dans le coté serveur j'ai créé un fichier « projet » contenant tous mes fichiers pour ce projet :

- Une page index.php à fin d'organiser tous mes pages pour ce projet.
- Fichier « BD » : contenant le fichier bd.php
- Fichier « core » : contenant les fichiers :
 - login.php : une page php pour se connecter au fichier chat.
 - chat.php : ma page principale de ce projet.
 - send.php : une page php à fin d'envoyer et insérer les messages dans la base de données.
 - recieve.php : une page consistée à récupérer les messages de la base de données.
 - signup.php : une page php à fin d'insérer nouveaux utilisateurs dans la base de données.
 - logout.php : une page php pour déconnexion du chat.
 - style.css : une page pour styler mes codes HTML.



PARTIE CODAGE

INDEX.PHP (FIGURE 3)

1. Ouvrir une session :
 - a. Tout d'abord, j'ai ouvert une session pour récupérer le nom d'utilisateur à fin d'utiliser dans tous les pages php en utilisant \$_SESSION['user'].
2. Inclure ma fichier base de données.
3. Inclure tous mes pages php qui existent dans le fichier « core ».

Figure 3 : index.php

```
index.php X
home > negar > .ssh > tchat > index.php
1  <?php
2  #j'ai ouvert une session pour récupérer le nom d'utilisateur à fin d'utiliser dans tous les pages php
3  session_start();
4  if (isset($_POST['login'])) {
5      $_SESSION['user']=$_POST['username'];
6  }
7
8  require("../BD/bd.php");
9
10 #dir_core est address dans la quelle j'ai mis toutes mes pages (.php)
11 $dir_core = './core/';
12 #création d'une variable page pour suivre mes pages dans le fichier core
13 $page = $_GET['page'];
14 #va chercher dans le fichier core et prend le nom de chaque page avec GET[page]
15 #et téléchargé la , et ajouter .php à la fin de son nom pour les inclure dans ce fichier (index)
16 include_once($dir_core.$page.".php");
17 ?>
18
```

Pour Quoi index.php ?

- Connexion à base de données une fois, pour tous les fichiers
- Gérer tous les pages php à partir d'une page index
- Écrire « 172.28.100.73/chat/index.php ?page=login » et éviter de taper « 172.28.100.73/chat/login.php »



BD.PHP (FIGURE4)

Dans cette page en utilisant PDO j'ai connecté à ma base de données qui est sur IP 172.28.100.121 à l'aide de MySQL.

```
index.php bd.php x
home > negar > .ssh > tchat > BD > bd.php
1  <?php
2  try {
3      $base = new PDO('mysql:host=172.28.100.121;port=3306;dbname=tchat','azad','886622');
4  } catch (\Exception $e) {
5      die('Erreur : ' . $e->getMessage());
6  }
7  ?>
8
```

Figure 4 : Base de données (bd.php)

LOGIN.PHP (FIGURE 5,6,7,8)

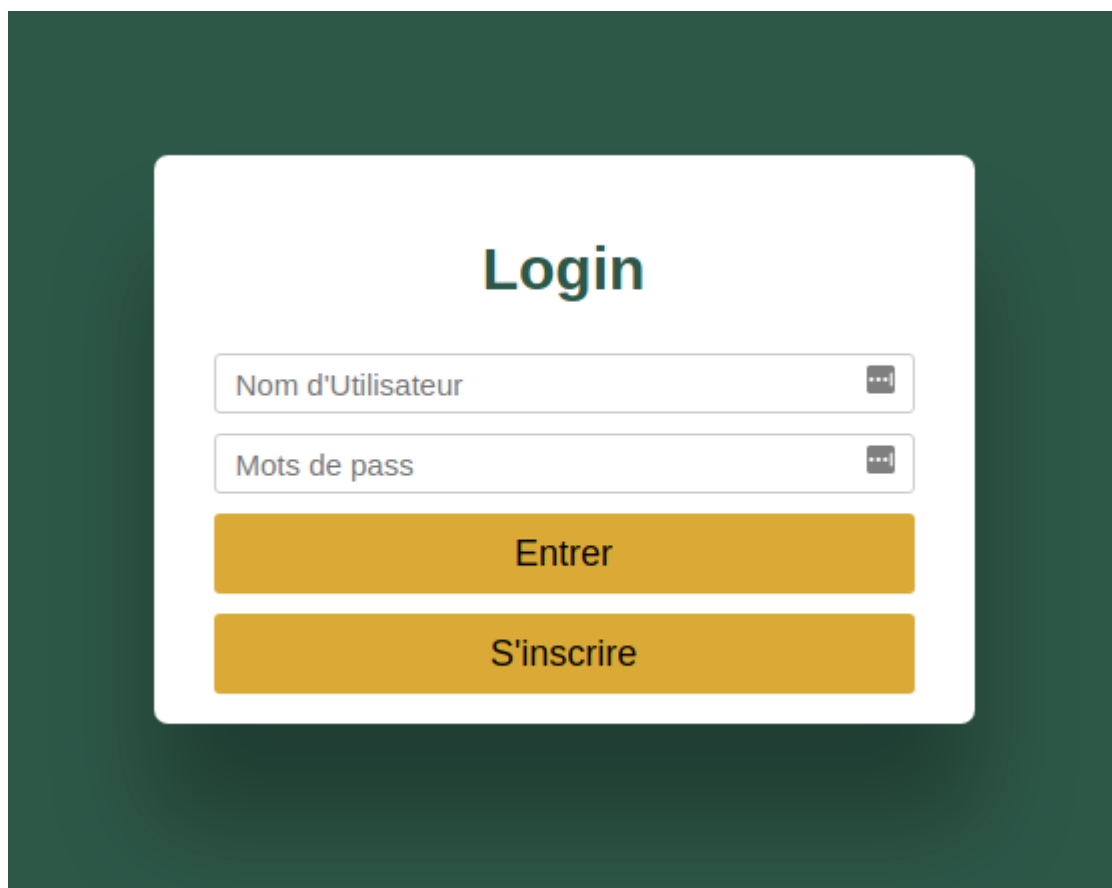


Figure 5 : Interface login.php

Ma page login.php est construit par trois parties :

1. PHP (Figure 6)
2. HTML (Figure 7)
3. Ajax (Figure 8)

PARTIE PHP

Ici j'ai utilisé le langage php afin de se faire connecter à la base de données et vérifier s'ils existent le nom d'utilisateur et mot de passe que mon utilisateur à saisi.

```
<?php
if(isset($_POST['login']) and $_POST['username'] and $_POST['pwd']) {
    try {
        $uname = trim($_POST['username']);
        $pass = trim($_POST['pwd']);
        $query = "SELECT * FROM `USERS` WHERE username='$uname' AND password='$pass'";
        $reponse = $base->query($query);
        if($ligne = $reponse->fetch()){
            $_SESSION['user'] = $uname;
            header('location:index.php?page=chat');
        }
    } catch (PDOException $e) {
        echo "Error : ".$e->getMessage();
    }
}
}??
```

Figure 6 : Partie PHP

Si oui il va ouvrir la page chat.php

PARTIE HTML

Dans cette partie j'ai créé une forme pour récupérer le nom d'utilisateur et son mot de passe.

J'ai mis aussi deux buttons, un pour connexion (qui est utilisé par mes code php) et l'autre

```
<body>
<div class='all'>
    <h1>Login</h1>
    <form action="" method="POST">
        <input type="text" name="username" placeholder="Nom d'Utilisateur" id="uname">
        <input type="password" name="pwd" placeholder="Mots de pass" id="pass">
        <button type="submit" name="login">Entrer</button>
        <button onclick="signup()">S'inscrire</button>
    </form>
</div>
```

Figure 7 : Partie HTML

pour inscription (qui est utilisé pour la partie ajax).



PARTIE AJAX

Dans cette partie j'ai écrit une fonctionne en utilisant l'Ajx pour envoyer des données à mon fichier signup.php - en arrière-plan.

```
<script>
// Ici j'ai créé une fonctionne pour pouvoir ajouter nouveaux utilisateur et mots de pass dans ma table USERS dans ma base de données en utilisant ajax
function signup(){
    var uname=document.getElementById("uname").value;
    var pass=document.getElementById("pass").value;
    var data ="username="+uname+"&password="+pass;
    var signxhr = new XMLHttpRequest();
    signxhr.open('POST','index.php?page=signup',true);
    signxhr.setRequestHeader("Content-type","application/x-www-form-urlencoded");
    signxhr.addEventListener('readystatechange',function(){
        if (signxhr.readyState === XMLHttpRequest.DONE){
            document.getElementById("uname").value = "";
            document.getElementById("pass").value = "";
        }
    });
    signxhr.send(data);
}
</script>
```

Figure 8 : Partie AJAX

SIGNUP.PHP (FIGURE 9)

Dans ce fichier j'ai récupéré le nom d'utilisateur et son mot de passe à partir de la page login.php

```
<?php
try {
    $nom=$_POST['username'];
    $pwd=$_POST['password'];
    $requet="INSERT INTO `USERS` (username,password) VALUES ('$nom','$pwd')";
    $reponse = $base->exec($requet);
} catch (PDOException $e) {
    echo "Error : ".$e->getMessage();
}
?>
```

Figure 9 : signup.php

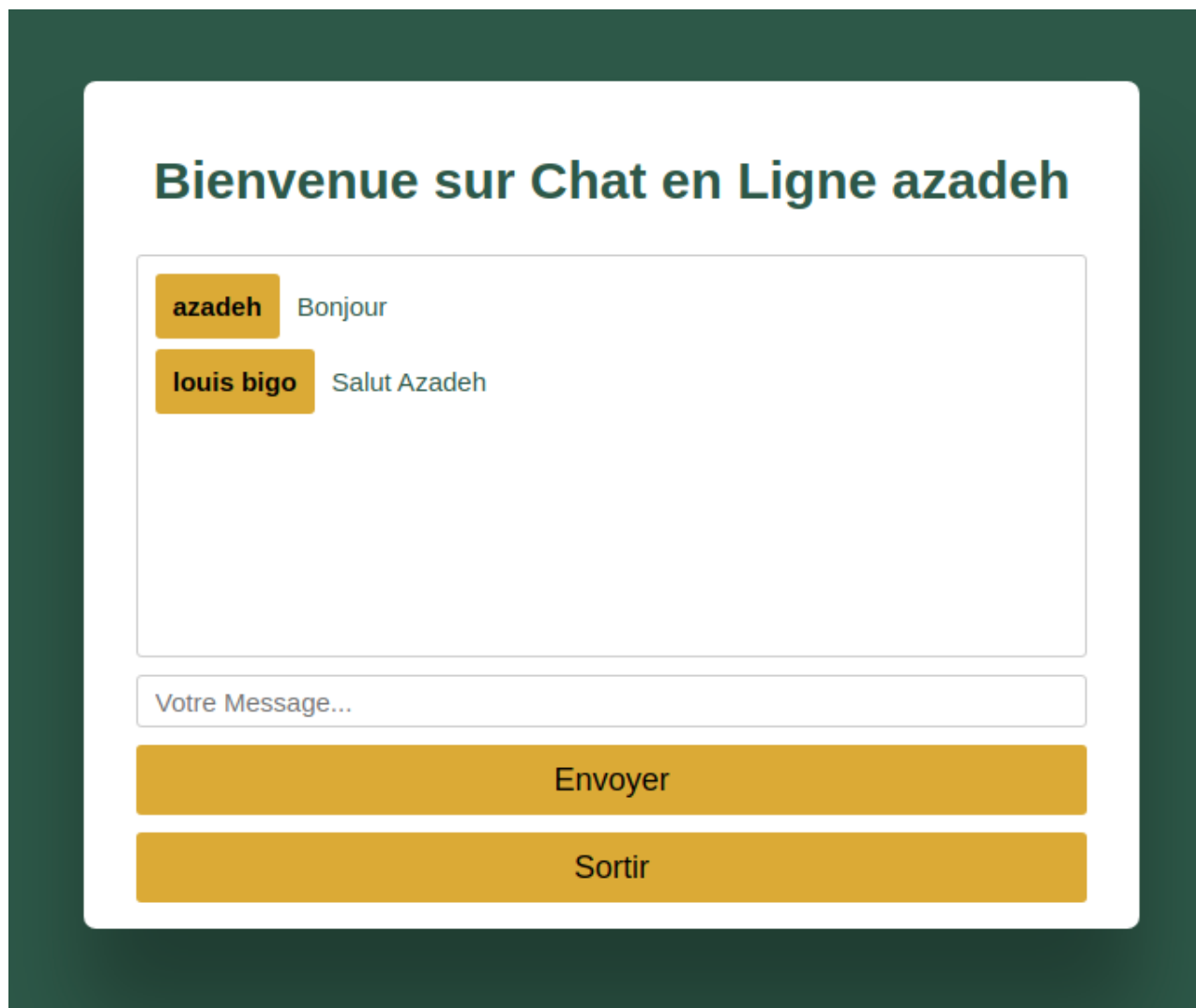


Figure 10 : Interface chat.php

Ma page chat contient deux parties : HTML et ajax.

Dans la partie Html j'ai créé :

Une division afin de montrer les messages.

Un élément input de type texte pour saisir nouveau message.

Un bouton de type submit pour envoyer le message écrit par utilisateur.

```
<div class='all'>
  <h1>Bienvenue sur Chat en Ligne <?php echo $_SESSION['user']; ?></h1>
  <div class="msgtext" id="msg">
  </div>
  <input type="text" name="msg" placeholder="Votre Message..." id="mesg">
  <button type="submit" name="envoyer" onclick="sendmsg()">Envoyer</button>
  <a href="index.php?page=logout" id="logout">Sortir</a>
</div>
```

Figure 11 : Partie HTML

Dans la partie script(ajax) j'ai créé :

- Une fonctionne getmsg en utilisant le méthode ajax pour récupérer tous les messages depuis ma base de données et les montrer dans ma « div » spécifié aux messages.
 - o J'ai créé une variable elem pour contrôler la barre de défilement dans la partie de messages
 - o J'ai appelé ma fonctionne getmsg () pour récupération les messages qu'ils existent déjà dans ma base de données
 - o J'ai fixé un intervalle de deux secondes, pour ma fonctionne getmsg afin de vérifier et récupérer les nouveaux messages
- Une fonctionne sendmsg () qui va envoyer les nouveaux messages à la page send.php à fin d'être insérer dans la base en arrière-plan.

```
<script>
// Création d'une fonctionne à fin de récupérer les messages depuis la base de données
function getmsg(){
  var xhr = new XMLHttpRequest();
  xhr.open('GET','index.php?page=recieve',false);
  xhr.send(null);
  //Lire la reponseText avec le format JSON
  var response = JSON.parse(xhr.responseText);
  var chat = '';
  for (var i = 0; i < response.length; i++) {
    chat+('<div id="inside"><span id="ch">'+response[i].name+'</span>'+response[i].message+'</div>');
  }
  document.getElementById("msg").innerHTML = chat;
  //Ici j'ai créé un variable elem pour contrôler la barre de défilement dans la partie de messages
  var elem = document.getElementById('msg');
  elem.scrollTop = elem.scrollHeight;
}
// J'ai appelé ma fonctionne getmsg() pour récupération les messages qu'ils existent déjà dans ma base de données
getmsg();
// J'ai fixé un intervalle de 2 secondes, pour ma fonctionne getmsg à fin de vérifier et récupérer les nouveaux messages
setInterval(getmsg,2000);

//Création d'une fonction pour envoyer un message à fichier send.php afin d'insertion le message dans la base de données
function sendmsg(){
  var message =document.getElementById("mesg").value;
  var sendxhr = new XMLHttpRequest();
  sendxhr.open('POST','index.php?page=send',true);
  sendxhr.setRequestHeader("Content-type","application/x-www-form-urlencoded");
  sendxhr.addEventListener('readystatechange',function(){
    if (sendxhr.readyState === XMLHttpRequest.DONE){
      document.getElementById("mesg").value = "";
    }
  });
  sendxhr.send('message='+message);
}
```

Figure 12: Partie AJAX

SEND.PHP, RECIEVE.PHP (FIGURE 13 ET 14)

Ces deux pages sont consistées à insérer les messages dans la base de données et récupérer les messages qui sont déjà insérer dans la base de données en utilisant PDO (exec () et query ())

```
<?php
try {
    $nom=$_SESSION['user'];
    $msg=$_POST['message'];
    $requet="INSERT INTO `message` (name, message) VALUES ('$nom', '$msg')";
    $reponse = $base->exec($requet);
} catch (PDOException $e) {
    echo "Error : ".$e->getMessage();
}
?>
```

Figure 13 : send.php

```
<?php
try {
    $requet="SELECT * FROM `message`";
    $reponse = $base->query($requet);
    $msg_list = $reponse->fetchAll(PDO::FETCH_ASSOC);
    $json_str = json_encode($msg_list);
    echo $json_str;
} catch (PDOException $e) {
    echo "Error : ".$e->getMessage();
}
?>
```

Figure 14 : recieve.php

LOGOUT.PHP (FIGURE 15)

En fin de ce projet j'ai créé une page log out en utilisant le langage php afin de supprimer la session et retourner à la page login si un utilisateur décide à déconnecter en cliquant sur le bouton « Sortir » du chat.

```
<?php
session_start();
unset($_SESSION["uname"]);
header("location:index.php?page=login");
?>
```

Figure 15 : logout.php



CONCLUSION

En faisant ce projet, j'ai appris beaucoup de choses comme utilisation d'Ajax, de JavaScript, PDO.

J'ai appris comment utiliser le PHP.

J'avais beaucoup de problème sur création un lien entre mes deux machines virtuelles en utilisant PDO, et j'ai appris comment résoudre ce problème.

Comme j'avais noté dans mon cahier des charges, je voulais écrire une fonctionne pour montrer les messages vus et les utilisateurs qui sont inactives, une fonctionne pour vérifier les exigences des utilisateurs dans la base de données et un client Python3 et un petit robot simple de discussion, mais malheureusement je n'ai pas pu les faire à cause de temps que j'ai perdu sur le côté technique que j'avais entre mes machines.

REMERCIEMENT

Je voudrais remercier Monsieur Louis bigo d'avoir partagé avec nous votre passion pour l'enseignement. J'ai grandement apprécié votre soutien, votre implication et votre expérience tout au long de ce projet.

