


DESCRIPTION D'UNE MISSION BTS SIO		
Prénom – Nom	Quentin Coqueran	N° mission
Option	SLAM	
Situation		Formation
Lieu de réalisation	Campus Montsouris 2 Rue Lacaze – 75014 PARIS	
Période de réalisation	Novembre 2020	Novembre 2020
Modalité de réalisation	VÉCUE	

Intitulé de la mission	Titre de la mission
	Création d'un chat
Description du contexte de la mission	Description en 2 à 3 lignes maxi
	Créer une page permettant l'envoi de message entre utilisateurs

Ressources et Outils utilisés	Liste des ressources disponibles et outils utilisés (Documentations, Matériels et Logiciels)
	<ul style="list-style-type: none"> - WampServer - Visual Studio Code - PhpMyAdmin - Une base de données message.sql
Résultat attendu	Résultat attendu avec la réalisation de cette mission L'affichage du prénom, message et de l'heure à laquelle le message a été posté
Contraintes	Contraintes : techniques budgétaires temps O.S. ou outils imposés...
	<ul style="list-style-type: none"> • Proposer un site permettant l'envoi de message • Créer une connexion Base de données / Site internet. • Vérifier que l'utilisateur ne rentre pas un nom ou message vide • Faire attention aux injections SQL

Compétences associées (Voir tableau)	Liste des intitulés du tableau de compétences (avec les références)
	A2.3.1 Identification, qualification et évaluation d'un problème
	A4.1.3 Conception ou adaptation d'une base de données

Description simplifiée des différentes étapes de réalisation de la mission en mettant en évidence la démarche suivie, les méthodes et les techniques utilisées
<p><u>Chapitre 1 : Début du développement de la page principale du projet</u></p> <p>Etape :</p> <p>1.1/ Tout d'abord nous créons le fichier index.php qui sera la seule page de ce projet, le fichier style.css qui comprendra le code CSS pour améliorer la partie physique du site.</p> <p>On se connecte à la base de données.</p> <pre>\$bdd = mysqli_connect("localhost", "root", "", "message");</pre> <p>1.2/ Pour envoyer des messages il faut envoyer son nom et son message dans les champs de saisie.</p> <pre><input class="form" placeholder="Entrez votre nom" type="text" name="nom"> <input class="form" placeholder="Ecrivez un message" type="text" name="message"></pre> <p>Résultat :</p> <div data-bbox="217 1265 1380 1346"> <input type="text" value="Entrez votre nom"/> <input type="text" value="Ecrivez un message"/> </div>
<p><u>Chapitre 2 : Réglages de problème et faille de sécurité</u></p> <p>Etape :</p> <p>2.1/ Etant donné qu'une requête SQL est composé de « " » ou de « ' », un message contenant l'un de ces deux caractères peu poser des problèmes lors de l'envoi de la requête, tout d'abord le message ne va pas s'enregistrer, mais en plus la base de données est exposée aux « injections SQL » qui consiste à modifier à partir du champ de saisie, la requête SQL qui est censé ajouter les messages.</p> <p>J'ai donc ajouter la fonction <code>mysqli_escape_string()</code>, aux variables qui vont récupérer les valeurs des deux champs, nom et message, cette fonction va permettre de transformer les quotes simple et double, afin de les afficher dans un message et ne pas interférer dans la requête SQL.</p>

```
$nom = htmlspecialchars(mysqli_escape_string($bdd, $_POST['nom'])) ;  
$message = htmlspecialchars(mysqli_real_escape_string($bdd, $_POST["message"]));
```

2.2/ La second faille consiste cette fois ci à envoyer du code dans les champs de saisie, on pourrait par exemple envoyer en message Bonjour , ce qui afficherais donc :

22:30:25 - **Nathan: Bonjour**

Et même si à première vue ça ne parait pas très grave, cela donne quand même à l'utilisateur la possibilité de modifier l'affichage du site, mais surtout de pouvoir envoyer du code JavaScript, et récupérer des informations via les cookies. Pour sa j'ai utilisé la fonction htmlspecialchars qui va transformer les balises < en < et > en > dans la base de données.

```
$nom = htmlspecialchars(mysqli_escape_string($bdd, $_POST['nom'])) ;  
$message = htmlspecialchars(mysqli_real_escape_string($bdd, $_POST["message"]));
```

2.3/ Dans le cas où un utilisateur ne renseigne pas son nom et son message et envoie un formulaire vide, le formulaire ne s'envoie pas et un message d'erreur s'affiche.

```
if(!empty($_POST['submit']))  
{  
    if(empty($_POST['nom']))  
    {  
        $nomVide = true;  
        $error = true;  
    }  
    if(empty($_POST['message']))  
    {  
        $error = true;  
        $messageVide = true;  
    }  
}
```

Le message d'erreur va du coup pouvoir s'afficher en fonction du problème rencontré :

```
<?php if(@$nomVide): ?>  
    <span class="error">Message anonyme interdit! Entrez votre nom.</span>  
    <br><br>  
<?php endif;?>  
<?php if(@$messageVide): ?>  
    <span class="error">Veuillez entrer votre Message!</span>  
    <br><br>  
<?php endif;?>
```

Si le formulaire ne rencontre aucun problème est que du coup la variable \$error renvoie false nous pouvons envoyer les messages dans la base de données grâce à la requête :

```




if(isset($_POST['submit']) && !isset($error))
{
    $nom = htmlspecialchars(mysqli_escape_string($bdd, $_POST['nom']));
    $message = htmlspecialchars(mysqli_real_escape_string($bdd, $_POST["message"]));

    $insertMessage = "INSERT INTO message(user, message, heure) VALUES ('$nom', '$message', now())"
    mysqli_query($bdd, $insertMessage);
}

```

Résultat :

Résultat dans la base de données :

☐  Éditer  Copier  Supprimer 54 Nathan Bonjour 22:30:25

Affichage sur le site :

22:30:25 - **Nathan:** Bonjour

Si un utilisateur souhaite envoyer un formulaire vide :

Message anonyme interdit! Entrez votre nom.

Veuillez entrer votre Message!.

Nous avons maintenant la possibilité d'envoyer des messages contenant des quotes sans soucis et sans risque d'injection SQL, de plus il est également impossible d'envoyer du code, la faille XSS et donc impossible à réaliser pour une personne mal intentionnée. Une personne ne peut également pas envoyer de message vide et anonyme.

Chapitre 3 : Affichage des messages

Etape :

3.1/ Il faut maintenant s'occuper d'afficher les messages stockés dans la base de données. Pour sa il suffit d'une simple requête pour récupérer l'ensemble des données (nom, message, heure) de la table message :

```

$requete = "SELECT * FROM message";
$requete = mysqli_query($bdd, $requete);

```

Puis nous utilisons une boucle while pour afficher chaque message a l'aide de la fonction mysqli_fetch_assoc() qui va pour chaque tour aller chercher la premier ligne et pointer le curseur sur la ligne d'après.

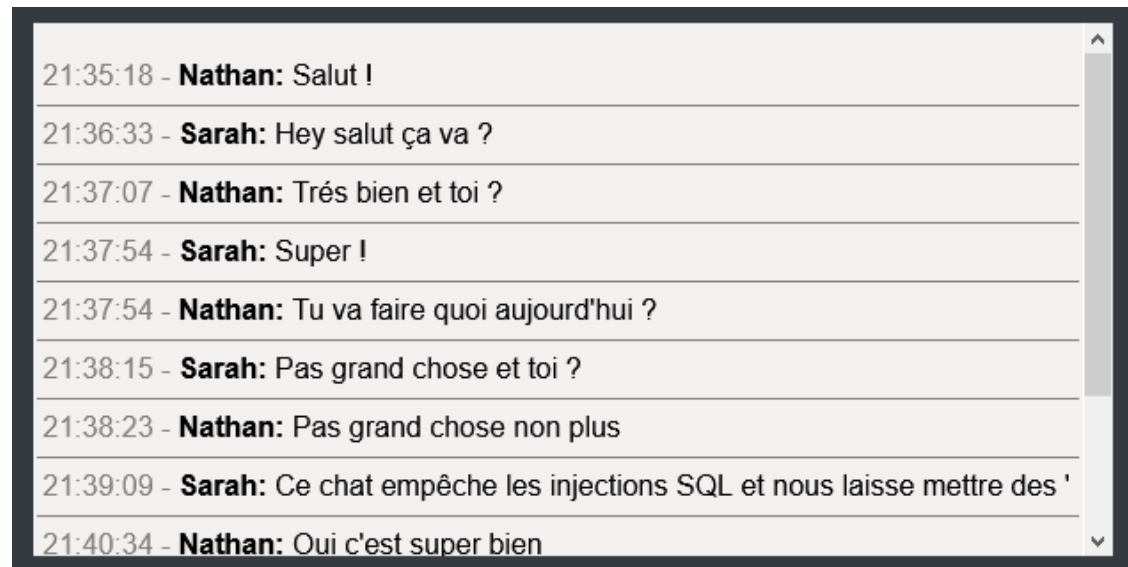
```

<?php while ($afficherMessage = mysqli_fetch_assoc($requete)): ?>
    <li class="li-chat">
        <span class="heure"><?php echo $afficherMessage['heure']; ?> -</span>
        <strong><?php echo $afficherMessage['user']; ?>:</strong>
        <span class="message"><?php echo $afficherMessage['message']; ?></span>
    </li>
    <div class="hr"></div>
<?php endwhile; ?>

```

Résultat :

Le rendu de notre affichage actuel est donc :



Le rendu final du site donne du coup ce résultat :

Chatter'en direct! ChatBox

21:35:18 - **Nathan**: Salut !
21:36:33 - **Sarah**: Hey salut ça va ?
21:37:07 - **Nathan**: Très bien et toi ?
21:37:54 - **Sarah**: Super !
21:37:54 - **Nathan**: Tu va faire quoi aujourd'hui ?
21:38:15 - **Sarah**: Pas grand chose et toi ?
21:38:23 - **Nathan**: Pas grand chose non plus
21:39:09 - **Sarah**: Ce chat empêche les injections SQL et nous laisse mettre des '
21:40:34 - **Nathan**: Oui c'est super bien

Message anonyme interdit! Entrez votre nom.
Veuillez entrer votre Message!.

Entrez votre nom

Ecrivez un message

Envoyer

Notre chat est maintenant fonctionnel.

Conclusion	Que pouvez-vous dire de cette mission : apport personnel, expérience, etc.
	<p>Cette mission ma vraiment permis d’apprendre à bien sécuriser les données envoyées par formulaire pour éviter les injections SQL, faille XSS et surtout éviter de laisser à l’utilisateur d’envoyer un formulaire vide.</p>

Productions associées

Liste des documents produits et description

Aperçu de la structure de la table message :

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/> 1	id	int(11)			Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Plus
<input type="checkbox"/> 2	user	varchar(255)	latin1_swedish_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/> 3	message	varchar(1000)	latin1_swedish_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/> 4	heure	time			Non	Aucun(e)			Modifier Supprimer Plus