


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 système de journalisation
Description du contexte de la mission	Description en 2 à 3 lignes maxi
	Enregistrer toute les connexions et déconnexions au site dans une table de la base de données

Ressources et Outils utilisés	Liste des ressources disponibles et outils utilisés (Documentations, Matériels et Logiciels)
	<ul style="list-style-type: none"> - Sublime Text - WampServer - PhpMyAdmin - Une base de données hopital.sql
Résultat attendu	Résultat attendu avec la réalisation de cette mission L'affichage de toutes les connexions de l'utilisateur X et un système de filtre par date, pour voir les connexions de l'utilisateur X de tel a tel a dates.
Contraintes	Contraintes : techniques budgétaires temps O.S. ou outils imposés...
	<ul style="list-style-type: none"> • Créer une connexion Base de données / Site internet.

Compétences associées
(Voir tableau)

Liste des intitulés du tableau de compétences (avec les références)

A1.1.3 Étude des exigences liées à la qualité attendue d'un service

A1.2.5 Définition des niveaux d'habilitation associés à un service

A2.3.2 Proposition d'amélioration d'un service

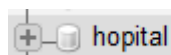
4.1.3 Conception ou adaptation d'une base de données

Description simplifiée des différentes étapes de réalisation de la mission
en mettant en évidence la démarche suivie, les méthodes et les techniques utilisées

Chapitre 1 : Création de la base de données

Etape :

1.1/ Il faut avant tout commencer créer la base de données, pour notre cas on crée la base de données hopital.



Puis dans la base de données hôpital nous allons créer la table user, pour les utilisateurs et connexion pour logger les connexions



2.2/ Dans la table « user » nous allons créer les champs : login en tant que clé primaire de la table, mdp, mail et level afin de définir si l'utilisateur a le droit d'accéder à la page admin.php ou non. Dans la table « connexion » nous allons créer les champs : idc en tant que clé primaire de la table, login, dateDeb, dateFin pour récupérer la date de connexion et la date de déconnexion.

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/> 1	login	varchar(50)	latin1_swedish_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/> 2	mdp	varchar(255)	latin1_swedish_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/> 3	mail	varchar(50)	latin1_swedish_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/> 4	level	int(1)			Non	1			Modifier Supprimer Plus

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/> 1	idc	int(4)			Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Plus
<input type="checkbox"/> 2	login	varchar(50)	latin1_swedish_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/> 3	dateDeb	datetime			Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/> 4	dateFin	datetime			Oui	NULL			Modifier Supprimer Plus

La clé dorée signifie que le champ est un clé primaire.

Chapitre 2 : Début du développement de la partie du projet

Etape :

2.1/ Tout d'abord nous créons le fichier formConnexion.php qui sera la page pour se connecter, qui redirigera l'utilisateur sur la page pagePrivee.php, qui elle pointera sur deux page, admin.php si le level de l'utilisateur est égal à 2 et sur deconnexion.php dans tous les cas. Ensuite il faut relier le site à la base de données, pour ça il faut faire :

```
$bdd = mysqli_connect("localhost", "root", "", "hopital");
```

2.2/ Pour accéder à la page admin.php, il faut se connecter avec son identifiant et son mot de passe

```
<body>
  <center>
    <h1> Connexion </h1>
    <form action="" method="post">
      <?php if(isset($erreur)) echo "<h2> $erreur </h2>"; ?>
      <label for="login">Entrer l'identifiant</label>
      <br><input type="text" id="login" name="login"><br><br>
      <label for="password">Entrer le mot de passe</label>
      <br><input type="password" id="password" name="mdp"><br><br>
      <input type="submit" value="Envoyer" name="envoyer">
    </form>
  </center>
</body>
```

Une fois que l'utilisateur envoie ses données par formulaire nous utilisons le code suivant :

```
<?php
session_start();
if(isset($_POST["envoyer"]))
{
    $login = $_POST["login"];
    $mdp = $_POST["mdp"];
    $bdd = mysqli_connect("localhost", "root", "", "hopital");
    $req = "SELECT * FROM user WHERE login='$login' AND mdp='$mdp'";
    $res = mysqli_query($bdd,$req);
    $ligne = mysqli_fetch_assoc($res);
    if(mysqli_num_rows($res)>0)
    {
        $_SESSION["login"] = $login;
        $_SESSION["level"] = $ligne["level"];
        $req = "INSERT connexion (login, dateDeb) VALUES ('$login',NOW())";
        $res = mysqli_query($bdd,$req);
        echo mysqli_error($bdd);
        $req = "SELECT idc FROM connexion ORDER BY idc DESC LIMIT 1";
        $res = mysqli_query($bdd,$req);
        $ligne = mysqli_fetch_assoc($res);
        $_SESSION["idc"] = $ligne["idc"];
        header("location:pagePrivee.php");
    }
    else
    {
        $erreur = "Désolé, l'identifiant ou le mot de passe saisi n'est pas valide";
    }
}
?>
```

Si le formulaire est envoyé alors on lance une requête SQL pour chercher dans la base de données si l'utilisateur ainsi que son mot de passe est bien enregistré dans la base de données. Si l'utilisateur est bel et bien enregistré alors le nombre que la fonction `mysqli_num_rows` va renvoyer sera supérieur à 0 est donc la condition pourra être exécutée, c'est-à-dire que l'utilisateur va récupérer l'identifiant et le niveau en variable de session, mais va également exécuter une autre requête SQL, celle qui va insérer dans la table connexion, l'identifiant et la date de connexion de l'utilisateur, on aura donc accès à sa date de connexion.

Résultat :

Connexion

Entrer l'identifiant

Entrer le mot de passe

2.3/ Si jamais les identifiant et mot de passe ne correspondent pas on aura le droit a un beau message d'erreur

Connexion

Désolé, l'identifiant ou le mot de passe saisie n'est pas valide

Chapitre 3 : Développement de la deuxième partie du projet

Etape :

3.1/ Une fois connecté nous arrivons sur la page, pagePrivee.php, qui dis à l'utilisateur son login, ainsi que son id de connexion, l'utilisateur aura également le droit à un bouton déconnexion, et si il a les permissions il pourra avoir le lien pour aller sur admin.php.

```
<body>
  <center>
    <h1> Page privée </h1>
    <?php
      if($_SESSION["level"] == 2)
      {
        echo "<a href='admin.php'> Admin </a><br>";
      }
      echo "Bonjour " . $_SESSION["login"] . " Votre id de connexion est " . $_SESSION["idc"];
    ?>
    <hr><br>
    <a href="deconnexion.php"> Deconnexion </a>
  </center>
</body>
```

Si l'utilisateur a le level 2 il peut voir le lien « Admin ».

Resultat :

Page privée

[Admin](#)

Bonjour nathan Votre id de connexion est 36

[Deconnexion](#)

3.2 / Si nous cliquons sur deconnexion, nous allons arriver sur la page deconnexion.php qui va executer ce code PHP :

```
<?php
    session_start();
    $idc = $_SESSION["idc"];
    $bdd = mysqli_connect("localhost", "root", "", "hopital");
    $req = "UPDATE connexion SET dateFin = now() WHERE idc = $idc";
    $res = mysqli_query($bdd, $req);
    session_destroy();
    header("location:formConnexion.php");
?>
```

La base de données va pouvoir rajouter la date de deconnexion de l'utilisateur, et deconnecter L'utilisateur grace à la fonction session_destroy, qui detruit les variables de session.

3.3 / Si nous avons accès à la page admin nous verrons une page avec un champ « rechercher par login » et deux champs « rechercher par date », on a même le droit à 2 boutons radio pour choisir en ordre croissant ou décroissant, on peut donc à partir d'un login afficher toutes les connexions entre la première date choisie et la deuxième :

```
<h1> Recherche par login</h1>
<form action="" method="post">
Login : <input type="text" name="login"><br><br>

<h1> Recherche par date</h1>

Date deb : <input type="date" name="datedeb"><br><br>
Date fin : <input type="date" name="datefin"><br><br>

Tri par ordre :
<input type="radio" name="tri" value="asc" checked> Croissant
<input type="radio" name="tri" value="desc"> Decroissant <br><br>

<input type="submit" value="Envoye" name="ok">
</form><hr>
```

Une fois que l'utilisateur a choisi un login et deux dates nous utilisons le code PHP suivant :

```

<?php
if(isset($_POST["ok"])): ?>
<h1> Table de journalisation </h1>
<table width="30%" border="1">
  <tr>
    <th> Login </th>
    <th>Date début</th>
    <th> Date fin </th>
  </tr>
<?php
$bdd = mysqli_connect("localhost", "root", "", "hopital");
if(isset($_POST["ok"])): ?>
  <?php
  $login = $_POST["login"];
  $tri = $_POST["tri"];
  $dateDeb = $_POST["dateDeb"];
  $dateFin = $_POST["dateFin"];
  if(empty($dateDeb) and empty($dateFin))
  {
    $req = "SELECT * FROM connexion WHERE dateDeb BETWEEN '$dateDeb' AND '$dateFin' AND login LIKE '%$login%' ORDER BY dateDeb $tri";
  }
  else if(empty($dateDeb) and empty($dateFin))
  {
    $req = "SELECT * FROM connexion WHERE login LIKE '%$login%' ORDER BY dateDeb $tri";
  }
  $res = mysqli_query($bdd, $req);
  echo mysqli_error($bdd);
  while($ligne = mysqli_fetch_assoc($res)): ?>
    <tr>
      <td><?=$ligne["login"];?></td>
      <td><?=$ligne["dateDeb"];?></td>
      <td><?=$ligne["dateFin"];?></td>
    </tr>
  <?php endwhile; ?>
<?php endif; ?>
</table>
<?php if(mysqli_num_rows($res)==0) echo "Aucune connexion pour cet utilisateur..."; ?>
<?php endif; ?>

```

Si l'utilisateur a envoyé le formulaire alors on récupère les données du formulaire qu'on Transmet aux variables \$login, \$tri etc... puis on fait une autre condition, si les deux champs ne sont pas vides alors on affiche toute la table connexion de l'utilisateur choisie, entre la date du champ 1 et la du champ 2, sinon si les champs date sont vides on affiche toute les dates sans distinction de l'utilisateur choisit, et si la requête ne retourne aucun résultat, alors on envoie l'erreur « aucune connexion pour cet utilisateur ».

Resultat :

Recherche par login

Login :

Recherche par date

Date deb :

Date fin :

Tri par ordre : ☒ Croissant ☐ Decroissant

Table de journalisation

Login	Date début	Date fin
nathan	2020-12-24 15:32:27	2020-12-24 15:34:54
nathan	2020-12-24 15:42:31	2020-12-24 15:46:11
nathan	2020-12-24 15:46:15	2020-12-24 15:50:32
nathan	2020-12-24 15:51:19	2020-12-24 15:55:38
nathan	2020-12-24 15:57:27	2020-12-24 16:00:19
nathan	2020-12-24 16:01:18	2020-12-24 16:06:12
nathan	2020-12-24 16:06:29	2020-12-24 16:06:32

Notre système de journalisation est maintenant terminé.

Conclusion

Que pouvez-vous dire de cette mission : apport personnel, expérience, etc.

Cette mission m'a permis d'apprendre plus sur le langage SQL, notamment la commande BETWEEN, qui permet d'afficher tous les résultats entre une donnée et une autre.

Productions
associées

Aperçu de la structure de la table connexion et user :

	#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/>	1	idc	int(4)			Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Plus
<input type="checkbox"/>	2	login	varchar(50)	latin1_swedish_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	3	dateDeb	datetime			Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	4	dateFin	datetime			Oui	NULL			Modifier Supprimer Plus

	#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/>	1	login	varchar(50)	latin1_swedish_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	2	mdp	varchar(255)	latin1_swedish_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	3	mail	varchar(50)	latin1_swedish_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	4	level	int(1)			Non	1			Modifier Supprimer Plus