



L'École Normale Supérieure de l'Enseignement Technique de Mohammedia

MOUSSA MOUMEN

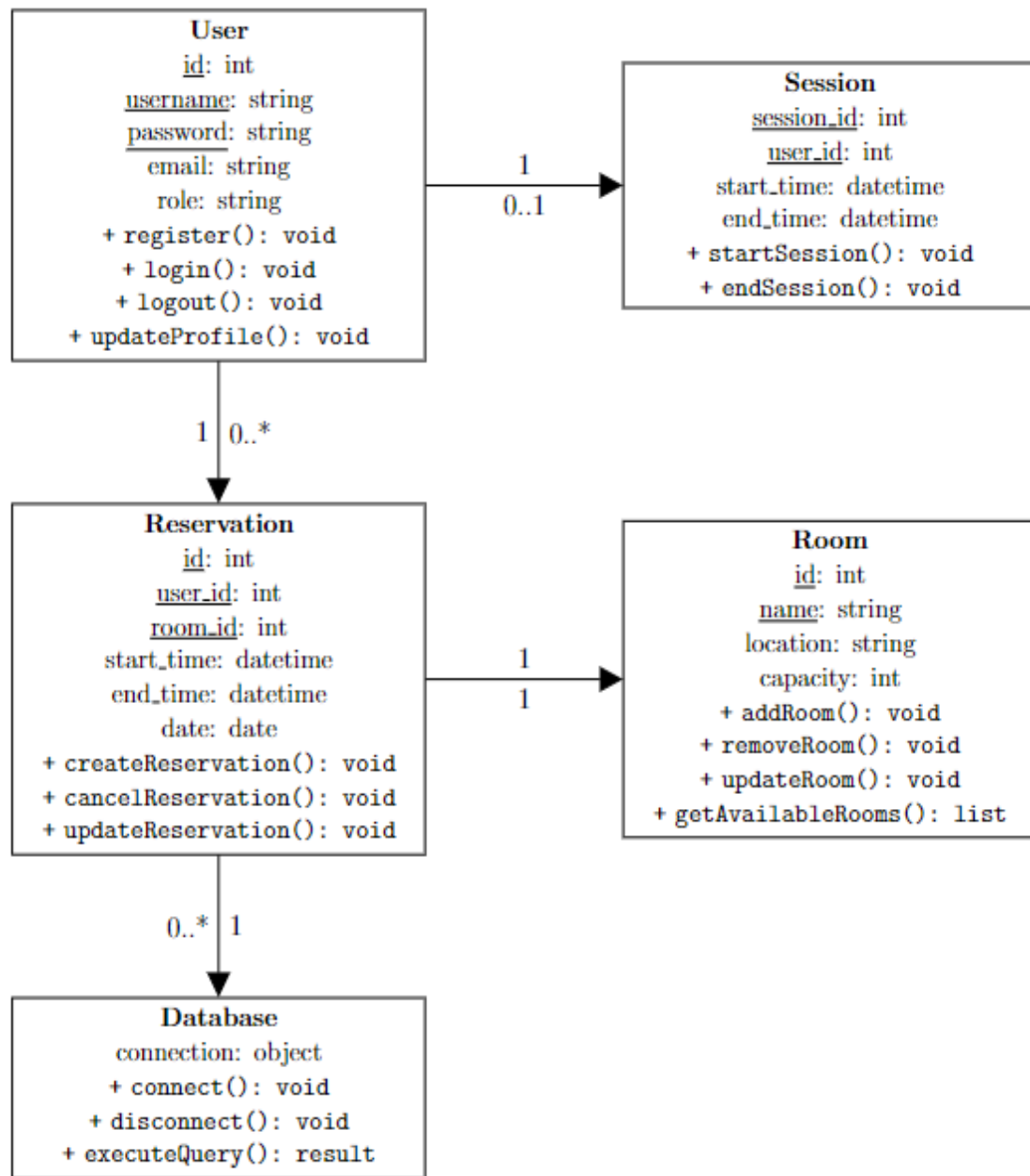
Master : SDIA

Rapport projet technologies de web

La structure du projet

```
reservation-salles-master
├── .idea
├── bdd
│   └── reservationsalles.sql
├── css
│   └── style.css
├── image
│   └── salle.jpg
├── include
│   ├── footer.php
│   ├── header.php
│   ├── php_affiche_d_fin.php
│   ├── php_connexion.php
│   ├── php_index.php
│   ├── php_inscription.php
│   ├── php_planning.php
│   ├── php_profil.php
│   ├── php_reservation.php
│   ├── php_reservation-form.php
│   ├── php_select_date.php
│   ├── php_select_h_debut.php
│   └── php_select_h_fin.php
└── police
    ├── connexion.php
    ├── index.php
    ├── inscription.php
    ├── planning.php
    ├── profil.php
    ├── reservation.php
    └── raservation-form.php
```

Le diagramme de classe



footer.php

Le fichier footer.php contient le code HTML du pied de page (footer) du site web. Il inclut le lien vers la page d'accueil, la page de réservation de salles, et vers mon compte GitHub.

```
<footer>
  <h2><a href="index.php">Accueil</a></h2>
  <h3>Réservation de salles</h3>
```

```
<h2><a href="https://github.com/">GitHub</a></h2>
</footer>
```

header.php

Ce fichier contient le code PHP et HTML du haut de page (header) du site web. Il affiche différents liens en fonction de si l'utilisateur est connecté ou non. S'il est connecté, il affiche des liens vers l'accueil, le planning, la réservation de salle, le profil et un bouton de déconnexion. Sinon, il affiche des liens vers l'accueil, l'inscription, la connexion et le planning.

Listing 1 : header.php

```
<header>
<?php
    if(isset($_SESSION["login"])){
        ?>
        <section id="HG">
            <h2><a href="index.php">Accueil</a></h2>
        </section>
        <section id="HD">
            <h2><a href="planning.php">Planning</a></h2>
            <h2><a href="reservation-form.php">Réserver
            une salle</a></h2>
            <h2><a href="profil.php">Profil</a></h2>
            <form method="POST">
                <input type="submit" name="deco"
                class="bouton" value="Déconnexion"/>
            </form>
        </section>
        <?php
            if(isset($_POST["deco"]))
            {
                session_destroy();
                header("Location:index.php");
            }
        else
        {
            ?>
            <section id="HG">
                <h2><a href="index.php">Accueil</a></h2>
            </section>
            <section id="HD">
                <h2><a href="inscription.php">Inscription</a></h2>
                <h2><a href="connexion.php">Connexion</a></h2>
                <h2><a href="planning.php">Planning</a></h2>
            </section>
        <?php
            }
        ?>
    </header>
```

php_affiche_d_fin.php

Ce fichier PHP génère un champ de saisie de date conditionnel en fonction de la présence de la variable `$_GET["date_debut"]`. Si cette variable est définie, il affiche un champ de saisie de date avec une valeur par défaut définie par `$date_select`. Sinon, il affiche simplement un champ de saisie de date avec la date minimale actuelle.

Listing 2 : php_affiche_d_fin.php

```
<?php
    if(isset($_GET["date_debut"]))
    {
        ?>
        <input type="date" id="fin" name="fin_date"
        min="<?php echo date('Y-m-d')?>" value="<?php
        echo \$date_select;?>" required>
        <small>Créneau d'une heure !</small>
        <?php
    }
    else
    {
        ?>
        <input type="date" id="fin" name="fin_date" min=
        "<?php echo date('Y-m-d')?>"/>
        <small>Créneau d'une heure !</small>
        <?php
    }
?>
```

php_connexion.php

Ce fichier PHP gère l'authentification de l'utilisateur. Si une session existe déjà, l'utilisateur est redirigé vers la page d'accueil. Sinon, si le formulaire de connexion est soumis, les informations de connexion sont vérifiées. Si le login et le mot de passe sont corrects, une session est démarrée et l'utilisateur est redirigé vers la page d'accueil. Si les informations sont incorrectes, un message d'erreur est affiché.

Listing 3 : php_connexion.php

```
<?php
    if(isset($_SESSION["login"]))
    {
        header("Location:index.php");
    }
    else
```

```

{
    if(isset($_POST["validcon"]))
    {
        $login = $_POST["login"];
        $mdp = $_POST["password"];

        $connexionbd = mysqli_connect("localhost", "root", "", "reservationsalles");
        $requete_user = "SELECT * FROM utilisateurs WHERE login='".$login'";
        $query_user = mysqli_query($connexionbd, $requete_user);
        $info_user = mysqli_fetch_all($query_user, MYSQLI_ASSOC);

        if(!empty($info_user))
        {
            if(password_verify($mdp, $info_user[0]["password"]))
            {
                session_start();
                $_SESSION["login"] = $info_user[0]["login"];
                $_SESSION["password"] = $info_user[0]["password"];
                $_SESSION["id"] = $info_user[0]["id"];
                mysqli_close($connexionbd);
                header("Location:index.php");
            }
            else
            {
                $msg_error = "Mauvais mot de passe";
            }
        }
        else
        {
            $msg_error = "Ce login n'est pas valide";
        }
    }
}

?>

```

php_index.php

Ce fichier PHP gère l’affichage du contenu principal de la page d’accueil. Si l’utilisateur est connecté, il affiche une section permettant de réserver une salle avec des instructions, ainsi qu’une image de la salle de réunion. Si l’utilisateur n’est pas connecté, il affiche une invitation à s’inscrire et se connecter pour réserver la salle, ainsi que la même image de la salle de réunion.

Listing 4 : php_index.php

```
<main id="accueil">
<?php
    if(isset($_SESSION["login"]))
    {
        ?>

        <section>
            <a href="reservation-form.php">Réserver la Broadcast
                room</a>
            <p>Créneau <b>d'une heure maximum</b>, n'oubliez pas de
                remplir la description !</p>
            <p><i>Merci de laisser la salle propre et ranger après
                votre passage</i></p>
        </section>

        <section>
            
        </section>
    <?php
    }
    else
    {
        ?>

        <section>
            <p><a href="inscription.php">Inscris</a> toi et <a href=
                "connexion.php">connectes</a> toi pour réserver la
                salle</p>
        </section>
        <section>
            
        </section>
    <?php
    }
?>
</main>
```

php_inscription.php

Ce script PHP gère le processus d'inscription des utilisateurs sur un site de réservation de salles. Tout d'abord, il vérifie si une session utilisateur est déjà active en cherchant la variable `$_SESSION["login"]`. Si une session est active, l'utilisateur est redirigé vers la page d'accueil (`index.php`).

Si aucune session n'est active, le script vérifie si le formulaire d'inscription a été soumis en vérifiant la présence de `$_POST["validinsc"]`.

Si le formulaire a été soumis, le script récupère le login saisi par l'utilisateur et se

connecte à la base de données MySQL. Il vérifie ensuite si le login existe déjà dans la base de données en exécutant une requête SQL.

Si le login n'existe pas, le script vérifie que le mot de passe et sa confirmation correspondent. Si c'est le cas, le mot de passe est haché avec la fonction `password_hash` et un nouvel enregistrement est inséré dans la table utilisateurs de la base de données.

Si l'insertion réussit, l'utilisateur est redirigé vers la page de connexion (`connexion.php`). Si les mots de passe ne correspondent pas, un message d'erreur est généré pour informer l'utilisateur. Si le login existe déjà, un autre message d'erreur est généré pour informer l'utilisateur que le login est déjà pris.

Listing 5 : `php_inscription.php`

```
<?php
    if(isset($_SESSION["login"]))
    {
        header("Location:index.php");
    }
    else
    {
        if(isset($_POST["validinsc"]))
        {
            $login = $_POST["login"];

            $connexionbd = mysqli_connect("localhost", "root", "", "reservationsalles");
            $requete_user = "SELECT * FROM utilisateurs WHERE login='$login'";
            $query_requete_user = mysqli_query($connexionbd, $requete_user);
            $verif_user = mysqli_fetch_all($query_requete_user);

            if(empty($verif_user))
            {
                if($_POST["password"] == $_POST["conf_password"])
                {
                    $mdp = password_hash($_POST["password"], PASSWORD_DEFAULT);

                    $insert_user = "INSERT INTO utilisateurs (login, password) VALUES ('$login', '$mdp')";
                    $query_insert_user = mysqli_query($connexionbd, $insert_user);
                    header("Location:connexion.php");
                }
                else
                {
                    $msg_error = "Les mots de passe ne correspondent pas";
                }
            }
        }
    }
}
```



```

    }
    else
    {
        $msg_error = "Ce login existe déjà";
    }
}

}

?>

```

php_planning.php

Ce script PHP génère un planning hebdomadaire des réservations pour une salle, affichant les heures de 8h à 19h pour chaque jour ouvrable (du lundi au vendredi). Il commence par se connecter à une base de données MySQL et récupère les réservations de la semaine en cours, en joignant les informations des utilisateurs qui ont fait ces réservations. Pour chaque heure de la journée et chaque jour de la semaine, le script génère une cellule dans le tableau. Si une réservation existe pour une heure et un jour donnés, les détails de la réservation (titre et login de l'utilisateur) sont affichés dans la cellule, avec un lien vers la page de détails de la réservation. Si aucune réservation n'existe pour cette heure et ce jour, un lien pour réserver un créneau est affiché.

Listing 6 : php_planning.php

```

<?php
$connexionbd = mysqli_connect("localhost", "root", "", "
    reservationsalles");
$requete_resa = "SELECT * FROM utilisateurs INNER JOIN reservations ON
    utilisateurs.id = reservations.id_utilisateurs WHERE week(debut) =
    week(curdate())";
//sélectionne toutes les réservations de la semaine en cours, en allant
    chercher le login de l'user qui fait la résa.
$query_resa = mysqli_query($connexionbd, $requete_resa);
$info_resa = mysqli_fetch_all($query_resa, MYSQLI_ASSOC);

    for($heure = 8; $heure <= 19; $heure++)//génération lignes des
        heures
        {
            ?>
            <tr>
                <td class="heure"><p><?php echo $heure . "h";?></p>
                </td>

            <?php
            for($jour = 1; $jour<=5; $jour++)//génération des
                cellules sous les jours
                {
                    if(!empty($info_resa))
                    {
                        foreach($info_resa as $resa => $Hresa)//sépare les
                            réservations

```

```

    {
        $JH = explode(" ", $Hresa["debut"]); //sélection la
            ligne correspondant à l'heure de début

        $H = explode(":", $JH[1]); //explose l'heure
        $heure_resa = date("G", mktime($H[0], $H[1], $H[2],
            0, 0, 0)); //récupère uniquement l'heure sans le
            0

        $J = explode("-", $JH[0]); //explose la date
        $jour_resa = date("N", mktime(0, 0, 0, $J[1], $J[2],
            $J[0])); //récupère le numéro du jour

        $case_resa = $heure_resa . $jour_resa; //crée un
            numéro de réservation

        $titre = $Hresa["titre"];
        $login = $Hresa["login"];
        $id = $Hresa["id"];

        $case = $heure . $jour; //Crée un numéro pour chaque
            cellules

        if($case == $case_resa)
        {
            ?>
            <td class="resa"><a href="reservation.php?evenement
                =<?php echo $id;?>"><p><?php echo $titre;?</p><p>
                <?php echo $login; </p></a></td>
            <?php
                break;
            }
            else //si pas de correspondance set $case à null
                pour éviter trop d'affichage
            {
                $case = null;
            }
        }
        if ($case == null)
        {
            ?>
            <td class="case"><a href="reservation-form.php?
                heure_debut=<?php echo $heure;?>&date_debut
                =<?php echo $jour;?>">Réserver un créneau</a></td>
            >
            <?php
                }
            }
            else
            {
                ?>
                <td class="case"><a href="reservation-form.php?
                    heure_debut=<?php echo $heure;?>&date_debut
                    =<?php echo $jour;?>">Réserver un créneau</a></

```

```

        td>
            <?php
            }
        }
    ?>
</tr>
<?php
}
?>

```

php_profil.php

Ce script PHP gère la modification des informations de profil des utilisateurs, en particulier le login et le mot de passe. Tout d'abord, il vérifie si l'utilisateur est connecté en contrôlant l'existence de la variable `$_SESSION["login"]`. Si l'utilisateur n'est pas connecté, il est redirigé vers la page d'accueil (`index.php`). Si l'utilisateur est connecté, le script vérifie si le formulaire de modification a été soumis en regardant si `$_POST["validmod"]` est défini et que les champs login et `old_password` ne sont pas vides. Si l'ancien mot de passe fourni par l'utilisateur est correct (vérifié à l'aide de `password_verify`), le script se connecte à la base de données MySQL et vérifie si le nouveau login existe déjà. Si le nouveau login est disponible ou identique à l'ancien, il met à jour le login de l'utilisateur. Si l'utilisateur a également fourni un nouveau mot de passe, le script vérifie que le mot de passe de confirmation correspond au nouveau mot de passe, puis hache le nouveau mot de passe et met à jour la base de données. Des messages d'erreur ou de succès sont générés en fonction des résultats des différentes vérifications et mises à jour.

Listing 7 : php_profil.php

```

<?php
    if(!isset($_SESSION["login"]))
    {
        header("Location:index.php");
    }
    else
    {
        if(isset($_POST["validmod"]) && !empty($_POST["login"]) && !
            empty($_POST["old_password"]))
        {
            if(password_verify($_POST["old_password"], $_SESSION
                ["password"]))
            {
                $login = $_POST["login"];
                $mdp = $_POST["nw_password"];
                $id = $_SESSION["id"];

                $connexionbd = mysqli_connect("localhost", "root"
                    , "", "reservationsalles");
                $requete_user = "SELECT * FROM utilisateurs WHERE
                    login='$login'";
            }
        }
    }
}

```

```

$query_requete = mysqli_query($connexionbd,
    $requete_user);
$info_user = mysqli_fetch_all($query_requete,
    MYSQLI_ASSOC);

    if(empty($info_user) || $login == $_SESSION[
        "login"])
    {
        $update_login = "UPDATE utilisateurs SET login='
            $login' WHERE id='$id'";
        $query_login = mysqli_query($connexionbd,
            $update_login);
        $_SESSION["login"] = $login;
        $msg_login = "Login modifié";
    }
    else
    {
        $msg_error = "Ce login est déjà pris";
    }
    if(isset($mdp) && !empty($mdp))
    {
        if($mdp ==
            $_POST["conf_password"])
        {
            $mdp_hash = password_hash($mdp, PASSWORD_DEFAULT
            );

            $update_mdp = "UPDATE utilisateurs SET password
                ='$mdp_hash' WHERE id='$id'";
            $query_mdp = $mysqli_query($connexionbd,
                $update_mdp);
            $_SESSION["password"] = $mdp_hash;
            $msg_mdp = "Mot de passe modifié";
        }
        else
        {
            $msg_error = "Les mots de passe ne
                correspondent pas";
        }
    }
}
else
{
    $msg_error = "Mauvais mot de passe";
}
}

```

?>

php_reservation.php

Ce script PHP permet d'afficher les détails d'une réservation spécifique pour un utilisateur connecté. Il commence par vérifier si une session utilisateur est active via `$_SESSION["login"]`. Si l'utilisateur n'est pas connecté, il est redirigé vers la page de planning (`planning.php`). Si l'utilisateur est connecté et qu'un identifiant d'événement (`evenement`) est passé via la méthode GET, le script récupère les informations de la réservation correspondante à partir de la base de données MySQL. Pour ce faire, il se connecte à la base de données et exécute une requête pour joindre les tables `reservations` et `utilisateurs` afin de récupérer les détails de la réservation (titre, login de l'utilisateur, description, heure de début et heure de fin). Les informations récupérées sont ensuite formatées pour être affichées ultérieurement sur la page.

Listing 8 : php_reservation.php

```
<?php
    if(isset($_SESSION["login"]))
    {
        if(isset($_GET["evenement"]) && !empty($_GET["evenement"]))
        {
            $id = $_GET["evenement"];

            $connexionbd = mysqli_connect("localhost", "root", "
            ", "reservationsalles");
            $requete = "SELECT * FROM reservations INNER JOIN
            utilisateurs ON utilisateurs.id = reservations.
            id_utilisateurs WHERE reservations.id = '$id'";
            $query = mysqli_query($connexionbd, $requete);
            $resa = mysqli_fetch_all($query, MYSQLI_ASSOC);

            $titre = $resa[0]['titre'];
            $login = $resa[0]['login'];
            $description = $resa[0]['description'];

            $debut = explode(" ", $resa[0]['debut']);

            $H = explode(":", $debut[1]);
            $heure_debut = $H[0] . ":" . $H[1]; //récupère
            seulement l'heure dans début

            $J = explode("-", $debut[0]);
            $jour = $J[2] . "-" . $J[1] . "-" . $J[0]; //récupère
            seulement la date dans début, mais formater j-m
            -a

            $fin = explode(" ", $resa[0]['fin']);

            $HF = explode(":", $fin[1]);
            $heure_fin = $HF[0] . ":" . $HF[1];
        }
    }
else
```

```

    {
        header("Location:planning.php");
    }
?>

```

php_reservation-form.php

vérifier si un utilisateur est connecté en vérifiant l'existence de `$_SESSION["login"]`. Si l'utilisateur n'est pas connecté, il est redirigé vers la page d'accueil (`index.php`). Si l'utilisateur est connecté et que le formulaire de réservation est soumis (`$_POST["validresa"]` est défini et tous les champs requis sont remplis), le script procède à différentes vérifications avant d'ajouter la réservation à la base de données. Les vérifications incluent :

- Vérification de la disponibilité du créneau choisi.
- Vérification de la disponibilité du créneau choisi.
- Vérification que la date de début n'est pas déjà passée.
- Vérification que la date de fin est le même jour que la date de début.
- Vérification que la durée de la réservation est d'une heure.
- Vérification que la durée de la réservation est d'une heure.
- Vérification que la réservation ne concerne pas un jour de week-end.

En fonction des résultats de ces vérifications, des messages d'erreur ou de succès sont générés pour informer l'utilisateur du résultat de sa tentative de réservation.

Listing 9 : php_reservation-form.php

```

<?php
    if(!isset($_SESSION["login"]))
    {
        header("Location:index.php");
    }
    else
    {
        if(isset($_POST["validresa"]) && !empty($_POST["titre"]) &&
            !empty($_POST["description"]) && !empty($_POST["
            debut_date"]) && !empty($_POST["debut_heure"]) && !empty(
            $_POST["fin_date"]) && !empty($_POST["fin_heure"]))
        {
            $debut = $_POST["debut_date"] . " " . $_POST["
                debut_heure"];
            $fin = $_POST["fin_date"] . " " . $_POST["fin_heure"
                ];
            $titre = $_POST["titre"];
            $description = $_POST["description"];
            $debut_str = strtotime($debut);

```

```

$fin_str = strtotime($fin);
$id = $_SESSION["id"];

//vérifie que l'heure de début choisie n'est pas
déjà enregistré
$connexionbd = mysqli_connect("localhost", "root", "
", "reservationsalles");
$requete_creneau = "SELECT * FROM reservations WHERE
debut='$debut'";
$query_creneau = mysqli_query($connexionbd,
    $requete_creneau);
$info_creneau = mysqli_fetch_all($query_creneau,
    MYSQLI_ASSOC);

//check si le jour n'est pas week-end
$semaine = explode("-", $_POST["debut_date"]);
$jour = date("N", mktime(0, 0, 0, $semaine[1],
    $semaine[2], $semaine[0]));

if(empty($info_creneau))
{
    if($debut_str < time()) //check date saisie pour
        début n'est pas déjà passée
        //time -2 heures !!!!!
        {
            $msg_error = "L'heure est déjà passée";
        }
        else
        {
            if($_POST["debut_date"] == $_POST["fin_date"])//
                vérifie que c'est le même jour
                {
                    $time_debut= explode(':', $_POST["debut_heure"]);//
                        transforme l'heure en tableau 2 entrées
                    $time_fin = explode(':', $_POST["fin_heure"]);

                    if($fin_str < $debut_str) //check si date de fin n'est
                        pas avant début
                        {
                            $msg_error = "La fin doit être le même jour";
                        }
                        else if($time_fin[0] - $time_debut[0] == 1)//regarde
                            si le créneau dure 1h
                            {
                                if($jour <= 5) //vérifie que le jour n'est pas week-end
                                    {
                                        $ajout = "INSERT INTO reservations (titre, description,
                                            debut, fin, id_utilisateurs) VALUES ('$titre',
                                            '$description', '$debut', '$fin', '$id')";
                                        $query_ajout = mysqli_query($connexionbd, $ajout);
                                        $msg_valid = "réservation prise en
                                            compte";
                                    }
                                    else

```

```

        {
            $msg_error = "Pas de réservation le week-end";
        }
    }
    else
    {
        $msg_error = "Le créneau doit faire 1h";
    }
    else
    {
        $msg_error = "Date de début et de fin
différente";
    }
}
}
else
{
    $msg_error = "Créneau déjà prit";
}
}
}
?>

```

php_select_date.php

Ce script PHP génère un champ de saisie de date en HTML en fonction de la valeur de `$_GET["date_debut"]`. Si `$_GET["date_debut"]` est défini, il récupère la valeur de `$_GET["date_debut"]` et génère un champ de saisie de date avec une valeur pré-remplie correspondant au début de la semaine en cours en fonction du jour sélectionné (1 pour lundi, 2 pour mardi, etc.). Si `$_GET["date_debut"]` n'est pas défini, il génère un champ de saisie de date avec la date actuelle comme valeur par défaut.

Listing 10 : php_select_date.php

```

<?php
if(isset($_GET["date_debut"]))
{
    $date_debut = $_GET["date_debut"];

    if($date_debut ==1)
    {
        $date_select = date( 'Y-m-d', strtotime('monday this
week'));
    }
    if($date_debut ==2)
    {
        $date_select = date( 'Y-m-d', strtotime('tuesday this
week'));
    }
    if($date_debut ==3)

```



```

        {
            $date_select = date( 'Y-m-d', strtotime('wednesday this
                                week'));
        }
        if($date_debut ==4)
        {
            $date_select = date( 'Y-m-d', strtotime('thursday this
                                week'));
        }
        if($date_debut ==5)
        {
            $date_select = date( 'Y-m-d', strtotime('friday this
                                week'));
        }
        ?>
        <input type="date" id="debut" name="debut_date" min = "<?php
            echo date('Y-m-d')?>" value="<?php echo $date_select;?>"
            required>

        <?php
    }
    else
    {
        ?>
        <input type="date" id="debut" name="debut_date" min = "<?php
            echo date('Y-m-d')?>" />
        <?php
    }
    ?>

```

php_select_h_debut.php

Ce script PHP génère des options pour un menu déroulant en HTML, représentant les heures disponibles pour la réservation d'une salle. Les heures vont de 8h à 19h incluses. Les options sont générées en fonction de la valeur de `$_GET["heure_debut"]`. Si cette valeur est définie, l'option correspondant à cette heure est sélectionnée par défaut dans le menu déroulant. Sinon, la première heure (8h) est sélectionnée par défaut.

Listing 11 : php_select_h_debut.php

```

<?php
if(isset($_GET["heure_debut"]))
{
    for($heure_select = 8; $heure_select<=19; $heure_select++)
    {
        if($heure_select < 10)
        {
            ?>
            <option value="<?php echo "0" . $heure_select . ":00
                ";?>" <?php if ($heure_select == $_GET["
                heure_debut"]){echo "selected";} ?>><?php echo "0
                " . $heure_select . ":00";?></option>

```

```

        <?php
    }
    else
    {
        ?>
        <option value="<?php echo $heure_select . ":00";?>"
            <?php if($heure_select == $_GET["heure_debut"]){
                echo "selected";}?>><?php echo $heure_select . "
                :00";?></option>
        <?php
    }
}
}
else
{
    for($heure_select = 8; $heure_select<=19; $heure_select++)
    {
        if($heure_select < 10)
        {
            ?>
            <option value="<?php echo "0" . $heure_select . "
                :00";?>"><?php echo "0" . $heure_select . "
                :00";?></option>
            <?php
        }
        else
        {
            ?>
            <option value="<?php echo $heure_select . ":00"
                ;?>"><?php echo $heure_select . ":00";?></
                option>
            <?php
        }
    }
}
?>

```

php_select_h_fin.php

Ce script PHP génère des options pour un menu déroulant en HTML, représentant les heures disponibles pour la fin d'une réservation de salle. Les heures vont de 9h à 20h incluses. Les options sont générées en fonction de la valeur de `$_GET["heure_debut"]`. Si cette valeur est définie, l'option correspondant à l'heure suivant immédiatement l'heure de début est sélectionnée par défaut dans le menu déroulant. Sinon, la première heure (9h) est sélectionnée par défaut.

Listing 12 : php_select_h_fin.php

```

<?php
if(isset($_GET["heure_debut"]))
{

```

```

        for($heure_fin = 9; $heure_fin<=20; $heure_fin++)
        {
            if($heure_fin < 10)
            {
                ?>
                <option value="<?php echo "0" . $heure_fin . "
                :00";?>" <?php if ($heure_fin == $_GET["
                heure_debut"]+1){echo "selected";} ?><?php
                echo "0" . $heure_fin . ":00";?></option>
                <?php
            }
            else
            {
                ?>
                <option value="<?php echo $heure_fin . ":00";?>"
                <?php if ($heure_fin == $_GET["heure_debut"
                ]+1){echo "selected";} ?><?php echo
                $heure_fin . ":00";?></option>
                <?php
            }
        }
    }
else
{
    for($heure_fin = 9; $heure_fin<=20; $heure_fin++)
    {
        if($heure_fin < 10)
        {
            ?>
            <option value="<?php echo "0" . $heure_fin . "
            :00";?>"<?php echo "0" . $heure_fin . ":00"
            ;?></option>
            <?php
        }
        else
        {
            ?>
            <option value="<?php echo $heure_fin . ":00";?>"
            ><?php echo $heure_fin . ":00";?></option>
            <?php
        }
    }
}
?>

```

connexion.php

Ce code PHP représente la page de connexion du site web. Il inclut un formulaire permettant aux utilisateurs de saisir leur login et leur mot de passe pour se connecter. Si des erreurs surviennent lors de la tentative de connexion, un message d'erreur est affiché. Il y a également un lien vers la page d'inscription pour les utilisateurs qui n'ont pas encore de compte.

Listing 13 : connexion.php

```
<?php
    session_start();
    include 'include/php_connexion.php';
?>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="css/style.css"/>
    <title>Connexion</title>
</head>
<body>
    <?php include 'include/header.php';?>

    <main>
        <h1>Formulaire de Connexion</h1>
        <form action="connexion.php" method="POST" id="form_co">
            <label for="login">Login :</label>
            <input type="text" id="login" name="login" required/>

            <label for="password">Mot de passe :</label>
            <input type="password" id="password" name="password"
                required>

            <input type="submit" name="validcon" class="bouton"
                value="Connexion">

            <?php
                if(isset($msg_error))
                {
                    ?>
                        <p class="msg_error">

            <?php
                echo $msg_error;
            ?>
                </p>

            <?php
                }
            ?>
        </form>
        <p>Tu n'as pas encore de comptes ? => <a href="inscription.
            php">inscris toi !</a></p>
    </main>

    <?php include 'include/footer.php';?>
</body>
</html>
```

index.php

Ce script PHP représente la page d'accueil du site web. Il démarre une session PHP, puis inclut les fichiers PHP nécessaires pour afficher l'en-tête, le contenu principal et le pied de page de la page d'accueil.

Listing 14 : index.php

```
<?php session_start(); ?>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="css/style.css"/>
    <title>Accueil</title>
</head>
<body>
    <?php
        include 'include/header.php';

        include 'include/php_index.php';

        include 'include/footer.php';
    ?>
</body>
</html>
```

inscription.php

Ce script PHP représente la page d'inscription du site web. Il démarre une session PHP, puis inclut le fichier PHP contenant le traitement de l'inscription. Ensuite, il génère une page HTML avec un formulaire permettant aux utilisateurs de saisir leur login et leur mot de passe pour créer un nouveau compte. Le formulaire demande également une confirmation du mot de passe. Si des erreurs surviennent lors de la tentative d'inscription, un message d'erreur est affiché.

Listing 15 : inscription.php

```
<?php
    session_start();
    include 'include/php_inscription.php';
?>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="css/style.css"/>
    <title>Inscription</title>
</head>
```

```

<body>
  <?php include 'include/header.php';?>

  <main>
    <h1>Formulaire d'inscription</h1>
    <form action="inscription.php" method="POST" class="form_pir">
      <label for="login">Login :</label>
      <input type="text" id="login" name="login" required/>

      <label for="password">Mot de passe :</label>
      <input type="password" id="password" name="password"
        required>

      <label for="conf_password">Confirmer Mot de passe :</label>
      <input type="password" id="conf_password" name="
        conf_password" required>

      <input type="submit" name="validinsc" class="bouton" value="
        Inscription">

      <?php
        if(isset($msg_error))
        {
          ?>
            <p class="msg_error">

          <?php
            echo $msg_error;
          ?>
            </p>

          <?php
            }
          ?>
        </form>
      </main>

      <?php include 'include/footer.php';?>
    </body>
  </html>

```

planning.php

Ce script PHP génère la page de planning d'une semaine pour le site web. Il démarre une session PHP, puis inclut le fichier PHP contenant la logique de génération du planning. Ensuite, il crée une page HTML avec un tableau affichant les jours de la semaine et leurs dates correspondantes. Le planning des réservations est inclus dans le corps du tableau.

Listing 16 : planning.php

```

<?php session_start(); ?>
<!DOCTYPE html>
<html>

```

```

<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="css/style.css"/>
    <title>Planning</title>
</head>
<body>
    <?php include 'include/header.php';?>

    <main>
        <h1>Planning <?php echo $jour_semaine = date('Y', time());?></h1>
        <h2>Semaine <?php echo $jour_semaine = date('W', time());?></h2>
        <table>
            <thead>
                <tr>
                    <th class="vide"></th>
                    <th class="jour">Lundi <?php echo $jour_semaine =
                        date('d/m', strtotime('monday this week'));?></th>
                    <th class="jour">Mardi <?php echo $jour_semaine =
                        date('d/m', strtotime('tuesday this week'));?></th>
                    <th class="jour">Mercredi <?php echo $jour_semaine =
                        date('d/m', strtotime('wednesday this week'));?>
                        </th>
                    <th class="jour">Jeudi <?php echo $jour_semaine =
                        date('d/m', strtotime('thursday this week'));?></th>
                    <th class="jour">Vendredi <?php echo $jour_semaine =
                        date('d/m', strtotime('friday this week'));?></th>
                </tr>
            </thead>
            <tbody>
                <?php include 'include/php_planning.php';?>
            </tbody>
        </table>
    </main>

    <?php include 'include/footer.php';?>
</body>
</html>

```

profil.php

Ce script PHP génère la page de profil d'utilisateur pour le site web. Il démarre une session PHP, puis inclut le fichier PHP contenant la logique de gestion du profil. Ensuite, il crée une page HTML avec un formulaire permettant à l'utilisateur de modifier son login et son mot de passe. Les champs de ce formulaire sont pré-remplis avec les valeurs actuelles du login de l'utilisateur. L'utilisateur peut saisir son ancien mot de passe, son nouveau mot de passe et le confirmer. Lorsque l'utilisateur soumet le formulaire, les données sont

envoyées à la même page (profil.php) pour traitement. Les éventuels messages de succès ou d'erreur sont affichés en fonction des opérations effectuées.

Listing 17 : profil.php

```
<?php
    session_start();
    include 'include/php_profil.php';
?>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="css/style.css"/>
    <title>Profil</title>
</head>
<body>
    <?php include 'include/header.php';?>

    <main>
        <form action="profil.php" method="POST" class="form_pir">
            <label for="login">Login :</label>
            <input type="text" id="login" name="login" required value="<?
                php echo $_SESSION['login'];?>" />

            <label for="old_password">Ancien mot de passe :</label>
            <input type="password" id="old_password" name="old_password"
                required>

            <label for="nw_password">Nouveau mot de passe :</label>
            <input type="password" id="nw_password" name="nw_password">

            <label for="conf_password">Confirmer mot de passe :</label>
            <input type="password" id="conf_password" name="
                conf_password">

            <input type="submit" name="validmod" class="bouton" value="
                Modifier">

        <?php
            if(isset($msg_login))
            {
                echo "<span class='msg_'>" . $msg_login . "
                    </span><br/>";
            }
            if(isset($msg_mdp))
            {
                echo "<span class='msg_'>" . $msg_mdp . "</
                    span><br/>";
            }
            if(isset($msg_error))
            {
                echo "<span class='msg_'>" . $msg_error . "
                    </span>";
            }
        }
    }
</main>
</body>
</html>
```



```

        }

        ?>

    </form>
</main>

<?php include 'include/footer.php';?>
</body>
</html>

```

reservation.php

Ce script PHP génère une page de réservation pour un utilisateur connecté. Il démarre une session PHP, puis inclut le fichier PHP contenant la logique de gestion de la réservation. Ensuite, il crée une page HTML avec les détails de la réservation, tels que le login de l'utilisateur, la date et l'heure de début et de fin de la réservation, ainsi que l'intitulé et la description de la réservation. Ces informations sont récupérées à partir des variables PHP définies dans le fichier `php_reservation.php`.

Listing 18 : reservation.php

```

<?php
    session_start();
    include 'include/php_reservation.php';
?>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="css/style.css"/>
    <title>Réservation</title>
</head>
<body>
    <?php include 'include/header.php';?>

    <main>
        <section id="reservation">
            <h1>Réserver par <u><?php echo $login;?></u></h1>
            <p>Le <?php echo $jour?> de <?php echo $heure_debut;?> à <?php echo $heure_fin;?></p>
            <section class="info_resa">
                <p><u>Intitulé</u> :</p>
                <?php echo $titre;?>
            </section>
            <hr>
            <section class="info_resa">
                <p><u>Description</u> :</p>
                <?php echo $description;?>
            </section>

        </section>
    </main>

```

```
<?php include 'include/footer.php';?>
</body>
</html>
```

reservation-form.php

Ce script PHP génère une page de formulaire de réservation pour la salle "Broadcast room". Il démarre une session PHP, puis inclut le fichier PHP contenant la logique de gestion du formulaire de réservation. Ensuite, il crée une page HTML avec un formulaire permettant à l'utilisateur de saisir les détails de la réservation, tels que le titre, la description, la date et l'heure de début et de fin de la réservation. Ces informations sont envoyées au script reservation-form.php lors de la soumission du formulaire. Le formulaire comprend également des listes déroulantes pour sélectionner les heures de début et de fin, ainsi qu'un message d'erreur qui s'affiche si la réservation ne peut pas être effectuée.

Listing 19 : reservation-form.php

```
<?php
    session_start();
    include 'include/php_reservation-form.php';
?>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="css/style.css"/>
    <title>Réserver une salle</title>
</head>
<body>
    <?php include 'include/header.php';?>

    <main>
        <h1>Réserver la Broadcast room</h1>
        <form action="reservation-form.php" method="POST" class="
            form_pir">
            <label for="titre">Titre <span class="oblig">*</span> :</
                label>
            <select id="titre" name="titre" required>
                <option value="How to">How to</option>
                <option value="Anglais">Anglais</option>
                <option value="Pitch">Pitch</option>
                <option value="Conférence">Conférence</option>
                <option value="Français">Français</option>
                <option value="Réunion">Réunion</option>
            </select>

            <label for="description">Description <span class="oblig">*</
                span>: </label>
            <input type="text" id="description" name="description">
```

```

        required>

        <label for="debut">Date et heure de début <span class="oblig
            ">*</span> :</label>
        <?php include 'include/php_select_date.php';?>

        <select id="debut" name="debut_heure" required>
        <?php include 'include/php_select_h_debut.php';?>
        </select>

        <label for="fin">Date et heure de fin <span class="oblig">*<
            /span> :</label>
        <?php include 'include/php_affiche_d_fin.php';?>

        <select id="fin" name="fin_heure" required>
        <?php include 'include/php_select_h_fin.php';?>
        </select>

        <small class="oblig">*</small> Champ obligatoire</small>

        <input type="submit" name="validresa" class="bouton" value="
            Réserver">

        <?php
            if(isset($msg_error))
            {
                echo "<span class='msg_'>" . $msg_error . "</
                    span><br/>";
            }
            if(isset($msg_valid))
            {
                echo "<span class='msg_'>" . $msg_valid . "</
                    span><br/>";
            }
        ?>
    </form>
</main>

    <?php include 'include/footer.php';?>
</body>
</html>

```

Le code CSS

Listing 20 : style.css

```

@font-face
{
    font-family: "Roboto";
    src: url('../police/Roboto-Light.ttf');
}

```

```

    }
@font-face
{
    font-family: "Roboto";
    font-style: italic;
    src: url('../police/Roboto-Italic.ttf');
}
@font-face
{
    font-family: "Roboto";
    font-weight: bold;
    src: url('../police/Roboto-Bold.ttf');
}
@font-face
{
    font-family: "SpaceMono";
    src: url('../police/SpaceMone-Regular.ttf');
}
@font-face
{
    font-family: "SpaceMono";
    font-style: italic;
    src: url('../police/SpaceMono-Italic.ttf');
}
@font-face
{
    font-family: "SpaceMono";
    font-weight: bold;
    src: url('../police/SpaceMono-Bold.ttf');
}
html
{
    margin: 0;
    height: 100%;
}
body
{
    display: flex;
    flex-direction: column;
    margin: 0;
    height: 100%;
}
a
{
    text-decoration: none;
    color: black;
}
h1
{
    font-family: "SpaceMono";
}
h2
{
    font-family: "SpaceMono";
}

```

```

    }
/* HEADER */
header
{
    display: flex;
    flex-direction: row;
    align-items: center;
    justify-content: space-evenly;
    background-color: #88c0d0;
    font-family: "SpaceMono";
}
#HG
{
    display: flex;
    flex-grow: 1;
    padding-left: 2vw;
}
#HD
{
    display: flex;
    flex-direction: row;
    flex-grow: 3;
    justify-content: space-evenly;
    align-items: center;
}
header a:hover
{
    color: #ffca4b;
}
/* MAIN */
main
{
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    flex-grow: 1;
    font-family: "Roboto";
}
main a:hover
{
    color: #275fa0;
    text-decoration: underline;
}
#accueil
{
    display: flex;
    flex-direction: row;
    justify-content: space-evenly;
    align-items: center;
}
#accueil section
{
    display: flex;

```

```

        flex-direction: column;
        align-items: center;
    }
#accueil a
{
    font-weight: bold;
}
#reunion
{
    width: 50vw;
    height: 70vh;
}
#reservation
{
    display: flex;
    flex-direction: column;
    justify-content: space-evenly;
    align-items: center;
    width: 40vw;
    height: 55vh;
    background-color: #88c0d0;
    padding-bottom: 5vh;
}
.info_resa
{
    display: flex;
    flex-direction: column;
    justify-content: space-evenly;
    align-items: center;
}
hr
{
    width: 5vw;
    border: 1px dashed black;
    margin-bottom: 0;
}
/* form */
form
{
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: space-evenly;
}
input
{
    text-align: center;
}
label
{
    font-weight: bold;
}
#deco
{

```

```

        font-family: "Roboto";
    }
    .form_pir
    {
        width: 40vw;
        height: 60vh;
        background-color: #88c0d0;
        margin-bottom: 5vh;
    }
    #form_co
    {
        width: 40vw;
        height: 40vh;
        background-color: #88c0d0;
        margin-bottom: 5vh;
    }
    .msg_
    {
        color: red;
    }
    .oblig
    {
        color: red;
    }
    .bouton
    {
        border-radius: 10px;
        border: 1px solid black;
    }
    .bouton:hover
    {
        border: 1px solid #ffca4b;
    }
    /* tableau */
    table
    {
        display: flex;
        flex-direction: column;
        border: 1px solid black;
        height: 90%;
        width: 70vw;
        margin-bottom: 10vh;
        border-spacing: 0;
        box-shadow: 2px 2px 10px #275fa0;
    }

    thead
    {
        display: flex;
        flex-direction: column;
        width: 100%;
        height: 10%;
        justify-content: space-around;
        align-items: center;
    }

```

```

    }

thead tr
{
    display: flex;
    flex-direction: row;
    justify-content: space-evenly;
    width: 100%;
    border-bottom: 1px solid black;
}
thead th
{
    border-left: 1px solid black;
    display: flex;
    flex-direction: row;
    width: 100%;
    justify-content: space-around;
    align-items: center;
}
tbody tr
{
    display: flex;
    flex-direction: row;
    justify-content: space-evenly;
    width: 100%;
    border-bottom: 1px solid black;
}

td
{
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    flex-grow: 1;
    width: 60vw;
    border-left: 1px solid black;
}

td p
{
    margin: 0;
}

.heure
{
    width: 15vw;
    background-color: #88c0d0;
    font-weight: bold;
}

.case
{
    width: 35vw;
    height: 15vh;
}

.resa

```



```

    {
        width: 35vw;
        height: 15vh;
        background-color: #275fa0;
    }
.resa a
{
    color: #ffca4b;
    display: flex;
    flex-direction: column;
    align-items: center;
}
.resa a:hover
{
    color: white;
    text-decoration: none;
}
.vide
{
    width: 30vw;
    background-color: #275fa0;
}
.jour
{
    background-color: #ffca4b ;
}
/* FOOTER */
footer
{
    display: flex;
    flex-direction: row;
    justify-content: space-evenly;
    align-items: center;
    background-color: #ffca4b;
    font-family: "SpaceMono";
}
footer a:hover
{
    color: #88c0d0;
}

```

reservationsalles.sql

Listing 21 : reservationsalles.sql

```

Dump
-- version 4.9.2
-- https://www.phpmyadmin.net/
--
-- Hôte : 127.0.0.1:3306
-- Généré le : ven. 26 mai 2024 à 23:38

```

```

-- Version du serveur : 10.4.10-MariaDB
-- Version de PHP : 7.3.12

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Base de données : `reservationsalles`
--

--
-- Structure de la table `reservations`
--

DROP TABLE IF EXISTS `reservations`;
CREATE TABLE IF NOT EXISTS `reservations` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `titre` varchar(255) NOT NULL,
  `description` text NOT NULL,
  `debut` datetime NOT NULL,
  `fin` datetime NOT NULL,
  `id_utilisateurs` int(11) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM AUTO_INCREMENT=27 DEFAULT CHARSET=latin1;

--
-- Déchargement des données de la table `reservations`
--

INSERT INTO `reservations` (`id`, `titre`, `description`, `debut`, `fin`, `id_utilisateurs`) VALUES
(20, 'Pitch', 'Pitch reservation salles', '2020-06-12 12:00:00', '2020-06-12 13:00:00', 1),
(19, 'How to', 'Garder son calme en codant', '2020-06-11 17:00:00', '2020-06-11 18:00:00', 1),
(18, 'How to', 'cours de la premi re promo', '2020-06-11 19:00:00', '2020-06-11 20:00:00', 1),
(22, 'R union', 'Important ! D but projet pro', '2020-06-12 18:00:00', '2020-06-12 19:00:00', 1),
(23, 'Conf rence', 'V n rer moi', '2020-06-16 13:00:00', '2020-06-16 14:00:00', 1),
(24, 'Pitch', 'Pr sentation application r volutionnaire', '2020-06-16 11:00:00', '2020-06-16 12:00:00', 1),
(25, 'How to', 'Ecoutez moi je suis ', '2020-06-16 10:00:00', '2020-06-16 11:00:00', 1);

```

```

2020-06-16 11:00:00', 3),
(26, 'How to', 'Bien se prÃ©senter', '2020-06-18 10:00:00', '2020-06-18
11:00:00', 3);

-----

--
-- Structure de la table `utilisateurs`
--

DROP TABLE IF EXISTS `utilisateurs`;
CREATE TABLE IF NOT EXISTS `utilisateurs` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `login` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM AUTO_INCREMENT=4 DEFAULT CHARSET=latin1;

--
-- D chargement des donn es de la table `utilisateurs`
--

INSERT INTO `utilisateurs` (`id`, `login`, `password`) VALUES
(1, 'Moussa', '$2y$10$ycbeRtFDMQRVuH675v5ZIu1YxXS73rv1J/
fNTA9QsFiu7hWMe7kaS'),
(2, 'Ahmed', '$2y$10$9xi/xJwvt8ejTlMzECtmr.GAWBJy2A3I./
df16u2LGjKi0QZ5YeKm'),
(3, 'Amin', '$2y$10$02c0fzZ7LZGgMnBVtgLwR0cIz32/Ak3/CG.rFORqvd/
qln0JHcARy');
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```