

# Projet Intranet – Mission 7

## Objectif :

### Table des matières

1. Question 1 : Créez un formulaire html pour la création d'un utilisateur depuis l'espace d'administration.....	2
2. Ajoutez les groupes 'Administrateur', 'Ressources Humaines', 'Logistique' et 'Informatique' manuellement dans votre base de données.....	3
3. Ajoutez le code PHP pour récupérer les groupes dans la base de données et les afficher dans la liste du formulaire.....	4
4. Toujours en PHP, récupérez les valeurs du formulaire via la méthode POST pour créer l'utilisateur dans la base de données lorsque le formulaire est validé. ....	7

## Question 1 : Créez un formulaire html pour la création d'un utilisateur depuis l'espace d'administration

On a bien créé le code html pour permettre à l'administrateur de créer un utilisateur :

```
<!-- FORMULAIRE D'INSCRIPTION -->
<div class="cc">
  <div class="inscription-box">
    <h2>Inscription</h2>
    <form action="../php/inscriptionUtilisateur.php" method="POST">
      <div class="form-group">
        <label for="nom">Nom :</label>
        <input type="text" id="nom" name="nom" placeholder="Entrez votre nom" required>
      </div>

      <div class="form-group">
        <label for="mail">Mail :</label>
        <input type="email" id="mail" name="mail" placeholder="Entrez votre email" required>
      </div>

      <div class="form-group">
        <label for="username">Nom utilisateur :</label>
        <input type="text" id="username" name="username" placeholder="Entrez votre nom d'utilisateur" required>
      </div>

      <div class="form-group">
        <label for="password">Mot de passe :</label>
        <input type="password" id="password" name="password" placeholder="Entrez votre mot de passe" required>
      </div>

      <button type="submit" class="sauvegarde">
        
      </button>

      <p>Vous avez déjà un compte ? <a href="connexion.html" class="login-link">Connectez-vous</a></p>
    </form>
  </div>
```

On a un peu changé après et cela donnerait plus ça :

```
<form action="" method="POST" >
  <div>
    <label for="nom">Nom :</label>
    <input type="text" id="nom" name="nom" required>
  </div>
  <div>
    <label for="prenom">Prénom :</label>
    <input type="text" id="prenom" name="prenom" required>
  </div>
  <div>
    <label for="login">Login :</label>
    <input type="text" id="login" name="login" required>
  </div>
  <div>
    <label for="password">Mot de passe :</label>
    <input type="password" id="password" name="password" required>
  </div>
  <div>
    <label for="listeGroupe">Groupe :</label>
    <select name="listeGroupe" id="listeGroupe">
      <option value="1">Groupe 1</option>
      <option value="2">Groupe 2</option>
    </select>
  </div>
  <div>
    <label for="photo">Photo de profil :</label>
    <input type="file" id="photo" name="photo">
  </div>
  <button type="submit">Créer l'utilisateur</button>
</form>
```

Le code HTML représente un formulaire de création d'utilisateur qui est encapsulé dans une balise `<main>`, ce qui indique qu'il constitue le contenu principal de la page. À l'intérieur de cette balise, un titre `<h1>` affiche le texte "Création Utilisateur", indiquant clairement à l'utilisateur l'objectif de la page.

Ensuite, un `<div>` avec la classe `admin-container` est utilisé pour contenir le formulaire. Ce `div` est divisé en deux parties, avec un bloc de gauche (`left-block`) qui contient tout le formulaire de création des informations utilisateur. Ce bloc commence par un sous-titre `<h2>` qui précise qu'il s'agit de la section pour la "Création des informations de l'utilisateur".

Le formulaire commence avec une balise `<form>` qui utilise la méthode POST pour envoyer les données à la même page (via l'attribut `action=""`). L'attribut `enctype="multipart/form-data"` est utilisé pour permettre l'upload de fichiers (comme une photo de profil).

Le formulaire inclut plusieurs champs de saisie, chacun étant encapsulé dans un `<div>` avec la classe `form-group`. Chaque champ est accompagné d'un label explicite. Le champ pour le "Nom" est un champ texte (`<input type="text">`), de même que le champ pour le "Prénom", le "Login", et le "Mot de passe", ce dernier étant de type mot de passe pour masquer l'entrée.

Enfin, le formulaire se termine par un bouton de soumission, `<button type="submit" class="sauvegarde">`, qui permet à l'utilisateur d'envoyer les données du formulaire. L'image à l'intérieur du bouton représente une icône de sauvegarde et est affichée dans le bouton (image pour indiquer où il faut cliquer pour valider).

Ainsi, le formulaire permet à l'utilisateur de renseigner des informations personnelles (nom, prénom, login et mot de passe), puis de soumettre le tout en cliquant sur le bouton de sauvegarde.

Et la vue sur la page inscriptionUtilisateur (vue que par l'utilisateur avec des restrictions vue que il y a des restrictions quand on n'est pas administrateur) :

Figure 1 : Vue page Inscription Utilisateur – Espace Administration

**Le css est amené à changer !**

AT Professionnel – Projet Intranet

## 2. Ajoutez les groupes 'Administrateur', 'Ressources Humaines', 'Logistique' et 'Informatique' manuellement dans votre base de données.

On a bien inséré nos utilisateurs en utilisation cette commande sql, cette commande SQL insère quatre groupes dans la table groupes, avec un code (codeGrp) et un libellé (libelle) :

```
INSERT INTO `groupes` (`codeGrp`, `libelle`) VALUES  
('ADM', 'Administrateur'),  
('INF', 'Informatique'),  
('LOG', 'Logistique'),  
('REH', 'Ressources Humaines');
```

Figure 2: Commanbde SQL pour insérer les groupes

Comme on peut voir ci-dessous les groupes ont bien été ajouté :

✓ Affichage des lignes 0 - 3 (total de 4, traitement en 0,0002 seconde(s).)

`SELECT * FROM `groupes``

☐ Profilage [ Éditer en ligne ] [ Éditer ] [ Expliquer SQL ] [ Créer le code source PHP ] [ Actualiser ]

☐ Tout afficher | Nombre de lignes : 25 ▼ Filtrer les lignes: Chercher dans cette

Options supplémentaires

				codeGrp	libelle
<input type="checkbox"/>	✎ Éditer	📄 Copier	🗑 Supprimer	ADM	Administrateur
<input type="checkbox"/>	✎ Éditer	📄 Copier	🗑 Supprimer	INF	Informatique
<input type="checkbox"/>	✎ Éditer	📄 Copier	🗑 Supprimer	LOG	Logistique
<input type="checkbox"/>	✎ Éditer	📄 Copier	🗑 Supprimer	REH	Ressources Humaines

Figure 3 : Groupes ajoutés dans la base de données

### 3. Ajoutez le code PHP pour récupérer les groupes dans la base de données et les afficher dans la liste du formulaire.

Cette fonctionnalité est **disponible sur notre page de création d'utilisateur côté administrateur**. On a bien ajouté les groupes dans la base de données :

```
<!-- Liste déroulante pour les groupes -->
<label for="listeGroupe">Choisissez un groupe :</label>
<select name="listeGroupe" id="listeGroupe">
  <option value="" disabled selected>Choisir un groupe</option> <!-- Option par défaut -->
  <?php
    // Récupérer les groupes depuis la base de données
    $stmt = $pdo->query("SELECT codeGrp, libelle FROM groupes");

    // Vérifier s'il y a des groupes dans la base de données
    if ($stmt->rowCount() > 0) {
      // Afficher les options pour chaque groupe
      while ($groupe = $stmt->fetch(PDO::FETCH_ASSOC)) {
        echo "<option value='" . htmlspecialchars($groupe['codeGrp']) . "'" .
              htmlspecialchars($groupe['libelle']) . "</option>";
      }
    } else {
      // Si aucun groupe n'est trouvé
      echo "<option disabled>Aucun groupe disponible</option>";
    }
  ?>
</select>
```

Figure 4 : Code pour récupérer les groupes

Le code commence par définir un **élément HTML <label>** pour une liste déroulante, intitulée **"Choisissez un groupe"**. Ce label est lié à l'élément **<select>** par l'attribut **for**, ce qui permet d'indiquer que la liste déroulante correspond à cette question spécifique. Ensuite, une balise **<select>** est utilisée pour **afficher la liste déroulante**. Cette balise possède un **name** et un **id** qui servent à **identifier et récupérer** la valeur sélectionnée lorsqu'un utilisateur **soumet le formulaire**.

À l'intérieur de la balise **<select>**, il y a un code PHP qui va **interagir** avec la **base de données**. La première partie de ce code utilise une requête **SQL SELECT** pour récupérer **les informations des groupes** depuis une **table nommée groupes**. Le résultat de cette requête est stocké dans une **variable \$stmt**, qui contient les **lignes des groupes récupérées**.

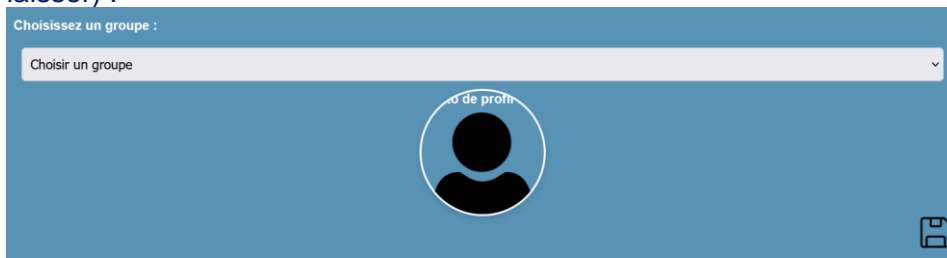
Ensuite, une vérification est effectuée pour savoir **s'il y a des groupes dans la base de données**. Si la requête renvoie des groupes (c'est-à-dire si le nombre de lignes retournées par **\$stmt->rowCount()** est **supérieur à 0**), une boucle **while** **parcourt chaque groupe récupéré** et **génère un élément <option>** pour chaque groupe. Les valeurs de chaque option sont dynamiquement générées à partir de la base de données, avec **value** représentant l'identifiant unique du groupe (**codeGrp**) et le texte affiché étant le nom du groupe (**libelle**). Cette génération permet à l'utilisateur de choisir parmi les différents groupes stockés dans la base de données.

AT Professionnel – Projet Intranet

Si **aucun groupe n'est trouvé** dans la base de données, un message "**Aucun groupe disponible**" est **affiché** dans la liste déroulante, et l'option est désactivée en utilisant l'**attribut disabled** pour empêcher la sélection.

**En résumé**, ce code génère une liste déroulante où l'utilisateur peut choisir un groupe parmi ceux récupérés dans la base de données, ou également avoir un message indiquant qu'aucun groupe n'est disponible.

Voici le résultat (la photo que profil est encore en cours de réflexion s'il est nécessaire de la laisser) :



On peut ensuite cliquer sur cette liste et choisir un groupe et choisir informatique par exemple :



Ou encore :

Choisissez un groupe :

Ressources Humaines

#### 4. Toujours en PHP, récupérez les valeurs du formulaire via la méthode POST pour créer l'utilisateur dans la base de données lorsque le formulaire est validé.

Le code PHP **récupère les données** soumises **via la méthode POST** dans le formulaire, en vérifiant que tous les champs nécessaires sont remplis. Il sécurise ensuite ces données, **notamment le mot de passe**, en utilisant `htmlspecialchars()` pour prévenir les attaques XSS et `password_hash()` pour **hacher le mot de passe**. Une requête SQL préparée est utilisée pour insérer les informations dans la base de données. Si **l'insertion est réussie**, un message de succès s'affiche ; sinon, un message d'erreur est montré. Si un champ est manquant, un message invite l'utilisateur à le remplir.

```
<!-- Ajout de l'utilisateur dans la bd -->
<?php
    if ($_SERVER["REQUEST_METHOD"] == "POST") {
        // Vérifier que tous les champs sont remplis
        if (!empty($_POST["nom"]) && !empty($_POST["prenom"]) && !empty($_POST["login"]) && !empty($_POST["password"])
            && !empty($_POST["listeGroupe"])) {

            // Récupérer les données du formulaire
            $nom = htmlspecialchars(string: $_POST["nom"]);
            $prenom = htmlspecialchars(string: $_POST["prenom"]);
            $login = htmlspecialchars(string: $_POST["login"]);
            $password = password_hash(password: $_POST["password"], algo: PASSWORD_DEFAULT); // Hachage du mot de passe
            $groupe = htmlspecialchars(string: $_POST["listeGroupe"]);

            // Requête d'insertion
            $sql = "INSERT INTO utilisateurs (nom, prenom, login, password, codeGrp) VALUES (:nom, :prenom, :login, :password, :codeGrp)";
            $stmt = $pdo->prepare(query: $sql);

            // Exécuter la requête
            if ($stmt->execute(params: [
                ':nom' => $nom,
                ':prenom' => $prenom,
                ':login' => $login,
                ':password' => $password,
                ':codeGrp' => $groupe
            ])) {
                echo "<p style='color: green;'>Utilisateur créé avec succès !</p>";
            } else {
                echo "<p style='color: red;'>Erreur lors de la création de l'utilisateur.</p>";
            }
        } else {
            echo "<p style='color: red;'>Veuillez remplir tous les champs.</p>";
        }
    }
?>
```

Test vérification :

AT Professionnel – Projet Intranet

Et quand on fait le test pour vérifier que le compte a bien été ajouté, si on clique sur le bouton sauvegarder en bas du formulaire :

**Création Utilisateur**

Création des informations de l'utilisateur

Nom :

Prénom :

Login :

Mot de passe :

Choisissez un groupe :

Informatique

L'utilisateur a bien été ajouté et le mot de passe est haché :

✓ Affichage des lignes 0 - 8 (total de 9, traitement en 0,0003 seconde(s).)

`SELECT * FROM `utilisateurs``

☐ Profilage [ [Éditer en ligne](#) ] [ [Éditer](#) ] [ [Expliquer SQL](#) ] [ [Créer le code source PHP](#) ] [ [Actualiser](#) ]

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes:  | Trier par clé : Aucun(e)

Options supplémentaires

	login	password	codeGrp
<input type="checkbox"/> <a href="#">Éditer</a> <a href="#">Copier</a> <a href="#">Supprimer</a>	aportolleau	aportolleau	INF
<input type="checkbox"/> <a href="#">Éditer</a> <a href="#">Copier</a> <a href="#">Supprimer</a>	axelcadix	axelcadixdu44	INF
<input type="checkbox"/> <a href="#">Éditer</a> <a href="#">Copier</a> <a href="#">Supprimer</a>	benjamindg	benjamindg44	LOG
<input type="checkbox"/> <a href="#">Éditer</a> <a href="#">Copier</a> <a href="#">Supprimer</a>	bonjour	\$2y\$10\$5WIPcO81EJZTxyNPLbeutzUuu0LJOqg7HmToieWF...	INF
<input type="checkbox"/> <a href="#">Éditer</a> <a href="#">Copier</a> <a href="#">Supprimer</a>	bonjour44	\$2y\$10\$n2hHlrirv4HX7ygz/3aNUuVB/adWSFx2EowTJI74go6...	LOG
<input type="checkbox"/> <a href="#">Éditer</a> <a href="#">Copier</a> <a href="#">Supprimer</a>	mkate	kate2025	ADM
<input type="checkbox"/> <a href="#">Éditer</a> <a href="#">Copier</a> <a href="#">Supprimer</a>	testquestion5	\$2y\$10\$n1vUBdT44rv5JUEXsr1XbepwfQK.37ZAGCqnuPUK00t...	INF