

# 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. ....	6

## 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 :

```
<div class="admin-container">
  <div class="left-block">
    <h2>Créer un utilisateur</h2>
    <form action="creation_utilisateur_Admin.php" method="POST"> <!-- Corrigez ici action="creation.php" -->

      <label for="login">Login :</label>
      <input type="text" id="login" name="login" required>

      <label for="nom">Nom :</label>
      <input type="text" id="nom" name="nom" required>

      <label for="prenom">Prénom :</label>
      <input type="text" id="prenom" name="prenom" required>

      <label for="leposte">Poste :</label>
      <select name="leposte" id="leposte">
        <?php foreach ($postes as $poste): ?>
          <option value="<?= htmlspecialchars($poste['idposte']); ?>"><?= htmlspecialchars($poste['libelleposte']); ?></option>
        <?php endforeach; ?>
      </select>

      <label for="tel">Téléphone :</label>
      <input type="text" id="tel" name="tel" required>

      <label for="mail">Email :</label>
      <input type="email" id="mail" name="mail" required>

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

      <label for="codeGrp">Groupe :</label>
      <select name="codeGrp" id="codeGrp">
        <?php foreach ($groupes as $groupe): ?>
          <option value="<?= htmlspecialchars($groupe['codeGrp']); ?>"><?= htmlspecialchars($groupe['libelle']); ?></option>
        <?php endforeach; ?>
      </select>

      <button type="submit" class="sauvegarde">
        
      </button>
    </form>
  </div>
</div>
```

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

AT Professionnel – Projet Intranet

"Login", et le "Mot de passe", ce dernier étant de type mot de passe pour masquer l'entrée. Mais également le groupe et le poste pour les privilèges accordés.

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) :

### Vue de la page :

### Création d'un utilisateur

Créer un utilisateur

Login :

Nom :

Prénom :

Poste :

Directeur

Téléphone :

Email :

Mot de passe :

Groupe :

Administrateur

Retour

## 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 1: 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 2 : 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 3 : 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 de la liste déroulante :

Groupe :

Administrateur



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

Groupe :

Informatique



Ou encore :

Groupe :

Logistique



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.

```
// Récupérer les listes des postes, utilisateurs et groupes
$sql_postes = "SELECT idposte, libelleposte FROM postes";
$stmt_postes = $pdo->query($sql_postes);
$postes = $stmt_postes->fetchAll(PDO::FETCH_ASSOC);

$sql_utilisateurs = "SELECT login FROM utilisateurs";
$stmt_utilisateurs = $pdo->query($sql_utilisateurs);
$utilisateurs = $stmt_utilisateurs->fetchAll(PDO::FETCH_ASSOC);

$sql_groupes = "SELECT codeGrp, libelle FROM groupes";
$stmt_groupes = $pdo->query($sql_groupes);
$groupes = $stmt_groupes->fetchAll(PDO::FETCH_ASSOC);

// Processus de création d'utilisateur
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    try {
        $login = htmlspecialchars($_POST['login']);
        $nom = htmlspecialchars($_POST['nom']);
        $prenom = htmlspecialchars($_POST['prenom']);
        $tel = htmlspecialchars($_POST['tel']);
        $mail = filter_var($_POST['mail'], FILTER_SANITIZE_EMAIL);
        $leposte = !empty($_POST['leposte']) ? htmlspecialchars($_POST['leposte']) : null;
        $codeGrp = !empty($_POST['codeGrp']) ? htmlspecialchars($_POST['codeGrp']) : null;
        $password = !empty($_POST['password']) ? $_POST['password'] : null;

        // Définir l'image par défaut si elle n'est pas fournie
        $chemin_image = '/images/user.png';

        // Vérifier si le login existe déjà
        $sql_check_user = "SELECT COUNT(*) FROM utilisateurs WHERE login = :login";
        $stmt_check_user = $pdo->prepare($sql_check_user);
        $stmt_check_user->execute([':login' => $login]);
        $user_exists = $stmt_check_user->fetchColumn();

        if ($user_exists > 0) {
            header("Location: creation.php?message=error&error=Ce login existe déjà.");
            exit;
        }

        // Démarrer la transaction
        $pdo->beginTransaction();

        // ♥ Insertion des données dans la table utilisateurs
        $hashed_password = password_hash($password, PASSWORD_DEFAULT);
        $sql_utilisateur = "INSERT INTO utilisateurs (login, password, codeGrp) VALUES (:login, :password, :codeGrp)";
        $stmt_utilisateur = $pdo->prepare($sql_utilisateur);
        $stmt_utilisateur->execute([
            ':login' => $login,
            ':password' => $hashed_password,
            ':codeGrp' => $codeGrp
        ]);
        $pdo->commit();
    } catch (PDOException $e) {
        header("Location: creation.php?message=error&error=" . $e->getMessage());
        exit;
    }
}
```

```
54 // Démarrer la transaction
55 $pdo->beginTransaction();
56
57 // ✓ Insertion des données dans la table utilisateurs
58 $hashed_password = password_hash($password, PASSWORD_DEFAULT);
59 $sql_utilisateur = "INSERT INTO utilisateurs (login, password, codeGrp) VALUES (:login, :password, :codeGrp)";
60 $stmt_utilisateur = $pdo->prepare($sql_utilisateur);
61 $stmt_utilisateur->execute([
62     ':login' => $login,
63     ':password' => $hashed_password,
64     ':codeGrp' => $codeGrp
65 ]);
66
67 // ✓ Insertion des informations personnelles dans la table infos_employes, y compris chemin_image
68 $sql_employe = "INSERT INTO infos_employes (login, nom, prenom, leposte, tel, mail, chemin_image)
69     VALUES (:login, :nom, :prenom, :leposte, :tel, :mail, :chemin_image)";
70 $stmt_employe = $pdo->prepare($sql_employe);
71 $stmt_employe->execute([
72     ':login' => $login,
73     ':nom' => $nom,
74     ':prenom' => $prenom,
75     ':leposte' => $leposte,
76     ':tel' => $tel,
77     ':mail' => $mail,
78     ':chemin_image' => $chemin_image // Insérer l'image par défaut
79 ]);
80
81 // Commit de la transaction
82 $pdo->commit();
83
84 // Redirection vers la page de succès
85 header("Location: creation_utilisateur_Admin.php?message=success");
86 exit;
87
88 } catch (Exception $e) {
89     // Rollback en cas d'erreur
90     $pdo->rollBack();
91     die("Erreur : " . $e->getMessage());
92 }
93 }
94 ?>
```

### **Test vérification :**

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 :



Créer un utilisateur

Login :

test

Nom :

test

Prénom :

test

Poste :

Employé

Téléphone :

1345678902

Email :

test@gmail.com

Mot de passe :

••••







Groupe :

Logistique



Retour

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

<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	tdurand	1761f5a2ae8ea9f0b53abf89034072af38b2008f4439233a6e...	LOG
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	test	\$2y\$10\$OfMNcnrm3wzA/Db2PKssK.klrUYs8tw2R2f8l6hWo8z...	LOG