

Procédure de réalisation du Formulaire de connexion

Auteur : FOUREY Mathis

1. Introduction

Ce document décrit le processus de création d'un formulaire de création de compte interactif. Ce projet portait sur la création d'un formulaire de connexion en HTML, CSS, JavaScript et PHP. Le formulaire intègre une gestion avancée des erreurs, une vérification de la robustesse des mots de passe, et une récupération sécurisée des données utilisateur. Ce projet visait à développer des compétences en validation de formulaires, gestion des interactions côté client et traitement des données côté serveur.

Technologies utilisées :

- **HTML5** pour la structure sémantique.
- **CSS3** pour le design et les styles visuels.
- **JavaScript** pour les validations en temps réel et les interactions dynamiques.
- **PHP** pour le traitement des données côté serveur.

2. Étapes de la réalisation

2.1. Structure HTML

La structure HTML se concentre sur une organisation claire avec des balises sémantiques. Le formulaire contient :

- **Champ e-mail** avec validation.
- **Champs mot de passe et confirmation** Une barre d'indicateur de force pour le mot de passe.

Exemple de code HTML :

```

<div class="form-container">
  <h2>CRÉATION DE COMPTE</h2>
  <form id="signupForm" action="./create-account.php" method="POST">
    <div class="form-group">
      <label for="email">Mon adresse mail*</label>
      <input type="email" id="email" name="email" required>
      <p id="emailError" style="display: none; color: red;">Adresse e-mail invalide</p>
    </div>
    <div class="form-group">
      <label for="password">Mon mot de passe*</label>
      <input type="password" id="password" name="password" required>
      <div id="strengthIndicator" class="strength-bar"></div>
      <p id="passwordStrength">Force : Aucune</p>
    </div>
    <div class="form-group">
      <label for="confirmPassword">Confirmation du mot de passe*</label>
      <input type="password" id="confirmPassword" name="confirmPassword" required>
      <p id="confirmPasswordError" style="display: none; color: red;">Les deux mots de passe ne correspondent pas</p>
    </div>
    <button type="submit">Créer le compte</button>
  </form>
</div>

```

2.2. Ajout des styles CSS

Les styles ont été conçus pour un design épuré, centré et responsive.

- **Couleurs principales** : Bleu (#0046b7) pour les boutons et en-têtes, fond clair (#f9f9f9) pour une meilleure lisibilité.
- **Barre de force du mot de passe** : Couleurs dynamiques indiquant la robustesse du mot de passe (rouge pour faible, vert pour excellent).
- **Structure de la page** : Centrage du formulaire avec des bordures arrondies.

```

body {
  font-family: Arial, sans-serif;
  background-color: #f9f9f9;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  margin: 0;
}

.form-container {
  width: 400px;
  padding: 20px;
  background-color: #fff;
  border: 1px solid #ccc;
  border-radius: 10px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
}

```

2.3. Fonctionnalités JavaScript

La validation côté client est assurée via JavaScript pour offrir des retours immédiats aux utilisateurs:

1. **Validation de l'e-mail** : Vérifie si l'adresse e-mail est valide.
2. **Barre de force du mot de passe** : Évalue la robustesse selon des critères (longueur, majuscules, chiffres, caractères spéciaux).
3. **Vérification des mots de passe** : Confirme si les deux mots de passe correspondent.

Exemple de code JavaScript :

```
const passwordInput = document.getElementById('password');
const confirmPasswordInput = document.getElementById('confirmPassword');
const passwordStrength = document.getElementById('passwordStrength');
const strengthIndicator = document.getElementById('strengthIndicator');
const confirmPasswordError = document.getElementById('confirmPasswordError');
const emailInput = document.getElementById('email');
const emailError = document.getElementById('emailError');

const checkPasswordStrength = (password) => {
  const criteria = [
    password.length >= 8,
    /[A-Z]/.test(password),
    /\d/.test(password),
    /[!@#$%^&*]/.test(password),
  ];
  const strength = criteria.filter(Boolean).length;

  console.log("Niveau de force :", strength); // Ajoute ceci pour vérifier le niveau

  switch (strength) {
    case 0:
      strengthIndicator.className = 'strength-bar strength-none';
      return 'Aucune';
    case 1:
      strengthIndicator.className = 'strength-bar strength-weak';
      return 'Faible';
    case 2:
      strengthIndicator.className = 'strength-bar strength-medium';
      return 'Moyenne';
    case 3:
      strengthIndicator.className = 'strength-bar strength-strong';
      return 'Bonne';
    case 4:
      strengthIndicator.className = 'strength-bar strength-excellent';
      return 'Excellente';
  }
};

passwordInput.addEventListener('input', () => {
  const strength = checkPasswordStrength(passwordInput.value);
  passwordStrength.innerHTML = `Force : ${strength}`;
});
```

2.4. Traitement PHP

Un script PHP simple traite les données et affiche un message de confirmation. La sécurité des données est assurée par htmlspecialchars() pour éviter les attaques XSS.

Exemple de code PHP :

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $email = htmlspecialchars($_POST["email"]);
    $password = htmlspecialchars($_POST["password"]);

    echo "<h2>Compte créé avec succès</h2>";
    echo "<p><strong>Email :</strong> $email</p>";
    echo "<p><strong>Mot de passe :</strong> (sécurisé)</p>";
}
?>
```

3. Résultat Final

- **Avant validation** : Les champs e-mail et mot de passe affichent des messages d'erreur en cas de saisie incorrecte.
- **Validation dynamique** : La barre de force change en fonction du mot de passe saisi.
- **Confirmation visuelle** : Les erreurs disparaissent lorsque l'utilisateur corrige ses entrées.

CRÉATION DE COMPTE

Mon adresse mail*

Mon mot de passe*

Force : Excellente

Confirmation du mot de passe*

Créer le compte