

Documentation du Projet : Générateur de Mot de Passe

1. Introduction

Ce projet est une application web permettant de générer un mot de passe aléatoire en fonction de la taille définie par l'utilisateur. L'application comprend un formulaire HTML, un script PHP pour la génération du mot de passe et une page de vérification.

2. Structure du Projet

Le projet est composé des fichiers suivants :

- index.html : Interface utilisateur permettant de choisir la taille du mot de passe.
- generateur.php : Script backend générant le mot de passe.
- verifier.html : Interface utilisateur permettant de vérifier un mot de passe.
- verifier.php : Fichier permettant de vérifier la force du mot de passe.

3. Interface Utilisateur

1. Page d'accueil (index.html) : Formulaire de saisie de la taille du mot de passe et le bouton de génération qui me permet d'envoyer au controller « verifier.php » la valeur de \$_POST['password'].

```
<h1>Générateur de mdp</h1>
<form action="generateur.php" method="post">
  <label for="pass">Taille du mot de passe : </label>
  <input type="text" id="pass" name="password" />
  <div class="button">
    <button type="submit">Générer</button>
  </div>
</form>
<a href="verifier.html">Verifier</a>
```

4. Fonctionnalités du Générateur

Je récupère la valeur saisie par l'utilisateur dans la et la stock dans la variable `$taille` qui me permettra de récupérer la taille du mot de passe demandé par l'utilisateur, je passe ensuite sur mon fonction `generer($taille)`.

Déclaration des variables :

`$car`, `$maj`, `$chi` et `$spe` sont des listes qui inclus les caractères à vérifier (minuscule, majuscules, caractères spéciaux et chiffres.

`$nb_car`, `$nb_maj`, `$nb_chi`, et `$nb_spe` sont des variables qui me serviront a stocker l'information permettant de vérifier si ces types de caractères sont contenu.

`$mdp` contient le mot de passe à générer.

```
$car = ["a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y", "z"];
$maj = ["A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z"];
$chi = ["0", "1", "2", "3", "4", "5", "6", "7", "8", "9"];
$spe = ["!", "@", "#", "$", "%", "^", "&", "*", "(", ")", "=", "_", "-", "+", "[", "]", "{", "}", ";", ":", "'", "\"", "<", ">", "/", "?", "\\", "|", "~", ""];

$nb_car = $nb_maj = $nb_chi = $nb_spe = $rand = 0;
$mdp= "";
```

Dans un premier temps je vérifie si `$taille` est bien supérieur ou égal à 14 :

```
if ($taille >= 14){
    echo "la taille du mot de passe est " . $taille;
```

Sinon, j'en avertis l'utilisateur :

```
else {
    return "Le mot de passe doit faire au moins 14 caractères";
}
```

En utilisant une boucle `while`, je parcours chaque élément de la liste. Grâce à la fonction `rand()` que j'effectue une première fois, je choisis aléatoirement un type de caractère, s'il est présent et que aucun autre ne l'est j'arrête la condition et continue la boucle. Sinon, j'ajoute un caractère aléatoire de la liste sélectionné grâce au précédent `rand()` contenue dans `$rand()` grâce à une nouvelle utilisation de `rand()`.

```
if ($taille >= 14){
    echo "la taille du mot de passe est " . $taille;
while(strlen($mdp) < $taille){
    $rand = rand(1, 4);
    switch ($rand){
        case 1 :
            if ($nb_car < 1 || ($nb_maj > 0 && $nb_chi > 0 && $nb_spe > 0)){
                $rand = rand(0,25);
                $mdp = $mdp.$car[$rand];
                $nb_car = $nb_car + 1;
            }
            break;

        case 2 :
            if ($nb_maj < 1 || ($nb_car > 0 && $nb_chi > 0 && $nb_spe > 0)){
                $rand = rand(0,25);
                $mdp = $mdp.$maj[$rand];
                $nb_maj = $nb_maj + 1;
            }
            break;

        case 3 :
            if ($nb_chi < 1 || ($nb_car > 0 && $nb_maj > 0 && $nb_spe > 0)){
                $rand = rand(0,9);
                $mdp = $mdp.$chi[$rand];
                $nb_chi = $nb_chi + 1;
            }
            break;

        case 4 :
            if ($nb_spe < 1 || ($nb_car > 0 && $nb_maj > 0 && $nb_chi > 0)){
                $rand = rand(0,31);
                $mdp = $mdp.$spe[$rand];
                $nb_spe = $nb_spe + 1;
            }
            break;
    }
}
}
```

5. Vérification du Mot de Passe

3. Page de vérification (verifier.php) : Je récupère la valeur saisie par l'utilisateur dans \$_POST['password'] et la stock dans la variable \$mdp qui me permettra de récupérer la taille du mot de passe demandé par l'utilisateur, je passe ensuite sur mon fonction generer(\$taille).

```
<h1>Verificateur de mdp</h1>
<form action="verifier.php" method="post">
  <label for="pass">Mot de passe : </label>
  <input type="password" id="pass" name="password" />
  <div class="button">
    <button type="submit">Verifier</button>
  </div>
</form>
```

Je déclare ensuite les variables qui me serviront à vérifier si le mot de passe est valide mais aussi à écrire un message personnalisé selon les erreurs détecté :

```
$erreur = 0;
$message = "";
$taille = "La longueur du mot de passe est inférieure à 14 caractères.";
$ecar = "\nIl n'y a pas de minuscule.";
$emaj = "\nIl n'y a pas de majuscule.";
$echi = "\nIl n'y a pas de chiffre.";
$espe = "\nIl n'y a pas de caractère spéciaux.";
$car = ["a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y", "z"];
$maj = ["A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z"];
$chi = ["0", "1", "2", "3", "4", "5", "6", "7", "8", "9"];
$spe = ["!", "@", "#", "$", "%", "&", "*", "(", ")", "-", "_", "=", "+", "[", "]", "{", "}", ":", ";", ",", "<", ">", "/", "?", "\\", "|", "~", "`"];
```

Je vérifie chaque critère en vérifiant chaque caractère grâce à une boucle for, cela me permet d'incrémenter le nombre total de chaque type de caractère :

```
if ($mdp){
    $ncar = $nmaj = $nchi = $nspe = 0;
    if (strlen($mdp) < 14){
        $erreur ++;
        $msge .= $etaille;
    }
    for ($i = 0; $i < strlen($mdp); $i++){
        for ($j = 0; $j < count($car); $j++){
            if ($mdp[$i] == $car[$j]){
                $ncar ++;
                break;
            }
        }
        for ($j = 0; $j < count($maj); $j++){
            if ($mdp[$i] == $maj[$j]){
                $nmaj ++;
                break;
            }
        }
        for ($j = 0; $j < count($chi); $j++){
            if ($mdp[$i] == $chi[$j]){
                $nchi ++;
                break;
            }
        }
        for ($j = 0; $j < count($maj); $j++){
            if ($mdp[$i] == $spe[$j]){
                $nspe ++;
                break;
            }
        }
    }
}
```

```
if ($ncar == 0){
    $erreur ++;
    $msge .= $ecar;
}
if ($nmaj == 0){
    $erreur ++;
    $msge .= $emaj;
}
if ($nchi == 0){
    $erreur ++;
    $msge .= $echi;
}
if ($nspe == 0){
    $erreur ++;
    $msge .= $espe;
}
}
```

Enfin, je transmets le message personnalisé et permet une redirection :

```
    if ($erreur > 0) {
        $msge = "Il y a {$erreur} erreur(s) dans votre mot de passe. " . $msge;
    } else {
        $msge = "Le mot de passe est valide.";
    }
    return $msge;
}
echo "resultat : " . verifier($mdp);
?><html>
<a href="index.html">Generer</a>
<a href="verifier.html">Verifier</a>
</html><?php
?>
```

6. Utilisation

1. Ouvrir index.html dans un navigateur.
2. Entrer la taille du mot de passe souhaitée.
3. Cliquer sur 'Générer'.
4. Copier le mot de passe généré affiché par generateur.php.
5. Se rendre sur verifier.php pour tester la force du mot de passe.

7. Conclusion

Ce projet offre une solution simple et efficace pour générer des mots de passe sécurisés et tester leur robustesse. Des améliorations possibles incluent l'ajout de critères de sécurité plus avancés et une interface utilisateur plus ergonomique.