

Virginie Exotique

Virginie Exotique

Introduction :

J'ai effectué un stage de 6 semaines dans un magasin d'alimentation générale appeler Virginie Exotique, qui se situe au 13 AVENUE ALBERT THOMAS, dans le département des PAVILLON SOUS BOIS.

Au cours de mon stage, j'ai eu l'opportunité de contribuer au développement d'un page de connexion et d'inscription pour sécuriser le mot de passe. Cette page vise à permettre aux utilisateurs de se connecter. Mon rôle principal a été de concevoir et d'implémenter cette page et de sécuriser le mot de passe du client lors de l'inscription, connexion en utilisant les technologies web telles que HTML, CSS, JavaScript, PHP, MySQL.

Présentation du candidat :

Je m'appelle Thiviyan, j'ai 19 ans et je suis étudiant à l'école IRIS à Paris en classe de 2^{ème} année de BTS SIO (Brevet de Technicien Supérieur Services Informatiques aux Organisations) Option SLAM (Solutions Logicielles et Applications Métier).

La forme juridique de l'entreprise est Société à responsabilité limitée. Son domaine d'activité est : commerce d'alimentation générale.

Activité effectuer

J'ai effectué un page de connexion et inscription.

La Page d'Inscription

1. Structure de la Page

DOCTYPE et Balise HTML

- Le document commence par la déclaration `<!DOCTYPE html>`, spécifiant qu'il s'agit d'un document HTML5.

Head Section

- Le document commence par la déclaration `<!DOCTYPE html>`, spécifiant qu'il s'agit d'un document HTML5.

Body Section

Logo

- Un logo est affiché en haut de la page avec un lien vers la page d'accueil (`index.html`).

Formulaire d'Inscription

- Le formulaire est encapsulé dans un div avec la classe `insc-form`.
- Affichage des messages d'erreur et de succès basés sur les paramètres GET (`reg_err`).

2. Affichage des Messages d'Erreur et de Succès

Gestion des Erreurs

- Utilisation de `isset($_GET['reg_err'])` pour vérifier la présence d'erreurs.
- Différents types d'erreurs sont gérés (`success`, `password`, `weak_password`, `email_length`, `pseudo_length`, `already`, `sx_miss`, `surname_miss`, `name_miss`, `dob_miss`, `ntel_miss`).
- Affichage des messages d'erreur correspondants dans des div avec des classes `success-stl` et `error-stl`.

Head Section

- Le document commence par la déclaration `<!DOCTYPE html>`, spécifiant qu'il s'agit d'un document HTML5.

3. Formulaire d'Inscription

Informations Personnelles

- Champs pour le nom (`nom\_user`), prénom (`prenom\_user`), date de naissance (`datenaissance\_user`), sexe (`sexe\_user`), et numéro de téléphone (`numtel\_user`).

Informations d'Authentification

- Champs pour le pseudo (`pseudo\_user`), email (`email\_user`), mot de passe (`password\_user`), et confirmation du mot de passe (`password\_retype`).

Coche

- Options pour s'abonner à la newsletter et accepter les conditions d'utilisation et la politique de confidentialité.

4. Footer

Menu

- Liens vers différentes sections du site : Rayon, Catalogue, Mes produits, Panier.

Nos Services

- Liens vers les services offerts : Drive, Livraison à Domicile, Livraison Express.

Article

- Liens vers la FAQ, Contacter-Nous, Magasin.

Copyright

- Mention des droits d'auteur et du designer du site.

Script PHP d'Inscription

Le script PHP présenté permet de gérer l'inscription d'un utilisateur dans une base de données MySQL. Voici une analyse détaillée de son fonctionnement :

1. Connexion à la Base de Données

```
<?php
require("contrôleur/config_bdd.php");

// $bdd = mysqli_connect('localhost',

if (!$bdd) {
    echo "Connection failed!";
}
```

Le script commence par inclure le fichier de configuration de la base de données et vérifie la connexion.

2. Vérification des Champs du Formulaire

```
10     if (
11         !empty($_POST['nom_user']) &&
12         !empty($_POST['prenom_user']) &&
13         !empty($_POST['datenaissance_user']) &&
14         !empty($_POST['sexe_user']) &&
15         !empty($_POST['numtel_user']) &&
16         !empty($_POST['pseudo_user']) &&
17         !empty($_POST['email_user']) &&
18         !empty($_POST['password_user']) &&
19         !empty($_POST['password_retype'])
20     ) {
```

Le script vérifie que tous les champs nécessaires du formulaire sont remplis.

3. Protection Contre les Attaques XSS

```
32     $nom_user = htmlspecialchars($_POST['nom_user']);
33     $prenom_user = htmlspecialchars($_POST['prenom_user']);
34     $datenaissance_user = date('Y-m-d', strtotime($_POST['datenaissance_user']));
35     $sexe_user = htmlspecialchars($_POST['sexe_user']);
36     $numtel_user = htmlspecialchars($_POST['numtel_user']);
37     $pseudo_user = htmlspecialchars($_POST['pseudo_user']);
38     $email_user = htmlspecialchars($_POST['email_user']);
39     $password_user = htmlspecialchars($_POST['password_user']);
40     $password_retype = htmlspecialchars($_POST['password_retype']);
41     $statut_user = htmlspecialchars('client');
```

Les données du formulaire sont nettoyées pour prévenir les attaques XSS.

4. Vérification de l'Existence de l'Utilisateur

```
43     // On vérifie si l'utilisateur existe
44     $check = $bdd->prepare('SELECT nom_user,prenom_user,datenaissance_user,pseudo_user, email_user,
45     $check->execute(array($email_user));
46     $data = $check->fetch();
47     $row = $check->rowCount();
```

Le script vérifie si un utilisateur avec le même email existe déjà dans la base de données.

5. Formatage des Données

```
49     $nom_user = strtoupper($nom_user); // on transforme
50     $prenom_user = ucfirst(strtolower($prenom_user));
51     $email_user = strtolower($email_user); // on transforme
```

Les noms, prénoms et emails sont formatés pour assurer une uniformité dans la base de données.

6. Vérifications Complémentaires et Insertion dans la Base de Données

```
63 if($row == 0){
64     if($nom_user){
65         if(strlen($prenom_user)){
66             if(strlen($datenaissance_user)){
67                 if($sexe_user){
68                     if($numtel_user){
69                         if(strlen($pseudo_user) <= 100){ // On verifie que la longueur du pseudo <=
70                             if(strlen($email_user) <= 100){ // On verifie que la longueur du mail <
71                                 if(filter_var($email_user, FILTER_VALIDATE_EMAIL)){ // Si l'email e
72                                     if (getStrength($password_user) > 1) { // Vérifiez la force du
73                                         if($password_user === $password_retype){ // si les deux mdp
74                                             // On hash le mot de passe avec Bcrypt, via un coût de
75                                             $cost = ['cost' => 12];
76                                             $password_user = password_hash($password_user, PASSWORD
77
78                                             // On stock l'adresse IP
79                                             $ip_user = $_SERVER['REMOTE_ADDR'];
80                                             /*
81                                             ATTENTION
82                                             Vérifiez bien que le champs token est présent dans votre
83                                             le .sql est dispo pensez à l'importer !
84                                             ATTENTION
85                                             */
86                                             // On insère dans la base de données
87                                             echo 'vu';
88                                             $insert = $bdd->prepare('INSERT INTO client_ve(nom_user,
89                                             $insert->execute(array(
90                                                 'nom_user' => $nom_user,
91                                                 'prenom_user' => $prenom_user,
92                                                 'datenaissance_user' => $datenaissance_user,
93                                                 'sexe_user' => $sexe_user,
94                                                 'numtel_user' => $numtel_user,
95                                                 'pseudo_user' => $pseudo_user,
96                                                 'email_user' => $email_user,
97                                                 'password user' => $password user,
```

Si l'utilisateur n'existe pas déjà, le script vérifie chaque champ et leur format, puis insère les données dans la base de données. Si tout est correct, l'utilisateur est redirigé vers une page de succès.

7. Fonction de Vérification de la Force du Mot de Passe

```
118 function getStrength($password) {
119     $hasSpecialChar = preg_match('/^[^A-Za-z0-9]/', $password);
120     if (!$hasSpecialChar) {
121         return 1; // Faible
122     } else if ($hasSpecialChar && strlen($password) < 8) {
123         return 2; // Moyen
124     } else if ($hasSpecialChar && strlen($password) >= 8) {
125         return 3; // Fort
126     }
127     return 0;
128 }
```

Cette fonction évalue la force du mot de passe basé sur la présence de caractères spéciaux et la longueur du mot de passe.

Script JavaScript de Vérification de la Force du Mot de Passe

Le script JavaScript présenté permet d'évaluer la force d'un mot de passe saisi par l'utilisateur en temps réel. Voici une analyse détaillée de son fonctionnement :

1. Écouteur d'Événements pour les Entrées Clavier

```
1 document.addEventListener("keyup", function (e) {
2     let password = document.querySelector('.p-input-container input').value;
3     let strengthValue = getStrength(password);
4     let container = document.querySelector('.strength-percent');
5
6     if (password.length === 0) {
7         container.classList.remove('weak', 'average', 'strong');
8     } else if (strengthValue === 1) {
9         container.classList.add('weak');
10        container.classList.remove('average', 'strong');
11    } else if (strengthValue === 2) {
12        container.classList.remove('weak');
13        container.classList.add('average');
14        container.classList.remove('strong');
15    } else if (strengthValue === 3) {
16        container.classList.remove('weak', 'average');
17        container.classList.add('strong');
18    }
19 });
```

Le script utilise un écouteur d'événements keyup pour détecter les entrées clavier dans un champ de saisie de mot de passe. À chaque frappe, il :

1. Récupère la valeur du mot de passe.
2. Évalue sa force via la fonction getStrength.
3. Met à jour la classe CSS d'un conteneur visuel pour indiquer la force du mot de passe (weak, average, strong).

2. Fonction de Vérification de la Force du Mot de Passe

```
21 function getStrength(password) {
22     const hasSpecialChar = /^[^A-Za-z0-9]/.test(password);
23     if (!hasSpecialChar) {
24         return 1; // Faible
25     } else if (hasSpecialChar && password.length < 8) {
26         return 2; // Moyen
27     } else if (hasSpecialChar && password.length >= 8) {
28         return 3; // Fort
29     }
30     return 0;
31 }
```

La fonction « getStrength » évalue la force du mot de passe basé sur deux critères :

1. Présence de caractères spéciaux.
2. Longueur du mot de passe.

Les niveaux de force sont définis comme suit :

- 1 (Faible) : Le mot de passe ne contient pas de caractères spéciaux.
- 2 (Moyen) : Le mot de passe contient des caractères spéciaux mais est inférieur à 8 caractères.
- 3 (Fort) : Le mot de passe contient des caractères spéciaux et a une longueur d'au moins 8 caractères.

La page de connexion du site web "Virginie Exotique" est conçue pour permettre aux utilisateurs de se connecter à leur compte. Cette page présente un formulaire de connexion où les utilisateurs peuvent saisir leur nom d'utilisateur et leur mot de passe.

1. Fonctionnalités Principales

1/Formulaire de Connexion: Le formulaire de connexion comprend des champs pour le nom d'utilisateur et le mot de passe.

2/Gestion des Erreurs: En cas d'erreur lors de la saisie des informations de connexion, des messages d'erreur appropriés sont affichés pour informer l'utilisateur de la nature de l'erreur (mot de passe incorrect, e-mail incorrect, compte inexistant, etc.).

3/Options de Navigation: Les utilisateurs ont la possibilité de réinitialiser leur mot de passe s'ils l'ont oublié, ainsi que de créer un nouveau compte s'ils n'en ont pas déjà un.

1. Fonctionnalités Principales

Technologies Utilisées :

HTML et CSS: Utilisés pour la structure et le style de la page web.

PHP: Utilisé pour la gestion des sessions et des requêtes vers la base de données.

JavaScript: Utilisé pour les interactions dynamiques sur la page, telles que la révélation du mot de passe et la validation des champs.

2. Fonction de Vérification de la Force du Mot de Passe

```

21  function getStrength(password) {
22      const hasSpecialChar = /^[^A-Za-z0-9]/.test(password);
23      if (!hasSpecialChar) {
24          return 1; // Faible
25      } else if (hasSpecialChar && password.length < 8) {
26          return 2; // Moyen
27      } else if (hasSpecialChar && password.length >= 8) {
28          return 3; // Fort
29      }
30      return 0;
31  }

```

La fonction « getStrength » évalue la force du mot de passe basé sur deux critères :

1.Présence de caractères spéciaux.

2.Longueur du mot de passe.

Les niveaux de force sont définis comme suit :

- 1 (Faible) : Le mot de passe ne contient pas de caractères spéciaux.
- 2 (Moyen) : Le mot de passe contient des caractères spéciaux mais est inférieur à 8 caractères.
- 3 (Fort) : Le mot de passe contient des caractères spéciaux et a une longueur d'au moins 8 caractères.

La Page de connexion

1. Structure de la Page

DOCTYPE et Balise HTML

- Le document commence par la déclaration `<!DOCTYPE html>`, spécifiant qu'il s'agit d'un document HTML5.

Head Section

- Le document commence par la déclaration `<!DOCTYPE html>`, spécifiant qu'il s'agit d'un document HTML5.

Body Section

Logo

- Un logo est affiché en haut de la page avec un lien vers la page d'accueil (`index.html`).

Formulaire de connexion

- Le formulaire est encapsulé dans un div avec la classe `login-form`.
- Affichage des messages d'erreur et de succès basés sur les paramètres GET (`login_err`).

2. Affichage des Messages d'Erreur et de Succès

Gestion des Erreurs

- Utilisation de `isset($_GET['login_err'])` pour vérifier la présence d'erreurs.
- Différents types d'erreurs sont gérés (`password`, `email`, `already`).
- Affichage des messages d'erreur correspondants dans des div avec des classes `error-stl`.

Head Section

- Le document commence par la déclaration `<!DOCTYPE html>`, spécifiant qu'il s'agit d'un document HTML5.

3. Formulaire de connexion

Informations

- Champs pour le username (`identifier`), mot de passe (`password`).

4. Footer

Menu

- Liens vers différentes sections du site : Rayon, Catalogue, Mes produits, Panier.

Nos Services

- Liens vers les services offerts : Drive, Livraison à Domicile, Livraison Express.

Article

- Liens vers la FAQ, Contacter-Nous, Magasin.

Copyright

- Mention des droits d'auteur et du designer du site.

Script PHP de connexion

Le script PHP présenté permet de gérer l'inscription d'un utilisateur dans une base de données MySQL. Voici une analyse détaillée de son fonctionnement :

1. Connexion à la Base de Données

```
<?php
require("contrôleur/config_bdd.php");

// $bdd = mysqli_connect('localhost',

if (!$bdd) {
    echo "Connection failed!";
}
```

Le script commence par inclure le fichier de configuration de la base de données et vérifie la connexion.

2. Vérification des Champs du Formulaire

```
5  if (!empty($_POST['identifier']) && !empty($_POST['password_user'])) {
```

Le script vérifie que tous les champs nécessaires du formulaire sont remplis.

3. Protection Contre les Attaques XSS

```
7      $identifier = htmlspecialchars($_POST['identifier']);
8      $password_user = htmlspecialchars($_POST['password_user']);
```

Les données du formulaire sont nettoyées pour prévenir les attaques XSS.

4. Vérification de l'Existence de l'Utilisateur

```
13 $check = $bdd->prepare('SELECT nom_user, prenom_user, datenaissance_user, pseudo_user, email_user, password_user, token_user FROM user_
14 $check->execute(array($identifiant, $identifiant));
15 $data = $check->fetch();
16 $row = $check->rowCount();
```

Le script vérifie si un utilisateur avec le même email existe déjà dans la base de données.

5. Formatage des Données

```
10 $identifiant = strtolower($identifiant);
```

Les pseudo ou emails sont formatés pour assurer une uniformité dans la base de données.

6. Vérifications Complémentaires et Connexion du client enregistré dans la Base de Données

```
19 if ($row > 0) {
20     if (filter_var($identifiant, FILTER_VALIDATE_EMAIL) || $identifiant == $data['pseudo_user']) {
21         if (password_verify($password_user, $data['password_user'])) { // Si le mot de passe est
22             $_SESSION['user_ve'] = $data['token_user'];
23             header('Location: pageuser.php'); // Redirection vers la page utilisateur
24             die();
25         } else {
26             header('Location: Connector.php?login_err=password');
27             die();
28         }
29     } else {
30         header('Location: Connector.php?login_err=identifiant');
31         die();
32     }
33 } else {
34     header('Location: Connector.php?login_err=already');
35     die();
36 }
```

Conclusion

En conclusion, mon stage m'a permis de contribuer au développement d'une page de requête d'avis, améliorant l'expérience utilisateur avec des technologies modernes comme JavaScript. J'ai renforcé mes compétences en développement web. Cette expérience me prépare bien pour mes futurs projets.

Remerciement

Je remercie l'entreprise, de m'avoir accepté au

sein de l'entreprise. Et pour la confiance dont ils ont fait preuve à mon égard.