

DOCUMENTATION CODE

FIDELILIUM



FIDELILIUM

Libérer & sécuriser la vie numérique

Formulaire de contact

1. Vérification du formulaire et de la case à cocher

```
// Vérifie si le formulaire a été soumis
if (isset($_POST['ok'])) {
    // Vérifie si la case à cocher "ma_checkbox" est cochée
    if (isset($_POST["ma_checkbox"])) {
```

Cette section vérifie si le formulaire a été soumis et si la case à cocher "ma_checkbox" est cochée.

3. Validation des Champs du Formulaire

```
if (!empty($_POST['nom']) && !empty($_POST['prénom'])
    && !empty($_POST['email'])
    && !empty($_POST['société']) && !empty($_POST['objet'])
    && !empty($_POST['message'])) {
```

Vérifie que tous les champs obligatoires du formulaire sont remplis.

4. Validation de l'Adresse E-mail

```
$email = filter_var($_POST['email'], FILTER_SANITIZE_EMAIL);
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
    $msg = "Adresse e-mail non valide";
    $statut = "error";
} else {
```

Nettoie et valide l'adresse e-mail. En cas d'erreur, défini un message d'erreur.

5. Récupération des Données du Formulaire

```
$nom = $_POST["nom"];  
$prenom = $_POST["prénom"];  
$telephone = $_POST["téléphone"];  
$societe = $_POST["société"];  
$fonction = $_POST["fonction"];  
$objet = $_POST["objet"];  
$message = htmlspecialchars($_POST["message"]);
```

Récupère les données du formulaire dans des variables.

6. Requête Préparée pour la Base de Données

```
$stmt = $db->prepare("INSERT INTO contact (nom, prenom, mail, tel, soci  
$stmt->bind_param("ssssssss", $nom, $prenom, $email, $telephone, $soci  
$stmt->execute();  
$stmt->close();
```

Utilise une requête préparée pour insérer les données dans une base de données, évitant ainsi les injections SQL.

7. Envoi d'un E-mail

```
$to = 'fidelilium@gmail.com';  
$subject = 'Formulaire de contact';  
mail($to, $subject, print_r($_POST, true), "From: $email");
```

Envoie un e-mail avec les données du formulaire au destinataire.
L'utilisation d'une bibliothèque de messagerie sécurisée est recommandée.

8. Gestion des Messages de Confirmation et d'Erreur

```
$msg = "Votre message a bien été envoyé, Vous allez être redirigé !";  
$statut = "success";
```

Gère les messages de confirmation et d'erreur en fonction du succès ou de l'échec du traitement du formulaire.

Page des actualités et produits et services

Les pages “actualités” et “produits et services” utilisent le même code, il suffit juste de modifier l’affichage en fonction de l’HTML souhaité et de changer la table utilisée dans le SELECT

1. Sélection des Actualités depuis la Base de Données

```
$query = "SELECT * FROM actualite";  
$result = $db->query($query);
```

Cette partie du code utilise une requête SQL SELECT pour récupérer toutes les lignes de la table 'actualite' dans la base de données.

2. Affichage des Actualités dans une Boucle While

```
while ($row = $result->fetch_assoc()) {  
    // Affiche une carte d'actualité avec les détails de chaque ligne  
    echo '<div class="card">';  
    echo '<div class="groupe gauche" data-aos="fade-right">';  
    echo '';  
    echo '<div class="groupe droite" data-aos="fade-left">';  
    echo '<div class="paragraphe">';  
    echo '<h3>' . $row['titre'] . '</h3>';  
    echo '<p class="p-block-3">' . $row['texte'] . '</p>';  
    echo '</div>';  
    echo '</div>';  
    echo '</div>';  
}
```

Cette section utilise une boucle `while` pour parcourir chaque ligne résultante de la requête SELECT. Elle affiche ensuite les détails de chaque actualité, tels que l'image, le titre et le texte, dans une structure HTML.

3. Affichage des Liens d'Administration pour les Administrateurs

```
if (isset($_SESSION['admin_logged_in']) && $_SESSION['admin_logged_in'])  
    echo "<div class='admin'>";  
    echo "<a href='modifier.php?id=" . $row['id'] . "' ><svg xmlns='ht";  
    echo "<a href='supprimer.php?id=" . $row['id'] . "' ><svg xmlns='ht";  
    echo "<a href='/admin/logout.php'><p>Encore connecté se déconnecter";  
    echo "</div>";  
    echo "<br>";  
}
```

Cette partie affiche des liens d'administration pour chaque actualité si l'utilisateur est connecté en tant qu'administrateur. Ces liens permettent de modifier ou de supprimer une actualité.

4. Fermeture de la Connexion à la Base de Données

```
$db->close();  
?>
```

Connexion Administrateur

1. Démarrage de la Session

```
session_start();
```

Cette ligne démarre ou restaure une session PHP. C'est nécessaire pour pouvoir utiliser les variables de session, telles que `\$_SESSION['admin_logged_in']`.

2. Vérification si l'Admin est déjà Connecté

```
if (isset($_SESSION['admin_logged_in']) && $_SESSION['admin_logged_in'])  
    header("Location: admin_page.php");  
    exit();  
}
```

Cette partie vérifie si l'administrateur est déjà connecté. Si c'est le cas, le script redirige automatiquement vers la page d'administration (`admin_page.php`). La fonction `exit()` est utilisée pour arrêter l'exécution du script après la redirection.

3. Vérification si le Formulaire de Connexion a été Soumis

```
if ($_SERVER['REQUEST_METHOD'] === 'POST') {  
    // ... (voir la suite ci-dessous)  
}
```

Cette condition vérifie si le formulaire de connexion a été soumis en vérifiant si la méthode de la requête est POST.

4. Informations d'Identification de l'Administrateur

```
$username = 'admin';  
$password = 'admin123';
```

Définition des informations d'identification de l'administrateur (nom d'utilisateur et mot de passe). Dans cet exemple, le nom d'utilisateur est 'admin' et le mot de passe est 'admin123'. Dans un environnement de production, ces informations devraient être stockées de manière plus sécurisée, par exemple dans une base de données.

5. Vérification des Informations de Connexion

```
if ($_POST['username'] == $username && $_POST['password'] == $password  
    $_SESSION['admin_logged_in'] = true;  
    header("Location: admin_page.php");  
    exit();  
} else {  
    echo "Identifiants incorrects. Veuillez réessayer.";  
}
```

Cette partie vérifie si les informations de connexion soumises via le formulaire (\$_POST['username'] et \$_POST['password']) correspondent aux informations d'identification de l'administrateur. Si la correspondance est réussie, la variable de session \$_SESSION['admin_logged_in'] est définie à true, et l'utilisateur est redirigé vers la page d'administration. Sinon, un message d'erreur est affiché.

CRUD

Création

1. Vérification si le Formulaire a été Soumis

```
if(isset($_POST['ok'])) {  
    // ...  
}
```

Cette condition vérifie si le formulaire a été soumis en vérifiant si la variable `\$_POST['ok']` est définie. Cela peut être le nom du bouton de soumission dans le formulaire.

2. Récupération des Données du Formulaire

```
$titre = $_POST["titre"];  
$texte = $_POST["texte"];  
$alt = $_POST["alt_text"];
```

Les données du formulaire (titre, texte, texte alternatif) sont récupérées à partir de la superglobale `\$_POST`.

3. Gestion de l'Image Téléchargée

```
$img_name = $_FILES['image']['name'];  
$img_size = $_FILES['image']['size'];  
$tmp_name = $_FILES['image']['tmp_name'];  
$error = $_FILES['image']['error'];  
  
$img_ex = pathinfo($img_name, PATHINFO_EXTENSION);  
$img_ex_lc = strtolower($img_ex);  
$allowed_exs = array("jpg", "jpeg", "png");
```

Les informations sur l'image téléchargée sont extraites de la superglobale `\$_FILES`. Ces informations incluent le nom de l'image, sa taille, le nom temporaire, et tout éventuel message d'erreur.

4. Vérification du Format de l'Image

```
if (in_array($img_ex_lc, $allowed_exs)) {  
    // ...  
} else {  
    echo "Erreur";  
}
```

Il vérifie si l'extension de l'image est parmi les extensions autorisées (jpg, jpeg, png). Si c'est le cas, le script continue, sinon, il affiche un message d'erreur.

5. Génération d'un Nouveau Nom d'Image Unique

```
$new_img_name = uniqid("IMG-", true).'.'.$img_ex_lc;  
$img_upload_path = __DIR__ . '/../../images/servicesetproduits/' . $new_
```

Un nouveau nom unique est généré pour l'image en utilisant la fonction `uniqid`. Le chemin de destination pour l'enregistrement de l'image est également défini.

6. Déplacement de l'Image Téléchargée vers le Répertoire de Destination

```
move_uploaded_file($tmp_name, $img_upload_path);
```

L'image téléchargée est déplacée depuis son emplacement temporaire vers le répertoire de destination avec le nouveau nom généré.

7. Insertion des Données dans la Base de Données

```
mysqli_query($db, "INSERT INTO produits VALUES ('','$titre','$texte','$
```

Les données, y compris le chemin de l'image, sont insérées dans une base de données MySQL. Le champ d'ID semble être un champ auto-incrémenté.

8. Redirection vers une Autre Page après l'Insertion

```
echo "Les informations ont été importées";  
header("Location: servicesetproduits_creation.php");
```

Après l'insertion réussie des données dans la base de données, un message est affiché, et l'utilisateur est redirigé vers une autre page (`servicesetproduits_creation.php`).

Modification

1. Récupération de l'ID du Produit

```
$id = $_GET['id'];
```

L'ID du produit à modifier est récupéré à partir de la requête GET. Cela signifie que l'ID est probablement inclus dans l'URL, par exemple, `modifier.php?id=1`.

2. Requête pour Afficher les Informations du Produit

```
$req = mysqli_query($db, "SELECT * FROM produits WHERE id = $id");  
$row = mysqli_fetch_assoc($req);
```

Une requête SQL est effectuée pour récupérer les informations du produit à partir de la base de données, en utilisant l'ID récupéré. Les données du produit sont stockées dans le tableau associatif `\$row`.

3. Vérification du Bouton Modifier

```
if (isset($_POST['ok'])) {  
    // ...  
}
```

Cette condition vérifie si le formulaire de modification a été soumis en vérifiant si le bouton "Modifier" a été cliqué (utilisation de `\$_POST['ok']`).

4. Traitement des Données du Formulaire

```
$titre = mysqli_real_escape_string($db, $_POST['titre']);  
$texte = mysqli_real_escape_string($db, $_POST['texte']);  
$salt = mysqli_real_escape_string($db, $_POST['alt_text']);
```

Les données du formulaire sont récupérées et échappées pour éviter les injections SQL.

5. Requête de Modification dans la Base de Données

```
$update_query = "UPDATE produits SET titre = '$titre', description = '$  
$req = mysqli_query($db, $update_query);
```

Une requête SQL de mise à jour est exécutée pour modifier les données du produit dans la base de données en fonction de l'ID.

6. Redirection après la Modification

```
if ($req) {  
    header("location: servicesetproduits_creation.php");  
} else {  
    $message = "Produit non modifié";  
}
```

Si la requête de modification est réussie, l'utilisateur est redirigé vers une autre page (`servicesetproduits_creation.php`). Sinon, un message d'erreur est stocké dans la variable `\$message`.

Suppression

1. Récupération de l'ID du Produit

```
$id = $_GET['id'];
```

L'ID du produit à supprimer est récupéré à partir de la requête GET.

2. Requête pour Récupérer les Informations du Produit

```
$query = "SELECT * FROM produits WHERE id= $id";  
$result = $db->query($query);  
while ($row = $result->fetch_assoc()) {  
    // ...  
}
```

Une requête SQL est effectuée pour récupérer les informations du produit à partir de la base de données, en utilisant l'ID récupéré. Les données du produit sont stockées dans le tableau associatif `\$row`.

3. Construction du Chemin de l'Image à Supprimer

```
$image_path = __DIR__ . '/../../images/servicesetproduits/' . $row['i
```

Le chemin complet de l'image associée au produit est construit en utilisant le chemin du répertoire des images et le nom de fichier stocké dans la base de données.

4. Suppression de l'Image

```
if (file_exists($image_path)) {  
    unlink($image_path); // Supprimer le fichier  
    // Ajoutez ici toute autre logique nécessaire, comme mettre à jour  
    echo "L'image a été supprimée avec succès."  
    // Suppression du produit dans la base de données  
    if (mysqli_query($db, "DELETE FROM produits WHERE id = $id")) {  
        header("Location: servicesetproduits_creation.php");  
    }  
} else {  
    echo "L'image n'existe pas ou a déjà été supprimée."  
}
```

Le script vérifie si le fichier image existe avant de le supprimer. Si l'image existe, elle est supprimée à l'aide de la fonction `unlink`. Ensuite, une requête SQL de suppression est exécutée pour supprimer le produit de la base de données. Si la suppression réussit, l'utilisateur est redirigé vers une autre page (`servicesetproduits_creation.php`).

5. Affichage des Images Restantes dans le Répertoire

```
$images_directory = __DIR__ . '/../../images/servicesetproduits/';  
$images = scandir($images_directory);
```

Le script utilise la fonction `scandir` pour obtenir la liste des fichiers dans le répertoire des images.