

Proyecto PHP

Mountain Connect

He creado el código de register.php en la que tiene un formulario dentro del html, el formulario recogerá los datos del usuario, y lo enviará de manera segura (al usar el POST), cada campo span MUESTRA UN ERROR.

En el nivel de experiencia, en caso de que haya un error en algún campo anterior, siga seleccionado el nivel de experiencia, y no se borre. Con el isset comprueba si existe y con el "selected". Se mantiene el valor seleccionado. Al igual pasa con la provincia seleccionada.

```
<option value="Principiante" <?=(isset($experiencia) && $experiencia === "Principiante") ? "selected" : "" ?>>Principiante</option>
```

El código register.php tiene todas las validaciones. Usando el método de la caja blanca.

Validaciones:

1. Evitar usuarios repetidos (nombres repetidos exactamente, no vale Lili y lili, esto querría decir que son diferentes usuarios) y te permite crearlo.

```
foreach ($_SESSION['users'] as $user){
    if ($user['username'] === $username) {
        $errors['username'] = "El nombre del usuario ya está en uso.";
        break;
    }
}
```

2. Evitar correos repetidos, en este punto, da igual mayúsculas o minúsculas, si pones (a@gmail.com y A@GMAIL.COM) te lo pillará todo en minúsculas, por lo que si hay un usuario con ese correo ya creado, no te permitirá crearlo, por lo que dará error

```
if (strtolower ($user['email']) === strtolower($email)) {
    $errors['email']= "El email ya está registrado / en uso";
    break;
}
```

3. Obligatorio todos los campos, no se puede quedar vacío ningún campo. Por ejemplo: Si el campo email está vacío, manda un error.

```
if (empty($email)) {
    $errors['email'] = "El email es obligatorio.";`
```

4. Formato de correo electrónico, php tiene su filtro de validación de correo electrónico.

```
elseif (!filter_var($email, FILTER_VALIDATE_EMAIL))
```

5. La contraseña tiene que tener más de 8 dígitos, strlen sirve para leer la longitud de una cadena de texto

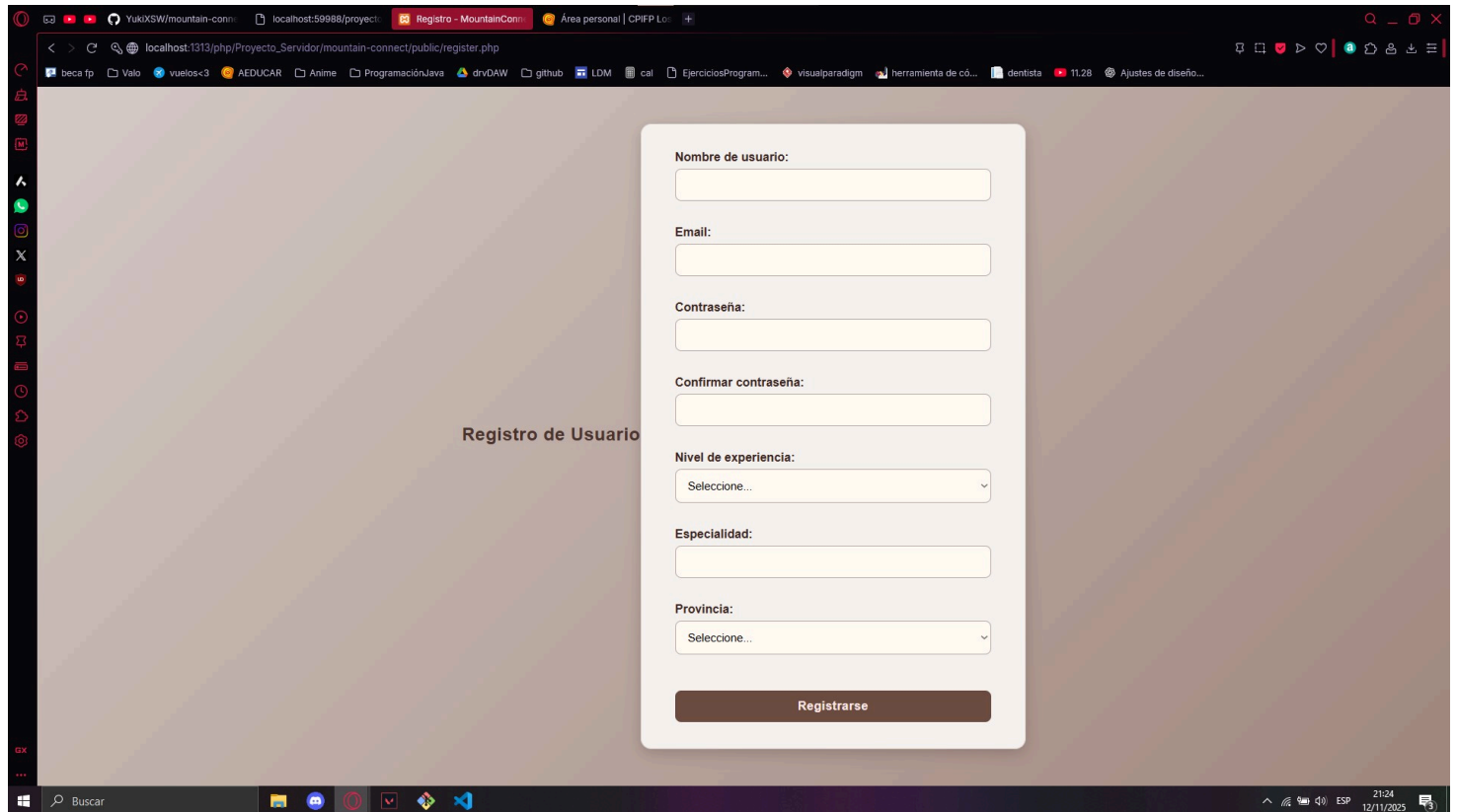
```
} elseif (strlen($password) < 8) {
    $errors['password'] = "La contraseña debe tener mínimo 8 caracteres."; # Si la contraseña tiene menos de 8 caracteres mostr
}
```

6. La contraseña confirmada tiene que coincidir con la contraseña

```
$password !== $confirm_password
```

Si no coincide envía un error

Aquí te presento la página de register.php



The screenshot shows a web browser window with the address bar displaying 'localhost:1313/php/Proyecto_Servidor/mountain-connect/public/register.php'. The page has a dark background with a light-colored registration form in the center. The form is titled 'Registro de Usuario' and contains the following fields:

- Nombre de usuario:
- Email:
- Contraseña:
- Confirmar contraseña:
- Nivel de experiencia:
- Especialidad:
- Provincia:

At the bottom of the form is a dark button labeled 'Registrarse'.

Al darle a registrar te tiene que llevar a la pagina de login.php.

En la que me he encontrado problemas como por ejemplo. Me creaba bien los usuarios, sin embargo a la hora de hacer login, me daba error en las credenciales, y en la unica que no me daba error era en la primera cuenta creada. El problema se situaba en el array temporal. Guarde el array con valores y por ello me daba error.

```
$users_registered = $_SESSION['users'] ?? [];
```

Aquí guardas los usuarios que ya se han registrado.

En la página de login.php, tiene que buscar si el usuario existe, y si no existe, mandar un error. Al igual que manda un error si la contraseña esta incorrecta. Sin embargo, si existe, directamente te manda a la pagina de profile.php.

Iniciar sesión

Usuario o Email:

Contraseña:

Entrar

En la página de profile.php hemos conectado header de includes *includes/header.php*

```
<?php include '../includes/header.php'; ?>
```

Y el footer *includes/footer.php*

```
<?php include '../includes/footer.php'; ?>
```

Y código html 🤖

 MountainConnect Inicio Rutas Perfil Cerrar sesión 



Bienvenido, a 🖐️

Email: a@gmail.com

Cerrar sesión