

Compte rendu projet en groupe CVVEN

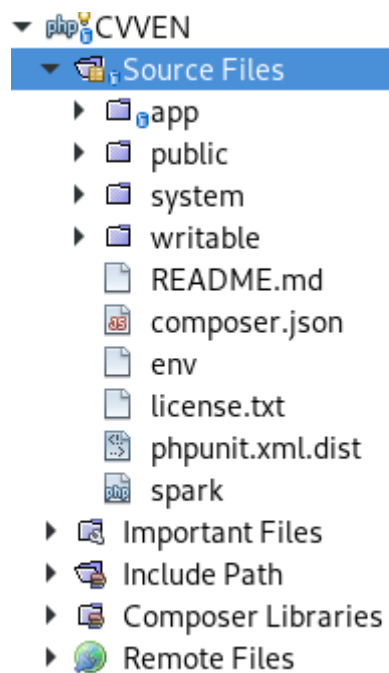
Objectif :

Crée une page de connexion et d'inscription, pouvoir modifier un utilisateur et hasher le mot de passe.

Framework :

Codigniter 4

Structure du projet :



Page d'inscription :

Inscription

Nom d'utilisateur:

test4321

Adresse e-mail:

test4321@gmail.com

Mot de passe:

●●●●●●●●

Confirmer le mot de passe:

●●●●●●●●|

S'inscrire

Base de donnée :

7	9	test4321	\$2y\$10\$3f0xeYU7xDzybz2mKsVkm.B7lPbVCmJCJV5dfAPPB7FZdooSVLU...	test4321@gmail.com	2024-04-28 02:21:31.012825
---	---	----------	--	--------------------	----------------------------

Gestion des erreurs :

J'ai rentrer le même nom d'utilisateur

Inscription

Nom d'utilisateur:

Ce nom d'utilisateur existe déjà.

Adresse e-mail:

Mot de passe:

Confirmer le mot de passe:

Les mots de passe ne correspondent pas.

Contrôler

```

<?php

namespace App\Controllers;

use App\Models\User_model; // Importez la classe User_model

class SignupController extends BaseController
{
    public function signup()
    {
        // Charge la vue d'inscription
        return view('Pages/Signup');
    }

    public function processSignup()
    {
        // Définition des règles de validation
        $validationRules = [
            'username' => [
                'rules' => 'required|is_unique[users.username]',
                'errors' => [
                    'required' => 'Le nom d\'utilisateur est requis.',
                    'is_unique' => 'Ce nom d\'utilisateur existe déjà.'
                ]
            ],
            'email' => [
                'rules' => 'required|valid_email|is_unique[users.email]',
                'errors' => [
                    'required' => 'L\'adresse e-mail est requise.',
                    'valid_email' => 'Le format de l\'adresse e-mail est incorrect.',
                    'is_unique' => 'Cette adresse mail existe déjà.'
                ]
            ],
            'password' => [
                'rules' => 'required|min_length[8]',

```

```

        'password' => [
            'rules' => 'required|min_length[8]',
            'errors' => [
                'required' => 'Le mot de passe est requis.',
                'min_length' => 'Le mot de passe doit comporter au moins 8 caractères.'
            ]
        ],
        'confirm_password' => [
            'rules' => 'required|matches[password]',
            'errors' => [
                'required' => 'La confirmation du mot de passe est requise.',
                'matches' => 'Les mots de passe ne correspondent pas.'
            ]
        ]
    ];

    // Validation des données
    if (!$this->validate($validationRules)) {
        $data['validation'] = $this->validator; // Passer le validateur lui-même
        return view('Pages/Signup', $data);
    }

    // Récupération des données du formulaire
    $username = $this->request->getPost('username');
    $email = $this->request->getPost('email');
    $password = $this->request->getPost('password');

    // Insertion de l'utilisateur dans la base de données
    $userModel = new User_model(); // Supposons que votre modèle s'appelle User_model
    $userData = [
        'username' => $username,
        'email' => $email,
        'password' => password_hash($password, PASSWORD_BCRYPT) // Hashage du mot de passe
    ];
    $userModel->insert($userData);

    // Redirection vers une page de confirmation
    return redirect()->to(site_url('login'))->with('success', 'Inscription réussie. Vous pouvez maintenant vous connecter.');
```

Vue :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Inscription</title>
  <style>
    /* Styles généraux */
    body {
      font-family: Arial, sans-serif;
      background-color: #f0f0f0;
      margin: 0;
      padding: 0;
    }

    .container {
      max-width: 400px;
      margin: 50px auto;
      background-color: #fff;
      border-radius: 5px;
      padding: 20px;
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    }

    h2 {
      text-align: center;
    }

    form {
      margin-top: 20px;
    }

    input[type="text"],
    input[type="email"],
    input[type="password"] {
```

```
input[type="email"],
input[type="password"] {
  width: 100%;
  padding: 10px;
  margin: 10px 0;
  border: 1px solid #ccc;
  border-radius: 5px;
  box-sizing: border-box;
}

button[type="submit"] {
  width: 100%;
  padding: 10px;
  border: none;
  border-radius: 5px;
  background-color: #4caf50;
  color: white;
  cursor: pointer;
}

button[type="submit"]:hover {
  background-color: #45a049;
}

.invalid-feedback {
  color: red;
  font-size: 0.8em;
}

</style>
</head>
<body>
```

```

</style>
</head>
<body>
<div class="container">
  <h2>Inscription</h2>
  <form action="<?= site_url('signup/processSignup') ?>" method="post">
    <?= csrf_field() ?>
    <label for="username">Nom d'utilisateur:</label><br>
    <input type="text" id="username" name="username" ><br>
    <?php if(isset($validation) && $validation->hasError('username')): ?>
      <div class="invalid-feedback"><?= $validation->getError('username') ?></div>
    <?php endif; ?>
    <label for="email">Adresse e-mail:</label><br>
    <input type="email" id="email" name="email" ><br>
    <?php if(isset($validation) && $validation->hasError('email')): ?>
      <div class="invalid-feedback"><?= $validation->getError('email') ?></div>
    <?php endif; ?>
    <label for="password">Mot de passe:</label><br>
    <input type="password" id="password" name="password" ><br>
    <?php if(isset($validation) && $validation->hasError('password')): ?>
      <div class="invalid-feedback"><?= $validation->getError('password') ?></div>
    <?php endif; ?>
    <label for="confirm_password">Confirmer le mot de passe:</label><br>
    <input type="password" id="confirm_password" name="confirm_password" ><br>
    <?php if(isset($validation) && $validation->hasError('confirm_password')): ?>
      <div class="invalid-feedback"><?= $validation->getError('confirm_password') ?></div>
    <?php endif; ?>
    <br>
    <button type="submit">S'inscrire</button>
  </form>
</div>

```

Connexion :

Connexion

Nom d'utilisateur:

Mot de passe:

Se connecter

S'inscrire

Gestion des erreurs :

Connexion

Nom d'utilisateur:

Ce nom d'utilisateur n'existe pas.

Mot de passe:

Le mot de passe est requis.

Se connecter

S'inscrire

Vue :

```
<body>
  <div class="container">
    <h2>Connexion</h2>
    <form action="<?= site_url('login/processlogin') ?>" method="post">
      <label for="username">Nom d'utilisateur:</label><br>
      <input type="text" id="username" name="username" ><br>
      <?php if(isset($validation) && $validation->hasError('username')): ?>
        <div class="invalid-feedback"><?= $validation->getError('username') ?></div>
      <?php endif; ?>
      <label for="password">Mot de passe:</label><br>
      <input type="password" id="password" name="password" ><br><br>
      <?php if(isset($validation) && $validation->hasError('password')): ?>
        <div class="invalid-feedback"><?= $validation->getError('password') ?></div>
      <?php endif; ?>
      <input type="submit" value="Se connecter">
    </form>
    <form action="<?= site_url('signup') ?>" method="get">
      <input type="submit" value="S'inscrire">
    </form>
  </div>
</body>
```

Modification du profil :

J'ai ajouter le lien profil dans la navbar



Profil

Votre nom d'utilisateur:

Nouveau nom d'utilisateur:

Votre adresse email:

Nouvelle adresse e-mail:

Nouveau mot de passe:

Confirmer le mot de passe:

Message :

Profil

Votre profil a été mis à jour avec succès.

Base de donnée :

7	9	testyanis854	\$2y\$10\$z6LS28FHvYdsOx0htoRA30HV9wJ3piuQxdPsZgEQv2nbWGV3Goe...	yanis4886@gmail.com	2024-04-28 02:21:31.012825
---	---	--------------	--	---------------------	----------------------------

Gestion des erreurs :

Profil

Votre nom d'utilisateur:

test4321

Nouveau nom d'utilisateur:

Le nom d'utilisateur est requis.

Votre adresse email:

test4321@gmail.com

Nouvelle adresse e-mail:

L'adresse e-mail est requise.

Nouveau mot de passe:

Le mot de passe est requis.

Confirmer le mot de passe:

La confirmation du mot de passe est requise.

Vue :

```
<div class="container">
  <h2>Profil</h2>
  <?php if (session()->getFlashdata('success')): ?>
    <div class="alert alert-success" role="alert">
      <? session()->getFlashdata('success'); ?>
    </div>
  <?php endif; ?>
  <?php if (session()->getFlashdata('error')): ?>
    <div class="alert alert-danger" role="alert">
      <? session()->getFlashdata('error'); ?>
    </div>
  <?php endif; ?>
  <form action="<? site_url('profil/enregistrement') ?>" method="post">
    <? csrf_field() ?>

    <label for="username_a">Votre nom d'utilisateur:</label><br>
    <input type="text" disabled="disabled" id="username_a" placeholder="<? session('user')['username'] ?>" ><br>

    <label for="username">Nouveau nom d'utilisateur:</label><br>
    <input type="text" id="username" name="username" ><br>
    <?php if(isset($validation) && $validation->hasError('username')): ?>
      <div class="invalid-feedback"><? $validation->getError('username') ?></div>
    <?php endif; ?>

    <label for="email_a">Votre adresse email:</label><br>
    <input type="email" disabled="disabled" id="email_a" placeholder="<? session('user')['email'] ?>" ><br>

    <label for="email">Nouvelle adresse e-mail:</label><br>
    <input type="email" id="email" name="email" ><br>
    <?php if(isset($validation) && $validation->hasError('email')): ?>
      <div class="invalid-feedback"><? $validation->getError('email') ?></div>
    <?php endif; ?>

    <label for="password">Nouveau mot de passe:</label><br>
    <input type="password" id="password" name="password" ><br>
    <?php if(isset($validation) && $validation->hasError('password')): ?>
      <div class="invalid-feedback"><? $validation->getError('password') ?></div>
    <?php endif; ?>

    <label for="confirm_password">Confirmer le mot de passe:</label><br>
    <input type="password" id="confirm_password" name="confirm_password" ><br>
    <?php if(isset($validation) && $validation->hasError('confirm_password')): ?>
      <div class="invalid-feedback"><? $validation->getError('confirm_password') ?></div>
    <?php endif; ?>
    <br>
    <button type="submit">Enregistrer</button>
  </form>
  <form action="<? site_url('profil/suppression') ?>" method="post">
    <button type="submit">Supprimer</button>
  </form>
</div>

</body>
</html>
```

Contrôleur:

k?php

```
namespace App\Controllers;

use App\Models\User_model; // Importez la classe User_model

class ProfilController extends BaseController
{
    public function index()
    {
        // Charge la vue du profil
        return view('Pages/profil');
    }

    public function enregistrement()
    {
        $validationRules = [
            'username' => [
                'rules' => 'required|is_unique[users.username]',
                'errors' => [
                    'required' => 'Le nom d\'utilisateur est requis.',
                    'is_unique' => 'Ce nom d\'utilisateur existe déjà.'
                ]
            ],
            'email' => [
                'rules' => 'required|valid_email|is_unique[users.email]',
                'errors' => [
                    'required' => 'L\'adresse e-mail est requise.',
                    'valid_email' => 'Le format de l\'adresse e-mail est incorrect.',
                    'is_unique' => 'Cette adresse mail existe déjà.'
                ]
            ],
            'password' => [
                'rules' => 'required|min_length[8]',
                'errors' => [
                    'required' => 'Le mot de passe est requis.',
                    'min_length' => 'Le mot de passe doit comporter au moins 8 caractères.'
                ]
            ],
            'confirm_password' => [
                'rules' => 'required|matches[password]',
                'errors' => [
                    'required' => 'La confirmation du mot de passe est requise.',
                    'matches' => 'Les mots de passe ne correspondent pas.'
                ]
            ]
        ];

        // Validation des données
        if (!$this->validate($validationRules)) {
            $data['validation'] = $this->validator; // Passer le validateur lui-même
            return view('Pages/profil', $data);
        }
        else{
            $username = $this->request->getPost('username');
            $email = $this->request->getPost('email');
            $password = $this->request->getPost('password');

            $id = session('user')['id'];

            $usermodel = new User_model();
            $modification = $usermodel->UpdateUser($id, $username, $password, $email);

            if ($modification == false) {
                session()->setFlashdata('error', 'La mise à jour de votre profil a rencontré une erreur.');
```

```

        $modification = $usermodel->updateUser($id, $password, $password, $email);

        if ($modification == false) {
            session()->setFlashdata('error', 'La mise à jour de votre profil a rencontré une erreur.');
```

}

```

        } else {
            session()->setFlashdata('success', 'Votre profil a été mis à jour avec succès.');
```

}

```

        return redirect()->to(site_url('profil'));
    }
}

public function delete(){
    $id = session('user')['id'];
    $usermodel = new User_model();
    $suppression = $usermodel->DeleteUser($id);

    if ($suppression == true) {
        // Si la suppression réussit, déconnectez l'utilisateur
        return redirect()->to(site_url('logout'));
    } else {
        // Si la suppression échoue, redirigez vers la page de profil avec un message d'erreur
        session()->setFlashdata('error', 'La suppression du profil a rencontré une erreur.');
```

return redirect()->to(site_url('profil'));

```

    }
}

```