

## TP 5 : Création d'un espace administrateur et abonné

Pour poursuivre notre site, nous allons maintenant prévoir une partie permettant à l'administrateur de se connecter au site pour accéder à des options non visibles par les visiteurs du site (par ex : le menu Ajouter une recette, le bouton modifier)

- La première étape consiste à créer un formulaire de création de compte
  - Créer une table **users** dans la base de données

Columns	
🔑	<b>id</b> int Primary Key
📦	<b>identifiant</b> varchar(100)
📦	<b>password</b> varchar(100)
📦	<b>mail</b> varchar(100)
📦	<b>create_time</b> datetime Cr...
📦	<b>isAdmin</b> tinyint(1)

Détail de la table users

- Créer un dossier **User** dans **Views**
- Créer le modèle associé **User** dans **Models** sur le modèle de **Recette**
- Créer une vue **inscription.php** dans ce dossier avec le formulaire associé

```
<h1>Inscription</h1>
<form action="?c=User&a=inscrire" method="post">
    <div class="mb-3">
        <label for="identifiant" class="form-label">Identifiant</label>
        <input type="text" class="form-control" name="identifiant" id="identifiant" required>
    </div>
    <div class="mb-3">
        <label for="mail" class="form-label">Adresse mail</label>
        <input type="email" class="form-control" name="mail" id="mail" required>
    </div>
    <div class="mb-3">
        <label for="pwd" class="form-label">Mot de passe</label>
        <input type="password" class="form-control" name="pwd" id="pwd" required>
    </div>
    <div class="mb-3">
        <button type="submit" class="btn btn-primary" id="enregistrer">Enregistrer</button>
    </div>
</form>
```

- Créer une vue **enregistrement.php** dans le dossier **User** pour afficher la réussite de l'enregistrement

```
<div class="alert alert-success">L'ajout de l'utilisateur a été effectué avec succès</div>
<a href="?c=home" class="btn btn-primary">Retour à l'accueil</a>
```

- Créer le contrôleur **UserController.php** avec une fonction **inscription** qui permet d'afficher la vue **inscription.php**

- Créer aussi la fonction **enregistrer** dans ce même contrôleur. NB : pour le mot de passe, merci d'utiliser **password\_hash(\$pwd,PASSWORD\_DEFAULT)** pour encoder le mot de passe dans la base de données
- Modifier le menu de navigation afin d'intégrer 2 boutons : **Inscription** et **Connexion**



```
<!-- menu de navigation -->
<nav class="navbar navbar-expand-lg bg-body-tertiary justify-content-between">
    <ul class="navbar-nav">
        ...
    </ul>
    <ul class="navbar-nav">
        <li class="nav-item">
            <a class="btn btn-outline-dark" href="?c=User&a=inscription">Inscription</a>
        </li>
        <li class="nav-item">
            <a class="btn btn-outline-dark" href="?c=User&a=connexion">Connexion</a>
        </li>
    </ul>
</nav>
```

- Créer l'utilisateur suivant :
  - Identifiant : admin
  - Mail : [contact@lacosina.fr](mailto:contact@lacosina.fr)
  - Password : azerty
- Modifier le contenu de la base de données, mettre l'utilisateur **admin** en tant qu'administrateur (**isAdmin = 1**)
- Créer un formulaire de connexion :
  - Créer une vue **connexion.php** dans le dossier **User** avec le formulaire associé

```
<h1>Connexion</h1>
<form action="?c=User&a=connecter" method="post">
    <div class="mb-3">
        <label for="identifiant" class="form-label">Identifiant</label>
        <input type="text" class="form-control" name="identifiant" id="identifiant" required>
    </div>
    <div class="mb-3">
        <label for="pwd" class="form-label">Mot de passe</label>
        <input type="password" class="form-control" name="pwd" id="pwd" required>
    </div>
    <div class="mb-3">
        <button type="submit" class="btn btn-primary" id="connexion">Se connecter</button>
    </div>
</form>
```

- Créer la fonction connexion dans le contrôleur **UserController.php**
- Créer aussi la fonction **verifieConnexion** qui permet de vérifier la connexion de l'utilisateur :
  - Récupération des données du formulaire
  - Requête de vérification de l'identifiant
  - Vérification de l'existence de l'utilisateur et du mot de passe associé à l'aide de la fonction **password\_verify**

- Si c'est le cas, revenir à la **page d'accueil**, sinon afficher le message d'erreur suivant : **Identifiant ou mot de passe incorrect**.
  - Tester la connexion avec un mot de passe incorrect ou correct
- Il reste désormais à garder la connexion active, effectivement maintenant que la connexion est fonctionnelle, celle-ci n'a pas d'effet car il faut activer le principe des sessions pour enregistrer la connexion de l'utilisateur :
  - Dans le fichier **index.php**, ajouter au début du fichier la fonction **session\_start()** pour initialiser la gestion des sessions
  - Dans ma fonction **verifieConnexion** du fichier **UserController.php**, au moment où les identifiants sont vérifiés, ajouter l'id, l'identifiant, et le mail de l'utilisateur connecté dans la session :

```
// Fonction permettant de vérifier la connexion d'un utilisateur
function verifieConnexion() {
    // récupération des données de formulaire
    $identifiant = $_POST['identifiant'];
    $pwd = $_POST['pwd'];

    // requête de vérification de l'identifiant
    // on cherche l'utilisateur par son identifiant
    // attention au piège, il faut utiliser findBy pour récupérer l'utilisateur
    // car find a besoin de l'id, par contre, findBy renvoie un tableau d'utilisateurs
    // et on prend le alors que le premier utilisateur trouvé d'où le [0]
    $user = $this->userModel->findBy(['identifiant' => $identifiant])[0];

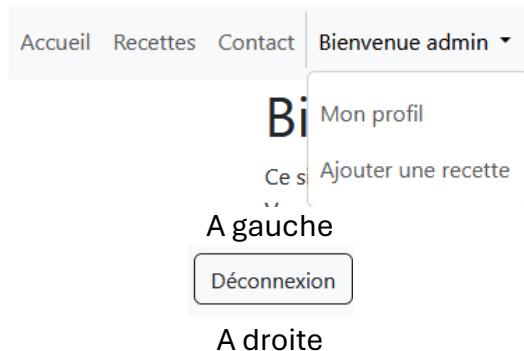
    // si l'utilisateur existe et le mot de passe est correct
    if($user && password_verify($pwd, $user['password'])) {
        // définition des variables de session
        $_SESSION['id'] = $user['id'];
        $_SESSION['identifiant'] = $user['identifiant'];
        $_SESSION['mail'] = $user['mail'];
        ...
    }
}
```

- Changer le bouton **Connexion** en **Déconnexion** et enlever le bouton **Inscription** si l'utilisateur est connecté

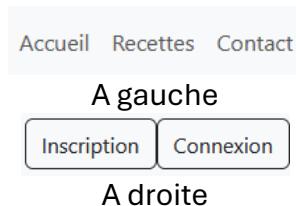
```
<ul class="navbar-nav">
<?php if(isset($_SESSION['identifiant'])) { ?>
    <li class="nav-item">
        <a class="btn btn-outline-dark" href='?c=User&a=deconnexion'>Déconnexion</a>
    </li>
<?php } else { ?>
    <li class="nav-item">
        <a class="btn btn-outline-dark" href='?c=User&a=inscription'>Inscription</a>
    </li>
    <li class="nav-item">
        <a class="btn btn-outline-dark" href='?c=User&a=connexion'>Connexion</a>
    </li>
<?php } ?>
</ul>
```

- Créer la fonction **deconnexion** dans le contrôleur **UserController.php**, celle-ci permet de supprimer la session à l'aide de la fonction **session\_destroy()** puis redirige vers l'accueil
- Nous allons modifier le menu, suivant si une personne est connectée ou pas :

- Menu version connectée :



- Menu version déconnecté :



- Il faut aussi prévoir que la **modification** des recettes ne peut se faire que si un utilisateur est connecté
- Créer la vue **profil.php** dans le dossier **User** : elle affichera le nom et le mail de l'utilisateur actuellement connecté (avec une image par défaut : `profil.png`)

### Profil de l'utilisateur : admin



**Identifiant :** admin  
**Email :** contact@lacosina.fr

*Profil de l'utilisateur*

---

[Retour à l'accueil](#)

```

<h1>Profil de l'utilisateur : <span id='profil_identifiant_titre'><?php echo
$user['identifiant']; ?></span></h1>
<div class="row">
    <div class="col">
        " class="card-img-top">
    </div>
    <div class="col">
        <p><b>Identifiant : </b><span id='profil_identifiant' data-id="<?php
echo $user['id']; ?>" contenteditable="true"><?php echo
$user['identifiant']; ?></span></p>
        <p><b>Email : </b><span id='profil_mail' data-id="<?php echo
$user['id']; ?>" contenteditable="true"><?php echo
$user['mail']; ?></span></p>
    </div>
</div>
<hr>
<div id="boutons">
    <button id="bouton_modifier_profil" class="btn btn-primary d-
none">Modifier le profil</button>
    <a href="?c=home" class="btn btn-primary">Retour à l'accueil</a>
</div>

```

- Une dose de Javascript pour finir :
  - Renommer le fichier **front.js** en **recipes.js**
  - Créer le fichier **users.js**
  - Les intégrer dans le fichier **header.php**
  - Le bouton de modification de profil a pour classe **d-none** (pour **display:none**), il est donc caché à l'ouverture de la page
  - Le fichier **users.js**, doit permettre d'afficher le bouton de modification quand l'identifiant ou le mail est modifié

```

// Écoute le chargement du DOM
document.addEventListener('DOMContentLoaded', () => {

    // Sélection du profil identifiant (contenu modifiable)
    let profil_identifiant = document.getElementById('profil_identifiant');
    // Sélection du profil mail (contenu modifiable)
    let profil_mail = document.getElementById('profil_mail');
    // Sélection du bouton de modification
    let modifier_profil = document.getElementById('bouton_modifier_profil');

    // Ajoute un écouteur d'événements pour afficher le bouton de modification
    // lorsque le contenu du profil est modifié
    profil_identifiant.addEventListener('input', (event) => {
        modifier_profil.classList.remove('d-none'); // Affiche le bouton de
modification
    });

    // Ajoute un écouteur d'événements pour afficher le bouton de modification
    // lorsque le contenu du profil est modifié
    profil_mail.addEventListener('input', (event) => {
        modifier_profil.classList.remove('d-none'); // Affiche le bouton de
modification
    });
});

```

- Créer un nouvel utilisateur :
  - Identifiant : robert
  - Mail : [robert@jmail.com](mailto:robert@jmail.com)
  - Mdp : roro87