

PROJET TUTORE S3 : LE QUARTIER DES PLAISIRS

Matthieu Schuler / Matthieu Tran

IUT D'ORSAY TP3A1

Table des matières

Introduction	2
Diagramme de cas d'utilisation	3
Schéma relationnel	3
Diagramme de Classe	4
Planification	5
Organigramme des tâches :	5
Grandes activités et tâches effectuées	6
Gantt	7
Parties techniques du code qui présentent un intérêt particulier	8
Ajout Recette	8
Partie Connexion/Inscription	16
Conclusion	20
Annexes	20
Comment tester notre site (Important)	20
Les fonctionnalités de notre site	21
Page d'accueil (dépend du nombre de recette)	21
Consulter une recette	22
Connexion ou Inscription	23
Ajouter une recette	24
Ajouter modérateur	24
Nos recettes (s'il y en a)	24
Rechercher recette	25
Supprimer Recette	25
Supprimer Commentaire	25

Introduction

Dans le cadre d'un projet universitaire en binôme, nous avons pour devoir de créer un nouveau site de recettes entre amis. Dans le contexte actuel dans lequel on est, c'est-à-dire une pandémie mondiale, quoi de mieux que de se faire de bons petits plats savoureux pour maintenir nos magnifiques silhouettes. L'objectif de ce projet est relativement simple, mettre en situation les compétences apprises en programmation Web et en base de données afin de pouvoir lier les deux afin de rendre un projet commun. De plus, ce projet va aussi nous permettre d'améliorer notre travail de groupe et par extension nous apprendre plus en profondeur comment séparer les tâches de ce dernier dans le but de gagner en rapidité. Nous allons avoir besoin d'utiliser plusieurs outils de communication afin d'améliorer l'efficacité de notre productivité du travail. Pour l'exemple, nous allons nous baser sur GitHub pour le partage de fichier.

Ce nouveau site de recette entre amis que l'on appellera « Le quartier des plaisirs » mettra en relation plusieurs acteurs. Il permettra, tout d'abord, à un utilisateur de pouvoir enregistrer une recette. Ensuite les autres utilisateurs pourront commenter la recette afin de donner leur avis sur cette dernière. De plus, chaque utilisateur déposant une recette sera considéré comme administrateur de cette dernière. Il pourra donc la modifier et la supprimer. Enfin, il y aura deux super Administrateurs, les créateurs du projet qui ont les droits d'accès sur toutes les fonctionnalités du site.

Database Connection in PHP with MySQL in Xampp Code



Diagramme de cas d'utilisation

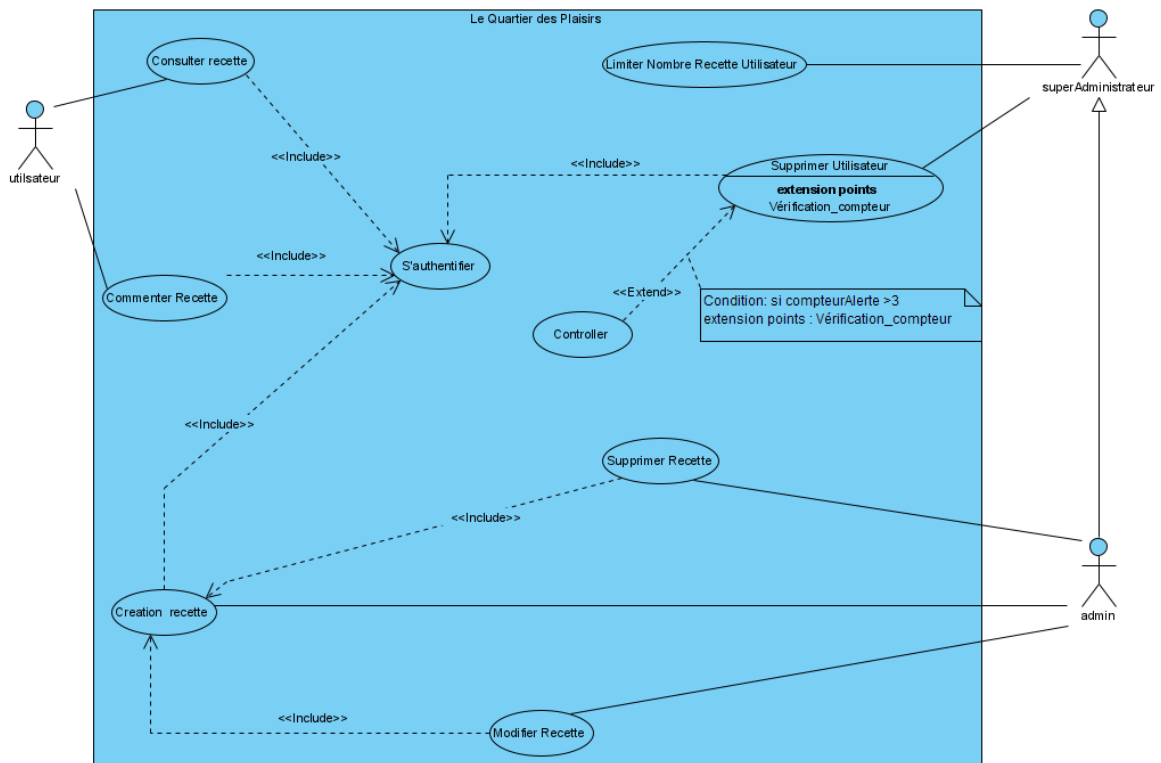


Schéma relationnel

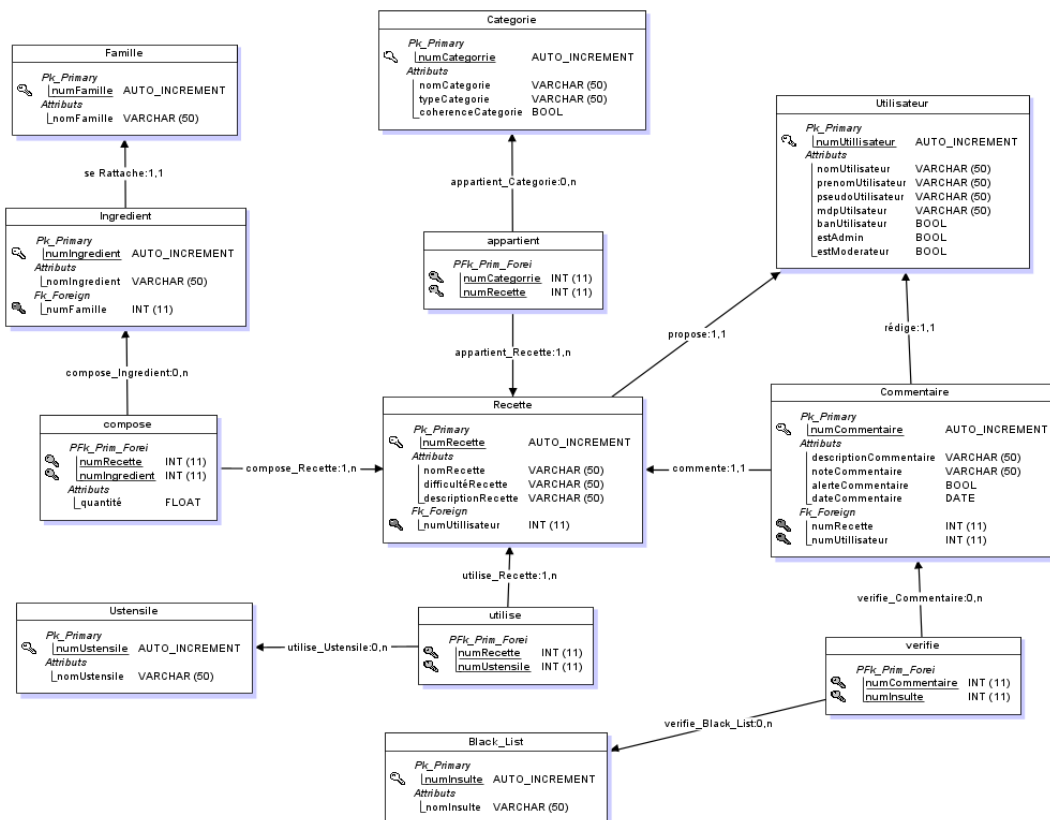
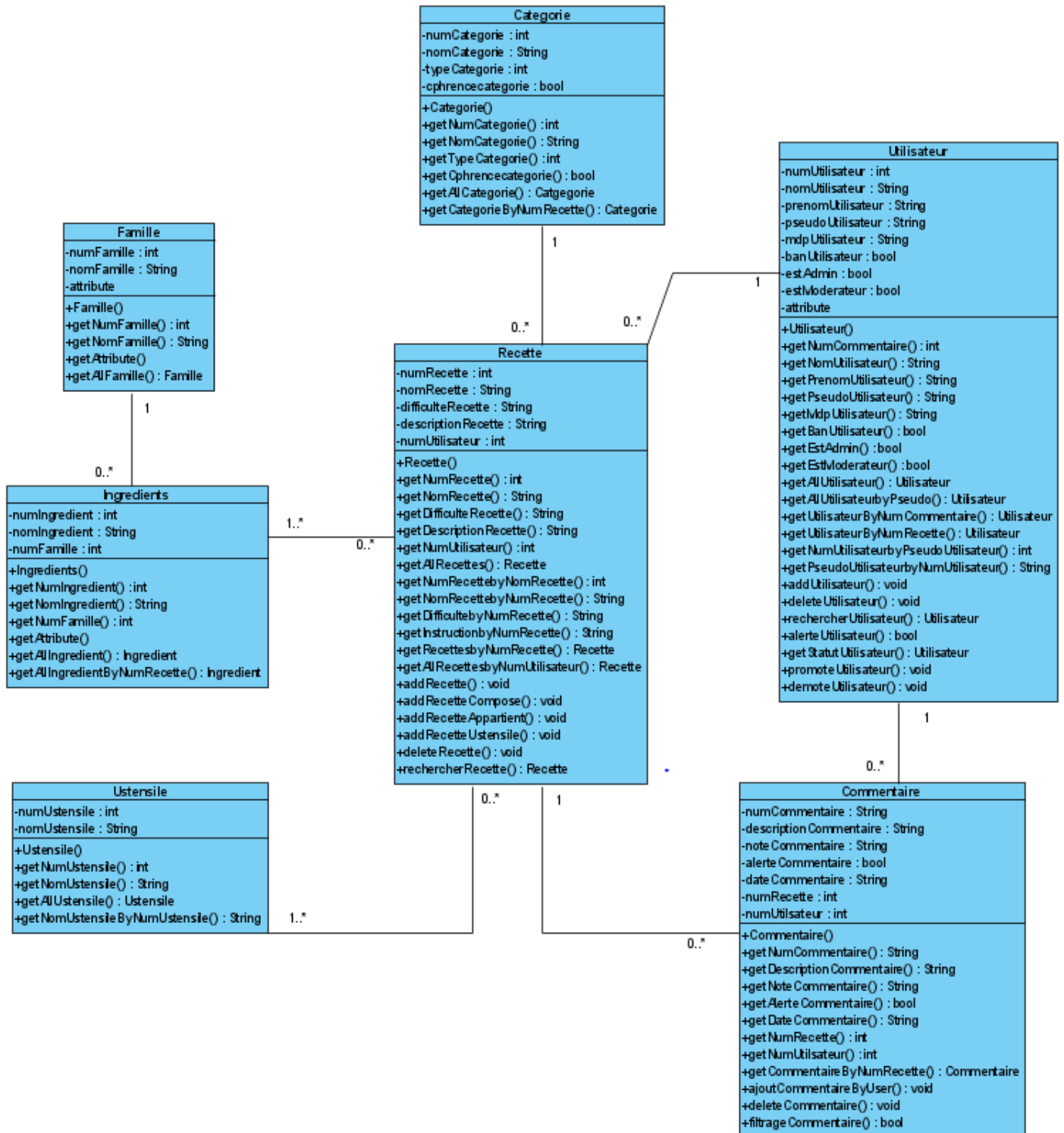


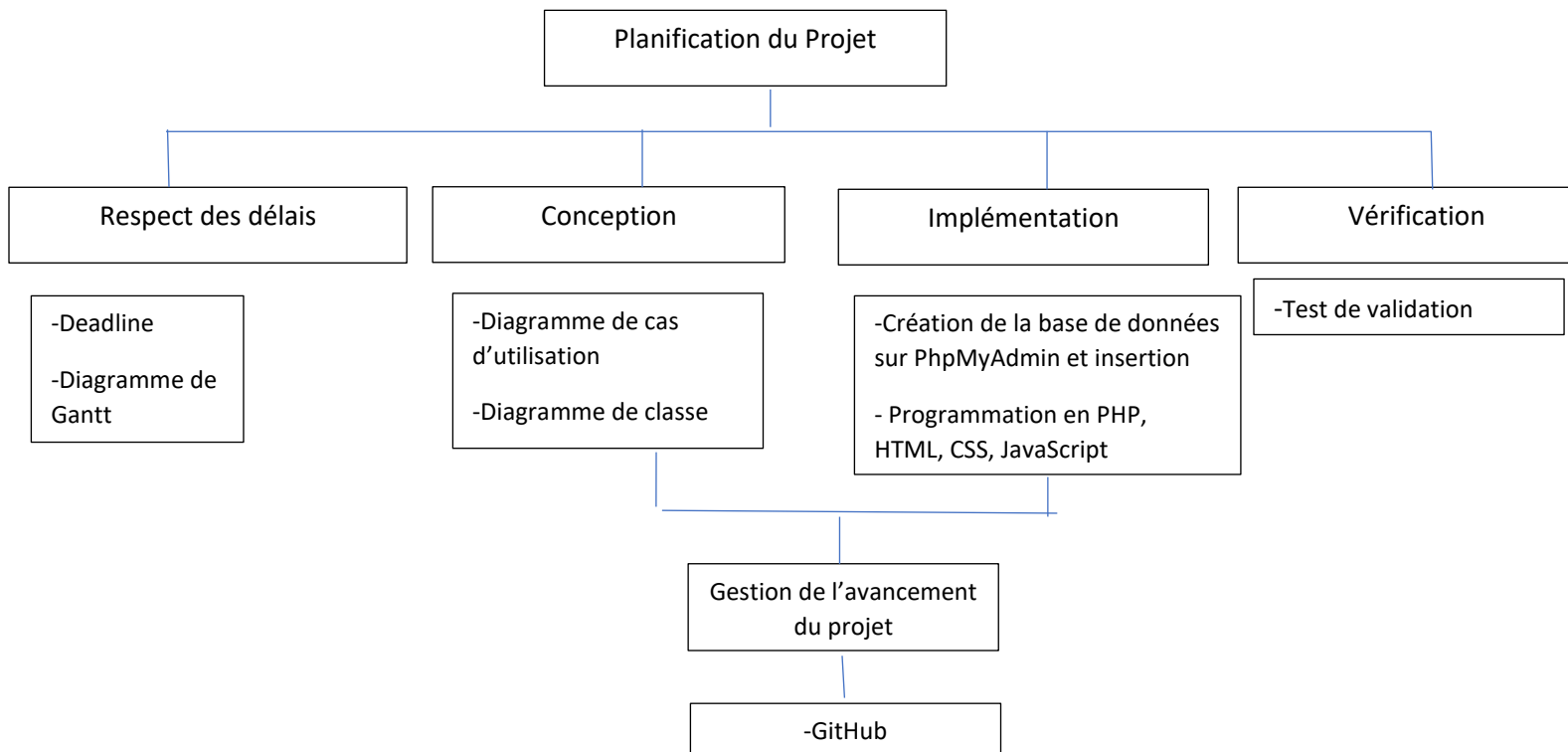
Diagramme de Classe



Planification

Pour avoir une meilleure vision des différentes tâches à effectuer durant notre projet, nous allons réaliser un organigramme des tâches.

Organigramme des tâches :



Grandes activités et tâches effectuées

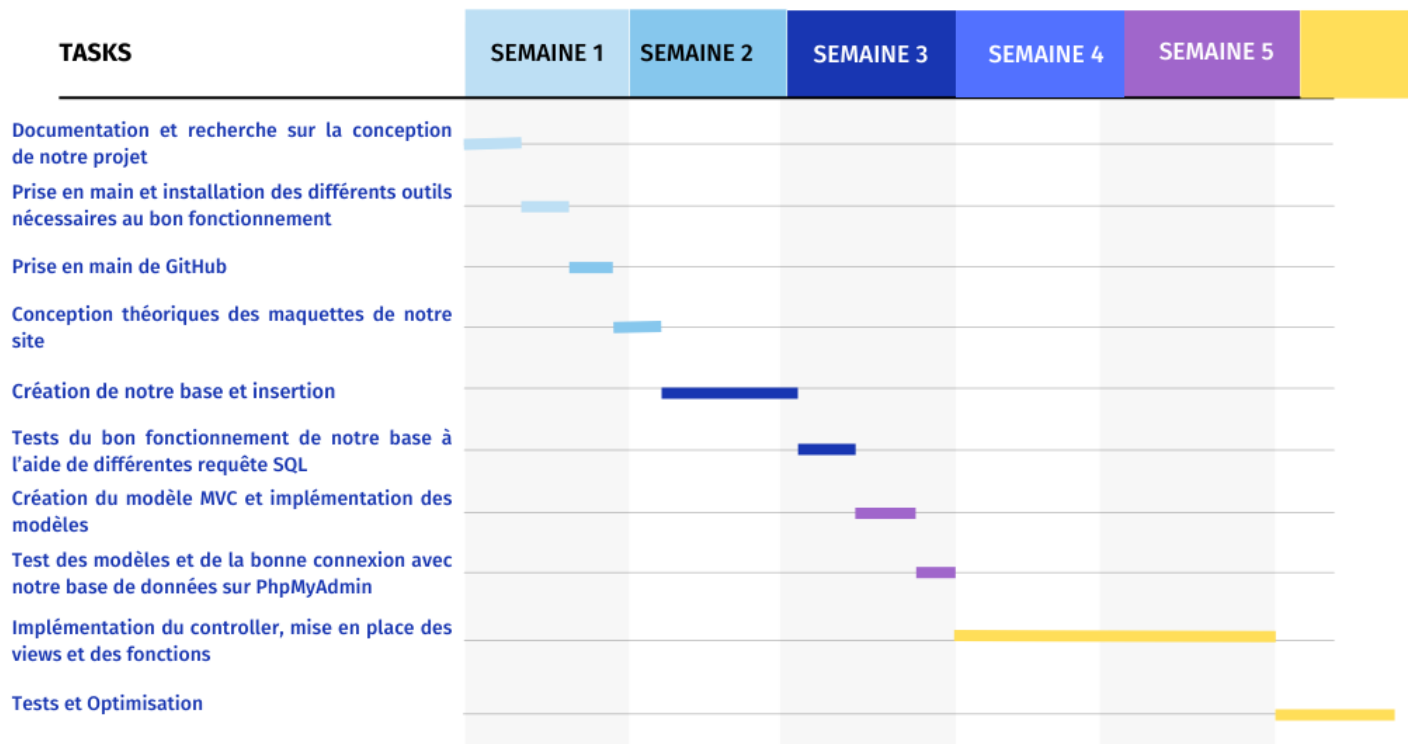
Taches	Durée (en jours)	Antécédent
Documentation et recherche sur la conception de notre projet (comment lier une base de données avec un site web en PHP) (A)	2	
Prise en main et installation des différents outils nécessaires au bon fonctionnement (XAMPP, Visual Studio Code, PhpMyAdmin, GIT) (B)	1	A
Prise en main de GitHub (C)	1	B
Conception théorique des maquettes de notre site (diagramme de cas d'utilisation, MCD, schéma relationnel, etc...) (D)	2	A/B
Création de notre base et insertion grâce aux scripts préparés à l'avance (E)	5	D/B
Tests du bon fonctionnement de notre base à l'aide de différentes requête SQL (F)	2	E
Création du modèle MVC et implémentation des modèles (G)	2	F
Test des modèles et de la bonne connexion avec notre base de données sur PhpMyAdmin (H)	1	G
Implémentation du controller, mise en place des views et des fonctions (I)	14	H
Tests et Optimisation (J)	5	I

Gantt



**Le Quartier
Des Plaisirs**

GANTT



Explication du Gantt :

Nous avons divisé le projet en 2 grandes parties :

- La première consiste à créer une base de données fonctionnelles pouvant fonctionner avec un site web géré en PHP
- La deuxième est donc d'implémenter la partie PHP permettant de répondre au besoin de créer un site de recette.

La bonne manipulation de l'outil Git est un point essentiel de la gestion de notre projet. En effet, cet outil nous permet de travailler chacun sur des parties différentes et donc d'avancer en parallèle.

Enfin l'implémentation de la base de données et de la partie PHP sont les parties centrales du projet qui nous ont demandé le plus de réflexion et sur laquelle nous avons consacré une grande partie du temps du projet.

Parties techniques du code qui présentent un intérêt particulier

Ajout Recette

Nous allons commencer par vous expliquer comment nous avons réalisé la fonctionnalité de l'ajout d'une recette. Voici le visuel attendu :



Ceci est la page d'accueil de notre site. Nous pouvons voir deux boutons situés dans le header. Un qui permet l'ajout de la recette et un deuxième qui octroie le droit d'ajouter un admin. Ce deuxième bouton est disponible uniquement car nous sommes connectés en tant qu'administrateur. Remarquons aussi que ces deux boutons sont disponibles uniquement car nous sommes connectés. Si cela n'est pas le cas, ces deux boutons ne sont pas visibles et un bouton « voir les recettes » serait à leur place.

→ Voici le code qui permet de réaliser cela :

```
<?php if (isset($_SESSION['user'])) { ?>
    <button onclick="window.location.href='routeur.php?action=create'" type="button" class="btn btn-sm btn-outline-primary ml-1">
        Ajouter une recette
    </button>
<?php } else { ?>
    <a href="#debut" class="btn btn-outline-light my-2 mr-3 my-lg-0">Voir les recettes</a>
<?php } ?>
<?php if (isset($_SESSION['Admin'])) {
    if ($_SESSION['Admin'] == 1) { ?>
        <button onclick="window.location.href='routeur.php?action=ajoutMod'" type="button" class="btn btn-primary ml-2">
            Ajouter un modérateur
        </button>
    <?php } ?>
<?php } ?>
```

La fonction `isset()` permet de tester si l'utilisateur est connecté et le cas échéant de savoir si c'est un administrateur et/ou un admin. Nous pouvons remarquer que ces valeurs sont contenues dans le tableau globale de `$_SESSION`. En effet lorsqu'un utilisateur se connecte, nous démarrons une session, ce qui permet de stocker plusieurs informations comme celles évoquées ci-dessus ou encore comme son `numUtilisateur` qui nous servira pour la suite

Ensuite, si nous appuyons sur le bouton « Ajouter une recette », la méthode `create` du `controller` est appelée et nous redirige vers la page `create.php` (cette méthode sera expliquée plus tard).

Voici ce qu'on obtient :

The screenshot shows a web application interface for adding a recipe. The header is orange and contains a logo, navigation links, a search bar, and a user account link. The main content area is titled "Partagez nous vos petits plaisirs" and contains a form with the following fields and buttons:

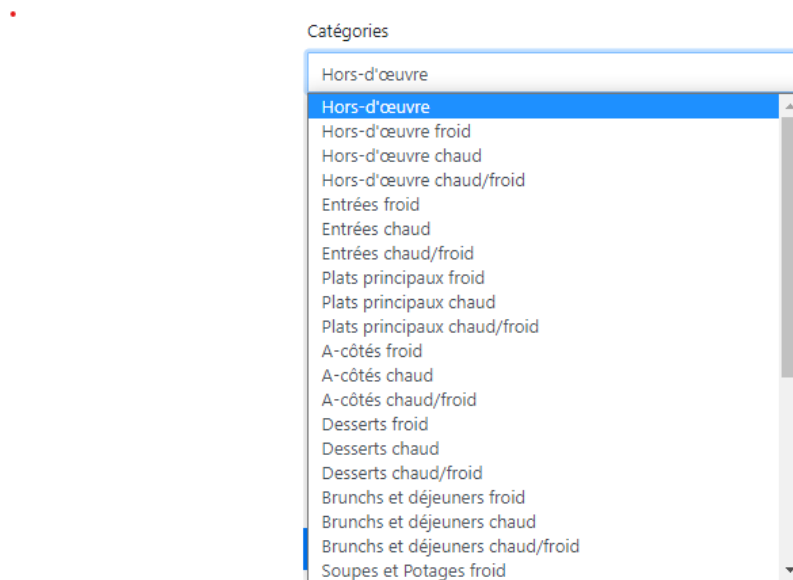
- Nom de la recette:** A text input field containing "Pâtes carbonara".
- Catégories:** A dropdown menu showing "Hors-d'œuvre".
- Difficulté:** A dropdown menu showing "Facile".
- Image:** A button "Choisir un fichier" and a text "Aucun fichier choisi".
- Instructions:** A text area containing "Instructions".
- Ingrédients:** A text input field containing "Aucun Ingredient".
- Quantités:** A text input field containing "... en g".
- Buttons:** A red "Supprimer" button, a blue "Ajouter un ingrédient" button, and a blue "Ajouter un ustensile" button.
- Ustensiles:** A text input field containing "Aucun Ustensiles".
- Buttons:** A red "Supprimer" button, a blue "Ajouter un ustensile" button, and a yellow "Ajouter la recette" button.

Ceci est donc un formulaire que nous avons mis avec la méthode `post` pour éviter que l'utilisateur change le numéro d'utilisateur à la création de la recette. Nous avons plusieurs champs qui utilisent les balises « `select` » et « `option` ». Prenons l'exemple du champs catégorie. Voici son code :

```
<div class="form-group col-md-6">
  <label for="numCategorie">Catégories</label>
  <select name="numCategorie" class="selectpicker form-control" required>
    <option name="numCategorie" value="1"><?php echo $lesCategories[0]['1']; ?></option>
    <?php foreach ($lesCategories as $key => $value) { ?>
      <option name="numCategorie" value="<?php echo $lesCategories[$key]['numCategorie']; ?>">
        <?php echo $lesCategories[$key]['nomCategorie'] . " " . $lesCategories[$key]['typeCategorie'] ?>
      </option>
    <?php } ?>
  </select>
</div>
```

Nous pouvons remarquer plusieurs éléments intéressants ici.

- ➔ Premièrement, le nom donné, « numCategorie » a été choisi pour éviter toute confusion dans notre code. Il correspond au même nom que nous avons mis dans la base de données. Cela va nous permettre de récupérer le numéro de la catégorie que l'utilisateur a choisi lorsque l'on va traiter l'envoi du formulaire.
- ➔ Ensuite, nous avons bouclé autour d'un tableau qui contient toutes les catégories et nous les affichons pour que l'utilisateur puisse choisir. Voici ce que cela donne :



Maintenant, passons à la partie la plus intéressante qui est l'ajout d'ingrédient et l'ajout d'ustensile. Ces deux fonctionnalités marchent de la même façon.

Nous voulions que lorsque l'on click sur le bouton « ajouter ingrédient », une nouvelle ligne apparaisse similaire à la précédente qui permet donc d'ajouter un autre ingrédient.

Voici ce que cela donne :

The screenshot shows a web form with two main sections: 'Ingrédients' and 'Ustensiles'. Each section has three rows of input fields. The 'Ingrédients' section has a header 'Ingrédients' and 'Quantités'. Each row contains a text input with the placeholder 'Aucun Ingredient', a number input with the placeholder '... en g', and a red 'Supprimer' button. Below the rows is a blue button labeled 'Ajouter un ingrédient'. The 'Ustensiles' section has a header 'Ustensiles' and three rows, each with a text input (placeholder 'Aucun Ustensiles') and a red 'Supprimer' button. Below these rows is a blue button labeled 'Ajouter un ustensile'.

Pour y arriver, nous avons dû utiliser du javascript, la classe d-none qui permet de ne pas afficher la div qui contient cette classe, un id selectIngredient qui permet que lorsque l'on supprime un ingrédient et que l'on veut en rajouter un, l'ingrédient supprimé ne soit pas de nouveau affiché (dans le code JavaScript) et une classe ingredient+un chiffre qui va permettre l'ajout et la suppression (dans le code JavaScript).

Voici le code HTML:

```
<?php for ($cpt = 1; $cpt < 10; $cpt++) { ?>
<div class="form-row d-none ingredient<?php echo $cpt ?>">
  <div class="form-group col-md-5">
    <select id="selectIngredient<?php echo $cpt ?>" name="numIngredient[]" class="selectpicker form-control" required>
      <option name="numIngredient" value="0"> Aucun Ingredient </option>
      <?php foreach ($lesIngredients as $key => $value) { ?>
        <option name="numIngredient" value="<?php echo $lesIngredients[$key]['numIngredient']; ?>">
          <?php echo $lesIngredients[$key]['nomIngredient'] ?>
        </option>
      <?php } ?>
    </select>
  </div>
  <div class="form-group col-md-5">
    <input type="number" class="form-control" name="quantite[]" min="0" max="10000" placeholder="... en g">
  </div>
  <div class="form-group col-md-2">
    <button class="btn btn-danger font-weight-bold mx-auto remove_ingredient">Supprimer</button>
  </div>
</div>
```

Dans la balise select, nous avons mis dans « name » numIngredient[]. Cela permet de mettre tous les numIngredients dans un tableau qui sera ensuite utiliser dans le traitement. Ensuite nous bouclons sur tous les ingrédients pour les afficher comme pour les catégories. On met aussi la quantité de chaque ingrédient dans un tableau de quantité.

Comme dit plus précédemment, nous avons dû faire un script JavaScript pour l'ajout et la suppression des ingrédients et des ustensiles.

Voici le code JavaScript :

```
<script type="text/javascript">
    $(document).ready(function() {
        // Fonctions pour les ingrédients
        var max_fields_ingredients = 10;
        var add_ingredient = $(".add_ingredient");
        var remove_ingredient = $(".remove_ingredient");
        var x = 0;

        $(add_ingredient).click(function(e) {
            e.preventDefault();
            if (x < max_fields_ingredients) {
                x++;
                $('#ingredient' + x).removeClass('d-none');
            } else {
                alert('Vous ne pouvez pas ajouter plus d\'ingrédients !');
            }
        });

        $(remove_ingredient).click(function(e) {
            e.preventDefault();
            if (x > 0) {
                if (x == 10)
                    x--;
                $('#ingredient' + x).addClass('d-none');
                var id = document.getElementById("selectIngredient" + x).selectedIndex;
                id = "0";
                x--;
            }
        });

        // Fonctions pour les ustensiles
        var max_fields_ustensiles = 10;
        var add_ustensile = $(".add_ustensile");
        var remove_ustensile = $(".remove_ustensile");
        var y = 0;
```

```
$(add_ustensile).click(function(e) {  
    e.preventDefault();  
    if (y < max_fields_ustensiles) {  
        y++;  
        $(' .ustensile' + y).removeClass('d-none');  
    } else {  
        alert('Vous ne pouvez pas ajouter plus d\'ustensiles !');  
    }  
});  
  
$(remove_ustensile).click(function(e) {  
    e.preventDefault();  
    if (y > 0) {  
        if (y == 10)  
            y--;  
        $(' .ustensile' + y).addClass('d-none');  
        var id = document.getElementById("selectUstensile" + x).selectedIndex;  
        id = "0";  
        y--;  
    }  
});  
});  
</script>
```

Si l'on veut faire simple, lorsqu'on ajoute un ingrédient, la fonction add_Ingrédient est appelé

```
$(add_ingredient).click(function(e) {  
    e.preventDefault();  
    if (x < max_fields_ingredients) {  
        x++;  
        $(' .ingredient' + x).removeClass('d-none');  
    } else {  
        alert('Vous ne pouvez pas ajouter plus d\'ingrédients !');  
    }  
});
```

Cette dernière enlève la classe d-none à la div qui a pour nom Ingredient+chiffre de l'ingrédient qu'on ajoute.

A l'inverse, si on supprime un ingrédient, la classe d-none est ajoutée à la div qui a pour nom Ingredient+chiffre de l'ingrédient qu'on supprime.

```
$(remove_ingredient).click(function(e) {  
    e.preventDefault();  
    if (x > 0) {  
        if (x == 10)
```

```
        x--;  
        $('<div>.ingredient' + x).addClass('d-none');  
        var id = document.getElementById("selectIngredient" + x).selectedIndex;  
        id = "0";  
        x--;  
    }</div>
```

Enfin, `var id = document.getElementById("selectIngredient" + x).selectedIndex;`

`id = "0";` permet de sélectionner « Aucun ingrédient » par défaut.

Passons ensuite à la partie traitement.

Pour envoyer le formulaire, il faut cliquer sur le bouton « Ajouter la recette ». Ce dernier va donc « submit » le formulaire et appelé la méthode choisie en début de fichier qui se situera dans le controller.

```
<form action="routeur.php" method="post" enctype="multipart/form-data">  
    <input type="hidden" name="action" value="created" readonly>
```

Voici le code de la méthode « created » :

```
public static function created()  
{  
    extract($_POST);  
    $cptIngredient = 0;  
    $cptQuantite = 0;  
    foreach ($quantite as $key => $value){  
        if($numIngredient[$key] != 0 )  
            $cptIngredient++;  
        if ($quantite[$key] != null)  
            $cptQuantite++;  
    }  
    $recette = Recette::getNumRecettebyNomRecette($nomRecette);  
    if ($recette == null && $numIngredient[0] != 0 && $numUstensile[0] != 0  
&& $cptQuantite != 0 && $cptQuantite == $cptIngredient) {  
        if (isset($_FILES['photo']['tmp_name'])) {  
            copy($_FILES['photo']['tmp_name'], 'view/image/' .  
$_FILES['photo']['name']);  
            $nomImage = $_FILES['photo']['name'];  
        }  
        Recette::addRecette($nomRecette, $difficulteRecette,  
$descriptionRecette, $_SESSION['numUtilisateur'], $nomImage);  
        $leNumRecette = Recette::getNumRecettebyNomRecette($nomRecette);  
        $leNumRecette = $leNumRecette['numRecette'];  
        Recette::addRecetteAppartient($numCategorie, $leNumRecette);  
        foreach ($numIngredient as $key => $value) {  
            if ($numIngredient[$key] == 0 || is_null($quantite[$key]))  
                continue;  
            Recette::addRecetteCompose($leNumRecette, $numIngredient[$key],  
$quantite[$key]);  
        }  
        foreach ($numUstensile as $key => $value) {  
            if ($numUstensile[$key] == 0)  
                continue;
```

```
        Recette::addRecetteUstensile($leNumRecette,  
$numUstensile[$key]);  
    }  
    header("Location:  
routeur.php?action=afficherRecette&numRecette=$leNumRecette");  
    die();  
}  
else if ($numUstensile[0] == 0 && $numIngredient[0] == 0) {  
    header("Location:  
routeur.php?action=create&ajout_err=oublieUstensileIngredient");  
    die();  
} else if ($numUstensile[0] == 0) {  
    header("Location:  
routeur.php?action=create&ajout_err=oublieUstensile");  
    die();  
} else if ($numIngredient[0] == 0) {  
    header("Location:  
routeur.php?action=create&ajout_err=oublieIngredient");  
    die();  
}  
else if ($cptQuantite==0 || $cptQuantite!=$cptIngredient) {  
    header("Location:  
routeur.php?action=create&ajout_err=oublieQuantite");  
    die();  
}  
else {  
    header("Location:  
routeur.php?action=afficherRecette&numRecette=$recette[0]&recetteExistante=  
true");  
    die();  
}  
}
```

Pour vérifier si l'utilisateur a bien mis des quantités, on Loop dans le tableau de quantités afin de regarder si le tableau contient des valeurs nulles. Et si ce n'est pas le cas, on augmente un compteur qui compte le nombre d'éléments dans cet array. On compte aussi le nombre d'ingrédient que l'utilisateur a mis et on incrémente aussi un compteur. On compare les deux et si, un des deux compteurs n'est pas le même que l'autre alors, cela veut dire que l'utilisateur a soit ajoute des ingrédients sans mettre de quantité ou bien, il a ajouté une quantité sans mettre le nom d'ingrédient. Si c'est le cas, alors, on envoie un message d'erreur qui dit que le nombre d'ingrédient n'est pas le même et que l'utilisateur doit recommencer.

```
if($cptQuantite==0 || $cptQuantite!=$cptIngredient) {  
    header("Location:routeur.php?action=create&ajout_err=oublieQuantite");  
    die();  
}
```

De même si le compteur de quantité est égal a 0.

Pour vérifier qu'on n'ajoute pas le même nom de recette on va appeler une fonction de la classe Recette qui cherche le nom de recette avec le numéro. Si cela retourne quelque chose alors,

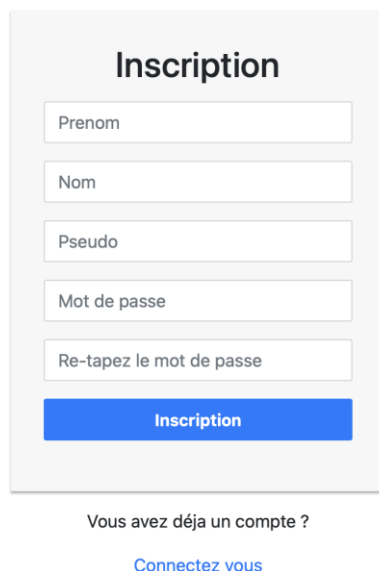
on redirige l'utilisateur vers la page de la recette avec un message d'erreur disant que la recette a déjà été créé.

On regarde aussi si l'array d'ustensile et l'array d'ingrédient ne sont pas égales à 0 afin d'ajouter la recette dans la base de données.

Ensuite, maintenant que toutes les conditions ont été réunies pour créer une recette, on ajoute dans les tables respectives, les attributs de la recette.

Pour stocker les photos, on va les stocker dans le serveur Webdev en copiant l'image dans un dossier 'Image' et le nom de l'image dans la table Recette qui contient un attribut « nom-image ». Ainsi, par la suite, pour afficher les images on va juste appeler l'attribut du nom de l'image qui correspond à la recette.

Partie Connexion/Inscription



The image shows a web form titled 'Inscription'. It contains five input fields: 'Prenom', 'Nom', 'Pseudo', 'Mot de passe', and 'Re-tapez le mot de passe'. Below these fields is a blue button labeled 'Inscription'. At the bottom of the form, there is a link that says 'Vous avez déjà un compte ? Connectez vous'.

Ceci est le form qui permet à l'utilisateur de s'inscrire sur le site. Il doit rentrer son prénom, nom et pseudo. Le nom et prénom ne sera pas affiché sur le site. Seul le pseudo permettra à l'utilisateur d'être identifié. Pour empêcher d'avoir plusieurs pseudos, on va appeler la fonction :

```
$requete = Utilisateur::getAllUtilisateurbyPseudo($pseudo);  
$row = $requete->rowCount();
```

Cette requête va permettre de récupérer des données sur des utilisateurs potentiels. On va récupérer le nombre de ligne de cette requête et si, elle renvoie un chiffre supérieur à 0 alors cela veut dire que le pseudo est déjà utilisé par un autre utilisateur. Sinon on va pouvoir continuer. On regarde aussi si le prénom et nom ne dépassent pas la limite autorisée par webdev. On a défini cette limite à 20 caractères.

```
if (strlen($prenom) <= 50) {  
    if (strlen($nom) <= 50) {
```

On fait aussi une petite vérification du mot de passe de l'utilisateur pour vérifier qu'il met bien son mot de passe. On hash aussi le mot de passe de l'utilisateur pour que seul lui puisse accéder à son compte. Même les administrateurs ne connaissent pas son mot de passe. Lorsque toutes les conditions ont été réunies. On peut créer l'utilisateur et on le redirige vers la page d'accueil.

Une autre fonctionnalité que l'on a ajoutée et que si une des erreurs citées précédemment se produit, on garde en mémoire les données qui sont bonnes comme le nom par exemple grâce au lancement de cookies. Par exemple si l'utilisateur enregistre un pseudo déjà présent dans la base de données alors, on va sauvegarder dans des cookies le nom et prénom de l'utilisateur.

```
public static function inscription_traitement()
{
    // Si les variables existent et qu'elles ne sont pas vides
    if (!empty($_POST['Prenom']) && !empty($_POST['Nom']) && $_POST['pseudo'] && !empty($_POST['password']) && !empty($_POST['password_retype'])) {
        // Patch XSS
        $pseudo = htmlspecialchars($_POST['pseudo']);
        $nom = htmlspecialchars($_POST['Nom']);
        $prenom = htmlspecialchars($_POST['Prenom']);
        $password = htmlspecialchars($_POST['password']);
        $password_retype = htmlspecialchars($_POST['password_retype']);
        // On vérifie si l'utilisateur existe

        $requete = Utilisateur::getAllUtilisateurbyPseudo($pseudo);
        $row = $requete->rowCount();

        // Si la requete renvoie un 0 alors l'utilisateur n'existe pas
        if ($row == 0) {
            if (strlen($prenom) <= 50) {
                if (strlen($nom) <= 50) {
                    if (strlen($pseudo) <= 20) { // On verifie que la longueur du pseudo <= 100
                        if ($password == $password_retype) { // si les deux mdp saisis sont bon
                            // On hash le mot de passe avec Bcrypt, via un coût de 12
                            $cost = ['cost' => 12];
                            $passwordHash = password_hash($password, PASSWORD_BCRYPT, $cost);
                            // On insère dans la base de données
                            Utilisateur::addUtilisateur($nom, $prenom, $pseudo, $passwordHash);
                            $numUtilisateur = Utilisateur::getNumUtilisateurbyPseudoUtilisateur($pseudo);
                            // On redirige avec le message de succès
                            $_SESSION['numUtilisateur'] = $numUtilisateur['numUtilisateur'];
                            $_SESSION['user'] = $pseudo;
                            header('Location: routeur.php?action=accueil');
                            die();
                        } else {
                            setcookie("pseudonyme", $_POST["pseudo"], time() + 5);
                            setcookie("nomUtilisateur", $_POST["Nom"], time() + 5);
                            setcookie("prenomUtilisateur", $_POST["Prenom"], time() + 5);
                            header('Location: routeur.php?action=inscription&reg_err=password');
                            die();
                        }
                    } else {
                        setcookie("nomUtilisateur", $_POST["Nom"], time() + 5);
                        setcookie("prenomUtilisateur", $_POST["Prenom"], time() + 5);
                        header('Location: routeur.php?action=inscription&reg_err=pseudo_length');
                        die();
                    }
                }
            }
        }
    }
}
```

```
        } else {
            setcookie("prenomUtilisateur", $_POST["Prenom"], time() + 5);
            setcookie("pseudonyme", $_POST["pseudo"], time() + 5);
            header('Location: routeur.php?action=inscription&reg_err=nom_length');
            die();
        }
    } else {
        setcookie("nomUtilisateur", $_POST["Nom"], time() + 5);
        setcookie("pseudonyme", $_POST["pseudo"], time() + 5);
        header('Location: routeur.php?action=inscription&reg_err=prenom_length');
        die();
    }
} else {
    setcookie("nomUtilisateur", $_POST["Nom"], time() + 5);
    setcookie("prenomUtilisateur", $_POST["Prenom"], time() + 5);
    header('Location: routeur.php?action=inscription&reg_err=already');
    die();
}
}
```

Connexion

☒ Remember username

Connexion

[Inscription](#)

Pour la fonctionnalité de connexions, on regarde si le pseudo existe dans la base de données avec la même méthode que l'on a fait pour l'inscription, On regarde ensuite si le mot de passe correspond avec celui de la base de données en utilisant une fonction existante de PHP. On démarre ensuite la session avec le numéro et pseudo de l'utilisateur si l'utilisateur se trouve dans la base de données et le mot de passe correspond aussi. On met aussi dans la variable de session le fait que l'utilisateur soit un administrateur ou non.

```
public static function connection_traitement()
{
    if (empty($_POST["rememberme"])) {
        setcookie("username", $_POST["Pseudo"], time() + 86400); // on set les cookies a un jour 86400
    } else {
        setcookie("username", null, time() - 1);
    }
    if (empty($_POST["Pseudo"]) && empty($_POST["password"])) // Si il existe les champs Pseudo, password et qu'ils sont pas vide
    {
        // Patch XSS
        $Pseudo = htmlspecialchars($_POST['Pseudo']);
        $password = htmlspecialchars($_POST['password']);
        $Pseudo = strtolower($Pseudo); // Pseudo transformé en minuscule
        // On regarde si l'utilisateur est inscrit dans la table utilisateurs
        $requete = Utilisateur::getAllUtilisateurbyPseudo($Pseudo);
        $data = $requete->fetch();
        $row = $requete->rowCount();
        // Si > à 0 alors l'utilisateur existe
        if ($row > 0) {
            if (password_verify($password, $data['mdpUtilisateur'])) {
                if ($data['estAdmin'] == 1)
                    $_SESSION['Admin'] = 1;
                if ($data['estModerateur'] == 1)
                    $_SESSION['Moderateur'] = 1;
                $_SESSION['numUtilisateur'] = $data['numUtilisateur'];
                $_SESSION['user'] = $data['pseudoUtilisateur'];
                header('Location: routeur.php?action=accueil');
            } else {
                setcookie("mdpDif", $_POST["Pseudo"], time() + 5); // si l'utilisateur se trompe de mot de passe mais que le pseudo est bon on initialise un cookie pour sauvegarder le pseudo
                header('Location: routeur.php?action=seConnecter&login_err=password');
                die();
            }
        } else {
            header('Location: routeur.php?action=seConnecter&login_err=Pseudo');
            die();
        }
    } else {
        header('Location: routeur.php?action=seConnecter&login_err=Pseudo');
        die();
    }
} // si le formulaire est envoyé sans aucune données
```

Une fonctionnalité utile à la connexion est la fonction « Remember Username ». Qui permet à l'utilisateur de sauvegarder son identifiant pour de futures connexions. On a défini cela grâce à des

cookies. Le navigateur garde en mémoire l'identifiant pendant 1 semaine. Voici le code correspondant :

```
setcookie("username", $_POST["Pseudo"], time() + 86400);
```

On a aussi implémenté une page qui permet à l'administrateur du site de définir ses modérateurs.

Vos modérateurs					
Numéro Utilisateur	Nom	Prénom	Pseudo	est Modérateur	Ajouter/Supprimer Modérateur
5	Cocagne	Theo	Comb	<input type="checkbox"/>	Valider
8	ds	dz	ds	<input checked="" type="checkbox"/>	Valider
2	Mont	Henry	hMont	<input checked="" type="checkbox"/>	Valider
4	Regulier	Corentin	Raps	<input type="checkbox"/>	Valider
1	Schuler	Matthieu	lershu	<input checked="" type="checkbox"/>	Valider
6	Tran	Matthieu	Aiern	<input type="checkbox"/>	Valider
3	Valogne	Aurelien	vAurelien	<input type="checkbox"/>	Valider

Cette page affiche tous les utilisateurs qui se sont inscrits sur le site et l'admin, qui a tous les droits peut ajouter un modérateur comme en enlever à sa guise.

Le rôle du modérateur est de pouvoir supprimer n'importe quelles recettes qui juge inapproprié ou bien des commentaires qui n'ont pas lieu d'être dans un blog de cuisine.

Conclusion

Tout d'abord, il faut savoir que ce semestre a été pour nous deux, éprouvant et intense. Ce projet nous a été donné à la veille des révisions pour les partiels et pendant des vacances qui se veulent assez festifs malgré le contexte sanitaire. C'est donc dans un contexte relativement stressant que nous avons abordé ce projet. Cependant, grâce aux nombreux outils que nous avons appris, nous avons su rester serein et nous organiser au mieux pour effectuer ce dernier dans de bonnes conditions. Nous avons fourni des efforts conséquents et sommes fiers de rendre ce projet sur lequel nous avons consacré de nombreuses heures.

En ce qui concerne la partie technique, il faut le dire, nous avons eu beaucoup de soucis mais, par la même occasion, énormément appris. En effet, du bon fonctionnement entre notre base de données et PHP, à la réalisation finale de notre site, nous sommes passés par de nombreux états. Nous avons notamment dû apprendre les bases du JavaScript, utiliser des sessions, des redirections nombreuses, Bootstrap, Git et GitHub. Ce projet a été assez différents des précédents car nous partions d'une base relativement solide et cela nous a donc permis de réaliser des fonctionnalités détaillées. Le fait de pouvoir utiliser nos connaissances d'une façon aussi concrètes nous a paru très satisfaisant.

Annexes

Comment tester notre site (Important)

- Pour avoir accès aux fonctions d'administrateurs :

➔ Connectez vous avec les identifiants suivants :

- Pseudo : admin
- Mot de passe : root

- Si vous commentez une recette avec un mot bannis, vous recevrez un avertissement.
- Au bout de 3 avertissements, votre compte sera banni et supprimé de la base de données. Toutes vos recettes et vos commentaires seront supprimés. Evitez donc de faire cela avec l'admin car vous ne pourrez plus revenir en arrière.
- Effectuez les fonctions citées ci-dessous.
- Pour tester si l'admin donne bien les droits de modérateurs à un utilisateur qu'il aura choisi :
 - ➔ Vous pouvez mettre les droits de modérations à l'utilisateur test en l'ajoutant avec le compte admin ou bien suivez les étapes suivantes
 - ➔ Créez un nouvel utilisateur en vous inscrivant
 - ➔ Déconnectez-vous et connectez-vous avec le pseudo admin
 - ➔ Appuyez sur le bouton « Ajoutez un modérateur »

- Ajoutez l'utilisateur que vous avez créé précédemment en tant que modérateur en cliquant sur le bouton validé à côté de son champ.
- Déconnectez-vous et connectez-vous avec l'utilisateur créé.
- Vous devriez avoir accès aux fonctions de modérateur (Supprimés les recettes et les commentaires)

Les fonctionnalités de notre site

1. Sans être connecté:

- Consulter les recettes des membres
- Rechercher une recette
- S'inscrire ou se connecter

2. Connecté sans être admin ou modérateur

- Consulter les recettes des membres
- Ajouter une Recette
- Consulter ses recettes
- Rechercher une recette
- Ajouter un commentaire
- Supprimer ses commentaires
- Supprimer ses recettes
- Se déconnecter

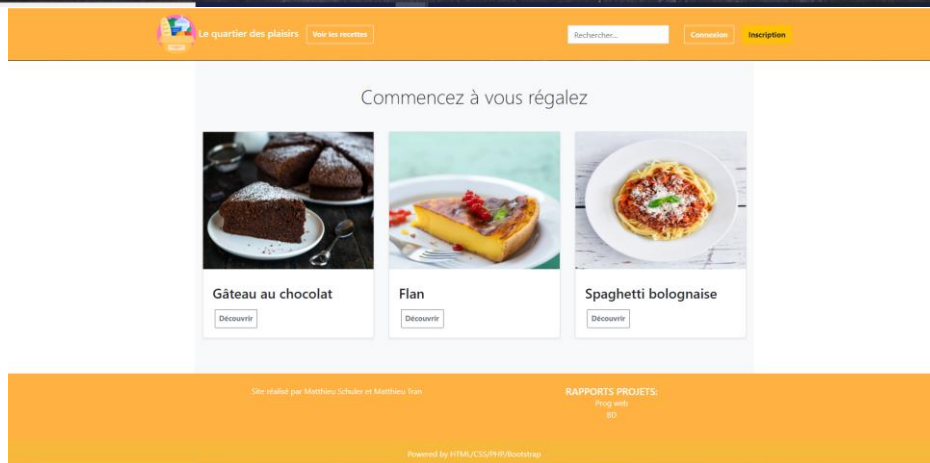
3. Connecté admin

- Consulter les recettes des membres
- Ajouter une Recette
- Consulter ses recettes
- Rechercher une recette
- Ajouter un commentaire
- **Supprimer n'importe quel commentaire**
- **Supprimer n'importe quelles recettes**
- **Ajouter des modérateurs**
- Se déconnecter

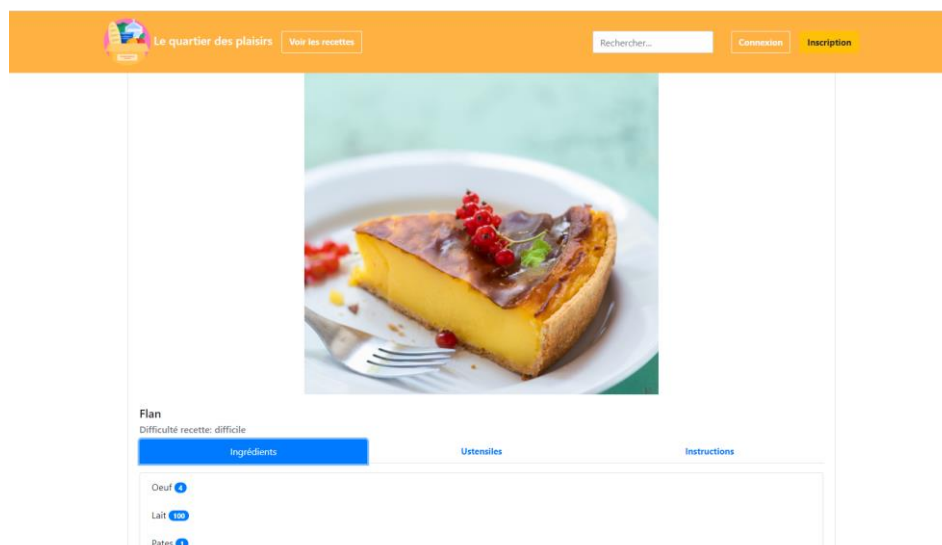
4. Connecté modérateur

- Consulter les recettes des membres
- Ajouter une Recette
- Consulter ses recettes
- Rechercher une recette
- Ajouter un commentaire
- **Supprimer n'importe quel commentaire**
- **Supprimer n'importe quelles recettes**
- Se déconnecter

Page d'accueil (dépend du nombre de recette)



Consulter une recette



Commentaires

1 / 5

1 Avis

5 ★
(0)

4 ★
(0)

3 ★
(0)

2 ★
(0)

1 ★
(1)

Veuillez vous connecter pour mettre
un commentaire

admin

test

★★★★★

2022-01-15 19:12:16

Connexion ou Inscription

Connexion

Pseudo

Mot de passe

☐ Remember username

Connexion

[Inscription](#)

Inscription

Prenom

Nom

Pseudo

Mot de passe


Re-tapez le mot de passe

Inscription

Vous avez déjà un compte ?

[Connectez vous](#)

Ajouter une recette

 Le quartier des plaisirs [Ajouter une recette](#) [Ajouter un modérateur](#) [Mon Compte](#)

Nom de la recette

Catégories

Difficulté

Image


Instructions

Ingrédients

Quantités


Utensiles

Ajouter modérateur

 Le quartier des plaisirs [Ajouter une recette](#) [Ajouter un modérateur](#) [Mon Compte](#)

Vos modérateurs					
Numéro Utilisateur	Nom	Prénom	Pseudo	est Modérateur	Ajouter/Supprimer Modérateur
5	Cocagne	Theo	Comb	<input type="checkbox"/>	<input type="button" value="Valider"/>
2	Mont	Henry	hMont	<input type="checkbox"/>	<input type="button" value="Valider"/>
4	Regulier	Corentin	Raps	<input type="checkbox"/>	<input type="button" value="Valider"/>
1	Schuler	Matthieu	lershu	<input checked="" type="checkbox"/>	<input type="button" value="Valider"/>
6	Tran	Matthieu	Aiern	<input type="checkbox"/>	<input type="button" value="Valider"/>
3	Valogne	Aurelien	vAurelien	<input type="checkbox"/>	<input type="button" value="Valider"/>

Nos recettes (s'il y en a)

 Le quartier des plaisirs [Ajouter une recette](#) [Ajouter un modérateur](#) [Mon Compte](#)

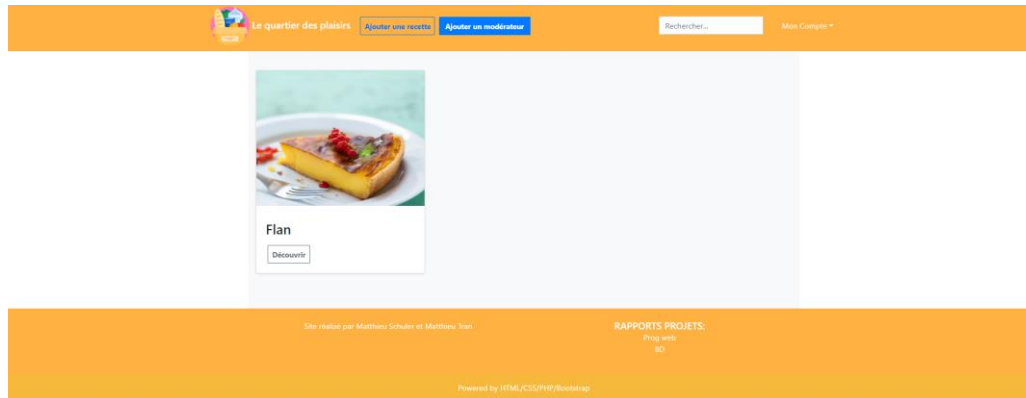
Désolé, vous n'avez pas de recettes pour le moment...

Site réalisé par Matthieu Schuler et Matthieu Tran

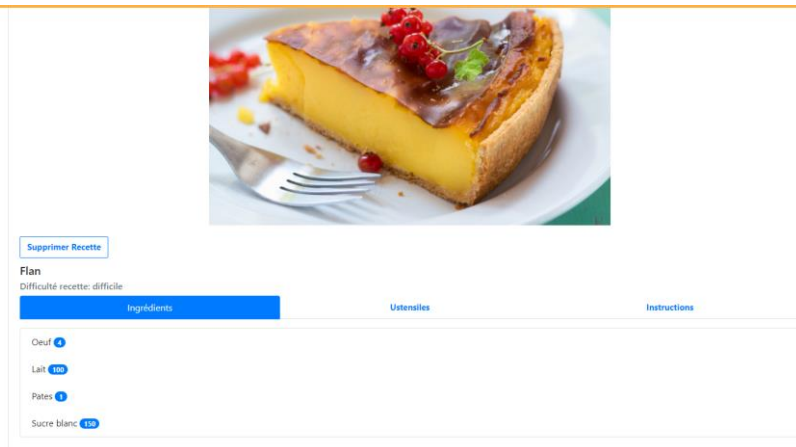
RAPPORTS PROJETS:
Prog web
BD

Powered by HTML/CSS/PHP/Bootstrap

Rechercher recette



Supprimer Recette



Supprimer Commentaire

admin	Supprimer
test	
★★★★★	2022-01-15 19:12:16