

DOSSIER DE PROJET PROFESSIONNEL POUR LE TITRE DE DEVELOPPEUR WEB & WEB MOBILE

Développement du site Web : KABSA FOOD pour Linda Bachia



Présenté par Azzouz Mohamed

Sommaire

Introduction

I.1 – Résumé du dossier de projet.....	3
--	---

Présentation

II.1 – Présentation personnelle.....	3
--------------------------------------	---

II.2 - Présentation de la formation.....	3
--	---

II.3 – Présentation entreprise.....	4
-------------------------------------	---

Projet Pro : Site web KABSA

III.1 – Présentation du site.....	5
-----------------------------------	---

III.2 – Cahier des charges.....	6
---------------------------------	---

III.3 – Développement de la partie Front-End.....	7
---	---

III.4 – Développement de la partie Back-End.....	11
--	----

III.5 – Difficulté rencontré.....	23
-----------------------------------	----

III.6 – Solutions mises en œuvre	23
--	----

III.7 – Conclusion.....	23
-------------------------	----

Annexe	24
--------------	----

I – INTRODUCTION

I.1 – Résumé du dossier de projet

Ce dossier présente la partie que j’ai travaillé sur mon projet professionnel de fin d’année lors de ma formation à La Plateforme_, Coding School.

II – PRESENTATION

II.1 Présentation personnelle

Je m’appelle Azzouz Mohamed, j’ai 21 ans et j’habite à Marseille. J’ai un BAC STI2D ainsi qu’un BTS SN. J’ai intégré la formation de La Plateforme_, Coding School en septembre 2019 à la suite d’une proposition de pôle emploi.

II.2 Présentation de la formation

La Plateforme_ est une école du numérique et des nouvelles technologies co-fondée avec le Club Top 20 réunissant les grandes entreprises de la Métropole Aix Marseille. Elle comprend une offre de formation diversifiées destinées à former des codeurs et développeurs web, des experts en sécurité, des ingénieurs spécialisés en Intelligence Artificielle, et des cadres d’entreprises au travers de cycles de formation continues.

La Plateforme_ est membre du programme Grande Ecole du Numérique. Elle est soutenue par de grandes entreprises du territoire comme le Crédit Agricole Alpes Provence, par la région Sud, le Département des Bouches du Rhône et la Métropole Aix Marseille Provence.

II.3 Présentation entreprise

Kabsa. C'est le nom du futur restaurant de Linda Bachia, une nouvelle adresse healthy à deux pas de la CMA CGM et de la tour La Marseillaise. Le concept proposé est innovant. « *Après avoir constaté qu'il était difficile de manger sainement aujourd'hui à l'extérieur, mais aussi difficile de retrouver une silhouette harmonieuse, je me suis intéressée au bien manger et à la santé de manière globale* », explique la jeune cheffe d'entreprise de 26 ans.

III – Projet Pro: Site web KABSA

III.1 – Résumé du projet

Ce projet consiste à créer un site web pour donner suite au projet de Linda Bachia.

Ce site est un restaurant healthy où une personne après avoir répondu à un petit questionnaire (pop-up), s'être inscrite et connecté peut commander des plats.

Le pop-up qui s'affichera dans la page accueil servira à dire à l'utilisateur combien de kcal il devra consommer en moyenne.

Le site s'adresse à trois types d'utilisateurs différents :

- Un utilisateur anonyme.
- Un utilisateur authentifié.
- Un administrateur

L'utilisateur anonyme peut seulement naviguer sur le site sans pouvoir commander de plats.

L'utilisateur authentifié, après s'être connecté peut choisir ses préférences (protéines et légumes) dans son profil pour avoir un résultat adapté lors de ses recherches.

L'administrateur lui pourra modifier les données du site (ajouter ou supprimer des plats et des catégories, supprimer des utilisateurs etc...).

III.2 – Cahier des charges

L'objectif du projet est qu'après une inscription et une connexion des utilisateurs, une base de données enregistre les inscrit, leurs préférences et affiche les plats en fonctions de cela.

Pour cela les utilisateurs devront compléter 2 pop-ups.

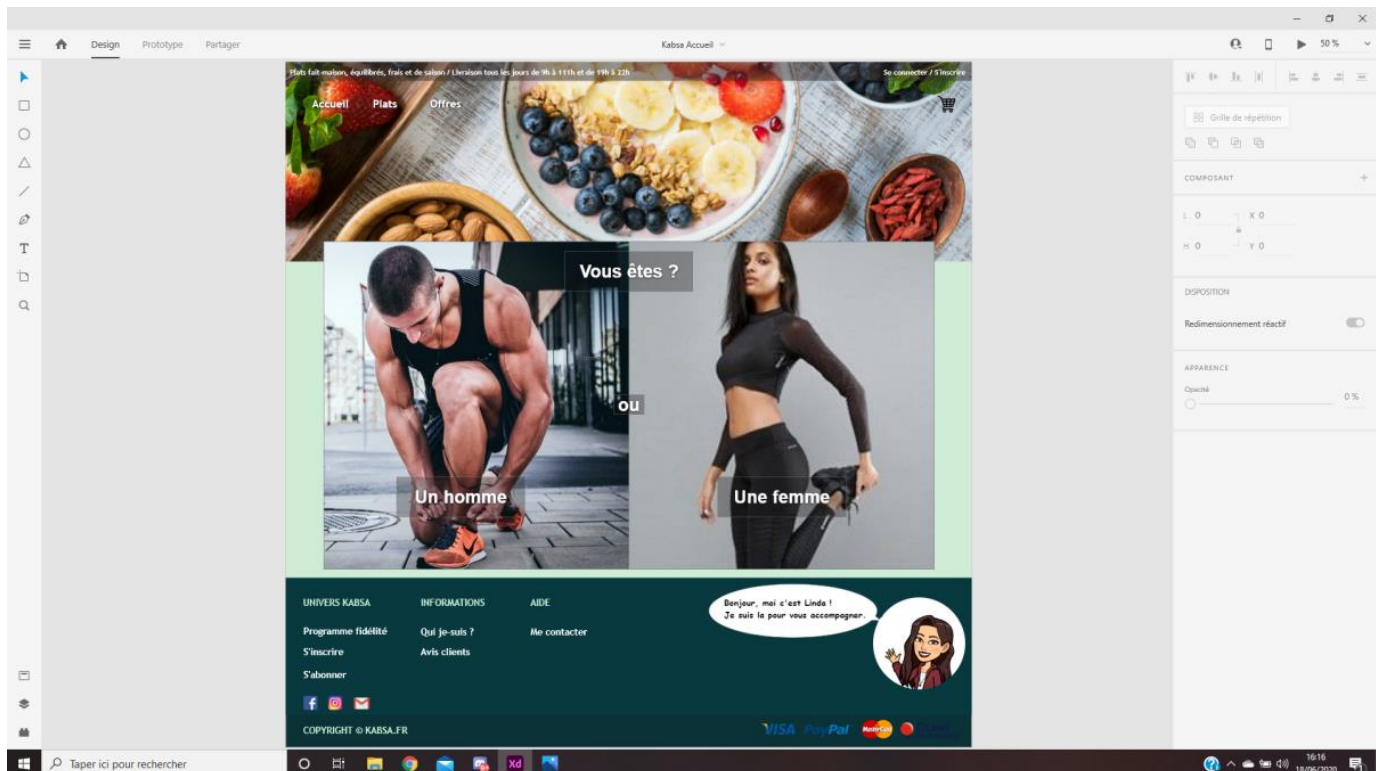
Le premier dans la page d'accueil qui servira a définir combien de kcal l'utilisateur devra consommé en moyenne et le deuxième dans la page profil qui servira à définir ses préférence pour ajuster les filtre lors de la recherche de plats.

Pour la page admin, je devais réaliser une page assez simple a comprendre pour que notre cliente puisse plus tard modifié les catégorie ainsi que les plats facilement.

Pour l'accessibilité, l'application sera développée en responsive pour permettre l'utilisateur d'utilisé le site sur un écran plus petit (tablettes et smartphone).

III.3 – Développement de la partie Front-End

III.3.1 Maquette de la page Accueil



Voici la maquette de l'accueil réalisé avec Adobe XD.

Sur cette maquette nous retrouvons le header contenant la redirection vers les pages plats et offres et dans le footer nous retrouvons la redirection vers les autres pages tel qu'inscription, contact etc...

Par la suite dans le site web nous aurons inscription et connexion dans le header et certaines pages tel que offres, qui suis-je et programme de fidélité disparaîtront.

Lors de la visite sur cette page un pop-up s'affiche et un formulaire apparaît. L'utilisateur devra alors se connecter et remplir ce formulaire

qui servira à lui dire combien de kcal il devra prendre en fonction de ce qu'il a rempli au préalable et il aura du coup des plats adapté pour lui.

Pour les couleurs nous avons choisi d'utiliser principalement du vert et du blanc car cela représente bien le but de ce restaurant.

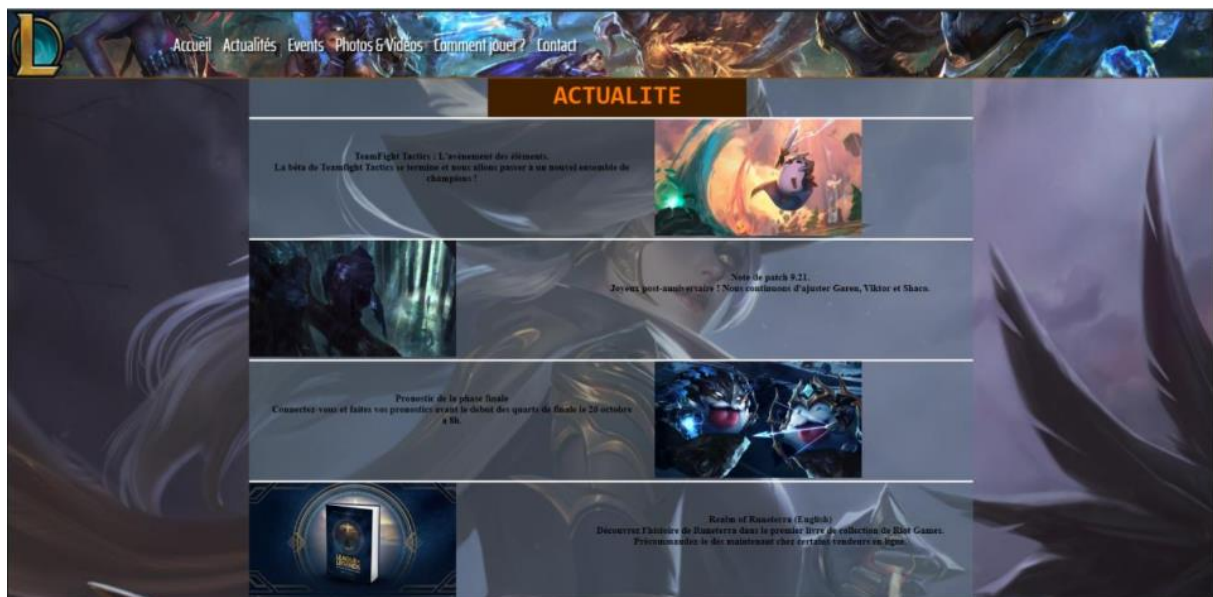
Pour coder les autres pages j'utilise les langages suivants :

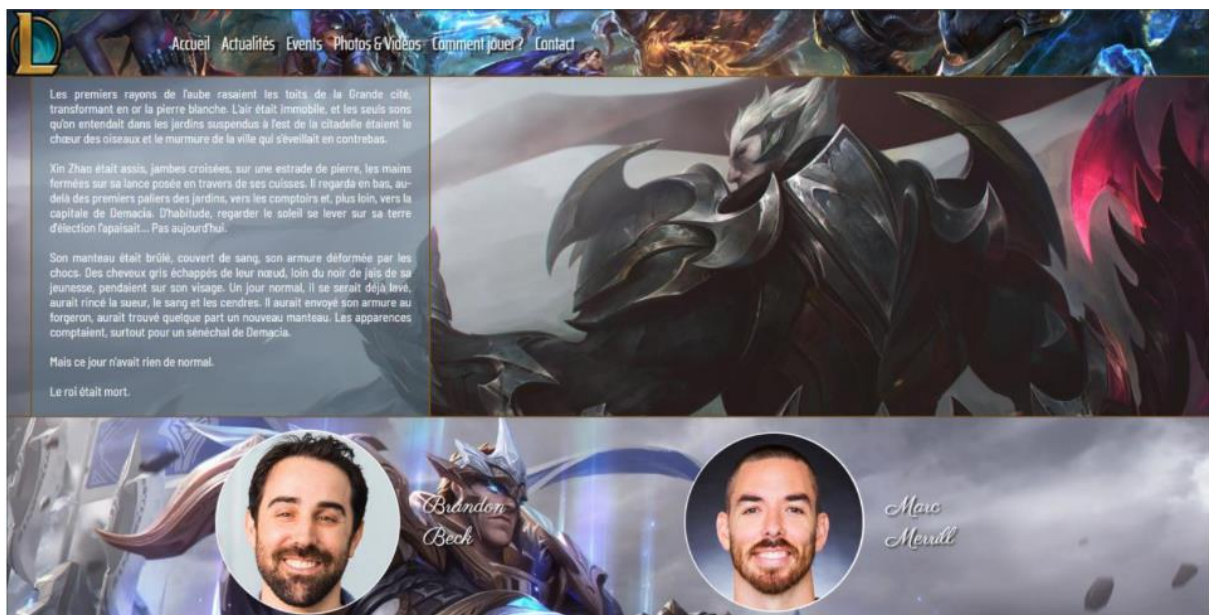
- HTML / CSS pour la structure du site
- Bootstrap pour le CSS de certaine page et le responsive
- Javascript/JQuery pour l'interaction utilisateur.

III.3.2 Réalisation d'une interface utilisateur web statique

Dans ce projet je n'ai pas de page web statique donc j'ai pris comme exemple un projet que j'ai fait lors de ma formation avec un camarade.

Voici les page Actualité et Accueil qui sont entièrement codé en HTML / CSS seulement pour mon projet Fansite consacré à League of Legend.





III.3.3 Réalisation d'une interface web dynamique

Pour illustrer une interface web dynamique je prends comme exemple mon formulaire de connexion que j'ai codé. L'utilisateur peut choisir d'afficher son mot de passe ou de le laisser cacher.

Connection

use your login and password

Identifiant

Mot de passe

AFFICHER

Valider

```

$(document).ready(function(){
    $('.show-password').click(function() {
        if($(this).prev('input').prop('type') == 'password') {
            //Si c'est un input type password
            $(this).prev('input').prop('type', 'text');
            $(this).text('cacher');
        } else {
            //Sinon
            $(this).prev('input').prop('type', 'password');
            $(this).text('afficher');
        }
    });
});

```

Dans la page connexion j'utilise le script ci-dessus qui servira à afficher ou masquer le mot de passe.

Ce script permet lorsque l'on click sur l'objet qui a comme class : show-password cela verifie si l'input qui le précède est de type password, si cette condition est validée alors le script change le type de cet input en type text et affiche comme text : cacher sinon cela change le type en password et affiche comme text : afficher.

III.4 – Développement de la partie Back-End

III.4.1 Arborescence de l'application

Accueil

- ➔ Inscription / Connexion
- ➔ Plats
- ➔ Page Profil
 - Modification des informations (Login, Mot de passe)
 - Ajout / Modification des préférences (Légumes et Protéines)
- ➔ Panier
 - Commande
- ➔ Admin Panel
 - Ajout de catégorie / sous-catégorie
 - Ajouts de nouveau plats
 - Liste de catégorie / sous-catégorie avec possibilité de les modifier ou supprimer
 - Liste des plats avec possibilité de les modifier ou supprimer
 - Liste des utilisateurs

III.4.2 Langage du développement de la partie back-end.

Pour cette partie j'utilise PHP pour communiquer avec la base de données avec les informations récupérer grâce aux requête (POST) faites au serveur.

J'ai choisi de travailler avec PHP car c'est le langage que j'ai le plus abordé au cours de ma formation et que maitrise le mieux.

III.4.3 Création de la base de données

Pour créer la base de données j'ai utilisé PhpMyAdmin car c'est l'outil que j'ai utilisé tout au long de ma formation.

La base de données se nomme « projetpro » et contient les tables suivantes :

- ➔ Utilisateurs
- ➔ Categorie
- ➔ Plats
- ➔ Panier
- ➔ Commentaire
- ➔ Ingredients
- ➔ goututilisateur

Avant de commencer la conception du site il faut créer la connexion à la base de données pour le stockage des données et pour cela j'utilise mon fichier bdd.php qui contient la classe bdd.

Cette classe contient les fonctions suivantes :

- ➔ connect : une fonction qui permet de se connecter à la base de données, elle contient :
 - Le nom de l'host, (si je suis en local : localhost).
 - Le nom de l'utilisateur ('root')
 - Le mot de passe ('').
 - Le nom de la base de données ('projetpro')
- ➔ close : une fonction qui permet de fermer la connexion à la base de données
- ➔ execute : une fonction qui prend en paramètre une requête et qui return le résultat (ex : SELEC * FROM table)
- ➔ executeonly : une fonction qui prend en paramètre une requête et qui return cette même requête (ex :INSERT INTO table VALUES('valeur1',...))
- ➔ executeassoc : une fonction qui prend en paramètre une requête et qui return le résultat sous forme de tableau associatif.

```

K: rnp
class bdd
{
    private $isConnected = false;
    private $connexion = "";

    public function connect()
    {
        $connect = mysqli_connect('localhost', 'root', '', 'projetpro');
        $this->connexion = $connect;
        return $this->connexion ;
    }

    public function close()
    {
        mysqli_close($this->connexion);
        $this->isConnected = false;
    }

    public function execute($query)
    {
        $requete = $query;
        $query = mysqli_query($this->connexion, $requete);
        $resultat = mysqli_fetch_all($query);
        return $resultat;
    }

    public function executeonly($query)
    {
        $requete = $query;
        $query = mysqli_query($this->connexion, $requete);
        return $query;
    }

    public function executeassoc($query)
    {
        $requete = $query; $query = mysqli_query($this->connexion, $requete);
        $resultat = mysqli_fetch_all($query, MYSQLI_ASSOC);
        return $resultat;
    }
}
?>

```

III.4.4 Fonctionnalité

III.4.4.1 Page Administrateur

Ajouter Catégorie / Ingredient

Catégorie :

Ajouter

Veuillez entrer un nom

Ingredient :

Ajouter

Liste Catégorie & Ingredients

Catégorie :
CAT 1
Modifier Supprimer

Ingredient :
Boeuf
Modifier Supprimer

Catégorie :

Ajouter Plat

Nom :



Veuillez renseigner ce champ.

Prix :

IMG 1 :
Choisir un fichier Aucun fichier choisi

IMG 2 :
Choisir un fichier Aucun fichier choisi

Catégorie :
CAT 1
Ajouter

LISTE DES PLATS							
Nom	Description	Prix	Catégorie	IMG 1	IMG 2		
Salade	Salade	10	CAT 1			Modifier	Supprimer

Seul l'administrateur après s'être connecté pourra avoir accès à cette page. Sur cette page il pourra ajouter de nouvelles catégories et ingrédients mais aussi de nouveaux plats.

Ces nouveaux ajouts seront affichés en dessous de leur formulaire spécifique.

Pour faire ces nouveaux ajouts j'utilise ma classe produit qui se trouve dans mon dossier class->produits.php.

Cette class contient toutes les fonctions qui permettent d'ajouter, de supprimer ou de modifier des produits.

<?php

```
class produits extends bdd
{
    private $id = NULL ;
    private $nom = NULL ;
    private $description = NULL ;
    private $prix = NULL ;
    private $id_categorie = NULL ;
    private $img1 = NULL ;
    private $img2 = NULL ;

    public function addCategorie($nom, $bdd)
    {
        if (strlen($nom) != 0)
        {
            $categorie = $bdd->execute("SELECT nom FROM categorie WHERE nom = '$nom'");

            if (empty($categorie))
            {
                $newCategorie = $bdd->executeonly("INSERT INTO categorie (nom) VALUES ('$nom')");
                return "newCategorie";
            }
            else
            {
                return "categorieExistante";
            }
        }
        else
        {
            return "info";
        }
    }
}
```

```
public function addIngredient($nom, $bdd)
{
    if (strlen($nom) != 0)
    {
        $ingredient = $bdd->execute("SELECT nom FROM ingredients WHERE nom = '$nom'");

        if (empty($ingredient))
        {
            $newIngredient = $bdd->executeonly("INSERT INTO ingredients (nom) VALUES ('$nom')");
            return "newIngredient";
        }
        else
        {
            return "ingredientExistant";
        }
    }
    else
    {
        return "info";
    }
}

public function addProduit($nom, $description, $prix, $id_categorie, $img1, $img2, $bdd)
{
    if (strlen($nom) != 0 && strlen($description) != 0 && !empty($prix) && !empty($img1) && !empty($img2))
    {
        $produit = $bdd->execute("SELECT nom FROM plats WHERE nom = '$nom'");

        if (empty($produit))
        {
            $newProduit = $bdd->executeonly("INSERT INTO plats (nom, description, prix, id_categorie, img1, img2) VALUES ('$nom', '$description', '$prix', '$id_categorie', '$img1', '$img2')");
            return "newProduit";
        }
        else
        {
            return "produitExist";
        }
    }
    else
    {
        return "info";
    }
}
```

```

public function updateProduits($updateNom, $updateDescription, $updatePrix, $updateCategorie, $updateImg1, $updateImg2, $id, $bdd)
{
    if (!empty($updateNom))
    {
        $update = $bdd->executeonly("UPDATE plats SET nom = '$updateNom' WHERE id = '$id' ");
        header('location:admin.php');
    }

    if (!empty($updateDescription))
    {
        $update = $bdd->executeonly("UPDATE plats SET description = '$updateDescription' WHERE id = '$id' ");
        header('location:admin.php');
    }

    if (!empty($updatePrix))
    {
        $update = $bdd->executeonly("UPDATE plats SET prix = '$updatePrix' WHERE id = '$id' ");
        header('location:admin.php');
    }

    if (!empty($updateCategorie))
    {
        $update = $bdd->executeonly("UPDATE plats SET id_categorie = '$updateCategorie' WHERE id = '$id' ");
        header('location:admin.php');
    }
}

```

```

    if (!empty($updateImg1))
    {
        $img = $bdd->execute("SELECT img1 FROM plats WHERE id = '$id' ");
        var_dump($img);
        $filePath = 'photoProduit/'.$img[0][0];
        if (file_exists($filePath))
        {
            unlink($filePath);
            $update = $bdd->executeonly("UPDATE plats set img1 = '$updateImg1' WHERE id = '$id' ");
            echo "UPDATE plats set img1 = '$updateImg1' WHERE id = '$id' ";
            header('location:admin.php');
        }
        else
        {
            echo "File does not exists";
        }
    }

    if (!empty($updateImg2))
    {
        $img = $bdd->execute("SELECT img2 FROM plats WHERE id = '$id' ");
        var_dump($img);
        $filePath = 'photoProduit/'.$img[0][0];
        if (file_exists($filePath))
        {
            unlink($filePath);
            $update = $bdd->executeonly("UPDATE plats set img2 = '$updateImg2' WHERE id = '$id' ");
            echo "UPDATE plats set img2 = '$updateImg2' WHERE id = '$id' ";
            header('location:admin.php');
        }
        else
        {
            echo "File does not exists";
        }
    }
}
}

```


Cette class contient les fonctions suivantes avec comme parametre les données dont on a besoin plus la requête (\$bdd) :

-addCategorie : qui servira à ajouter une nouvelle catégorie si celle si n'existe pas déjà. En fonction de ce que return la fonction j'affiche un message, si la fonction return :

-newCategorie : alors j'affiche « categorie ajouter »

-categorieExistante : alors j'affiche « Cette Categorie existe déjà »

-info : alors j'affiche « Veuillez entrer un nom »

-addIngredient : qui servira à ajouter de nouveau ingrédients pour constituer les filtres dans la page plats. Cette fonction return les mêmes choses que la fonction précédente seuls les noms change.

-newIngredient, ingredientExistant, info.

-addProduit : qui servira à ajouter un nouveau produit. Elle return aussi les mêmes choses que les fonctions précédentes.

-newProduit, produitExist, info.

-updateProduits : qui servira à modifier les valeurs du produits (nom, description, prix, catégorie, image) une fois que l'administrateur aura appuyer sur « Modifier », lors de cette action un formulaire similaire a celui d'ajout de produit s'ouvrira. Dans cette fonction lorsque l'administrateur voudra modifier les images du produits, la fonction supprimera la ou les photos qui ont été modifier grâce à la fonction « unlink » de php. J'utilise cette fonction pour éviter d'avoir un dossier d'image rempli de photo inutilisé.

III.4.4.2 Fonctionnement de la page

Je créer mes formulaires avec la méthode POST pour pouvoir récupérer les données entrées, ensuite après avoir appuyer sur les inputs de type submit qui serviront à envoyer les données j'utilise ma class produits où il a mes fonctions me permettant d'ajouter les infos entrées par l'administrateur dans la base de données grâce aux requête INSERT INTO.

```
if (strlen($nom) != 0)
{
    $categorie = $bdd->execute("SELECT nom FROM categorie WHERE nom = '$nom'");

    if (empty($categorie))
    {
        $newCategorie = $bdd->executeonly("INSERT INTO categorie (nom) VALUES ('$nom')");
        return "newCategorie";
    }
}
```

L'administrateur pourra aussi les modifier ou les supprimer grâce aux requête UPDATE et DELETE

```
$idCat = $categorie[$i][0];
$nomCat = $categorie[$i][1];

if (isset($_POST["updateCategorie$idCat"]) AND strlen($_POST['upCat']) != 0)
{
    $nameCategorie = $bdd->execute("SELECT nom FROM categorie WHERE nom = '".$_POST['upCat']."' ");
    if (empty($nameCategorie))
    {
        $requeteUpdateCat = $bdd->executeonly("UPDATE categorie set nom = '".$_POST['upCat']."' WHERE nom = '$nomCat.' ");
        header('location:admin.php');
    }
    else
    {
        echo "Ce nom existe";
    }
}

if (isset($_POST["deleteCategorie$idCat"]))
{
    $requeteDeleteCat = $bdd->executeonly("DELETE FROM categorie WHERE id = '$idCat.'");
    header('location:admin.php');
}
```

Lorsque l'administrateur voudra modifier quelque chose je vérifie au préalable si le nom n'existe pas déjà dans la base de données

J'utilise \$idCat pour définir quelle catégorie sera modifier ou supprimer. Je récupère cette variable grâce à la requête suivantes :

```
$categorie = $bdd->execute("SELECT * FROM categorie");
```

III.4.4.3 Sécurité des pages

Pour la sécurité de cette page étant donné que je ne code pas en PDO, j'utilise la fonction « required » de php dans mon formulaire pour obliger le remplissage du formulaire complet (exemple formulaire Ajout Plat) et dans ma fonction pour ajouter les catégories ainsi que les ingrédients je vérifie bien que l'input n'est pas vide en utilisant la fonction « empty » et affiche un message d'erreur si la case est vide (exemple formulaire ajout categorie).

Pour sécuriser l'ajout de photo je restreins l'ajout a seulement certain type de fichier et à une taille limite.

```
$tailleMax = 2097152 ;  
$extensionsValides = $arrayName = array('jpg', 'jpeg', 'png');
```

Pour sécuriser le mot de passe d'un utilisateur lors de son inscription, je hash son mot de passe grâce à la fonction « password_hash » de php.

```
$password = password_hash($password, PASSWORD_BCRYPT, array('cost' => 12));
```

Et lors de la connexion, pour vérifier qu'il a bien rentré le bon mot de passe, je recupere les donnee dans ma table utilisateurs en fonction de son identifiant et j'utilise la fonction « password_verify » de php pour comparer le mot de passe entré dans l'input et celui de la table utilisateurs.

```
if(password_verify($password,$log[0][2]))
```

III.4.4.4 Jeu d'essai

Pour illustrer un jeu d'essai J'ai choisi de prendre comme exemple mon formulaire d'ajout de produit.

| Scenario | Données | Résultat attendu | Résultat obtenu |
|--|---|--|--|
| Scenario de base | Nom : Salade
Description : Salade
Prix : 10
IMG 1 : img1.jpg
IMG 2 : img2.jpg
Categorie : Cat1 | Message :
PRODUIT
AJOUTER

Ajout des données dans la table Produits | Message :
PRODUIT
AJOUTER

Ajout des données dans la table Produits |
| Erreur 1 :
Nom existant | Nom : Salade
Description : Salade
Prix : 10
IMG 1 : img1.jpg
IMG 2 : img2.jpg
Categorie : Cat1 | Message :
Ce nom existe

Rien ne se passe au niveau de la bdd | Message :
Ce nom existe

Rien ne se passe au niveau de la bdd |
| Erreur 2 :
Un ou plusieurs champs sont manquant | Nom : Salade
Description :
Prix :
IMG 1 : img1.jpg
IMG 2 : img2.jpg
Categorie : Cat1 | Une notification apparait sur les champs manquants :
« Veuillez renseigner ce champ » | Une notification apparait sur les champs manquants :
« Veuillez renseigner ce champ » |
| Erreur 3 :
L'importation d'un/des photos n'a pas fonctionné | Nom : Salade
Description : Salade
Prix : 10
IMG 1 : img1.jpg
IMG 2 : img2.jpg
Categorie : Cat1 | Message :
Erreur durant l'importation de votre photo | Message :
Erreur durant l'importation de votre photo |
| | | | |

| | | | |
|--|---|---|---|
| Erreur 4 :
Mauvais format
d'une/des photos | Nom : Salade
Description : Salade
Prix : 10
IMG 1 : img1.gif
IMG 2 : img2.jpg
Categorie : Cat1 | Message :
Votre photo
doit être au
format jpg,
jpeg ou png. | Message :
Votre photo
doit être au
format jpg,
jpeg ou png. |
| Erreur 5 :
Taille de photo
trop grande | Nom : Salade
Description : Salade
Prix : 10
IMG 1 : img1.jpg
IMG 2 : img2.jpg
Categorie : Cat1 | Message :
Votre photo
de profil ne
doit pas
dépasser 2Mo | Message :
Votre photo
de profil ne
doit pas
dépasser 2Mo |

III.4.4.5 Ajout d'un modal Bootstrap

Pour ma page profil je devais coder un pop-up qui servirait à définir les goûts des utilisateurs pour après leurs proposer des plats adaptés. Pour ce faire je me suis aidé de la documentation sur le site Bootstrap.

How it works

Before getting started with Bootstrap's modal component, be sure to read the following as our menu options have recently changed.

- Modals are built with HTML, CSS, and JavaScript. They're positioned over everything else in the document and remove scroll from the `<body>` so that modal content scrolls instead.
- Clicking on the modal "backdrop" will automatically close the modal.
- Bootstrap only supports one modal window at a time. Nested modals aren't supported as we believe them to be poor user experiences.
- Modals use `position: fixed`, which can sometimes be a bit particular about its rendering. Whenever possible, place your modal HTML in a top-level position to avoid potential interference from other elements. You'll likely run into issues when nesting a `.modal` within another fixed element.
- Once again, due to `position: fixed`, there are some caveats with using modals on mobile devices. [See our browser support docs](#) for details.
- Due to how HTML5 defines its semantics, [the autofocus HTML attribute](#) has no effect in Bootstrap modals. To achieve the same effect, use some custom JavaScript:

```
$('#myModal').on('shown.bs.modal', function () {
  $('#myInput').trigger('focus')
})
```

Copy

Keep reading for demos and usage guidelines.

Traduction :

Comment ça marche

Avant de commencer avec le composant Modal de Bootstrap, veuillez lire ce qui suit étant donné que les options de notre menu ont changé.

- Les modals sont construits avec HTML, CSS et Javascript. Ils sont positionnés par-dessus tout autre chose dans le document et suppriment le défilement du <body> pour que ce soit le contenu du modal que l'on puisse faire défiler à la place.
- Cliquer à l'extérieur du modal le fermera automatiquement.
- Bootstrap supporte seulement une fenêtre modal à la fois. Les modals imbriqués ne sont pas supportés car nous pensons qu'ils sont d'une faible expérience pour les utilisateurs.
- Les modals utilisent « position : fixed », qui peut parfois être un peu particulier lors de son rendu. À chaque fois que c'est possible, placez votre modal HTML en hauteur pour éviter de potentielles interférences avec d'autres éléments. Vous rencontrerez des erreurs quand vous imbriquez un .modal sans un autre élément fixé.
- Encore une fois, à cause de « position : fixed », il y a quelques avertissements pour l'utilisation des modals sur les appareils mobiles. Consultez notre documentation pour plus de détails.
- En raison de la façon qu'HTML5 définit sa sémantique, l'attribut HTML autofocus n'a pas d'effet sur les modals de Bootstrap. Pour avoir le même effet, utilisez un peu de javascript personnalisé.

III.5 Difficulté rencontré

En raison du confinement et du manque d'organisation, j'ai pris du retard sur le projet car je travaillé beaucoup moins souvent chez moi aux vues des tentation qu'il y avait (jeux-vidéos).

J'ai aussi perdu la motivation de travailler lorsque j'ai appris que j'étais maintenant seul sur le projet car l'un de mes camarades avait trouvé un CDI et a arrêté la formation tandis que l'autre voulait redoubler.

III.6 Solution mise en œuvre

Je me suis donc concentré sur des fonction basique et simple à faire et tout en essayant d'améliorer ma façon de coder, j'essaye chaque jour d'avancer dans le projet en ayant pour objectif de le finir.

III.7 Conclusion

J'ai bien aimé travailler sur ce projet malgré les difficultés rencontré, le manque d'organisation et de temps. Ce projet ma permis d'avoir une nouvelle façon de coder et j'ai appris de nouvelle chose comme les modals de bootstrap ou bien le Javascript.

ANNEXE

Base de données

| Table | Action | Lignes | Type | Interclassement | Taille | Perte |
|---|--------------|-----------|---------------|--------------------------|-----------------|--------------|
| <input type="checkbox"/> categorie | ★ | 4 | MyISAM | latin1_swedish_ci | 2,1 kio | 44 o |
| <input type="checkbox"/> commentaire | ★ | 0 | MyISAM | latin1_swedish_ci | 1 kio | - |
| <input type="checkbox"/> goututilisateur | ★ | 3 | MyISAM | latin1_swedish_ci | 2,2 kio | - |
| <input type="checkbox"/> ingredients | ★ | 1 | MyISAM | latin1_swedish_ci | 2 kio | - |
| <input type="checkbox"/> panier | ★ | 2 | MyISAM | latin1_swedish_ci | 2,1 kio | 21 o |
| <input type="checkbox"/> plats | ★ | 1 | MyISAM | latin1_swedish_ci | 2,2 kio | 116 o |
| <input type="checkbox"/> utilisateurs | ★ | 7 | MyISAM | latin1_swedish_ci | 2,5 kio | - |
| 7 tables | Somme | 18 | MyISAM | latin1_swedish_ci | 14,1 kio | 181 o |

CLASSE

Class user.php

```
<?php
class user extends bdd
{
    private $id = NULL;
    private $login = NULL;
    private $password = NULL;

    public function inscription($login, $password, $confPassword, $adresse, $bdd)
    {
        if (strlen($login) != 0 && strlen($password) != 0)
        {
            if ($password == $confPassword )
            {
                $user = $bdd->execute("SELECT login FROM utilisateurs WHERE login = '$login'");

                if (empty($user))
                {
                    $password = password_hash($password, PASSWORD_BCRYPT, array('cost' => 12));
                    $newUser = $bdd->executeonly("INSERT INTO utilisateurs (login, password, adresse) VALUES ('$login', '$password', '$adresse')");
                    return "userCheck";
                }
                else
                {
                    return "userExist";
                }
            }
            return "mdpFaux";
        }
        return "logVide";
    }
}
```

```

public function connexion($login,$password, $bdd)
{

    $log = $bdd->execute("SELECT * FROM utilisateurs WHERE login = '$login'");

    if(empty($log))
    {
        if($login == $log[0][1])
        {
            if(password_verify($password,$log[0][2]))
            {
                $_SESSION['id'] = $log[0][0];
                $_SESSION['login'] = $log[0][1] ;

                header('Location:index.php');
            }
            else
            {
                echo "ERREUR";
            }
        }
        else
        {
            echo "ERREUR";
        }
    }
    else
    {
        return false;
    }
}

```

```

public function addPreferenceGout($idUser, $boeuf, $poulet, $dinde, $saumon, $thon, $calamar, $haricots, $pommeDeTerre, $brocolis, $avocats, $choux,
    $salade, $poivrons, $champignons, $lentilles, $bdd)
{
    $gout = $bdd->executeonly("INSERT INTO goututilisateur (id.utilisateur, boeuf, poulet, dinde, saumon, thon, calamar, haricots, pommeDeTerre,
        brocolis, avocats, choux, salade, poivrons, champignons, lentilles) VALUES ('$idUser', '$boeuf', '$poulet', '$dinde', '$saumon', '$thon', '$
        calamar', '$haricots', '$pommeDeTerre', '$brocolis', '$avocats', '$choux', '$salade', '$poivrons', '$champignons', '$lentilles')");
    echo "INSERT INTO goututilisateur (id.utilisateur, boeuf, poulet, dinde, saumon, thon, calamar, haricots, pommeDeTerre, brocolis, avocats,
        choux, salade, poivrons, champignons, lentilles) VALUES ('$idUser', '$boeuf', '$poulet', '$dinde', '$saumon', '$thon', '$calamar', '$haricots
        ', '$pommeDeTerre', '$brocolis', '$avocats', '$choux', '$salade', '$poivrons', '$champignons', '$lentilles')";
}

public function disconnect()
{
    session_unset();
    session_destroy();
    header('location:index.php');
}
}

```

Page Inscription

```
<?php
    session_start();

    require 'class/bdd.php';
    require 'class/user.php';

    $user = new user();
    $bdd = new bdd();

    $bdd->connect();

    if (isset($_SESSION['id']))
    {
        header('Location:index.php');
    }
}

<!DOCTYPE html>
<html>
<head>
    <title>Inscription</title>
    <link rel="stylesheet" type="text/css" href="css/style.css">
</head>
<body id="bodyInscription">
    <main id="mainInscription">
        <section id="inscription">
            <h1 id="titleInscription">INSCRIPTION <span>Bienvenue</span></h1>
            <form method="post" action="" id="formInscription">
                <label for="login" class="labelInscription">Login :</label><input type="text" name="login" class="inputInscription" required><br />
                <label for="mdp" class="labelInscription">Mot de passe :</label> <input type="password" name="password" class="inputInscription" required><br />
                <label for="confMdp" class="labelInscription">Confirmer votre Mot de Passe :</label> <input type="password" name="confPassword" class="inputInscription" required><br />
                <label for="adresse" class="labelInscription">Adresse :</label> <input type="text" name="adresse" class="inputInscription"><br />
                <input type="submit" value="Inscription" name="newUser">
            </form>
        </section>
    </main>
</body>
</html>
```

```
<?php

if (isset($_POST["newUser"]))
{
    if ($user->inscription($_POST['login'], $_POST['password'], $_POST['confPassword'], $_POST['adresse'], $bdd) == "userCheck")
    {
        echo "Compte créer";
        header('Location:connexion.php');
    }
    elseif ($user->inscription($_POST['login'], $_POST['password'], $_POST['confPassword'], $_POST['adresse'], $bdd) == "userExist")
    {
        echo "ce login existe";
    }
    elseif ($user->inscription($_POST['login'], $_POST['password'], $_POST['confPassword'], $_POST['adresse'], $bdd) == "mdpFaux")
    {
        echo "mot de passe !=";
    }
    elseif ($user->inscription($_POST['login'], $_POST['password'], $_POST['confPassword'], $_POST['adresse'], $bdd) == "logVide")
    {
        echo "champ manquant";
    }
}
}

</section>
</main>
</body>
</html>
```

Page connexion

```
<?php
    session_start();
    require 'class/bdd.php';
    require 'class/user.php';

    $user = new user();
    $bdd = new bdd();

    $bdd->connect();

    if (isset($_SESSION['id']))
    {
        header('Location:index.php');
    }
?>

<!DOCTYPE html>
<html>
<head>
    <title>Connexion</title>
    <link rel="stylesheet" type="text/css" href="css/style.css">
</head>
<body id="bodyConnexion">
    <main id="mainConnexion">
        <section id="connexion">
            <h1 id="titleConnexion">Connexion <span> use your login and password</span></h1>

            <form action="" method="post" id="formConnexion">

                <label for="login" class="labelConnexion">Identifiant</label>
                <input type="text" name="login" class="inputConnexion" required<br>

                <label for="password" class="labelConnexion">Mot de passe</label>
                <input type="password" name="password" class="inputConnexion" id="password">
                <span class="show-password">afficher</span>

                <input type="submit" name="send" class="inputConnexion">
            </form>
        </section>
    </main>
</body>
</html>
```

```
<?php
    if(isset($_POST["send"]))
    {
        $user->connexion($_POST["login"],$_POST["password"], $bdd);
    }
?>
</section>
</main>
<script type="text/javascript" src="js/jquery-3.5.1.min"></script>
<script type="text/javascript" src="js/buttonPassword.js"></script>
</body>
</html>
```

Page Profil

```
1 <?php
2
3 session_start();
4 require 'class/bdd.php';
5 require 'class/user.php';
6
7 $user = new user();
8 $bdd = new bdd();
9
10 $bdd->connect();
11
12 $userInfo = $bdd->execute("SELECT * FROM utilisateurs WHERE id = '". $_SESSION['id']."' ");
13
14
15 var_dump($userInfo);
16
17 ?>
18
19 <!DOCTYPE html>
20 <html>
21 <head>
22 <title>Profil</title>
23
24 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="
sha384-Gn5384xqQ1aoWA+9588RXPxPg6fy4IWvTNh0E263XmFc7lSxwI6GfAW/dA1S6JXm" crossorigin="anonymous">
25
26 <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KcKbR/rE9/Qp6GaAZGjwFDMVNA/GpGFF93hXp65KkN" crossorigin="anonymous"
></script>
27 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-AplNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q
" crossorigin="anonymous"></script>
28 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR65Pejh4U02d8j0t6vLEHfe/JQGiRRSQQxSfFFWp11MquVdAyjUar5+76PVCmY1"
crossorigin="anonymous"></script>
29
30 <link rel="stylesheet" type="text/css" href="css/style.css">
31
32
33 </head>
34 <body>
```

```
<main>
<div id="profil">
  <form method="post" action="">
    <div>
      <label>login</label> : <input type="text" name="updateLogin" placeholder="<?php echo $userInfo[0][1]; ?>"><br />
      <label>Mot de passe</label> : <input type="password" name="updatePassword"><br />
      <label>Confirmation mot de passe</label> : <input type="password" name="updateConfPassword">
      <br />
      <input type="submit" name="updateInfo" value="Modifier">
    </div>
  </div>
  <div>
    <?php
    if (isset($_POST['updateInfo']))
    {
      $loginUpdate = $_POST['updateLogin'];
      $passwordUpdate = $_POST['updatePassword'];
      $updateConfPassword = $_POST['updateConfPassword'];

      $login = $bdd->execute("SELECT login FROM utilisateurs WHERE login = '$loginUpdate'");

      $id = $_SESSION['id'];

      if (!empty($loginUpdate) && !empty($login))
      {
        echo "Ce Login est déjà prit";
        header("location:profil.php");
      }
      elseif (empty($login) && !empty($loginUpdate))
      {
        $updateLogin = $bdd->executeonly("UPDATE utilisateurs SET login = '$loginUpdate' WHERE id = '$id'");
        header("location:profil.php");
      }
    }
  </div>
</div>
```



```

        <div class="choice" id="avocat">
            <input id="avocat" type="checkbox" name="avocat" value="avocat" />
            <label for="avocat">Avocat</label>
        </div>

        <div class="choice" id="choux">
            <input id="choux" type="checkbox" name="choux" value="choux" />
            <label for="choux">Choux</label>
        </div>

        <div class="choice" id="salade">
            <input id="salade" type="checkbox" name="salade" value="salade" />
            <label for="salade">Salade</label>
        </div>

        <div class="choice" id="poivrons">
            <input id="poivrons" type="checkbox" name="poivrons" value="poivrons" />
            <label for="poivrons">Poivrons</label>
        </div>

        <div class="choice" id="champignons">
            <input id="champignons" type="checkbox" name="champignons" value="champignons" />
            <label for="champignons">Champignon</label>
        </div>

        <div class="choice" id="lentilles">
            <input id="lentilles" type="checkbox" name="lentilles" value="lentilles" />
            <label for="lentilles">Lentilles</label>
        </div>
    </section>
</form>
</section>

<input type="submit" name="preference" value="Enregistrer">

```

```

<?php
if (isset($_POST["preference"]))
{
    if (isset($_POST["boeuf"]))
    {
        $boeuf = $_POST["boeuf"] ;
        $boeuf = 1;
    }
    else
    {
        $boeuf = 0 ;
    }

    if (isset($_POST["poulet"]))
    {
        $poulet = $_POST["poulet"] ;
        $poulet = 1;
    }
    else
    {
        $poulet = 0 ;
    }

    if (isset($_POST["dinde"]))
    {
        $dinde = $_POST["dinde"] ;
        $dinde = 1;
    }
    else
    {
        $dinde = 0 ;
    }

    if (isset($_POST["saumon"]))
    {
        $saumon = $_POST["saumon"] ;
        $saumon = 1;
    }
    else
    {
        $saumon = 0 ;
    }
}

```

```

if (isset($_POST["thon"]))
{
    $thon = $_POST["thon"] ;
    $thon = 1;
}
else
{
    $thon = 0 ;
}

if (isset($_POST["calamar"]))
{
    $calamar = $_POST["calamar"] ;
    $calamar = 1;
}
else
{
    $calamar = 0 ;
}

if (isset($_POST["haricots"]))
{
    $haricots = $_POST["haricots"] ;
    $haricots = 1;
}
else
{
    $haricots = 0 ;
}

if (isset($_POST["pommeDeTerre"]))
{
    $pommeDeTerre = $_POST["pommeDeTerre"] ;
    $pommeDeTerre = 1;
}
else
{
    $pommeDeTerre = 0 ;
}

```

```

if (isset($_POST["brocolis"]))
{
    $brocolis = $_POST["brocolis"] ;
    $brocolis = 1;
}
else
{
    $brocolis = 0 ;
}

if (isset($_POST["avocat"]))
{
    $avocat = $_POST["avocat"] ;
    $avocat = 1;
}
else
{
    $avocat = 0 ;
}

if (isset($_POST["choux"]))
{
    $choux = $_POST["choux"] ;
    $choux = 1;
}
else
{
    $choux = 0 ;
}

if (isset($_POST["salade"]))
{
    $salade = $_POST["salade"] ;
    $salade = 1;
}
else
{
    $salade = 0 ;
}

```



```

    if (isset($_POST["poivrons"]))
    {
        $poivrons = $_POST["poivrons"] ;
        $poivrons = 1;
    }
    else
    {
        $poivrons = 0 ;
    }

    if (isset($_POST["champignons"]))
    {
        $champignons = $_POST["champignons"] ;
        $champignons = 1;
    }
    else
    {
        $champignons = 0 ;
    }

    if (isset($_POST["lentilles"]))
    {
        $lentilles = $_POST["lentilles"] ;
        $lentilles = 1;
    }
    else
    {
        $lentilles = 0 ;
    }

    $user->addPreferenceGout($userInfo[0][0], $boeuf, $poulet, $dinde, $saumon, $thon, $calamar, $haricots, $pommeDeTerre, $brocolis, $avocat, $choux, $salade, $
    poivrons, $champignons, $lentilles, $bdd);

}
?>

</main>
<script type="text/javascript" src="js/jquery-3.5.1.min.js"></script>
<script type="text/javascript" src="js/profil.js"></script>

</body>
</html>

```

Page admin (version pc et mobile)

Version PC sans CSS

Ajouter Category / Ingredient

Categorie :

Ajouter

Ingredient :

Ajouter

Liste Category & Ingredients

Categorie :
CAT 1

Modifier Supprimer

Ingredient :
Boeuf

Modifier Supprimer

Categorie :

Ajouter

Ajouter Plat

Nom :
Salade

Description :
salade

Prix :



IMG 1 :
Choisir un fichier true-dama....1203.jpg

IMG 2 :
Choisir un fichier Aucun fichier choisi

Categorie :
CAT 1

Ajouter

Liste DES PLATS

Nom	Description	Prix	Category	IMG 1	IMG 2		
Salade	Salade	10	CAT 1			Modifier	Supprimer

Liste User

Version Mobile

Ajouter Catégorie / Ingredient

Catégorie :

Ingredient :

Liste Catégorie & Ingredients

Catégorie :

Ingredient :

Catégorie :

Ajouter Plat

Nom :

Description :

Prix :

IMG 1 :
 true-dama....1203.jpg

IMG 2 :
 Aucun fichier choisi

Catégorie :

LISTE DES PLATS	
Nom	Salade
Description	Salade
Prix	10
Categorie	CAT 1
Photo1	
Photo2	
	<input type="button" value="Modifier"/>
	<input type="button" value="Supprimer"/>