

# Projet 2019 : création d'une application participative "recettes de cuisine" (MarmitUM)

Vous travaillerez **en binôme ou en trinôme**.

Durée : 18 h de TP prévues du 13 novembre au 18 décembre.

**À rendre sous forme d'archive** (fichier zip, tar.gz ou autre) **par email** :

[mathias.chouet@umontpellier.fr](mailto:mathias.chouet@umontpellier.fr) , **le 20 décembre** à 18h au plus tard.

Lors de la dernière séance de TP (18 décembre) vous devrez me **présenter en 10mn** maximum votre projet, même s'il n'est pas terminé (j'en tiendrai compte).

## Énoncé

Ce projet consiste à construire une application Web qui gère des recettes de cuisine.

Une **recette de cuisine** est caractérisée par :

- un titre
- un type de recette : entrée, plat ou dessert
- une catégorie : cuisine française, sénégalaise, cambodgienne...
- un temps de préparation (nombre de minutes)
- un nombre de personnes (ex: "pour 6 personnes")
- le nom ou pseudo de l'auteur
- l'adresse email de l'auteur
- le texte de la recette (liste des ingrédients, instructions étape par étape)
- une photo
- une date d'ajout

L'application permettra de **lister** les recettes existantes et **rechercher** parmi elles, **saisir** de nouvelles recettes, **supprimer** des recettes existantes.

Les données seront stockées **dans un fichier JSON**.

La suppression de recettes nécessitera une **authentification** (login, mot de passe).

L'application devra avoir un **style** et une **ergonomie** agréables, et sera **dynamique**.

Les échanges seront effectués à travers des **Webservices**, en utilisant le format **JSON**.

L'application devra être consultable sur un appareil mobile (design adapté aux petits écrans).

# Avant de commencer

## Déroulement du développement

**Vous êtes libres de l'organisation, de la présentation et du style** : des consignes de base sont données mais vous pouvez changer ce qui vous plaît : soyez créatifs !

Consultez la documentation sur Internet, relisez le cours et les corrigés. Vous pouvez communiquer entre groupes, mais votre code doit rester personnel (pour un binôme / trinôme). Gardez toujours la console de votre navigateur ouverte (F12) !

**Votre application devra comporter 5 pages, accessibles à travers un menu :**

- Accueil
- Liste statique des recettes
- Liste dynamique des recettes, avec recherche
- Ajout d'une recette
- Connexion

Vous pouvez ajouter d'autres pages mais cela ne vous rapportera pas de points.

## 1 - Squelette de l'application Web (3 points)

### Organisation du code

Votre application sera composée de plusieurs pages (HTML, PHP), plusieurs Webservices (PHP), du code côté client (Javascript) et du style (CSS).

Organisez proprement votre répertoire de travail : créez un répertoire pour le projet, puis des sous-répertoires si besoin. Nommez vos fichiers pour qu'il soit facile de comprendre leur rôle (ex: nommez votre webservice de liste des recettes `liste_recettes.php` plutôt que `projet_1.php`).

Vous n'oublierez pas d'indenter les blocs de code pour faciliter la lisibilité et éviter les erreurs.

### Première page - Accueil

a) Créez une page d'accueil en HTML, avec un entête et un pied de page. L'entête contiendra le titre de l'application, et un logo de votre choix. Le pied de page contiendra vos noms et la date de dernière mise à jour du site.

Le centre de la page contiendra un texte de bienvenue qui présente brièvement l'application. Remplissez la partie `<head>` comme vous l'avez appris : titre, codage de caractères etc.

b) Ajoutez un menu en haut de la page. Ce menu servira à passer d'une page à l'autre. Pour l'instant il n'y a qu'une page, mais vous en ajouterez par la suite. Vous enrichirez au fur et à mesure le menu par de nouveaux liens (un par page).

c) Incluez dans le <head> de la page un fichier Javascript, que vous remplirez plus tard.

d) Incluez aussi un fichier CSS, et créez un style basique pour votre page (ex: couleur de fond, bordures, padding...) . Agrémentez votre style autant que possible, mais arrangez-vous pour que ça ne fasse pas mal aux yeux !

## 2 - “Base de données” (2 points)

a) Ce que nous appellerons “base de données” est en fait un fichier JSON, qui contiendra les données des recettes. Créez un fichier `data.json`, contenant une liste vide [ ]

b) Créez un fichier “`bdd.php`”. Ce fichier doit contenir deux fonctions :

- une fonction `get_recettes()` qui lit les données depuis le fichier `data.json` et les retourne sous la forme d'un tableau PHP (array)
- une fonction `add_recette($r)` qui prend en argument un tableau contenant les données d'une nouvelle recette, et ajoute ces données au fichier `data.json`

Par la suite vous réutiliserez `bdd.php` dans vos Webservices afin de consulter et ajouter facilement des recettes.

## 3 - Affichage des données en liste (3 points)

a) Créez une nouvelle page pour **visualiser la liste** des recettes. Elle utilisera `bdd.php` pour obtenir les données des recettes. Les recettes seront affichées avec toutes leurs caractéristiques et leur photo, les unes au dessus des autres.

b) Ajoutez un lien vers cette page dans le menu, et assurez-vous que l'on puisse passer d'une page à l'autre (page d'accueil et liste des recettes) à l'aide du menu.

**Important** : si plusieurs de vos pages contiennent des parties communes (un menu, un entête ou un pied de page par exemple !), vous pouvez utiliser le mécanisme “include” de PHP afin d'éviter de répéter le code du contenu commun dans chaque page.

## 4 - Liste dynamique, recherche (3 points)

### Liste dynamique

- a) Créez une nouvelle page et ajoutez-la au menu. Cette page affichera également la liste des recettes, mais cette fois **de façon dynamique en utilisant AJAX** (Javascript + Webservices).
- b) Créez un Webservice pour lire les données depuis la base de données. Les données devront être renvoyées au format JSON (rien ne doit être affiché à l'écran).
- c) Placez un bouton "afficher les données" dans la page. Un appui sur ce bouton devra appeler le Webservice et afficher les données dans la page, avec le même rendu (toutes les caractéristiques plus l'image) que dans la page statique réalisée au point 3.

### Recherche de recettes

- d) Modifiez le Webservice pour permettre de faire une recherche parmi les recettes. Pour cela, ajoutez **un paramètre** au service ainsi qu'à la fonction `get_recettes()`, afin de ne renvoyer que les données dont un des champs correspond au paramètre.  
Pour filtrer les données, vous pouvez utiliser [`strpos\(\)`](#) ou [`preg\_match\(\)`](#).

Par exemple, si le paramètre vaut "dessert", le service renverra les recettes de type dessert. Si le paramètre vaut "marie", le service renverra la recette publiée par Marie Tournelle, mais aussi celle dont le texte contient "Faire chauffer au bain-marie".

- e) Ajoutez un champ de texte à côté du bouton "afficher les données". Lorsqu'on appuie sur le bouton, la valeur du champ de texte doit être passée au service comme paramètre de recherche, et seuls les résultats correspondant à la recherche doivent être affichés.

## 5 - Ajout et suppression de données (4 points)

### Ajout d'une recette

- a) Créez un Webservice qui permet d'ajouter une nouvelle recette dans la base de données, en utilisant `bdd.php`.
- b) Sur la même page que la liste dynamique, créez un formulaire HTML pour ajouter une recette. Pour gérer l'image, utilisez un simple champ de texte dans lequel on saisira une URL. La validation de ce formulaire devra appeler le Webservice d'ajout, et actualiser la liste des recettes afin d'afficher les données qu'on vient d'ajouter.

### Suppression d'une recette

a) Modifiez la fonction `add_recette($r)` : lorsqu'on ajoute une recette, il faut maintenant lui ajouter automatiquement un identifiant unique `id`, généré avec le code PHP suivant:

```
md5(rand())
```

b) Ajoutez une fonction `del_recette($id)` dans `bdd.php`, qui permette de supprimer une recette de la base de données en prenant en paramètre son identifiant.

c) Créez un Webservice qui permette de supprimer une recette, en prenant en paramètre son identifiant (`id`). Vous utiliserez bien sûr `del_recette($id)` que vous venez d'écrire.

d) Modifiez la liste dynamique pour afficher à côté de chaque recette un bouton "supprimer". Un appui sur ce bouton devra appeler le Webservice de suppression, et actualiser la liste des recettes à l'écran. Vous pourrez utiliser la fonction Javascript "confirm" pour demander confirmation avant de supprimer.

Note: vous devrez stocker quelque part l'id de la recette pour pouvoir demander sa suppression - une bonne idée est d'utiliser les attributs HTML "data" ([aide en anglais](#)).

## 6 - Authentification (3 points)

Nous voulons maintenant ajouter un système d'authentification pour les utilisateurs de l'application. Un utilisateur sera identifié par un nom d'utilisateur (login) et un mot de passe. L'objectif est de n'autoriser l'ajout et la suppression de données qu'aux utilisateur authentifiés.

### Authentification

a) Créez un Webservice d'authentification. Ce service recevra deux paramètres (nom d'utilisateur et mot de passe) et devra contrôler la validité du couple login / mot de passe. Il devra renvoyer :

- le nom d'utilisateur si la personne est connue et a fourni le bon mot de passe
- un code d'erreur dans le cas contraire

Vous utiliserez le mécanisme des [sessions PHP](#) pour définir si un utilisateur est connecté ou non. Si un utilisateur a fourni un couple login / mot de passe correct, une session sera créée. Les utilisateurs et leurs mots de passe seront stockés **dans un fichier JSON dédié**.

b) Créez un formulaire d'authentification avec deux champs (nom d'utilisateur, mot de passe) et un bouton pour se connecter. L'appui sur le bouton de connexion devra appeler le Webservice d'identification, et en fonction de sa réponse modifier l'affichage : si l'utilisateur est correctement identifié, on affichera un message de bienvenue (ex: Bonjour Claude) et un bouton pour se déconnecter. Sinon, on affichera une erreur et on proposera à nouveau de se connecter.

Lorsqu'on change de page, l'état de connexion devra être conservé. Pour cela, on utilisera les sessions PHP pour déterminer si l'utilisateur est connecté ou non.

### Restriction d'accès pour la suppression

c) Toujours en utilisant les sessions PHP, modifiez le Webservice de suppression pour qu'il ne soit accessible qu'aux utilisateurs authentifiés. Dans le cas contraire, il devra maintenant renvoyer une erreur.

Dans la liste dynamique, lors d'un appui sur le bouton de suppression d'une recette, vous devrez afficher une erreur si l'utilisateur n'est pas authentifié.

Note: on pourrait simplement masquer les boutons de suppression de l'interface, mais cela n'empêcherait pas un vilain hacker d'appeler le Webservice directement !

## 7 - Style et ergonomie (2 points)

2 points seront accordés au style et à l'ergonomie de votre application. Vos pages devront être agréables à lire et utiliser l'espace correctement. Un utilisateur doit pouvoir comprendre au premier coup d'œil comment utiliser votre application et trouver toutes les informations dont il a besoin. Par exemple, si la police est trop petite, s'il faut faire défiler la page pour trouver le login ou le menu, ce n'est pas très pratique...

Ce n'est **pas obligatoire** pour avoir tous les points, mais vous pouvez utiliser un framework CSS pour agrémenter votre style. Parmi les plus connus, il y a [Bootstrap](#) (facile à utiliser) ou [Materialize](#) (plus difficile), mais il en existe beaucoup d'autres.

Pour utiliser un de ces frameworks, téléchargez le(s) fichier(s) CSS et incluez-le(s) à vos pages comme vous incluriez n'importe quel autre fichier CSS.

L'ajout de classes CSS aux éléments de votre page permet de bénéficier de l'un ou l'autre des styles prévus par les frameworks (consultez la documentation).

**Attention:** un framework CSS mal utilisé peut rendre votre site moins beau !

## 8 - Navigation mobile (1 point)

Cette application doit pouvoir être visible sur un appareil mobile, tel qu'un téléphone ou une tablette. Ces appareils ont un petit écran, souvent orienté verticalement.

Modifiez le style si nécessaire pour que l'application s'affiche correctement sur un petit écran vertical. Vous utiliserez pour cela une ou plusieurs **media-query CSS**. Pour vérifier le rendu, réduisez la fenêtre de votre navigateur et assurez-vous que les pages restent lisibles.