



LES RECETTES DU PROGR@MMEUR

Projet web réalisé par *Manon Seznec, Ghaïs Mokhtar-Slimane et Grégory Almirall*

Janvier 2020

Sommaire :

I- Introduction.	2
II- L'organisation.	3
2.1 Utilisation de la méthodologie Agile.	3
2.2. Outil de versionning.	5
2.3. Outil de communication.	5
2.4. Google Drive.	6
III- Choix techniques.	6
3.1 Architecture projet.	6
3.2 Visual studio code.	6
3.3 Looping.	6
3.4 Serveur.	6
3.5 Bootstrap.	6
IV- Conceptions.	7
4.1. La Base de données.	7
4.1.1 Convention de nommage.	7
4.1.2 Réflexion sur les tables.	7
4.1.3 Du MCD au MLD.	7
4.2. La charte graphique.	9
4.3. La maquette.	9
4.4. UX Design.	17
V- Présentation des différentes pages.	18
5.1 Header.	18
5.2 Footer.	18
5.3 Accueil.	18
5.4. Inscription.	19
5.5 Login.	19
5.6 Recettes.	19
5.7 Contact.	20
5.8 A Propos.	20
5.9 Mes Recettes.	20
5.10 Créer Recette.	20
5.11 Voir Recette (le template).	21
5.12 Admin Contact.	21
5.13 Admin User.	22
VI- Tests et extrait de codes commentés.	22
6.1 Les principes de bases du Test.	22

6.2 Mise en place d'une architecture MVC.	22
6.3 Vérification et déploiement :	23
6.4 Démonstration du Code et des tests unitaires.	24
6.4.1 Tests de Base de Données.	24
6.4.2 Test du code en Architecture MVC :	25
VII- SECURITE	33
7.1 Fichier .htaccess	33
7.2 Niveau utilisateur :	33
7.3 Filtre des données.	33
7.4 Contrôle.	33
7.5 Token.	33
7.6 MVC.	34
VIII- Conclusion.	34

I- Introduction.

Dans le pays de la gastronomie, le site de recettes de cuisine est devenu une véritable institution. Certains sites peuvent rassembler jusqu'à 70 000 recettes et 12 millions de visiteurs uniques chaque mois.

En tant que prestataire web, il nous a été demandé de réaliser un site permettant de gérer des recettes de cuisine où il sera possible de manipuler plusieurs types de contenu (ingrédients, unités, produits, etc.), de s'inscrire, créer des recettes ainsi que d'autres fonctionnalités.

Ce sera un site communautaire où les "programmeurs" pourront partager entre eux leurs meilleures recettes.

II- L'organisation.

2.1 Utilisation de la méthodologie Agile.

Afin de répondre au mieux à la demande et d'avancer en toute intelligence dans ce projet, nous avons choisi d'avancer de manière Agile.

Pour mieux maîtriser les impondérables, nous avons divisé les tâches, nous y avons défini le périmètre, le temps alloué et la personne ou l'équipe en charge de manière optimale.

Aussi, nous sommes restés ouverts afin de s'adapter tout le long du projet.

Nous avons commencé par faire une étude, analyser la demande exacte du projet et séparer le travail en plusieurs sprints :

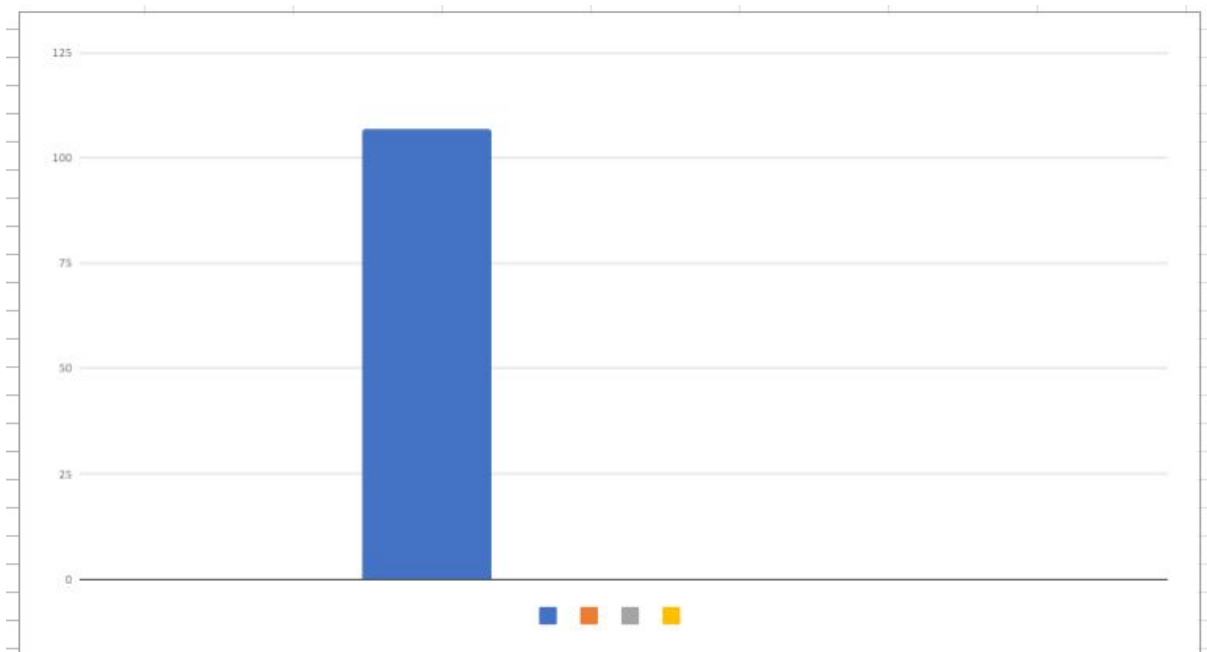
1. le travail à faire en amont => du 14 au 16 janvier.
2. la partie "pages web" à réaliser => du 17 au 24 janvier.
3. les écrits, la présentation à préparer => du 25 au 29 janvier, jour de la soutenance.

Chaque sprint a été lui-même divisé en tâche que vous pouvez voir sur ce "Kanban" où le poids sera la difficulté générale de la tâche à accomplir.

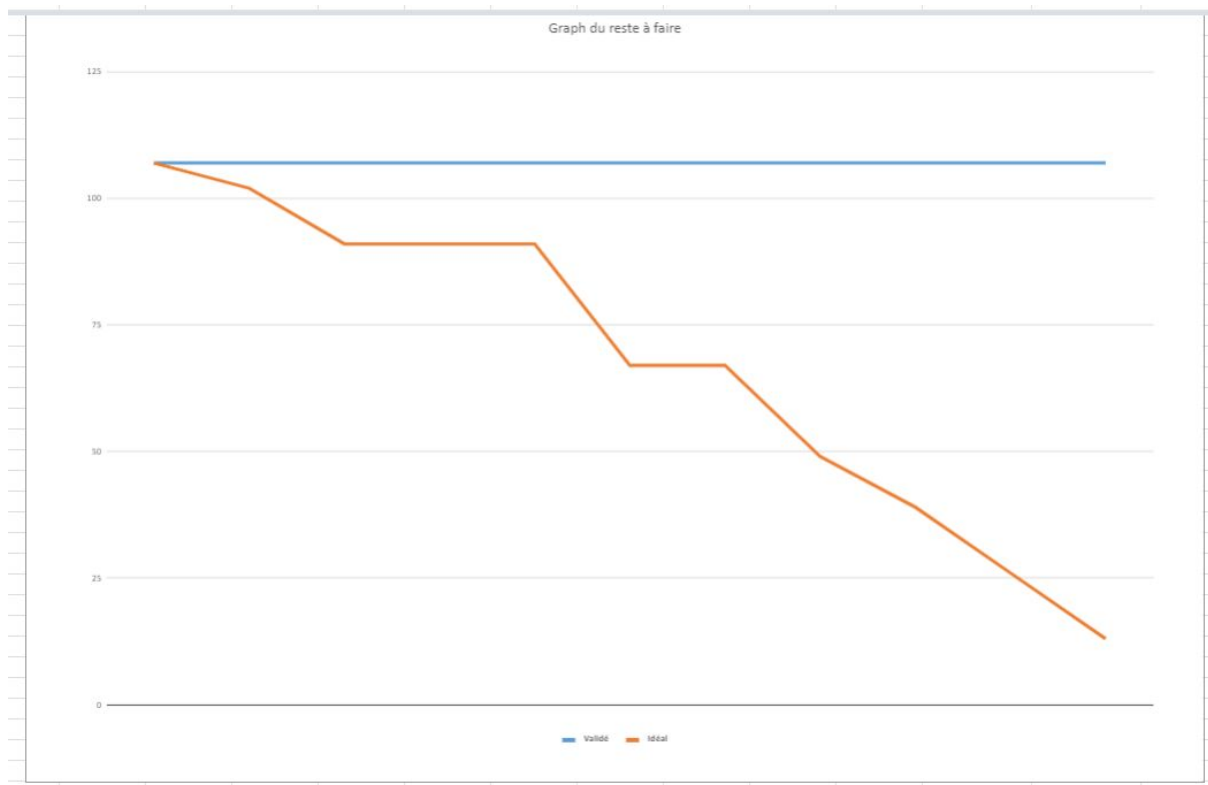
Kanban initial :

	Tâches	Poids	Acteur	Date limite	A faire	En cours	A valider	Validé	Finalisés
En Amont	BDD->MCD->SQL	5	Gh/Gr	16/01/2020	0				0
	Plan du document à rendre -> drive	3	Gh/Gr	15/01/2020	0				0
	Charte graphique / Maquette	3	Manon	16/01/2020	0				0
	Header/Footer Html Css	3	Manon	16/01/2020	0				0
	Mettre en place Github	2	Groupe	15/01/2020	0				0
Partie Web	Accueil html/php/sql	8	Manon	19/01/2020	0				0
	Inscription / Login html/php/sql	8	Gregory	19/01/2020	0				0
	Espace Privée html/php/sql	8	Ghais	19/01/2020	0				0
	Template recette html/php/sql	5	Gregory	21/01/2020	0				0
	Pdf Mail en PHP	5	Ghais	21/01/2020	0				0
	Formulaire Ajouter html/php/sql + formulaire mo	8	Manon	21/01/2020	0				0
	Recettes html/php/sql/JS ?	13	Groupe	23/01/2020	0				0
	A propos	5	Gregory	22/01/2020	0				0
	Contact	5	Ghais	22/01/2020	0				0
	Compte administrateur	13	Groupe	24/01/2020	0				0
Presentation	Powerpoint	5	Groupe	26/01/2020	0				0
	Finir le Doc	8	Groupe	26/01/2020	0				0
									0
									0

Graphique du Kaban :



Graphique du reste à faire :



Etat d'avancement :

Etat	Total	15/01/2020	16/01/2020	17/01/2020	18/01/2020	19/01/2020	20/01/2020	21/01/2020	22/01/2020	23/01/2020	24/01/2020
Validé	107	107	107	107	107	107	107	107	107	107	107
Idéal	107	102	91	91	91	67	67	49	39	26	13

Outre les échanges quotidiens, nous avons fait des réunions à la fin de chaque sprint et début des nouveaux afin de redéfinir les tâches quand nécessaire et discuter de nos ressentis sur ce qui nous avait été alloué en début de projet pour le redistribuer quand nécessaire.

2.2. Outil de versionning.

Nous avons fait le choix de travailler sur un outil de versionning, Github. Cela nous permet de ne pas perdre tout notre code en cas de fausse manipulation et nous pourrions aussi grâce à Github travailler ensemble mais dans des lieux différents. Pour plus de commodités, nous avons travaillé sur une seule branch master.

2.3. Outil de communication.

Nous échangerons aussi dans un espace de travail sur SLACK ou nous ferons quotidiennement le point sur les avancées et les difficultés rencontrées dans notre projet.

2.4. Google Drive.

Google Drive servira enfin dans sa fonction de partage pour notre rapport, le powerpoint ainsi que le fichier Excel d'évolution de projet que nous avons mis en place.

III- Choix techniques.

3.1 Architecture projet.

Nous avons choisi d'organiser notre code en MVC. Manon ayant déjà réalisé un projet web. Elle a mis en place l'architecture générale du projet afin de donner une vision global de notre site et qu'on puisse commencer à créer nos pages.

3.2 Visual studio code.

Nous avons choisi d'harmoniser notre travail en codant sur Visual Studio Code, principalement pour ses fonctions d'échanges avec Github.

3.3 Looping.

Looping nous sert à conceptualiser la base de donnée et les relations établies.

3.4 Serveur.

Tout ce projet repose sur des serveurs apaches installés via Xampp ou Wampserver. Configuration d'un serveur virtuel pointant vers notre nom de domaine.

3.5 Bootstrap.

Afin de rendre notre site responsive et plus agréable graphiquement, nous utiliserons le framework Bootstrap.

IV- Conceptions.

4.1. La Base de données.

4.1.1 Convention de nommage.

Il n'y a, et ceux pour toutes les tables et colonnes, aucun espace, caractère spécial. Seules les lettres de l'alphabet et le underscore sont utilisés, le tout en minuscule.

Toutes les tables commencent par "rp_" (pour recette programmeur) et sont au singulier. Il n'y a qu'un mot par table qui se veut explicite. Chaque table commence par une colonne ID.

Exemple de table : rp_recette

Les colonnes commencent par les trois premières lettres de la table, suivi d'un underscore, les mots utilisés ici se veulent aussi être explicite.

Exemple de colonne : rec_nom

Les clés étrangère sont elles aussi préfixées par les trois premières lettres de la table, un underscore suivi de fk, puis l'id de la table associative.

Exemple de clé étrangère : rec_fk_difficulte_id

4.1.2 Réflexion sur les tables.

La réflexion s'est articulée autour de la table "rp_recette" qui sera le noyau de notre base de donnée.

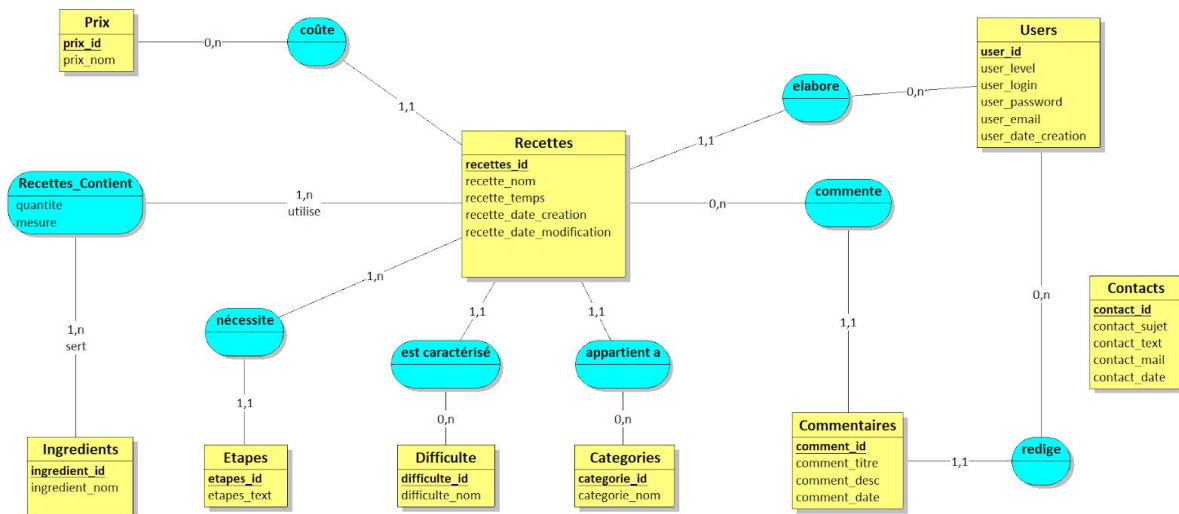
Afin d'éviter les redondances, comme il est recommandé dans une création de base de données, les colonnes Prix, Catégories et Difficulté par exemple sont sorties de la table "rp_recette" et se transforment elles-mêmes en tables.

De même la table "rp_ingredient" est séparée de la table "rp_quantite" où se trouvera la quantité et l'unité de mesure de l'ingrédient pour éviter les répétitions et pouvoir les séparer pour de possibles futures fonctionnalités.

Chaque colonne ID des tables représente la clé primaire et est auto-incrémentée.

4.1.3 Du MCD au MLD.

Ceci est notre MCD avant application de notre convention de nommage finale et nos améliorations de la base de donnée au fil du projet :

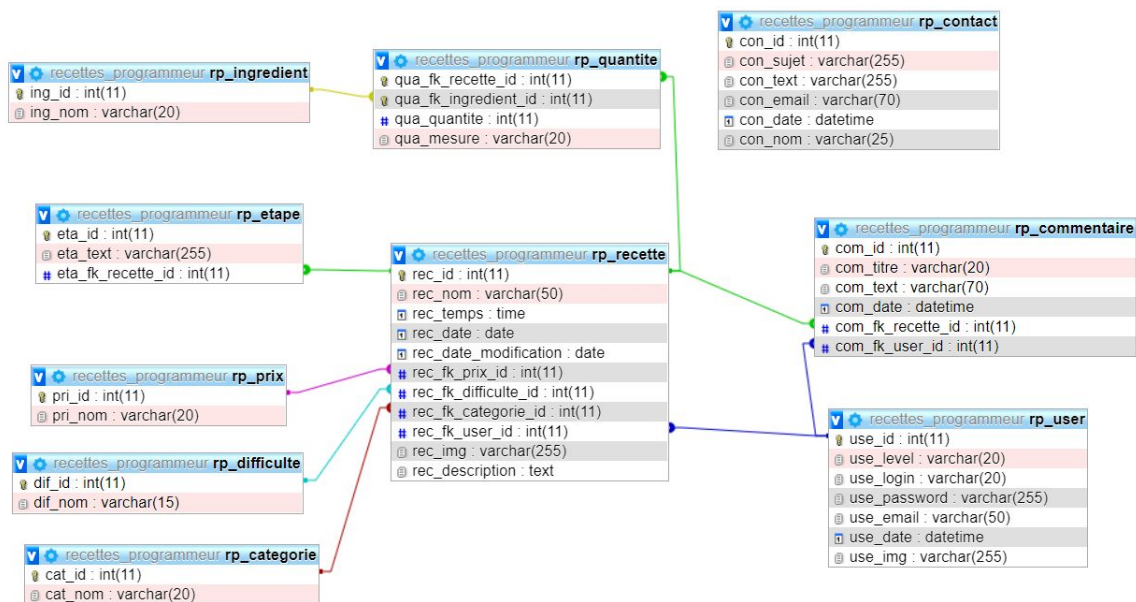


Qui nous a amené à ce MLD :

```

Ingrédients = (ingredient_id SMALLINT, ingredient_nom VARCHAR(20));
Difficulté = (difficulte_id SMALLINT, difficulte_nom VARCHAR(15));
Prix = (prix_id SMALLINT, prix_nom VARCHAR(20));
Catégories = (categorie_id SMALLINT, categorie_nom VARCHAR(20));
Users = (user_id SMALLINT, user_level VARCHAR(20), user_login VARCHAR(20), user_password VARCHAR(20), user_email VARCHAR(50), user_date_creation DATETIME);
Contacts = (contact_id SMALLINT, contact_sujet VARCHAR(255), contact_text VARCHAR(255), contact_mail VARCHAR(70), contact_date DATETIME);
Recettes = (recettes_id VARCHAR(50), recettes_nom VARCHAR(50), recettes_temps TIME, recettes_date_creation DATE, recettes_date_modification DATE, #prix_id, #difficulte_id, #categorie_id, #user_id);
Commentaires = (comment_id SMALLINT, comment_titre VARCHAR(20), comment_desc VARCHAR(70), comment_date DATETIME, #recettes_id, #user_id);
Etapes = (etapes_id SMALLINT, etapes_text VARCHAR(255), #recettes_id);
Recettes_Contient = (#recettes_id utilise, #ingredient_id sert, quantite SMALLINT, mesure VARCHAR(20));
  
```

Ceci est le MCD final tiré du concepteur de phpmyadmin.



4.2. La charte graphique.

- **Identité du Site:**

- Site très photographique enrichie par la communauté avec des images de plats et végétaux, avec un peu d'humain sur la page d'accueil ainsi que quelques parallèles entre la cuisine et le développement :
 - exemple : les pages connexion et inscription.
- Rappel de la programmation dans les textes et titres du site ainsi que la page 404.

- **Les couleurs utilisées :**

- 2 couleurs neutres = blanc et gris foncé.
- Une dominante de 4 couleurs surtout réparties entre les titres, les liens et les boutons:

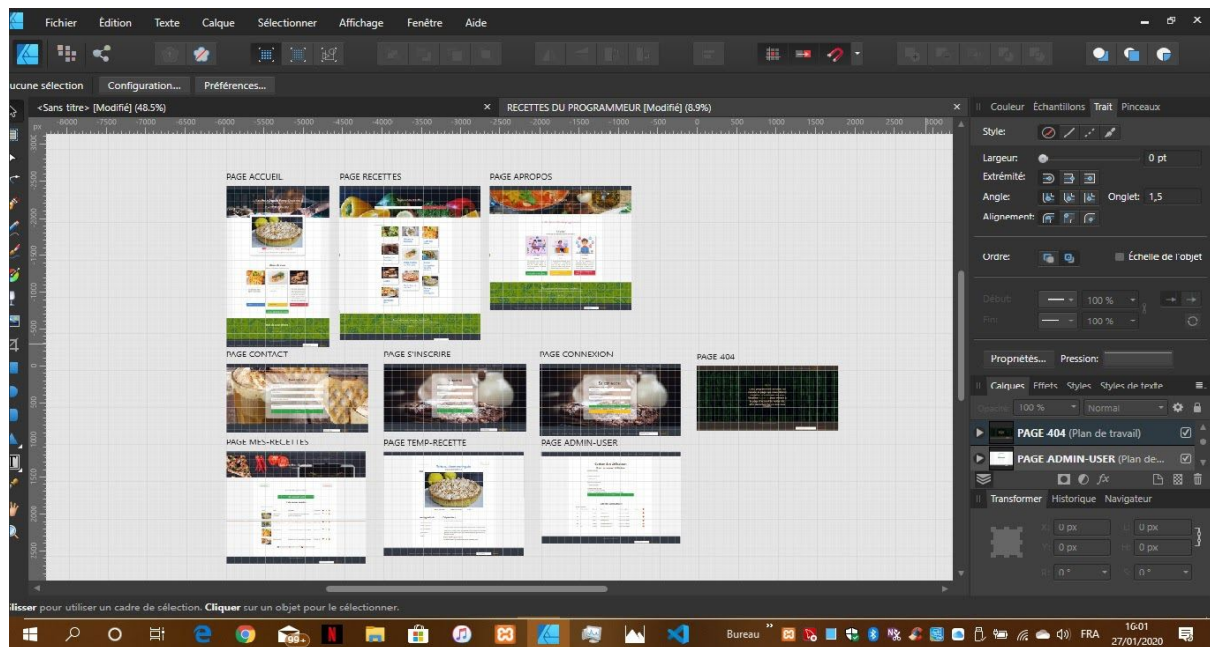
```
--colorGreen: #6E8C03;  
--colorYellow: #F29F05;  
--colorOrange: #F27405;  
--colorRed: #F22613;
```

- **La typographie:** Roboto.
- **Le logotype :** Pour le menu connexion et les différentes actions sur les données : voir, modifier et supprimer, un oeil, update et sens interdit.
- **Le positionnement des titres:** centrés dans une bannière photographique ainsi qu'un bouton ou un lien.
- **Le logo** sera une tasse fumante à gauche de la barre de navigation.
- La mise en page des différents éléments : dimensions, colonnes, espaces, marges, placement du logo, les différents éléments visuels (bandeau, boutons...) sera calculé pour répondre au mieux aux attentes du visiteur.

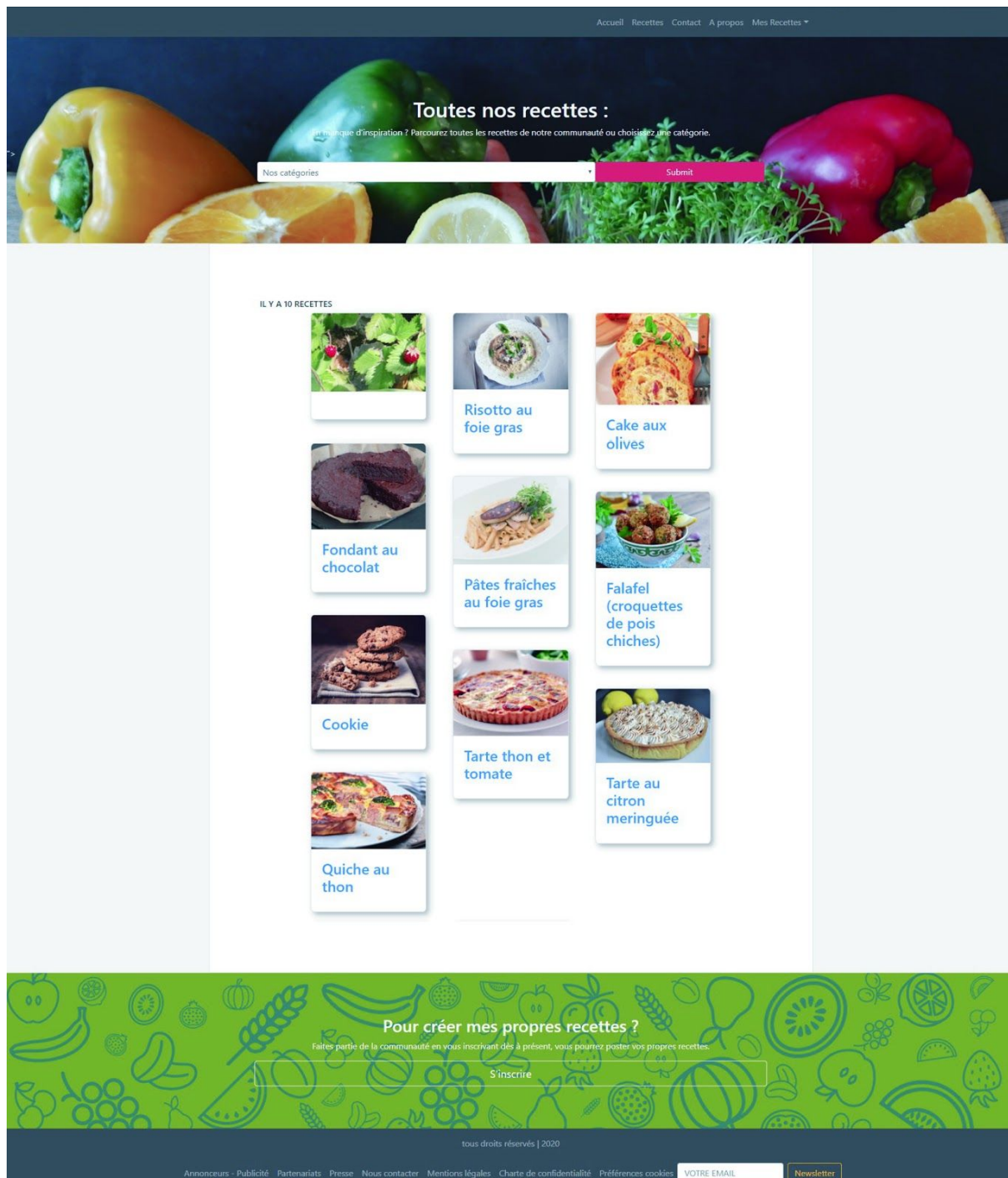
4.3. La maquette.

Maquette de 10 planches préparées sur Affinity Designer présentant les différentes pages de la 1ère version du site :

Maquette générale :



Exemple pour la page d'accueil:



Exemple pour le template recette:


[Accueil](#) [Recettes](#) [Contact](#) [A propos](#) [Mes Recettes](#)

[Toutes les recettes](#) [Mes recettes](#)

Tarte au citron meringuée

Créé par antoine le 2020-01-23

Dessert



Temps: 01:00:00 Difficulté : facile Coût : €

Les ingrédients

- farine 250 grammes
- beurre 50 grammes
- sucres 200 grammes
- jaune d'oeuf 2
- citron 1

Préparation:

Battez l'oeuf avec les sucres et le sel

Ajoutez la farine en 1 fois, pétrissez du bout des doigts

Ajoutez le beurre mou en morceaux, pétrissez rapidement et formez une boule

Faites cuire 10 minutes à 180°C, puis enlevez le "poids" et remettez 8-10 minutes pour bien cuire le centre. La pâte doit être dorée (sinon prolongez la cuisson de quelques minutes)

Note du Programmeur:

Un dessert excellent qui saura plaire aux grands comme aux petits.

tous droits réservés | 2020

[Annonceurs](#) [Publicité](#) [Partenariats](#) [Presse](#) [Nous contacter](#) [Mentions légales](#) [Charte de confidentialité](#) [Préférences cookies](#) [Newsletter](#)

Exemple de formulaire :

Accueil Recettes Contact A propos Mes Recettes ▼

Se connecter

Veillez entrer votre adresse mail*

Veillez entrer votre mot de passe:*

[Mot de passe oublié ?](#)

Se connecter

S'inscrire

tous droits réservés | 2020

Annonces - Publicité Partenariats Presse Nous contacter Mentions légales Charte de confidentialité Préférences cookies

Newsletter

4.4. UX Design.

Nous avons essayé de nous rapprocher le plus possible de ce que préconise le UX design, aujourd'hui très prisé pour la conception de site web.

Pour cela nous avons commencé par définir le public visé afin de mettre en avant l'utilisateur dans la conception du site. Sur les différentes pages, des appels au visiteur pour aller plus loin dans sa découverte du site et éviter un nombre de rebond trop important, les liens sont visibles et les pages expliquées pour qu'il y ait la meilleure compréhension possible.

V- Présentation des différentes pages.

La réalisation de ce site a été faite avec toujours à l'esprit, que le visiteur, qui deviendra membre, puisse toujours être captivé, et puisse naviguer avec aisance.

5.1 Header.

Description :

Le header, qui sera le même pour tout le site est seulement composé de la barre de navigation.

Celle-ci permet dès à présent de naviguer sur l'ensemble du site.

La dernière partie de la barre de navigation est un sous-menu différent selon le niveau d'autorisation de l'utilisateur.

En effet, si **l'utilisateur n'est pas encore connecté**, il a alors deux possibilités, se connecter ou s'inscrire.

L'utilisateur membre mais pas administrateur, pourra voir ses recettes ou se déconnecter.

Enfin **l'administrateur** a droit à un sous-menu plus riche avec la possibilité de gérer, en plus de ses recettes, les contacts dans "admin contact" et gérer les utilisateurs dans "admin users".

Futures fonctionnalités:

L'administrateur pourra aussi gérer toutes les recettes du site dans "admin recettes". Possible mise en place d'un forum, d'une page shopping, d'une page liste de courses.

5.2 Footer.

Description :

Le footer contient des liens vers les pages administratives ainsi qu'un formulaire d'inscription à la newsletter du site.

5.3 Accueil.

Description :

La page d'accueil, outre le titre du site et une rapide description contient dans une grande section, une recette mise en avant, sélectionné par l'administrateur du site, afin de faire une promotion de recette.

Elle est suivi d'un menu du jour avec la sélection d'une 'entrée, un plat principal, et un dessert.

Autant les recettes mises en avant que celles du menu du jour sont cliquables et mènent vers la recette en question.

Enfin, il y a une invitation à cliquer sur un bouton pour se rendre vers toutes les recettes.

5.4. Inscription.

Description :

S'inscrire renvoie vers un formulaire d'inscription.

Une fois le pseudo, l'email et le mot de passe envoyés, une vérification se fait pour ne pas qu'il y ait de doublons. Si tel est le cas, un message s'affiche afin de changer de pseudo ou email. Autrement, l'inscription est prise en compte dans la base de donnée, et l'utilisateur peut commencer à créer ses propres recettes et avoir accès à la page "mes recettes".

Futures fonctionnalités:

Lorsque la communauté sera assez grande, possible vérification par envoie d'email.
Bouton d'inscription directement avec son compte gmail/facebook.

5.5 Login.

Description :

Formulaire pour se connecter à l'aide de son email et son mot de passe.

S'il est inscrit en base de donnée, cela le renvoie directement vers "mes recettes".

Futures fonctionnalités:

mise en place du liens vers mot de passe oublié pour que l'utilisateur puisse le changer si nécessaire.

5.6 Recettes.

Description :

Accessible pour tout le monde, toutes les recettes du site sont présentées sous forme de mini photo et titre. Il suffit d'un clique sur le titre pour la voir.
le sélecteur nos catégories a été ajouté mais n'est pas fonctionnel pour l'instant.
Une invitation en fin de page à s'inscrire renvoie vers la page "inscription".

Futures fonctionnalités :

Le sélecteur renverra vers une page de recette par catégorie.
Il faudra aussi rajouter une fonction recherche par nom de recette, d'ingrédient et autres caractéristiques.

5.7 Contact.

Description :

La page contact permet à tout le monde de contacter l'administrateur du site grâce à un formulaire. Le nom, l'email, le sujet et le message sont envoyés en base de donnée. Les administrateurs les retrouvent dans la page "admin contact" afin de pouvoir y répondre.

5.8 A Propos.

Description :

"A Propos" donne une brève présentation du site et de l'équipe qui l'a créé.

5.9 Mes Recettes.

Description :

"Mes recettes" affiche pour l'utilisateur connecté, la liste des recettes avec son nom, son temps de préparation et sa date de publication.
Celui-ci a pour l'instant la possibilité de la visualiser ou de la supprimer.
Un bouton lui permet d'en créer une nouvelle.

Futures fonctionnalités:

L'utilisateur connecté pour aussi modifier sa recette.
Lorsque la communauté sera plus grande, une page profil pourrait voir le jour.

5.10 Créer Recette.

Description :

Une fois sur cette page, un formulaire permet de créer sa recette en détail avec :

- un titre.
- un message descriptif ou d'accompagnement.
- la catégorie de la recette.
- le coût.
- le temps de préparation.
- les ingrédients.
- les étapes.

Une fois envoyée, la recette apparaît en base de donnée et l'utilisateur est retourné vers "mes recettes".

Deux boutons sont présent sur la page création pour plus d'ergonomie, un pour revenir vers "mes recettes" et un pour aller vers toutes les "recettes".

Futures fonctionnalités:

Pour l'instant, il n'est possible que de mettre un ingrédient, il sera très rapidement possible d'en mettre plusieurs, le bouton d'ajout existe déjà, il nous faut faire le lien à la base de donnée.

Il sera possible de télécharger la photo de la recette.

5.11 Voir Recette (le template).

Description :

Peu importe dans quel parti du site on se trouve, si l'utilisateur clique sur une recette, il accède au template de la recette.

Une vue simple et épurée avec toutes les informations inscrites lors de la création.

De plus, un bouton permet de revenir vers toutes les recettes.

Un deuxième bouton apparaît si c'est un utilisateur connecté avec un lien vers "mes recettes".

Futures fonctionnalités:

Nous donnerons la possibilité aux utilisateurs inscrits et connectés de noter la recette, ou juste mettre un like avec un pouce, comme sur facebook.

Il sera aussi possible de commenter la recette, de l'envoyer par mail, de l'imprimer ou encore de la télécharger en pdf.

5.12 Admin Contact.

Description :

Les administrateurs ont accès sur cette page à tous les contacts créés par les utilisateurs du site dans la page contact.

Il lui est possible de supprimer les contacts une fois lu et répondu.

Futures fonctionnalités:

L'administrateur pourra répondre directement en cliquant sur le mail du contact.

5.13 Admin User.

Description :

Comme pour les contacts, ici l'administrateur a accès à tous les utilisateurs du site et peut supprimer leur compte s'il estime que c'est nécessaire.

VI- Tests et extrait de codes commentés.

Accueil / Login / Inscription / Recettes / Contact / A propos / Mes recettes / Créer recette / Voir recette (template) / Admin contact / admin user / header / footer.

6.1 Les principes de bases du Test.

1. Vérifier les données émises par l'utilisateur.
2. Vérifier chaque partie de notre code après écriture pour voir s'il s'exécute comme souhaité :
 - Affectation de variables
 - If de contrôle
 - Boucle d'itération
 - Fonctions Modèle BDD du MVC
 - Fonctions natives de PHP
3. Vérifier les Outputs en sortie pour l'utilisateur en BDD ou sur le Navigateur

6.2 Mise en place d'une architecture MVC.

Utilisation d'un routeur Web qui fait les liens à travers l'architecture MVC d'un site web :

Les *fonctions* **Modèles** agissent sur la Base de données. Le CRUD permet de lire les données de la base de données, les créer, les modifier ou enfin, les supprimer.

Pour chaque pages du site dans le **View**, nous avons des sections spécifiques.

Le traitement du **Controller** sont les fichiers PHP qui servent à contrôler toutes les données en transitions de l'utilisateur vers la BDD et inversement (Client/Serveur).

6.3 Vérification et déploiement :

Pour chaque liens entre le MVC et chaque action de l'utilisateur, il y a des transitions de données entre les fichiers **model**, **view** et **controller**. Ils sont à vérifier par un affichage sur le navigateur ou la Base de données.

Nous devons effectué la méthodologie suivante de la Vue au Traitement:

principe n°1. Vérifier les données rentrés par les utilisateurs.

1. Routing web : Au clique du bouton submit de la **View**, id_formulaire de l'hidden message transmet les données à la page **Controlleur** traitement-\$id_formulaire-.php
2. Récupération des données du formulaire HTML à travers le Traitement-id_form et Affectation à des variables php:
3. Test : affichage sur le navigateur avec echo ou var_dump(array).

Dès lors, plusieurs types de traitement sont possibles selon les fonctionnalités à implémenter, représentés ici dans un sens chronologique :

4. CRUD sur la BDD :

- Insertion des données dans la BDD.
- Sélection à partir des variables du formulaire se trouvant dans la BDD.
- Modification des données en BDD.
- Supprimer de la BDD.

5. Réutilisation des données pour l'afficher sur le Navigateur web selon le choix du visiteurs.

6. Mise en forme HTML , CSS (bootstrap), jQuery en lien avec le Maquettage.

6.4 Démonstration du Code et des tests unitaires.

6.4.1 Tests de Base de Données.

Mise en place de plusieurs recettes et tests afin de voir l'implémentation dans la BDD, notamment au niveau des clés étrangères.

Table des recettes :

d	rec_nom	rec_temps	rec_date	rec_date_modification	rec_fk_prix_id	rec_fk_difficulte_id	rec_fk_categorie_id	rec_fk_user_id	rec_img	rec_description
1	Tarte au citron meringuée	01:00:00	2020-01-23	NULL	1	2	4	4	assets\upload\tartecitronmeringue.jpg	Un dessert excellent qui saura plaire aux grands c...
2	Falafel (croquettes de pois chiches)	00:30:00	2020-01-22	NULL	1	3	3	4	assets\upload\talafel.jpg	Les meilleurs Falafels qui raviront vos papilles !
3	Cake aux olives	00:30:00	2020-01-23	NULL	2	2	2	4	assets\upload\cakeauxolives.jpg	Une délicieuse entrée légère et au bon goût.
4	Tarte thon et tomate	00:35:00	2020-01-23	NULL	1	1	2	10	assets\upload\tartethontomate.jpg	Hyper rapide, pour ceux qui aiment bien cuisiner s...
5	Pâtes fraîches au foie gras	01:00:00	2020-01-23	NULL	3	2	3	10	assets\upload\patesfoiegras.jpg	Un pur régal !
6	Risotto au foie gras	01:15:00	2020-01-23	NULL	3	2	3	10	assets\upload\risotto.jpg	A déguster accompagné d'une flûte de Crémant de Lo...
7	Quiche au thon	01:00:00	2020-01-24	NULL	2	1	3	4	assets\upload\quichethon.jpg	Pour ceux qui n'ont pas beaucoup de temps.
■ Console de requêtes SQL										
38	Cookie	01:00:00	2020-01-24	NULL	1	2	4	12	assets\upload\cookie.jpg	Très bonne recette que j'ai déjà testée plusieurs ...
39	Fondant au chocolat	02:00:00	2020-01-24	NULL	1	1	4	12	assets\upload\fondantchocolat.jpg	Délicieux pour les enfants !!

Table Utilisateur :

use_id	use_level	use_login	use_password	use_email	use_date	use_img
4	10	antoine	\$2y\$10\$53fHHrOYDEPChNHxKVMC8ud1kBfcQ.btAgbQ.tKk/m5...	antoine@gmail.com	2020-01-19 10:55:28	NULL
10	1	Sophie	\$2y\$10\$cSkuaTi0psNnqwlP92u0eGig5Zeu2NgB74IKv4FUr...	Sophieaix@gmail.com	2020-01-23 21:43:29	NULL
11	1	Lilou	\$2y\$10\$QWITm8hfPBSwBFbsqrKm9OH8jfrRXsKwJCVVFPZpPE...	Lilou@yahoo.fr	2020-01-23 21:44:03	NULL
12	1	Alain	\$2y\$10\$0Tt4wbN04Y/eFOWpcipJ8.9fDmRjndW7pHyIV2V06j9...	alain@gmail.com	2020-01-23 21:44:34	NULL

Table quantité :

	qua_fk_recette_id	qua_fk_ingredient_id	qua_quantite	qua_mesure
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	31	1	250	grammes
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	31	2	50	grammes
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	31	3	200	grammes

Table Ingrédient :

	ing_id	ing_nom
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	1	farine
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	2	beurre
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	3	sucré
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	4	jaune d'oeuf
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	5	eau

6.4.2 Test du code en Architecture MVC :

Page faisant le lien entre la vue, le header et footer, ainsi le traitement correspondant.

```
inscription.php
1  <?php
2
3  require_once "private/controller/traitement.php";
4
5  require_once "private/view/header.php";
6  require_once "private/view/section-inscription.php";
7  require_once "private/view/footer.php";
8
```

Fichier traitement : Appel des fonctions modèles, des fonctions view et fonctions controller (filterInfo, email etc).

Création du chemin général des traitements, et utilisation de la variable du formulaire à la fin pour appeler la page de traitement correspondante dans son sous-dossier.

```
traitement.php
private > controller > traitement.php
1  <?php
2
3  error_reporting(E_ALL); // NIVEAU MAXIMUM DE DETECTION DES PROBLEMES
4  ini_set("display_errors", "1"); // AFFICHER DANS LE NAVIGATEUR LES ERREURS
5
6  // CHARGER LES PARAMETRES DE MON PROJET
7  require_once "private/mon-projet.php";
8
9  // TESTER SI IL Y A UN FORMULAIRE A TRAITER
10 // ON VA DETECTER SI L'INFO identifiantFormulaire EST PRESENTE
11
12 require_once "private/model/fonctions-model.php";
13 require_once "private/view/fonctions-view.php";
14 require_once "private/controller/fonctions-controller.php";
15
16 // MAINTENANT JE PEUX UTILISER/APPELER LA FONCTION filterInfo
17 // <input type="hidden" name="identifiantFormulaire" value="newsletter-create">
18 $identifiantFormulaire = filterInfo("identifiantFormulaire");
19 // TEST1: EST-CE QU'IL Y A UN FORMULAIRE A TRAITER?
20 if ($identifiantFormulaire != "")
21 {
22     // ON VEUT DES SOUS-DOSSIERS POUR RANGER SES FICHIERS DE TRAITEMENT
23     $cheminFichier = "private/controller/*/traitement-$identifiantFormulaire.php";
24
25     $tabFichierTraitement = glob($cheminFichier);
26     foreach($tabFichierTraitement as $fichierTraitement)
27     {
28         require_once $fichierTraitement;
29     }
30 }
```

Identifiant formulaire avec un hidden message pour appeler le traitement correspondant.

```
<input id="passwordNoHash" class="form-control" type="password" name="passwordDouble" placeholder="VOTRE PASSWORD">
</div>

<button type="submit" class="btn btn-success">Submit</button>

<!-- INFOS TECHNIQUES -->
<input type="hidden" name="identifiantFormulaire" value="user-inscription">
<div class="inscriptionFeedback">
```

INSCRIPTION

Récupération des données du visiteur voulant s'identifier.

Insertion en BDD dans table USER :

```
private > controller > user > traitement-user-create.php
2
3 // ETAPE0: VERIFIER LE level DU VISITEUR. RETROUVER LE $level A PARTIR DE LA SESSION
4 $level = lireSession("use_level");
5 if ($level >= 10)
6
7 // ETAPE1: RECUPERER ET FILTRER LES INFOS DU FORMULAIRE
8 $login = filtrerInfo("login");
9 $email = filtrerInfo("email");
10 $password = filtrerInfo("password");
11
12 // ETAPE2: IF VERIFIER QUE LES CHAMPS NE SONT PAS VIDES
13 if ( ($login != "") && ($email != "") && ($password != ""))
14 {
15 // ON VA VERIFIER SI L'EMAIL EST DEJA UTILISE PAR UN UTILISATEUR DEJA CREE
16 $nbEmailDuplicate = compterLigneSQL("rp_user", "use_email", $email);
17 $nbLoginDuplicate = compterLigneSQL("rp_user", "use_login", $login);
18 if (($nbEmailDuplicate == 0) && ($nbLoginDuplicate == 0))
19 {
20 // ETAPE3: COMPLETER LES INFOS MANQUANTES
21 $dateInscription = date("Y-m-d H:i:s"); // FORMAT DATETIME SQL
22 $level = 10; // USER ACTIF IMMEDIATEMENT
23
24 // SECURITE: ON HASHE LE MOT DE PASSE
25 $passwordHash = password_hash($password, PASSWORD_DEFAULT);
26
27 insererLigneSQL("rp_user", [
28 "use_login" => $login,
29 "use_email" => $email,
30 "use_password" => $passwordHash, // ON STOCKE LE MOT DE PASSE HASHE
31 "use_level" => $level,
32 "use_date" => $dateInscription,
33 ]);
34 //MESSAGE A AFFICHER EN FEEDBACK
35 $userFeedback = "VOTRE USER EST CREE. MERCI $login. ($email)";
36 }
37 else
38 {
39 $userFeedback = "DESOLE CET EMAIL OU LOGIN EST DEJA PRIS ($email)";
40 }
41 }
```

Inscription refusé si nom de compte est déjà en BDD

The screenshot shows a web form titled "S'inscrire" (Sign up) with a background image of a person's hands. The form contains four input fields: "Veuillez entrer votre pseudo:" (filled with "antoine"), "Veuillez entrer votre adresse mail*" (filled with "antoine@gmail.com"), "Veuillez entrer votre mot de passe:*" (filled with "*****"), and "Veuillez confirmer votre mot de passe:*" (filled with "*****"). Below the fields is a green button labeled "S'inscrire". At the bottom, a message reads: "DESOLE CET EMAIL OU LOGIN EST DEJA PRIS (antoine@gmail.com)".

LOGIN

- Vérifier les réponses du formulaire (email, login, password).
- Si email existant en BDD +
 - password crypté de la table équivaut au password entrée dans le formulaire +
 - Que le level en BDD est plus de 1 (actif) => ÉCRIRE EN SESSION.
 - Sinon renvoi vers la page inscription :

```
private > controller > user > traitement-user-login.php
3 // ETAPE1: RECUPERER ET FILTRER LES INFOS DU FORMULAIRE
4 $email = filtrerInfo("email");
5 $passwordNoHash = filtrerInfo("passwordNoHash");
6 // ETAPE2: VERIFIER SI LES CHAMPS NE SONT PAS VIDES
7 if ( ($email != "") && ($passwordNoHash != ""))
8 { // ON NE DEVRAIT EN TROUVER QUE 0 OU 1 LIGNE
9   // (PAS DE DOUBLON...)
10  $listeLigne = lireLigneSQL("rp_user", $email, "use_email");
11  // LECTURE DES RESULTATS
12  $password;
13  foreach($listeLigne as $tabLigne)
14  {
15    extract($tabLigne);
16    var_dump($tabLigne);
17    //ON VERIFIE QUE LES MOTS DE PASSE CORRESPONDENT
18
19    if (password_verify($passwordNoHash, $use_password))
20    {
21      if ($use_level > 0)
22      {
23        // OK LE COMPTE EST ACTIF
24        $loginFeedback = "BIENVENUE $use_login";
25        // ON VA MEMORISER DANS UNE SESSION LES INFOS DE L'UTILISATEUR
26        echreSession("use_level", $use_level);
27        echreSession("use_login", $use_login);
28        echreSession("use_email", $use_email);
29        echreSession("use_id", $use_id);
30
31        header("Location: mes-recettes.php");
32      }
33      else
34      {
35        // KO LE COMPTE EST DESACTIVE
36        $loginFeedback = "Désolé, votre compte est désactivé $login";
37      }
38    }
39    else
40    {
41      // KO LES MOTS DE PASSE NE CORRESPONDENT PAS
42      $loginFeedback = "LE MOT DE PASSE EST INVALIDE";
```

Créer une recette : 4 Tables en liens

Lors de la création de la recette, nous devons insérer les données dans les tables :

recette, ingrédient, quantité, et étapes.

Ces tables sont associatives, nous devons récupérer le last_id insérer dans la table précédente et le réutiliser à la prochaine insertion.

- Déclaration des variables, si un nom de recette est bien présent :
 - => l'insérer en BDD dans la table recette pour l'user actif :

```
traitement-recette-create.php •
private > controller > recette > traitement-recette-create.php
14 $rec_id = filterInfo("rec_id");
15 $rec_nom = filterInfo("rec_nom");
16 $rec_fk_prix_id = filterInfo("rec_fk_prix_id");
17 $rec_fk_difficulte_id = filterInfo("rec_fk_difficulte_id");
18 $rec_temps = filterInfo("rec_temps");
19 $rec_fk_categorie_id = filterInfo("rec_fk_categorie_id");
20 $qua_fk_recette_id = $rec_id;
21 $rec_description = filterInfo("rec_description");
22
23 $ing_id = filterInfo("ing_id");
24 $ing_nom = filterInfo("ing_nom");
25 $qua_fk_ingredient_id = $ing_id;
26 $qua_quantite = filterInfo("qua_quantite");
27 $qua_mesure = filterInfo("qua_mesure");
28
29 if ( ($rec_nom != "")
30 {
31
32
33 $datePublication = date("Y-m-d H:i:s");
34 $rec_fk_user_id = lireSession("use_id"); // ON RECUPERE
35
36 $objetPDOStatement = insererLigneSQL("rp_recette", [
37     "rec_nom" => $rec_nom,
38     "rec_temps" => $rec_temps,
39     "rec_description" => $rec_description,
40     "rec_img" => $rec_img,
41     "rec_fk_user_id" => $_SESSION["use_id"],
42     "rec_fk_prix_id" => $rec_fk_prix_id,
43     "rec_fk_difficulte_id" => $rec_fk_difficulte_id,
44     "rec_fk_categorie_id" => $rec_fk_categorie_id,
45     "rec_date" => $datePublication,
46     // "re_fk_us_id" => $user_id,
47     ]);
48
49 $last_rec_id = $objetPDOStatement->monLastInsertId;
50
51 // DONNER LE MESSAGE A AFFICHER EN FEEDBACK
52 $recetteFeedback = "Votre recette a bien été publiée.";
53
```

Ingrédient

- Si l'ingrédient n'existe pas encore dans la BDD, il faut l'insérer.
 - Sinon, le lire et le sélectionner pour la suite :
 - Récupérer le dernier ingrédient ID

```
private > controller > recette > traitement-recette-create.php
50
57 // Etape 2 INGREDIENT :
58
59 $ing_nom_count = compterLigneSQL("rp_ingredient", "ing_nom", $ing_nom);
60
61 $ing_nom_lire = lireLigneSQL("rp_ingredient", $ing_nom, "ing_nom");
62
63 // Si aucun ingrédient avec ce nom est en BDD, alors l'insérer
64 if($ing_nom_count < 1 ) {
65
66     $objetPDOStatement = insererLigneSQL("rp_ingredient", [
67         "ing_nom"          => $ing_nom,
68
69     ]);
70
71     // dernière id de l'ingrédient
72     $last_ing_id = $objetPDOStatement->monLastInsertId;
73
74
75 } // 2. Si déjà en BDD, récupérer son ID
76 else{
77
78     foreach($ing_nom_lire as $ing_col_lire){
79
80         extract($ing_col_lire);
81     };
82
83
84
85     $last_ing_id = $ing_id;
86
87 }
88 }
```

QUANTITÉ ET ÉTAPES

- Récupération des derniers ID : Recette et Ingrédient.

Insérer les quantités et mesure de chaque ingrédient dans la table quantité.

- De même, nous inscrivons en BDD chaque étapes de la recette.
- Enfin, le navigateur inscrit que la recette est bien publié pour l'utilisateur.

```
private > controller > recette > traitement-recette-create.php
89
90
91 // ETAPE 3: Insérer les quantités et mesures de chaque ingrédient pour une recette
92
93 $objetPDOStatement = insererLigneSQL("rp_quantite", [
94     "qua_fk_recette_id"      => $last_rec_id,
95     "qua_fk_ingredient_id"  => $last_ing_id,
96     "qua_quantite"          => $qua_quantite,
97     "qua_mesure"            => $qua_mesure,
98     ]);
99
100 // ETAPE 4: Insérer toutes les etapes d'une recette dans la table ETAPE
101
102 for($i = 0;$i<count($etapes);$i++){
103
104     if($etapes[$i] != ""){
105         $objetPDOStatement = insererLigneSQL("rp_etape", [
106
107             "eta_text" => $etapes[$i],
108             "eta_fk_recette_id" => $last_rec_id,
109         ]);
110     }
111 }
112
113 // Affichage de bonne création de la recette sur le Navigateur.
114 <<<CODEHTML
115
116 <p>VOTRE ARTICLE EST PUBLIE</p>
117 <div><a href="template-recette.php?id=$rec_id">CLIQUER ICI POUR VOIR VOTRE ARTICLE</a></div>
118
119 CODEHTML;
120
121
```


VIEW – MES RECETTES

Si l'utilisateur est connecté en SESSION =>

Lecture de ses recettes et Retranscription dans un tableau :

```
ivate > view > section-mes-recettes.php
69 // LECTURE de mes Recettes, récupération SESSION ID
70 $listeRecette = lireLigneSQL("rp_recette", $_SESSION["use_id"], "rec_fk_user_id" );
71 echo //
72 <<<CODEHTML
73 <br>
74 <table class="table table-hover">
75 <caption><div class="row justify-content-center"> :Voir ma recette
76  :Modifier ma recette |
77  :Supprimer ma recette</div></caption>
78 <thead>
79 <tr>
80 <th scope="col">Photo</th>
81 <th scope="col">Nom</th>
82 <th scope="col">Description</th>
83 <th scope="col">Publication</th>
84 <th scope="col"></th>
85 <th scope="col"></th>
86 <th scope="col"></th>
87 <tr>
88 </thead>
89 <tbody>
90 CODEHTML;
91
92 foreach($listeRecette as $tabRecette)
93 {
94     extract($tabRecette);
95     echo
96     <<<CODEHTML
97     <tr>
98     <td scope="row"> </td>
99     <td>$rec_nom</td>
100    <td>$rec_description</td>
101    <td> $rec_date</td>
102    <td><a href="template-recette.php?rec_id=$rec_id"><a href="mes-recette-update.php?rec_id=$rec_id"><a href="?identifiantFormulaire=recette-delete&rec_id=$rec_id" class="text-danger">
106    CODEHTML;
107 }
```

ADMIN : Liste et administration des membres

```
81 // $listeLigne EST COMME UN TABLEAU ORDONNE
82 $listeLigne = lireTableSQL("rp_user", "use_id");
83 echo
84 <<<CODEHTML
85     <tr>
86         <th scope="col">Id</b></th>
87         <th scope="col">Authorisation</th>
88         <th scope="col">Pseudo </th>
89         <th scope="col">Email</th>
90         <th scope="col">Date d'inscription</th>
91         <th scope="col">Avatar</th>
92         <th scope="col"></th>
93     </tr>
94 </thead>
95 <tbody>
96 CODEHTML;
97 // Pour chaque ligne d'utilisateur, l'insérer dans un tableau
98 foreach($listeLigne as $tabLigne)
99 {
100     extract($tabLigne);
101     echo
102     <<<CODEHTML
103         <tr>
104             <th scope="row">$use_id</th>
105             <td>$use_level</td>
106             <td>$use_login </td>
107             <td>$use_email</td>
108             <td>$use_date</td>
109             <td>$use_img</td>
110             <td><a href="?identifiantFormulaire=user-delete&use_id=$use_id" class="text-danger"></td>
111         </tr>
112     <<<CODEHTML;
113 }
```

VII- Sécurité.

7.1 Fichier .htaccess

Fichier de configuration du serveur web au niveau des répertoires.

Nous avons bloqué le répertoire "Private" là où les traitements s'effectuent.
A terme, dans la prochaine mise-à-jour du site, cela permettra de ne rendre visible depuis le navigateur que le fichier index.php qui servira de contrôleur pour l'ensemble des fichiers du site.
A la racine du projet, un 2ème fichier .htaccess dirige l'utilisateur vers la page 404 si la page visée est introuvable.

7.2 Niveau utilisateur :

L'administrateur peut voir tous les messages déposés sur le site, et tous les utilisateurs inscrits. Avec la possibilité de les supprimer si nécessaire.

7.3 Filtre des données.

Nous filtrons les données reçues afin d'avoir que les données qui nous intéressent dans le bon format sans possibilité d'injection de code.

7.4 Contrôle.

Après le premier filtre, nous contrôlons les données, comment elles seront utilisées, traitées et modifiées.

7.5 Token.

Afin d'éviter le hacking et l'injection SQL, nos fonctions génériques du **model** et celles du **controller** sont toutes préparées avant d'être exécutées.

VIII- Conclusion.

Ce site internet, dans sa **Version 1**, est déjà, avec l'ajout de quelques petites fonctionnalités, complètement opérationnel.

Il nous a permis de progresser et d'approfondir nos connaissances en matière de langages web et de bases de données.

La connexion entre pages et la gestion d'un projet global était aussi une nouveauté pour deux personnes de l'équipe, mais cela a permis à la troisième de transmettre son savoir et donc le conforter aussi car nous ne pouvons bien expliquer que ce nous maîtrisons parfaitement.

La dimension projet et travaille en groupe a aussi parfaitement été appréhendé et nous a appris l'utilisation de Github pour deux d'entre nous, une personne de l'équipe ayant déjà l'expérience du versionning.

Des fonctionnalités pourront être ajoutées comme une liste de courses, en ajoutant des recettes à réaliser, les ingrédients et leurs quantités s'intégreront directement à cette liste. En plus de l'ajout d'images, des vidéos pourront aussi être postées.

Nous avons dû parfois revoir nos objectifs à la hausse ou à la baisse et redistribuer les cartes en fin de sprint afin de coller au mieux aux objectifs de notre équipe. Nous avons dû nous faire confiance et beaucoup échangé, c'était une belle expérience humaine.