

## Rapport du projet tutoré de S3

### Sommaire

<b>I - Introduction .....</b>	<b>1</b>
Contexte : .....	1
Objectifs : .....	1
Définition des acteurs : .....	2
Diagramme de cas d'utilisation : .....	3
<b>II – Modèle conceptuel de donnée et dictionnaire des données .....</b>	<b>3</b>
Modèle conceptuel de données : .....	3
Dictionnaire de données : .....	4
<b>III – Schéma relationnel et code SQL .....</b>	<b>6</b>
Schéma relationnel : .....	6
Code SQL : .....	6
CODE SQL (intégré au PHP) : .....	15
<b>IV – Documentation technique du produit .....</b>	<b>16</b>
Manuel d'utilisation : .....	16
<b>V - Conclusion .....</b>	<b>17</b>

## **I - Introduction**

### Contexte :

Le contexte du projet est le suivant. Nous traversons une période où le peuple vacille entre liberté et confinement, dans un duel acharné et sans fin contre l'ennemie du peuple : la COVID 19. Chacun se fait du souci pour ses proches, pour ses projets, etc... Ce qui est notre cas évidemment. Mais on a également pensé à notre gentil voisin, Monsieur Carcenac, fruitier et légumier de père en fils depuis le 19<sup>-ème</sup> siècle. Il a besoin, pour surmonter la crise, de pouvoir continuer à vendre ses produits frais dans les alentours. Alors, nous nous sommes concertés avec lui et nous avons décidé de l'aider. Vous voulez savoir comment ? Voyons cela dans les objectifs !

### Objectifs :

Pour aider Monsieur Carcenac, nous avons choisi de créer un service web, qui permettra à des clients d'effectuer des commandes, en ligne, de produits d'agriculture, c'est-à-dire qu'ils commanderont les bons fruits et légumes de notre cher agriculteur. Ce service web, fièrement nommé **UberPrimeur**, devra réunir les fonctionnalités suivantes (*ci-après seront citées les fonctionnalités principales et utiles aux*

*clients. Nous ne parlerons pas, par exemple, du changement d'adresse mail, bien que cette dernière existe.) :*

- Permettre aux clients de créer un compte et de se connecter par la suite à leur compte sur le site.
- Permettre aux clients de demander une livraison hebdomadaire d'un panier complet qui sera constitué au gré des saisons par les sois de notre bon Mr. Carcenac. Il correspondra à un nombre de personnes (par exemple, un panier pour 4, ou pour 6), et sera livré chaque samedi au domicile du client.
- Permettre aux clients de commander ponctuellement une liste de fruits et légumes, choisis directement sur le site, et de se faire livrer chez eux.
- Permettre, étant donné qu'il existe un minimum de prix pour une livraison, à plusieurs voisins de se rassembler pour faire une commande groupée.
- Permettre au webmaster, ici Monsieur Carcenac, de gérer ses stocks de produits, y compris l'ajout d'un nouveau produit sur le site, et d'accéder aux informations de son entreprise concernant ce site (nombre de clients inscrits, etc...).

Monsieur Carcenac proposera également une recette par semaine aux utilisateurs du site, qu'il choisira selon ses envies.

### Définition des acteurs :

Parmi les acteurs de ce projet, on décompte :

#### **Les utilisateurs :**

- Les clients qui utiliseront le service web pour effectuer des commandes
- Le webmaster qui gèrera son entreprise via le site et sa page webmaster

#### **Les acteurs internes :**

- L'équipe projet, composée d'un chef de projet, Valentin MELUSSON, et de deux membres, Thomas JALLU et Lhukas NELHOMME

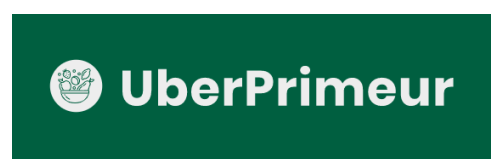
#### **Les acteurs externes :**

- L'entreprise qui se charge de la livraison, choisie et gérée par Monsieur Carcenac.

Pour ce projet, nous avons également créer un logo pour ce cher Monsieur Carcenac. Le voici :

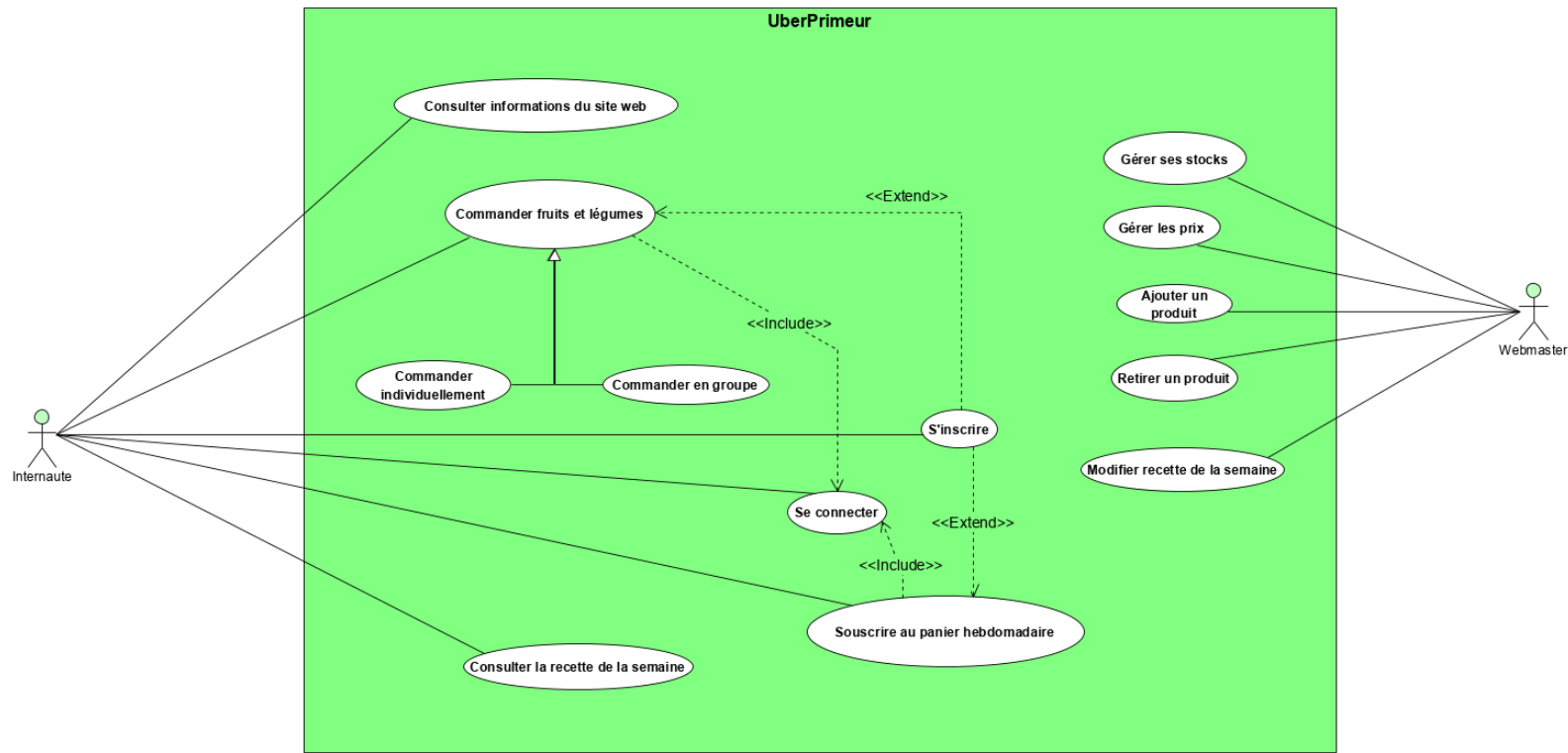


*Logo pour la page d'accueil*



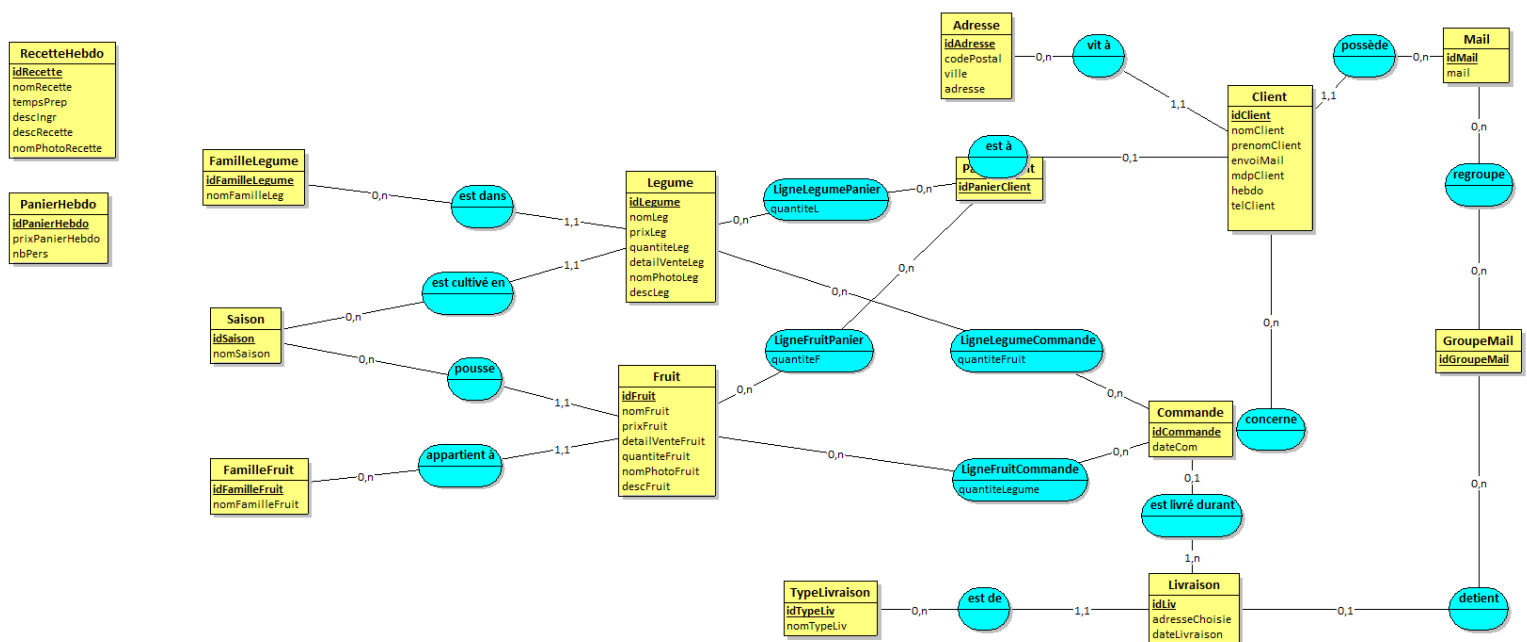
*Logo pour le menu de navigation*

## Diagramme de cas d'utilisation :



## II – Modèle conceptuel de donnée et dictionnaire des données

### Modèle conceptuel de données :



Dictionnaire de données :

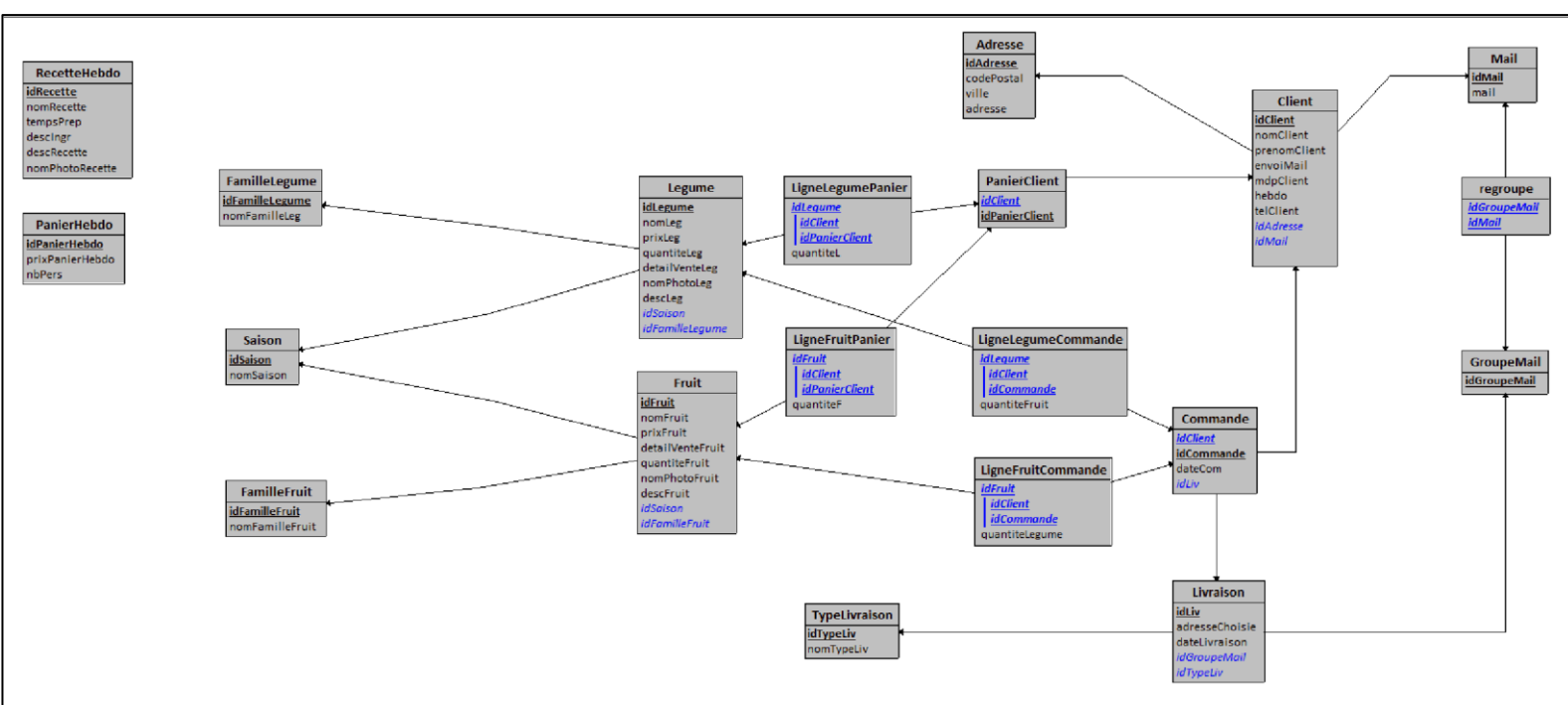
Nom	Type	Signification
idClient	Attribut	Clé primaire de la table Client, propre à chaque client, unique
nomClient	Attribut	Nom de famille du client
prenomClient	Attribut	Prénom du client
envoiMail	Attribut	Booléen permettant de savoir si le client souhaite ou non que son adresse mail soit utilisée pour le notifier
mdpClient	Attribut	Mot de passe du client, stocké de manière sécurisée
Hedbo	Attribut	Chaine de caractère permettant de savoir si le client s'est abonné au service proposant un panier hebdomadaire, et la taille du panier par conséquent ( petit, moyen ou grand)
telClient	Attribut	Numéro de téléphone du client
idAdresse	Attribut	Clé primaire de la table Adresse, propre à chaque adresse, unique
codePostal	Attribut	Code postal de l'adresse
ville	Attribut	Nom de la ville de l'adresse
adresse	Attribut	Adresse exacte (numéro, rue etc...)
idLegume	Attribut	Clé primaire de la table Legume, propre à chaque légume, unique
nomLeg	Attribut	Nom du légume
prixLeg	Attribut	Prix du légume selon les détails de vente (au kilo, barquette etc..)
quantiteLeg	Attribut	Quantité de ce légume dans les stocks
detailVenteLeg	Attribut	Attribut permettant de savoir si le légume est vendu au kilo, à la barquette, etc...
nomPhotoLeg	Attribut	Nom de la photo du légume dans le dossier de photos
descLegume	Attribut	Description du légume (culture, origine etc...)
idFamilleLeg	Attribut	Clé primaire de la table FamilleLegume, propre à chaque famille de légume, unique
nomFamilleLeg	Attribut	Nom de la famille de légume
idSaison	Attribut	Clé primaire de la table Saison, propre à chaque saison, unique
nomSaison	Attribut	Nom de la saison
idFruit	Attribut	Clé primaire de la table Fruit, propre à chaque fruit, unique
nomFruit	Attribut	Nom du fruit
prixFruit	Attribut	Prix du fruit selon les détails de vente (au kilo, barquette etc..)

detailVenteFruit	Attribut	Attribut permettant de savoir si le fruit est vendu au kilo, à la barquette, etc...
quantiteFruit	Attribut	Quantité de ce fruit dans les stocks
nomPhotoFruit	Attribut	Nom de la photo du fruit dans le dossier de photos
descFruit	Attribut	Description du fruit (culture, origine etc..)
idFamilleFruit	Attribut	Clé primaire de la table FamilleFruit, propre à chaque famille de fruit, unique
nomFamilleFruit	Attribut	Nom de la famille de fruit
idPanierHebdo	Attribut	Clé primaire de la table PanierHebdo, propre à chaque panier hebdomadaire, unique
prixPanierHebdo	Attribut	Prix du panier hebdomadaire
nbPers	Attribut	Nombre de personne vis-à-vis de la quantité de fruits et légumes du panier (ex : panier pour 8 personnes, nbPers = 8)
idRecette	Attribut	Clé primaire de la table RecetteHebdo, propre à chaque recette hebdomadaire, unique
nomRecette	Attribut	Nom de la recette hebdomadaire
tempsPrep	Attribut	Temps de préparation de la recette hebdomadaire
descIngr	Attribut	Description des ingrédients nécessaires
descRecette	Attribut	Description des étapes de la recette
nomPhotoRecette	Attribut	Nom de la photo de la recette dans le dossier de photos
idPanierClient	Attribut	Clé primaire de la table PanierClient, propre à chaque panier client, unique
idCommande	Attribut	Clé primaire de la table Commande, propre à chaque commande, unique
dateCom	Attribut	Date à laquelle le client à réaliser la commande
idTypeLiv	Attribut	Clé primaire de la table TypeLiv, propre à chaque type de livraison, unique
nomTypeLiv	Attribut	Nom du type de livraison (groupée, ponctuelle, etc..)
idLiv	Attribut	Clé primaire de la table Livraison, propre à chaque livraison, unique
adresseChoisie	Attribut	Adresse de livraison choisie
dateLiv	Attribut	Date estimée de livraison
idGroupeMail	Attribut	Clé primaire de la table GroupeMail, propre à chaque groupe de mail, unique
mail	Attribut	Mail de la personne
idMail	Attribut	Clé primaire de la table Mail, propre à chaque mail et unique
prixMinLiv	Paramètre	20€, représente le prix minimum du panier pour demander une livraison de fruits ou légumes

nbMaxGrp	Paramètre	10, nombre maximal de personne pour UNE commande groupée
jourLivPanier	Paramètre	Jour durant lequel le panier est livré chaque semaine. Ce jour est fixe, et vaut : <b>Samedi</b>
TVA	Paramètre	20%, taxe sur la valeur ajoutée
creaneauLivGrp	Paramètre	30 minutes, durée du créneau au cours duquel le livreur sera à l'adresse indiquée lors de la livraison groupée pour que les concernés viennent chercher leurs produits
nbCommandes	Calculable	Nombre de commande effectuées depuis la mise en production du site, information disponible uniquement pour le webmaster, obtenue en comptant le nombre de ligne de la table Commande
chiffreAffaire	Calculable	Argent total récolté par des ventes de marchandises depuis la mise en production du site, information disponible uniquement pour le webmaster, obtenue en faisant la somme du prixTotal de chaque commande (sans compter les coûts de productions, charges etc...)
nbClientInscrits	Calculable	Nombre de client ayant créé un compte depuis la mise en production du site, information disponible uniquement pour le webmaster, obtenue en comptant le nombre de ligne de la table Client
prixPanierTotal	Calculable	Prix total du panier, calculé en faisant la somme pour chaque produit du panier de : (quantité de ce produit * prix du produit)
prixComTotal	Calculable	Prix total de la commande, calculé en faisant la somme pour chaque produit de la commande de : (quantité de ce produit * prix du produit)

### III – Schéma relationnel et code SQL

#### Schéma relationnel :



### Code SQL :

/ ! \

*La table est créée sur **phpMyAdmin**. Il y a donc une syntaxe particulière, légèrement différente de celle que nous utilisons et les clés primaires / étrangères sont renseignées à la fin (voir fin code). /*

! \

#### Table Adresse :

```
CREATE TABLE `adresse` (  
  `idAdresse` int(11) NOT NULL,  
  `codePostal` varchar(10) NOT NULL,  
  `ville` varchar(50) NOT NULL,  
  `adresse` varchar(50) NOT NULL  
)
```

#### Table Client :

```
CREATE TABLE `client` (  
  `idClient` int(11) NOT NULL,  
  `nomClient` varchar(40) NOT NULL,  
  `prenomClient` varchar(40) NOT NULL,  
  `mdpClient` varchar(50) NOT NULL,  
  `envoiMail` tinyint(1) NOT NULL DEFAULT '0',  
  `hebdo` varchar(10) NOT NULL DEFAULT '0',  
  `telClient` varchar(10) NOT NULL,  
  `adresseCli` int(11) NOT NULL,  
  `mailCli` int(11) NOT NULL  
)
```

#### Table Commande :

```
CREATE TABLE `commande` (  
  `idCommande` int(11) NOT NULL,  
  `clientCom` int(11) NOT NULL,  
  `dateCom` date NOT NULL,  
  `livraisonCom` int(11) NOT NULL  
)
```

#### **Table FamilleFruit :**

```
CREATE TABLE `famillefruit` (  
  `idFamilleFruit` int(11) NOT NULL,  
  `nomFamilleFruit` varchar(40) NOT NULL  
)
```

#### **Table FamilleLegume :**

```
CREATE TABLE `famillelegume` (  
  `idFamilleLegume` int(11) NOT NULL,  
  `nomFamilleLeg` varchar(40) NOT NULL  
)
```

#### **Table Fruit :**

```
CREATE TABLE `fruit` (  
  `idFruit` int(11) NOT NULL,  
  `nomFruit` varchar(30) NOT NULL,  
  `prixFruit` int(11) NOT NULL,  
  `detailVenteFruit` varchar(30) NOT NULL,  
  `quantiteFruit` int(11) NOT NULL,  
  `nomPhotoFruit` varchar(30) NOT NULL,  
  `descFruit` text NOT NULL,  
  `saisonFruit` int(11) NOT NULL,  
  `familleFruit` int(11) NOT NULL
```



**Table GroupeMail :**

```
CREATE TABLE `groupemail` (  
  `idGroupeMail` int(11) NOT NULL  
)
```

**Table GroupeMailLiv :**

```
CREATE TABLE `groupemailliv` (  
  `idGroupe` int(11) NOT NULL,  
  `idMailG` int(11) NOT NULL  
)
```

**Table Legume :**

```
CREATE TABLE `legume` (  
  `idLegume` int(11) NOT NULL,  
  `nomLeg` varchar(30) NOT NULL,  
  `prixLeg` int(11) NOT NULL,  
  `quantiteLeg` int(11) NOT NULL,  
  `detailVenteLeg` varchar(30) NOT NULL,  
  `nomPhotoLeg` varchar(30) NOT NULL,  
  `descLeg` text NOT NULL,  
  `saisonLeg` int(11) NOT NULL,  
  `familleLeg` int(11) NOT NULL  
)
```

**Table LigneLegCommande :**

```
CREATE TABLE `lignelegcommande` (  
  `legumeLigne` int(11) NOT NULL,  
  `clientLigneLegCom` int(11) NOT NULL,  
  `commandeLegFruitCom` int(11) NOT NULL,  
  `quantiteLegCom` int(11) NOT NULL
```

**Table LigneFruitCommande :**

```
CREATE TABLE `lignefruitcommande` (  
  `fruitLigne` int(11) NOT NULL,  
  `clientLigneFruitCom` int(11) NOT NULL,  
  `commandeLigneFruitCom` int(11) NOT NULL,  
  `quantiteFruitCom` int(11) NOT NULL  
)
```

**Table LignePanierFruit :**

```
CREATE TABLE `lignepanierfruit` (  
  `fruitLigneP` int(11) NOT NULL,  
  `clientLigneP` int(11) NOT NULL,  
  `panierLigneP` int(11) NOT NULL,  
  `quantiteFruitP` int(11) NOT NULL  
)
```

**Table LignePanierLeg :**

```
CREATE TABLE `lignepanierleg` (  
  `legLigneP` int(11) NOT NULL,  
  `clientLigneLegP` int(11) NOT NULL,  
  `panierLigneLegP` int(11) NOT NULL,  
  `quantiteLegP` int(11) NOT NULL  
)
```

**Table Livraison :**

```
CREATE TABLE `livraison` (  
  `idLivraison` int(11) NOT NULL,  
  `adresseChoisie` int(11) NOT NULL,  
  `dateLivraison` date NOT NULL,  
  `groupeMailLiv` int(11) NOT NULL,
```

JALLU Thomas  
NELHOMME Lhukas  
MELUSSON Valentin  
`typeLiv` int(11) NOT NULL  
)

Projet tutoré S3

2020/2021

**Table Mail :**

```
CREATE TABLE `mail` (  
  `idMail` int(11) NOT NULL,  
  `mail` varchar(40) NOT NULL  
)
```

**Table PanierClient :**

```
CREATE TABLE `panierclient` (  
  `idPanierClient` int(11) NOT NULL,  
  `clientPanier` int(11) NOT NULL  
)
```

**Table PanierHebdo :**

```
CREATE TABLE `panierhebdo` (  
  `idPanierHebdo` int(11) NOT NULL,  
  `prixPanierHebdo` int(11) NOT NULL,  
  `nbPers` int(11) NOT NULL  
)
```

**Table RecetteHebdo :**

```
CREATE TABLE `recettehebdo` (  
  `idRecette` int(11) NOT NULL,  
  `nomRecette` varchar(50) NOT NULL,  
  `tempsPrep` int(11) NOT NULL,  
  `descIngr` text NOT NULL,  
  `descRecette` text NOT NULL,  
  `nomPhotoRecette` varchar(25) NOT NULL  
)
```

**Table Saison :**

```
CREATE TABLE `saison` (  
  `idSaison` int(11) NOT NULL,  
  `nomSaison` varchar(20) NOT NULL  
)
```

**Table TypeLivraison :**

```
CREATE TABLE `typelivraison` (  
  `idTypeLiv` int(11) NOT NULL,  
  `nomTypeLiv` varchar(30) NOT NULL  
)
```

**Déclaration des index :**

- ALTER TABLE `adresse`  
 ADD PRIMARY KEY (`idAdresse`);
- ALTER TABLE `client`  
 ADD PRIMARY KEY (`idClient`),  
 ADD KEY `mailClient` (`mailCli`),  
 ADD KEY `adresseCli` (`adresseCli`);
- ALTER TABLE `commande`  
 ADD PRIMARY KEY (`idCommande`, `clientCom`),  
 ADD KEY `clientCom` (`clientCom`),  
 ADD KEY `livraisonCom` (`livraisonCom`);
- ALTER TABLE `famillefruit`  
 ADD PRIMARY KEY (`idFamilleFruit`);
- ALTER TABLE `famillelegume`  
 ADD PRIMARY KEY (`idFamilleLegume`);

- ALTER TABLE `fruit`  
    ADD PRIMARY KEY (`idFruit`),  
    ADD KEY `familleFruit` (`familleFruit`),  
    ADD KEY `saisonFruit` (`saisonFruit`);
  
- ALTER TABLE `groupemail`  
    ADD PRIMARY KEY (`idGroupeMail`);
  
- ALTER TABLE `groupemailliv`  
    ADD PRIMARY KEY (`idGroupe`),  
    ADD KEY `idMailG` (`idMailG`);
  
- ALTER TABLE `legume`  
    ADD PRIMARY KEY (`idLegume`),  
    ADD KEY `familleLeg` (`familleLeg`),  
    ADD KEY `saisonLeg` (`saisonLeg`);
  
- ALTER TABLE `lignefruitcommande`  
    ADD PRIMARY KEY (`fruitLigne`, `clientLigneFruitCom`, `commandeLigneFruitCom`),  
    ADD KEY `clientLigneFruitCom` (`clientLigneFruitCom`),  
    ADD KEY `commandeLigneFruitCom` (`commandeLigneFruitCom`);
  
- ALTER TABLE `lignelegcommande`  
    ADD PRIMARY KEY (`legumeLigne`, `clientLigneLegCom`, `commandeLegFruitCom`),  
    ADD KEY `clientLigneLegCom` (`clientLigneLegCom`),  
    ADD KEY `commandeLegFruitCom` (`commandeLegFruitCom`);
  
- ALTER TABLE `lignepanierfruit`

- ```
ADD PRIMARY KEY (`fruitLigneP`,`clientLigneP`,`panierLigneP`),  
ADD KEY `clientLigneP` (`clientLigneP`),  
ADD KEY `panierLigneP` (`panierLigneP`);
```
- ALTER TABLE `lignepanierleg`  

```
ADD PRIMARY KEY (`legLigneP`,`clientLigneLegP`,`panierLigneLegP`),  
ADD KEY `clientLigneLegP` (`clientLigneLegP`),  
ADD KEY `panierLigneLegP` (`panierLigneLegP`);
```
  - ALTER TABLE `livraison`  

```
ADD PRIMARY KEY (`idLivraison`),  
ADD KEY `typeLiv` (`typeLiv`),  
ADD KEY `groupeMailLiv` (`groupeMailLiv`);
```
  - ALTER TABLE `mail`  

```
ADD PRIMARY KEY (`idMail`);
```
  - ALTER TABLE `panierclient`  

```
ADD PRIMARY KEY (`idPanierClient`,`clientPanier`),  
ADD KEY `clientPanier` (`clientPanier`);
```
  - ALTER TABLE `panierhebdo`  

```
ADD PRIMARY KEY (`idPanierHebdo`);
```
  - ALTER TABLE `recettehebdo`  

```
ADD PRIMARY KEY (`idRecette`);
```
  - ALTER TABLE `saison`  

```
ADD PRIMARY KEY (`idSaison`);
```

```
- ALTER TABLE `typelivraison`  
  
    ADD PRIMARY KEY (`idTypeLiv`);
```

### CODE SQL (intégré au PHP) :

Ci-dessous, vous trouverez un florilège des différentes lignes de code SQL intégré au PHP utilisé pour le site. Tout n'est pas présent car le site n'est pas encore complet mais les principales requêtes utilisées sont présentes.

```
SELECT idFamilleFruit FROM famillefruit WHERE nomFamilleFruit = '$nomfamille'  
SELECT idSaison FROM saison WHERE nomSaison = '$nomsaison'  
INSERT INTO Fruit VALUES (null, '$nom', $prix, '$detail', $stockI, '$nomPh',  
null, '$saison', '$famille')  
SELECT idFamilleLegume FROM famillelegume WHERE nomFamilleLeg = '$nomfamille'  
SELECT idSaison FROM saison WHERE nomSaison = '$nomsaison'  
INSERT INTO Legume VALUES (null, '$nom', $prix, $stockI, '$detail', '$nomPh',  
null, '$saison', '$famille')  
$mail = mysqli_real_escape_string($co, $mail);  
SELECT idMail FROM mail WHERE mail = '$mail'  
SELECT mdpClient FROM client WHERE mailCli = '$idmail'  
SELECT * FROM fruit  
SELECT * FROM client JOIN mail ON (mailCli = idMail) WHERE mail = '$mail'  
INSERT INTO mail VALUES (null, '$mail')  
INSERT INTO adresse VALUES (null, '$cp', '$ville', '$addr')  
INSERT INTO client VALUES (null, '$nom', '$prenom', '$mdp', '$choix', '$hebdo', '$tel',  
'$idAddr', '$idmail')  
SELECT * FROM legume  
UPDATE mail SET mail = '$mail' WHERE mail.idMail = $idcli  
DELETE FROM Fruit WHERE nomFruit = '$nom'  
DELETE FROM Legume WHERE nomLeg = '$nom'  
SELECT quantiteFruit, idFruit FROM Fruit WHERE nomFruit = '$nom'  
UPDATE Fruit SET quantiteFruit = '$newQuant' WHERE idFruit = '$id'  
SELECT quantiteLeg FROM Legume WHERE nomLeg = '$nom'  
UPDATE Legume SET quantiteLeg = '$newQuant' WHERE nomLeg = '$nom'  
SELECT nomClient, prenomClient FROM client WHERE envoiMail = true
```

## STOCKAGE MOT DE PASSE :

Le mot de passe doit être stocké de manière sécurisée. Pour cela, nous avons choisis la méthode du Hash and Salt. Cela consiste à utiliser une fonction de Hachage d'une chaîne de caractère, en ajoutant un cette chaîne un « grain de sel » c'est-à-dire une autre chaîne confectionnée par nos soins et que seuls nous détenons. Ainsi, le mot de passe dans la base de données n'est qu'une image du vrai mot de passe, que seul l'utilisateur connaîtra.

Fonction de hachage : **php md5**

Le grain de sel restera secret pour la sécurité de nos utilisateurs.

## Exemple de données insérées :

|                          |                         | idClient | nomClient | prenomClient | mdpClient                        | envoiMail | hebd  | telClient  | adresseCli | mailCli |
|--------------------------|-------------------------|----------|-----------|--------------|----------------------------------|-----------|-------|------------|------------|---------|
| <input type="checkbox"/> | Éditer Copier Supprimer | 4        | Melusson  | Valentin     | 003da4a9e2da2bba2c98f723c4b99924 | 1         | moyen | 0646221047 | 2          | 9       |
| <input type="checkbox"/> | Éditer Copier Supprimer | 5        | Jallu     | Thomas       | b6fbc8fbb104ed7f118fe11d6e0a4e3  | 1         |       | 0641475895 | 4          | 11      |
| <input type="checkbox"/> | Éditer Copier Supprimer | 6        | Nelhomme  | Lhukas       | 7a361e1af69473f54dd6baf4cb51f860 | 1         |       | 0655864412 | 6          | 13      |
| <input type="checkbox"/> | Éditer Copier Supprimer | 8        | Nelhomme  | Severine     | 171348e6c119826593fbcde48186ed19 | 1         |       | 0645552147 | 8          | 15      |

|                          |                         | idFruit | nomFruit                  | prixFruit | detailVenteFruit      | quantiteFruit | nomPhotoFruit   | descFruit | saisonFruit | familleFruit |
|--------------------------|-------------------------|---------|---------------------------|-----------|-----------------------|---------------|-----------------|-----------|-------------|--------------|
| <input type="checkbox"/> | Éditer Copier Supprimer | 2       | Pomme Elstar du verger    | 4         | au kilo               | 130           | pommes.jpg      |           | 1           | 1            |
| <input type="checkbox"/> | Éditer Copier Supprimer | 3       | Poires Williams du verger | 6         | au kilo               | 100           | poires.jpg      |           | 1           | 1            |
| <input type="checkbox"/> | Éditer Copier Supprimer | 4       | Clementines bio           | 8         | au kilo               | 100           | clementines.jpg |           | 1           | 1            |
| <input type="checkbox"/> | Éditer Copier Supprimer | 5       | Kiwi bio                  | 7         | par barquette de 6    | 130           | kiwi.jpg        |           | 1           | 1            |
| <input type="checkbox"/> | Éditer Copier Supprimer | 6       | Fraises bio               | 8         | par barquette de 12   | 100           | fraise.jpg      |           | 1           | 1            |
| <input type="checkbox"/> | Éditer Copier Supprimer | 7       | Myrtilles bio             | 11        | par barquette de 500g | 100           | myrtilles.jpg   |           | 1           | 1            |
| <input type="checkbox"/> | Éditer Copier Supprimer | 8       | Raisin sans pepins bio    | 10        | au kilo               | 100           | raisin.jpg      |           | 1           | 1            |
| <input type="checkbox"/> | Éditer Copier Supprimer | 9       | Framboises bio            | 8         | par barquette de 500g | 100           | framboises.jpg  |           | 1           | 1            |

|                          |                         | idLegume | nomLeg           | prixLeg | quantiteLeg | detailVenteLeg | nomPhotoLeg    | descLeg | saisonLeg | familleLeg |
|--------------------------|-------------------------|----------|------------------|---------|-------------|----------------|----------------|---------|-----------|------------|
| <input type="checkbox"/> | Éditer Copier Supprimer | 1        | Carottes bio     | 6       | 120         | au kilo        | carottes.jpg   |         | 1         | 1          |
| <input type="checkbox"/> | Éditer Copier Supprimer | 2        | Courgettes bio   | 6       | 100         | au kilo        | courgettes.jpg |         | 1         | 1          |
| <input type="checkbox"/> | Éditer Copier Supprimer | 3        | Patates douces   | 4       | 100         | au kilo        | pdouces.jpg    |         | 1         | 1          |
| <input type="checkbox"/> | Éditer Copier Supprimer | 4        | Coeur de boeuf   | 7       | 100         | au kilo        | cdeboeuf.jpg   |         | 1         | 1          |
| <input type="checkbox"/> | Éditer Copier Supprimer | 5        | Pommes de terres | 3       | 100         | au kilo        | patates.jpg    |         | 1         | 1          |
| <input type="checkbox"/> | Éditer Copier Supprimer | 6        | Poireaux bio     | 9       | 100         | au kilo        | poireaux.jpg   |         | 1         | 1          |
| <input type="checkbox"/> | Éditer Copier Supprimer | 7        | Navets bio       | 11      | 100         | au kilo        | navets.jpg     |         | 1         | 1          |
| <input type="checkbox"/> | Éditer Copier Supprimer | 8        | Choux fleurs     | 6       | 100         | au kilo        | choufleurs.jpg |         | 1         | 1          |

## IV – Documentation technique du produit

### Manuel d'utilisation :

Pour installer la base de données, différents moyens sont possibles.

Tout d'abord, il est possible de réaliser l'import via des lignes de code. En effet, le code entier de la base de données comprenant également l'initialisation des index est donné un peu plus haut dans ce rapport. Il vous suffit d'exécuter ce code et vous obtiendrez la base de données.

Il est également possible de le faire via l'import de l'outil phpMyAdmin. Pour cela, il suffit de suivre ces quelques étapes :

1. Ouvrez **phpMyAdmin**.



2. Cliquez sur le nom de la base de données qui recevra les informations importées. La page se rafraîchira pour afficher les informations relatives à la base de données sélectionnée.
3. Cliquez sur l'onglet « Importer ».
4. Cliquez sur le bouton « Afficher » de la section « Fichier à importer » (File to import).
5. Votre navigateur vous invitera à repérer sur votre ordinateur le fichier de la base de données.
6. Sélectionner le fichier SQL fourni, puis cliquez sur le bouton « Exécuter ».

Vous aurez ainsi la base de données installée et fonctionnelle, possédant des caractéristiques identiques à la base de données que nous utilisons.

En ce qui concerne la maintenance, deux méthodes sont également possibles. En effet, s'il est nécessaire d'apporter des modifications à la base pour conserver sa stabilité il est possible via phpMyAdmin de mettre à jour les caractéristiques de la base par des lignes de code SQL. Ainsi, il est possible de garantir l'intégrité de la base de données de manière manuelle.

Il est enfin possible de faire des modifications directement grâce aux interfaces que propose l'outil phpMyAdmin. Il vous suffit alors de choisir la donnée, table ou autre à modifier et à suivre les étapes indiquées par le système.

## V - Conclusion

Pour conclure, on peut dire que la base de données est assez bien faite, et ne présente pas d'erreurs qui ne puissent être réglées. Ce constat est évidemment valable à l'heure actuelle et il n'est pas possible de prévoir si elle rencontrera des problèmes par la suite.

Ce projet a été très enrichissant, à différentes échelles. Tout d'abord car il représente la concrétisation du module de base de données avancées du troisième semestre en nous permettant d'appliquer les acquis à une situation qui se rapproche du monde professionnel. De plus, il nous permet de nous confronter à des problèmes qui ne peuvent être appréhendés théoriquement, et nous oblige donc à apprendre à surmonter des problèmes sans qu'il n'y ait une solution prédéfinie, à être réactif et à construire la base de données intelligemment pour qu'elle soit stable.

Les problèmes que nous avons rencontrés sont par exemple le fait que le modèle théorique que nous avons construit de la base ne permettait pas, une fois implémenté, de répondre aux attentes de l'application web. Il a alors fallu modifier certaines caractéristiques sans pour autant rendre obsolète la base dans son ensemble.

Nous avons également remarqué que, après s'être réparti les tâches entre membres de l'équipe projet, nous nous retrouvions confrontés à des imprévus qui forçaient chacun des membres à modifier certaines données de la base. Il a alors fallu mettre en commun ces modifications tout en conservant l'harmonie de la base.

Ce projet a donc permis de faire la passerelle entre ce que nous apprenons en cours et la pratique réelle, de ne pas être démunis face à des imprévus et de pouvoir construire une base de données dont les caractéristiques dépendent des objectifs et non d'un pattern prédéfini.