

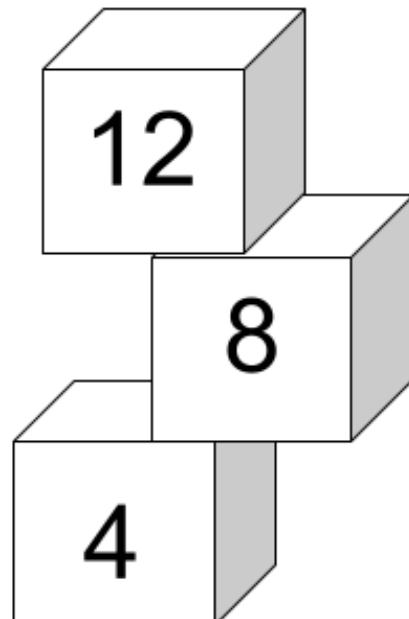
2020/2021

Afpa - DWWM

Arnaud BOISJARDIN

# Sugar Meter

## Projet tutoré





## Table des matières

---

<b>Introduction</b>	<b>4</b>
<b>I - Présentation du projet</b>	<b>5</b>
Contexte du projet	5
Cahier des charges	5
Charte graphique	5
<b>II - Environnement de travail</b>	<b>6</b>
Organisation : Agile	6
Outils de communication	7
Outils de développement	7
<b>III - Développer la partie front-end d'une application web ou web mobile</b>	<b>7</b>
a. Maquetter une application	8
Logigramme de notre application	9
User story	10
User Case	10
Inventaire du contenu Hiérarchie visuelle ( primaire, secondaire tertiaire)	11
Maquette	12
b. Réaliser une interface utilisateur web statique et adaptable	16
Template	16
Accueil	17
Inscription	19
c. Développer une interface utilisateur web dynamique	20
Enrichir notre base de données	20
Logigramme du code JavaScript	21
Affichage	23
d. Réaliser une interface utilisateur avec une solution de gestion de contenu	24
Connexion	24
Recherche et ajout à la consommation	25
Gestion de mes produits consommés dans la journée	26
<b>IV - Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité</b>	<b>27</b>
a. Créer une base de données	27
MCD (modèle conceptuel de données)	27
MLD (modèle logique de données)	28
Code SQL (Structured Query Language)	29
SGBD (système de gestion de base de données)	30
Relations (type InnoDB)	30
b. Développer les composants d'accès aux données	31
PDO (PHP Data Objects)	31



c. Développer la partie back-end d'une application web ou web mobile	32
Inscription	32
Désinscription	32
Connexion	33
Déconnexion	33
d. Élaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce	35
<b>Conclusion</b>	<b>38</b>
<b>Annexes</b>	<b>39</b>
CNIL Réglementation CNIL 1er octobre 2020, Cookies et traceurs	39
Sécurité	40
Google présente l'avancée de son projet « Privacy Sandbox »	41
Internet en 2021 : les 10 chiffres clés	42
Le mobile en 2021 : les 10 chiffres clés	43



## Introduction

---

Dans le cadre de ma formation Développement Web et Web Mobile (DWWM,) j'ai réalisé un projet tutoré de fin de formation, durant lequel j'ai pu suivre les différentes étapes du développement d'une application web (mobile-first donc gestion du responsive et de l'architecture d'amélioration progressive).

C'est un projet de sensibilisation à la nutrition qui vise à mettre en place un site qui permet (via l'API « Open Food Facts ») de récupérer, grâce à son code barre, la quantité de sucre d'un produit, puis de calculer dynamiquement le nombre de grammes de sucre ingérés en fonction de la quantité de produit consommé.

Un système d'inscription et d'identification est nécessaire afin de pouvoir cumuler ses consommations sur une journée, ou plus (statistiques), et ainsi de maîtriser sa consommation de sucre.

L'organisation du projet se fait selon une démarche Agile, on établit un product backlog que l'on divise en sprint hebdomadaire. Ce projet couvre la partie Front-end et la partie Back-end ainsi que la composante accès aux données. Il a été réalisé en PHP/MySQL, HTML/CSS/JS/AJAX/Bootstrap/JQuery. afin de respecter le cahier des charges.



## I - Présentation du projet

---

### Contexte du projet

Réaliser une application web mobile de sensibilisation à la nutrition destinée à connaître la quantité de sucre consommée quotidiennement.

### Cahier des charges

*Sugar Meter* consiste en l'affichage de la quantité de sucre consommée quotidiennement, au travers de produits alimentaires industriels, sous la forme explicite de nombres de morceaux de sucre. Afin de connaître la quantité de sucre ingérée chaque jour, l'utilisateur procède selon les étapes suivantes :

1. Ajout d'un produit alimentaire dans son espace personnel « garde manger »
2. Sélection d'un produit de son «garde manger »
3. Indication de la quantité de produit consommée
4. Consultation de la quantité totale de sucre absorbée à un instant de la journée
5. Consultation de la quantité totale de sucre consommée sur une journée écoulée

*Sugar Meter* sera développée sous la forme d'une application web mobile destinée à être utilisée principalement sur smartphones. L'expérience utilisateur devra donc être adaptée à ce type d'appareils.

Il est important de noter que la composition de la majorité des aliments industriels est déjà référencée dans la base de données communautaire [Open Food Facts](#). Toutes les données nutritionnelles utilisées par *Sugar Meter* devront donc être issues de cette base de données de référence.

### Charte graphique

L'ergonomie de *Sugar Meter* devra être dans l'esprit des applications smartphones natives. Ainsi, l'interface graphique et la palette couleurs devront être proches des recommandations « Flat design » et la proposition inhérente [Material design](#) de Google. Afin de correspondre au contexte de la nutrition, la palette couleurs sera dans des teintes vives vertes ou bleues.



## II - Environnement de travail

### Organisation : Agile

La réalisation *Sugar Meter* sera menée en démarche projet « agile ».

Le projet tutoré c'est organiser en 7 sprints de 1 semaine

Les fonctionnalités souhaitées seront donc définies et priorisées dans un tableau « product backlog » qui sera mis à jour et complété régulièrement.

Le développement des fonctionnalités s'effectuera selon un rythme périodique « sprint » hebdomadaire. Pour assurer le suivi de ces développements, les développeurs établiront un tableau collaboratif « sprint backlog ».

Afin d'assurer le suivi de la réalisation en télétravail, 2 rendez-vous d'équipe hebdomadaires seront organisés :

- Atelier technique : rendez-vous informel destiné à discuter des solutions techniques envisagées et des difficultés techniques rencontrées
- Revue de projet : rendez-vous formel destiné à faire un point :
  - collectif sur l'achèvement et l'ajout des fonctionnalités dans le « product backlog », mais aussi de revoir la priorisation
  - sur l'avancement des tâches du « sprint backlog » lors d'un « tour de table » : tâches faites, en cours et restant. Attributions des nouvelles tâches.



Trello





## Outils de communication

**Plateforme Métis Afpa** (échange avec le formateur)



**Discord** (étudiants de la formation)



**Git** (dépôt document de la formation)



## Outils de développement

*Sugar Meter* devra reposer sur les technologies web traditionnelles et l'architecture « open source » la plus communément disponible chez les hébergeurs web :

- **Frontend : compatibilité aux normes HTML 5 / ECMAScript 2018 / CSS 3**

- **Backend : compatibilité à l'architecture serveurs PHP 7.4 / MySQL 5.7**

Le navigateur ciblé sera « Google Chrome mobile », dans sa dernière version, car il est le plus utilisé actuellement sur les smartphones.

**Wampserver** : Environnement Serveur -> PHP, MySQL, Apache.

**Windows 10 / Visual studio code** : outil de développement (EDI).

**Google Drive / Dropbox / Git** : outils google et stockage.

**Moqups** : maquettage de l'application.

**Looping** : logiciel de modélisation conceptuelle de données (MCD/MLD).



### III - Développer la partie front-end d'une application web ou web mobile

#### a. Maquetter une application

##### Mobile-first (amélioration progressive).

La version mobile (responsive) doit être pensée dès les premières phases du design ainsi que de l'interface mobile et d'adapter l'affichage pour grands écrans ensuite. Le plus petit des designs aura seulement les caractéristiques essentielles.

##### Desktop (dégradation progressive).

Elle intègre toutes les complexités au début, puis enlève une à une lors de la conception pour des appareils plus petits. Le problème est que le noyau et les éléments complémentaires fusionnent et deviennent difficiles à distinguer et à séparer.

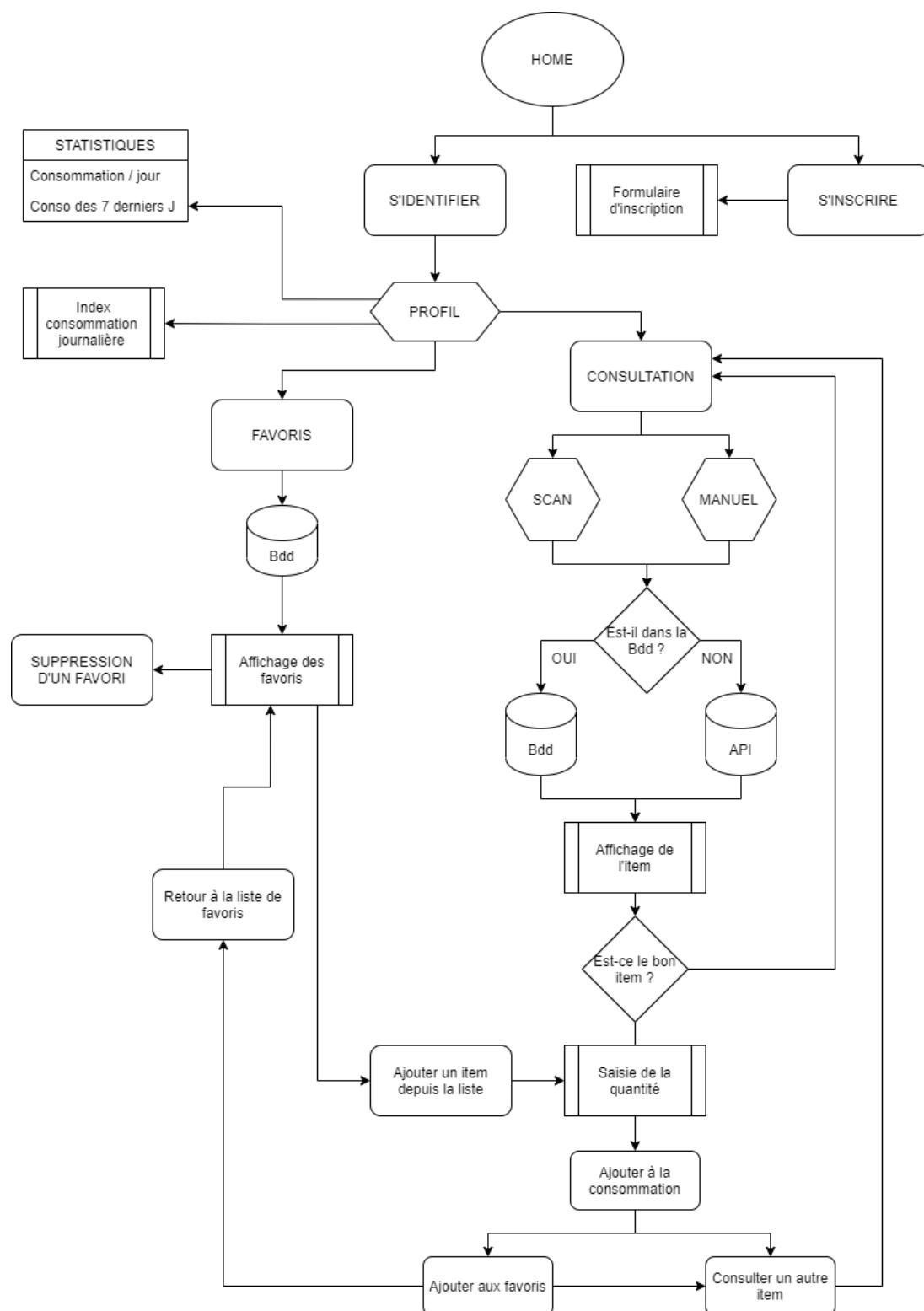
##### Mobile-First est synonyme de Content-First

Le mobile a de nombreuses limites comme la taille de l'écran ou la taille de la bande passante. Concevoir quelque chose en prenant en compte ces paramètres vous oblige à établir des priorités pour votre contenu et donc centré sur l'utilisateur.

- Construire la maquette mobile d'abord, puis l'utiliser comme modèle pour les tailles plus élevées.
- Agrandir les cibles tactiles, Apple recommande un carré de 44 pixels.
- Ne pas compter sur les survols
- Pensez «application», le menu off-canvas, des widgets extensibles, des appels AJAX ou d'autres éléments sur l'écran avec lesquels les utilisateurs peuvent interagir sans rafraîchir la page.
- Évitez les grands graphiques, des images qui sont lisibles sur leurs écrans de poche.



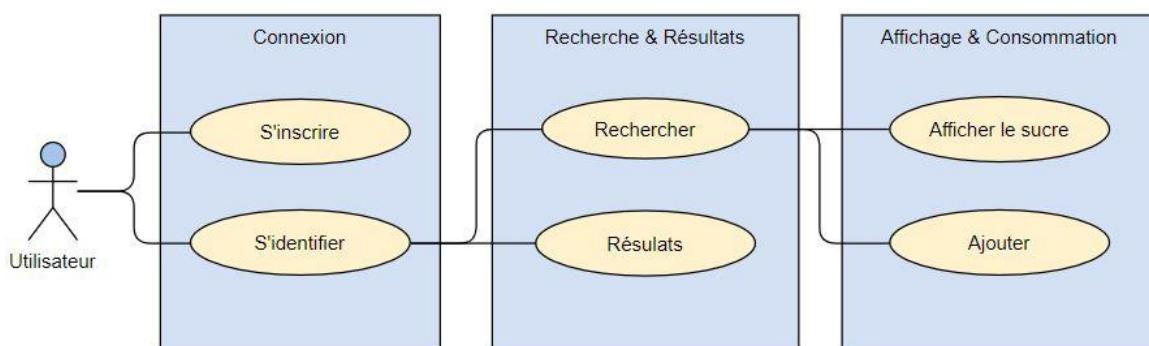
## Logigramme de notre application



## User story

ID	PERSONA	FONCTIONNALITE	MoSCoW	Effort	Business value
1	Romane	S'inscrire sur l'application web	S	L	90
2	Kévin	Choisir un produit alimentaire d'après le numéro de son code barre	M	M	80
3	Romane	Avoir une photo d'un produit alimentaire choisi	C	S	40
4	Kévin	Récupérer la quantité de sucre, en pourcentage, d'un produit alimentaire depuis Open Food Facts	M	S	70
5	Romane	Ajouter un produit alimentaire à une liste de favoris	C	M	50
6	Jean-Pierre	Obtenir des statistiques sur un utilisateur et son « garde manger »	C	M	40
7	Romane	Supprimer un produit alimentaire de la liste des favoris	C	M	40
8	Jean-Pierre	Obtenir des statistiques de consultation de l'application web	S	L	50
9	Jean-Pierre	Sauvegarder les comptes et préférences utilisateurs	S	M	70
10	Kévin	Connaître la quantité de sucre consommée sur une journée entière	M	S	70
11	Romane	Suivre l'évolution journalière de sa consommation de sucre (période de 7 jours depuis hier)	C	M	60
12	Romane	Connaître la quantité de sucre consommée depuis le début de la journée	C	L	70
13	Kévin	Saisir une quantité d'un produit alimentaire et afficher la quantité de sucre correspondante	M	M	70
14	Kévin	Convertir une quantité de sucre en nombre de morceaux de sucre	S	S	40
15	Romane	Sélectionner un produit alimentaire dans ses favoris	C	M	30
16	Romane	Se désinscrire de l'application web	S	M	40
17	Jean-Pierre	Afficher un message de consentement conforme à la CNIL	M	L	60
18	Romane	Avoir une alerte lors du dépassement d'un seuil de consommation de sucre	W	S	30
19	Kévin	Trier les produits du « garde manger » d'après leur utilisation décroissante	C	S	40
20	Kévin	Afficher une présentation et une note d'utilisation de l'application web aux utilisateurs non inscrits	S	M	40
21	Romane	Scanner le code barre d'un produit alimentaire avec son smartphone	M	L	30
22	Jean-Pierre	Disposer de l'historique des inscriptions / désinscriptions des utilisateurs	S	M	30
23	Kévin	Avoir un bon référencement naturel sur Google quant aux outils nutritionnels	M	M	80
24	Romane	Visualiser la quantité de sucre qu'elle s'apprête à manger sans l'ajouter obligatoirement à sa consommation journalière	S	S	70
25	Kévin	S'identifier sur l'application	S	M	90

## User Case



## Inventaire du contenu Hiérarchie visuelle ( primaire, secondaire tertiaire)

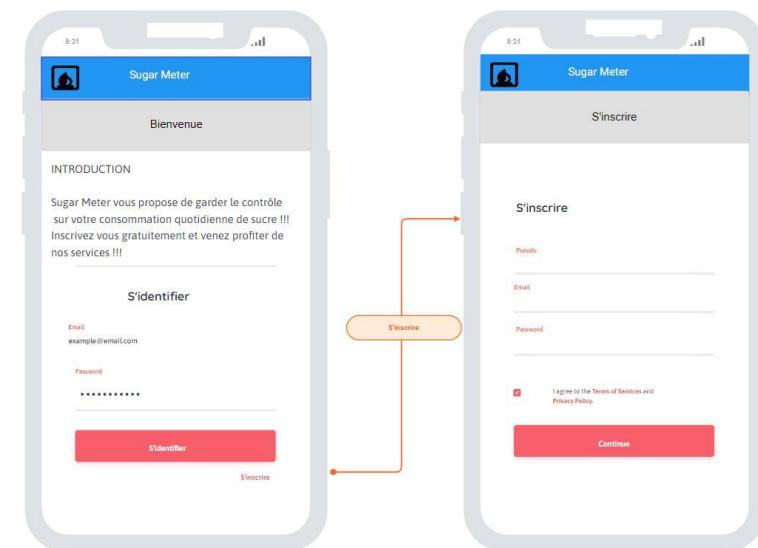
Inventaire du contenu		Hiérarchie visuelle	
User story	Éléments	Mobile	Desktop
User story	Logo	Secondaire	Secondaire
	Nom du site	Secondaire	Secondaire
	Mon profil	Secondaire	Secondaire
	Mes consommations	Secondaire	Secondaire
	Mes statistiques	Secondaire	Secondaire
	Déconnexion	Secondaire	Secondaire
Navigation	Cnil cookie	Primaire	Primaire
	Jumbotron info	-	Primaire
	Information alimentation	Secondaire	Secondaire
	Principe du site	Primaire	Primaire
Informer l'utilisateur	Pseudo	Primaire	Primaire
	E-mail	Primaire	Primaire
	Password	Primaire	Primaire
	Password retype	Primaire	Primaire
S'inscrire	E-mail	Primaire	Primaire
	Password	Primaire	Primaire
Se désinscrire	Modal de confirmation	primaire	primaire
	Code Barre	Primaire	Primaire
Rechercher un produit	Quantité consommée	Primaire	Primaire
	Nom	Primaire	Primaire
	Image	Primaire	Primaire
	Quantité de sucre en gramme	Primaire	Primaire
Afficher le résultat	Nombre de morceau de sucre	Primaire	Primaire
	Ajouter à sa consommation	Primaire	Primaire
Afficher la quantité de sucre déjà consommé	Quantité de sucre en g du jour	Secondaire	Secondaire
	Nom	Primaire	Primaire
Liste des produits consommés dans la journée	Image	Primaire	Primaire
	Quantité de produit en gramme	-	Secondaire
	Quantité de sucre ingérée	Primaire	Primaire
	Quantité de sucre 7 derniers jours	Primaire	Primaire
Statistique			





## Maquette

On commence donc par la taille la plus petite.



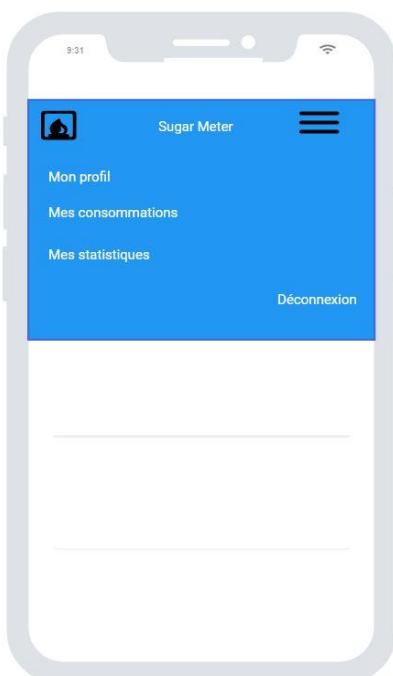
### Page d'accueil

- Présentation site
- S'identifier

### Page d'inscription

- Pseudo
- Email
- Password

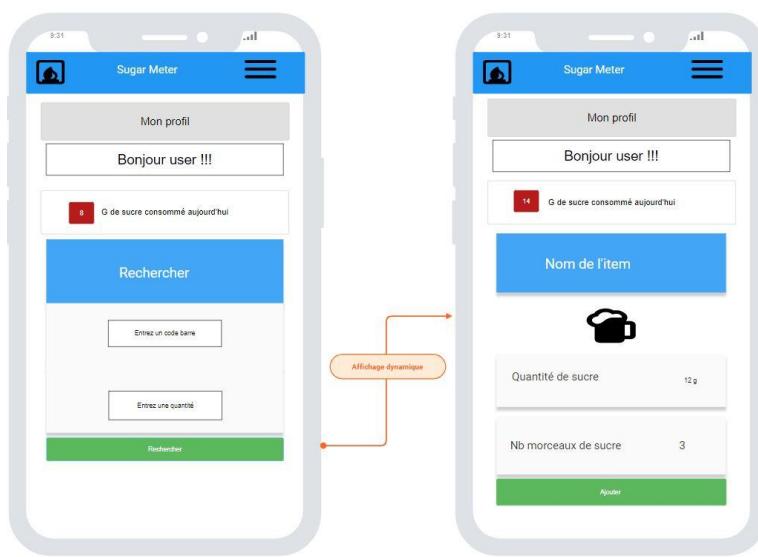
**Une fois connecté, nous sommes redirigés vers le profil du user et avons accès au menu de navigation.**



- **Logo & nom du site :** Lien retour page d'accueil
- **Navbar toggle icon :** Navbar format mobile
- **Navigation :**
  - Mon profil
  - Mes consommations
  - Mes statistiques
- **Déconnexion :** Clôture la session et redirige vers la page d'accueil



## Profil / rechercher un produit / l'ajouter à sa consommation



### Mon Profil

- Accès navbar toggle
- Message bienvenue user
- Compteur sucre / jour

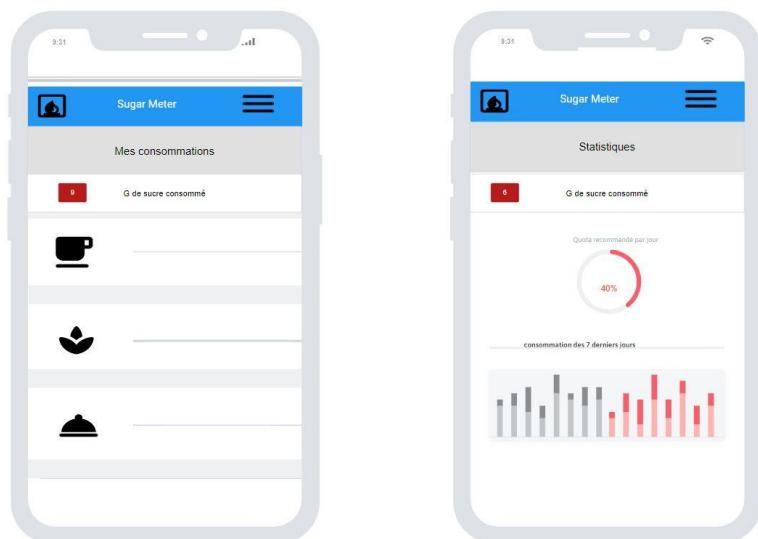
### Form recherche item

- Saisie code barre
- Saisie quantité consommée

### Résultat recherche

- Nom item
- Image
- Quantité de sucre
- Nb morceaux de sucre
- Ajouter à sa consommation

## Mes consommations / Mes statistiques



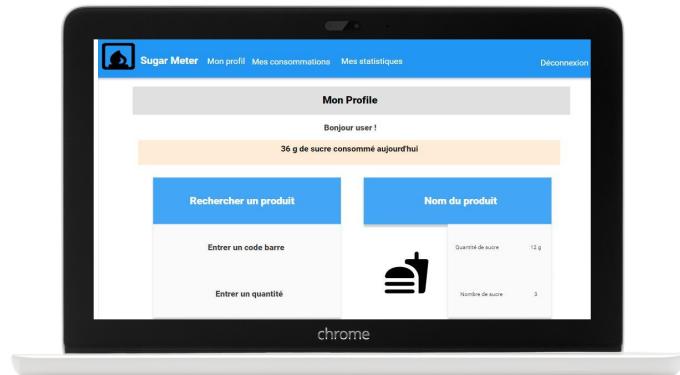
### Mes consommations

- total sucre du jour
- liste items consommés du jour

### Mes statistiques

- total sucre du jour
- total sucre des consommations des 7 derniers jours

## Affichage desktop



## Structure / MVC (Model-View-Controller)

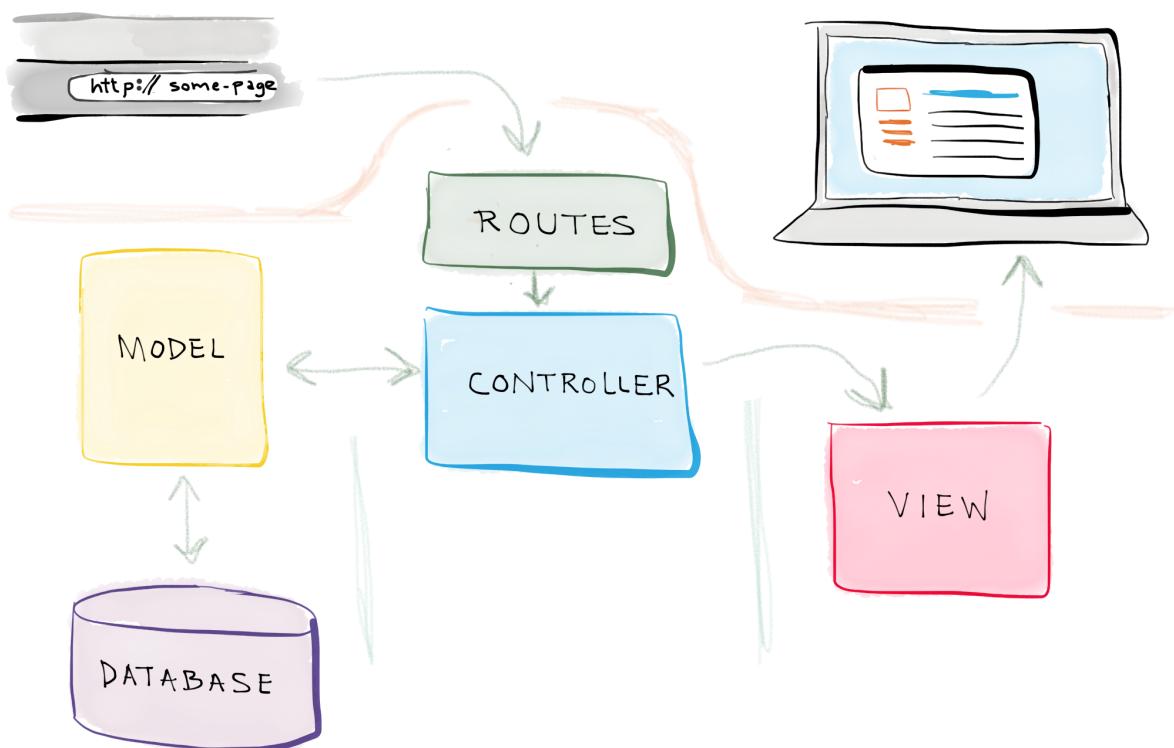
C'est un style d'architecture très populaire permettant d'être efficace et structuré lors du développement d'un projet.

Il permet une réutilisation de code, une plus grande souplesse pour le développement du site entre les différents développeurs ainsi qu'un gain de temps de maintenance et d'évolution du site.

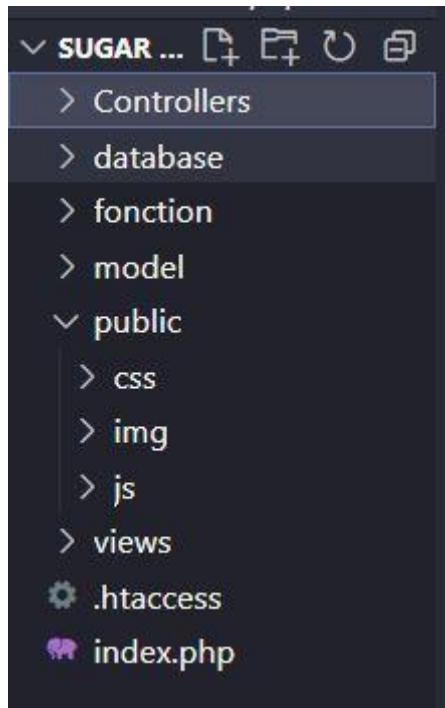
**Le M, modèle (model)** représente les données et ne fait rien d'autre, il ne dépend donc pas des vues et des contrôleurs.

**Le V, vue (view)** affiche les données que le modèle contient à l'utilisateur, ex : formulaire, bouton, etc... (Un modèle peut contenir plusieurs vues)

**Le C, contrôleur (controller)** traite les données de l'utilisateur, il dépend et interagit avec le modèle ainsi que la vue (Dans certains cas il est possible que la vue & le contrôleur soient le même objet).



## Répertoire



Fichier htaccess

```
.htaccess
1 RewriteEngine On
2
3 RewriteCond %{REQUEST_FILENAME} !-f
4 RewriteCond %{REQUEST_FILENAME} !-d
5
6 RewriteRule ^(.*)$ index.php?page=$1
7
8
```

J'utilise le fichier index pour router  
Le fichier htaccess me permet de définir des pretty URL (SEO)

```
/*
 * @author      Arnaud BOISJARDIN
 * @version     v.1.0.0
 * @copyright   Copyright (c) 2021, AB
 */
require_once "Controllers/consommationController.php";
$ConsommationController = new ConsommationController;

require_once "Controllers/utilisateurController.php";
$utilisateurController = new UtilisateurController;

define("URL", str_replace("index.php", "", (isset($_SERVER['HTTPS']) ? "https" : "http") .
"://$_SERVER[HTTP_HOST]$_SERVER[PHP_SELF]"));

try {
    if (empty($_GET['page'])) {
        require "views/accueil.php";
    } else {

        $url = explode("/", filter_var($_GET['page']), FILTER_SANITIZE_URL);

        switch ($url[0]) {

            case "accueil":
                $utilisateurController->afficherConnexion(); //ok
                break;

            case "connexion_traitement":
                $utilisateurController->connexionTraitement(); //ok
                break;

            case "deconnexion":
                $utilisateurController->deconnexion(); //ok
                break;
        }
    }
}
```



## b. Réaliser une interface utilisateur web statique et adaptable

### Template

- Éviter la répétition de code
- Charte graphique du site homogène
- Faciliter toutes modifications
- Search Engine Optimisation

```

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sugar Meter - <?= $title ?></title>
    <link rel="icon" type="image/png" href="public/img/favLogoSugar.png" />
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
        integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">
    <link type="text/css" rel="stylesheet" href="public/css/sugar.css">
</head>

> <body class="bgBody"> ...
</body>

</html>

```

```

<nav class="navbar sticky-top navbar-light shadow-sm navbar-expand-lg gradRb shadow">
    <a href="#">
        
    </a>
    <a class="navbar-brand font-weight-bolder ml-2" href="#"><strong> Sugar Meter</strong></a>
    <?php if (Securite::verifAccessSession()) : ?>
        <!-- Partie du menu accessible depuis une session ouverte-->
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavAltMarkup"
            aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
            <div class="navbar-nav">
                <a class="nav-item nav-link active" href="#"><strong>Mon profil</strong><span class="sr-only" style="font-size: small;">(current)</span></a>
                <a class="nav-item nav-link" href="#"><strong>Mes consommations</strong></a>
                <a class="nav-item nav-link" href="#"><strong>Mes statistiques</strong></a>
            </div>
            <a href="deconnexion" class="btn btn-rounded text-dark float-right ml-auto connect">Déconnexion</a>
        </div>
    <?php endif; ?>
</nav>

```

```

<main class="contener">
    <h1 class="rounded border bg-light text-center border-dark p-2 m-2"><?= $title ?></h1>
    <?= $content ?>
</main>
<!-- Footer -->
<footer class="text-center fixed-bottom gradRbTop">
    <!-- Copyright -->
    <div class="text-center p-1" style="background-color: rgba(0, 0, 0, 0.2);>
        © 2021 Copyright : AkB
    </div>
</footer>
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
    integrity="sha384-KJ3o2DKtIkVYIK3UENzm7KCKr/rE9/Pqg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
    integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibx39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
    integrity="sha384-JZR6Spejh4U02d8j0t6vLEHfe/JQGIRRSQQxSffWp11MqudAyjuar5+76PVCmYl" crossorigin="anonymous"></script>
<!-- JavaScript -->
<script type="" src="public/js/sugar.js"></script>
</body>
</html>

```





## Accueil

### Gestion du responsive

- Utilisation des breakpoints de bootstrap, pour un affichage optimal.
- J'ai défini un seul breakpoint pour passer de une colonne à deux avec : col-lg  $\geq$ 992px.
- Le contenu est en une colonne en version mobile et en 2 colonnes pour le desktop.

**Le Jumbotron n'est visible que dans la version d'écran : col-md  $\geq$ 768px.**

```
<?php ob_start() ?>

> <section class="jumbotron jumbotron-fluid d-none d-md-block">...
</section>

> <section class="row mx-auto">...
</section>

<?php

$content = ob_get_clean();
$title = "Bienvenue";
require "template.php";
?>
```

Ajout de champs required et d'un champ caché avec une variable \$csrf Token

```
<form action="connexion_traitement" method="post">
    <h2 class="text-center">Connexion</h2>
    <div class="form-group">
        <input type="email" name="email" class="form-control" placeholder="Email" required="required" autocomplete="on">
    </div>
    <div class="form-group">
        <input type="password" name="password" class="form-control" placeholder="Mot de passe" required="required" autocomplete="on">
    </div>
    <div class="form-group">
        <input type="hidden" name="csrf" value=<?php echo $csrf; ?>>
    </div>
    <div class="form-group">
        <button type="submit" class="btn btn-primary btn-block">Connexion</button>
    </div>
</form>
```



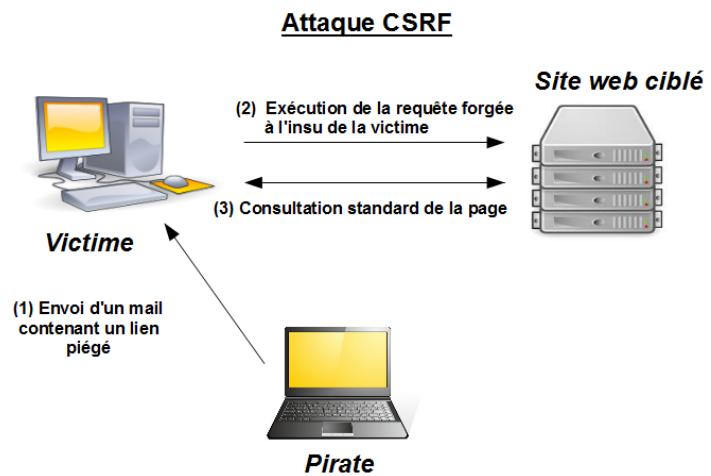
## CSRF (Cross Site Request Forgery)

Elle consiste à effectuer des opérations sur un site sans le consentement d'un utilisateur.

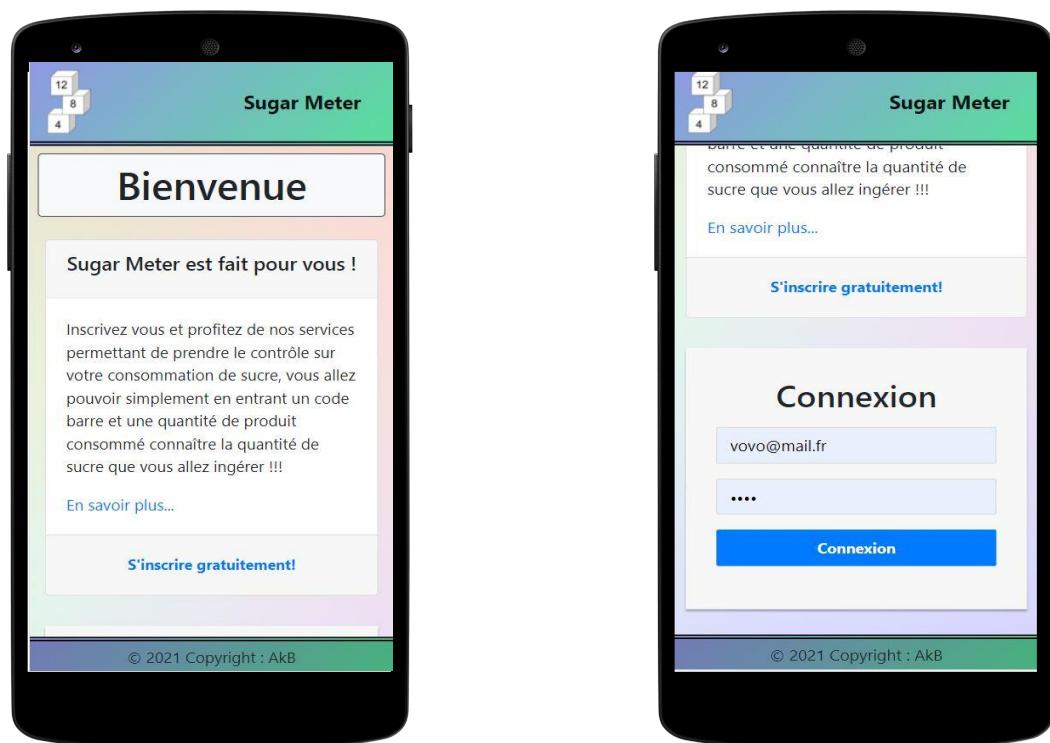
Pour se protéger d'attaques du type CSRF.

**On va générer un token dans un champ caché qui sera récupéré lors de l'appel de notre action POST du formulaire.**

Si le token est non présent lors de la création du formulaire et qu'il est mis sur l'action du contrôle alors par défaut il aura la page d'erreur.



- Une partie message de bienvenue avec but du site et lien vers ANSES pour les recommandations de consommation de sucre
- Lien vers la page d'inscription
- Module de connexion



## Inscription



- **Le pseudo** est utilisé pour s'adresser à l'utilisateur
- **L'email** comme login pour la connexion
- Afin de s'assurer de la bonne saisie du password on demande à l'utilisateur de le saisir à nouveau
- J'ai ajouté une checkbox d'acceptation des termes et conditions

```
<form action="inscription_traitement" method="post">
    <h2 class="text-center">Inscription</h2>
    <div class="form-group">
        <input type="text" name="pseudo" class="form-control" placeholder="Pseudo" required="required" autocomplete="off">
    </div>
    <div class="form-group">
        <input type="email" name="email" class="form-control" placeholder="Email" required="required" autocomplete="off">
    </div>
    <div class="form-group">
        <input type="password" name="password" class="form-control" placeholder="Mot de passe" required="required" autocomplete="off">
    </div>
    <div class="form-group">
        <input type="password" name="password_retype" class="form-control" placeholder="Re-tapez le mot de passe" required="required" autocomplete="off">
    </div>
    <div class="form-group">
        <p><input type="checkbox" required name="terms"> J'accepte les <u>Termes and Conditions</u></p>
    </div>
    <div class="form-group">
        <button type="submit" class="btn btn-primary btn-block">Inscription</button>
    </div>
</form>
```

## Form





## c. Développer une interface utilisateur web dynamique

### Enrichir notre base de données

Lors d'une recherche, on vérifie que le code barre existe dans la base de données,

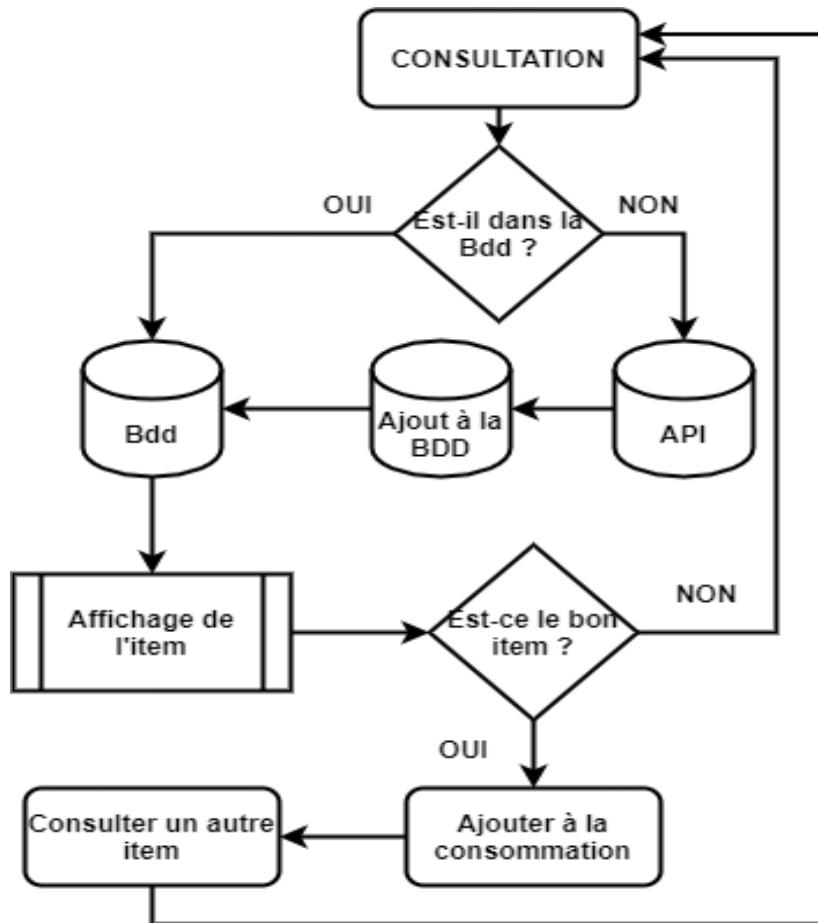
- **Si il existe**, on récupère les informations du produit dans notre base de données puis on calcule la quantité de sucre consommée (quantité de produit consommé x le pourcentage de sucre du produit) et enfin on affiche les informations.
- **Si il n'existe pas**, on récupère les informations sur l'API Open Food Facts, puis on les insère dans notre base de données pour l'enrichir, ensuite on récupère les informations du produit qui viennent d'être insérées dans notre base de données puis on calcule la quantité de sucre consommée (quantité de produit consommé x le pourcentage de sucre du produit) et enfin on affiche les informations.

```
<div class="row mx-auto">
    <!-- Zone de recherche d'un produit-->
    <div class="col-lg-6 mt-3 mb-5">
        <div class="card mx-auto h-100">
            <div class="card-header">
                <h5 class="card-title text-center name">Rechercher un produit</h5>
            </div>
            <div class="card-body border border-black">
                <input type="number" placeholder="Entrez un code barre" id="codeBarre" required minlength="6" maxlength="13">
                <hr>
                <input type="number" placeholder="Quantité en g ou ml" id="quantite" required minlength="1" maxLength="10000" style="width: 100%; height: 30px;">
            </div>
            <div class="card-footer">
                <a class="btn btn-light btn-rounded border border-dark d-block" type="button" onclick="getValue();">Rechercher</a>
            </div>
        </div>
    </div>
</div>
```





## Logigramme du code JavaScript



```
function getValue() {  
    // Sélectionner l'élément input et récupérer sa valeur  
    var barCode = document.getElementById("codeBarre").value;  
  
    fetch("/sugar/fonction/itemInBdd.php?barCode=" + barCode)  
        .then((reponse) => reponse.text())  
        Complexity is 7 It's time to do something...  
        .then((nb) => {  
            console.log(nb);  
  
            existeBdd = nb == 1;  
  
            if (!existeBdd) { ...  
            } else {  
                // si mon produit existe en bdd  
                console.log(barCode);  
                afficherBdd(barCode);  
            } // fin fetch // fin fonction  
        });  
}
```



## Affichage des informations relatives au produit ainsi que le calcul du nombre de gramme de sucre et son équivalent en morceau de sucre.

```

✓ function afficherBdd(barCode) {
    var itemQuantite = document.getElementById("quantite").value;

    fetch("/sugar/fonction/infoFromBdd.php?barCode=" + barCode)
        .then((reponse) => reponse.json())
        Complexity is 7 It's time to do something...
        .then((data) => {
            console.log(data);
            let nameCol = data.nameCol;
            let imageUrl = data.imageUrl;
            let sucreCent = data.sucreCent;
            let ProduitId = data.ProduitId;
            // Calcule quantité saisie par le pourcentage de sucre
            var sucreQuantite = ((sucreCent / 100) * itemQuantite).toFixed(1);

            // Calcule nombre de morceau de sucre
            var morceau = (sucreQuantite / 4).toFixed(1);

            // récupéré l'Id avec ces informations
            const html = ` ... `;
            $(".product").html(html); // on insère le html dans la div de class product

            document
                .getElementById("conso_Ajout")
                .addEventListener("click", function() {
                    ajouterConso();
                });
        });
}

```

## Injection d'une card HTML intégrant les variables relatives au produit.

```

const html = `

<div class="card">

    <div class="card-header">
        <h5 class="card-title text-center">${nameCol}</h5>
    </div>

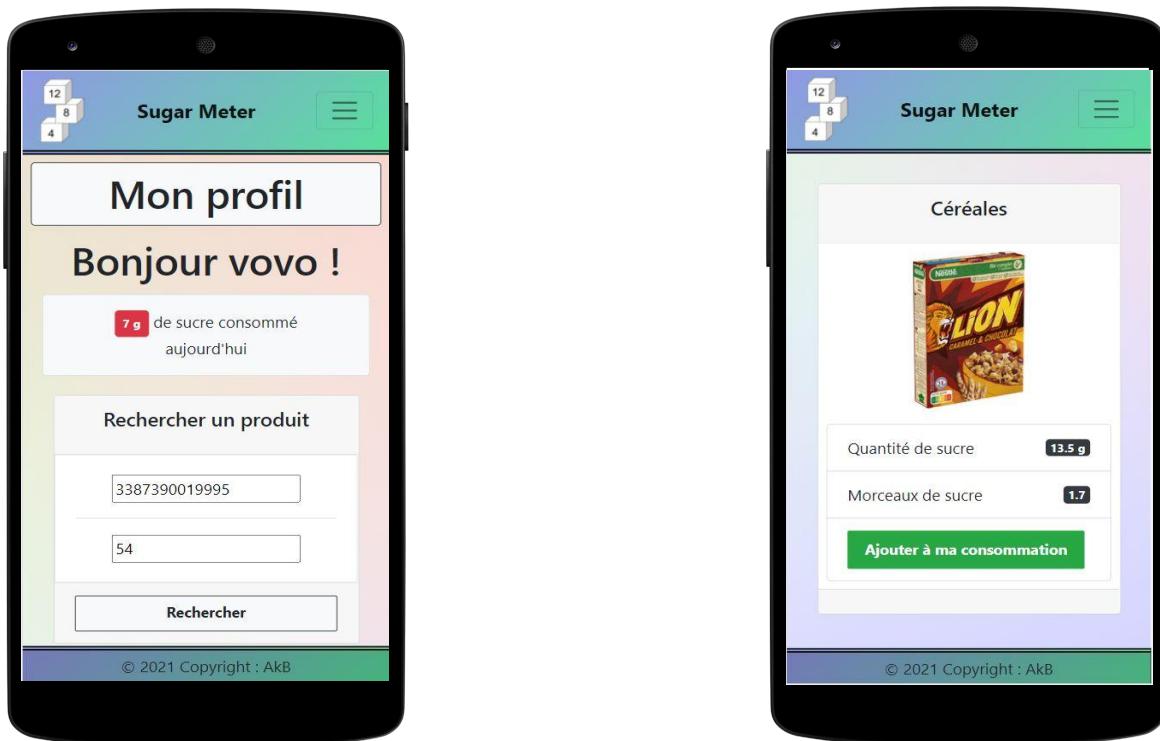
    <div class="row">
        <div class="card-img-top col-sm-4">
            
        </div>
        <div class="col-sm-8">
            <ul class="list-group p-2">
                <li class="list-group-item d-flex justify-content-between align-items-center">
                    Quantité de sucre
                    <span class="badge bg-dark text-light rounded">${sucreQuantite} g</span>
                </li>
                <li class="list-group-item d-flex justify-content-between align-items-center">
                    Morceaux de sucre
                    <span class="badge bg-dark text-light rounded">${morceau}</span>
                </li>
                <li class="list-group-item d-flex justify-content-between align-items-center">
                    <button id="conso_Ajout" class="btn btn-success btn-lg" type="button">Ajouter à ma consommation</button>
                </li>
            </ul>
        </div>
    </div>
    <div class="card-footer">
    </div>
</div>
`;

```

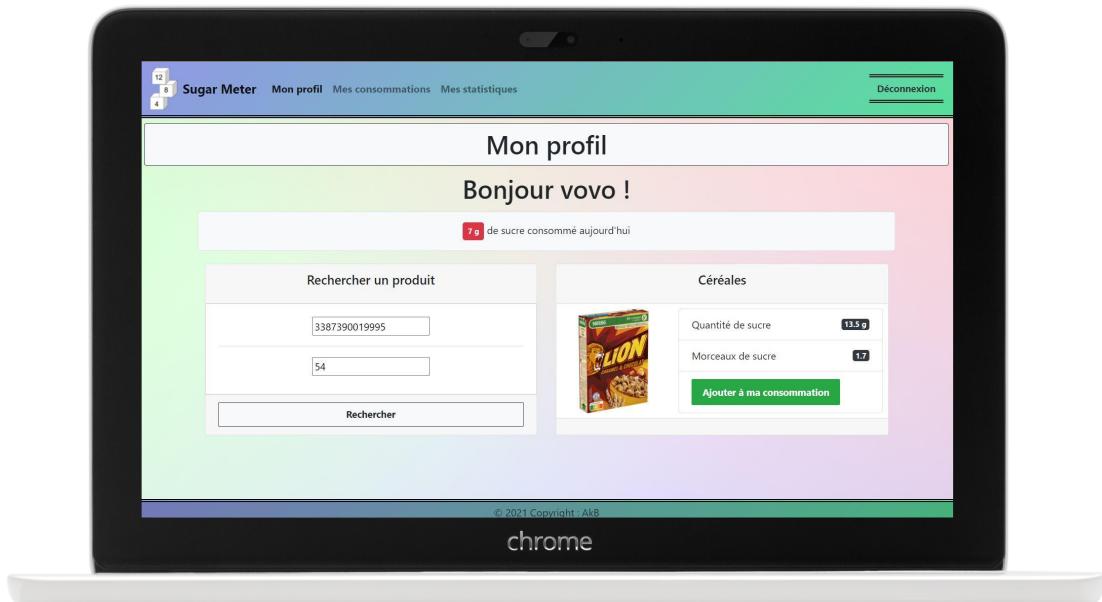


## Affichage

Captures d'écran du formulaire de recherche et de la card générée avec ses informations.

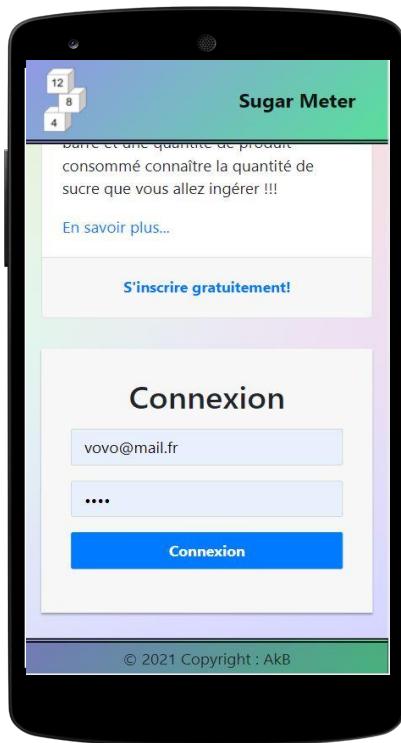


## Desktop



**d. Réaliser une interface utilisateur avec une solution de gestion de contenu**

### Connexion



```
// Si > à 0 alors l'utilisateur existe
if ($row > 0) {
    // Si le mail est bon niveau format
    if (filter_var($email, FILTER_VALIDATE_EMAIL)) {
        // Si le mot de passe est le bon
        if (password_verify($password, $data['password'])) {
            // On créer la session et on redirige sur profile.php
            $_SESSION['user'] = $data['token'];
            header('Location: profil');
            die();
        } else {
            header('Location: accueil?login_err=password');
            die();
        }
    } else {
        header('Location: accueil?login_err=email');
        die();
    }
} else {
    header('Location: accueil?login_err=already');
    die();
}
```

### Vérification

- de l'**existence de l'utilisateur**
- si l'**email est au bon format**
- si le **mot de passe est bien le bon**

**Si toutes les vérifications sont bonnes alors la session est créée et on redirige vers le profil de l'utilisateur**

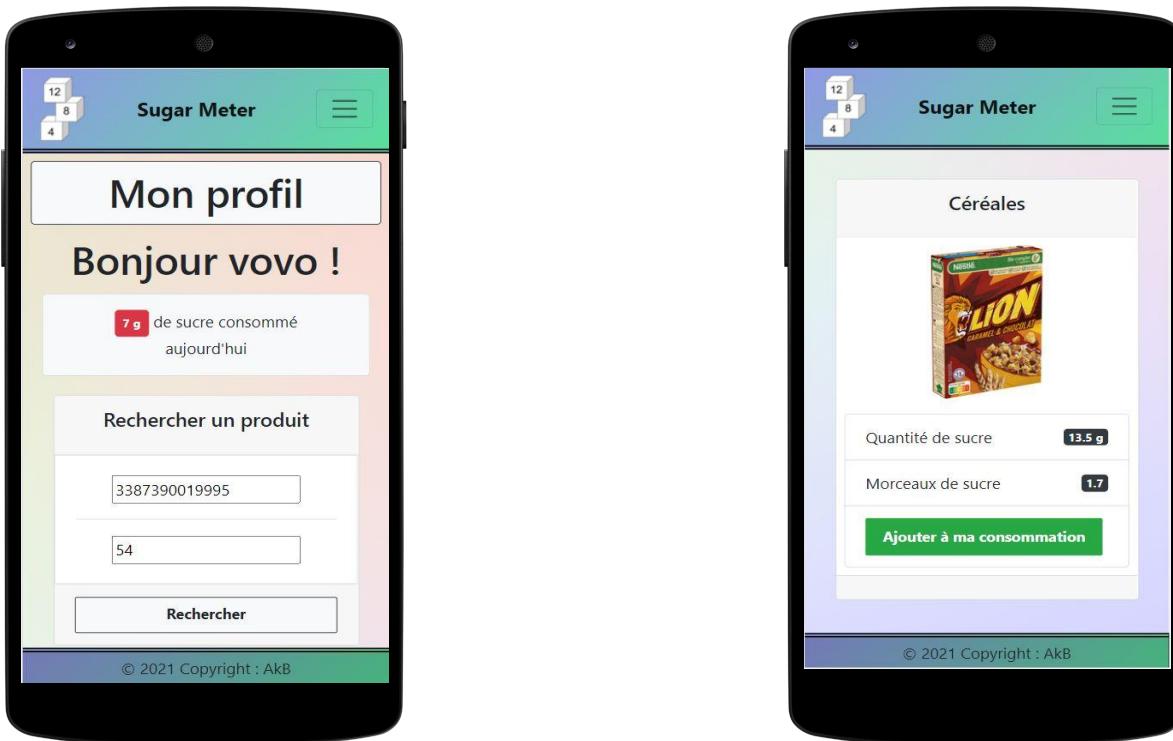
```
session_start();

// si la session existe pas soit si l'on est pas connecté on redirige
if (!isset($_SESSION['user'])) {
    header('Location: accueil');
    die();
}
```

**Si la session n'existe pas ou si on n'est pas connecté on est redirigé vers l'accueil.**



## Recherche et ajout à la consommation



```
function ajouterConso() {  
    fetch(`  
        /sugar/fonction/insererConso.php?itemQuantite=${itemQuantite}&sucreQuantite=${sucreQuantite}&ProduitId=${ProduitId}`)  
    )  
    .then((reponse) => reponse.json())  
    Complexity is 3 Everything is cool!  
    .then((reponse) => {  
        console.log(reponse["valid"]);  
        if (reponse["valid"]) {  
            document.location.href = "profil";  
            alert("Produit ajouté"); // si l'insertion s'est bien passée  
        } else {  
            document.location.href = "profil";  
            alert("Erruer produit non ajouté"); // si l'insertion s'est bien passée  
        }  
    });  
}
```



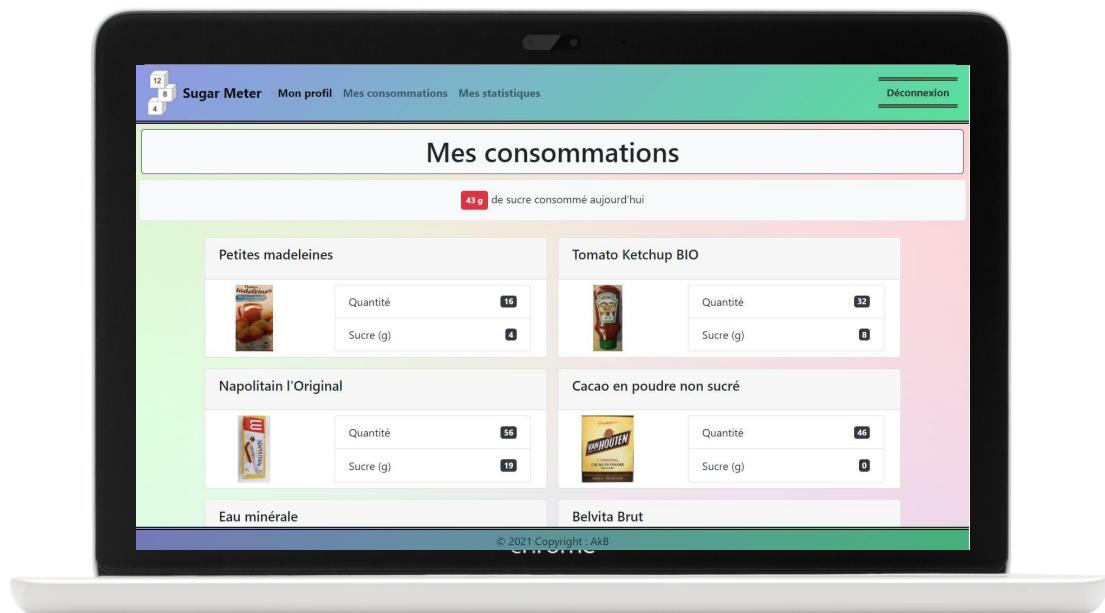
## Gestion de mes produits consommés dans la journée



### Mes consommations

- Nous retrouvons sur chaque page en rouge le compteur de sucre en gramme de la journée.
- La liste de tous les produits consommés dans la journée, le nom, l'image (pour l'identifier rapidement), la quantité de produit consommée et donc la quantité de sucre ingérée.

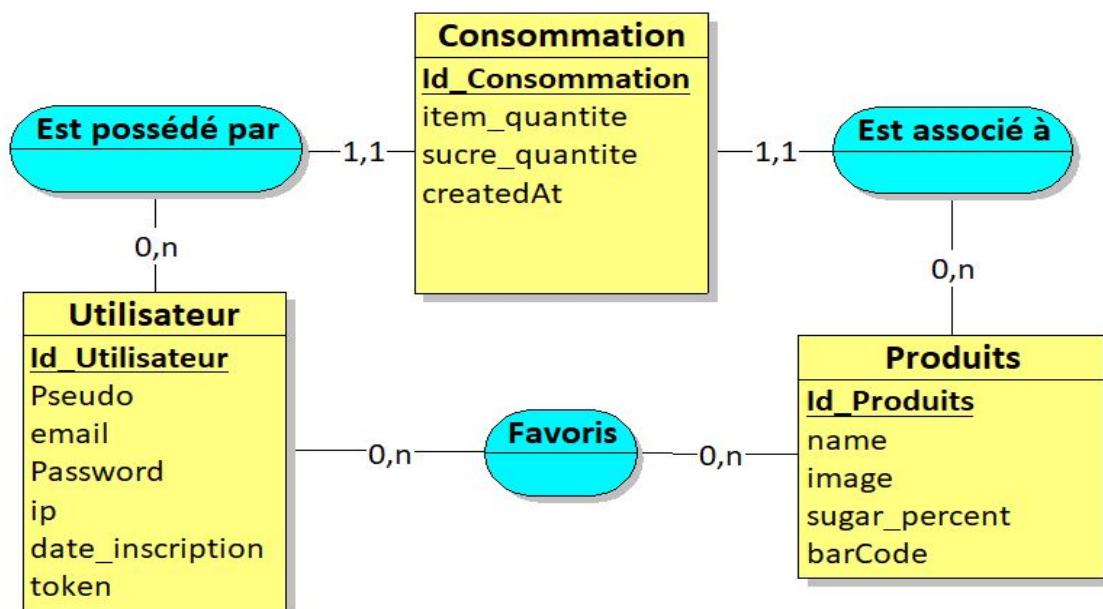
### Affichage de la liste des consommations du jour format ≥768px



## IV - Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité

### a. Créer une base de données

#### MCD (modèle conceptuel de données)



#### Utilisateur :

- peut avoir zéro ou plusieurs consommations
- peut avoir zéro ou plusieurs favoris

#### Produits :

- peut être associé à zéro ou plusieurs favoris
- peut être associé à zéro ou plusieurs consommations

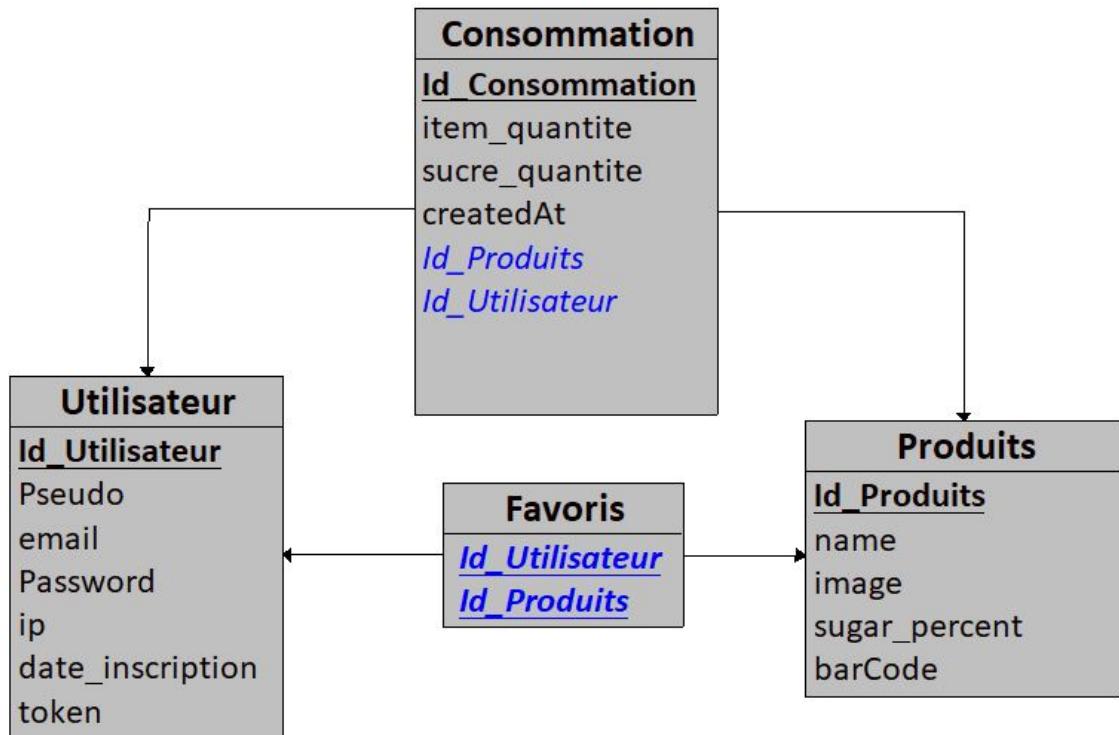
#### Consommation :

- est forcément possédée par un utilisateur et un seul
- est forcément associée à un produit et un seul

**Favoris** : couple de clés étrangères utilisateur/produit



## MLD (modèle logique de données)



Afin de modéliser le MCD et le MLD, j'ai utilisé le logiciel Looping qui m'a permis de réaliser un MCD puis de le transformer en MLD

**Le MLD textuel permet la vérification des interactions et des paramètres donnés**

**Utilisateur = (Id\_Utilisateur INT, Pseudo VARCHAR(100), email VARCHAR(50), Password VARCHAR(255), ip VARCHAR(20), date\_inscription DATETIME, token TEXT);**

**Produits = (Id\_Produits INT, name VARCHAR(50), image VARCHAR(255), sugar\_percent INT, barCode VARCHAR(13));**

**Consommation = (Id\_Consommation INT, item\_quantite INT, sucre\_quantite DECIMAL(15,2), createdAt DATE, #Id\_Produits, #Id\_Utilisateur);**

**Favoris = (#Id\_Utilisateur, #Id\_Produits);**



## Code SQL (Structured Query Language)

```
-- Structure de la table `consommation`
-- 

DROP TABLE IF EXISTS `consommation`;
CREATE TABLE IF NOT EXISTS `consommation` (
    `Id_Consummation` int(11) NOT NULL AUTO_INCREMENT,
    `item_quantite` int(11) DEFAULT NULL,
    `sucre_quantite` int(11) DEFAULT NULL,
    `createdAt` datetime DEFAULT CURRENT_TIMESTAMP,
    `Id_Produit` int(11) NOT NULL,
    `Id_Utilisateur` int(11) NOT NULL,
    PRIMARY KEY (`Id_Consummation`),
    KEY `Id_Utilisateur` (`Id_Utilisateur`),
    KEY `Id_Produit` (`Id_Produit`)
) ENGINE=InnoDB AUTO_INCREMENT=93 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

```
-- Structure de la table `utilisateur`
-- 

DROP TABLE IF EXISTS `utilisateur`;
CREATE TABLE IF NOT EXISTS `utilisateur` (
    `id_Utilisateur` int(11) NOT NULL AUTO_INCREMENT,
    `pseudo` varchar(100) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
    `email` varchar(50) COLLATE utf8mb4_unicode_ci NOT NULL,
    `password` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
    `ip` varchar(20) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
    `date_inscription` datetime DEFAULT CURRENT_TIMESTAMP,
    `token` text COLLATE utf8mb4_unicode_ci NOT NULL,
    PRIMARY KEY (`id_Utilisateur`),
    UNIQUE KEY `email` (`email`)
) ENGINE=InnoDB AUTO_INCREMENT=37 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

```
-- Structure de la table `produit`
-- 

DROP TABLE IF EXISTS `produit`;
CREATE TABLE IF NOT EXISTS `produit` (
    `Id_Produit` int(11) NOT NULL AUTO_INCREMENT,
    `name` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
    `image` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
    `sugar_percent` int(11) DEFAULT NULL,
    `barCode` char(13) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
    PRIMARY KEY (`Id_Produit`)
) ENGINE=InnoDB AUTO_INCREMENT=131 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```



## **SGBD (système de gestion de base de données)**

Un système de gestion de base de données est un logiciel système servant à stocker, à manipuler ou gérer, et à partager des données dans une base de données, en garantissant la qualité, la pérennité et la confidentialité des informations, tout en cachant la complexité des opérations (en anglais DBMS pour database management system).

Nous utilisons MySQL avec l'outil d'administration gratuit et open source phpMyAdmin. J'ai utilisé WampServer qui est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement notre base de données.

### **Relations (type InnoDB)**

Les consommations sont liées à un utilisateur et un produit (clés étrangères).

L'utilisateur à la possibilité de supprimer son compte.

Dans ces conditions, il faut supprimer les informations liées à l'utilisateur ainsi que ses consommations, mais ne doit pas effacer les informations produit (on ne souhaite pas appauvrir notre base de données).

On ajoute donc une contrainte "ON DELETE CASCADE" à la table des consommations uniquement sur la clé étrangère utilisateur.

La même contrainte est ajouté à la table des favoris pour une future intégration de la fonction favoris.

```
-- Contraintes pour la table `consommation`
--
ALTER TABLE `consommation`
  ADD CONSTRAINT `consommation_ibfk_1` FOREIGN KEY (`Id_Produit`) REFERENCES `produit` (`Id_Produit`),
  ADD CONSTRAINT `consommation_ibfk_2` FOREIGN KEY (`Id_Utilisateur`) REFERENCES `utilisateur` (`id_Utilisateur`) ON DELETE CASCADE;

-- Contraintes pour la table `favoris`
--
ALTER TABLE `favoris`
  ADD CONSTRAINT `favoris_ibfk_1` FOREIGN KEY (`Id_Utilisateur`) REFERENCES `utilisateur` (`id_Utilisateur`) ON DELETE CASCADE,
  ADD CONSTRAINT `favoris_ibfk_2` FOREIGN KEY (`Id_Produit`) REFERENCES `produit` (`Id_Produit`);
COMMIT;
```

**Iperius Backup ou autre** peuvent être utilisés pour sauvegarder les bases de données MySQL et ainsi s'assurer d'avoir des sauvegardes régulières de notre base de données.



## b. Développer les composants d'accès aux données

### PDO (PHP Data Objects)

PDO est une extension définissant l'interface pour accéder à une base de données avec PHP. Elle est orientée objet, la classe s'appelant PDO constitue une couche d'abstraction qui intervient entre l'application PHP et un système de gestion de base de données (SGDB) tel que MySQL.

**PDO comporte trois classes :**

- la classe PDO, qui représente une connexion à la base de données ;
- la classe PDOStatement, qui représente d'une part une requête SQL préparée et d'autre part le jeu de résultats de la requête une fois qu'elle est exécutée. Cette classe offre des méthodes de parcours et de comptage du jeu de résultats ;
- la classe PDOException, qui représente une erreur émise par PDO.

```
class Database //patern singleton
{
    private static $instance = null;

    /**
     * Connection à la bdd avec pdo
     * Static
     * @return PDO
     */
    public static function getBdd(): PDO
    {
        if (self::$instance === null) {
            self::$instance = new PDO("mysql:host=localhost;dbname=projet_sugar;charset=UTF8", "xxx", "xxx", [
                PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
                PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC
            ]);
        }

        return self::$instance;
    }
}
```

**Je défini le mode:** `PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION`

Quand il y a une erreur je ne veux pas qu'il reste en mode silencieux par défaut mais qu'il me donne un rapport d'erreur.

**Je défini le mode:** `PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC`

Mode d'exploitation en tableau associatif

**On vérifie si linstanciation a déjà été faite, si elle existe déjà alors on renvoi linstance.**



## c. Développer la partie back-end d'une application web ou web mobile

### Inscription

```
// Traitement de l'inscription
public function inscriptionTraitement()
{
    // Si les variables existent et qu'elles ne sont pas vides
    if (!empty($_POST['pseudo']) && !empty($_POST['email']) && !empty($_POST['password']) && !empty($_POST['password_retype'])) {
        // Patch XSS
        $pseudo = htmlspecialchars($_POST['pseudo']);
        $email = htmlspecialchars($_POST['email']);
        $password = htmlspecialchars($_POST['password']);
        $password_retype = htmlspecialchars($_POST['password_retype']);

        // On hash le mot de passe avec Bcrypt, via un coût de 12
        $cost = ['cost' => 12];
        $password = password_hash($password, PASSWORD_BCRYPT, $cost);

        // on stock l'adresse IP
        $ip = $_SERVER['REMOTE_ADDR'];

        // on insère dans la base de données
        $insert = $this->pdo->prepare('INSERT INTO utilisateur(pseudo, email, password, ip, token) VALUES(:pseudo, :email, :password, :ip, :token)');
        $insert->execute(array(
            'pseudo' => $pseudo,
            'email' => $email,
            'password' => $password,
            'ip' => $ip,
            'token' => bin2hex(openssl_random_pseudo_bytes(64))
        ));
    }
}
```

- **Utilisation de htmlspecialchars** (faille XSS) pour contrer les insertions de script, il permet de convertir les entités HTML spéciales en caractères. `strip_tags ()`.
- **Hachage du mot de passe.**
- **Requête préparer.**

### Désinscription

```
$deleteaccount = $this->pdo->prepare("DELETE FROM utilisateur WHERE id_Utilisateur = $data[id_Utilisateur]");
$deleteaccount->bindParam(':id_Utilisateur', $data['id_Utilisateur'], PDO::PARAM_INT);
$deleteaccount->execute(array($data['id_Utilisateur']));
header('Location: deconnexion');
```

Entraîne la suppression des consommations liées.



## Connexion

```
// Patch XSS
$email = htmlspecialchars($_POST['email']);
$password = htmlspecialchars($_POST['password']);

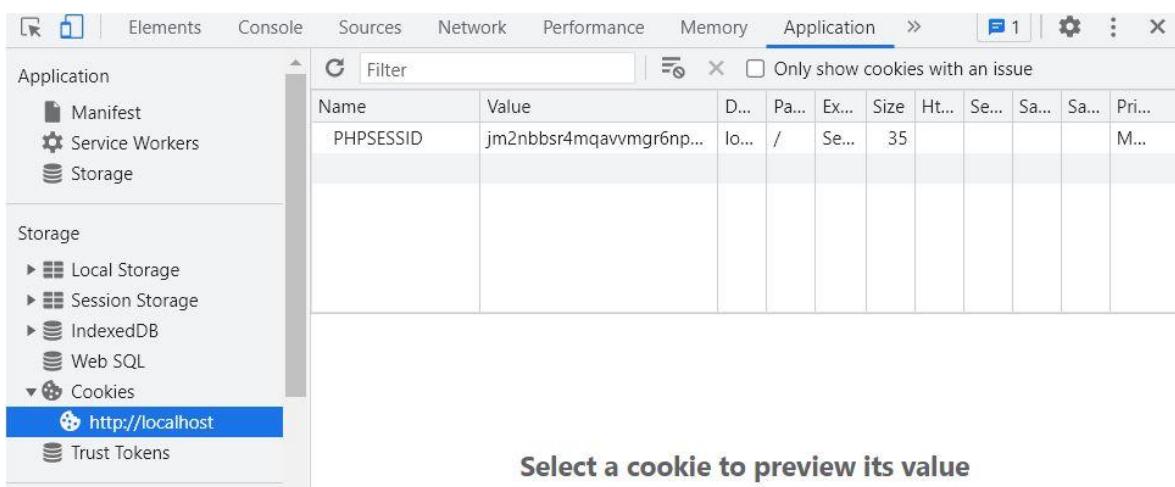
$email = strtolower($email); // email transformé en minuscule

// On regarde si l'utilisateur est inscrit dans la table utilisateurs
$check = $this->pdo->prepare('SELECT pseudo, email, password, id_Utilisateur, token FROM utilisateur WHERE email = ?');
$check->execute(array($email));
$data = $check->fetch();
$row = $check->rowCount();
```

## Déconnexion

```
// Gérer la déconnexion
public function deconnexion()
{
    session_start(); // démarrage de la session
    session_destroy(); // on détruit la/les session(s)
    header('Location: accueil'); // On redirige
    die();
}
```

**Une session en PHP** correspond à une façon de stocker des données différentes pour chaque utilisateur en utilisant un identifiant de session unique. Les identifiants de session vont généralement être envoyés au navigateur via des cookies de session et vont être utilisés pour récupérer les données existantes de la session.



The screenshot shows the Chrome DevTools Application tab open. The left sidebar lists 'Manifest', 'Service Workers', and 'Storage'. Under 'Storage', 'Local Storage' and 'Session Storage' are expanded, while 'IndexedDB', 'Web SQL', and 'Trust Tokens' are collapsed. 'Cookies' is also listed under 'Storage'. A blue bar highlights 'http://localhost'. The main area displays a table of cookies. One cookie is visible: 'PHPSESSID' with value 'jm2nbbsr4mqavvmgr6np...', expiration date 'Se...', size '35', and priority 'M...'. A message at the bottom says 'Select a cookie to preview its value'.

Name	Value	D...	Pa...	Ex...	Size	Ht...	Se...	Sa...	Sa...	Pri...
PHPSESSID	jm2nbbsr4mqavvmgr6np...	lo...	/	Se...	35					M...



## Sécurité

### L'injection SQL

Modification d'une requête SQL existante grâce à un champ mal filtré. Ce type d'attaque permet de récupérer des informations sur la base de données, 'bypasser' l'authentification etc. L'attaquant se sert soit de l'url ou bien d'un formulaire.

Faire des requêtes SQL préparées via PDO en PHP. Ce système de requête préparée est très puissant puisqu'il se charge de faire le nettoyage contre les injections SQL.

### La faille XSS

La faille XSS, « Cross-Site Scripting », permet d'injecter du code directement dans une page par l'URL, en POST, en GET et bien évidemment par l'intermédiaire d'un formulaire. Le plus souvent, cette faille est utilisée dans l'optique de capturer les cookies d'identification des utilisateurs sur un site Internet.

Si votre formulaire n'est pas sécurisé pour afficher les données soumises par l'utilisateur, alors rien n'empêche personne de mettre ce qu'il veut dans vos champs. Un simple `strip_tags` permet de pallier à ce problème (transforme les balises en string)

<https://stackoverflow.com/questions/46483/htmlentities-vs-htmlspecialchars/3614344>

12 Answers

Active Oldest Votes



`htmlspecialchars` may be used:

381

1. When there is no need to encode all characters which have their HTML equivalents.



If you know that the page encoding match the text special symbols, why would you use `htmlentities`? `htmlspecialchars` is much straightforward, and produce less code to send to the client.



For example:

```
echo htmlentities('<Il était une fois un être>');
// Output: &lt;Il &eacute;tait une fois un &ecirc;tre&gt;;
//           ^^^^^^          ^^^^^^

echo htmlspecialchars('<Il était une fois un être>');
// Output: &lt;Il était une fois un être&gt;;
//           ^           ^
```

The second one is shorter, and does not cause any problems if ISO-8859-1 charset is set.

2. When the data will be processed not only through a browser (to avoid decoding HTML entities),



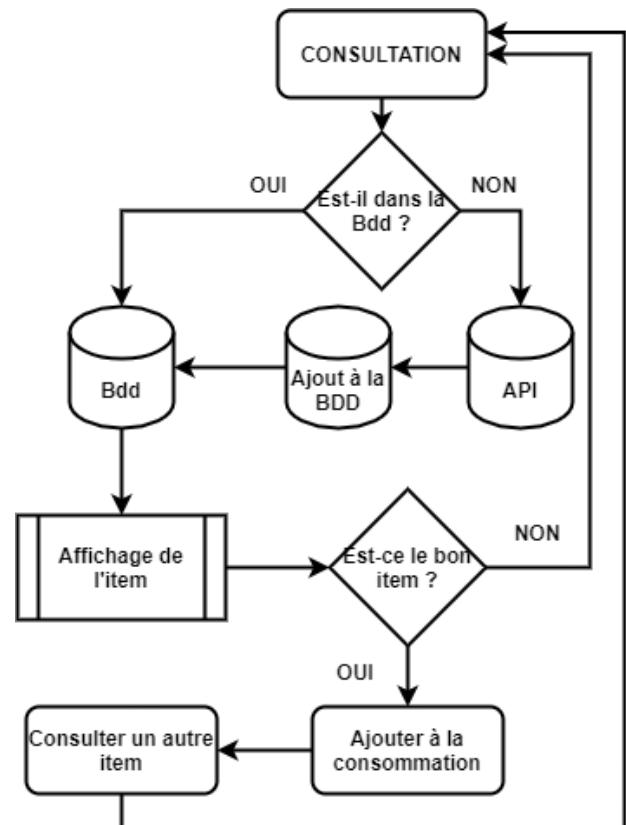
## d. Élaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce

Gérer les consommations

Existe-t-il dans la base de données

```
// on donne un code barre à partir de sugar-meter
// on reçoit un bool qui indique si ce produit existe dans sugar-meter
$barcode = $_GET['barcode'];
$sql = " SELECT COUNT(*) AS nb FROM produit WHERE barcode = :barcode ";
$data = $pdo->prepare($sql);
$data->bindValue(':barcode', $barcode, PDO::PARAM_STR);
$result = $data->execute();
$result = $data->fetch(PDO::FETCH_ASSOC);
$nb = $result['nb'] ?? 0;

echo json_encode([$nb]);
```



Si il n'existe pas on récupère sur l'API open food fact les informations produit puis on les insère dans notre base de données.

```

if (!isset($_GET['barcode']) || !isset($_GET['name']) || !isset($_GET['image']) || !isset($_GET['sugar_percent'])) {
    echo json_encode(['erreur' => '', 'msg' => "ERREUR le code barre, le nom, l'image et le pourcentage de sucre sont obligatoires"]);
    die;
}

// ventilation
$barcode = $_GET['barcode'];
$name = $_GET['name'];
$image = $_GET['image'];
$sugar_percent = $_GET['sugar_percent'];

// nettoyage
$barcode = htmlspecialchars(strip_tags(trim($barcode)));
$name = htmlspecialchars(strip_tags(trim($name)));
$image = htmlspecialchars(strip_tags(trim($image)));
$sugar_percent = htmlspecialchars(strip_tags(trim($sugar_percent)));

$sql = " INSERT INTO `produit` (`name`, `image`, `sugar_percent`, `barcode`) VALUES (:name, :image, :sugar_percent, :barcode) ";
$data = $bdd->prepare($sql);
$data->bindValue(':barcode', $barcode, PDO::PARAM_STR);
$data->bindValue(':name', $name, PDO::PARAM_STR);
$data->bindValue(':image', $image, PDO::PARAM_STR);
$data->bindValue(':sugar_percent', $sugar_percent, PDO::PARAM_INT);

$result = $data->execute();

// NOTE : $result renvoie un boolean, vrai si ça a fait le INSERT, false sinon
if ($result) {
    $strResult = '1';
    echo "ok";
} else {
    $strResult = '0';
    echo "pas ok";
}

echo json_encode(['valid' => $strResult]);

```

In 21 Col 69 - Spaces: 4 - LTFE: 8

On peut donc maintenant utiliser les informations nouvellement insérées dans la base de données.

```
$barcode = $_GET['barcode'];

$sql = " SELECT name, image, sugar_percent, Id_Produit FROM produit WHERE barcode = :barcode ";
$data = $bdd->prepare($sql);
$data->bindValue(':barcode', $barcode, PDO::PARAM_STR);
$result = $data->execute();
$count = $data->rowCount();
$result = $data->fetch(PDO::FETCH_ASSOC);

// VENTILATION
$name = $result['name'];
$image = $result['image'];
$sugar_percent = $result['sugar_percent'];
$Id_Produit = $result['Id_Produit'];

// NOTE : je masque les noms des colonnes pour qu'on ne voit pas les noms de mes colonnes de BDD dans le JS
echo json_encode([
    'nameCol' => $name,
    'imageUrl' => $image,
    'sucreCent' => $sugar_percent,
    'ProduitId' => $Id_Produit,
    ...
]);
```

Et pour finir, on insère dans la base de données la consommation de l'utilisateur connecté

```
if (!isset($_GET['itemQuantite']) || !isset($_GET['sucreQuantite']) || !isset($_GET['ProduitId'])) {
    echo json_encode(['erreur' => '', 'msg' => "ERREUR veuillez entrer une quantité"]);
    die;
}

$itemQuantite = $_GET['itemQuantite'];
$sucreQuantite = $_GET['sucreQuantite'];
$ProduitId = $_GET['ProduitId'];
$UtilisateurId = $data['id_Utilisateur'];

$sql = " INSERT INTO consommation (item_quantite, sucre_quantite, Id_Produit, Id_Utilisateur) VALUES (:item_quantite, :sucre_quantite, :Id_Produit,
$query = $pdo->prepare($sql);
$query->bindValue(':item_quantite', $itemQuantite, PDO::PARAM_STR);
$query->bindValue(':sucre_quantite', $sucreQuantite, PDO::PARAM_STR);
$query->bindValue(':Id_Produit', $ProduitId, PDO::PARAM_STR);
$query->bindValue(':id_Utilisateur', $UtilisateurId, PDO::PARAM_STR);
$result = $query->execute();

// NOTE : $result renvoie un boolean, vrai si ça a fait le INSERT, false sinon
if ($result) {
    $strResult = '1';
} else {
    $strResult = '0';
}
echo json_encode(['valid' => $strResult]);
```

Utilisation de bindValue et de contrôle des erreurs éventuelles



Ici, on souhaite récupérer les informations de deux tables différentes, on utilise donc une jointure INNER JOIN.

Puis, à l'aide d'un foreach on itère les consommations de la journée pour un utilisateur

```
$query = $this->pdo->prepare("SELECT name, image, item_quantite, sucre_quantite
    FROM consommation c
    INNER JOIN produit p ON p.Id_Produit = c.Id_Produit
    WHERE id_Utilisateur = $data[id_Utilisateur] AND CAST(createdAt AS DATE) = CAST( curdate() AS DATE)
    ORDER BY createdAt DESC");
$query->bindParam(':id_Utilisateur', $data['id_Utilisateur'], PDO::PARAM_INT);
$query->execute();
$consommations = $query->fetchAll(PDO::FETCH_ASSOC);
return $consommations;
```

De la même manière, les statistiques quant à eux reprennent les informations des 7 derniers jours.

```
$query = $this->pdo->prepare("SELECT item_quantite, sucre_quantite, createdAt
    FROM consommation c
    INNER JOIN produit p ON p.Id_Produit = c.Id_Produit
    WHERE id_Utilisateur = $data[id_Utilisateur] AND DATE_SUB(CURDATE(), INTERVAL 7 DAY) <= createdAt
    ORDER BY createdAt DESC");
$query->bindParam(':id_Utilisateur', $data['id_Utilisateur'], PDO::PARAM_INT);
$query->execute();
$stats = $query->fetchAll();
return $stats;
```

### Le compteur de sucre (en gramme) des consommations du jour utilise :

- Des opérations dans la requête SQL (sum)
- Le CAST fonction de transtypage qui permet de convertir une données d'un type en un autre
- Le curdate est une fonction qui retourne la date courante

```
$query = $this->pdo->prepare("SELECT SUM(sucre_quantite) AS Total_sucre
    FROM consommation
    WHERE id_Utilisateur = $data[id_Utilisateur] AND CAST(createdAt AS DATE) = CAST( curdate() AS DATE)");
$query->bindParam(':id_Utilisateur', $data['id_Utilisateur'], PDO::PARAM_INT);
$query->execute();
$sucreT = $query->fetchColumn();
return $sucreT;
```



## Conclusion

---

Ce projet tutoré bien que très court en temps de réalisation m'a permis de revoir et mettre en application une grande partie des enseignements suivis au cours de cette formation.

Je regrette cependant de ne pas avoir trouvé de stage en adéquation avec ma formation, j'espère pouvoir lors de mon premier emploi dans ce domaine, compléter mon expertise avec la dimension plus opérationnelle que peut offrir une entreprise.

### Développer la partie front-end d'une application web ou web mobile

- a. Maquetter une application
- b. Réaliser une interface utilisateur web statique et adaptable
- c. Développer une interface utilisateur web dynamique
- d. Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce

### Développer la partie back-end d'une application web ou web mobile

- a. Créer une base de données
- b. Développer les composants d'accès aux données
- c. Développer la partie back-end d'une application web ou web mobile
- d. Élaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce



## Annexes

---

**CNIL** Réglementation CNIL 1er octobre 2020, Cookies et traceurs

**Il faut informer l'utilisateur avec :**

- l'ensemble des finalités d'usage liées aux traceurs
- une liste des responsables du traitement des données

**Il faut permettre à l'utilisateur de :**

**1- Consentir par un acte positif clair**

- un silence est un refus
- les cases sont décochées par défaut

**2- Faire un choix par finalité**

- Consentement indépendant et spécifiques
- Boutons « tout accepter » et « tout refuser »

**3- Choix avec le même degré de simplicité**

- deux boutons au même niveau
- le choix conservé pour 12 mois

**4- Revenir sur sa décision à tout moment**

- un lien en pied de page ou autre

Une ambiguïté persiste sur la gestion des cookies analytics

- les traceurs exemptés de consentement, certains traceurs de mesure d'audience
  - Les traceurs destinés à l'authentification auprès d'un service.
  - Les traceurs destinées à garder en mémoire le contenu d'un panier.
  - Certains traceurs visant à générer des statistiques de fréquentation.
  - Les traceurs permettant aux sites payant de limiter l'accès gratuit.
  - les traceurs de personnalisation de l'interface utilisateur
  - les traceurs permettant l'équilibrage de la charge des équipements.





## Sécurité

### Parameter Tampering

Attaque basée sur un formulaire en passant certains paramètres dans l'URL ou changement d'un champ d'un formulaire sans l'autorisation de l'utilisateur.

### cURL

Le cURL est une librairie permettant de faire des requêtes URL aux clients tout cela en ligne de commande. C'est un mécanisme très puissant puisque cela vous permet de récupérer le contenu d'une page web, d'envoyer des fichiers, de s'authentifier et donc bien évidemment de soumettre des données dans un formulaire HTML. En effet, il est tout à fait possible de soumettre un formulaire sans y accéder par un navigateur où plutôt soumettre des données sur l'action du formulaire. Cela s'applique également pour les formulaires en Ajax qui contiennent une URL.

cURL peut être utilisé dans des objectifs sains comme l'utilisation d'API REST ou bien de façon néfaste comme pirater un site Internet en contournant les sécurités mis en place côté client.

On peut soumettre le formulaire autant de fois que l'on veut avec un tel script en y ajoutant une boucle et il serait alors judicieux de le protéger avec un captcha par exemple.

### OverPosting

Poster des propriétés qui n'apparaissent pas dans le formulaire mais qui sont contenues dans le modèle du formulaire.

### Clics multiples

Quand le formulaire met trop de temps à s'exécuter, les utilisateurs ont tendance à cliquer une nouvelle fois sur le bouton "Envoyer". Cela a pour conséquence d'envoyer plusieurs fois la demande à notre méthode qui reçoit le POST et de potentiellement créer autant d'enregistrements. Pour corriger cela, il suffit d'ajouter un code JavaScript permettant sa désactivation lors du 1er clic :

```
if ($('#idForm').valid()) { $("#btnSubmit").attr("disabled", "disabled"); return true; };
```

**Le captcha** est une mesure de sécurité qui fait que la réponse ne peut pas être générée par un ordinateur mais par l'utilisateur.



## Google présente l'avancée de son projet « Privacy Sandbox »

**L'objectif :** éliminer les cookies tiers en les remplaçant par des techniques viables qui garantissent la confidentialité des utilisateurs, tout en permettant aux annonceurs de continuer à diffuser des annonces adaptées et ciblées.

### Lancement de tests pour le ciblage par groupe (FLoC)

L'idée serait de remplacer le ciblage individuel (qui fonctionne avec les cookies tiers) par un ciblage de groupe via l'apprentissage fédéré par cohortes (FLoC, Federated Learning of Cohorts). Concrètement, l'utilisation de cohortes d'audience permet de cibler collectivement plutôt qu'individuellement des utilisateurs ayant des historiques de navigation et des intérêts similaires.

### La création d'audiences sans cookies tiers pour les annonceurs

Au sein de l'initiative Privacy Sandbox, la création d'audience sans utiliser de cookies tiers est aussi étudiée afin de permettre aux annonceurs de mettre en place du remarketing d'une autre façon. Chrome propose une solution baptisée FLEDGE (appelée précédemment TURTLEDOVE), qui permettrait d'introduire un serveur tiers de confiance dans le processus, afin de stocker des informations sur les enchères et les budgets d'une campagne.

Cette solution sera disponible pour des tests plus tard dans l'année et Google souhaite laisser la possibilité à des sociétés d'ad tech d'utiliser l'API avec un modèle où ils peuvent apporter leur propre serveur. Google invite aussi les annonceurs à participer aux tests d'enchères dans Privacy Sandbox afin de s'assurer que les enchères publicitaires fonctionneront correctement une fois les cookies tiers supprimés.

### D'autres sujets étudiés pour allier gestion des données et sécurité

- La mesure des conversions : Pour évaluer la manière dont les API de mesure de conversion peuvent être utilisées avec les produits de mesure de Google afin de répondre à des cas d'utilisation
- La prévention de la fraude publicitaire : Chrome avait lancé des tests de l'API Trust Token qui permet de vérifier l'authenticité du trafic sans exposition de l'identité au cours de la vérification.



## Internet en 2021 : les 10 chiffres clés

Sur 7,83 milliards d'individus sur la planète, on recense :

- 5,22 milliards de mobinautes (+1,8 %),
- 66,6 % de la population mondiale utilise mobile,
- 93 millions de nouveaux utilisateurs de mobile (+1,8 % en un an),
- 96,43 Mbps de vitesse de connexion moyenne d'une ligne fixe à Internet (+31,1 %),
- 4,66 milliards d'internautes (59,5 %),
- 4,32 milliards d'utilisateurs d'Internet à partir d'un mobile,
- 316 millions d'internautes supplémentaires en 2020 (+7,3 %),
- 4,2 milliards d'utilisateurs actifs des réseaux sociaux (53,6 %),
- 6h54 passées en moyenne sur Internet (+4 %),
- 63,4 % des internautes utilisent le navigateur Chrome (-0,4 %).

L'étude précise que la pandémie de la Covid-19 a eu un impact sur la déclaration du nombre d'utilisateurs d'Internet. Alors que le taux de pénétration atteint en janvier 2021 59,5 % de la population mondiale avec un total de 4,66 milliards d'internautes, ces chiffres pourraient être plus importants que ceux énoncés. Il sera intéressant d'en suivre l'évolution dans les prochaines mises à jour du rapport.



## Le mobile en 2021 : les 10 chiffres clés

À partir des données fournies par App Annie, l'étude montre que le mobile est devenu « *notre premier écran* », avec 3h39 passées sur ce type d'appareil, qui se place désormais devant la télévision (3h34). Sur 5,22 milliards de mobinautes dans le monde, qui représentent 66,66 % de la population mondiale, on compte :

- 92,6 % d'utilisateurs qui accèdent à Internet à partir d'un mobile,
- 52,8 % du temps passé en moyenne sur Internet dans le monde depuis un appareil mobile, au 3e trimestre 2020 (+3,9 %),
- 4h10 passées en moyenne sur un appareil mobile Android dans le monde (+20 %),
- 3h39 de temps passé en moyenne par jour sur Internet sur mobile dans le monde,
- 8,02 milliards de connexions mobiles (+0,9 % en un an),
- 76,3 % des connexions mobiles s'effectuent depuis un modèle de type smartphone, 20,1 % par des « *feature phones* » et 3,6 % par d'autres appareils comme des tablettes,
- 163 % de connexions mobiles en Afrique du Sud, soit le taux le plus élevé par rapport à la population mondiale (moyenne : 102,4 % dans le monde),
- 9,4 GB utilisés par mois en moyenne sur un smartphone dans le monde,
- 218 milliards d'applications mobiles téléchargées en 2020 (chiffres App Annie),
- 143 milliards de dollars dépensés dans des apps mobiles (+ 20 % sur un an), soit en moyenne 23,62 dollars par smartphone dans le monde.

Si le volume de smartphones représente 7,67 milliards de devices dans le monde, l'étude recense 10,75 milliards d'appareils connectés à courte portée et 1,36 milliard de téléphones fixes.

la grande étude annuelle *Digital Report 2021*, publiée par Hootsuite et We Are Social

