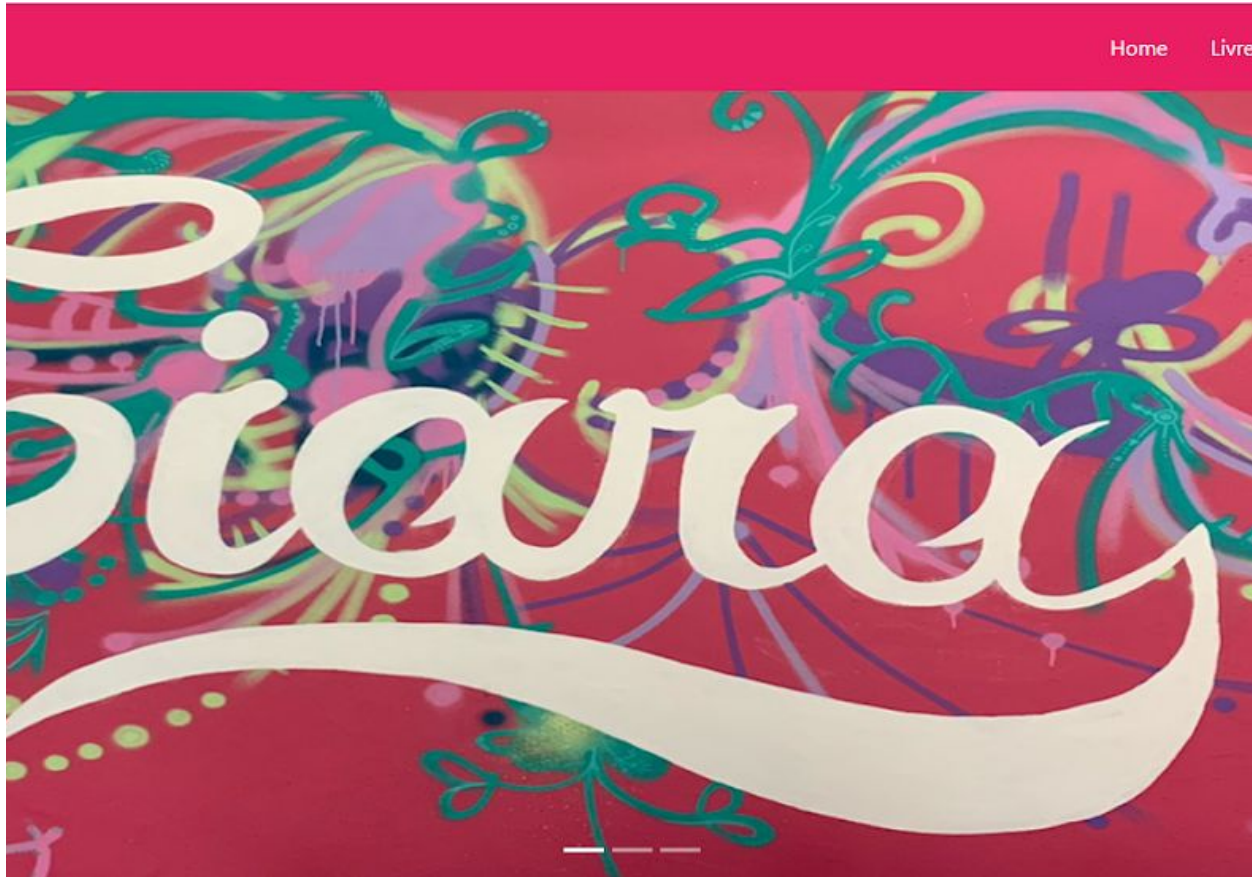


# Ciara Cut

*Projet professionnel de réservation pour un salon de coiffure*



**ANTHONY REBOLI**

**Coding School - Dev Web & Web mobile**

# SOMMAIRE

<b>SOMMAIRE</b>	<b>1</b>
<b>INTRODUCTION</b>	<b>3</b>
<b>Présentation de l'entreprise</b>	<b>3</b>
<b>Présentation du personnel</b>	<b>3</b>
<b>Présentation de l'équipe</b>	<b>4</b>
<b>Formation</b>	<b>4</b>
<b>Organisation du projet</b>	<b>5</b>
<b>Les différents sites créer</b>	<b>5</b>
<b>Détail du projet professionnel</b>	<b>5</b>
<b>Choix technique et méthodologie</b>	<b>6</b>
<b>Les bases de données</b>	<b>6</b>
<b>Cahier des charges</b>	<b>7</b>
<b>Sécurité</b>	<b>7</b>
<b>Protéger toutes les variables qu'un utilisateur peut modifier directement ou indirectement.</b>	<b>8</b>
<b>Recherche sur des sites anglophones</b>	<b>9</b>
<b>Les infos des pages</b>	<b>9</b>
Pour la page CLIENT	10
Pour la page STOCK	10
<b>Détail technique back-end</b>	<b>11</b>
Pour la page client	11
La page stock :	15
<b>Un jeu d'essai</b>	<b>19</b>
<b>Les compétences acquises</b>	<b>19</b>
<b>Compétences WEB</b>	<b>20</b>
<b>Débogage du site en général</b>	<b>20</b>
<b>Les rendez-vous client</b>	<b>21</b>
<b>Le Front</b>	<b>21</b>
<b>Conclusion</b>	<b>25</b>



## INTRODUCTION

Ce dossier a pour objectif d'expliquer mon projet professionnel pour le client Ciara cut. Le projet consiste globalement à créer un site de réservation et de panel admin pour la gestion de ces clients.

S'ajoute à ce projet, la création d'une page stock pour la vérification des stocks en temps réel et la gestion de l'inventaire.

La cliente a demandé qu'il y est un accès privé au livre d'or des client.

Ainsi elle peut gérer complètement son site avec Les réservations , la clientèle et stock.

## Présentation de l'entreprise

L'entreprise Ciara cut est un salon de coiffure tenu par une seule personne, la dirigeante. Son fonctionnement de travail n'est pas moderne et non digitalisé : Toute sa gestion notamment ses comptes et le stock de marchandise est faite sur papier. Dans le but de se moderniser, elle a souhaité créer un site internet dynamique avec une page vitrine.

## Présentation du personnel

Je suis un étudiant de l'école La Plateforme Marseille qui forme aux métiers de développeur web & web mobile. L'objectif est d'apprendre le code HTML / CSS / PHP / SQL / JAVASCRIPT.

Concernant mon parcours, j'ai fais de l'intérim dans le domaine de l'industrie en tant que monteur tuyauteur pour monter ou démonter des brides. Après une décision personnelle j'ai fais une reconversion dans le développement web, ce qui m'amène à me présenter aujourd'hui au titre RNCP .

Étant un "gamer" je me suis toujours posé la question de comment fonctionne un jeu vidéo et j'ai découvert un univers vaste et très intéressant.

## Présentation de l'équipe

Nous avons travaillé ce projet en équipe composée de :

GREGORY FAUVEL pour le planning, la réservation et la boutique.

THOMAS SAUSSOL pour la partie Connexion, Inscription et livre d'or.

Moi même pour la partie Produit, stock, client, fidélité et debug general.

## Formation

L'école La Plateforme est une association de loi 1901 fondée par Cyril Zimmermann en 2019 et financée pour son lancement par les entreprises du club top 20 (qui regroupe les 50 plus grosses entreprises de la Région PACA). Les formations qu'elle propose sont totalement gratuites et ils intègrent tous les publics de tous âges.

La réponse à un secteur sous tension : La plateforme a été fondée pour répondre à une demande non comblée du marché de l'emploi.

Le manque de développeur autonome se fait cruellement ressentir sur le marché des nouvelles technologies. Le but de ce centre est de fournir à ce marché de nouveau profil venant de tout horizon et de tous âges.

Une nouvelle pédagogie : La pédagogie de La Plateforme est une nouvelle pédagogie qui travaille sur l'autonomie. Ceci avec une liberté de travail au niveau horaire et dans le choix de ses collaborateurs.

Les runtracks : Les runtracks sont mise en place lors de la découverte des nouveaux langages. Ces runtracks sont présenté sous la forme d'un pitch qui dure en moyenne une demi heure et une série d'exercices sont mis en place dans la journée qui regroupe tout le panel des propriétés vue dans le pitch.

Exemple pour le module javascript : lundi javascript, mardi ajax ,mercredi jquery, jeudi json, vendredi un big job regroupant tous ce qui a été vu dans la semaine.

Pas de cours magistraux mais beaucoup de mise en situation : La Plateforme ne donne pas de cours mais organise des mises en situation avec des projets auxquels nous devons nous inscrire entre 1 et 5 personnes par projet avec des deadlines à tenir.

Ceci a pour but d'apprendre à travailler en groupe sur des projets et augmenter la le partage de connaissance ainsi que les compétences transverses.

Google est notre ami : Étant donné que nous ne suivons pas de cours nous devons nous débrouiller pour aller à la recherche des informations. Pour cela, nous nous tournons naturellement sur de la recherche internet ce qui a pour but de développer notre autonomie.

## Organisation du projet

La première étape est de demander au client l'idée du site et les fonctionnalités nécessaires pour son projet.

Faire le dispatch des tâches pour que tout le monde est une partie à travailler et que l'on a un plan pour commencer les maquettes.

Le maquettage des pages en démonstration pour voir les fonctionnalités générales.

Le maquettage de la base de données avec genmymodel.

La programmation du site.

## Les différents sites créer

J'ai effectué plusieurs projets avec ma formation La Plateforme :

- E-Commerce Boutique de thé
- E-Commerce Boutique de liseuse
- Réservation camping
- Réservation paintball
- Jeux clicker
- Ciara Cut projet professionnel

## Détail du projet professionnel

Pour le projet Ciara Cut j'ai utilisé PHP/SQL/CSS /HTML/Java Script. J'ai travaillé en équipe avec Gregory FAUVEL et THOMAS SAUSSOL pendant le confinement.

Pour mettre en place l'organisation du travail on a créé un github pour sauvegarder et partager nos pages.

Pour bien placer l'organisation des pages on a utilisé l'application TRELLO qui permet de gérer des événements en équipe pour suivre le déroulement des étapes.

J'ai pris les pages clients,Stock,Fidélité.

## Choix technique et méthodologie

Avant de commencer le projet, j'ai fait plusieurs recherches concernant les choix techniques. J'ai mis en place une méthode de travail simple mais efficace qui utilise php .J'ai utilisé Git comme logiciel de versioning, couplé à GitHub, pour stocker mon code versionné sur un serveur distant et pouvoir ainsi travailler facilement. J'utilisais principalement 2 branches, une pour moi et l'autre pour les merges.

## Les bases de données

Je commence par faire une maquette avec le site mygenmodel qui permet de bien structurer la base de données avant de la créer en brut.

PHOTO DE LA MAQUETTE

EXPLICATION

Puis je la créer via PHPmyadmin

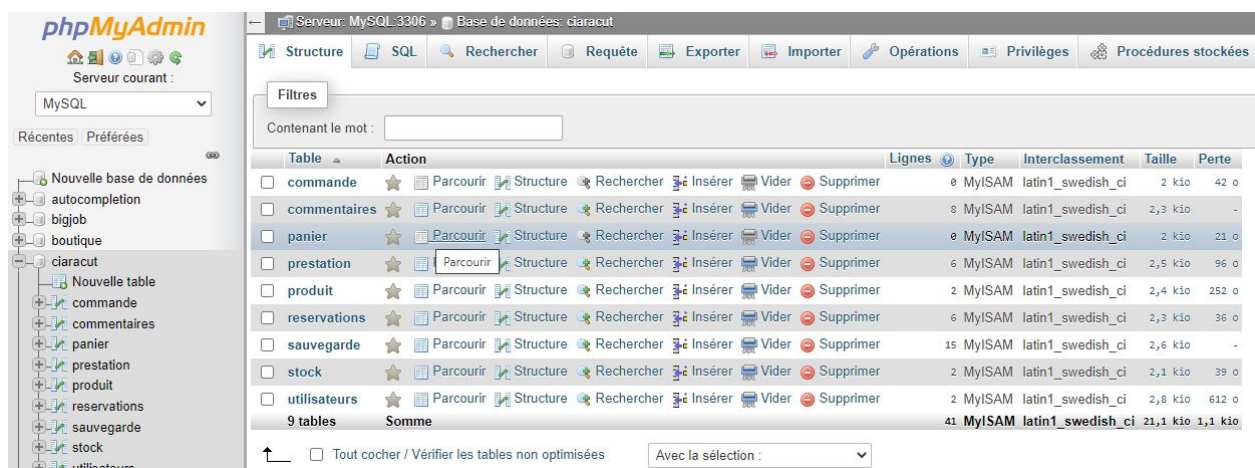


Table	Action	Lignes	Type	Interclassement	Taille	Perte
commande	Parcourir Structure Rechercher Insérer Vider Supprimer	0	MyISAM	latin1_swedish_ci	2 kio	42 0
commentaires	Parcourir Structure Rechercher Insérer Vider Supprimer	8	MyISAM	latin1_swedish_ci	2,3 kio	-
panier	Parcourir Structure Rechercher Insérer Vider Supprimer	0	MyISAM	latin1_swedish_ci	2 kio	21 0
prestation	Parcourir Structure Rechercher Insérer Vider Supprimer	6	MyISAM	latin1_swedish_ci	2,5 kio	96 0
produit	Parcourir Structure Rechercher Insérer Vider Supprimer	2	MyISAM	latin1_swedish_ci	2,4 kio	252 0
reservations	Parcourir Structure Rechercher Insérer Vider Supprimer	6	MyISAM	latin1_swedish_ci	2,3 kio	36 0
sauvegarde	Parcourir Structure Rechercher Insérer Vider Supprimer	15	MyISAM	latin1_swedish_ci	2,6 kio	-
stock	Parcourir Structure Rechercher Insérer Vider Supprimer	2	MyISAM	latin1_swedish_ci	2,1 kio	39 0
utilisateurs	Parcourir Structure Rechercher Insérer Vider Supprimer	2	MyISAM	latin1_swedish_ci	2,8 kio	612 0
9 tables	Somme	41	MyISAM	latin1_swedish_ci	21,1 kio	1,1 kio

J'ai créé une nouvelle base de données puis j'ai créé plusieurs tables stock, produit, utilisateurs.

Qui permet de stocker des informations transmises par les input du site.

## Cahier des charges

- Le look du site est à la demande du client (Style girly Trois tons de couleurs principales dont le noir, blanc, rose fushia).
- Un code couleur de référence a été établi par le client pour la couleur global du site je lui ai proposé le référentiel de couleur Materialize.
- Des informations précises de l'utilisateur à l'inscription du site ont été exigées par le client.
- Une remise de 10% à la date d'anniversaire de l'utilisateur du site valable pendant 1 semaine.
- Un fichier client dans la partie admin a été ajouté pendant la période du développement.
- Affichage des informations complètes de ses clients ainsi que ses derniers rendez-vous et dernières prestations.
- Elle veut pouvoir supprimer les messages indésirables du livre d'or et pouvoir aussi envoyer des messages à ses clients.
- Gestion du stock de son entreprise.
- Gestion des rendez-vous de ses clients.

## Sécurité

Pour la sécurité du site j'ai d'abord veiller à ce que tous mes formulaires soient bien remplis grâce à la technique REQUIRED.

J'ai veillé à ce que le mot de passe de l'utilisateur soit en mode HASH c'est à dire que le mot de passe devient crypté et personne d'autre peut y avoir accès.

L'accès au post commentaire du livre d'or est autorisé que si le paiement de la prestation est effectué.

J'ai mis en place des htmlspecialchars à tous les formulaires du site.

J'utilise le pdo pour préparer les requêtes et éviter les injection sql.



Il y a plusieurs formes de menace possible :

- **Faible XSS**

Elle consiste à injecter un contenu malveillant en langage de script dans un site web dans le but de déclencher différentes actions:

```
input = <script>alert(ruben);</script>
$message = input
echo $message
```

Comment se protéger ?

Protéger toutes les variables qu'un utilisateur peut modifier directement ou indirectement.

Pour les cookies, utilisez HttpOnly (rend les cookies inaccessibles à document.cookie et le flag Secure (ne transmettra pas le cookie via une voie non sécurisée HTTP, mais seulement par HTTPS)

Utiliser la fonction htmlspecialchars(), qui filtre les symboles (<, ", &,...)

- **Requête SQL**

Le but de cette attaque est de modifier une requête SQL par le biais d'input, que ce soit pour se logger, ou pour laisser un message dans un forum.

En code ça donne quoi ?

Requete : SELECT \* FROM utilisateurs WHERE login='\$login' AND password='\$password'  
login = admin'#

La Requete devient SELECT \* FROM utilisateurs WHERE login='admin'

Car le # met en commentaire la suite.

Le login ' OR 1=1' marche aussi.

Comment s'en protéger ? Faire des requêtes préparées grâce à PDO

- **Faible upload**

La faille upload va permettre d'exécuter un script en téléchargeant un fichier php là où le site le permettra.

Comment s'en protéger ?

Limiter l'upload à un type de fichier souhaité (peut être contourné par la double extension)

Renommer le fichier uploader

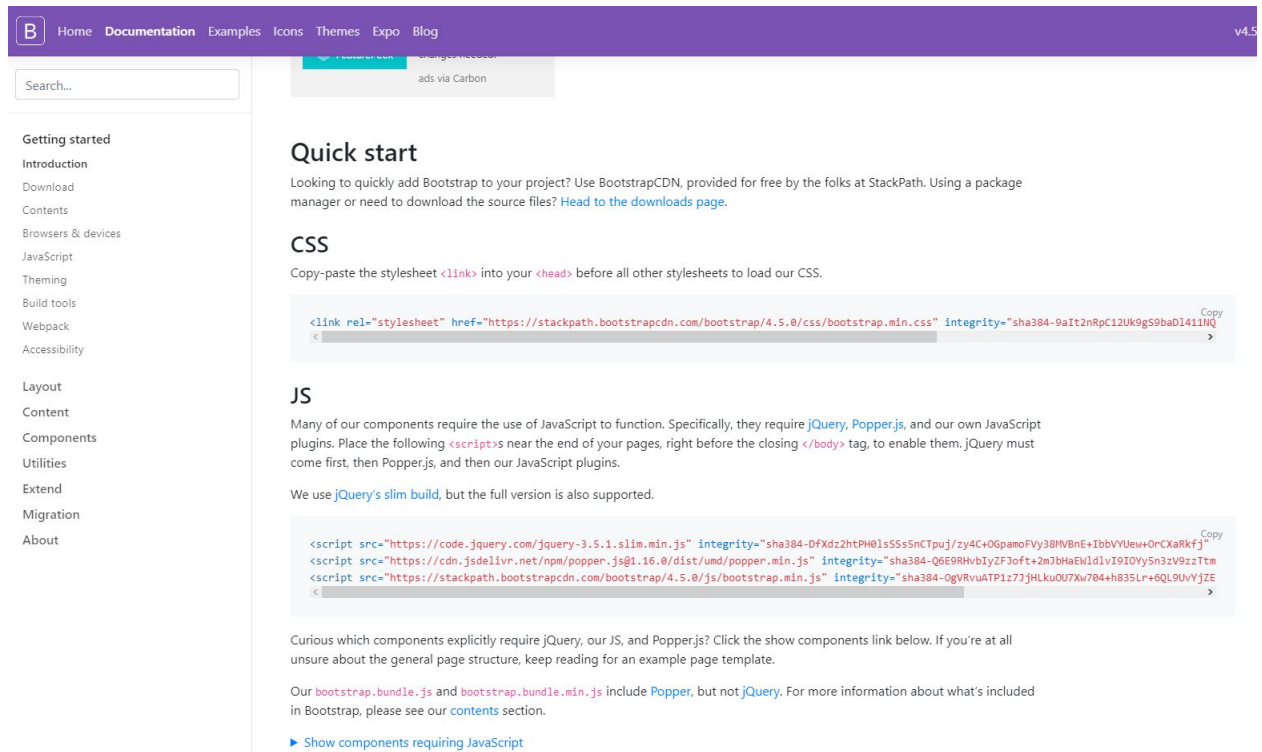
Ne pas stocker le fichier dans un endroit important

Checker la taille du fichier

Pour résumer bien protéger le site est primordial pour la cliente et pour l'intégralité de son salon. Dès multitude de test sont effectués pour tester les failles.

## Recherche sur des sites anglophones

Pour tout type de recherche on se retrouve sur les informations en anglais, Front ou back. Pour utiliser bootstrap les documents sont en anglais.



The screenshot displays the Bootstrap 4.5.0 documentation page. The top navigation bar includes links for Home, Documentation, Examples, Icons, Themes, Expo, and Blog. A search bar is located on the left. The main content area is titled 'Quick start' and provides instructions on how to add Bootstrap to a project using BootstrapCDN. It includes a code snippet for including the CSS file and another for including the JavaScript files. The left sidebar shows the navigation menu with categories like 'Getting started', 'Layout', 'Content', 'Components', 'Utilities', 'Extend', 'Migration', and 'About'.

**Quick start**

Looking to quickly add Bootstrap to your project? Use BootstrapCDN, provided for free by the folks at StackPath. Using a package manager or need to download the source files? [Head to the downloads page](#).

**CSS**

Copy-paste the stylesheet `<link>` into your `<head>` before all other stylesheets to load our CSS.

```
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" integrity="sha384-9aIt2nRc12Uk9gS9baD1411HQ" >
```

**JS**

Many of our components require the use of JavaScript to function. Specifically, they require [jQuery](#), [Popper.js](#), and our own JavaScript plugins. Place the following `<script>`s near the end of your pages, right before the closing `</body>` tag, to enable them. jQuery must come first, then Popper.js, and then our JavaScript plugins.

We use [jQuery's slim build](#), but the full version is also supported.

```
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+06pamofV38MBNtE+IbbVvUew+OrCkARkfj" ><script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E9RHvbIyZF3oft+2mJbHaEWldlvI9IOYy5n3zV9zzTtm" ><script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js" integrity="sha384-OgVRvuATP1z7JHHLKoOU7xw704+h835Lr+6QL9UvVjZE" >
```

Curious which components explicitly require jQuery, our JS, and Popper.js? Click the show components link below. If you're at all unsure about the general page structure, keep reading for an example page template.

Our `bootstrap.bundle.js` and `bootstrap.bundle.min.js` include [Popper](#), but not [jQuery](#). For more information about what's included in Bootstrap, please see our [contents](#) section.

[► Show components requiring JavaScript](#)

Un visuel rapide pour votre projet? Utiliser bootstrapCDN donnez gratuitement par les utilisateurs. Utiliser les packs ou télécharger la source complète. Beaucoup de nos composants utilise JavaScript comme fonction.

Il utilise aussi jquery, popper.js et d'autres plugins. Place le contenu dans les balise `<script>` à la fin de ta page. JQuery est a mettre avant tout dans le script.

## Les infos des pages

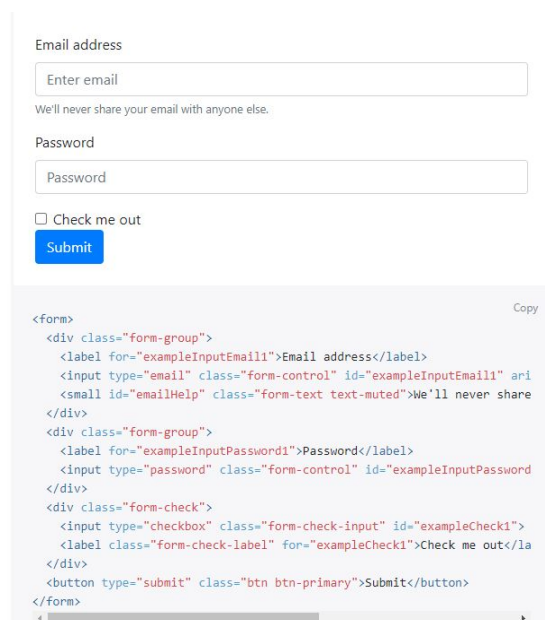
## Pour la page CLIENT

La page client a été créée pour répondre à la demande du client qui lui permet de voir tous ses clients et de les rechercher avec une barre de recherche qui permet de cibler les clients. Dans cette page on retrouve aussi une inscription pour les personnes qui ne peuvent pas s'inscrire comme les personnes âgées. En cliquant sur le client on se retrouve sur la page profil pour voir les détails de l'utilisateur.

Si c'est le jour d'anniversaire d'un client un gâteau apparaît à côté de son nom ce qui permet d'avoir une réduction de 10% pendant 1 semaine.

Je fais le CSS à l'aide de bootstrap qui permet avec des classes préconfigurées d'ajouter des éléments.

Exemple pour les formulaires :



The image shows a Bootstrap form example. It includes an email input field with a label 'Email address', a password input field with a label 'Password', a checkbox labeled 'Check me out', and a blue 'Submit' button. Below the form, the corresponding HTML code is displayed, using Bootstrap classes like 'form-group', 'form-control', 'form-check', and 'btn-primary'.

```
<form>
  <div class="form-group">
    <label for="exampleInputEmail1">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail1" ari
    <small id="emailHelp" class="form-text text-muted">We'll never share
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1">Password</label>
    <input type="password" class="form-control" id="exampleInputPassword
  </div>
  <div class="form-check">
    <input type="checkbox" class="form-check-input" id="exampleCheck1">
    <label class="form-check-label" for="exampleCheck1">Check me out</la
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

## Pour la page STOCK

La cliente a demandé de voir les stocks des objets du salon.

J'ai fais en sorte de voir tous les produits et de pouvoir rechercher des produits avec une barre de recherche pour voir la quantité restante.

Si on clique sur l'objet en question on se retrouve sur la fiche technique et on peut changer la quantité.

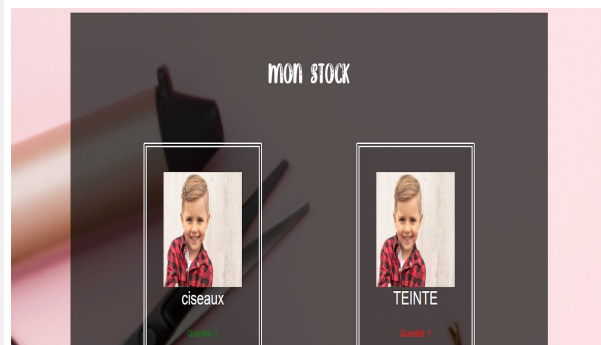
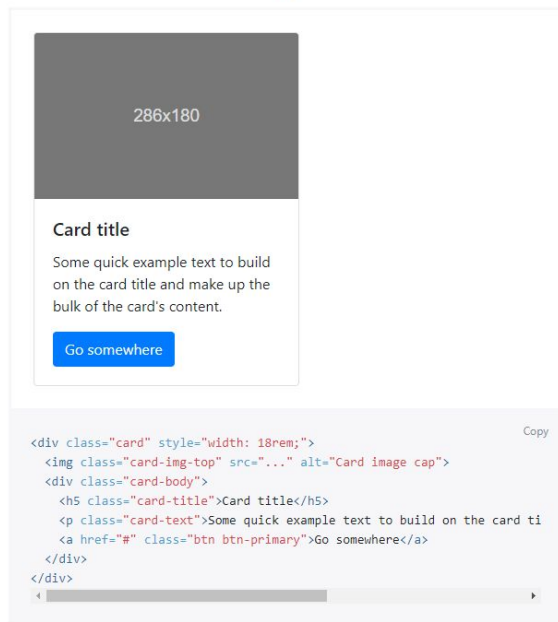
La particularité de mon stock et de changer de couleur en fonction si le stock dépasse un nombre défini par la cliente. Si la couleur est verte c'est qu'il y a assez de stock en boutique, en rouge c'est que la quantité est insuffisante donc elle pourra acheter le nécessaire.

Avec un formulaire on peut créer un produit pour les rajouter dans la base de données et l'afficher.

Je fais le CSS à l'aide de bootstrap qui permet avec des classes préconfigurer d'ajouter des éléments.

Exemple pour les photo du stock :

Below is an example of a basic card with mixed content and a fixed width. Cards have no fixed width to start, so they'll naturally fill the full width of its parent element. This is easily customized with our various [sizing options](#).



## Détail technique back-end

### Pour la page client

J'ai inclus la barre nav pour différencier les page.sg

```
<header class="header1">
...
<?php include("../include/bar-nav.php");?>
</header>
```

Dans l'autre page se trouve :

```
<?php
if (isset($_SESSION['login'])==false)
{
?>
<ul class="nav justify-content-end bg-text-white">
<li class="nav-item"><a class="nav-link text-white active" href="index.php">Home</a></li>
<li class="nav-item"><a class="nav-link text-white active" href="../sources/tarif.php">Tarifs</a></li>
<li class="nav-item"><a class="nav-link text-white active" href="../sources/livreor.php">Livre d'or</a></li>
<li class="nav-item"><a class="nav-link text-white active" href="connexion.php">Connexion</a></li>
<li class="nav-item"><a class="nav-link text-white active" href="inscription.php">Inscription</a></li>
</ul>

<?php
}
elseif(isset($_SESSION['login'])==true)

{
    if($_SESSION['login'] == "vanessa")
    {
?>
<ul id="menu-deroulant" class="nav justify-content-end text-white">
<li class="nav-item"><a class="nav-link text-white active" href="index.php">Home</a></li>
<li class="nav-item"><a class="nav-link text-white active" href="../sources/livreor.php">Livre d'or</a></li>
<li class="nav-item"><a class="nav-link text-white active" href="../sources/tarif.php">Tarifs</a></li>
<li class="nav-item"><a class="nav-link text-white active" href="../sources/profil.php">Profil</a></li>
<li class="nav-item"><a class="nav-link text-white active" href="index.php?deconnexion=true">Déconnexion</a></li>
<li class="list-group-item bg-dark"><a class="nav-link text-white" href="#">Administrateur</a>
<ul class="list-group">
<li class="list-group-item"><a class="nav-link active" href="../sources/client.php">Mes clients</a></li>
<li class="list-group-item"><a class="nav-link active" href="../sources/planning.php">Mon Agenda</a></li>
<li class="list-group-item"><a class="nav-link active" href="../sources/boutiqueprestation.php">Boutique/Paiement</a>
<li class="list-group-item"><a class="nav-link active" href="../sources/stock.php">Mon stock</a></li>
</ul>
</li>
```

J'ai créé la condition qui permet de savoir si l'utilisateur est connecté avec la session que j'ai créé auparavant dans connexion.

J'ai créé une liste <ul> qui me permet de faire la barre de navigation qui va permettre au utilisateur de se déplacer de page en page.

J'ai fais une condition que si l'admin est connecté un menu spécial admin apparaît pour pouvoir gérer le panel admin comme le stock, les clients, le planning.

J'ai créé un formulaire qui va pouvoir inscrire des clients exclusif à l'admin :

```

<form method="post">

    <input type="text" name="nom" required placeholder="Nom"/>
    <input type="text" name="prenom" required placeholder="Prénom"/>
    <input type="email" name="email" required placeholder="Email"/>
    <input type="password" name="mdp" required placeholder="mdp"/>
    <input type="date" name="date" required placeholder="Date"/>
    <input type="text" name="tel" required placeholder="tel"/>

    <br><input type="submit" name="formI" required value="S'inscrire"/>

</form>

```

Par method post ce qui permet de pas voir les informations du formulaire dans la barre du navigateur.

J'ai fais plusieurs input avec différents types :

- type texte : Pour pouvoir écrire du text.
- Type email : Pour pouvoir écrire exclusivement une email avec @.
- Type password : Pour pouvoir écrire le mot de passe sans qu'il soit visible.
- Type date : Pour mettre que des dates et pas de text.
- Type submit : Pour pouvoir envoyer le formulaire et vérifier les required.

A quoi sert le required : Il permet de ne pas valider le formulaire si cette input n'est pas compléter.

```

if(isset($_POST['formI']))
{
    $nom=$_POST['nom'];
    $prenom=$_POST['prenom'];
    $email=$_POST['email'];
    $date=$_POST['date'];
    $mdp=$_POST['mdp'];
    $tel=$_POST['tel'];

    $login=$_POST['login'];
    $requete1="SELECT * FROM utilisateurs WHERE login='$nom'";

    $requete1Q=mysqli_query($connexion,$requete1);
    $requete1R=mysqli_fetch_all($requete1Q);

    if (empty($requete1R) )
    {
        include("../include/mdpaleatoire.php");
        $hash = password_hash($mdp, PASSWORD_BCRYPT, ['cost' => 12]);
        $requete2="INSERT INTO utilisateurs VALUES(NULL,'$nom', '$nom', '$prenom', '$email', '$hash', '$date', $tel, NULL, NULL)";
        var_dump($requete2);

        $requete2Q=mysqli_query($connexion,$requete2);
        echo "votre inscription est confirmer";

    }
    else
    {
        echo "login déjà existant";
    }
}

```

J'ai défini les \$\_POST pour faire des raccourci \$xxx=\$\_POST['Xxx'].

J'ai créé une condition avec le `if(isset($_POST['forml']))` qui me permet de faire l'action quand il clique sur le bouton s'inscrire.

J'ai fait une requête SQL qui me permet de voir si il y a un utilisateur avec le nom inscrit dans l'input dans la base de donnée. J'ai récupéré le résultat avec un `fetch all`. Qui me permet de récupérer les informations sous forme de tableau.

J'ai fait une condition pour détecter si le résultat est vide . `if (empty($requete1R) )`

Si le résultat est vide je fais une requête sql qui permet d'insérer les informations dans la base de données. Je crypte le mot de passe avec la fonction `hash` qui permet de transformer le mot de passe pour qu'il ne soit plus visible.

Si le résultat est rempli je lui affiche que le login existe déjà.

J'ai fait en sorte de savoir combien il y a de client inscrit sur le site pour cela j'effectue une requête qui compte le nombre d'id dans la table utilisateurs.

```
<?php
$count="SELECT count(id) FROM utilisateurs";
$countQ=mysqli_query($bdd,$count);
$countR=mysqli_fetch_all($countQ);
$CR=$countR[0][0];

echo "<div id =\"compteclient\"> Il y a $CR clients inscrits </div>";
```

Pour la barre de recherche j'ai fait une requête `SELECT * FROM utilisateurs WHERE nom LIKE '$R%' ORDER BY id DESC`.

```
if(isset($_POST['search']))
{
    $R=$_POST['search'];
    $requete="SELECT * FROM utilisateurs WHERE nom LIKE '$R%' ORDER BY id DESC";
    $queryR=mysqli_query($bdd,$requete);
    while($data= mysqli_fetch_assoc($queryR))
    {
        include("../include/infouseradmin.php");
    }
}
else
{
    $user="SELECT * FROM utilisateurs";
    $reservQ=mysqli_query($bdd,$user);

    while($data= mysqli_fetch_assoc($reservQ))
    {
        include("../include/infouseradmin.php");
    }
}
?>
```

Un Like dans une requête permet de trouver un mot qui commence par la lettre sélectionnée



avant le %.

Puis j'ai fais un ORDER BY qui indique que l'on choisi l'ordre d'affichage DESC ou ASC selon les besoins.

Dans la boucle du while j'ai fais un include qui me permet d'afficher les informations de l'utilisateur.

```
<?php
$i=0;

$nom=$data['nom'];
$prenom=$data['prenom'];
$date=$data['date'];
$id=$data['id'];
echo "<div id='infouser' class='table table-dark'>";
echo "Nom:&nbsp; $nom &nbsp;";
echo "Prénom:&nbsp; $prenom &nbsp;";
echo "<a href='profil.php?U=$id' target='_blank'><button class='btn btn-light' name='p";
$datejour=date('Y-m-d');

if($date == $datejour)
{
    echo "<img src='../upload/gat.png' style=' height: 50px; width: 100px;'>";
}
echo "</div>";

$i++;
```

Pour expliquer ce morceau de code je vous explique que dans la boucle le \$data change en fonction du nombre de boucle qui est fait.

Du coup j'affiche toute les informations avec les echos.

Ensuite j'ai créé un condition qui permet de regarder si le jour de son anniversaire dans la base de donnée et la même qu'aujourd'hui j'ai fais apparaître un gâteau pour signaler à l'admin que c'est l'anniversaire du client ce qui lui fait une remise de 10% sur la somme à payer.

Si l'admin a pas utilisé la barre de recherche affiche tous les clients .

**La page stock :**



```

if (!isset($_GET['p']))
{
    ?>

    <form id="autostock" method="post">
    <label class="title2">Le moteur de recherche du stock</label>
    <input class="form-control mdb-autocomplete" name="search" type="text" placeholder="Recherche par type" id="searchBox">
    </form>
    <div id="response"></div>
    <button type="button" class="btn btn-dark" data-toggle="modal" data-target="#exampleModal6">
    Générer mon stock
    </button>
    <!-- Modal -->
    <div class="modal fade" id="exampleModal6" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">
    <div class="modal-dialog" role="document">
    <div class="modal-content">
    <div class="modal-header">
    <h5 class="modal-title" id="exampleModalLabel">Gestion stock</h5>
    <button type="button" class="close" data-dismiss="modal" aria-label="Close">
    <span aria-hidden="true">&times;</span>
    </button>
    </div>
    <div class="modal-body">
    .....
    <?php
    include("../include/imagestock.php");
    ?>

```

Pour commencer j'ai fait une condition s'il existe pas de GET j'ai créé un formulaire qui permet de récupérer les informations et la photo à inscrire dans la base de données pour le stock.

Dans l'include voici le code :

```

<?php
if(isset($_POST['description']) & isset($_POST['nom']) )
{
    $dossier = '../upload/stock/';

    $fichier = basename($_FILES['mon_fichier']['name']);
    $taille_maxi = 1000000;
    $taille = filesize($_FILES['mon_fichier']['tmp_name']);
    $extensions = array('.png', '.gif', '.jpg', '.jpeg');
    $extension = strrchr($_FILES['mon_fichier']['name'], '.');
    $chemin=$fichier;
    var_dump($chemin);

    //Début des vérifications de sécurité...
    if(!in_array($extension, $extensions)) //Si l'extension n'est pas dans le tableau
    {
        $erreur = 'Vous devez uploader un fichier de type png, gif, jpg, jpeg, txt ou doc...';
    }
    if($taille>$taille_maxi)
    {
        $erreur = 'Le fichier est trop gros...';
    }
    if(!isset($erreur)) //S'il n'y a pas d'erreur, on upload
    {
        //On formate le nom du fichier ici...
        $fichier = strtr($fichier,
            'AAAAACEEEEEIIIIIOOOOOUUUUYaaaaaaceeeeiifiiioooooouuuuyy',
            'AAAAACEEEEEIIIIIOOOOOUUUUYaaaaaaceeeeiifiiioooooouuuuyy');
        $fichier = preg_replace('/([^\a-z0-9]+)/i', '-', $fichier);
        if(move_uploaded_file($_FILES['mon_fichier']['tmp_name'], $dossier . $fichier)) //Si la fonction renvoie TRUE, c'est que ça a
        {
            try
            {
                $bdd = mysqli_connect("localhost", "root", "", "ciaracut");
            }
            catch(Exception $e)
            {
                die('Erreur : '.$e->getMessage());
            }

            $nom=$_POST['nom'];
            $image=$chemin;
            $description=$_POST['description'];
            $req = $bdd->prepare("INSERT INTO `produit`(`nom`, `image`, `description`) VALUES ('$nom','$image','$description')");

```

Ce code me permet de récupérer un fichier de l'ordinateur et le mettre dans un fichier sur

serveur.

Il faut lui indiquer :

- le nom de l'input file
- la taille du fichier.
- Le type de fichier

J'ai fais des vérifications de la taille, du type, des caractères spéciaux puis j'utilise `move_uploaded_file` qui permet de déplacer un fichier dans un dossier prévu à cet effet.

Puis j'ai fais une requête avec un insert pour ajouter les informations fournies à la base de données. En cas d'erreur il m'affiche echec de l'upload.

```
<label class="title2">Mon stock</label>
<div id="produit">
<?php
$all="SELECT * FROM stock inner join produit on stock.id_produit = produit.id";

$allQ=mysqli_query($bdd,$all);

// echo"<div id='bloccardstock' class='card' style='width: 15rem;'>";
echo "<div>";

while($data= mysqli_fetch_assoc($allQ))
{
    $i=1;
    $did=$data['id'];
    $img=$data['image'];
    $dnp=$data['nom'];
    $cmb=$data['quantiteproduit'];

    echo" <div id='bloccardstock' class='card' style='width: 15rem;'>";
    echo "<a href='stock.php?p=$did'><img id='photostock' src='../upload/stock/$img'></a>";
    echo" <div class='titrestock'>";
    echo "<h1 class='dnp'>Nom:$dnp</h1><br>";
    echo "</div>";

    if ($cmb < 3)
    {
        echo"<div id='insufisant'>";
    }
    else
    {
        echo"<div id='stockquant'>";
    }

    echo " Quantité: $cmb";
    echo "</div>";
    echo "</div>";

    $i++;
}
echo"</div>";
```

J'effectue la requête `SELECT * FROM stock inner join produit on stock.id_produit = produit.id` qui permet de récupérer les informations avec les `id_produit` dans `stock` et l'`id` dans `produit` identique.

Ensuite je fais un `while` qui me permet d'afficher les informations si le nombre de quantité est

inférieur à 3 la quantité se transformera en rouge grâce au CSS.

Si l'admin clique sur un objet je fait la requete : "SELECT \* FROM `stock` inner join produit on stock.id\_produit = produit.id where stock.id\_produit = '\$idproduit'" Qui me permet de pouvoir chercher l'objet avec le GET qui correspond avec l'id de l'objet pour avoir les infos.

```
<form class="form-group" id="quantite" method="post">
  <div>
    <label>Quantité:</label>
    <input name="nombre" class="form-control" type="number">
    <br>
  </div>
  <div>
    <button name="stockV" type="submit" class="btn btn-dark">Valider</button>
  </div>
</form>
<?php
$i++;
if(isset($_POST['stockV']))
{
    $qtt=$_POST['nombre'];
    $idproduit=$_GET['p'];
    $update="UPDATE stock SET quantiteproduit = $qtt where id_produit = $idproduit";
    $updateQ=mysqli_query($bdd,$update);
    header("refresh:0");
}
}
```

Je fais une condition pour que quand l'admin click sur modifier la quantité change et actualise la page.

## Un jeu d'essai

Ordre	Scénarios	Cas de test : Valeurs	Cas de test : Résultats attendus
1	Scénario nominal	FORM: image: titre: description:	Insert dans la bdd
2	Premier cas d'échec	FORM: titre: description:	Empêche la suite de la requête
3	Deuxième cas d'échec	FORM : image: pas de bon format  titre: description:	Empêche la suite de la requête

Ce tableau représente plusieurs possibilités selon le choix de l'utilisateur .

## Les compétences acquises

Avec le projet professionnel j'ai acquis des compétences pour écouter la demande de la cliente.

A faire des tests à chaque étape de la création du site.

vérification précise exemple :

Stock : Bien afficher et modifier la quantité des produits.

Client : Mettre à jour en temp réel les client et vérifier si c'est son anniversaire.

## Compétences WEB

### PHP :

Pour faire les conditions nécessaires au fonctionnement du site.

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

**SQL :** Pour faire des requêtes à la base de données.

Maquettage : Qui nous a permis à convaincre le client de commencer le projet.

GitHub : Pour sauvegarder et partager à mes collaborateurs.

CSS/BOOTSTRAP : Qui permet de désigner le site avec facilité.

## Débogage du site en général

Une fois le merge de toutes les pages effectué, on fait le test global pour repérer les éventuels bugs ou des fonctionnalités oubliées.

On commence par le côté client pour finir sur le coté admin.

La première partie : Le test se fait sans être connecté pour voir si on peut accéder à des accès interdits et si la connexion et l'inscription fonctionne correctement.

Du coup on a accès à la connexion / Inscription / tarifs/ Livre d'or (visuel)

Toute les autres pages ont bloqué l'accès.

Pour la deuxième partie : On se connecte en tant qu'utilisateur sans grade pour débloquent le profil / Livre d'or pour écrire un commentaire.

Avec la fonction qui vérifie que l'utilisateur a déjà payé au salon et donc inscrit dans la base de données pour éviter que tout le monde commente de mauvaise information qui peut dénigrer le salon.

Pour la troisième partie: On se connecte en tant qu'admin ce qui lui donne tout les pouvoirs pour gérer son site.

les pages :

- Client : Qui permet de gérer tous les clients et de voir le profil des clients.
- Stock : Qui permet de gérer tous les produits en stock, créer et modifier.
- Planning: Qui permet de créer des rendez vous et de voir son planning.
- Paiement: Pour encaisser les clients via un panel.

## Les rendez-vous client

Nous avons échangé avec la cliente pour la première fois par téléphone pour qu'elle nous explique l'idée de son projet et les fonctionnalités attendues.

Elle voulait un site de réservation classique. Après notre premier échange, je lui ai fait une proposition avec une maquette, une première version sans design, avec les fonctionnalités de base.

Après avoir vu notre travail, elle a validé la maquette et on a commencé à développer le projet.

Avec plusieurs échanges via meet, pendant le confinement, elle a changé d'avis et elle voulait gérer son site et ses clients, ce qui nous a amené à faire un panel admin pour répondre à la demande.

L'utilisateur peut seulement s'inscrire, changer son profil ou mettre un commentaire pour les clients qui ont déjà passés commande.

## Le Front

Les différentes pages ont été effectuées avec le CSS et BOOTSTRAP.

C'est quoi bootstrap ?

BOOTSTRAP est un framework front qui permet de faire du responsive et du personnalisé pour chaque projet.

Comment marche bootstrap ?

Rapide et efficace il se découpe en plusieurs morceaux de code déjà écrit qu'on peut retrouver sur le site officiel.

Il suffit de mettre des classes prédéfinies sur l'élément qu'on veut pour pouvoir le styliser comme l'on veut.

Pour l'intégrer au page il faut mettre le script cdn :

Pour le css :

```
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"
integrity="sha384-9alt2nRpC12Uk9gS9baDl411NQAпFmC26EwAOH8WgZl5MYxPfc+NcPb1dK
Gj7Sk" crossorigin="anonymous">
```

Pour le js :

```
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaR
kfj" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
integrity="sha384-Q6E9RHvblyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooA
o" crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"
integrity="sha384-OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR0J
KI" crossorigin="anonymous"></script>
```

Une fois mis dans le head on peut utiliser le framework.

Exemples :

Exemple pour les photos on peut utiliser un modal qui permet de les afficher proprement et responsive avec un style définit :

```
<div class="card" style="width: 18rem;">
  
  <div class="card-body">
    <h5 class="card-title">Card title</h5>
    <p class="card-text">Some quick example text to build on the card title and make up the bulk
of the card's content.</p>
    <a href="#" class="btn btn-primary">Go somewhere</a>
```

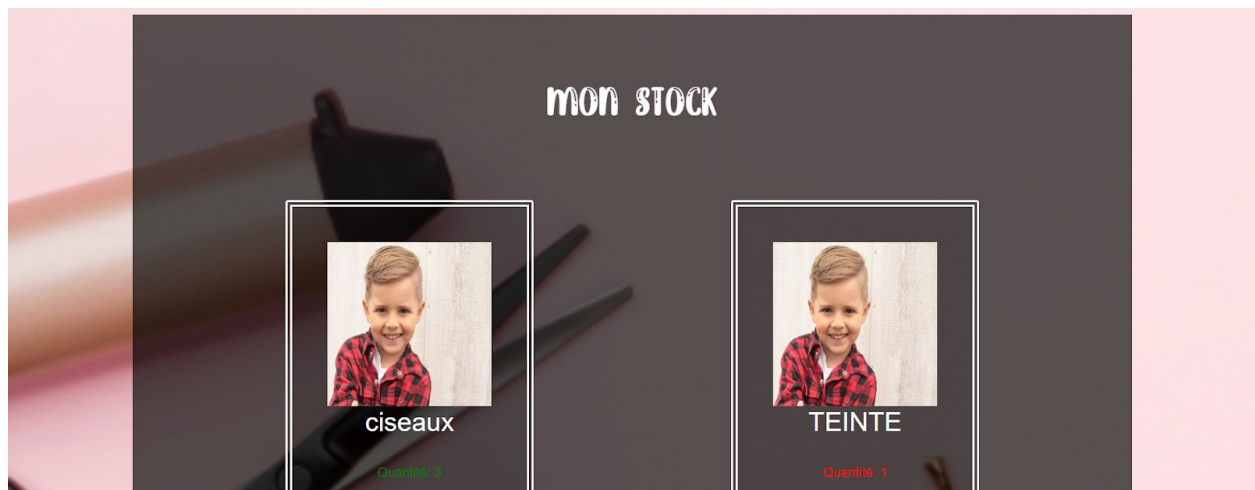
</div>  
</div>

Les classes utilisés sont créés pour avoir un style déjà créé il suffit de regarder la librairie mis à disposition et de choisir son style pour mettre les bonnes classes pour des résultats rapide et efficace.

Les différentes catégories :

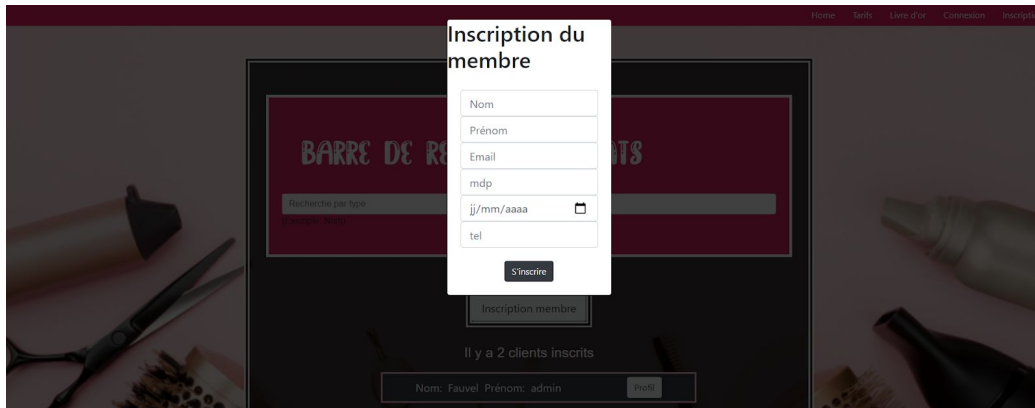
Alerts	,Badge,	Breadcrumb	Buttons,Button	group,
Card	Carousel	Collapse,	Dropdowns	Forms,
Input group,	Jumbotron	List group,	Media object,	Modal
Navs	,Navbar	,Pagination	Popovers	Progress
Scrollspy	Spinners	Toasts	Tooltips	

J'ai utilisé pour la page stock le card et le Forms qui m'a permis d'avoir ce résultat :





Pour la page client j'ai fait un form aussi pour le rendre plus beau pour la cliente qui creer des comptes. Avec un pop up qui apparaît en cliquant sur le bouton inscription membre.



### Et le css dans tout cela?

Sans faire de bootstrap on peut très bien personnaliser son site avec le fichier .css qui permet de créer un design avec des classes et différents attribut. On peut très bien changer la couleur du background , tu text ou créer des div pour faire d'autre style.

### et le responsive ?

On peut faire du responsive en utilisant des @media query pour différente taille d'écran.

```
/* Set the background color of body to tan */
```

```
body {  
  background-color: tan;  
}
```

```
/* On screens that are 992px or less, set the background color to blue */
```

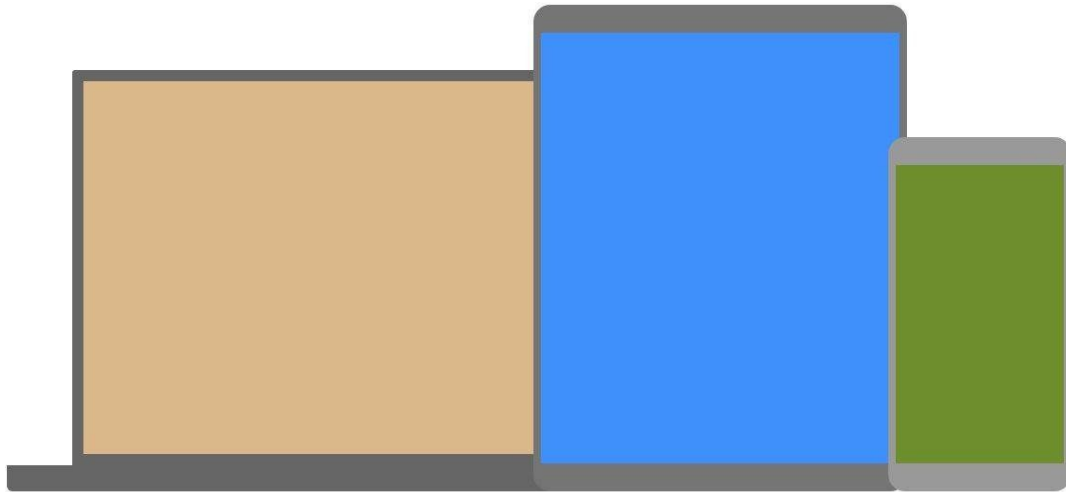
```
@media screen and (max-width: 992px) {  
  body {  
    background-color: blue;  
  }  
}
```

```
/* On screens that are 600px or less, set the background color to olive */
```

```
@media screen and (max-width: 600px) {  
  body {  
    background-color: olive;  
  }  
}
```

```
}  
}
```

ce qui permet d'avoir le résultat suivant :



Cela permet de s'adapter à tous les types : en étant précis sur les pixel ou autre.

## Conclusion

Pour le projet CIARA CUT, nous avons avancé en équipe malgré la situation particulière du coronavirus. On a finalisé le projet et la cliente nous a fait part de sa satisfaction du résultat.

On va héberger le site sur OVH après les dernières vérifications de sécurité.

Cette période m'a appris à être structuré et m'adapter à toutes les situations.

Le travail via meet a été très bénéfique pour l'équipe.