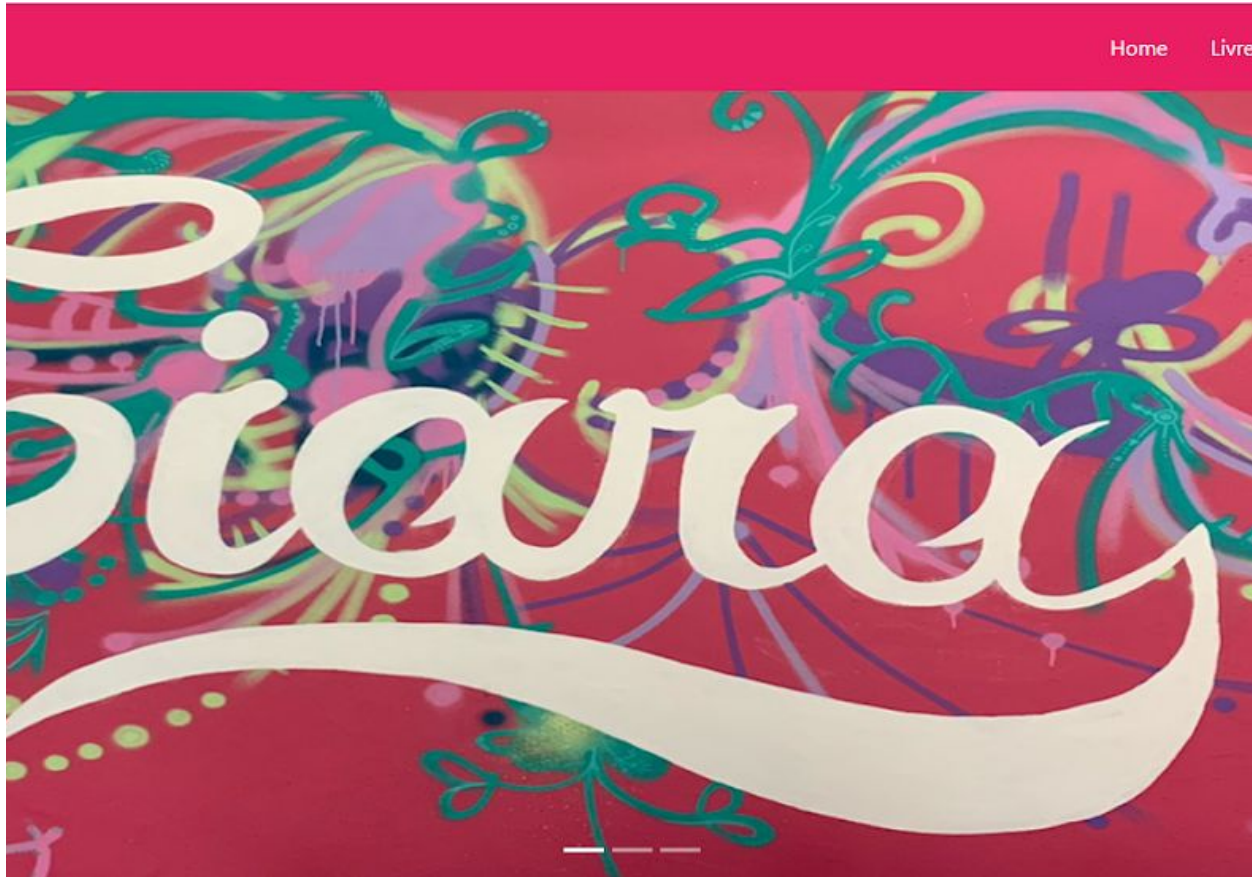


CiaraCut

Projet pro de réservation pour salon de coiffure



REBOLI

SOMMAIRE

1 Introduction	p3
1.2 Présentation Entreprise	p3
1.1 Présentation personnel	p3
1.1 Présentation d'équipes	p3
1.4 formation	p
1.5 Comment organisé le projet	p
1.6 Détail du projet pro	p
1.7 Choix techniques et méthodologie	
1.8 base de donnée	
1.9 cahier des charges	p
1.6 Sécurité	p
1.7 Recherche sur Site anglophone	p
2 Partie Front-end	p
2.1 client	p
2.2 stock	p
3 Partie back-end	p
3.1 base de données	p
3.2 client	p
3.3 stock	p

INTRODUCTION

Je suis là pour expliquer mon projet pro pour le client ciaracut qui a demandé un site de réservation et de panel admin pour pouvoir gérer c'est client .

Ainsi qu'une page stock pour pouvoir vérifier les stock en temps réel pour pouvoir gérer 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 Ciaracut est un salon de coiffure tenu par une personne qui fait sa depuis ... et qui écrit sur un papier c'est compte, son stock et elle a voulu se moderniser avec un site internet dynamique avec une page vitrine.

Présentation personnel

Je suis un étudiant de la plateforme qui est une formation pour apprendre le code HTML / CSS / PHP / SQL / JAVASCRIPT. Je suis un ancien intérimaire qui travailler dans le domaine de l'industrie en tant que monteur tuyauteur ce qui permet de monter ou démonter des brides. Après une décision personnel je me suis reconverti dans le web ce qui m'a permit de me présenter aujourd'hui au RNCP . Étant un gamer je me suis toujours demandé comment fonctionne un jeu et j'ai découvert un univers vaste et très intéressant.

Présentation d'Équipes

Pour ce projet pro j'ai travailler en équipe :

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

THOMAS SAUSSOL pour Connexion , Inscription , livre d'or.

ANTHONY REBOLI pour Produit , stock , client , fidélité et debug general.

Formation

La Plateforme coding school est une formation qui permet d'apprendre le HTML/PHP/SQL/JAVA SCRIPT/BOOTSTRAP/CSS.

La Formation nous apprend à rechercher par nous même les solution a nos problemes.

COMMENT EST ORGANISER LE PROJET

- La première étapes est de demandé au client l'idée du site et les fonctionnalités nécessaire pour son projet .
- Faire le dispatch des tâches pour que tout le monde est une partie a travailler et que on n'a un plan pour commencer les maquettes.
- Le maquettagage des pages en démonstration pour voir les fonctionnalité general.
- Le maquettagage de la base de donné avec genmymodel.
- La programmation du site.

LES DIFFERENT SITE CREER

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

E-Commerce Boutique de thé.

E-Commerce Boutique de liseuse.

Réservation camping.

Réservation paintball.

Jeux clicker.

CiaraCut projet pro

DÉTAIL DU PROJET PRO

Pour le projet CiaraCut 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 n'a créer un github pour sauvegarder et partager nos pages.

Pour bien placer l'organisation des page on n'a utilisé l'application TRELLO qui permet de gerer des evenement en équipe pour suivre le déroulement des étapes.

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

Choix techniques 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 branche, une pour moi et l'autre pour les merges.

LES BASES DE DONNÉE

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

PHOTO DE LA MAQUETTE

EXPLICATION

Puis je la créer via PHPmyadmin

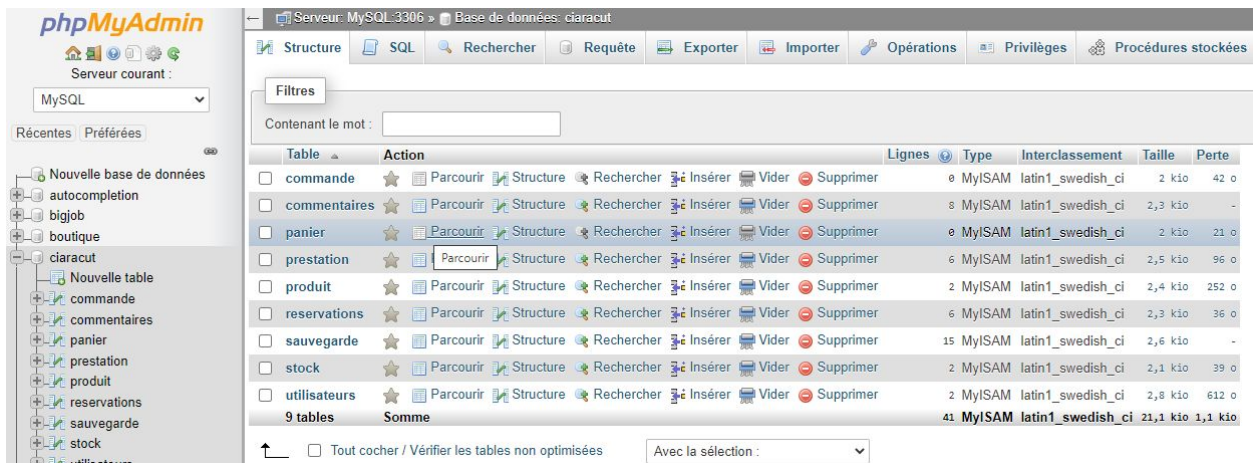


Table	Action	Lignes	Type	Interclassement	Taille	Perte
commande	Parcourir Structure Rechercher Insérer Vider Supprimer	0	MyISAM	latin1_swedish_ci	2 kio	42 0
commentaires	Parcourir Structure Rechercher Insérer Vider Supprimer	8	MyISAM	latin1_swedish_ci	2,3 kio	-
panier	Parcourir Structure Rechercher Insérer 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

Je crée une nouvelle base de données puis je créer plusieurs tables stock, produit, utilisateurs.

Cahier des charges

- Le look du site est aussi à la demande du client (Style girly Trois ton de couleurs principales dont le noir ,blanc,rose fushia).
- Un code couleur de référence a été établie 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 complète de ses clients ainsi que ses derniers rendez-vous et dernières prestations.
- Elle veut pouvoir supprimer les messages indésirables du livre 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 veillé 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.

Accès au post commentaire livre or autorisé que si paiement de la prestation effectué

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

Recherche sur des site anglophone

Pour tout type de recherche on se retrouve sur les informations en anglais , **Front** ou **BACK** .

Pour utiliser bootstrap les doc sont en anglais.

The screenshot shows the Bootstrap 4.5.0 documentation page. The top navigation bar is purple with links: Home, Documentation, Examples, Icons, Themes, Expo, Blog. A search bar is on the left. The left sidebar contains a table of contents with categories like Getting started, CSS, JS, and more. The main content area is titled 'Quick start' and includes sections for 'CSS' and 'JS'. The 'CSS' section shows a code snippet for including Bootstrap CSS with an integrity hash. The 'JS' section shows code snippets for including jQuery, Popper.js, and Bootstrap JS, also with integrity hashes. A 'Show components requiring JavaScript' link is at the bottom.

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="sha384-9aIt2nRc12Uk9gS9baD1411NQ" >
```

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+OGpamoFVy38MVBnE+IbbVYUew+OrCxaRkfj" ><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-OgVRvuATP177JHHLKuOU7Xw704+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? Utilisez bootstrapCDN donnez gratuitement par les utilisateur. Utilisez le pack ou téléchargez la source complète. Beaucoup de nos composants utilisent JavaScript comme fonction.

Il utilise aussi jQuery, Popper.js et d'autres plugins. Place le contenu dans les balises `<script>` à la fin de la page. jQuery est à mettre avant tout dans le script.

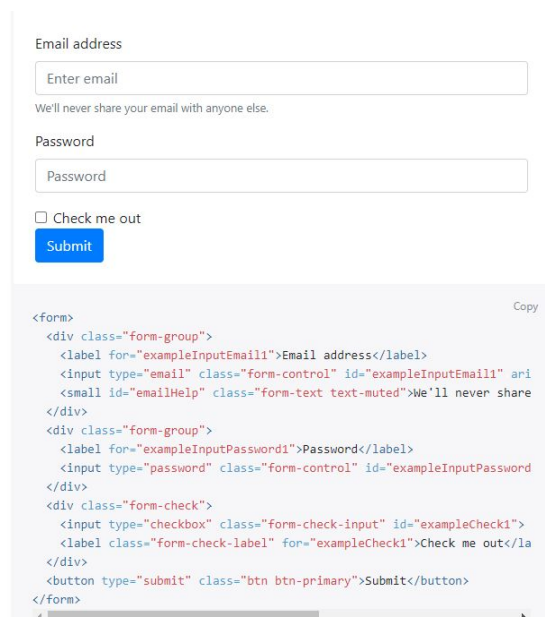
Pour la page CLIENT

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 l'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 address' field with a placeholder 'Enter email' and a small text note 'We'll never share your email with anyone else.' Below it is a 'Password' field. At the bottom, there is a checkbox labeled 'Check me out' and a blue 'Submit' button. The form is styled with Bootstrap classes like 'form-group', 'form-control', and 'btn-primary'.

```
<form>
<div class="form-group">
  <label for="exampleInputEmail">Email address</label>
  <input type="email" class="form-control" id="exampleInputEmail" 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 objet du salon.

J'ai fait en sorte de voir tout les produit et de pouvoir rechercher des produit 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.

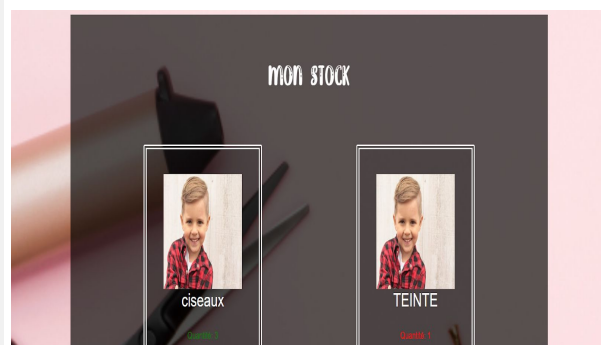
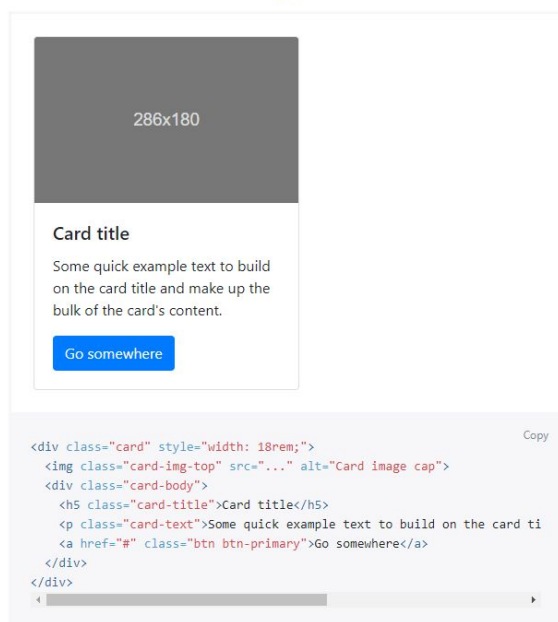
La particularité de mon stock et de changer de couleur en fonction si le stock dépasser un Nombre défini par la cliente . Si la couleur est verte c'est qu'il y a assez de stock en boutique , En rouge c'est que la quantité est insuffisante donc elle pourra acheter le nécessaire.

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

Je fais le CSS à l'aide de bootstrap qui permet avec des classe 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 include la barre nav pour différencier les page.

```
<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>
</ul>
```

Je créer la condition qui permet de savoir si l'utilisateur est connecter avec la session que j'ai créer au par avant dans connexion.

Je creer une liste qui me permet de faire la barre de navigation qui va permettre au utilisateur de se déplacer de page en page.

Je fait 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.

Je crée 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.

Je fait plusieurs input avec différent type :

- 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 date et pas de text.
- Type submit: Pour pouvoir envoyer le formulaire et vérifier les required.
-

A quoi sert le required : Il permet de pas valider le formulaire si cette input 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 confirmée";

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

```

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

Je crée une condition avec le if(isset(\$_POST['formI'])) qui me permet de faire l'action quand il clique sur le bouton s'inscrire.

Je fais une requête sql qui me permet de voir si il y a un utilisateur avec le nom rentré dans l'input dans la base de données. Je récupère 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 infos 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.

Je 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 je 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 je fait un ORDER BY qui indique qu'on choisit l'ordre d'affichage DESC ou ASC selon les besoins.

Dans la boucle du while je fait un include qui me permet d'afficher les infos de

l'utilisateurs.

```
<?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 bout de code je vous explique que dans la boucle le \$data change en fonction du nombre de boucle qui la fait.

Ducoup j'affiche toute les informations avec les echos.

Ensuite je créer un condition qui permet à regarder si le jour de son anniversaire dans la base de donnée et la même qu'aujourd'hui je fait apparaitre un gâteaux pour signaler à l'admin que c'est anniversaire du client ce qui lui fait une remise de 10% de la somme à payer.

Si l'admin a pas utilisé la barre de recherche s'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 je fait une condition si il existe pas de GET je créer un formulaire qui permet de recuperer les info et la photo a inscrire dans la base de donnée 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,
            'AAAAACeFfEIIIIIOO0000UUUYYaasaaaceeeiiiioooooouuuuyy',
            'AAAAAACeFfEIIIIIOO0000UUUYYaasaaaceeeiiiioooooouuuuyy');
        $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 .

Je fait des vérification de la taille , du type , des caractère spécial puis j'utilise `move_uploaded_file` qui permet de déplacer un fichier dans un dossier prévu à cette effet.

Puis je fait une requete avec un insert pour ajouter les information fournis à la base de donnée. En cas d'erreur il me marque 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 fais la requête : `"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 clique sur modifier la quantité change et actualise la page.