

Salon de coiffure

“Ciara cut”



Sommaire

1/ Présentation personnelle.	p.1
1.1/ Présentation équipe.	p.1
1.2/ Présentation Entreprise.	p.2
1.3/ Connaissance Informatiques.	p.3
1.4/ Cahier des charges.	p.4
1.5/ Organisation et répartition du travail .	p.5
1.6/ Sécurité.	p.6
1.7/ Description d'une situation de travail ayant nécessité une recherche à partir de site anglophone.	p.7
2/ Partie Front-end (index,boutique,réservations).	p.17
3/ Partie back-end (réservations).	p.22
3.1/ Partie back-end (boutique).	p.24
Conclusion .	p.30

1/ Présentation personnelle :



Bonjour je m'appelle **Grégory Fauvel** J'ai suivi une formation Coding School dans le domaine du développement web à l'école la Plateforme à Marseille .Grâce à cette formation j'ai acquis et compris les bases des langages du code comme le **HTML,CSS,PHP,MYSQL,Javascript,Jquery.**

Curieux et passionné, j'ai trouvé une reconversion professionnelle qui me convient parfaitement.
Maintenant je souhaite en faire mon futur métier.

1.1/ Présentation de l'équipe :



Le projet "Ciarat cut" a été réalisé par Anthony Reboli pour les parties (clients, stock et fidélité). Thomas Saussol a réalisé les parties livreor ainsi que l'inscription et la connexion du site .Ma partie travaillée sont l'index du site, la réservation ainsi que la boutique personnelle du site .

Page: 1



1.2 Présentation de l'entreprise :



Le client travaillant seul dans son entreprise veut une partie administrateur gérée par elle même pour faciliter la gestion globale de son entreprise.

Le projet "Ciarat cut " est un site de coiffure vitrine et dynamique avec une partie gestion Administrateur personnelle qui inclut une gestion de client avec leur profil leur trois derniers rendez-vous au salon ainsi qu'un fichier client, réservation de rendez-vous modifiable, effaçable ,le client peut entrer pas plus de trois rendez-vous par heure, boutique prestation avec chiffre d'affaire en espèce ou en chèque de la semaine en cours, gestion de stock des produits de l'entreprise avec quantité affiché rouge si le produit est inférieur a 3 puis vert si c'est supérieur à 3 .

Tout le site a été conçu à la demande du client et de ses exigences.

Le site est composé d'une partie vitrine qui met en avant le profil du client, elle veut mettre en avant ses forfaits, un slide avant/après de ses différentes prestations, son tableau de tarifs visible par les clients, un livre or du site ainsi que l'actualité de sa page professionnelle Facebook sur la page d'accueil du site.

1.3 Connaissances informatiques :

Languages HTML et CSS.

J'ai utilisé le **HTML/CSS** comme structure des pages à développer et pour l'intégration. La partie front-end du site a été conçu en **HTML et CSS avec la méthode flexbox**.

Languages PHP My SQL et javascript.

Le dynamique des pages est codé principalement en **PHP** .J'ai fais des modifications back-end en php comme par exemple la barre de navigation suivant l'utilisateur connecté. La partie réservation du site est codé en **javascript ,jquery,ajax**.

L'utilisation du framework Bootstrap

Toutes les pages du site sont conçues avec le **framework Bootstrap**.J'ai réutilisé la documentation responsive et pour l'interaction utilisateur grâce au langage.

Les **formulaires** du sites sont dans des **popups** en bootstrap avec la méthode **Modal**.

Le choix des couleurs ainsi que la disposition des éléments dans le site sont définis par le client. Les slides utilisés dans l'index ainsi que sont contenu du site sont en bootstrap

Tout les boutons du site sont en **class button bootstrap**.

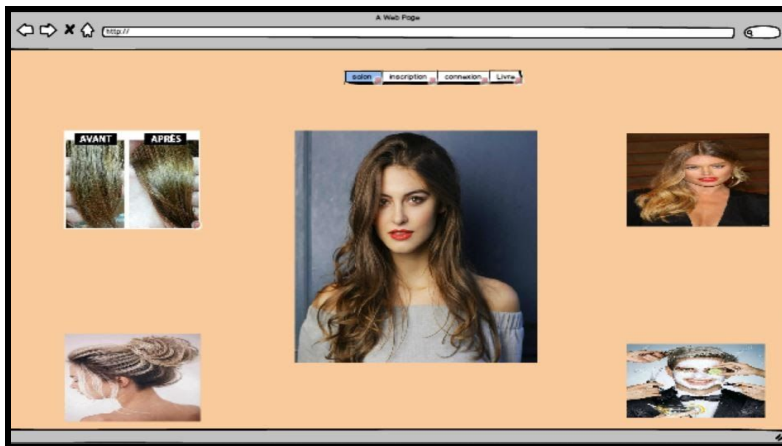
Le **tableau des tarifs** ainsi que **les réservations** du site sont codés en **bootstrap TABLE**. L'affichage des prestations dans la boutique partie administrateur est en méthode **CARD** bootstrap.

1.4 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).
- Pour le 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 la date anniversaire numero de téléphone.
- Une **remise de 10%** à la date d'**anniversaire** de l'utilisateur du site valable pendant 1 semaine.
- Un **accès au livre or** de son site avec un post commentaire que **si le client** a fait une **prestation au magasin** en récupérant le nom et le prénom du client dans la page paiement du site de sa boutique de prestations.
- Affichage des **informations** complète de ses clients ainsi que ses derniers rendez-vous et dernières prestations suivi d'un **fichier client** .
- Tableau des **tarifs** visible dans la barre de navigation et non dans la page d'accueil.
- Affichage **page professionnelle Facebook** dans l'index de son site.
- 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.
- **Boutique personnelle** sorte de **caisse rapide** qui affiche le **chiffre d'affaire** de la semaine en espèce ou en chèque

1.5 Organisation et répartition du travail.

Maquettage d'une application.



Choix techniques et méthodologie Avant de commencer le projet, j'ai fait plusieurs recherches concernant les choix techniques pour des salons de coiffure (ex: Pascal Coste, Jean Claude biguine). Pour le **maquettage** j'utilise

Balsamiq pour le premier visuel du site puis **Gen my model** pour les maquettes et schéma de **base de donnée**. Mon **IDE** est **Sublime Text** pour coder mon site. Ma base de donnée est **PHP MY ADMIN**. J'ai mis en place une méthode de travail. J'ai utilisé Git comme logiciel de versioning. J'utilise principalement **1 branche personnelle** ainsi qu'une **branche Master** pour le projet final. Chaque membre de l'équipe **commit** dans leur branche personnelle pour pouvoir travailler sa partie pour ensuite **push** sur le master une fois le site fini. Une branche **fusion** a été créée pour faire les tests tout au long du développement du site. Trello est l'outil de gestion de projet en ligne qu'on a choisi pour travailler en commun et y déposer nos travaux.

1.6 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 **PREPARE** à mes requêtes en PDO. J'ai mis en place des **htmlspecialchars** à tous les formulaires du site, période de **débug du site**.

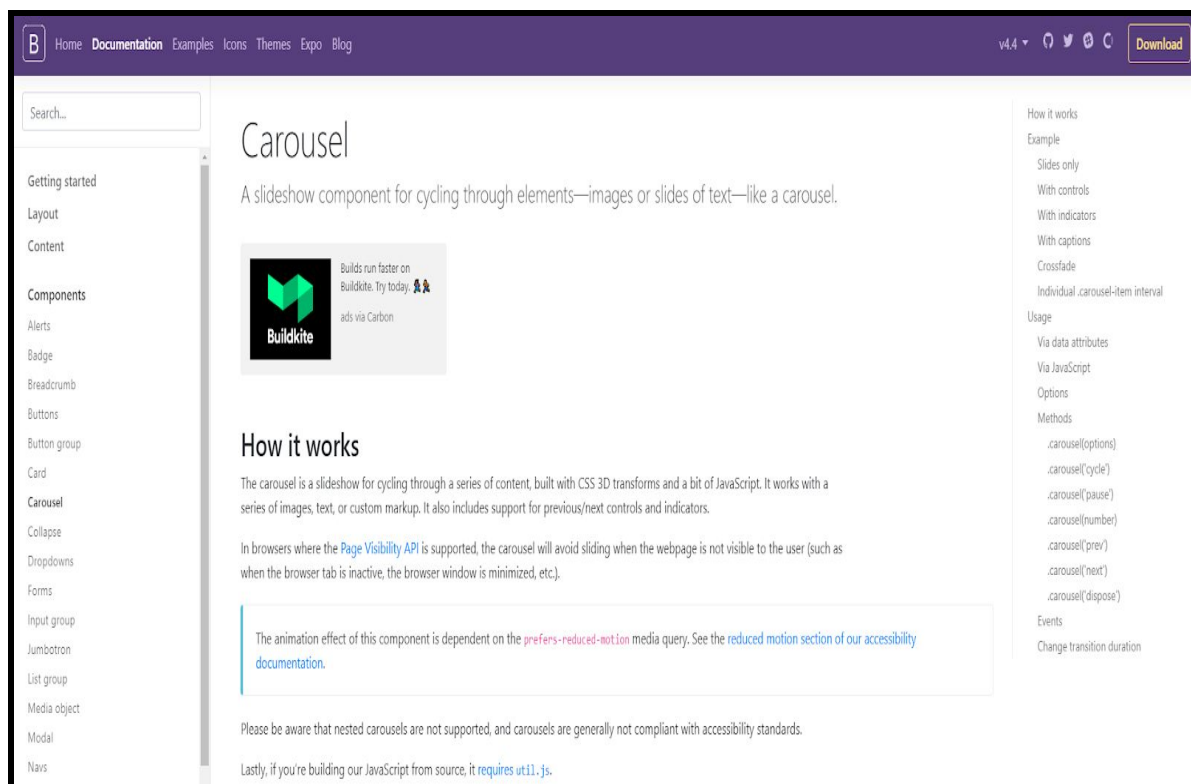
Voici l'arborescence de l'application dans laquelle seront intégrées les différentes fonctionnalités :

Partie Administrateur → Inscription → Connexion → Afficher la page home l'index du site ➤ ➤ Panneau d'administration → Afficher le profil utilisateur → Modifier les paramètres utilisateur → Afficher la page tarifs → Afficher la page livreor ➤ Gestion des clients, ➤ gestion des rendez-vous, ➤ gestion boutique des prestations ➤ gestion du stock.

Partie utilisateurs → Inscription → Connexion → Afficher la page home l'index du site ➤ ➤ Afficher le profil utilisateur → Modifier les paramètres utilisateur → Afficher la page tarifs → Afficher la page livreor → poster un commentaire.

1.7 Description d'une situation de travail ayant nécessité une recherche à partir de site anglophone

Pour la documentation Bootstrap du carousel elle est entièrement en anglais je vous propose de vous la traduire en français pour mieux comprendre cette documentation .



Carrousel

Un composant de diaporama pour parcourir les éléments (images ou diapositives de texte) comme un carrousel.

Comment ça fonctionne?

Le carrousel est un diaporama pour parcourir une série de contenu, construit avec des transformations CSS 3D et un peu de JavaScript. Il fonctionne avec une série d'images, de texte ou de balisage personnalisé. Il inclut également la prise en charge des commandes et indicateurs précédents / suivants.

Dans les navigateurs où l'API de visibilité de page est prise en charge, le carrousel évitera de glisser lorsque la page Web n'est pas visible pour l'utilisateur (par exemple, lorsque l'onglet du navigateur est inactif, la fenêtre du navigateur est réduite, etc.)

Exemple:

Les carrousels ne normalisent pas automatiquement les dimensions des diapositives.

En tant que tel, vous devrez peut-être utiliser des utilitaires supplémentaires ou des styles personnalisés pour dimensionner le contenu de manière appropriée. Bien que les carrousels prennent en charge les commandes et indicateurs précédents / suivants, ils ne sont pas explicitement requis. Ajoutez et personnalisez comme bon vous semble. La classe `.active` doit être ajoutée à l'une des diapositives, sinon le carrousel ne sera pas visible. Veillez également à définir un identifiant unique sur le `.carousel` pour les commandes facultatives, en particulier si vous utilisez plusieurs carrousels sur une seule page. Les éléments de contrôle et d'indicateur doivent avoir un attribut `data-target` (ou `href` pour les liens) qui correspond à l'id de l'élément `.carousel`. Diapositives uniquement

2 Partie Front-end (index,boutique,réservation).

La barre de navigation en bootstrap:

Pour le design du site j'utilise la documentation Bootstrap est une collection d'outils utiles à la création du design de sites et d'applications web. Les listes déroulantes sont des superpositions contextuelles basculantes pour afficher des listes de liens et plus encore. Ils sont rendus interactifs avec le plugin JavaScript de la liste déroulante Bootstrap inclus. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript. Le .navcomposant de base est construit avec Flexbox et fournit une base solide pour la construction de tous les types de composants de navigation. Il comprend des remplacements de style (pour travailler avec des listes). Un remplissage de lien pour les zones de hit plus grandes et un style de base désactivé. Il offre une barre de **navigation responsive** et design. La barre de navigation annonce un hamburger qui affiche le menu déroulant en verticale pour la version mobile du site ce qui facilite la lecture du menu à l'utilisateur. J'utilise une **CLASS** pour mon **CSS** à la **barre de navigation** pour l'afficher avec les exigences du client. La partie menu déroulant est entièrement codé en Bootstrap. Dans la barre une condition a été codé à la condition de l'utilisateur. Si l'administrateur du site est connecté la partie admin du site avec son panel (réservation de rendez-vous, gestion client, gestion de stock ainsi que sa boutique personnelle).

Documentation Bootstrap pour la barre de navigation.

Aligné à droite avec `.justify-content-end`:

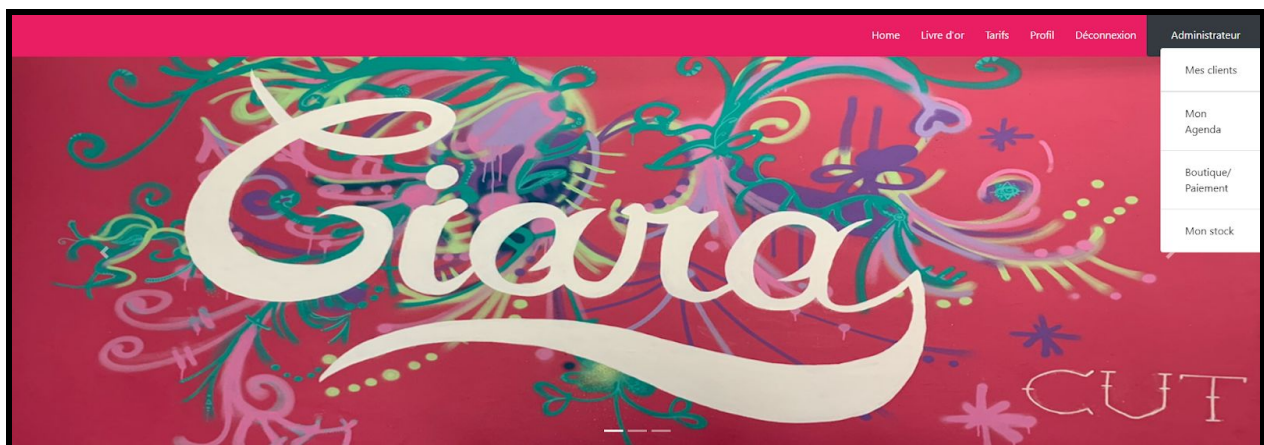
actif Lien Lien désactivé

```
<ul class="nav justify-content-end">
  <li class="nav-item">
    <a class="nav-link active" href="#">Active</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link disabled" href="#" tabindex="-1" aria-disabled="true">Disabled</a>
  </li>
</ul>
```

Copie

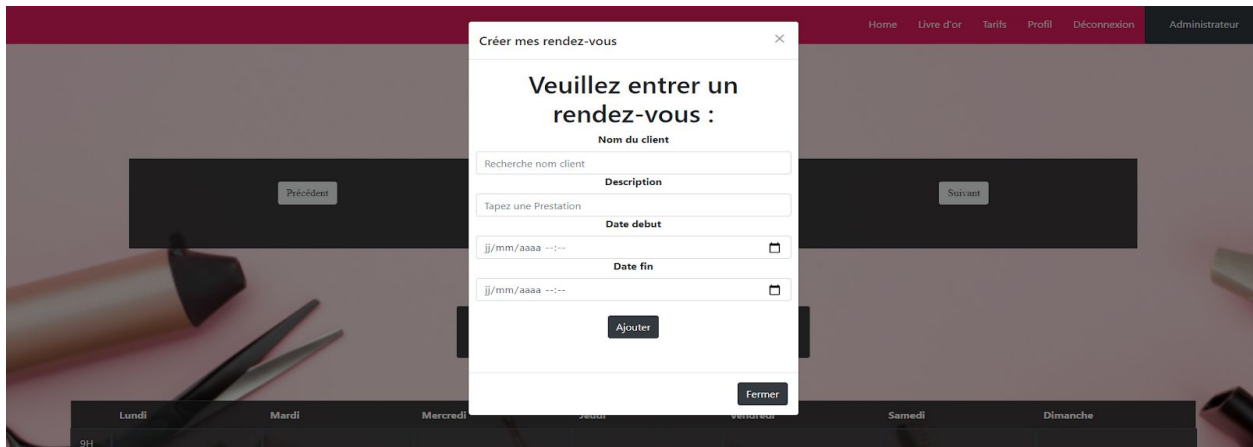
Barre de navigation fait en bootstrap avec les exigences du client.

Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce.



Popup Modal avec la documentation de Bootstrap.

Développer une interface utilisateur web dynamique:



Le client dispose des **popups modal bootstrap** que j'ai récupéré dans la **documentation Bootstrap** pour la partie réservation de son site. Ils s'ouvrent pour la création de la réservation ainsi que la modification et la suppression du rendez-vous. **Les formulaires en bootstrap:**

Adresse e-mail

Nous ne partagerons jamais votre e-mail avec qui que ce soit.

Mot de passe

☐ Regarde moi

```
<form>
  <div class="form-group">
    <label for="exampleInputEmail">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail" aria-describedby="emailHelp">
    <small id="emailHelp" class="form-text text-muted">We'll never share your email with anyone else.</small>
  </div>
  <div class="form-group">
    <label for="exampleInputPassword">Password</label>
    <input type="password" class="form-control" id="exampleInputPassword">
  </div>
  <div class="form-group form-check">
    <input type="checkbox" class="form-check-input" id="exampleCheck1">
    <label class="form-check-label" for="exampleCheck1">Check me out</label>
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

Les contrôles de formulaire de **Bootstrap** développent nos **styles** de formulaire avec des classes. Il nous offre des formulaires design et **responsives**. Pour cela je réutilise la documentation Bootstrap.

Créer mes rendez-vous

Veuillez entrer un
rendez-vous :

Nom du client

Recherche nom client

Description

Tapez une Prestation

Date debut

jj/mm/yyyy --:--

Date fin

jj/mm/yyyy --:--

Ajouter

Fermer

L'exemple suivant montre le code pour différents boutons: Les classes `btn-sm` et `btn-lg` définissent les différentes tailles et couleurs: **Les boutons** utilisés sont le mode **dark** le plus souvent dans le site. Bootstrap comprend plusieurs styles de boutons prédéfinis, chacun servant son propre objectif sémantique, avec quelques extras ajoutés pour plus de contrôle.

Exemple de boutonven bootstrap:

PrimarySecondarySuccessDangerWarningInfoLightDarkLink

```
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>

<button type="button" class="btn btn-link">Link</button>
```

Copy

Exemple de tableau en bootstrap:

#	First	Last	Handle
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

```
<table class="table table-dark">
  <thead>
    <tr>
      <th scope="col">#</th>
      <th scope="col">First</th>
      <th scope="col">Last</th>
      <th scope="col">Handle</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">1</th>
      <td>Mark</td>
      <td>Otto</td>
      <td>@mdo</td>
    </tr>
    <tr>
      <th scope="row">2</th>
      <td>Jacob</td>
      <td>Thornton</td>
      <td>@fat</td>
    </tr>
    <tr>
      <th scope="row">3</th>
      <td>Larry</td>
      <td>the Bird</td>
      <td>@twitter</td>
    </tr>
  </tbody>
</table>
```

Copy

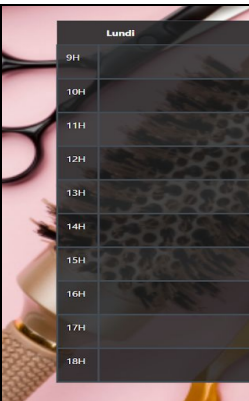


MES TARIFS			
	Court	Mi-long	Long
Brushing	15€	20€	25€
Coupe Brushing	25€	35€	45€
Forfait mèches	55€	65€	75€
Forfait Couleur	45€	55€	65€
Forfait Permanente	65€	75€	85€
Forfait ombré hair Tye and Dye	95€	130€	150€
Lissage Brésilien Soin Botox capillaire	99€	130€	De 150€ à 250€
Coupe Homme	15€	15€	15€
Coupe Enfants -10 ans	10€	10€	10€
Coupe Fillette -10 ans	25€	25€	25€
Forfait Mariage 1er essai	150€	150€	150€
Forfait Mariage 2em essai	180€	180€	180€
Chignon cocktail	50€	60€	70€

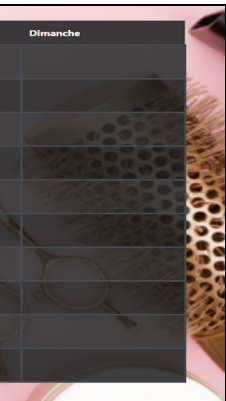


Le Tableau des tarifs s'affichent avec les propriétés **bootstrap** et la class table il prend en compte le nom des prestations ainsi que leur prix. Le Tableau de réservation prend en compte les jours et les heures d'ouverture du salon de coiffure ainsi que les tarifs s'affichent avec les propriétés bootstrap avec la class table .

Tableau des réservations crée en bootstrap.



	Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi	Dimanche
9H							
10H							
11H							
12H							
13H							
14H							
15H							
16H							
17H							
18H							



Documentation slide bootstrap:

Un composant de diaporama pour parcourir les éléments (images ou diapositives de texte) comme un carrousel. Le carrousel est un diaporama pour parcourir une série de contenu, construit avec des transformations CSS 3D et un peu de JavaScript. Il fonctionne avec une série d'images, de texte ou de balisage personnalisé. Il inclut également la prise en charge des commandes et indicateurs précédents / suivants ce qui me permet d'ajouter ou de réduire la quantité d'images dans le slide.



Slide des prestations avant/après de l'index du site:



Le client désire un aperçu de ses prestations avant/après et le slide est le meilleur moyen. En insérant les photos personnelles du client et en reprenant la **class carroussel**, j'ai pu réaliser un avant /après des prestations du client avec un titre de ses différentes prestations. On peut redimensionner les photos et le slide à la taille souhaitée.



Pour la boutique j'ai utilisé la méthode **CARD** pour l'affichage des catégories de ses prestations . Avec cette **documentation** cela me permet d'afficher mes prestation avec une présentation imagé façon carte de produit et un descriptif complet. Le nom,l'image,descriptif,leprix.

3 Partie back-end : (réservation base de donnée).

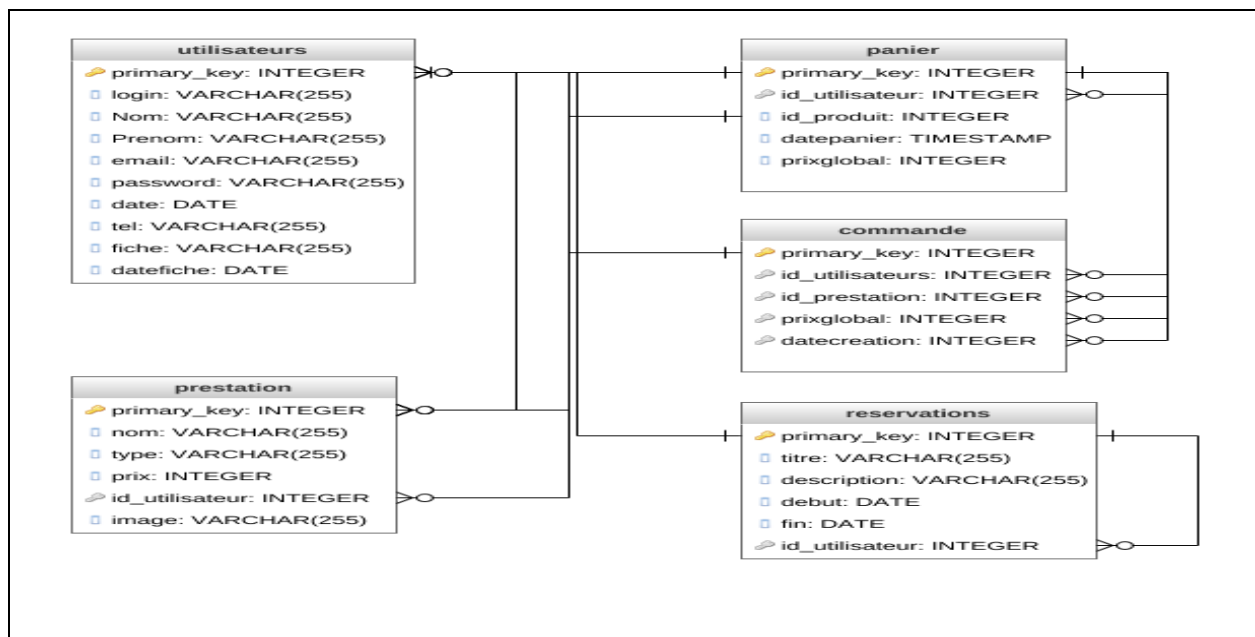
Créer une base de données .

Présentation de la **base de donnée ciaracut**, pour ma partie elle est composée **5 tables**.utilisateurs et réservation pour la partie rendez-vous.
utilisateurs,prestation,commande puis panier pour la partie boutique.



Table	Engine	Character Set	Collation	Size
commande	MyISAM	latin1_swedish_ci		2,6 kio 689 o
commentaires	MyISAM	latin1_swedish_ci		2,5 kio 484 o
panier	MyISAM	latin1_swedish_ci		2,1 kio 84 o
prestation	MyISAM	latin1_swedish_ci		2,8 kio 228 o
produit	MyISAM	latin1_swedish_ci		2,2 kio -
reservations	MyISAM	latin1_swedish_ci		2,7 kio 644 o
sauvegarde	MyISAM	latin1_swedish_ci		3,9 kio -
stock	MyISAM	latin1_swedish_ci		2,1 kio -
utilisateurs	MyISAM	latin1_swedish_ci		3,3 kio 484 o
9 tables	Somme			64 MyISAM latin1_swedish_ci 24,3 kio 2,5 kio

-Schéma de la base de donnée avec l'application Gen my model.



Pour la partie réservation le client nous demande de pouvoir insérer un rendez-vous elle peut ajouter **3 clients par heure** du lundi au **vendredi de 9H à 18H**. Un modal popup modifier ou supprimer s'affiche alors quand le rendez-vous est validé. Le modal récupère directement les informations insérées. Toute la partie réservation est codé en **javascript** avec la fonction **ajax**. Ajax est une architecture informatique qui permet de construire des applications Web et des sites web dynamiques interactifs sur le poste client en se servant de différentes technologies ajoutées aux navigateurs web AJAX. Il excelle dans l'échange des données avec un serveur et permet de mettre à jour les parties d'une page Web - sans recharger la page entière.. La fonction **AJAX** ici récupère l' URL des mes requêtes en php pour pouvoir les executer par la suite .



Elle prend la méthode de mon formulaire ici c'est la méthode POST. **Data** elle recupère l' **ID** de mes post et enfin la fonction succes qui elle valide la fonction ajax si tout le formulaire est rempli.

Un **tableau multidimensionnel** codé en **php** récupère le nombre de jour et l'heure d'ouverture et de fermeture de l'entreprise. J'ai besoin de 2 **boucles for** 1 qui prend en compte les jours, l'autre boucle for prend elle en compte les heures . Une requête **count()** me compte et me donne le nombre de **réservations** .

Tableau PHP (1er partie).

```
<table id="tableauplanning" class="table table-striped table-dark">
  <thead>
    <tr>
      <th></th>
      <th>Lundi</th>
      <th>Mardi</th>
      <th>Mercredi</th>
      <th>Jeudi</th>
      <th>Vendredi</th>
      <th>Samedi</th>
      <th>Dimanche</th>
    </tr>
  </thead>
  <tbody>
<?php
for($ligne =9; $ligne <= 18; $ligne++ )
{
  echo "<tr>";
  echo "<td class='ligne2'>". $ligne. "</td>";

  for($colonne = 1; $colonne <= 7; $colonne++)
  {
    echo "<td class='ligne'>";
    foreach($result as $value)
    {
      $jour=date("w", strtotime($value[3]));
      $h=date("H", strtotime($value[3]));

      if($h==$ligne && $jour== $colonne)
      {
        $idreserv=$value[0];
        ?>
        <div class="lien">
          <?php echo "Client: ".$value[1]. "<br>Prestations: ".$value[2]. ""?>
          <button type="button" id="<?php echo $idreserv;?>" class="material-icons btnRes" data-toggle="modal" data-target="#exampleModal"
            title="Modifier">create</button>
        </div>
```


Tableau PHP (2em partie).

Développer les composants d'accès aux données.

```
<?php
$connexion = new PDO('mysql:host=localhost;dbname=ciaracut', 'root', '');
$requete = $connexion->prepare("SELECT * FROM reservations WHERE id='$idreserv'");
$requete->execute();
$test = $requete->fetchAll();
foreach ($test as $teste) {
    $idtest=$teste;
    ?>
    <div class="modal fade" id="exampleModal" 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">Modifier ou supprimer mes rendez-vous</h5>
    <button type="button" class="close" data-dismiss="modal" aria-label="Close">
    <span class="material-icons" aria-hidden="true">close</span>
    </button>
    </div>
    <div class="modal-body">
    <?php include("../include/reservation-form2.php"); ?>
    </div>
    <div class="modal-footer">
    <button type="button" class="btn btn-dark" data-dismiss="modal">Fermer</button>
    </div>
    </div>
    </div>
    <?php
    }
}
}
echo "</td>";
}
echo "</tr>";
```

Je fais une requête qui **SELECT *** pour récupérer tout les rendez-vous entre la de début et date de fin de ma table reservation dans ma base de donnée pour pouvoir les afficher dans mon tableau. Ensuite toujours dans la boucle foreach je code le popup modal pour m'afficher le formulaire de modification et de suppression de rendez-vous pour chaque rendez-vous effectué.

```
<?php
$db=mysqli_connect("localhost","root","","ciaracut");
mysqli_set_charset($db, "utf8");
$date="SELECT * FROM reservations LEFT JOIN utilisateurs ON utilisateurs.id = reservations.id_utilisateur
WHERE \"\$dateselec\" BETWEEN DATE_FORMAT(debut, \"%Y-%m-%d\") AND DATE_FORMAT(fin, \"%Y-%m-%d\")";
$query=mysqli_query($db, $date);
$result=mysqli_fetch_all($query);

$data2 ="SELECT count(*) FROM reservations LEFT JOIN utilisateurs ON utilisateurs.id = reservations.
id_utilisateur WHERE \"\$dateselec\" BETWEEN DATE_FORMAT(debut, \"%Y-%m-%d\") AND DATE_FORMAT(fin, \"%Y-%m-%d\")";
$query2=mysqli_query($db, $data2);
$result2=mysqli_fetch_all($query2);
```



```
$("#insert").click(function(){
    $.ajax(
        {
            url: '../functions/insert.php',
            type: 'POST',
            data: { validator:"go", nom: $("#nom").val(), description: $("#description2").val(), datedebut: $("#datedebut2").val(), datefin: $("#datefin2").val() },
            success: function(data)
            {
                location.reload(true);
            },
        },
    )
})
```

Mes conditions en PHP sont codées à part puis récupérer par **Ajax** avec le **paramètre URL** qui lui va lire mes conditions en PHP puis la méthode **POST** ou **GET** de mes formulaires avec le paramètre **TYPE**. Les **id** de mes **inputs** sont récupéré par le **paramètre data** de la fonction ajax dans mon formulaire d'envoi. La fonction **reload()** rafraichit la page systématiquement.

```
<article><h1>Veuillez entrer un rendez-vous :</h1></article>
<form class="form-row d-flex d-column justify-content-center align-items-center" method="post">
<label for="formGroupExampleInput"><b>Nom du client</b></label>
<input id="nom" class="form-control" name="search" type="text" placeholder="Recherche nom client" required/>
<label for="formGroupExampleInput"><b>Description</b></label>
<input id="description2" class="form-control" id="formGroupExampleInput" type="text" placeholder="Tapez une Prestation" name="description" required/>
<label for="formGroupExampleInput"><b>Date debut</b></label>
<input id="datedebut2" id="formGroupExampleInput" class="form-control" type="datetime-local" name="datedebut"/>
<label for="formGroupExampleInput"><b>Date fin</b></label>
<input id="datefin2" id="formGroupExampleInput" class="form-control" type="datetime-local" name="datefin"/>
<input id="insert" class="btn btn-dark m-4" type="button" value="Ajouter" name="valider"/>
</form>
```

J'ai effectué les mêmes opérations pour **modifier** et **supprimer** les rendez vous. J'ai en premier temps codé toutes mes requêtes en PHP avec les conditions pour les mettre dans le paramètre ajax en récupérant les méthodes post de tous mes formulaires.

3.1 Partie back-end : (boutique base de donnée).

Pour la base de donnée j'utilise toujours **PHP MY ADMIN**, j'ai effectué des recherches sur le web pour comprendre le fonctionnement d'une boutique et imaginer la base de donnée je me suis inspirer de Cdiscount et Amazon. Pour cela j'ai besoin de **5 tables** Utilisateur -> prestation -> commande -> panier -> sauvegarde.

[illegible]

sauvegarde	
🔑	primary_key: INTEGER
🔑	id_utilisateur: INTEGER
🔑	id_produit: INTEGER
🔑	datepanier: TIMESTAMP
🔑	prixtotal: INTEGER
🔑	mode: VARCHAR(255)
🔑	nom: VARCHAR(255)
🔑	prenom: VARCHAR(255)

J'ai ajouter une table **sauvegarde** qui me permet d'avoir un **historique** de mes **paniers**. Elle se declenche à l'insertion du panier via un déclencheur dans la base de donnée. Cela me permettra par la suite de savoir qui a fait une prestation dans l'entreprise. J'établie par la suite un passage au salon une fois la **prestation payée** en récupérant son nom et son prénom donnera à l'utilisateur de pouvoir poster un commentaire dans le livre or du site.

Partie back-end : (boutique).

Une fois la prestation créée par l'administrateur, je récupère les prestations en faisant un **SELECT *** de ma table prestation pour récupérer toutes les prestations de la table.

```
<?php
    session_start();
    $connexion= mysqli_connect("localhost", "root", "", "ciaracut");
    $requete = "SELECT * FROM prestation ";
    $query = mysqli_query($connexion, $requete);
    $resultat = mysqli_fetch_all($query);
?>
```

Je récupère le résultat avec **FETCH_ALL** et je fais une boucle **FOREACH** pour parcourir toutes mes prestations et pouvoir les affichées dans ma boutique et dans des **CARDS** en **BOOTSTRAP**.

```
<div id="contpresta">
    <?php
    $i=0;
    foreach ($resultat as $values)
    {
        if (!empty($values)) {
            $did=$values[0];
            $img=$values[5];
            $nom=$values[1];
            $type=$values[2];
            echo " <div class=\"cardproduit\" class=\"card\" style=\"width: 15rem\";>";
            echo "<h1 class=\"nomp\">$nom </h1><br>";
            echo "<a href=\"produit.php?p=$did\"><img class=\"card-img-top\" src=\"";
            echo "../upload/$img\"></a>";
            echo "<p class=\"card-text\">Détails:{$values['2']}</p>";
            echo "<p class=\"card-text\">Prix:{$values['3']}€</p>";
            echo "</div>";
        }
    }
?>
</div>
```

Partie back-end : (page prestation).

Ensuite une fois la **prestation validée** elle bascule sur la page **produit** qui elle donne les informations du produit sélectionnée dans la boutique puis je récupère l' **ID** de la prestation ici dans l'exemple suivant c'est le "p" cette page sert de rétractations pour l'administrateur si la prestation selectionnée n'est pas la bonne.

```
<?php
$connexion = new PDO('mysql:host=localhost;dbname=ciaracut', 'root', '');
$retour=$_GET['p'];
$id_utilisateurs=$_SESSION['id'];
$reponse = $connexion->query("SELECT * FROM prestation WHERE prestation.id=$retour ");
$rep=$reponse->fetchAll();
```

C'est à ce moment là que la prestation entre dans la table commande de ma base de donnée l'admin peut revenir en arrière grâce au **bouton supprimer**, une fois celui ci appuyer on retourne systématiquement à la boutique grâce au **header("location:boutiqueprestation.php")**;

```
if (!empty($_GET['p']))
{
    $i = 0;
    foreach ($rep as $val)
    {
        if (!empty($val))
        {
            $did=$val[0];
            $img=$val[5];
            $nom=$val[1];
            $type=$val[2];
            echo " <div id=\"formprestation2\" class=\"card\" style=\"width: 30rem;>
                ";
            echo "<H1 class=\"title\">Ma Selection</H1>";
            echo "<h3 class=\"nomp\">$nom </h3><br>";
            echo "<a href=\"panier.php?p=$did\"><img class=\"card-img-top\" src=\"
                ../upload/$img\"></a>";
            echo "<p class=\"card-text\">{$val['2']}</p>";
            echo "<p class=\"card-text\">Prix:{$val['3']}€</p>";
            include("../include/quantite2.php");
            $i ++;
        }
        else
        {
            echo "Veuillez choisir une prestation!";
        }
    }
}
```


Partie back-end : (panier).

Je me suis inspiré des paniers dans les sites e-commerce tels que Amazon ou Cdiscount, j'ai supprimé le bouton quantité car la prestation sera toujours unique. Je l'ai remplacé par un bouton supprimer la prestation si le produit est en trop dans le panier. Une fois la prestation validée celle-ci entre dans la page panier du site. Je récupère tout de la table **COMMANDE**, je fais un **INNER JOIN** avec la table **PRESTATION** où l'**ID** de la commande est égale à l'**ID** de la prestation. Je fais un **FETCH ALL** pour récupérer toutes les prestations dans mon panier. Le client peut cumuler plusieurs prestations dans son panier suivi d'un montant total de son panier. J'ai créé un bouton retour sur la boutique simple lien **HREF** sur la page boutiqueprestation.

```
$connexion = new PDO('mysql:host=localhost;dbname=ciaracut', 'root', '');
$id_utilisateurs=$SESSION['id'];
$rep= $connexion->query("SELECT * FROM commande INNER JOIN prestation ON commande.id_prestation=prestation.id ");
$test = $rep->fetchAll();
```

Je fais un **FOREACH** pour parcourir ma table panier et l'afficher dans un tableau si le produit est sélectionné avec la méthode **EMPTY** "si le panier est plein".

```
$i=0;
foreach ($test as $values)
{
    if (!empty($values))
    {
        $did=$values[0];
        $img=$values[10];
        $nom=$values[6];
        $prixglobal=$values[3];
        echo " <div id=\"cardpanier\" class=\"card\" style=\"width: 18rem\";>";
        echo " <h1 class=\"nomp\">$nom </h1><br>";
        echo " <img class=\"card-img-top\" src=\"../upload/$img\">";
        echo " <p class=\"card-text\">Détail:{$values['7']}</p>";
        echo " <p class=\"card-text\">Prix:{$values['3']}€</p>";
        include("../include/quantite.php");
        echo "</div>";
    }
    $i++;
}
```

J'affiche un montant total avec la requête suivante et la fonction **SUM()** qui calcule la somme total des prix de ma commande:

```
<?php
$connexion = new PDO('mysql:host=localhost;dbname=ciaracut', 'root', '');
$req=$connexion->query("SELECT SUM(prixglobal) FROM commande WHERE id_utilisateurs=$
    id_utilisateurs");
$total = $req->fetchAll();
?>
```

Si la prestation est validée dans le panier , alors celle-ci entre dans la table **PANIER** puis j'ai mis en place un déclencheur dans la base de donnée pour la table **SAUVEGARDE** avec la requête suivante: Les **déclencheurs** ou **triggers** (en anglais) sont une fonctionnalité avancée très pratique des bases de données. Ils permettent de **déclencher une action lorsque certains paramètres sont réunis** dans une table. Un déclencheur peut se produire lorsque :

- des données sont insérées (INSERT) ;
- des données sont modifiées (UPDATE) ;
- des données sont supprimées (DELETE) ;
- Juste avant (BEFORE) ou juste après (AFTER) l'action sur les données.

```
$connexion=mysqli_connect("localhost","root","","ciaracut");
$id_utilisateurs=$values[1];
$id_produits=$values[2];
$prixtotal=$total[0][0];
$req="INSERT INTO panier (id_utilisateurs,id_produit,datepanier,prixtotal) VALUES ('$
    id_utilisateurs','$id_produits',NOW(),'$prixtotal')";
$query=mysqli_query($connexion,$req);
$id = mysqli_insert_id($connexion);
header("location:paiement.php?id=".$id);
```


Partie back-end : (paiement).

Pour la partie paiement , je dois récupérer le **prix total** du panier pour afficher le prix total du panier à payer j'effectue la **requête** suivante et affiche la somme à payer :

```
$id_utilisateurs=$_SESSION['id'];
$id = $_GET['id'];
$connexion=mysqli_connect("localhost","root","","ciaracut");
$req=("SELECT prixtotal FROM panier ");
$query2=mysqli_query($connexion,$req);
$res=mysqli_fetch_all($query2);
?>

<h1 id="montant">Le montant total à payer est de: <?php echo $res[0][0]?>€</h1>
```

Pour régler la prestation je met en place un **formulaire de paiement** avec comme informations le Nom et le Prénom du client de l'administrateur , le mode de paiement ici c'est chèque ou espèce car l'administrateur ne possède pas de moyen paiement par carte bancaire.

```
<H2 class id="title">En espèce ou chèque</H2><br>

<form id="paiementbloc" method="post">

  <div class="form-column d-flex align-items-center flex-column justify-content-center">
    <h1 class="card-title">Formulaire de paiement</h1>
    <div class="col-10">
      <label for="validationDefault01">Nom</label>
      <input name="nom" type="text" class="form-control" placeholder="Nom client">
      <small class="exemplescript" class="form-text text-muted">(Exemple:Dupont)</small><br>
    </div>
    <div class="col-10">
      <label for="validationDefault01">Prénom</label>
      <input name="prenom" type="text" class="form-control" placeholder="Prénom client">
      <small class="exemplescript" class="form-text text-muted">(Exemple:Madeleine)</small><br>
    </div>
    <div class="form-group col-md-10">
      <label for="inputState"> Paiement</label>
      <select name="mode" id="inputState" class="form-control">
        <option selected>Choisir...</option>
        <option value="espece" required>Espece</option>
        <option value="cheque" required>Chèque</option>
      </select>
    </div>
    <button name="payer" class="btn btn-light" type="submit">Payer</button>
  </div>
</form>
```

Une fois le formulaire validé par le paiement le **PANIER** se vide dans la base de donnée puis grâce à la sauvegarde si l'administrateur entre le nom et le prénom de son client qui a réglé une prestation en boutique, ses informations entrent dans la table **SAUVEGARDE** de la base de donnée via un déclencheur. Les **déclencheurs** ou **triggers** (en anglais) sont une fonctionnalité avancée très pratique des bases de données. Ils permettent de **déclencher une action lorsque certains paramètres sont réunis** dans une table.

Un déclencheur peut se produire lorsque :

- des données sont insérées (INSERT) ;
- des données sont modifiées (UPDATE) ;
- des données sont supprimées (DELETE) ;
- Juste avant (BEFORE) ou juste après (AFTER) l'action sur les données.

Je vérifie les informations puis je récupère les valeurs des inputs entrées dans le formulaire de paiement celui ci pourra ensuite accéder au formulaire du **livre or** du site si son Nom et Prénom figure dans la table sauvegarde .Il pourra alors poster un **commentaire** sur les prestations faites dans l'entreprise.

```
$requete4="SELECT count(id) FROM sauvegarde WHERE nom = '$nom' and prenom = '$prenom';  
$requete4Q=mysqli_query($connexion,$requete4);  
$requete4R=mysqli_fetch_all($requete4Q);  
if($requete4R[0][0] != 0 OR $_SESSION['login'] == 'vanessa')  
{  
    ?>  
    <form id="comment" method="POST" action="livreor.php">  
    <h1 class="title">Le livre or de ciaracut</h1>  
    <label for="formGroupExampleInput"><b>Laisser vos commentaires:</b></label>  
    <div class="form-group">  
    <textarea name="message" class="form-control" id="exampleFormControlTextarea1" rows="3"></textarea>  
    </div>  
    <input class="btn btn-light" type="submit" name="submit" value ="Poster"/>  
    </form>
```

Conclusion du projet.

Le projet **CIARA CUT** est très intéressant car ce site est très complet pour la partie administrateur et a été codé en partie pendant une période inédite celle **du confinement**. Cette épisode de la formation m'a permis d'avoir une meilleure **organisation de travail** et de faire face aux multiples problèmes rencontrés .

Une **réorganisation complète pour l'équipe** a été établie pour respecter les deadlines pendant le confinement avec des heures fixes et régulière afin de pouvoir présenter le projet dans les temps pour le passage au titre.

Organiser des **meets** avec le client une fois par semaine pour qu'elle puissent voir l'avancé du projet et si ont respectaient les consignes données. J'ai appris la méthode du **télétravail**.qui était pour moi encore inconnue à ce jour.

Comprendre les **attentes d'un client** et respecter ses **exigences**, établir un **cahier des charges** c'est une excellente fin de formation cela nous a préparé pour une future insertion en entreprise .

FIN.

Page: 30

