

# Dossier de projet

---

Création d'un site web pour un salon de coiffure

- Présentation de site web
- Résumé

**Présenté par Thomas SAUSSOL**





# Sommaire

<b>1. Introduction</b>	<b>3</b>
1.1 Résumé du dossier de projet	3
<b>2. Présentation</b>	<b>3</b>
2.1 Présentation personnelle	3
2.2 Présentation de l'entreprise	
2.3 Présentation de l'équipe	3
2.4 Présentation de la formation La Plateforme Marseille	3
<b>3. Projet: site web Ciaracut</b>	<b>3</b>
3.1 Présentation du projet	3
3.2 Cahier des charges	3
3.3 Principes	3
3.4 Planification	3
3.5 Organisation	3
>3.5.1 Utilisation de wireframes	3
3.6 Développement de la partie front-end	3
>3.6.1 Réaliser une interface utilisateur web statique et adaptable	3
>3.6.2 Développer une interface utilisateur web dynamique	3
3.7 Développement de la partie back-end	3
>3.7.1 Développer les composants d'accès aux données	3
>3.7.2 Développer la partie back-end d'une application web ou web mobile	3
3.8 Jeu d'essai: inscription et connexion d'un utilisateur	3
3.9 Veille et sécurité	3
>3.9.1 Exemple d'un composant	3
>3.9.2 Comment utiliser ce composant?	3
>3.9.3 Comment fonctionne ce composant?	3
3.10 Difficultés rencontrées	3
3.11 Solutions mises en oeuvre	3
3.12 Conclusion	3



## 1. Introduction

### 1.1 Résumé du dossier de projet

Ce dossier présente mon projet pro que j'ai réalisé durant mon année à la Plateforme située à Marseille La Joliette. J'ai réalisé le projet avec Gregory Fauvel et Anthony Reboli. Ce projet nous a permis d'approfondir nos connaissances sur le PHP. Personnellement j'ai pu améliorer mes compétences en codage et dans la façon de structurer mon code.

## 2. Présentation

### 2.1 Présentation personnelle

Je m'appelle **Thomas SAUSSOL**, j'ai 21 ans et je suis étudiant à La Plateforme à Marseille la Joliette. J'habite à Brue-Auriac dans le Var mais je suis en colocation à Marseille proche de l'école.

J'ai suivi La Formation à la Plateforme en Septembre 2019, après une année en pâtisserie qui s'est mal déroulée. J'ai donc décidé de garder la pâtisserie en tant que passion et de faire un métier dans le domaine du web qui est ma deuxième passion.

Maintenant, je souhaite devenir développeur web et mobile.

### 2.2 Présentation de l'entreprise

L'entreprise pour laquelle j'ai réalisé le Projet pro, s'appelle Ciaracut. C'est une coiffeuse qui est à Marseille 20 Boulevard de la Concorde dans le 9ème.

## 2.3 Présentation de l'équipe

J'ai effectué le projet de Ciaracut avec Gregory Fauvel et Anthony Reboli. Ils sont tous deux étudiants à la Plateforme dans le même cursus que moi.

## 2.4 Présentation de la formation La Plateforme Marseille

La Coding School est une formation web qui s'adresse à tous ceux qui souhaitent s'ouvrir les portes des métiers du numérique, sans préjugés de niveau ou de ressources.

Avec ou sans bac, sans limite d'âge et totalement gratuite, la Coding School offre leur chance à ceux qui veulent faire du numérique leur avenir. Le modèle pédagogique unique de l'école La Plateforme s'adapte aux besoins de chacun.

Membre de la Grande École du Numérique, la Coding School délivre 2 titres RNCP de niveau 5 et niveau 6 (équivalents BAC+2 et BAC+3/4). Les évaluations se font par contrôles continus puis la validation des titres sont faites par des jurys.

L'intégration professionnelle est réalisée au travers de projets professionnels tutorés, portés par des entreprises dès la première année.

La Coding School c'est aussi une pédagogie active et inductive centrée sur l'apprenant et orientée projet. La formation s'effectue en présentiel, 1200h/an, dans un lieu dédié, construit spécifiquement pour catalyser l'apprentissage.

### 3. Projet: site web Ciaracut

#### 3.1 Présentation du projet

Nous avons eu l'annonce de Roxan Roumégas, directeur de La Plateforme, au mois de Mars que nous devons réaliser un projet pour finaliser notre première année (passage au titre RNCP de niveau V). J'ai donc lancé des demandes aux entreprises en leur expliquant mon projet. Je me suis mis avec Gregory Fauvel et Anthony Reboli. La cliente exerce le métier de coiffeuse. Durant la première réunion nous avons établi un cahier des charges que je vous présenterai juste après. Elle nous a fait part de ses souhaits, voulant avoir un site de réservations simple pouvant créer des rendez-vous manuellement en passant par l'application. Elle a exprimé les fonctionnalités de chaque parties. Il y a déjà une partie Administrateur et une partie Client. Puis nous ai venu l'idée de créer l'accès pour le livre d'or du client, pour pouvoir le commenter, que si celui ci avait pris un rendez-vous. L'administrateur peut ensuite avoir une vision globale de ce que ses clients commentent et peut à souhait y répondre ou bien supprimer les messages disgracieux. L'administrateur de son côté, a tout un panel lui servant à ajouter des prestations nouvelles, à gérer ses stocks et gérer ses rendez vous sur l'agenda. L'administrateur peut aussi pour les personnes âgées leurs créer une fiche client car celles ci ne peuvent pas ou ne savent pas se servir d'un site internet.

Nous nous sommes départagés les différentes parties à

effectuer. Pour ma part j'ai réalisé la partie inscription, connexion, livre d'or.

### 3.2 Cahier des charges

Avant de commencer avec la cliente de CIARACUT, nous avons prédéfini les pages de bases pour son site. Nous sommes partis sur l'idée que celui-ci aurait un livre d'or pour que les clients puissent mettre des commentaires après leur passage chez elle. Ils peuvent visualiser le livre d'or sans pouvoir mettre de commentaire.

Puis dans un second temps, nous avons mis les pages de base, comme la page d'inscription du client, la page de connexion, la page d'accueil du site selon le client. Il y a également, la page avec les tarifs d'affichés et une page profil où le client pourra changer ses informations personnelles.

Pour la partie gestion du site de l'administrateur, nous avons mis en place, la page de la clientèle avec une barre de recherche de tous les clients inscrits sur le site. La gestion des stocks, son agenda et la boutique avec un moyen de paiement sont aussi disponibles sur l'interface de l'admin. Quand ce sera l'anniversaire du client, nous avons fait en sorte qu'une remise de 10% pour une durée de 1 semaine soit effectué quand le client prend et paie un rendez-vous.

Pour la charte graphique, la cliente voulait de base 3 couleurs, le rose fuschia (code `#e91e63`), le noir (code `#000000`) et le blanc (code `#ffffff`). Comme on dit, jamais plus de 3 couleurs mais nous avons besoin d'une couleur pour créer un léger contraste. J'ai donc pris l'initiative de proposer la

couleur grise anthracite (code `#303030`), la cliente a de suite été satisfaite. Nous avons adapté le background en respectant les couleurs dominantes du site.

Le site a été conçu selon les exigences de la coiffeuse.

En règle générale, le site a été conçu pour une utilisation simple et rapide car la cliente travail seule dans son entreprise et qu'elle avait un besoin évident pour avoir une gestion globale de son entreprise.

### 3.3 Principes

L'objectif du projet est qu'à partir d'une inscription et d'une connexion, le client puisse créer son compte pour prendre un rendez vous, voir les prestations fournies par la coiffeuse. Une base de données enregistre et gère les inscrits et aussi gère et affiche les pages personnalisées des utilisateurs. Le client pourra demander un rendez vous et ainsi avoir accès au livre d'or. Le client pourra changer aisément ses informations de connexions sur sa page profil.

L'application est simple d'utilisation autant du côté client qu'administrateur.

Pour le design nous avons choisi d'utiliser Bootstrap, c'est-à-dire que l'affichage s'adapte en fonction de la taille d'écran et du terminal utilisé.

### 3.4 Planification

J'ai tout d'abord commencé avec mon groupe par planifier à l'aide d'un *brainstorming*: pour voir quelles tâches essentielles doivent être présentes sur le futur site de la coiffeuse.

Je me suis inspiré de mes travaux passés, effectués à La Plateforme afin de réaliser l'ensemble de mes parties que j'ai modifié en fonction.



Pour sauvegarder mes projets j'ai utilisé Github, nous avons créé un repository avec des branches de chaque personnes. Quand j'avais fini j'ai "COMMIT" et "PUSH".

### 3.4.1 Rythme de travail et conditions

Nous avons commencé le projet avec le groupe milieux Mars juste avant le confinement. Je me suis adaptée en fonction du mode de vie en confinement avec ma famille. C'est-à-dire que j'ai travaillé sur le projet surtout en matinées et quelquefois en demi-journées. Avec l'ensemble du groupe nous avons organisé des Meet par Google pour se voir ainsi que faire des réunions avec la cliente pour la tenir au courant de notre avancement.

Durant le confinement j'ai gardé la méthodologie apprise à La Plateforme depuis Septembre 2019.

### 3.4.1 Méthodologie de travail

J'ai organisé chez moi, un espace de travail dégagé et rangé, pour travailler dans de bonnes conditions. J'ai commencé par ouvrir un Trello. Trello est un outil de gestion de projet en ligne. Il repose sur une organisation des projets en planches listant des cartes, chacune représentant des tâches. J'y ai mis les pages avec une date à laquelle j'ai fini les différentes pages.



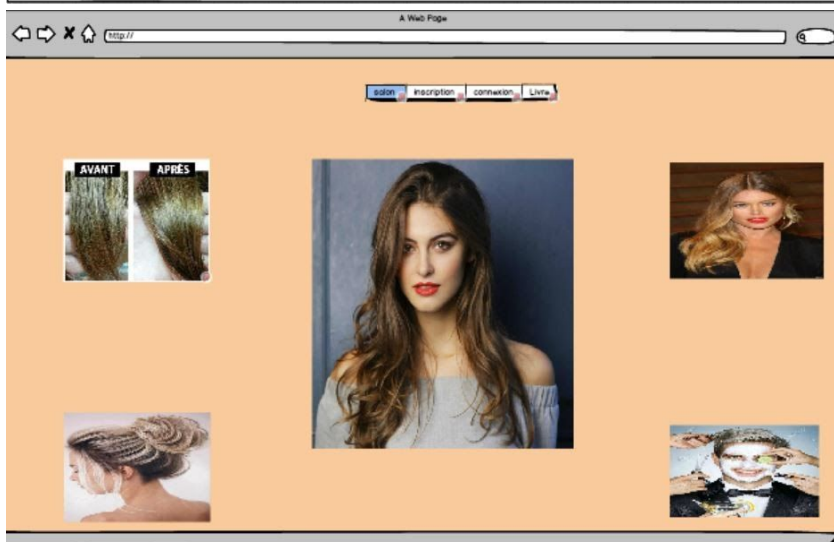
## 3.5 Organisation

### 3.5.1 Utilisation de wireframes

Dès le début j'ai dû maquetter le site pour qu'il y ait une marche à suivre selon ce que voulait la cliente. J'ai utilisé le Wireframe Balsamiq pour modéliser les différentes pages choisies. Il permet au concepteur d'organiser des widgets prédéfinis.



Voici la première maquette de la page de connexion que j'avais imaginée.



Voici également la première maquette que nous avons imaginé avec le groupe pour la cliente. Celui-ci a été modifié avec le temps et les souhaits de celle-ci.

## 3.6 Développement de la partie front-end

### 3.6.1 Réaliser une interface utilisateur web statique et adaptable

Pour gérer l'interface utilisateur, j'ai créé sur ma page inscription et ma page connexion deux formulaires reliés à PhpMyadmin pour que cela enregistre les données du client.

```

<div class="form-column d-flex justify-content-center flex-column align-items-center" method="post">
  <label for="formGroupExampleInput"><b>Login</b></label>
  <input class="form-control col-6" type="text" name="login" required placeholder="Login"/>
  <small class="exemplescript" class="form-text text-muted">(Exemple: Login)</small><br>
  <label for="formGroupExampleInput"><b>Nom</b></label>
  <input class="form-control col-6" type="text" name="nom" required placeholder="Nom"/>
  <small class="exemplescript" class="form-text text-muted">(Exemple: Nom)</small><br>
  <label for="formGroupExampleInput"><b>Prénom</b></label>
  <input class="form-control col-6" type="text" name="prenom" required placeholder="Prénom"/>
  <small class="exemplescript" class="form-text text-muted">(Exemple: Prénom)</small><br>
  <label for="formGroupExampleInput"><b>Email</b></label>
  <input class="form-control col-6" type="email" name="email" required placeholder="Email"/>
  <label for="formGroupExampleInput"><b>Mot de passe</b></label>
  <input class="form-control col-6" type="password" name="pass1" required placeholder="Mot de passe"/>
  <label for="formGroupExampleInput"><b>Confirmer mot de passe</b></label>
  <input class="form-control col-6" type="password" name="pass2" required placeholder="Confirmer votre mot de passe"/>
  <label for="formGroupExampleInput"><b>Date anniversaire</b></label>
  <input class="form-control col-6" type="date" name="date" required placeholder="Date"/>
  <label for="formGroupExampleInput"><b>Votre téléphone</b></label>
  <input class="form-control col-6" type="text" name="tel" required placeholder="Votre tel"><br>
  <input class="btn btn-light" type="submit" name="signin" required value="S'inscrire"/>

```

Sachant que l'on a utilisé Bootstrap, j'ai implémenté d'autres Class en plus des class de Bootstrap, sur différents labels ou inputs pour les modifier à ma guise.

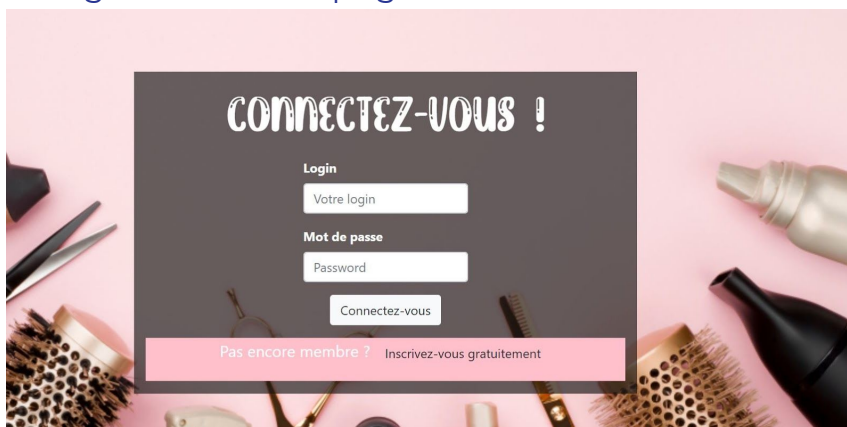
J'ai créé les Class, pour modifier le du css original de Bootstrap en plaçant la balise `<link rel="stylesheet" type="text/css">`

`href="../css/ciaracut.css">` juste après les scripts et les links de la bibliothèque.

```
/*PAGE INSCRIPTION*/
#inscription{
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
}
#continscription{
  background-image: url("../upload/panierfond.jpg");
  background-repeat: no-repeat;
  background-size: cover;
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  color: white;
  min-height: 690px;
}
#main{
  margin: 3%;
  display: flex;
  flex-direction: column;
  justify-content: center;
  background-color: #191717a8;
  width: 50%;
}
```

```
#contconnexion{
  background-image: url("../upload/panierfond.jpg");
  background-repeat: no-repeat;
  background-size: cover;
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  color: white;
  min-height: 641px;
}
.inscri{
  background-color: pink;
  text-align: center;
  color: white;
  font-size: 20px;
  width: 100%;
}
#notcon{
  color: black;
  font-size: 30px;
  font-family: police;
}
```

J'ai donc pu agencer les différents formulaires d'inscription et de connexion en utilisant le "display: flex", le "flex direction", le "justify content" et le "align items". Tout ceci m'a permis de modéliser aisément toute la partie css des formulaires d'inscription et de connexion du site de Ciaracut. la couleur de base des formulaires était neutre donc blancs. La cliente demandait 4 couleurs dont le principal est le rose fuschia. J'ai donc adapté son souhait au codage de ces deux pages.



Pour l'adaptation sur

les différents écrans, Bootstrap prend directement en charge tout cela. Effectivement le css de Bootstrap est déjà codé de façon à être responsive.

### 3.6.2 Développer une interface utilisateur web dynamique

Pour réaliser l'interface côté client, j'ai tout d'abord commencé par lire la documentation de Bootstrap. Ceci est une librairie créée pour les développeurs. Elle est constituée d'une collection d'outils utiles à la création du design (graphisme, animation et interactions avec la page dans le navigateur, etc.) de sites et d'applications web. 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 en option. C'est l'un des projets les plus populaires sur la plate-forme de gestion de développement GitHub.



Je me suis baladé sur la documentation, dans le but de rechercher une barre de navigation adaptée pour le site du salon de coiffure selon l'utilisateur qui est présent dessus.

Pour commencer je suis allé dans la section "Components" puis dans la section "Navbar".

J'ai sélectionné la navbar qui me correspondait et je l'ai utilisé dans mon IDE PhpStorm sur la page navbar.php. Ensuite j'ai effectué cette même tâche pour les différentes condition d'affichage de cette navbar.

Exemple :

- Quand l'utilisateur n'est pas connecté au site, cela va afficher la redirection vers le Home, les tarifs du salon, la visualisation du livre d'or sans pouvoir rentrer de commentaires, une case connexion et une case inscription.
- Quand l'utilisateur est connecté au site, cela va afficher la redirection vers le Home, les tarifs du salon, son profil, le livre d'or qu'il pourra remplir pour complimenter la cliente sur ses réalisations effectuées et une partie

déconnexion que quand celui ci cliquera dessus sa sessions sera sauvegardée et fermée.

Une fois la structure de la barnav mise et adaptée selon les conditions d'utilisation de l'utilisateur, la condition qui reste à faire est de l'adapter selon la taille de l'écran. De base c'est Bootstrap qui gère l'adaptation de la navbar sur le site en fonction de la width de l'écran utilisé. Nous nous sommes mis d'accord, pour un côté esthétique, de réaliser un menu hamburger quand la taille de l'écran est inférieur à 450 et 650 pixels..

```
@media only screen and (max-width:450px) {
```

```
#menu-deroulant {
```

```
display: flex;
```

```
flex-direction: column;
```

```
flex-wrap: wrap;
```

```
width: auto;
```

```
}
```

```
}
```

```
@media only screen and (max-width:650px) {
```

```
#menu-deroulant {
```

```
display: flex;
```

```
flex-direction: column;
```

```
flex-wrap: wrap;
```

```
width: auto;
```

```
}
```

Le Javascript est aussi intégré à Bootstrap pour effectuer des évènements.

Mais qu'est ce que le JavaScript?

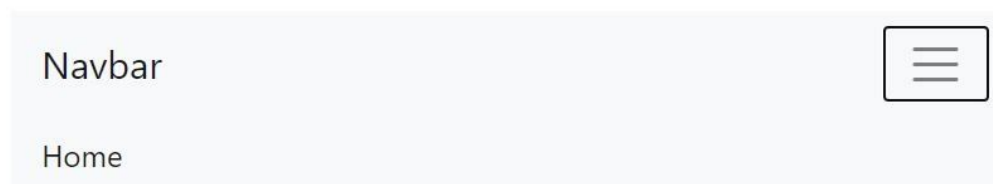
- C'est un langage de script qui permet de rendre les pages web interactives.
- C'est un langage qui fonctionne côté client et ne demande aucune ressource au niveau du serveur.
- C'est un langage interprété directement par le navigateur

L'adaptation à l'écran que j'ai effectuée, en plus des class de Bootstrap par rapport, m'a permis effectivement d'annoncer un hamburger quand l'écran se réduit.



Quand on clique  
liste déroulante.

dessus, le menu apparaît avec une



Bien sûr c'est une démonstration, vous la verriez sur le site en totale fonctionnement.

L'animation de la liste qui se déroule est gérée par le code déjà intégré de







forme de fichier .sql et ainsi transférer facilement ses données.

De plus celui-ci accepte la formulation de requêtes **SQL** directement en **langage SQL**. Cela permet de tester ses requêtes par exemple lors de la création d'un site et ainsi de gagner un temps précieux.

The screenshot shows a database management tool interface. On the left, a tree view shows the database structure under 'ciaracut', including tables like 'commande', 'commentaires', 'panier', 'prestation', 'produit', 'reservations', 'sauvegarde', 'stock', and 'utilisateurs'. The main panel displays a table with columns: Table, Action, Lignes, Type, Interclassement, Taille, and Perte. It lists 9 tables, all of type MyISAM with the latin1\_swedish\_ci character set.

Table	Action	Lignes	Type	Interclassement	Taille	Perte
commande	Parcourir Structure Rechercher Insérer Vider Supprimer	0	MyISAM	latin1_swedish_ci	1 kio	-
commentaires	Parcourir Structure Rechercher Insérer Vider Supprimer	1	MyISAM	latin1_swedish_ci	2 kio	-
panier	Parcourir Structure Rechercher Insérer Vider Supprimer	0	MyISAM	latin1_swedish_ci	1 kio	-
prestation	Parcourir Structure Rechercher Insérer Vider Supprimer	8	MyISAM	latin1_swedish_ci	2,4 kio	-
produit	Parcourir Structure Rechercher Insérer Vider Supprimer	2	MyISAM	latin1_swedish_ci	2 kio	-
reservations	Parcourir Structure Rechercher Insérer Vider Supprimer	4	MyISAM	latin1_swedish_ci	2,2 kio	-
sauvegarde	Parcourir Structure Rechercher Insérer Vider Supprimer	13	MyISAM	latin1_swedish_ci	2,5 kio	-
stock	Parcourir Structure Rechercher Insérer Vider Supprimer	2	MyISAM	latin1_swedish_ci	2 kio	-
utilisateurs	Parcourir Structure Rechercher Insérer Vider Supprimer	3	MyISAM	latin1_swedish_ci	2,3 kio	-
<b>9 tables</b>	<b>Somme</b>	<b>33</b>	<b>MyISAM</b>	<b>latin1_swedish_ci</b>	<b>17,5 kio</b>	<b>0</b>

Sur ma page d'IDE, pour accéder au site, j'ai codé un formulaire d'inscription en reliant bien la table inscription à la base de donnée de Ciaracut

```
Mysqli_connect("localhost","root","","ciaracut");
```

Puis dans un second, temps j'ai mis avec le "isset" que quand la page apparaît, que le client ne soit pas déjà connecté alors le site va afficher le formulaire avec les inputs à remplir.

Une fois celui ci remplit, vu que j'ai codé une partie de mon code en POO, la class `public function register($login, $nom, $prenom, $email, $pass1, $pass2, $date, $tel)` va enregistrer et vérifier les information prescrites et les insérer dans la Base de données avec la requête INSERT INTO.

```
query("INSERT INTO utilisateurs VALUES(NULL, '$login', '$nom', '$prenom', '$email', '$hash', '$date', '$tel', NULL, NULL)");
```

Une fois cela effectué, toutes les informations de l'utilisateur sont sauvegardées dans la base de données.

En finalité, pour que le client puisse se connecter sur son espace personnel, il doit dans un premier temps accéder au menu du site de coiffure, puis cliquer sur "Connexion".

Il doit rentrer son Login/Identifiant ainsi que son Mot de passe.

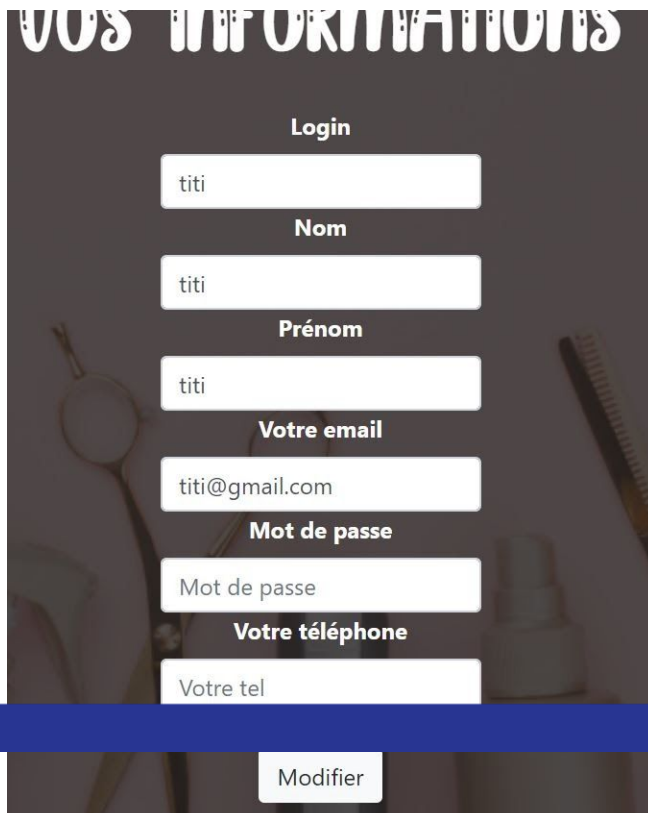
Tout cela en back est géré par la fonction en POO qui s'intitule `public function connect.` Celle-ci va vérifier le login entré et le mot de passe. Si tout est bon alors la condition est respectée, donc le programme va aller vérifier et chercher toutes les informations de l'utilisateur pour lui afficher sa page personnelle.

```
$this->id = $donnees['id'];  
$this->login = $login;  
$this->nom = $donnees['nom'];  
$this->prenom = $donnees['prenom'];  
$this->email = $donnees['email'];  
$this->pass1 = $donnees['pass1'];  
$this->date = $donnees['date'];  
$_SESSION['login'] = $login;  
$_SESSION['pass1'] = $pass1;  
$msg = "ok";
```

Si aucune de ces caractéristiques n'est respectée, alors un message appelé par une condition va s'afficher demandant à l'utilisateur de retaper son mot de passe.

```
else
{
    $msg = "Login ou mot de passe incorrect";
}
```

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



The screenshot shows a web form titled "VOS INFORMATIONS" with a dark background. The form contains several input fields with labels above them: "Login" (containing "titi"), "Nom" (containing "titi"), "Prénom" (containing "titi"), "Votre email" (containing "titi@gmail.com"), "Mot de passe" (containing "Mot de passe"), and "Votre téléphone" (containing "Votre tel"). At the bottom of the form is a button labeled "Modifier".

Dans cette partie, je vais vous présenter la situation quand un utilisateur veut changer ses informations personnelles. Quand celui ci est connecté, il peut à tout moment modifier ses informations. Pour cela, la page profil sert à ça. Côté utilisateur le formulaire avec

les informations rentrées seront déjà insérées.

Côté back, j'ai codé en POO l'update des informations de l'utilisateur.

```
<?php

if (isset($_POST['update']))
{
    $profil_update = $user->update($_POST['login'],
    $_POST['lastname'], $_POST['firstname'], $_POST['email'],
    $_POST['pass'], $_POST['tel']);

    if ($profil_update == "erreur2")
    {
        ?><p>Mot de passe trop court (5 caractères minimums)</p>
    }
}

echo "Le changement a bien été effectué!";

?>
```

Pour la fonction update, dans le fichier function.php, je vais sélectionner toutes les informations que l'utilisateur pourra changer de sa session avec la fonction `connectdb()->query("SELECT *FROM utilisateurs WHERE`

`login='$log'");`. Cela va chercher le login de l'utilisateur et afficher toute les données demandées en condition.

Le client change les informations qu'il veut et la commande UPDATE permet d'effectuer des modifications sur des lignes existantes. Très souvent cette commande est utilisée avec WHERE pour spécifier sur quelles lignes doivent porter la ou les modifications.

Exemple:

```
$update = $this->connectdb()->query("UPDATE utilisateurs SET
login='$login', nom='$nom', prenom='$prenom', email='$email',
password='$hash' , tel='$tel' ,fiche= NULL, datefiche= NULL
WHERE login='$log'");

$this->login = $log;

$this->nom = $nom;

$this->prenom = $prenom;

$this->email = $email;

$this->password = $pass1;

$this->tel = $tel;

}
```

Si jamais une information est changée par l'utilisateur, alors le programme va la changer.

```
if (isset($_POST['update']))
{
```

```

$profil_update = $user->update($ POST['login'],
$ POST['lastname'], $ POST['firstname'], $ POST['email'],
$ POST['pass'],$ POST['tel']);

if ($profil_update == "erreur2")

{

?><p>Mot de passe trop court (5 caractères minimums)</p>

<?php

}

echo "Le changement a bien été effectué!";

}

```

Si le changement effectué est bon alors l'update est effectuée dans la base de donnée, et le message suivant le echo "Le changement a bien été effectué!" va s'afficher.

L'utilisateur devra alors se connecter avec son nouvel identifiant et son nouveau de passe par exemple si celui ci les changera.

Dans sécurité: L'utilisateur a accès au livre d'or que si sa commande passée au salon est validée par sa soeur, il pourra donc y inscrire des commentaires.

### 3.8 Jeu d'essai: inscription et connexion d'un utilisateur

Je vais vous présenter un jeu d'essai, réalisé sur une fonctionnalité qui pour moi est essentielle à l'accès d'un utilisateur au site.

Ordre	Scénario	Cas de test: valeurs	Cas de test: résultats attendus
1	Scénario nominal	<b>Login:</b> toto <b>Nom:</b> Saussol <b>Prénom:</b> Thomas <b>Email:</b> toto@gmail.com <b>Mot de passe:</b> 123456 <b>Confirmer le mot de passe:</b> 123456 <b>Date d'anniversaire:</b> 09/01/99 <b>Votre téléphone:</b> 06 56 32 34 56	<pre>header ("Refresh: 1;URL=connexion.php");</pre> <p>Redirection vers la page connexion.php pour se connecter</p> <p>L'inscription a été effectuée.</p>
2	Scénario d'exception: Mots de passes rentrés différents	<b>Login:</b> toto <b>Nom:</b> Saussol <b>Prénom:</b> Thomas <b>Email:</b> toto@gmail.com <b>Mot de passe:</b> 123456 <b>Confirmer le mot de passe:</b> 126666 <b>Date d'anniversaire:</b> 09/01/99 <b>Votre téléphone:</b> 06 56 32 34 56	<p>Message d'erreur apparu "Mots de passes rentrés différents"</p> <p>Action requise: se réinscrire du début</p>
3	Scénario d'exception: Mot de passe trop court	<b>Login:</b> toto <b>Nom:</b> Saussol <b>Prénom:</b> Thomas <b>Email:</b> toto@gmail.com <b>Mot de passe:</b> 123456 <b>Confirmer le mot de passe:</b> 1234	<p>Message d'erreur apparu "Mot de passe trop court"</p> <p>Action requise: se réinscrire du début</p>

		<b>Date d'anniversaire:</b> 09/01/99 <b>Votre téléphone:</b> 06 56 32 34 56	
--	--	--	--

Commentaires :

- Les scénarios et les jeux d'essai sont conçus avant la réalisation de l'application (codage et tests)
- 1 cas d'utilisation est décrit par plusieurs scénarios : 1 scénario nominal, des scénarios alternatifs, des scénarios d'exception
- 1 jeu d'essai est destiné à valider l'ensemble des scénarios d'un cas d'utilisation
- 1 cas de test est un ensemble de valeurs destiné à tester un scénario
- 1 scénario peut être validé par plusieurs cas de test
- L'ordre des cas de test est important, car un cas de test peut affecter l'état initial de la base de données
- Le jeu d'essai comporte une ligne de tableau par cas de test

### 3.9 Veille et sécurité

#### 3.9.1 Exemple d'un composant

Voici, en trois parties, un exemple concret d'un composant que j'ai codé :

- D'un point de vue utilisateur
- Comment utiliser ce composant ?
- Comment fonctionne ce composant ?



Le composant est une notification dont le but est d'avertir l'utilisateur d'une information (généralement pour remplir tous les champs des formulaires), sans perturber sa navigation au sein de l'application.

Je me suis inspiré des différents site où il fallait s'inscrire. Sur tous y était mis l'information quand un input était vide ou mal rempli. C'est un système de sécurité du coté de la BDD pour qu'aucun champ ne soit vide. Je l'ai appris durant ma formation.

D'un point de vue utilisateur

Design de l'alerte:



En gris anthracite, il y a le background du formulaire pour se connecter. avec la notification arrivant sur chaque champ non rempli. Celle-ci est composée d'un cadre en forme de bulle rectangulaire, d'un message au centre, et à gauche d'un caractère spécial. Celui-ci est couplé à une couleur de fond permettant à l'utilisateur d'identifier en un clin d'œil le type de notification.

Dans l'exemple ci-dessus, la notification correspond à un champ non renseigné.

### 3.9.2 Comment utiliser ce composant?

Pour la page de connexion ci dessus, la notification est une 'info':

- Pour renseigner correctement tous les champs.
- Pour ne pas oublier de mettre un arobase dans l'email (sinon il n'est pas pris en compte).

Les notifications sont affichées en-dessous de chaque champ.

Une notification apparaît quand on clique sur "Connectez-vous" . Elle reste ensuite sur l'écran :

- 5 secondes si elle est de type 'info'. Puis elle disparaît.

### 3.9.3 Comment fonctionne ce composant?

Lorsque j'ai réalisé ces notifications, je voulais qu'elles soient simples à paramétrer et mettre en place, accessibles à quelqu'un qui ne connaîtrait pas le code. Pour cela, j'ai représenté chaque champ du formulaire relié à:

- un login
- un mot de passe

Si la variable "\$\_Post" contient les informations, alors je les traite. Je place sur le bon formulaire grâce au "name" de la balise "input". Et je fais les vérifications ensuite.

Le fonctionnement : sécurité page de connexion

J'ai créé mon formulaire avec le framework Bootstrap. Ce formulaire fonctionne avec la méthode POST. Il présente les différents champs de saisie et le bouton pour soumettre le formulaire. Le 'name' du bouton se nomme 'submit', il enregistre toutes les valeurs des champs. Je reçois les données en PHP grâce à

une session 'start' qui sert à savoir si une connexion est démarrée, pour restituer ensuite les valeurs de cette session. Je récupère la valeur du \$POST, si elle est remplie. Puis je vais l'extraire et la placer vers le bon formulaire.

Pour sécuriser le site, j'ai utilisé sur chaque formulaire, le 'required' à la fin de chacun de mes 'input'. Cela permet de mettre en forme les éléments obligatoires pour remplir correctement un formulaire.

J'ai aussi utilisé, le mot de passe et la confirmation du mot de passe dans le formulaire d'inscription. Si le mot de passe est différent, alors un message d'infos précise que le mot de passe ne correspond pas.

Ensuite, quand un utilisateur s'inscrit par exemple, son compte est créé dans la base de données. Puis, la fonction INSERT permet d'insérer toutes les données des champs.

Chaque informations rentrées par l'utilisateur sont directement stockées dans la base de données. '\$prenom', '\$nom', '\$mail', '\$mdp', sont stockés dans un tableau.

Le but de cette méthode est d'arriver à la fin d'une inscription en PHP. Ceci permet de choisir une quantité d'informations à récupérer, limiter et sécuriser les données, vérifier si elles sont utiles.

Exemple :

- Enlever les espaces avant et après.
- Vérifier que le mail est bien une adresse email.
- Confirmer le mot de passe et le crypter pour garantir la sécurité de ma base de donnée et de mon site.

### 3.10 Difficultés rencontrées

Durant le projet pro, j'ai rencontré un certains nombre de problème.

J'avais codé l'inscription et la connexion en POO dans le fichier functions.php que j'intègre en "INCULDE". Mais quand la cliente lors de la première réunion nous a fait part de ses souhaits pour l'inscription, elle avait oublié de nous parle du numéro de téléphone à intégrer dans la base de données.

Ensuite il y a eu le problème dans la base de donné de mettre la date en "DATE" ou en Varchar mais en séparant pour le Varchar le jour et le mois d'anniversaire.

Mais mes coéquipier n'ont pas été habitués au POO et pour moi c'était encore nouveau, et ils ont eu du mal avec leurs code pour récupérer les informations de l'utilisateurs.

De base nous étions partis sur un site de réservation de client, mais pour des raisons de sécurités du livre d'or nous avons préférés après après maintes réflexions nous diriger sur un système de panel admin. Ça veut dire que le client ne peut accéder au livre d'or que si et seulement si celui ci a prit une des prestations proposées.

L'administrateur peut si besoin inscrire son client si celui ci est trop âgé et ne sait pas utiliser internet.

Mes coéquipiers ont travaillé sur la récupération de la page de la cliente sur Facebook et ont eu du mal, ils pensaient passer par une Api mais sont au final allé chercher l'iframe de Facebook dans l'API du site.

### 3.11 Solutions mises en oeuvre

Pour réaliser la navbar du site j'ai utilisé Bootstrap la librairie accessible à tous.

**Introduction**

Get started with Bootstrap, the world's most popular framework for building responsive, mobile-first sites, with BootstrapCDN and a template starter page.

**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="sha384-4X9KwZC4o9eWE/T6cu2pNtQ1dJX1jW68uKtg1MbYbd9VTW5q936l8/CPw0X" />
```

**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="sha384-4X9KwZC4o9eWE/T6cu2pNtQ1dJX1jW68uKtg1MbYbd9VTW5q936l8/CPw0X" />
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-99T41A3dfi3P0kE71Sf6PO2fU9M9E9RA3aE1AI14KOh3pDp01z0vaz29kF1gE" />
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js" integrity="sha384-OEGEGEPSPZ4x2XRDZc7Q7TGuhRUQ91OCEpGGWgJ0p26mrGUVo5eZ6d7qq91CGAR8" />
```

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.

Voici donc la documentation de bootstrap afin de bien implémenter les fonctionnalités à la navbar. La documentation est utile à tout début de projets pour une langue de développeur encore inconnue ou simplement

pour aller chercher les liens diverses au bon fonctionnement des animations par exemple.

Pour rappel, le link Css Bootstrap se met en haut dans la partie du `<Head>`. Le script Javascript se met plutôt en pied de page juste avant la fermeture du html.

La documentation nous dit que beaucoup de leurs composants nécessitent l'utilisation de Javascript pour fonctionner. Plus précisément, ils nécessitent jQuery, Popper.js, et leurs propres plugins Javascript. Ils disent de placer les scripts suivants près de la fin de nos pages, juste avant la balise `</body>` de fermeture, pour les activer. jQuery doit venir en premier, puis Popper.js, et ensuite leurs plugins Javascript.

Ils finissent par dire que la version slim de jQuery, mais la version complète est également supportée.

Au final, la librairie nous montre comment la structure du site doit être.

```
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstr

    <title>Hello, world!</title>
  </head>
  <body>
    <h1>Hello, world!</h1>

    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrit
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js" integr
  </body>
</html>
```

Copy

On remarque que tout est bien implémenté, je peux ensuite modifier à ma guise sans toucher les links et les scripts de bootstrap.

## CONCLUSION

La réalisation de ce projet m'a appris beaucoup de choses quand à la gestion du côté admin, ainsi que la concrétisation d'un projet de A à Z.

J'étais enthousiaste à l'idée d'organiser et gérer le développement du site de Ciaracut . Cependant, je ne me rendais pas compte de ce que cela impliquerait en terme de temps. J'ai pour objectif d'approfondir le côté



design pour pouvoir être polyvalent en entreprise : allier les profils développeur web et web designer.

Actuellement, pour le Front-end et le Web Design j'aime découvrir de nouvelles choses mais j'aime tout autant découvrir et apprendre de nouvelles technologies en back-end.