



# *Best Hair By Fa*

## **Titre Professionnel Développeur Web et Web mobile** **Niveau III RNCP 31114**



## Remerciements



Il m'est particulièrement agréable avant de présenter mon travail, d'exprimer toute ma gratitude envers les personnes qui de près ou de loin m'ont apporté leur sollicitude. Mention spéciale à Kimly TEA.

J'adresse ma reconnaissance à toute l'équipe de La MANU : notre campus manager madame Sarah LOUVET qui a toujours été insufflée de la motivation et de l'énergie aux bons moments, nos formateurs messieurs Jean-Philippe DENET et Amadou CAMARA pour les explications et les conseils qui ont un grand impact dans la réussite du projet réalisé.

Un grand merci à tous les apprenants de la promo1 de La MANU Amiens, particulièrement monsieur Nicolas VALLOIS pour son aide et sa disponibilité.

## Sommaire

|  |    |
|--|----|
| REMERCIEMENTS .....                              | 2  |
| COMPÉTENCES COUVERTES PAR LE PROJET.....         | 4  |
| RÉSUMÉ.....                                      | 5  |
| INTRODUCTION GÉNÉRALE.....                       | 6  |
| CAHIER DES CHARGES PROJET.....                   | 7  |
| MAQUETTE.....                                    | 15 |
| CONCEPTION DE LA BASE DE DONNÉES .....           | 19 |
| MISE EN PLACE DE LA BASE DE DONNÉES.....         | 21 |
| RÉALISATION COMPORTANT DES EXTRAITS DE CODE..... | 27 |
| PRÉSENTATION DU JEU D’ESSAI.....                 | 30 |
| SITUATION AYANT NÉCESSITE UNE RECHERCHE .....    | 33 |
| EXTRAITS DU SITE ANGLOPHONE + TRADUCTION .....   | 34 |
| CONCLUSION.....                                  | 35 |

## 1- Liste des compétences couvertes par le projet

Le tableau suivant récapitule les compétences du référentiel couvertes par le projet.

| N°<br>Fiche<br>AT | Activités types   | N°<br>Fiche<br>CP | Compétences professionnelles  |
|-------------------|---|-------------------|---|
| 1                 | Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité | 1                 | Maquetter une application   |
|                   |   | 2                 | Réaliser une interface utilisateur web statique et adaptable  |
|                   |   | 3                 | Développer une interface utilisateur web dynamique  |
|                   |   | 4                 | Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce            |
| 2                 | Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité  | 5                 | Créer une base de données   |
|                   |   | 6                 | Développer les composants d'accès aux données   |
|                   |   | 7                 | Développer la partie back-end d'une application web ou web mobile                                   |
|                   |   | 8                 | Elaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce |

## 2- Résumé

Avec une expérience dans le milieu commercial et après plusieurs discussions avec une coiffeuse à domicile que j'accompagne dans ses commandes d'extensions de cheveux naturels, j'ai décidé de les commercialiser via un site pour faciliter mes échanges avec celle-ci et développer ma propre clientèle.

Le but de mon projet est de réaliser un site de vente en ligne d'extensions de cheveux 100% naturels de qualité supérieure. Il proposera un large choix qui sera accessible aux particuliers et aux professionnels.

Après une étude concurrentielle d'autres sites équivalents ou plus importants, j'ai fait évoluer ma maquette. Je m'en suis inspirée aussi pour mettre en place la navigation.

Sur mon site, les utilisateurs pourront facilement retrouver les différents types de produits, par catégorie ou fonction de recherche. Ils pourront également me contacter en cas de besoin ou pour un conseil sur le choix des produits.

Pour pouvoir passer une commande, les clients devront s'inscrire via un formulaire. Ils auront ensuite la possibilité d'accéder à leur espace client, le mettre à jour ou le supprimer s'ils le souhaitent. Suite à leurs commandes, les clients seront invités à laisser un commentaire sous la fiche du/des produit(s) commandé(s).

Le projet sera développé en étant le plus précis possible pour que le lecteur soit éclairé sur les différentes étapes de ce travail qui est vraiment passionnant.

## Introduction générale

La réalisation d'un site web doit être impérativement précédée d'une méthodologie d'analyse et de conception qui a pour objectif de permettre de formaliser les étapes préliminaires du développement d'un site afin de rendre ce développement plus fidèle aux besoins du client.

La phase d'analyse permet de lister les résultats attendus, en termes de fonctionnalités et la phase de conception permet de décrire de manière non ambiguë, le plus souvent en utilisant un langage de modélisation, le fonctionnement futur du système, afin d'en faciliter la réalisation.

### 3- Cahier des charges

#### 1. Présentation du projet

“Best Hair by Fa” est un site de vente en ligne de cheveux 100 % naturels de qualité supérieure, actuellement en cours de création. L’enregistrement des statuts pour la création d’une entreprise est prévu.

Ce document décrit le contexte, les besoins fonctionnels et les objectifs du projet. Un premier découpage des étapes nécessaires à la réalisation d’un tel projet donne lieu dans ce document à un planning prévisionnel. Ce document a pour finalité de définir le projet de manière simple et détaillée et de définir les objectifs auxquels devra répondre le site.

#### 1-2 Les objectifs du site :

Le site de “Best Hair by Fa” sera la seule plateforme de vente de l’entreprise, et doit être en mesure de proposer une expérience de qualité aux visiteurs du site. Le back-office du site doit également permettre une gestion quotidienne des activités e-commerce (suivi des commandes, mise à jour du catalogue produit).

#### 1-3 La cible adressée par le site :

“Best Hair by Fa” cible les particuliers, les coiffeuses indépendantes et à domicile. Le discours marketing, les modes de tarification pratiqués ou l’image de marque qui sera développée ciblera principalement une clientèle relativement jeune (25-35 ans) utilisant les supports digitaux de manière intensive.

#### 1-4 Les Objectifs quantitatifs après 1 an :

- Nombre de pièces vendues : 3000
- Visiteurs par jour : 500, dont plus de 30 % en référencement naturel
- Taux de conversion : supérieur à 1,2 %
- Nombre de compte clients : 1000

### 1-5 Périmètre du projet :

- L'activité vise pour uniquement la France métropolitaine.
- Le site sera disponible en français uniquement.
- Le site sera intégralement ''Responsive Design''.
- L'ensemble des fonctionnalités détaillées dans ce document seront accessibles depuis un mobile.

### 2. Description graphique et ergonomique :

#### 2-1 Charte graphique :

- La charte graphique doit être moderne et épurée.
- La couleur dominante du site sera le blanc : #F8F9FA
- La couleur secondaire, utilisée pour les titres, les boutons, et autres éléments de navigation sera le gris
- La troisième couleur, utilisée entre autres pour les à plats, sera le bleu clair

#### 2-2 Inspirations :

Je me suis inspirée de ces deux sites :

- <https://www.miami-hair-shop.com/>
- <https://www.koryn-hair.fr/>

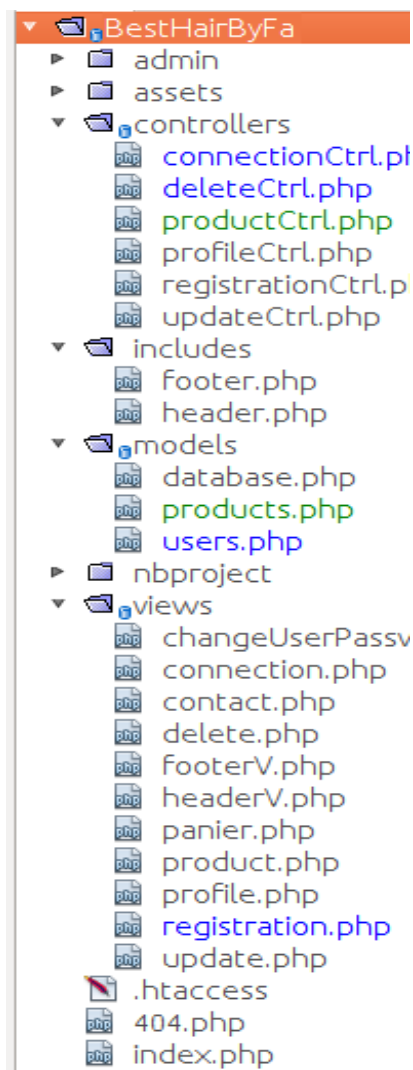


### 3. Description fonctionnelle et technique :

#### 3-1 Arborescence du site :

L'arborescence d'un site présente les différentes pages de celui-ci, organisées logiquement et hiérarchiquement sous forme d'un arbre. La première page doit être la page d'accueil (la racine), et les autres pages apparaissent ensuite dans un ordre logique.

L'arborescence aide l'internaute à comprendre la structure du site. La consultation et la mémorisation des pages seront ainsi plus faciles, rapides et efficaces.



### 3.2- Description fonctionnelle et technique :

Le site de “Best Hair By Fa” doit bien évidemment comporter une boutique en ligne, accessible depuis l’onglet “tous les produits”. Cette boutique en ligne doit par défaut afficher l’ensemble des produits mis en ligne depuis le back-office.

Une liste de filtre doit permettre d’affiner et de simplifier la recherche.

Depuis la liste de chaque produit, les utilisateurs doivent pouvoir :

- Ajouter le produit au panier
- Consulter la fiche du produit

#### Fiches produits :

Chaque produit est composé des informations structurées suivantes :

- Nom du produit
- Description du produit
- Référence du produit
- Taille du produit
- Prix du produit
- L'utilisateur pourra depuis ces pages produits effectuer un ajout et poster un commentaire.

#### Espace client :

Un utilisateur sera dans l’obligation de créer un compte client pour compléter un achat. Afin de créer un compte, l'utilisateur devra renseigner son nom, son prénom, sa date de naissance, son numéro de téléphone, son mail et son mot de passe.

### 3.3- Description fonctionnelle du Back-office :

Depuis le back-office, je vais effectuer l’ensemble des tâches quotidiennes nécessaires au bon fonctionnement du site.

### Fonctionnalités standards :

A la connexion sur le site, chaque utilisateur a accès à une page d'accueil lui affichant les différentes catégories de produits proposés par le site.

Pour pouvoir passer commande, les utilisateurs seront obligés de s'inscrire via un formulaire.

L'utilisateur inscrit doit pouvoir avoir accès à son espace client simplement et rapidement. Il aura aussi la possibilité de le modifier ou le supprimer s'il le souhaite.

- Publication d'articles : Mise en ligne de nouveaux articles
- Création de page : Elle doit être facilement réalisable. Le positionnement d'une page dans l'arborescence du site doit être facilement éditable.

### Fonctionnalités E commerce :

#### 1. Gestion du front :

- Gestion du catalogue de produits : Ajout et retrait des produits affichés dans la boutique.
- Gestion des promotions et des codes promos : Création et suppression de codes promo et offres temporaires sur des produits ou groupes de produits.
- Mise à jour de la page d'accueil : possibilité de mettre en avant des produits, promotions et articles.

#### 2. Gestion du Back-office :

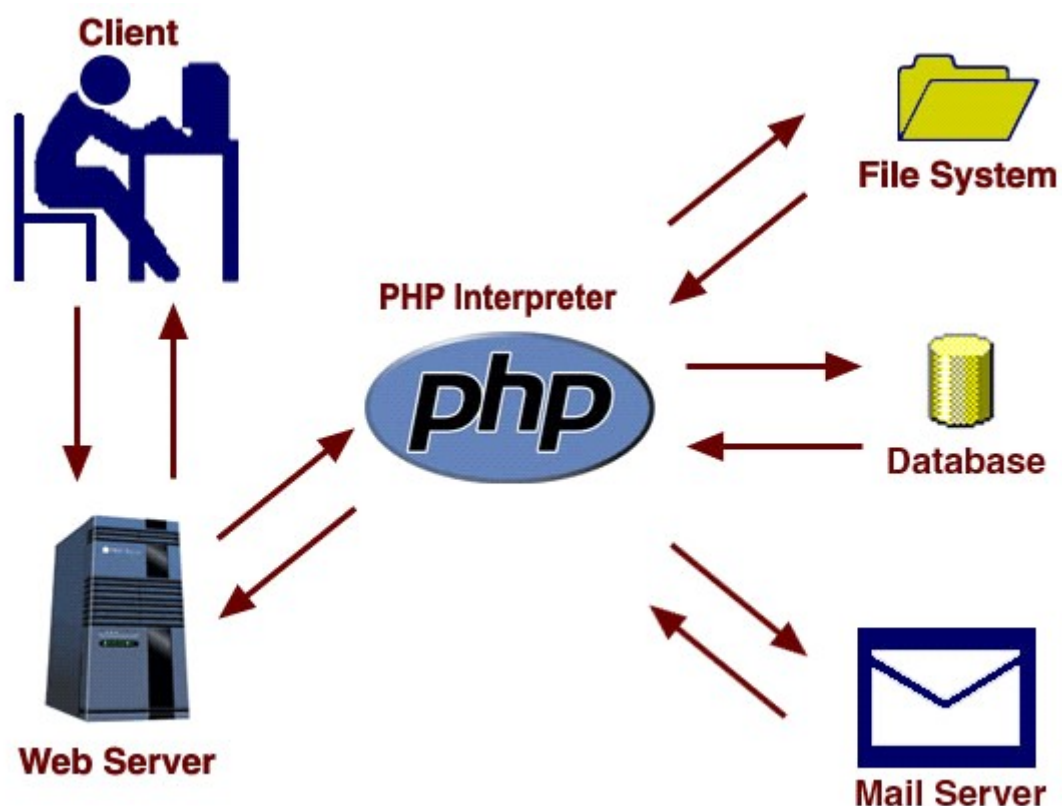
- Suivi des commandes : Affichage et modification de l'état (commande en cours de validation, validée, en cours de livraison, etc.)
- Suivi et gestion des paiements : Possibilité d'effectuer des remboursements et suivre le montant des commandes par période
- Gestion des stocks : Manuelle dans un premier temps. Il doit être possible de suivre depuis le Back-office le niveau des stocks pour chacun des produits.

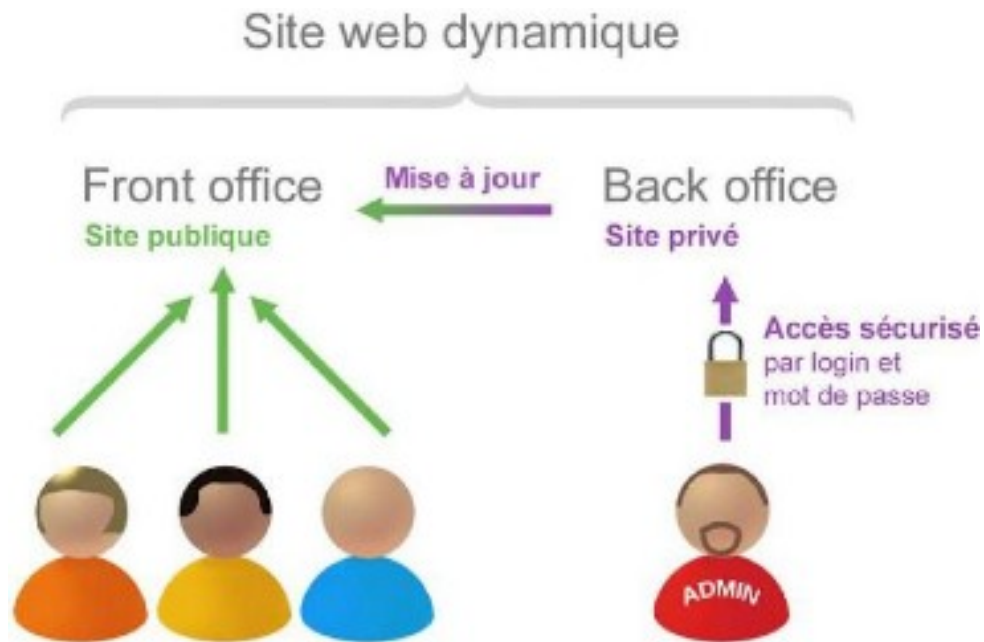
### 3.4- Choix des outils de développement :

Le développement d'un site web dynamique demande l'inclusion de script et de code pour garantir le dynamisme des pages et la liaison avec la base de données.

#### Le langage de programmation : PHP7

Pour réaliser mon site, j'ai utilisé le langage de programmation PHP dédié à la création des applications web dynamique, celui-ci nous l'avons manipulé dans un environnement de développement intitulé **PHP7** qui est un langage de script libre, très puissant, rapide et principalement exécuté par le compilateur php. Un script PHP est multiplateforme, très bon support des bases de données (MySQL, Oracle, Microsoft).





### Autres outils :

Les outils de développement web : HTML (HyperText Markup Langage), CSS (Cascade Style Sheet) seront aussi utilisés pour réaliser les interfaces du site. Ainsi Apache HTTP Server, souvent appelé Apache, est un logiciel permettant à des utilisateurs d'accéder à des pages web.

### Choix du Système de Gestion de Base de Données : MySQL

MySQL (My Structured Query Language) est un Système de Gestion des Bases de données (SGBD) Open Source très rapide, robuste et multi-utilisateurs. Le serveur MySQL supporte le langage de requêtes SQL, langage standard de choix des SGBD modernes. Il est facilement accessible en réseaux et supporte des connexions sécurisées grâce au protocole SSL. La portabilité du serveur MySQL lui permet de s'exécuter sur toutes les plateformes et d'être intégré à plusieurs serveurs web.

#### 4- Prestations attendues :

-Design : Réalisation de maquette pour la page d'accueil, la page d'inscription et la page des produits.

- Intégration : Intégration de ces maquettes
- Développement du site et du back-office
- Création et paramétrage de la base de données du site
- Hébergement : le projet sera développé, l'hébergeur doit être sélectionné et l'hébergement sera paramétré après.
- Marketing : Création de compte sur les réseaux sociaux :

- \*faire connaître ‘’Best Hair by Fa,’’

- \*vendre mes produits sur internet,

- \*donner à mes clients des conseils sur mes produits ou services,

- \*générer des contacts et de nouveaux prospects

#### Planning :

La date de notre examen étant fixée du 20 au 21 mars 2019, je fournirais une première version fonctionnelle tout en respectant les compétences du référentiel couvertes par le projet. Le projet sera ensuite développé intégralement.

#### Suivi du projet :

Le projet sera organisé et suivi sur un board [Trello](#), afin de garantir une bonne compréhension et lisibilité de l'avancement.

#### Propriété :

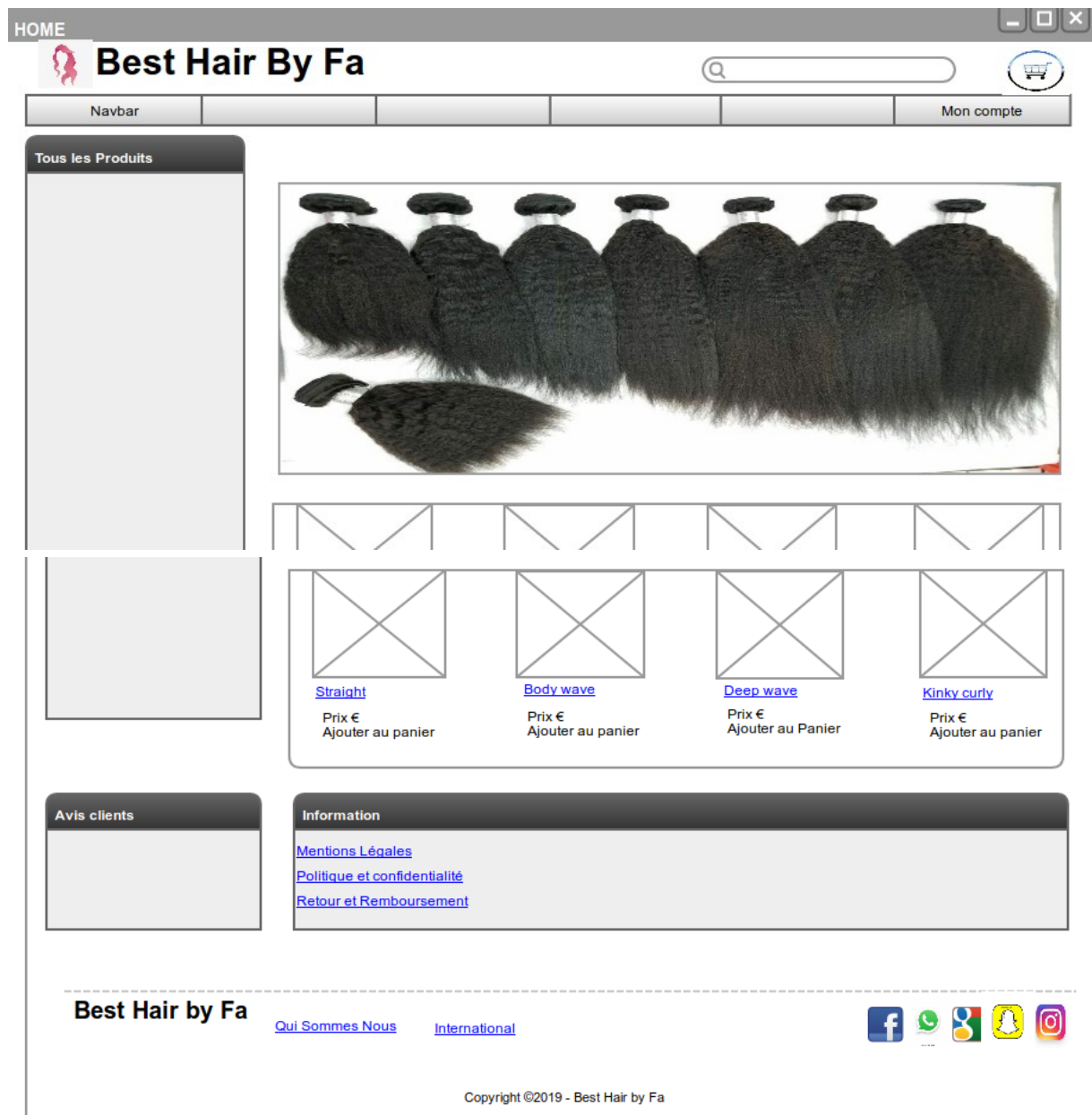
‘’Best Hair By Fa’’ sera propriétaire de l'ensemble des contenus de base (texte, plaquettes, logos, images et photos) et autres contenus créés pour le site.

## 4- Maquettage du site

Avant de me lancer dans le codage à proprement parler, j'ai réalisé une étude concurrentielle d'autres sites de vente de cheveux naturels, d'envergure équivalente ou plus importante.

Puis j'ai effectué un zoning et recenser les pages que je souhaite réaliser.

Voici la proposition de zoning que j'ai réalisé sur le site <https://cacoo.com> pour la page d'accueil, la page de connexion et la page de produits :





HOME

Best Hair By Fa

Navbar

JE SUIS DEJA ENREGISTRE

Email

Mot de passe

Mot de passe oublié

oK

JE SUIS UN NOUVEAU CLIENT

INFORMATIONS PERSONNELS

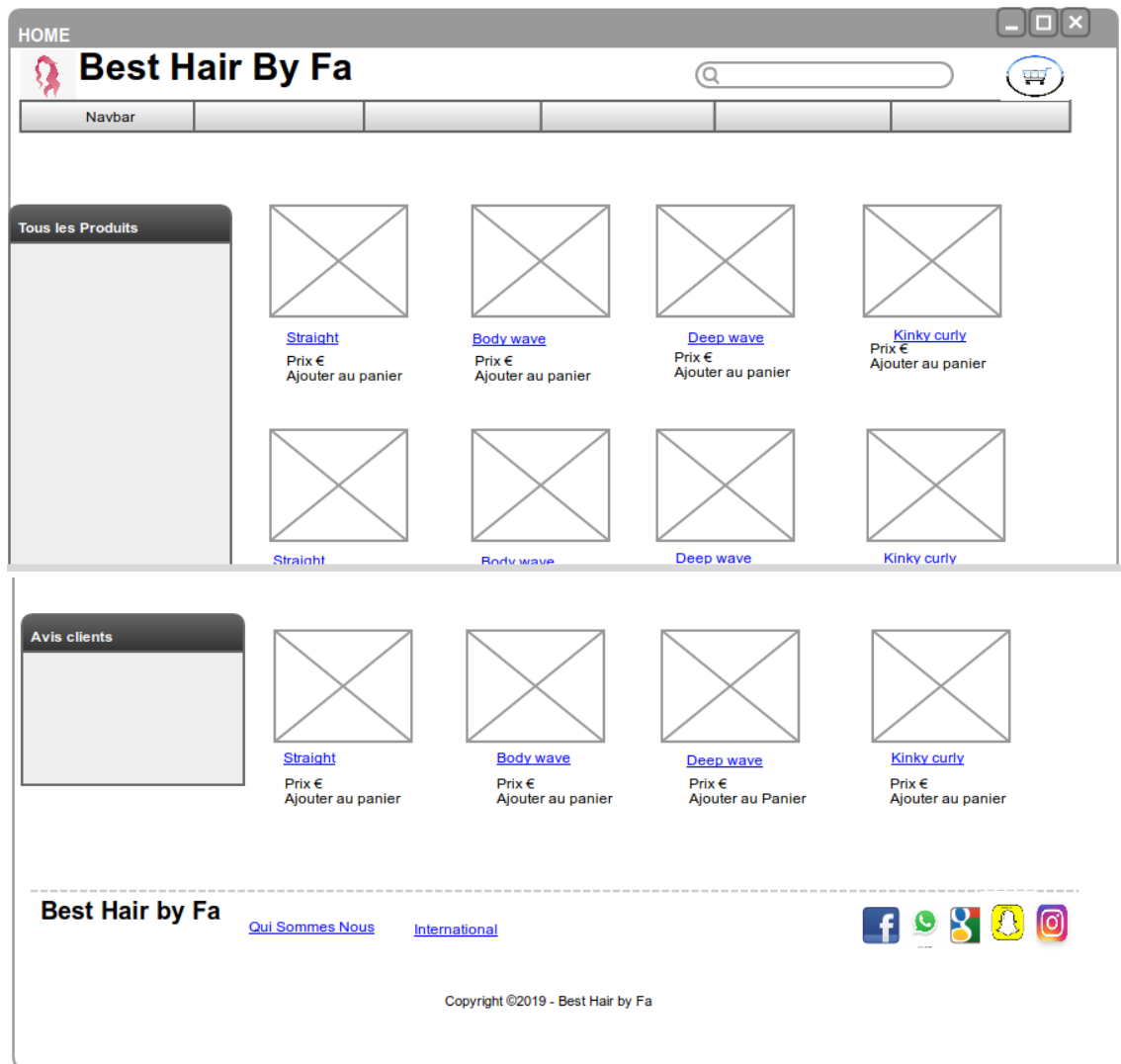
PARAMÈTRES DE CONNEXION

ENVOYER

Best Hair by Fa

[Qui Sommes Nous](#)[International](#)

Copyright ©2019 - Best Hair by Fa

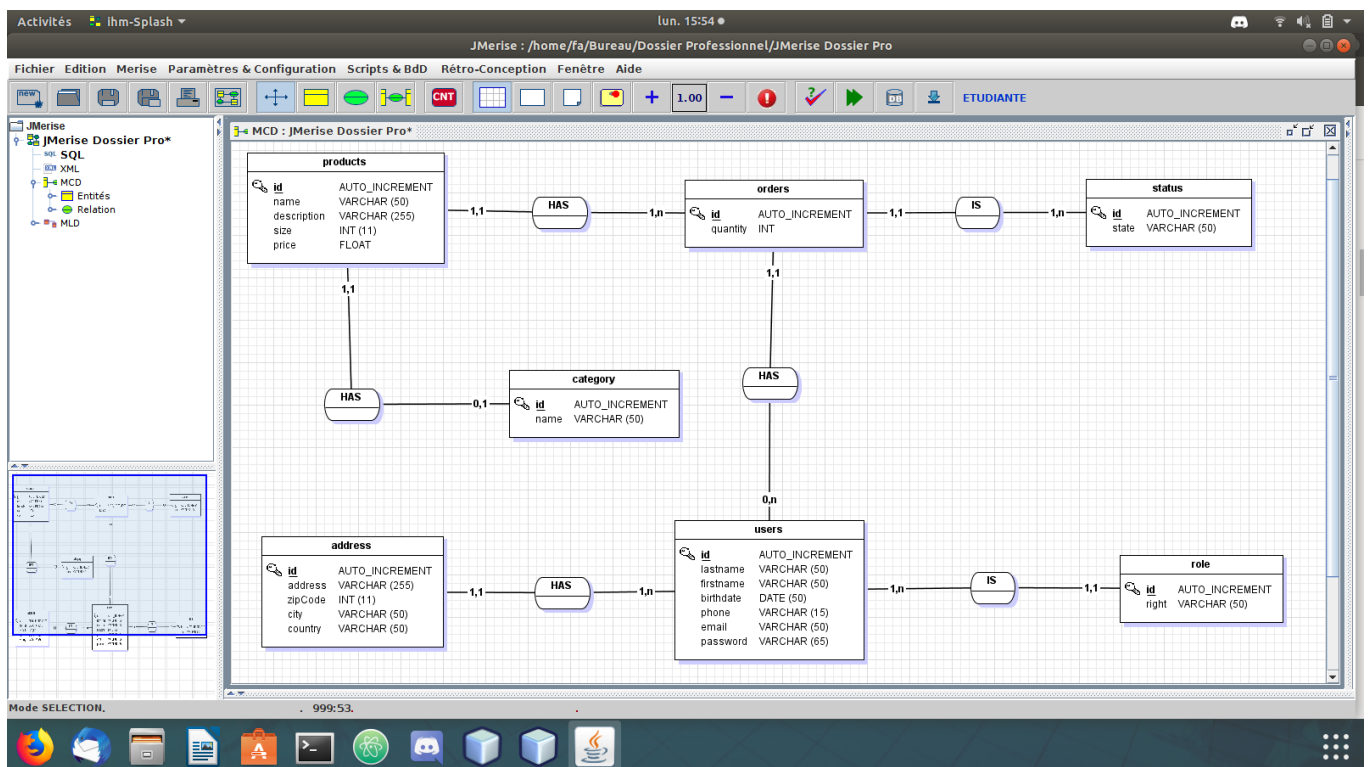


## 5- Conception de la base de données

Pour la création de la base de données (MySQL pour rappel), j'ai utilisé le logiciel de modélisation JMerise afin de permettre le stockage des nombreuses données issues du site.

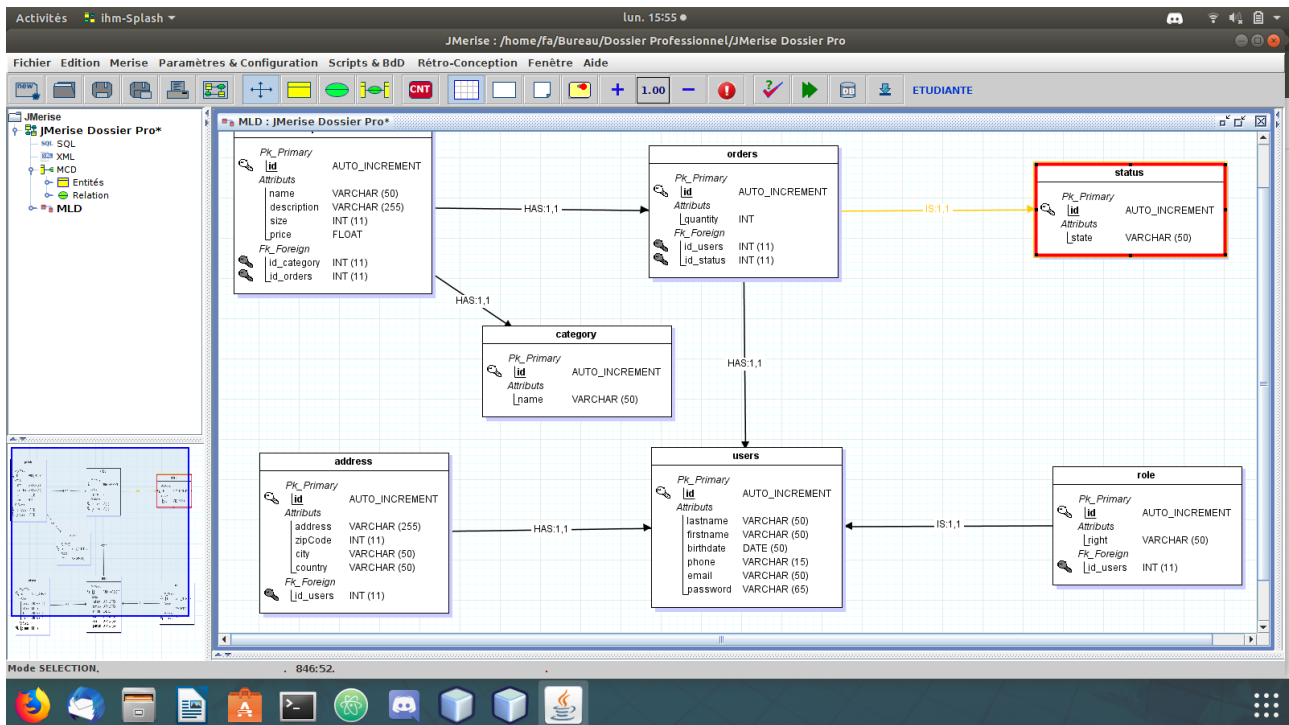
### -Le Modèle Conceptuel de Données (M.C.D) :

Le MCD ci-dessous montre la structure des tables utilisées dans la base de données et les relations entre eux.



## -Le Modèle Logique des Données (M.L.D) :

Le schéma du MLC a respecté les règles des cardinalités : telle entité est associée à telle autre.



Sur ce schéma on voit l'exemple de l'entité utilisateurs (users) qui peut avoir 1 et 1 seule commande (orders), par contre 1 commande ne peut être possédée que par 1 et 1 seul utilisateur.

## 6- Mise en place de la base de données

### 1- Mise en place de la base de données :

Une fois que la conception a été validée, j'ai mise en place ma base de données en effectuant un import du fichier SQL généré depuis le logiciel de modélisation JMerise vers le terminal. Je l'ai écrit selon le formalisme de SQL et j'ai ensuite mis en place les requêtes de création des tables.

Pour cela, il a fallu tout d'abord créer la base de données `bestHairByFa` dans le terminal :

Les contraintes seront écrites après la création des tables, de sorte de pouvoir les retirer (DROP), les modifier (ALTER, MODIFY), sans mettre en danger l'intégrité des autres données présentes dans la base.

```
CREATE DATABASE `bestHairByFa` ;
```

Ensuite, utiliser la base de données :

```
USE `bestHairByFa` ;
```

```
#-----  
#    Script MySQL.  
#-----
```

```
#-----  
# Table: users  
#-----
```

```
CREATE TABLE users(  
  id          Int Auto_increment NOT NULL ,  
  lastname    Varchar (50)       NOT NULL ,  
  firstname   Varchar (50)       NOT NULL ,  
  birthdate   Date               NOT NULL ,  
  phone       Varchar (15)       NOT NULL ,  
  email       Varchar (50)       NOT NULL ,  
  password    Varchar (65)       NOT NULL  
  ,CONSTRAINT users_PK PRIMARY KEY (id)  
)ENGINE=InnoDB;
```

```
#-----  
# Table: products  
#-----
```

```
CREATE TABLE products(  
  id          Int Auto_increment NOT NULL ,  
  name        Varchar (50)       NOT NULL ,  
  description  Varchar (255)     NOT NULL ,  
  size        Int                NOT NULL ,  
  price       Float NOT NULL  
  ,CONSTRAINT products_PK PRIMARY KEY (id)  
)ENGINE=InnoDB;
```

```
#-----  
# Table: category  
#-----
```

```
CREATE TABLE category(  
  id          Int Auto_increment NOT NULL ,  
  name        Varchar (50)       NOT NULL ,  
  id_users    Int                NOT NULL ,  
  id_products Int NOT NULL  
  ,CONSTRAINT category_PK PRIMARY KEY (id)  
  
  ,CONSTRAINT category_users_FK FOREIGN KEY (id_users) REFERENCES users(id)  
  ,CONSTRAINT category_products0_FK FOREIGN KEY (id_products) REFERENCES  
products(id)  
  ,CONSTRAINT category_users_AK UNIQUE (id_users)  
)ENGINE=InnoDB;
```

```
#-----  
# Table: status  
#-----
```

```
CREATE TABLE status(  
  id          Int Auto_increment NOT NULL ,  
  state       Varchar (50)       NOT NULL  
  ,CONSTRAINT status_PK PRIMARY KEY (id)  
)ENGINE=InnoDB;
```

```
#-----  
# Table: orders  
#-----
```

```
CREATE TABLE orders(  
  id          Int Auto_increment NOT NULL ,  
  quantity    Int      NOT NULL ,  
  id_users    Int      NOT NULL ,  
  id_status   Int      NOT NULL ,  
  id_products Int      NOT NULL  
  ,CONSTRAINT orders_PK PRIMARY KEY (id)  
  
  ,CONSTRAINT orders_users_FK FOREIGN KEY (id_users) REFERENCES users(id)  
  ,CONSTRAINT orders_status0_FK FOREIGN KEY (id_status) REFERENCES status(id)  
  ,CONSTRAINT orders_products1_FK FOREIGN KEY (id_products) REFERENCES  
products(id)  
)ENGINE=InnoDB;
```

```
#-----  
# Table: role  
#-----
```

```
CREATE TABLE role(  
  id          Int Auto_increment NOT NULL ,  
  right       Varchar (50)      NOT NULL ,  
  id_users    Int              NOT NULL  
  ,CONSTRAINT role_PK PRIMARY KEY (id)  
  
  ,CONSTRAINT role_users_FK FOREIGN KEY (id_users) REFERENCES users(id)  
)ENGINE=InnoDB;
```

```
#-----  
# Table: address  
#-----
```

```
CREATE TABLE address(  
  id          Int Auto_increment NOT NULL ,  
  address     Varchar (255)      NOT NULL ,  
  zipCode     Int NOT NULL ,  
  city        Varchar (50)      NOT NULL ,  
  country     Varchar (50)      NOT NULL ,  
  id_users    Int              NOT NULL  
  ,CONSTRAINT address_PK PRIMARY KEY (id)
```

,CONSTRAINT address\_users\_FK FOREIGN KEY (id\_users) REFERENCES users(id)  
)ENGINE=InnoDB;

Par mesure de sécurité, j'ai vérifié dans PHPMyadmin si ma base de données avait bien été créée. Le tableau ci-dessous nous le confirme.

| Table           | Action  | Lignes    | Type          | Interclassement          | Taille         | Perte      |
|-----------------|---|-----------|---------------|--------------------------|----------------|------------|
| address         | Afficher Structure Rechercher Insérer Vider Supprimer | 0         | InnoDB        | latin1_swedish_ci        | 32 Kio         | -          |
| category        | Afficher Structure Rechercher Insérer Vider Supprimer | 0         | InnoDB        | latin1_swedish_ci        | 16 Kio         | -          |
| orders          | Afficher Structure Rechercher Insérer Vider Supprimer | 0         | InnoDB        | latin1_swedish_ci        | 48 Kio         | -          |
| products        | Afficher Structure Rechercher Insérer Vider Supprimer | 0         | InnoDB        | latin1_swedish_ci        | 48 Kio         | -          |
| role            | Afficher Structure Rechercher Insérer Vider Supprimer | 0         | InnoDB        | latin1_swedish_ci        | 32 Kio         | -          |
| status          | Afficher Structure Rechercher Insérer Vider Supprimer | 0         | InnoDB        | latin1_swedish_ci        | 16 Kio         | -          |
| users           | Afficher Structure Rechercher Insérer Vider Supprimer | 13        | InnoDB        | latin1_swedish_ci        | 16 Kio         | -          |
| <b>7 tables</b> | <b>Somme</b>  | <b>13</b> | <b>InnoDB</b> | <b>latin1_swedish_ci</b> | <b>208 Kio</b> | <b>0 o</b> |

☐ Tout cocher Pour la sélection :   
 Imprimer Dictionnaire de données  
 Nouvelle table  
 Nom:  Nombre de colonnes:    
 Exécuter

Ensuite, il faut « peupler » la base de données :

Le tableau ci-dessous nous montre un exemple de peuplement de la table utilisateur.

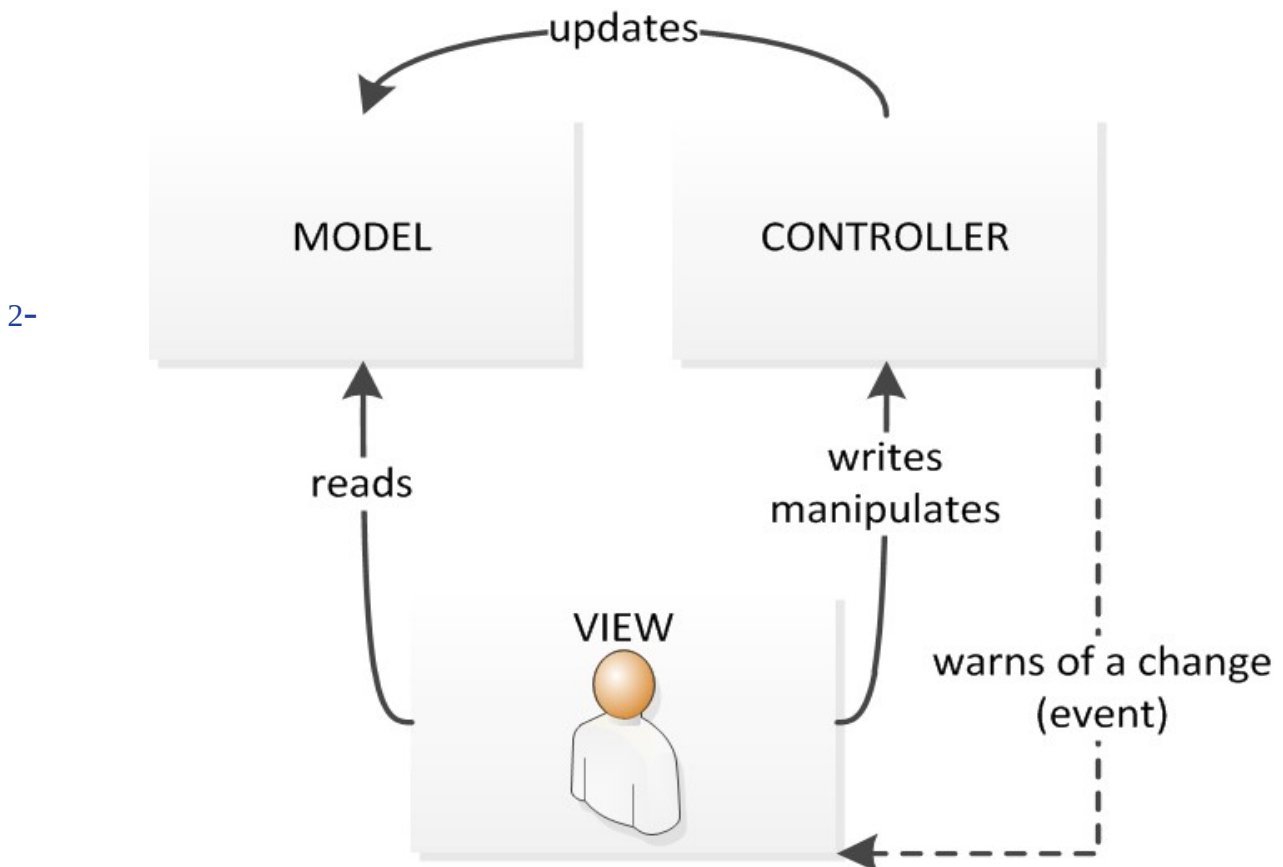
```

public $password;
//Methode magique __construct
public function __construct() {
    parent::__construct();
}
/**
 *
 * @return boolean
 */
    public function addUser(){
        // requête permettant d'insérer les valeurs de l'enregistrement.
        $query = 'INSERT INTO `users`(`lastname`, `firstname`, `birthdate`, `phone`, `email`, `password`) VALUES (:lastname, :firstname,
        // On prépare la requête.
        $addUser = $this->db->prepare($query);
        // On attribue les valeurs via bindValue et on récupère les attributs de la classe via $this
        $addUser->bindValue(':lastname', $this->lastname, PDO::PARAM_STR);
        $addUser->bindValue(':firstname', $this->firstname, PDO::PARAM_STR);
        $addUser->bindValue(':birthdate', $this->birthdate, PDO::PARAM_STR);
        $addUser->bindValue(':phone', $this->phone, PDO::PARAM_INT);
        $addUser->bindValue(':email', $this->email, PDO::PARAM_STR);
        $addUser->bindValue(':password', $this->password, PDO::PARAM_STR);
        // On execute la requete une fois prête.
        if ($addUser->execute()){
            return true;
        }
    }
  
```



Pour le développement de `bestHair ByFa` l'architecture va suivre celle du MVC (Modèle/Vue/Contrôleur). C'est une façon de d'organiser l'interface graphique d'un programme, elle consiste à distinguer 3 entités distinctes qui sont : le modèle, la vue et le contrôleur ayant chacun un rôle précis dans l'interface.

- Un modèle (Model) : contient les données à afficher.
- Une vue (View) : contient la présentation de l'interface graphique.
- Un contrôleur (Controller) : contient la logique concernant les actions effectuées par l'utilisateur (gestion des événements et synchronisation).



## 2-Connexion à la base de données :

La classe Database est une classe abstraite. Elle est héritée par tous les modèles. Elle permet de factoriser la connexion à la base de données pour entrer les identifiants une seule et unique fois.

```
<?php
try
{
/**
 * Classe permettant de se connecter à la base de données
 * Ses enfants hériteront de ses méthodes et attributs.
 */
class Database {
    public $db;

    public function __construct(){
        $this->db = new PDO('mysql:dbname=bestHair;host=localhost', 'fa', 'Magatte2012', [PDO::ATTR_ERRMODE=>PDO::ERRMODE_EXCEPTION]);
    }
}

// On teste les erreurs avec le try/catch, si tout est bon, on est connecté à la base de données
}
catch (Exception $e)
{
    die('Erreur : ' . $e->getMessage()); // Autrement, un message d'erreur est affiché
}
?>
```

## 6- Réalisations comportant les extraits de code

Le premier exemple de réalisation va être l'inscription d'un nouveau client.

### models/users.php

```

    .php
/** extends permet l'heritage entre 2 classes.
 * Donc users hérite de Database ce qui permet d'utiliser son constructeur et donc d'établir la connection à la base de d
 */
class users extends Database{
    public $id;
    public $lastname;
    public $firstname;

    public $birthdate;
    public $phone;
    public $email;
    public $password;
    //Methode magique __construct
    public function __construct() {
        parent::__construct();
    }
    /**
     *
     * @return boolean
     */
    public function addUser(){
        // requête permettant d'insérer les valeurs de l'enregistrement.
        $query = 'INSERT INTO `users`(`lastname`, `firstname`, `birthdate`, `phone`, `email`, `password`) VALUES (:lastname
        // On prépare la requête.
        $addUser = $this->db->prepare($query);
        // On attribue les valeurs via bindValue et on récupère les attributs de la classe via $this
        $addUser->bindValue(':lastname', $this->lastname, PDO::PARAM_STR);
        $addUser->bindValue(':firstname', $this->firstname, PDO::PARAM_STR);
        $addUser->bindValue(':birthdate', $this->birthdate, PDO::PARAM_STR);
        $addUser->bindValue(':phone', $this->phone, PDO::PARAM_INT);
        $addUser->bindValue(':email', $this->email, PDO::PARAM_STR);
        $addUser->bindValue(':password', $this->password, PDO::PARAM_STR);
        // On execute la requete une fois prête.
        if ($addUser->execute()){
            return true;
        }
    }
}

```

views/registration.php

```

<?php
    session_start();
    // inclue le fichier header.php qui se trouve dans le dossier includes.
    include'../views/headerV.php';
    require'../controllers/registrationCtrl.php';
    ?>
    <div class="container">
        <div class="row">
            <div class="col-sm-8 m-auto">
                <div class="card bg-light mt-5">
                    <h1 class="card-title text-center text-black mt-3 py-4">JE SUIS UN NOUVEAU CLIENT</h1>
                    <p class="text-center">Merci de bien renseigner tous les champs de saisie</p>
                    <?php
                        if(isset($errors) && count($errors) != 0){
                            foreach ($errors as $error){
                                ?>
                                <div class="alert alert-danger">
                                    <p><?= $error ;?></p>
                                </div>
                                <?php
                                    }
                            class="card-body bg-light m-auto py-2">
                            <form action="registration.php" method="post" class="form">
                                <label for="lastname">Nom</label>
                                <input class="form-control" type="text" id="lastname" name="lastname" value="<?= (isset($_POST['lastname'])) ? $_POST['lastname'] : '' ?>" />
                                <label for="firstname">Prénom</label>
                                <input class="form-control" type="text" name="firstname" id="firstname" value="<?= (isset($_POST['firstname'])) ? $_POST['firstname'] : '' ?>" />
                                <label for="birthdate">Date de naissance</label>
                                <input class="form-control" type="date" name="birthdate" id="birthdate" value="<?= (isset($_POST['birthdate'])) ? $_POST['birthdate'] : '' ?>" />
                                <label for="phone">Téléphone</label>
                                <input class="form-control" type="text" name="phone" id="phone" value="<?= (isset($_POST['phone'])) ? $_POST['phone'] : '' ?>" />
                                <label for="emailAddUser">Mail</label>
                                <input class="form-control" type="email" name="email" id="email" value="<?= (isset($_POST['email'])) ? $_POST['email'] : '' ?>" />
                                <label for="Mot de passe">Mot de passe</label>
                                <input class="form-control" type="password" name="password" id="password" value="<?= (isset($_POST['password'])) ? $_POST['password'] : '' ?>" />
                                <div class="mb-5 mt-5">
                                    <button class="btn btn-secondary col-5 mr-3" type="submit" name="submit" id="submit">Valider</button>
                                    <a href="javascript:history.go(-1)" class="btn btn-secondary col-5 ml-3">Retour</a>
                                </div>
                            </form>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

controllers/registrationCtrl.php

```

<?php
// inclue une fois le fichier database.php qui est dans le dossier models.
// inclue une fois le fichier UserClass.php qui est dans le dossier models.
require '../models/database.php';
require '../models/users.php';

// Regex permettant la sécurisation du code.
$regexText = '/^[A-Za-zàèìòùâëïôûäëïöüÿÄÉÍÓÚáëíöüÿÀÉÍÓÚāāñōāāñōäëíöüÿÆÏÖÜŸÇÇßøÀÀÆæ" \'\'-]+$/i';
$regexPhone = '/^\d{10}$/';
$regexDate = '/^[0-9]{4}-[0-9]{2}-[0-9]{2}$/';
$regexLogin = '/^[a-zA-Z0-9_\.- @éèàùëïôê]{6,20}$/';
$regexMail = '/^[A-z0-9._%+-]+\[\@]{1}[A-z0-9.-]+\[\.]{1}[A-z]{2,4}$/i';
$formError = array();

// On instancie l'objet user dans la classe users.
$user = new users();

// On vérifie avec la condition si $_POST comporte plus de 0 données alors on exécute le contenu du if.
if (COUNT( $_POST ) > 0){
    // on initialise les variable et on sécurise par htmlspecialchars
    $lastname = htmlspecialchars($_POST['lastname']);
    $firstname = htmlspecialchars($_POST['firstname']);
    $birthdate = htmlspecialchars($_POST['birthdate']);
    $phone = htmlspecialchars($_POST['phone']);
    $email = htmlspecialchars($_POST['email']);
    $password = password_hash($_POST['password'], PASSWORD_BCRYPT); //hachage du password.
    // Vérification du nom selon la regex + filtrage de l'email;
    if (!empty($_POST['lastname'])){
        if(preg_match($regexText, $lastname)){
            $user->lastname = $lastname;
        } else {
            $formError['lastname'] = 'La saisie de votre nom est invalide';
        }
    }
}

```

cf

suite cf-annexe

## 7-Présentation du jeu d'essai

L'insertion de données dans ces tables sera gérée par la partie utilisateur (création de profil, modification de profil, ...)

- un formulaire d'inscription, qui vérifie la bonne saisie des informations de l'utilisateur et envoie ces données à la base de données.
- un formulaire, qui après inscription permet à l'utilisateur de se connecter à son espace.

### JE SUIS UN NOUVEAU CLIENT

Merci de bien renseigner tous les champs de saisie

Nom

Prénom

Date de naissance

Téléphone

Mail

Mot de passe

### JE SUIS DEJA INSCRIT

Adresse email :

Mot de passe :

[\\* J'ai oublié mon mot de passe](#)

- un profil utilisateur qui, après connexion, affiche les données reçues depuis la base de données.

- un formulaire de modification du profil utilisateur, gère l'affichage des données à modifier puis l'update suite à la validation de la modification.

TEST

Testeur

2000-02-01

602030405

test@test.fr

## Mon PROFIL

[Modifier](#)  
[Supprimer](#)  
[Se Deconnecter](#)

## MISE A JOUR

Merci de bien renseigner tous les champs de saisie

Nom

Prénom

Date de naissance

Téléphone

Mail

✓ Affichage des lignes 0 - 0 (total de 1, Traitement en 0.0004 secondes.)

SELECT \* FROM `users`

Profilage [ Éditer en ligne ] [ Modifier ] [ Expliquer SQL ] [ Créer cc

☐ Tout afficher
 

Nombre de lignes :

Filtrer les lignes:

Options

|   | id | lastname | firstname | birthdate  | phone     | email        | password   |
|---|----|----------|-----------|------------|-----------|--------------|--|
| <input type="checkbox"/> <span>Modifier</span> <span>Copier</span> <span>Effacer</span> | 44 | TEST     | Testeur   | 2000-02-01 | 602030405 | test@test.fr | \$2y\$10\$V5VnnRYaTV8vT3jXzjdIn.hsCVqhfiy9EB3f/jmWSgO... |

↑
☐ Tout cocher
 

Pour la sélection : Modifier Copier Effacer Export

☐ Tout afficher
 

Nombre de lignes :

Filtrer les lignes:

Opérations sur les résultats de la requête

Imprimer
Copier dans le presse-papier
Export
Afficher le graphique
Créer une vue

Pour chaque formulaire, j'ai créé une classe qui envoie une requête en méthode POST au fichier.php correspondant.

Au clic sur le bouton « submit » :

Etape1 :

- il vérifie que tous les champs sont bien remplis
- une requête est envoyée à la base de données pour vérification

Etape2 :

- On vérifie que les 2 champs ont la même valeur
- Si les données saisies dans les champs sont conformes (en utilisant htmlspecialchars + Regex).

Une fois la saisie conforme, on se connecte et on a accès au profil utilisateur.



## 8-Situation ayant nécessité une recherche à partir de site anglophone

Cette compétence a été mis en œuvre tout au long de notre formation.

Compréhension orale : tutoriels vidéo sur tous les langages qu'on a étudiés

Expression écrite :

- Requêtes effectuées en anglais sur les moteurs de recherche.
- [stackoverflow.com](https://stackoverflow.com)
- [W3schools.com](https://www.w3schools.com)

## 9- Extrait du site anglophone plus traduction en français :

Traduction effectuée depuis le site tutorial.net

### Overflow

As already explained, a block level element in CSS will, by default, take up all the horizontal space as well as all the vertical space it needs to fit the content inside of it. However, what happens if we define a vertical size that is not enough to contain the content? Let's have a look:

Translate

Source:

As already explained, a block level element in CSS will, by default, take up all the horizontal space as well as all the vertical space it needs to fit the content inside of it. However, what happens if we define a vertical size that is not enough to contain the content? Let's have a look:

Translation [French]:

Comme déjà expliqué, un élément de niveau bloc en CSS occupera, par défaut, tout l'espace horizontal ainsi que tout l'espace vertical dont il a besoin contenu à l'intérieur de celui-ci. Cependant, que se passe-t-il si nous définissons une taille verticale qui n'est pas suffisante pour contenir le contenu? Regardons:

**Important:** Please don't submit machine-translations (e.g. from Google Translate) - only manual translations will be accepted!

Submit translation

## Conclusion

Le développement de mon site m'a permis de comprendre toutes les étapes nécessaires à la création d'un site web : e la création de la maquette au développement front-end en passant par le cahier des charges et la structure 'MVC'.

Compte tenu de l'avancée sur le projet, je considère avoir dans l'ensemble collé au cahier des charges que je m'étais fixé au début.

Ce projet m'a permis de me plonger plus amplement sur le développement d'un projet web d'envergure. J'ai pu apprendre beaucoup de mes camarades et de mes formateurs, autant sur l'aspect technique que sur l'esprit d'équipe. Étant dès le début dans un environnement agile basé sur l'entraide et la communication, j'ai pu énormément progresser dans un environnement idéal.