
Global Prestation Annexes

SOMMAIRE

Introduction	3
1.1 Résumé du dossier de projet	3
Présentation	
2.1 Présentation personnelle	3
2.2 Présentation de La Plateforme	4
Projet entreprise	4
3.1 Présentation de l'entreprise	4
3.2 Présentation du projet	5
3.3 Compétences utilisées	6
3.4 Cahier des charges	7
3.6 Spécifications techniques	10
Développement	10
4.1 Réalisations	11
4.1.1 Maquettage du site	11
4.1.3 Développement de la partie back-end	11
4.3 Veille et sécurité	21
4.4 Difficultés rencontrées	23
Conclusion	24

Introduction

1.1 Résumé du dossier de projet

Ce dossier présente une partie du travail réalisé lors de ma formation à La Plateforme à Marseille, à savoir le développement d'un site pour une entreprise regroupant un

ensemble de compétences pour mon titre RNCP de niveau 5.

Vous y retrouverez donc un panel de compétences mises en avant lors de ce projet d'une durée de 5 mois, ainsi que son déroulement, les attentes attendues par la personne en charge de ce projet et sa réalisation finale.

Présentation

2.1 Présentation personnelle

Je m'appelle ZIZZO Paolo, j'ai 19 ans et j'habite à Coudoux.

Sortant d'un Bac Professionnel S.N (systèmes numériques) au lycée

Pierre Mendès France, je voulais continuer mes études dans le domaine du développement web que j'affectionne particulièrement. C'est pourquoi, en cherchant sur internet, je suis tombé sur l'école de La Plateforme qui propose une formation en tant que développeur web / web mobile sur 2 ans (la première année s'effectue à l'école, la deuxième année s'effectue sous forme d'alternance).

2.2 Présentation de La Plateforme

La Plateforme est une école du numérique et des nouvelles technologies co-fondée avec le Club Top 20 réunissant les grandes entreprises de la Métropole Aix Marseille.

Elle comprend une offre de formations diversifiées destinées à former des codeurs et développeurs web, des experts en sécurité, des ingénieurs spécialisés en Intelligence Artificielle, et des cadres d'entreprises au travers de cycles de formations continues.

Je suis actuellement la formation "Coding School" qui s'adresse à tous les passionnés sans diplômes et sans limite d'âge. C'est une formation gratuite qui offre donc ses chances à toutes et à tous.

Projet entreprise

3.1 Présentation de l'entreprise

Global Prestations annexes est une holding de transport de marchandises proposant un accompagnement structurel au nouveau créateur d'entreprise transport en leur proposant :

- La gestion intégrale de l'administratif (facture, comptabilité, litiges...)
- La mise à disposition du gestionnaire transport (capacité professionnelle – attestataire transport)
- Logiciels de suivi activité (géolocalisation, lettre de voiture dématérialisée, contrôle des heures conducteurs...)
- Proposition de marché et accompagnement vers développement stratégique avec des affrètements transports, contrat client

Lorsqu'une personne souhaite devenir transporteur il faut au préalable avant de s'immatriculer au niveau des greffes de tribunal de commerce, faire une demande d'autorisation d'exercer la profession de transporteur routier de marchandises à la DREAL ou DRIEA suivant les régions. La DREAL (Direction Régionale de l'Environnement, de l'Aménagement et du Logement) ou DRIEA (Direction régionale et interdépartementale de l'Équipement et de l'Aménagement) est une autorité

dépendante du Préfet de chaque Région qui contrôle et surveille que les activités dans le domaine du transport respecte les lois entrée en vigueur, auquel cas la DREAL ou DRIEA a le pouvoir de faire cesser toute activité ou mettre en demeure de conformité les établissements en infraction.

3.2 Présentation du projet

Afin de faciliter la mise en relation entre entrepreneurs et gestionnaires, commissionnaires transport, l'idée d'une plateforme, façon "Le Bon Coin" est née du constat qu'aucun site gratuit spécialisé à cet effet n'existait !

Ce projet professionnel consiste donc à créer une plateforme "gestionnairetransport.com" par laquelle la mise en relation entre gestionnaires, commissionnaires et entrepreneurs transports est facilitée. Des utilisateurs inscrits pourront déposer des annonces et fournir une attestation certifiant leur éligibilité (vérifié par l'admin avant d'être postée).

La plateforme a pour principal but d'être simple d'utilisation et de faciliter au maximum les échanges entre les différents utilisateurs. A l'arrivée sur le site l'utilisateur pourra voir l'ensemble des annonces postées ainsi qu'un filtre permettant de faciliter sa recherche mais également un système de favoris afin de sauvegarder des annonces pour y revenir plus tard.

Pour terminer, ce site possède une interface administrative accessible au gérant du site permettant :

- la consultation de l'ensemble des annonces en attente de validation
- l'archivage de la base des utilisateurs sous forme de liste (possibilité de bannir un utilisateur 30 jours / permanent)
- le pilotage des services en les ajoutant et/ou en supprimant (comme

l'option payante de mise en avant des annonces)

- la visualisation et la gestion des paiements effectués par les utilisateurs.

3.3 Compétences utilisées

Pour réaliser ce projet, de nombreuses connaissances étaient nécessaires car le but de ce projet étant de mettre en pratique l'ensemble des compétences / outils utilisé(e)s pendant l'année.

Langages HTML et CSS

J'ai utilisé HTML / CSS comme structures des pages à développer et pour l'intégration.

Le CSS m'a aidé pour appliquer différents style à mes formulaires et à mes pages.

Langages PHP et SQL

Le PHP est utilisé pour les requêtes avec la base de données et l'affichage des informations récupérées. De nombreuses requêtes avec la base sont réalisées pour les formulaires de connexion / inscription, puis aussi pour la partie profil.

i

Langages JavaScript / jQuery / JSON

Le JavaScript a été utilisé pour la vérification des formulaires, si un champ n'est pas rempli ou incorrect, une erreur sera affichée sans rafraichissement de la page.

Pour cela, j'utilise Ajax, une communication est faite lorsque j'appuie sur un bouton de validation du formulaire avec un fichier php, qui lui va exécuter une requête sql et renvoyer le résultat sous forme de tableaux json.

En fonction du résultat j'affiche un contenu différents dynamiquement en JavaScript.

3.4 Cahier des charges

Les objectifs du site Internet :

Proposer un outil gratuit qui permettrait au créateur de société transport de pouvoir utiliser la capacité de transport d'une tierce personne en :

- facilitant la mise en relation entre futur transporteur et gestionnaire transport.
- recruter nos futures filiales (intégration d'un groupe avec mise à disposition du gestionnaire transport).

La cible :

- Créateur de société transport
- Société transport en développement
- Gestionnaire transport pour mise à disposition de leur capacité
- Expert comptable et avocat pour pub

L'architecture du site Internet :

ACCUEIL:

proposer des profils directement comme présenté ci-dessous Filtrer un attestataire pour la capacité marchandises/voyageur/commissionnaire Permettre le dépôt d'annonce pour les gestionnaires transport Intégration latérale bandeau pub (avec renvoi sur site correspondant) + Modèle de document officiel payant Proposition des cookies / CGV /

Contact

Les annonces des gestionnaires :

Accueil : Visuel synthétique :

- Photo

-
- Pseudo
 - Lieu d'action
 - Type de capacité
 - Prix de la mise à disposition
 - Téléphone
 - Email

Dépôt d'annonce des gestionnaires :

- Nom
 - Prénom
 - Pseudo
 - Date naissance
 - Adresse
 - Email
 - Tel
 - Type de capacités avec possibilité d'intégrer les attestations pour vérification
 - Tarif € ou à négocier
 - Lieu d'action (France entière, paris, Bouches du Rhône) Remplissage libre
 - Descriptif de parcours / expériences /objectifs
 - Téléchargement photos Jpeg/ PDF
 - Possibilité de faire payer comme le principe de LEBONCOIN avec remonter en tête de liste / Mise en évidence / Rajout de photo... Modérateur : Global Prestations Annexes
- se réserve le droit après contrôle des pièces d'accepter ou refuser toute diffusion sur le site

Concernant l'organisation du projet, nous étions 3 au départ puis nous avons finis à 2

(car la deuxième personne à été basculé en deuxième promotion). Afin de se répartir le mieux les tâches, nous avons fait en fonction des compétences de chacun, l'un pour la partie "annonces" du projet et moi pour la partie profil et informations personnelles .

Nous avons utilisé GitHub afin de pouvoir envoyer nos fichiers versionnés dessus et de pouvoir les récupérer depuis n'importe quel appareil mais surtout de pouvoir mettre une description lors d'un commit / push d'un fichier afin de savoir ce qui a été fait pour chaque push de fichier.

GitHub m'a été très utile notamment lors du confinement pour que je puisse voir et récupérer les parties de mes coéquipiers afin mettre en commun les différents codes...

Toutes les 2 semaines , nous avons organisé avec la cliente, des réunions via google meet afin qu'elle puisse voir notre avancé, donner son avis en voyant notre site grâce au partage d'écran. Ainsi nous avons ajouté des fonctionnalités en fonctions de ce qu'elle demandait.

3.6 Spécifications techniques

Versioning et stockage

Pour ce projet nous avons utilisé Git comme logiciel de versionning, pour stocker notre code versionné sur un serveur distant afin de pouvoir travailler plus facilement dessus chacun de notre côté. Pour cela nous avons créé plusieurs branches sur GitHub, avec nos noms puis la branche master qui elle est réservée pour le projet final. Pour chaque nouvelle tâche effectuée, je faisait un push sur ma branche via l'outil TortoiseGit afin de mettre à jour mon avancé.

Technologies

- PHP : Langage back-end
- HTML / CSS : Structure du site
- JavaScript / jQuery : Création contenu dynamique
- PHPmyAdmin : Base de données
- Bootstrap : Responsive et design du site

Développement

4.1 Réalisations

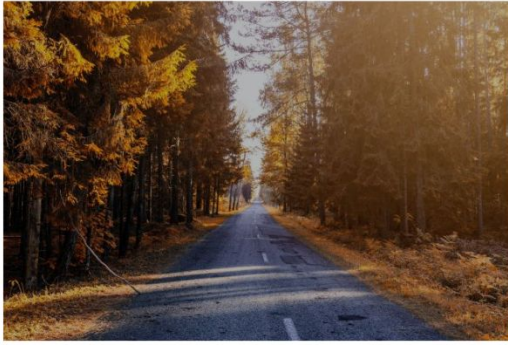
4.1.1 Maquettage du site

En début de projet, nous avons commencé à réaliser une maquette du site via l'application Balsamiq

afin de faire un aperçu du design et du jeu de couleur de celui-ci. Au final nous n'avons pas choisis Balsamiq car au niveau du design, il était pour nous moins représentatif.

4.1.3 Développement de la partie front-end.

J'ai fait les pages d'inscription/connexion en partant d'un template formulaire trouvé sur internet que je trouvais très complet. J'ai ensuite utilisé ce formulaire pour chaque page qui en avait besoin (connexion, inscription, profil...). Les formulaires sont différents de ceux réalisés dans la maquette mais ils sont plus complets et ont donc été choisis.



CONNEXION

VALIDER

Besoin d'un compte ? [INSCRIVEZ-VOUS](#)
Mot de passe oublié ? [CLIQUEZ ICI](#)

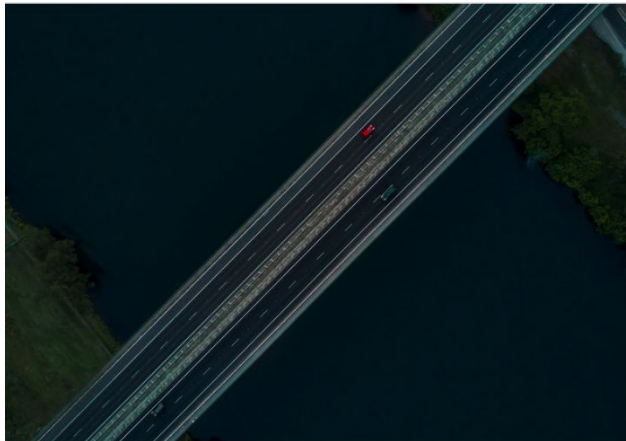
L'ensemble du site possède le panel de couleurs gris et vert de code hexadécimal suivant :

#17A68

#0D836

et #21202

Comme expliqué précédemment, mon JavaScript permettra de vérifier si les formulaires sont bien remplis lors de leurs validation et si les champs complétés sont dans les normes cela permettra au JavaScript d'effectuer une requête avec la base de données et d'afficher du contenu dynamiquement en fonction du résultat. (exemple : login/mot de passe incorrect, encadrement des inputs en rouge).



CONNEXION

***Login ou mot de passe incorrect**

VALIDER

Besoin d'un compte ? [INSCRIVEZ-VOUS](#)
Mot de passe oublié ? [CLIQUEZ ICI](#)



INSCRIPTION

*Des champs sont vides

Mr ☐

Mme ☐

kzen ckle

Prénom

Adresse

Email

Photo de profil (optionnel)

Date de naissance :

Rechercher

j / mm / aaaa

Login (5 caractères minimum)

Mot de passe

Confirmation mot de passe

VALIDER

Vous avez un compte ? [SE CONNECTER](#)

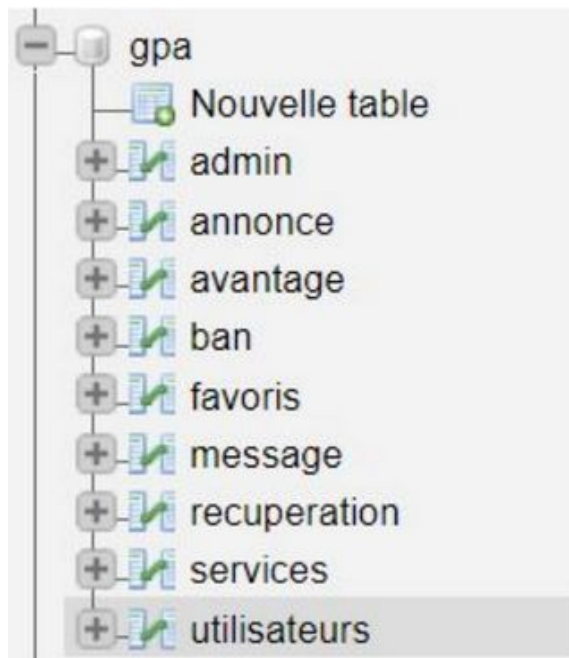
4.1.3 Développement de la partie back-end

Pour la partie back-end, nous avons tout d'abord commencer à créer la base de données avec phpmyadmin. J'ai créé les différentes tables que j'avais maquettée au préalable avec les différentes informations qu'elles contiennent.

Pour exemple, voici la table utilisateurs qui contiendra donc toutes les informations correspondantes aux inscrits du site.

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/>	1 id	int(11)			Non	Aucun(e)		AUTO_INCREMENT	 Modifier  Supprimer  Plus
<input type="checkbox"/>	2 genre	varchar(5)	latin1_swedish_ci		Non	Aucun(e)			 Modifier  Supprimer  Plus
<input type="checkbox"/>	3 nom	varchar(20)	latin1_swedish_ci		Non	Aucun(e)			 Modifier  Supprimer  Plus
<input type="checkbox"/>	4 prenom	varchar(20)	latin1_swedish_ci		Non	Aucun(e)			 Modifier  Supprimer  Plus
<input type="checkbox"/>	5 adresse	varchar(255)	latin1_swedish_ci		Non	Aucun(e)			 Modifier  Supprimer  Plus
<input type="checkbox"/>	6 email	varchar(255)	latin1_swedish_ci		Non	Aucun(e)			 Modifier  Supprimer  Plus
<input type="checkbox"/>	7 naissance	varchar(10)	latin1_swedish_ci		Non	Aucun(e)			 Modifier  Supprimer  Plus
<input type="checkbox"/>	8 login	varchar(15)	latin1_swedish_ci		Non	Aucun(e)			 Modifier  Supprimer  Plus
<input type="checkbox"/>	9 password	varchar(255)	latin1_swedish_ci		Non	Aucun(e)			 Modifier  Supprimer  Plus
<input type="checkbox"/>	10 profil	varchar(255)	latin1_swedish_ci		Non	Aucun(e)			 Modifier  Supprimer  Plus
<input type="checkbox"/>	11 date	date			Oui	NULL			 Modifier  Supprimer  Plus

La base de données finale comprend 9 tables :



Une fois la base de données créée, nous avons commencé à développer la partie connexion / inscription afin d'inscrire des utilisateurs. Lorsqu'un utilisateur s'inscrit, il doit renseigner l'ensemble des informations suivantes : genre, nom, prénom, adresse, email, login, mot de passe et photo de profil (optionnel).

```

<form class="login100-form validate-form" enctype="multipart/form-data">
  <span class="login100-form-title p-b-70">
    Inscription
  </span>
  <div class="genre rs1-wrap-input100 validate-input m-b-20">
    <div>
      Mr
      <input id="homme" value="Homme" type="radio" name="genre">
      <span class="focus-input100"></span>
    </div>
    <div>
      Mme
      <input id="femme" value="Femme" type="radio" name="genre">
      <span class="focus-input100"></span>
    </div>
  </div>
  <div class="flex-row">
    <div class="nom wrap-input100 rs1-wrap-input100 validate-input m-b-20 taille2" >
      <input id="nom" class="input100" type="text" name="nom" placeholder="Nom">
      <span class="focus-input100"></span>
    </div>
    <div class="prenom wrap-input100 rs1-wrap-input100 validate-input m-b-20 taille2">
      <input id="prenom" class="input100" type="text" name="prenom" placeholder="Prénom">
      <span class="focus-input100"></span>
    </div>
  </div>
  <div class="naissance wrap-input100 rs1-wrap-input100 validate-input m-b-20 taille2">
    <input id="naissance" class="input100-2" type="date" name="date" placeholder="Date de naissance"
      min="1930-01-01" max='<?php echo $date_max; ?>'>
    <span class="focus-input100"></span>
  </div>
</div>
<div class="login wrap-input100 rs1-wrap-input100 validate-input m-b-20 taille1">
  <input id="login" class="input100" type="text" name="login" placeholder="Login (5 caractères minimum)">
  <span class="focus-input100"></span>
</div>
<div class="password1 wrap-input100 rs1-wrap-input100 validate-input m-b-20 taille1" >
  <input id="password1" class="input100" type="password" name="password1" placeholder="Mot de passe">
  <span class="focus-input100"></span>
</div>
<div class="password2 wrap-input100 rs1-wrap-input100 validate-input m-b-20 taille1" >
  <input id="password2" class="input100" type="password" name="password2" placeholder="Confirmation mot de p">
  <span class="focus-input100"></span>
</div>
<div class="container-login100-form-btn">
  <input type="button" id="validate_register" class="responsive_button login100-form-btn" value="Valider">
</div>
<div class="text-center m-t-40">
  Vous avez un compte ?
  <a href="connexion.php" class="txt3">
  Se connecter
</a>

```

Une fois les données renseignées, je récupère

le contenu des champs en JavaScript afin des les passer en paramètre via Ajax pour pouvoir les insérer dans la base avec des requêtes SQL.

Tout d'abord, je regarde si le contenu des champs n'est pas vide, si c'est le cas, je stocke leur contenu dans des variables.

```

function inscription(){
$(".wrap-input100").css({"border": "solid 1px #E6E6E6"})
$(".genre").css({"color" : "black"})
if($("#nom").val() != "" && $("#prenom").val() != "" && $("#adresse").val() != "" &&
$("#email").val() != "" && $("#naissance").val() != "" && $("#login").val() != "" &&
$("#password1").val() != "" && $("#password2").val() != "" &&
$("#homme").is(':checked') || $("#femme").is(':checked') )
{
$(".champ_vide").remove()
genre = document.querySelector('input[name="genre"]:checked').value;
nom = $("#nom").val()
prenom = $("#prenom").val()
adresse = $("#adresse").val()
email = $("#email").val()
naissance = $("#naissance").val()
login = $("#login").val()
password1 = $("#password1").val()
password2 = $("#password2").val()

```

Une fois les vérifications faites, je passe en données le contenu des champs et les envoies à un autre fichier php qui va effectuer les requêtes d'insertion en base de données.

Une fois inscrit, j'envoie l'utilisateur sur la page de connexion.

Pour la connexion le principe est le même que pour l'inscription, nous avons un fichier php qui comprend un formulaire avec les champs "login/pseudo" et "mot de passe" "mot de passe oublié" et un dernier champ "inscrivez vous" au cas où l'utilisateur se retrouve sur la page de connexion sans avoir de compte.


```

<div class="login wrap-input100 rs1-wrap-input100 validate-input m-b-20 taille1">
  <input id="login" class="input100" type="text" name="login" placeholder="Login">
  <span class="focus-input100"></span>
</div>
<div class="password1 wrap-input100 rs1-wrap-input100 validate-input m-b-20 taille1" >
  <input id="password" class="input100" type="password" name="password" placeholder="Mot de passe">
  <span class="focus-input100"></span>
</div>
<div class="container-login100-form-btn">
  <input type="button" id="validate_connect" class="responsive_button login100-form-btn" value="valider">
</div>
<div class="text-center m-t-40 ">
  <p>Besoin d'un compte ? <a class="txt3" href="inscription.php">Inscrivez-vous</a></p>
  <p>Mot de passe oublié ? <a href="" class="txt3">Cliquez ici</a></p>

```

ensuite le Javascript intervient pour pouvoir permettre l'insertion dans les champs

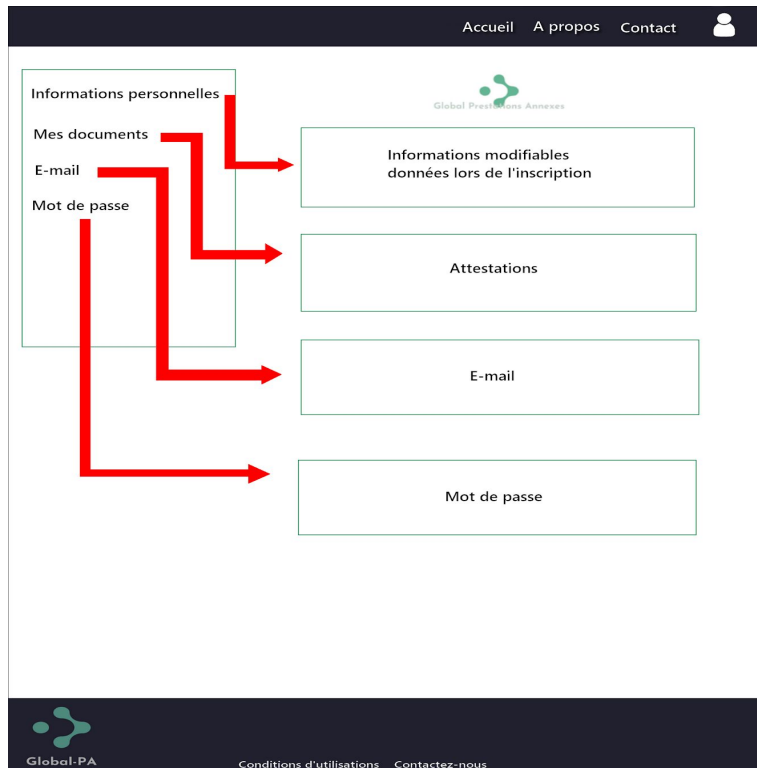
```

function connexion(){
  if ($("#login").val() != "" && $("#password").val() != "")
  {
    login = ($("#login").val())
    password = ($("#password").val())
    $.ajax({
      url: '../fonctions/fonction_connexion.php',
      type: 'POST',
      data: {login: login, password: password},
      success: function(data){
        data=data.trim()
        if(data == "erreur")
        {
          $(".champ_vide").remove()
          $(".login100-form-title").after('<div class="champ_vide">*Login ou mot de passe incorrect</div>')
        }
        else
        {
          window.history.go(-1);
        }
      }
    });
  }
}

```

Ensuite toujours dans la partie Profil utilisateur nous avons la possibilité de modifier les informations personnelles de l'utilisateur comme le mot de passe le login l'adresse mail l'adresse et autre.

Pour cela j'ai mis en place une interface assez simple qui va suivre la maquette ci dessous



j'ai donc créer un fichier PHP dans lequel j'ai créé les premiers formulaire et un autre fichier en Javascript qui va permettre l'affichage sans rafraichissement de la page mais aussi pour la modification des données de profil de l'utilisateur

```
<body id="test" class="accueil">
<main class="ie-stickyFooter">
  <div id="page">
    <div id="header_content">
      <?php include('../sources/header.php');?>
    </div>

    <section class="formulaireprofil">
      <ul class="infoperso">
        <li><button id="infoperso"> informations personnelles</a></li>
        <li><button id="infoperso2"> informations de confidentialités </a></li>
      </ul>

      <div class="forminfoperso">
      </div>
    </section>
  </div>
</main>
</body>
```

Nous pouvons voir ici deux dans la même classe ce qui va permettre l'affichage des "grands" formulaire à gauche ce qui laissera la place pour dérouler d'autres formulaire sur la partie de droite grâce au <div class="forminfoperso"> ce qui donnera ceci:

- informations personnelles
- informations de confidentialités

paolo
zizzo
paolo
26 / 08 / 2000
1rue de la gérome
ok

Je me suis servi du JavaScript pour l’affichage instantané comme expliqué ci-dessous mais aussi pour récupérer la valeur des champs préalablement rentrée lors de l’inscription (prénom, nom, login, date de naissance, adresse)

Pour ceci j’ai créé une fonction onclick qui va agir sur mon premier formulaire de mon fichier .php bien évidemment je renseigne l’endroit ou sont mes classes donc dans mon body, donc je traduis ce code par « dans mon body je clique sur mon formulaire « infoperso » ect... ».

```
<ul class="infoperso">
  <li><button id="infoperso"> informations personnelles</a></li>
```

```
$(document).ready(function(){
  $("body").on("click", "#infoperso", function (){
```

Une fois avoir fait ceci j’ai rédigé un formulaire comprenant tous les champs dont j’avais besoin en utilisant des id différent à chaque fois :

```
$(".formmodif1").remove()
$(".forminfoperso").append('<form class="formmodif1" method="post" action=""></form>')
```

Je nomme mon formulaire “formmodif1” qui se mettra ensuite dans la div que j’ai créée précédemment “forminfoperso”, on voit bien qu’après avoir nommé mon formulaire je fais appel

à un “.remove()”, ceci va permettre à la suppression de celui-ci (suppression visuelle) sinon à chaque click sur un “grand” formulaire les “petits” formulaire à droite s’afficheront à l’infini.

De plus la fonction append qui succède mon “forminfoperso” va permettre l’ajout de mon formulaire dans cette div.

Une fois avoir fait cela je peux rédiger mon formulaire :

```
$(".formmodif1").append('<input id="nom" class="input1" type="text"
required placeholder="Nom" value="'+result[nom]+' " name="Nom"> ');

$(".input1").after('<input id="prenom" class="input2"
type="text" required placeholder="Prénom" value="'+result[prenom]+' "
name="Prénom"> ');

$(".input2").after('<input id="login" class="input3"
type="text" required placeholder="login" value="'+result[login]+' "
name="login"> ');

$(".input3").after('<input id="naissance" class="input4"
type="date" required placeholder="Date de naissance"
value="'+result[naissance]+' " name="date"> ');

$(".input4").after('<input id="adresse" class="input5"
type="text" required placeholder="Adresse" value="'+result[adresse]+' "
name="adresse"> ');

$(".input5").after('<input class="input7" type="button"
name="valider" value="ok" id="valider_form1"> ');

}

})

});
```

Une fois mon formulaire finit je vais pouvoir faire la vérification des champs (vérifier que tout les champs sont bien remplis et ensuite renvoyer les données entrées vers la base de données.

je rappelle ma fonction “onclick”

```
$("#body").on("click","#valider_form1", function () {
```

cette fois ci ma fonction va agir sur la validation de mon formulaire d’ou l’id “valider_form1”.

```
$("#nom").css({"border": "1px solid #E6E6E6"})

$("#prenom").css({"border": "1px solid #E6E6E6"})

$("#login").css({"border": "1px solid #E6E6E6"})

$("#adresse").css({"border": "1px solid #E6E6E6"})

$("#naissance").css({"border": "1px solid #E6E6E6"})
```

Je renseigne ensuite mes champs et intègre mon css

```
if($("#nom").val() != "" && $("#prenom").val() != "" && $("#login").val() != "" && $("#naissance").val() != "" && $("#adresse").val() != "")
{
    nom = $("#nom").val()
    prenom = $("#prenom").val()
    login = $("#login").val()
    naissance = $("#naissance").val()
    adresse = $("#adresse").val()

    $.ajax({
        url: '../fonctions/fonction_modif.php',
        type: 'POST',
        data: {nom: nom, prenom: prenom, login: login, naissance:
naissance, adresse: adresse},
```

```
        success: function(data) {  
  
            console.log(data);  
  
        }  
  
    });
```

mon if va me servir à la vérification de mon formulaire, c'est à dire que si mes champs sont bien remplis, le code ci dessus s'effectuera, sinon j'afficherai les erreurs comme ceci:

```
else  
  
    {  
  
        if($("#nom").val() === "")  
  
        {  
  
            $('#nom').css({"border":"1px solid #C0392B"})  
  
            $('#nom').attr("placeholder","*Veuillez entrer un nom")  
  
            $('#nom').addClass("erreur_form")  
  
        }  
  
        if($("#prenom").val() === "")  
  
        {  
  
            $('#prenom').css({"border":"1px solid #C0392B"})  
  
            $('#prenom').attr("placeholder","*Veuillez entrer un  
prenom")  
  
            $('#prenom').addClass("erreur_form")  
  
        }  
  
        if($("#login").val() === "")  
  
        {  
  
            $('#login').css({"border":"1px solid #C0392B"})
```

```

        $('#login').attr("placeholder","*Veuillez entrer un
login")

        $('#login').addClass("erreur_form")
    }

    if($("#adresse").val() === "")
    {

        $('#adresse').css({"border":"1px solid #C0392B"})

        $('#adresse').attr("placeholder","*Veuillez entrer un
adresse")

        $('#adresse').addClass("erreur_form")
    }

    if($("#naissance").val() === "")
    {

        $('#naissance').css({"border":"1px solid #C0392B"})

        $('#naissance').attr("placeholder","*Veuillez entrer un
naissance")

        $('#naissance').addClass("erreur_form")
    }

    }

});
});

```

une fois ceci fais je vais devoir convertir l'ajax en json pour renvoyer les données sous forme de tableau comme ceci

```

$.ajax({

    url: '../fonctions/fonction_modif.php',

    type: 'POST',

```

```
success: function(data) {  
  
    for(i=0; i < JSON.parse(data).length; i++)  
    {  
        var result = JSON.parse(data)[i];  
        for(j=0; j < Object.keys(result).length; j++)  
        {  
            var id = Object.keys(result)[0]  
            var nom = Object.keys(result)[2]  
            var prenom = Object.keys(result)[3]  
            var adresse = Object.keys(result)[4]  
            var naissance = Object.keys(result)[6]  
            var login = Object.keys(result)[7]  
        }  
    }  
}
```

Nous avons maintenant réussi la modification des informations personnelles le système sera le même pour les autres formulaire.

4.3 Veilles et sécurité

Concernant la sécurité, il est tout d'abord important de vérifier les données rentrées par l'utilisateur dans les formulaires, que ce soit pour s'inscrire, se connecter ou déposer une annonce. Car ses données sont ensuite récupérées et insérées dans la base de données, il faut donc faire attention à ce qui est récupéré. De plus, ceci doit se faire sans perturber la navigation au sein du site.

C'est pourquoi j'utilise JavaScript afin de vérifier que tous les champs soient bien

remplies.

CONNEXION

*Des champs sont vides

VALIDER

Besoin d'un compte ? INSCRIVEZ-VOUS

Mot de passe oublié ? CLIQUEZ ICI

```
if ($("#login").val() != "" && $("#password").val() != "")
{
    login = $("#login").val()
    password = $("#password").val()
    $.ajax({
        url: 'fonctions/fonction_connexion.php',
        type: 'POST',
        data: {login: login, password: password},
        success: function(data){
            data=data.trim()
            if(data == "erreur")
            {
                $(".champ_vide").remove()
                $(".login100-form-title").after('<div class="champ_vide">*Login ou mot de passe incorrect</div>')
            }
            else
            {
                document.location.reload(true);
            }
        }
    });
}
else
{
    $('#login').css({"border":"1px solid #C0392B"})
    $('#password').css({"border":"1px solid #C0392B"})
    $(".champ_vide").remove()
    $(".login100-form-title").after('<div class="champ_vide">*Des champs sont vides</div>')
}
```

De plus, côté inscription, lorsqu'un utilisateur renseigne son login et son email, nous vérifions si ceux-là n'existent pas déjà dans ma base de données car ils doivent être unique à chaque utilisateur. Concernant l'email je regarde grâce à une fonction si celle-ci possède le bon format "...@...fr/.com". Sinon j'indique à l'utilisateur que le format n'est

pas le bon. De même en ce qui concerne le numéro de téléphone (les deux formats acceptés étant 06 et 07).

4.4 Difficultés rencontrées

Les délais

Tout d'abord, ce que j'ai trouvé assez difficile pour ce projet était d'établir des dates pour la réalisation de celui-ci car c'était la première fois que je réalisais un projet de cette ampleur avec un réel enjeu derrière.

Il était donc assez difficile pour moi de visualiser et de calculer le temps que j'avais pour réaliser chaque fonction dont le site avait besoin.

De plus, par le confinement, il m'a fallu trouver dans les premiers temps une autre façon de m'organiser et de trouver la motivation d'avancer dans le projet. Car en plus du projet professionnel, je devais également réaliser des projets à côté pour l'école car j'apprenais encore de nouveaux langages au début de celui-ci (JavaScript / jQuery / json).

Conclusion

J'étais vraiment motivé à l'idée de réaliser un projet que je trouvais en plus de ça vraiment très intéressant. Nous avons 2 propositions de projet et c'est celui-ci que nous avons choisis (le deuxième était un site e-commerce de ventes de sacs pour ordinateurs).

Cependant, je ne me rendais pas compte ce que cela impliquerait en terme de temps et de travail.

Finalement, ce projet m'a permis de mettre en pratique mes compétences que j'ai pu acquérir durant mon année de formation.

j'ai pu apprendre à travailler avec un certain rythme et une certaine assiduité. La réalisation de ce projet permet de savoir à quoi s'attendre en ce qui concerne le monde du travail car il fallait créer un site en fonction d'une demande d'un client, d'un temps donné (4 à 6 mois) et donc d'un cahier des charges, c'était donc très formateur.