

Mémoire d'entreprise

LICENCE PRO ACSID

Analyse et Conception des Systèmes d’Information Décisionnels

Promotion 2019

**Apprenti** : BAKARI Naïm

Maître d'apprentissage : ALI SAID Ridjali

Référents école : MOKRANE Abdenour, LEJOUX Valérie, LEFEVRE Frédéric

Table des matières

[I. Présentation du candidat 1](#_Toc11146948)

[II. Remerciement 2](#_Toc11146949)

[III. Introduction 3](#_Toc11146950)

[IV. Présentation de l’entreprise 3](#_Toc11146951)

[a. Présentation de EXPRESSUR – VISION 3](#_Toc11146952)

[b. Structure organisationnelle de l’entreprise 3](#_Toc11146953)

[c. Identification de l’entreprise 4](#_Toc11146954)

[d. Activité de l’entreprise 4](#_Toc11146955)

[V. Activité 5](#_Toc11146956)

[a. Environnement de travail 5](#_Toc11146957)

[b. Présentation de la mission 5](#_Toc11146958)

[1. Les outils mis à ma disposition et l’existant 5](#_Toc11146959)

[2. Analyse du projet/du cahier des charges 6](#_Toc11146960)

[c. Plan d’action 7](#_Toc11146961)

[d. Phase de conception 9](#_Toc11146962)

[e. Versionnage de l’application 10](#_Toc11146963)

[f. Phase de développement 10](#_Toc11146964)

[1. Les requête SQL 15](#_Toc11146965)

[2. Organisations des pages 16](#_Toc11146966)

[3. La page d’accueil 19](#_Toc11146967)

[4. Liste & recherche d’offres 20](#_Toc11146968)

[5. Les offres où le candidat a postulé 20](#_Toc11146969)

[6. Ajouter un utilisateur (entreprise ou interne) 21](#_Toc11146970)

[7. Ajouter une offre 22](#_Toc11146971)

[8. Les différentes listes 23](#_Toc11146972)

[9. Entreprise qui envoie une offre. 24](#_Toc11146973)

[VI. Test de fonctionnement 26](#_Toc11146974)

[1. Test de l’inscription/connexion d’un utilisateur 27](#_Toc11146975)

[2. Test d’ajout d’une offre 28](#_Toc11146976)

[VII. Conclusion 29](#_Toc11146977)

## Présentation du candidat

Je m’appelle BAKARI Naïm, j’ai 21 ans.

Ayant eu un bac sciences et technologies de l'industrie et du développement durable (STI2D) à 18 ans en 2015, j’ai voulu continuer dans le domaine de la technologie et plus précisément dans l’informatique. J’ai donc poursuivi mes études en BTS Systèmes Numériques option Informatique et Réseaux (SNIR). Après l’obtention de mon BTS SNIR à 20 ans, j’ai voulu poursuivre mes études pour à terme pouvoir faire de l’informatique mon métier dans le domaine du web. J’ai donc décidé de me lancer dans une licence pro Analyste Concepteur en Système d’Information et de Décision car je suis intéressé par les bases de données (celle-ci étant très importante lors de la conception d’un site web).

En BTS j’ai travaillé en stage dans une petite association qui se nomme Monde en Marge Monde en Marche. Ma Mission consistait à créer un site web interne qui permettrait de regrouper toutes les données sur les locataires et les logements de l’association. Une partie du site avait déjà été commencée mais par manque de temps les membres de l’association n’ont pas pu finir le site. C’est pour cela que l’association m’a recruté. C’est à ce moment-là que j’ai su qu’il fallait que je fasse du web mon métier.



## Remerciement

Avant de commencer ce mémoire, je tiens à remercier mon maître d’apprentissage monsieur ALI SAID Ridjali, qui est aussi le chef d’entreprise, pour avoir accepté de m’accueillir durant une année dans son entreprise.

J’aimerais aussi remercier le personnel CFI/CNAM ainsi que tout le corps enseignant pour m’avoir accueilli afin de poursuivre mes études en Licence Professionnelle.

Je tiens particulièrement à remercier madame Valérie LEJOUX et monsieur Frédéric LEFEVRE de m’avoir aidé et conseillé pour rédiger mon mémoire d’entreprise.

## Introduction

La formation qui m’a plongé au cœur de la vie d’entreprise durant cette année 2018/2019 m’a permis de m’initier au métier de concepteur/développeur web. Le développeur full stack est polyvalent car il peut s’occuper de toutes les tâches que le codage d’un site comporte. Pour ma part j’ai donc fait usage des code html, PHP, css, le JavaScript ainsi que le SQL pour la base de données. J’ai été chargé de réaliser un site web qui interagissait avec une base de données afin de stocker des informations confidentielles.

L’élaboration de ce mémoire a pour but de rendre compte de manière fidèle et analytique de l’année passée au sein de l’entreprise. Il apparaît donc logique de présenter à titre préalable l’entreprise et son secteur d’activité, d’un point de vue structurel et fonctionnel. Ensuite je présenterai les différentes tâches que j’ai réalisé durant cet apprentissage ainsi que le savoir-faire acquis lors de cette expérience.

Enfin je conclurai ce mémoire par un bilan global, les apports humains et techniques de ce mémoire, afin de souligner les compétences que j’ai pu développer et de faire le lien entre cette expérience et mon projet professionnel.

## Présentation de l’entreprise

### Présentation de EXPRESSUR – VISION

EXPRESSUR – VISION est une PME, sous la forme juridique SAS, créée le 25 juillet 2016 à Noisiel dans le 77. Le président de cette entreprise s’appelle ALI SAID Ridjali. C’est un Ingénieur en réseaux et Télécommunications. Le dirigeant de l’entreprise se nomme Monsieur ALI Moumini et le conseiller spécial se nomme ALI MOUSSA Jacques.

### Structure organisationnelle de l’entreprise

BAKARI Naïm

Dans la direction générale, nous retrouvons le dirigeant de l’entreprise Monsieur ALI Moumini, le conseiller spécial ALI MOUSSA Jacques ainsi que le président de l’entreprise Monsieur ALI SAID Ridjali. L’entreprise a décidé d’externaliser la comptabilité c’est à dire qu’une autre entreprise gère les comptes d’Expressur-Vision. Dans le pôle informatique il y a donc mon tuteur, qui est aussi le président de l’entreprise et moi. Enfin, dans le pôle formation il y a des personnes qui donnent des formations en bureautique (Word, Excel par exemple). Celles-ci visent énormément de personnes (les demandeurs d’emplois, les étudiants, les particuliers, les artisans, les salariés).

### Identification de l’entreprise

Nom : EXPRESSUR – VISION

Date de création : 25 juillet 2016

Adresse : 2 Allée Montesquieu, 77186 Noisiel

Forme juridique : Société par actions simplifiée (SAS)

Numéro SIREN : 821 717 089

Numéro SIRET : 821 717 089 00014

Code NAF : Conseil en systèmes et logiciels informatiques (6202A)

Nombre de salarié : 3 salariés

Capital social : 4K€

CA  2017-2018 : 123 000 €

(Source : societe.com)

### Activité de l’entreprise

Cette entreprise exerce une activité de prestation de services informatiques, de conseils auprès des entreprises en matière de recrutement spécialisé dans les métiers des systèmes de l’information. Expressur-Vision a aussi un pôle de formation pour pouvoir former des personnes à la bureautique. Expressur-Vision travaille avec différents partenaires comme le CNFPI, Montréal Associates, Progressive Recruitment, Espace Freelance et Hays plc.



Pour que la prestation soit possible, il était impératif de mettre un place une plateforme où les entreprises partenaires, les candidats, pour pouvoir postuler pour des offres, ainsi que Expressur-vison, puissent communiquer entre eux. C’est alors qu’est venue l’idée d’un site web et c’est pour cela que j’ai été recruté.

## Activité

### Environnement de travail

Lorsque je suis en entreprise, je partage mon bureau avec d’autres employés. Je suis sur un ordinateur de bureau qui fonctionne sous linux qui m’a été assigné durant toute la durée de ma mission au sein de l’entreprise.

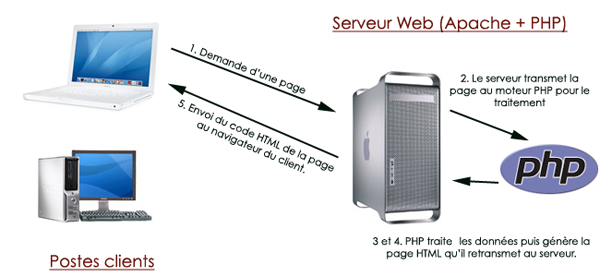
### Présentation de la mission

L’entreprise m’a recruté dans le but de créer une application web car ce besoin devenait très urgent étant donné qu’il n’y avait aucun site mis à disposition qui permettait de mettre en ligne les offres à pourvoir de nos partenaires. En effet, certaines entreprises partenaires ont déjà des postes non pourvus.

Avant de décrire ma mission il est nécessaire de savoir quels outils sont à ma disposition ainsi que de décrire ce qui existait déjà.

#### Les outils mis à ma disposition et l’existant

A ma disposition j’ai un IDE (Integrated Development Environment, soit un environnement de développement en français) qui se nomme Netbeans. Cet IDE me permet de coder de manière plus simple et précise le HTML, le CSS, le JavaScript ainsi que le PHP. Ce dernier permet entre autres de lier le site web à la base de données pour rendre le site dynamique. Pour tester l’application web en local j’utilise LAMP (Linux Apache MySql PHP). Ce logiciel est composé de quatre éléments : il fonctionne sous le système d’exploitation Linux, Apache est le serveur web, MySql est la base de données et enfin PHP le langage utilisé pour pouvoir communiquer entre l’application web et la base de données. Sans ce logiciel nous ne pouvons pas tester le site en tant que site web dynamique car le navigateur web ne comprend pas le PHP. Le PHP est traduit par le serveur puis ensuite il transmet l’information au navigateur qui peut maintenant l’interpréter.



**Figure 1 Schéma global d'une requête web**

#### Analyse du projet/du cahier des charges

##### Contexte

L’une des activités d’Expressur est de trouver le meilleur candidat pour une offre d’emplois. Pour cela nous avons besoin d’un outil telle qu’une application qui pourra être utilisée en interne par l’équipe RH de l’entreprise, mais qui devra être aussi accessible de l’extérieur à travers le site de l’entreprise.

##### 2.2 Le projet

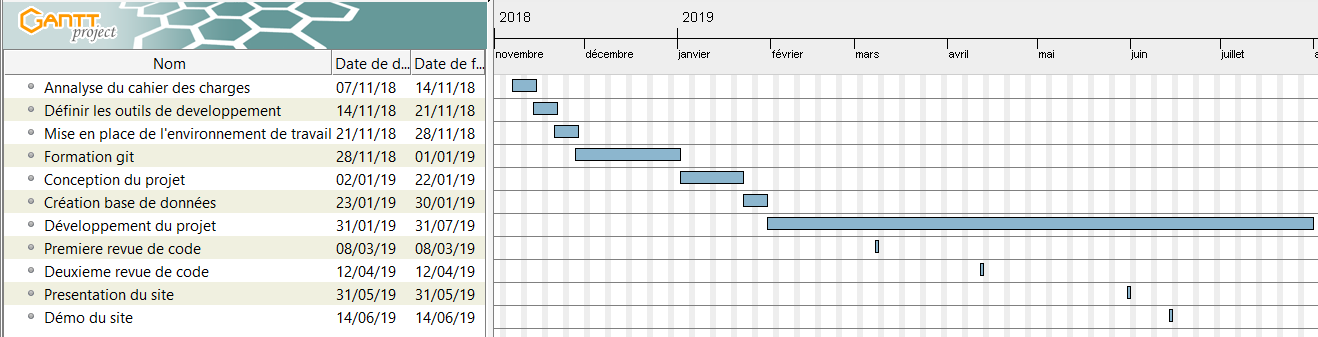
###### 2.2.1 Objectif

Une partie du site devra être accessible par le public, tandis qu’une autre partie sera uniquement accessible pour les utilisateurs internes (l’équipe RH de l’entreprise Exressur ou bien l’administrateur). La partie accessible par le public concernera la consultation des offres d’emplois. Une base de données sera nécessaire pour stocker les offres d’emplois, les candidats, les entreprises qui proposent des offres ainsi que les utilisateurs internes. Sur le site web, le grand public pourra consulter les offres d’emplois, il y aura la possibilité de filtrer les annonces selon différents critères (catégories, villes) ainsi que la possibilité de faire une recherche par mot clé. Les candidats pourront postuler pour une ou plusieurs offres et pourront aussi déposer leur CV au format PDF et Word (.doc et .docx). Les candidats pourront aussi consulter les offres auxquelles ils auront postulé et pourront aussi se désengager de l’offre. Les entreprises pourront ajouter, supprimer, modifier les offres ainsi que voir quel candidat a postulé pour quelle offre. Les utilisateurs internes devront s’occuper de la gestion du site, de la supervision des offres ainsi que celle des candidats. En revanche, les entreprises devront remplir un formulaire de demande d’inscription directement sur le site web. Ensuite, après que les RH auront pris contact avec les entreprises, un utilisateur interne ou l’administrateur pourra créer un utilisateur entreprise.

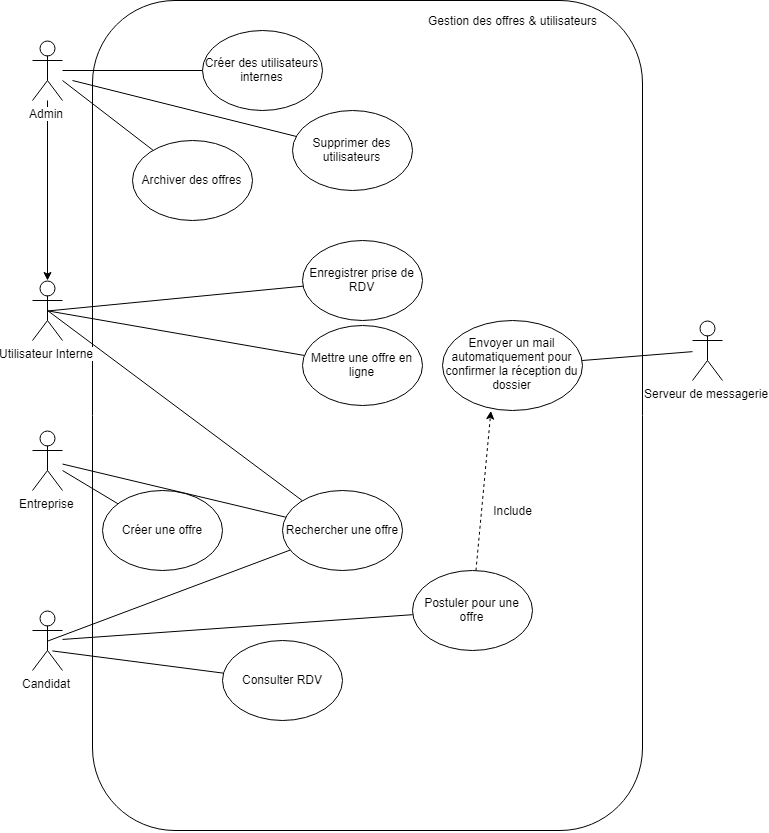
###### 2.2.2 Contrainte

Pour pouvoir créer ce site web, il y aura certaines contraintes auxquelles je devrai m’adapter. Pour commencer, le site web devra être accessible de l’extérieur. Pour que le site web soit sécurisé, chaque utilisateur devra s’authentifier via un mail et un mot de passe. Les candidats devront créer eux même leur compte tandis qu’uniquement l’administrateur pourra créer des utilisateurs internes.

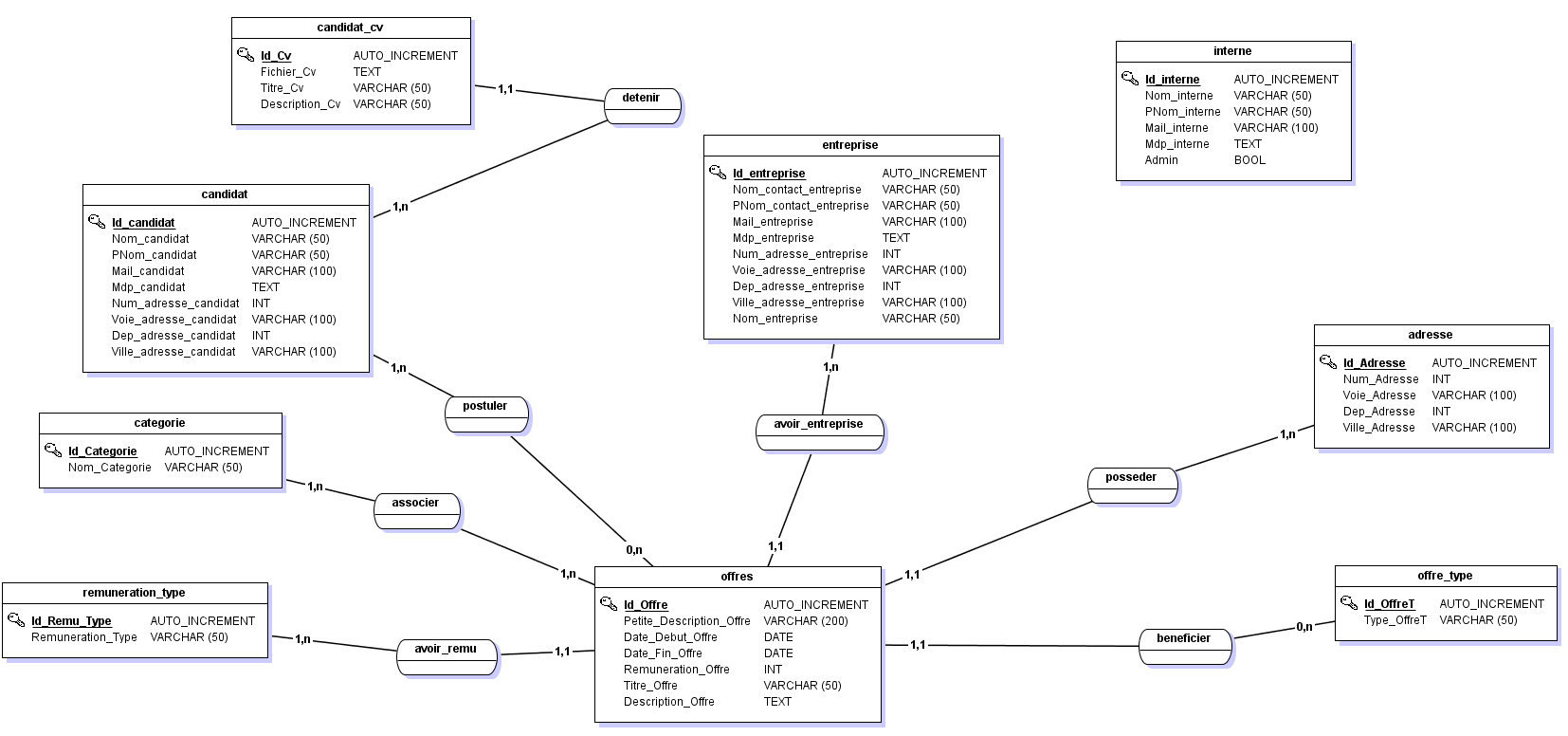
### Plan d’action

Avant tout, j’ai commencé par planifier mon travail.

J’ai commencé mon alternance le 7 novembre et elle se terminera le 31 juillet. J’ai réellement commencé à coder à partir de février, avant cette date, je me suis consacré à la conception et à ma formation notamment sur git. Pour pouvoir illustrer tous les cas réalisés par l’application web, j’ai décider de faire un diagramme de cas d’utilisation (Use case) afin de facilité la compréhension du cahier des charges :

 **Figure 3 Use case**

### Phase de conception

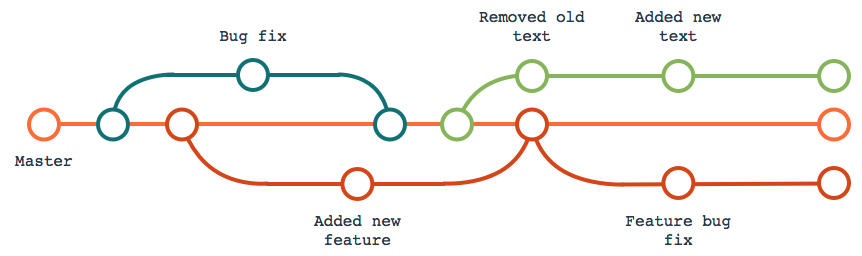
J’ai réalisé une multitude de modèle conceptuel de données (MCD) afin de pouvoir réaliser la base de données qui me permettra de stocker toutes les informations concernant l’application web. Voici le MCD utilisé actuellement :

Pour que la base de données soit optimal, j’ai décidé de créer trois tables différentes pour bien différencier les acteurs qui interagissent avec le système (Administrateur et Interne ensemble, Candidat et Entreprise). Les entreprises possèdent une ou plusieurs offres. Les candidats peuvent postuler à une ou plusieurs offres. Ces derniers on un ou plieurs CV. Les offres ont une seule adresse tandis qu’une adresse peut posséder plusieurs offres. Une offre bénéficie d’un type (CDD, CDI, etc) ainsi qu’un type de rémunération (par jour, par mois, etc). Une offre peut avoir une ou plusieurs catégories.

### Versionnage de l’application

Une fois que j’ai fini de concevoir la première version de la base de données, j’ai commencé à me former à GIT.

Git est un logiciel qui permet de gérer les versions du code source d’un projet. Par exemple lorsque j’écris du code et que je le sauvegarde. Si je partage le code sur un serveur distant (serveur principal contenant le code du projet, vers GIT hub) et modifie le code, GIT (dans mon cas via GIT hub) pourra me montrer les lignes différentes entre les deux versions de mon code. Nous avons utilisé GIT hub car nous sommes deux sur le projet : Mon tuteur et moi.

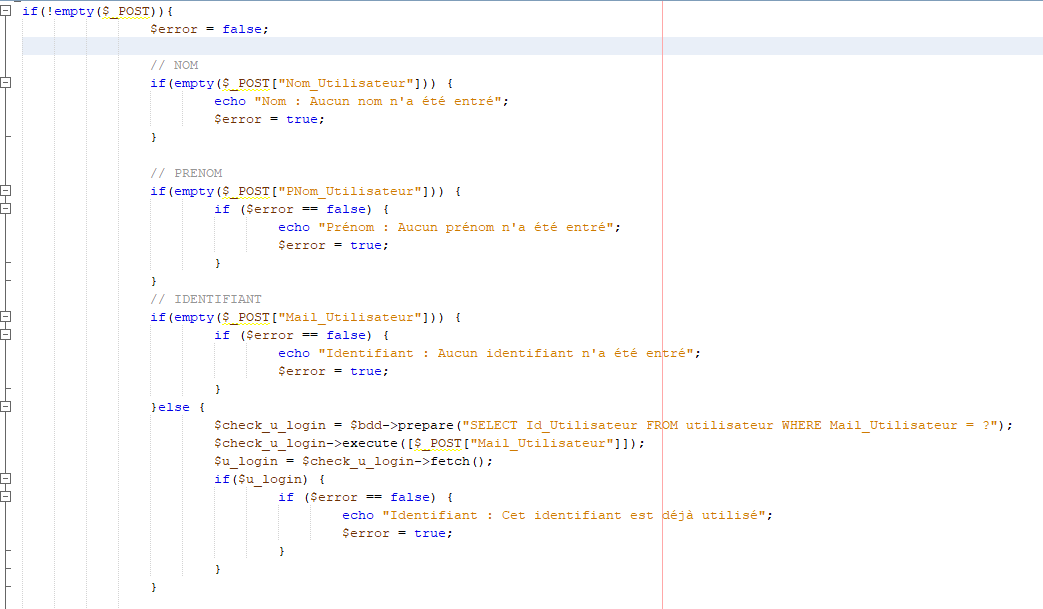


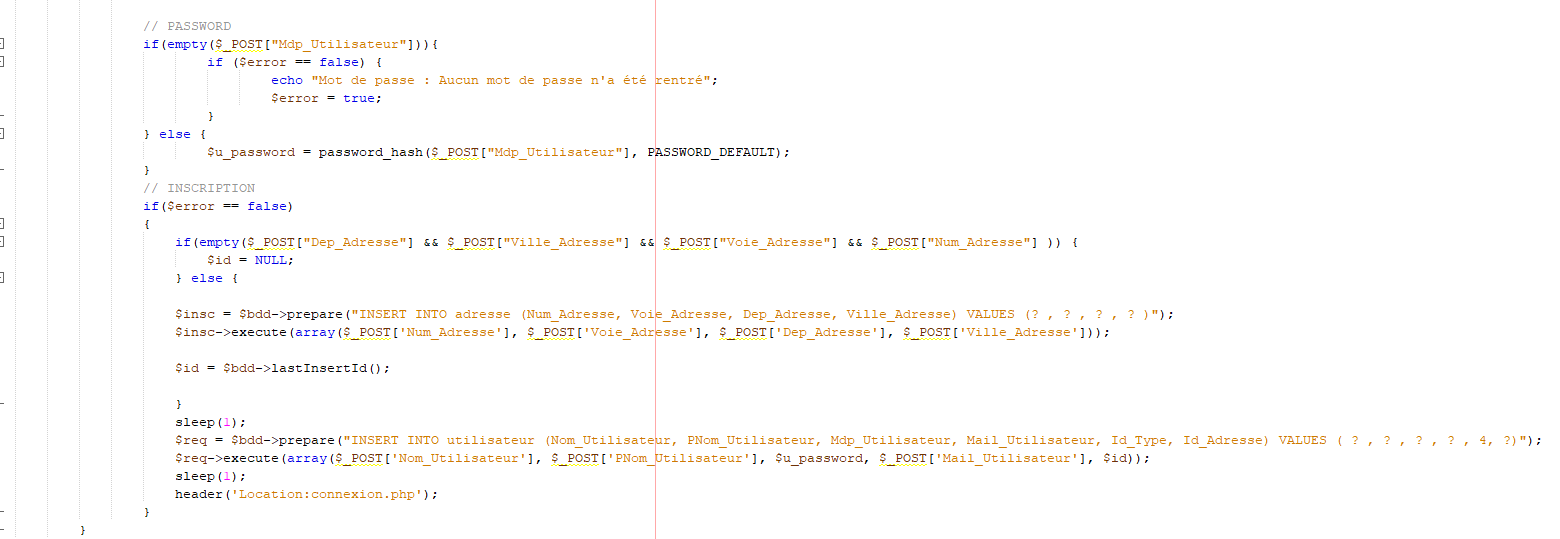
**Figure 5 Schéma git**

Au fur et à mesure de ma formation à GIT, je me suis rendue compte que certain IDE permet de gérer GIT sans à avoir à entrer des lignes de commande, tout est visuel comme l’IDE Netbeans (l’une des raisons de mon choix vers cet IDE). J’ai choisi cette méthode, à chaque fin de journée je pousse mon travail sur GIT hub.

### Phase de développement

Une fois ma formation à git terminée, mon environnement de travail installé et fonctionnel, ma première version de ma base créée, j’ai enfin commencé à coder. J’ai tout d’abord commencé à chercher diffèrent Template Bootstrap sur internet puis je l’ai fait valider par mon tuteur. Bootstrap est un ensemble d’outils pour faciliter la création du design d’un site. Bootstrap est une bibliothèque qui contient des codes HTML, CSS ainsi que des formulaires, des boutons, des extensions JavaScript. Ensuite j’ai commencé à réaliser la base de données J’ai commencé par supprimer tous les éléments inutiles du Template, puis j’ai commencé la mise en forme. Une fois fais, j’ai enfin commencé à utiliser le PHP pour pouvoir faire une page de connexion et d’inscription, j’ai amélioré cette dernière au fil du temps car au début je ne savais pas comment ajouter des informations d’un formulaire dans plusieurs table (ici table utilisateur et adresse). L’inscription doit être extrêmement sécuriser avec un mot de passe qui doit être hacher lors de l’insertion dans la base de données. Voici un aperçu du code php pour l’inscription :



Avant toutes choses, je vérifie si le futur utilisateur a complété tous les champs obligatoires (soit via PHP comme ci-dessus soit via html), ensuite je vérifie si le mail est déjà utilisé au pas. S’il n’y a aucun problème alors nous pouvons passer à la seconde partie : 

Dans le PHP je vérifie si le mot de passé est saisie. Si c’est le cas je récupère le mot de passe saisie par l’utilisateur, puis grace à la ligne de code suivante :

*$u\_password = password\_hash($mot\_de\_passe, PASSWORD\_DEFAULT) ;*

Le mot de passe est de la manière la plus sure actuellement. A savoir, le *PASSWORD\_DEFAULT* permet de hashé le mot de passe de la même manière que *PASSWORD\_BCRYPT*. La seul différnce c’est que *PASSWORD\_DEFAULT* est prévue pour le futur, c’est-à-dire que lorsque une méthode de hashage plus puissante que l’actuel sortiras, elle seras automatiquement mis à jour à la mise à jour du PHP.

Après que le mot de passe soit hashé, il faut ensuite tout mettre dans la base. Si’il n’y a pas d’erreur, alors on verifie si il y a une adresse. S’il n’y a pas d’adresse alors on entre dans une variable *$id* la valeur *NULL*, si non on entre l’adresse dans la base dans la bonne table et ensuite on prend l’id de l’adresse grace à *lastInsertId()* pour stocker la valeur dans la variable *$id*.

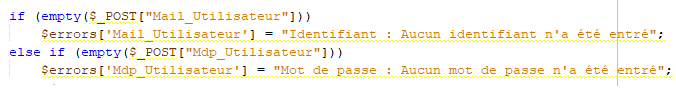
Ensuite nous entrons les valeurs de l’utilisateur dans sa table et en insérant à la fin le contenue de la variable *$id*. Les valeurs que l’utilisateur a entré dans le formulaire sont récuperés grace à la methode POST. Il éxiste deux sortes de methodes (GET et POST). Ici j’ai préféré utiliser la methode POST car celle-ci cache les informations qui transitent par rapport à la methode GET qui elle fait transiter les informations via l’URL. De plus, cette dernière permet de faire transiter moins d’informations car elle est limitée à 255 caractères. Voici une partie du formulaire d’inscription au site :

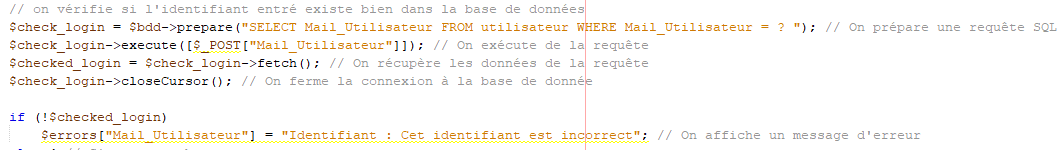


Tout d’abord en html les formulaires sont définis par les balises *<form> </form>*, à l’intérieur de cette balise nous mettons la méthode choisie (ici POST). Les balises *<input>* permettent à l’utilisateur de saisir toutes sortes de données selon le type de cette balise que l’on place à l’intérieur de la balise par exemple une date (*type= "date"*), du texte(*type="text"*), on y ajoute un *required=""* pour que lorsque l’utilisateur oublie de saisir l’information, le site le fais savoir à l’utilisateur sans supprimer les autres informations qu’il a saisi. A la fin d’un formulaire, on utilise un bouton de type submit pour que les données saisies par l’utilisateur soient récupérées grâce à la méthode utilisée (ici POST), puis les requêtes écris en PHP sont exécutés par le serveur. Les informations sont ensuite stockées dans la base de données.



Apres l’inscription, j’ai mis en place la connexion au site. Même principe que pour l’inscription, nous avons un formulaire avec les balises *<form> </form>* ainsi que les balises *<input>*. Coté PHP ça commence de la même manière : on vérifie tout d’abord si les inputs sont saisis.



Si l’identifiant (ici le mail) et le mot de passe sont entrés alors on va récupérer toutes les informations disponibles sur l’utilisateur par rapport à son mail. S’il n’y a rien cela veut dire que le mail n’est pas correct, alors, on affiche un message d’erreur.

Si l’identifiant existe, dans ce cas, on doit vérifier le mot de passe. Pour pouvoir vérifier si le mot de passe est correct, on fait une requête pour récupérer le mot de passe de l’identifiant que l’utilisateur a entré. Ensuite il suffit de la comparer avec le mot de passe que l’utilisateur à entrer grâce à la fonction *password\_verify()*. Si le mot de passe n’est pas identique alors on met un message d’erreur, sinon on entre toutes les informations que l’on a besoin sur l’utilisateur dans des variable superglobal *$\_SESSION*.

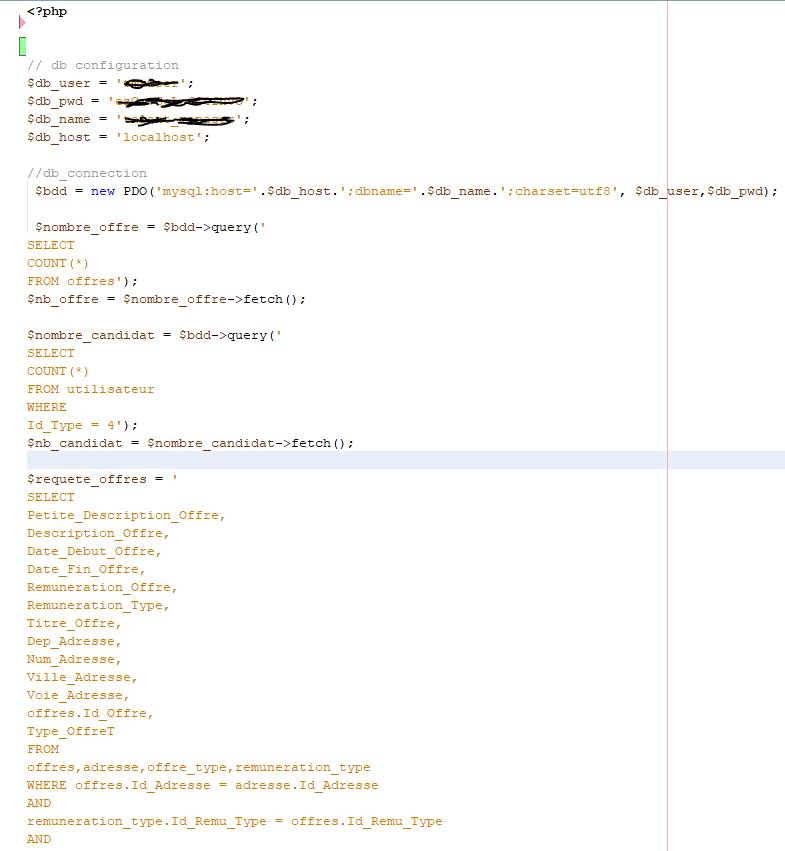
Une fois que l’utilisateur est authentifié, le PHP redirige l’utilisateur vers une page que ce dernier n’a pas le temps de voir pour qu’il soit redirigé vers le bon espace. Un candidat sera redirigé vers la page d’accueil tandis que l’administrateur, les utilisateurs internes ou bien les entreprises seront redirigées vers un autre espace de gestion. Cette page contient juste plusieurs conditions.

Dans le code Le PHP je vérifie le type de l’utilisateur (Administrateur, candidat, interne ou entreprise) puis il redirige l’utilisateur vers le bon espace. 

Lors de la conception du site, dans un premier temps, j’ai écrit certaines informations directement dans la base de données car elles ne changent pas, comme par exemple, le type des offres (il y aura toujours les mêmes types d’offres sauf si on doit en ajouter un mais on peut le faire directement via PHP MyAdmin). J’ai aussi rempli d’autres champs pour pouvoir tester le site avant que j’aie pu mettre en place un espace qui permet de remplir ses champs depuis le site par exemple l’insertion des offres d’emplois.

#### Les requête SQL

Lors de la conception du site, j’ai préféré mettre toutes les requêtes SQL dans le même fichier afin d’éviter de trop encombrer les pages du site. Ce fichier est donc appelé dans les autres pages pour que ces derniers puissent avoir accès aux requêtes.



Voici le fichier où il y a toutes les requêtes du site. Les quatre premières lignes servent à mettre dans des variables les différentes informations pour que le site puisse se connecter à la base de données. La ligne qui suit est justement la connexion à notre base. La première information est l’hôte (ici localhost), puis il y a le nom de la base de données, puis le codage des caractères, puis le nom d’utilisateur et enfin le mot de passe. Ensuite il y a toutes les requêtes pour le bon fonctionnement du site.

Je vais maintenant expliquer le développement du site page par page.

#### Organisations des pages

Avant de décrire les pages il faut comprendre comment j’ai codé ces pages. Sachant que le site comporte plusieurs pages et celles-ci doivent avoir la même charte graphique, par ailleurs, j’ai décidé de découper les pages en plusieurs partie. La première partie est l’entête de la page avec la barre de navigation. Cette page inclus la page où il y a toutes les requêtes du site.



Ensuite il y a la page en elle-même (la page d’accueil par exemple). Mes pages contiennent différentes "section" représentés entre les balises *<section> </section>*. Ces balises servent à regrouper un même sujet ou une même fonctionnalité. Elle est apparue avec la version 5 de HTML. À l’intérieur de la section où il y a les offres (je détaillerais ci-dessous), à la fin de la section, j’inclus une page regroupant les catégories ainsi que les villes des offres.



Voici ce que contient cette page.



Cette page contient donc deux tableaux, l’un pour les catégories et l’autre pour les lieux. Pour pouvoir créer un tableau dynamique il suffit d’ouvrir le PHP (*<?php ?>*) et de faire une boucle. Ici j’ai choisi la boucle WHILE, c’est la plus adapté, puis nous prenons les résultats de la requête que nous avons exécuter avant et nous les mettons dans un tableau PHP grâce au *fetch()*. Puis on affiche les résultats de la manière suivante :

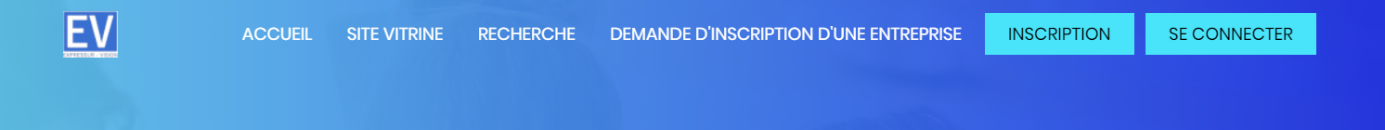
$nom\_de\_la\_variable [nom\_du\_champs\_voulue]. Grâce à la boucle nous avons un tableau HTML avec toutes les valeurs du tableau PHP. Voici à quoi ressemble les tableaux sur le site web :

Enfin, j’inclus le bas de page avec l’email de l’entreprise, le numéro ainsi que d’autres informations.



Voici une page type du site que j’ai réalisé lorsqu’un candidat n’est pas connecté :

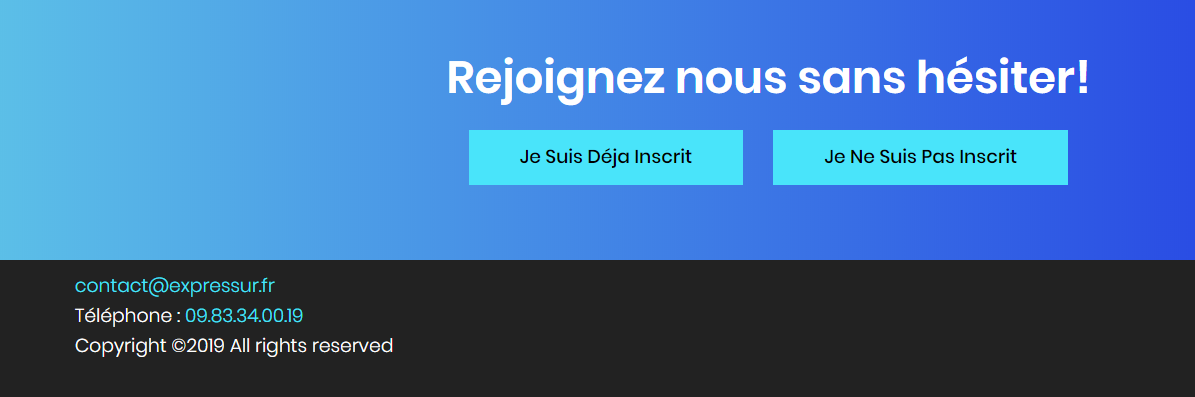
L’entête du site :



La page en elle-même :

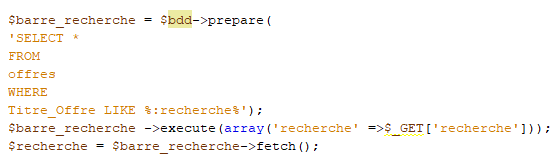


Le pied de page :



Lorsqu’un candidat se connecte il y a le haut de la page qui change, la page en elle-même comporte les informations liées au candidat et le pied de page ne comporte plus la mention "Rejoignez-nous sans hésiter ! ".

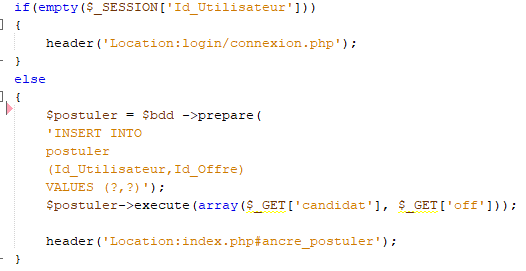
#### La page d’accueil

La page d’accueil permet d’accéder à toutes les fonctionnalités proposées dans le diagramme de cas d’utilisation vu plus tôt. La page d’accueil commence par une barre de recherche afin de rechercher les offres par mot clé. Pour que la recherche fonctionne, c’est un formulaire html ainsi qu’une requête SQL de type.

En PHP on peut exécuter une requête de plusieurs façon (*querry()*, *prepare()*). Ici j’utilise un *prepare()* car il y a un élément dans la requête qui n’est pas encore connu. Cet élément sera remplacé par le mot clé qu’aura entré un utilisateur dans la barre de recherche. Ensuite, il y a la liste des offres. Avant tout je vérifie si la personne est connectée garce à une des variables superglobal de session. Cette vérification me permet de faire en sorte que lorsqu’un candidat est connecté, il y a des variables ainsi que des vérifications pour savoir si le candidat à postuler pour une offre ou pas, et si la personne qui visite le site n’est pas connecté alors le site va afficher les offres normalement. Voici ce qui permet de faire la vérification pour voir si un utilisateur est connecter ou pas :



Empty en PHP détermine si une variable est vide. Lorsque qu’un candidat veut postuler pour une offre il lui suffit de cliquer sur postuler. Lorsque le candidat a cliqué, le PHP passe en paramètre dans l’URL grâce à la méthode GET l’ID de l’offre ainsi que l’ID du candidat de la manière suivante :

Ensuite, depuis le code le PHP redirige le candidat vers une page qui va nous permettre d’entrer les paramètres dans la base de données. Si personne n’est connecté alors le PHP redirige la personne vers la page de connexion, sinon le PHP remplis la base avec les bons paramètres, le candidat est redirigé à l’endroit où il était avec une mention qui dis que le candidat a déjà postulé pour cette offre.

Sur la page d’accueil, j’ai décidé, avec l’approbation de mon tuteur, que la liste d’offre devait être limité à un certain nombre d’offres (10) car sinon la page d’accueil serait vraiment trop surchargée. Ce nombre d’offres pourra être modifier lorsque l’administrateur du site le voudra. Alors, j’ai créé une autre page consacrée à la liste des offres avec toutes les offres misent sur le site.

#### Liste & recherche d’offres

Dans cette page, c’est le même principe que sur la page d’accueil c’est-à-dire qu’il y a une liste d’offre sauf qu’ici elle n’a pas de limite. Lorsqu’on clique sur un lieu ou une catégorie (peut importer où on est dans le site), il y a le nom de la ville ou de la catégorie qui passe en paramètre dans l’URL du site.

#### Les offres où le candidat a postulé

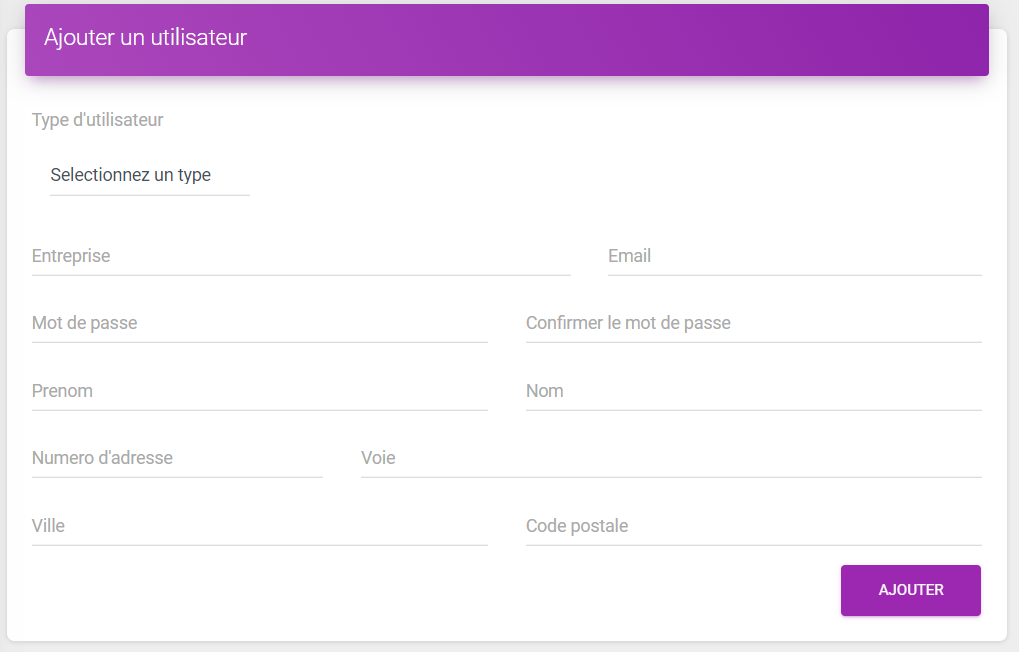
Lorsqu’un candidat postule, il doit avoir accès aux offres auquel il a postulé, c’est pour cela que j’ai décidé de créer une page dédier à cet effet. Cette page permet au candidat de voir les offres auquel il a postulé ainsi que le détail de l’offre mais surtout de se désengager de l’offre (la supprimer de l’espace mais pas du site). Pour cela il suffit de faire une requête SQL avec tous les champs que l’on a besoin pour l’affichage de l’offre ainsi qu’une condition WHERE qui indique l’ID de l’utilisateur connecter :

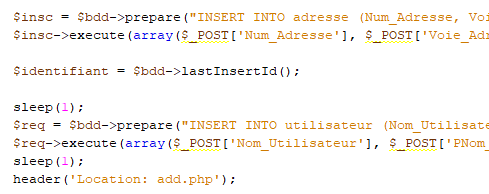
Il n’y a pas écrit WHERE mais AND car il y a plusieurs WHERE dans ma requête.

Maintenant je vais parler de la partie coté administration du site.

#### Ajouter un utilisateur (entreprise ou interne)

Pour pouvoir créer une offre, j’ai tout d’abord besoin de créer un utilisateur de type entreprise car une offre est forcément reliée à celle-ci. Pour que se soit assez simple, lorsque je créer un utilisateur (interne ou entreprise), la personne entre le nom, le prénom (du contact de l’entreprise ou bien le nom de l’utilisateur interne), l’adresse, éventuellement le nom de l’entreprise ainsi que son type d’utilisateur. J’ai décider que la personne qui inscrira un utilisateur rentreras un mot de passe lui-même. Une fois fais l’application enverra un mail en essayant d’inciter la personne à changer de mot de passe.

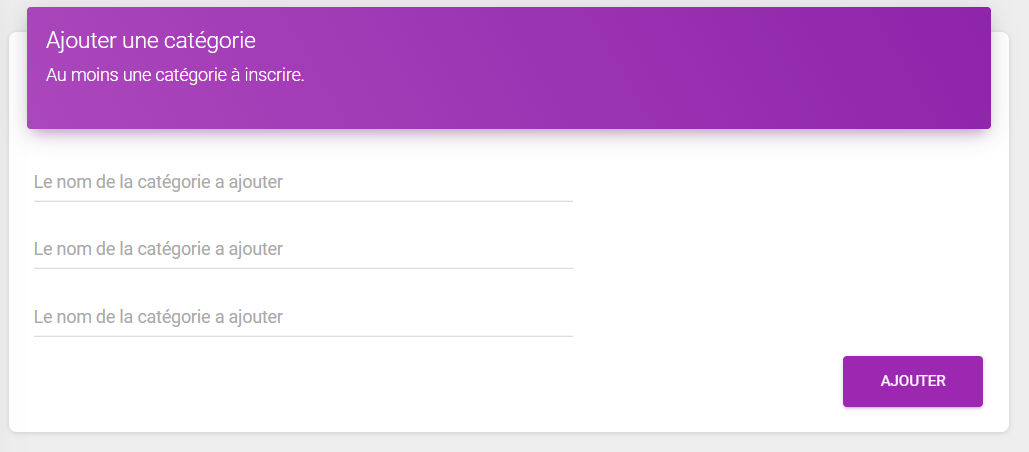


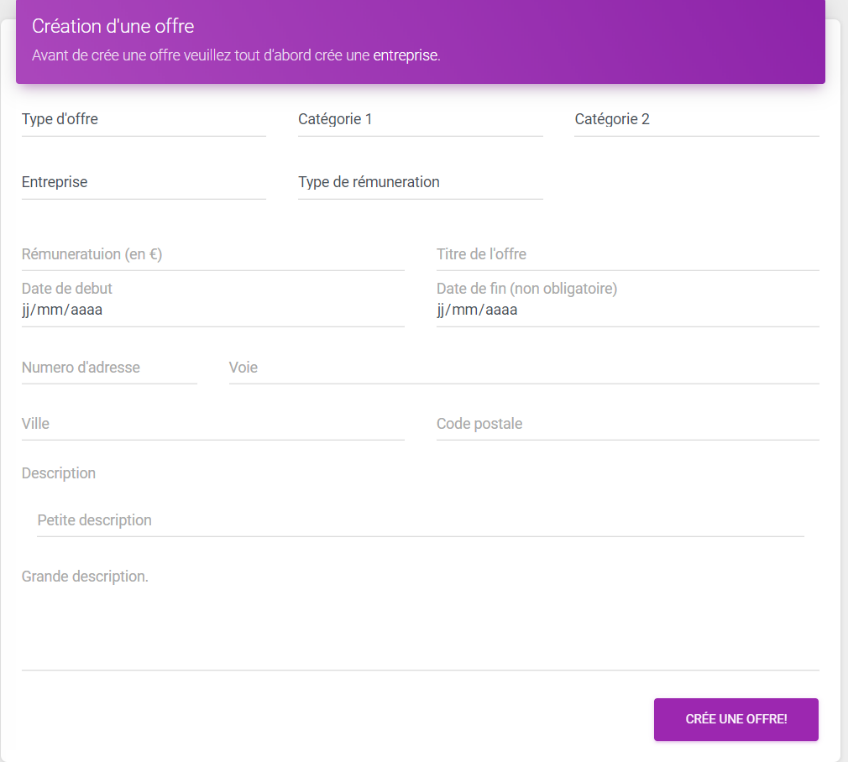
Dans ce formulaire on peut remarquer qu’il y a plusieurs tables à remplir, bien évidemment, il est impossible de faire une requête SQL d’insertion avec plusieurs tables à l’intérieur, alors, on utilise le PHP. Grâce à *lastInsertId(),* on peut prendre le dernier ID que la base de données à enregistrer.

Ici je prépare une requête (pas de query car on ne sait pas à l’avance ce que l’on va insérer dans la table) puis on l’exécute en remplaçant les " ?" par les champs qu’a remplis l’utilisateur grâce aux POST. On récupère ensuite l’ID de la requête d’avant et on recommence pour la deuxième requête. On termine par exécuter en remplaçant les ? par les champs sauf pour l’ID ou dans ce cas on met $identifiant.

#### Ajouter une offre

Une fois qu’une entreprise est en base, on va pouvoir créer une offre. Pour ce faire nous devons tout d’abord créer des catégories et ensuite l’utilisateur pourra créer l’offre.

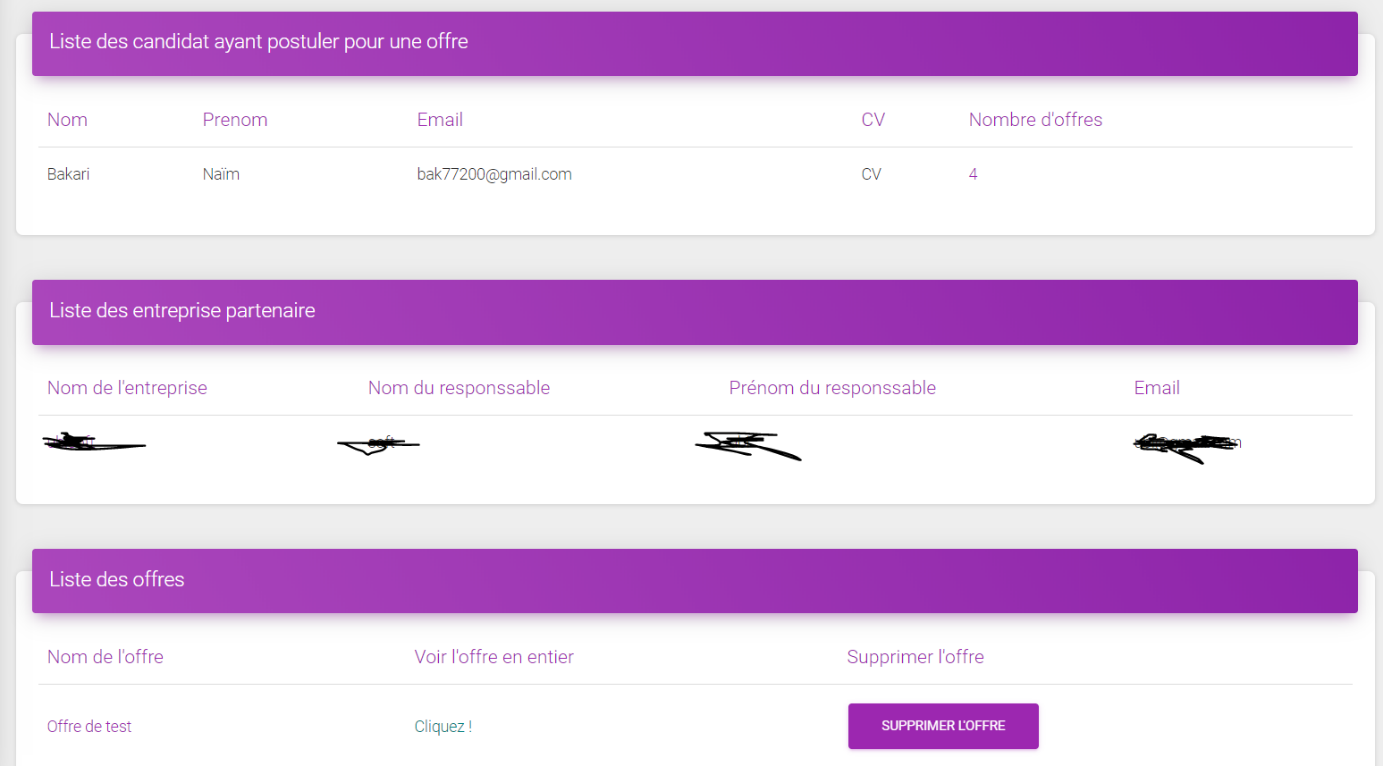


Et maintenant on crée l’offre. Pour créer une offre on a besoin de renseigner le type de l’offre (CDD, CDI), Une ou plusieurs catégories, le nom de l’entreprise, le type de rémunération ( par jours, par mois), la rémunération en €, le Titre de l’offre, la date de début et éventuellement de fin, l’adresse et enfin une description.

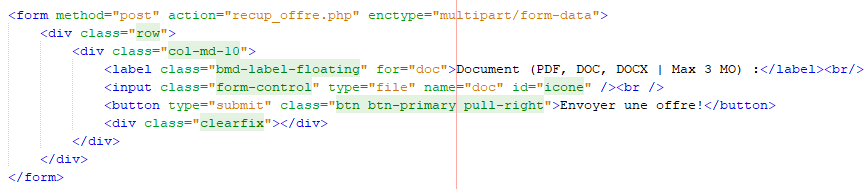
Il y a encore une fois plusieurs tables donc c’est le même principe que pour ajouter un utilisateur, avec *lastInsertId(),* sauf qu’ici on l’utilise plusieurs fois car on a trois tables différentes (on l’utilise donc deux fois).

#### Les différentes listes

Enfin, j’ai créé une page où il y a toutes sorte de liste. Une liste des nom et coordonné des entreprises partenaire, une liste des candidats ayant postuler pour une ou plusieurs offres, d’ailleurs, lorsqu’on clique sur le candidat, on a une liste de toutes les offres aux quels le candidat à postuler pour pouvoir assurer un suivi, on a aussi un bouton pour pouvoir voir son CV. Enfin, il y a une liste de toutes les offres sur le site pour pouvoir les supprimer.



#### Entreprise qui envoie une offre.

Dans le cahier des charges il est stipulé qu’une entreprise ne peut pas créer une offre elle-même. Elle envoie donc un document (PDF, doc, docx) via l’application web. Pour ce faire j’utilise un formulaire d’envoie de fichier vers le serveur (formulaire d’upload). Voici donc le formulaire :

J’utilise donc la méthode post, action signifie que lorsque l’utilisateur aura appuyé sur "Envoyer une offre !", qui est un bouton d’envoie (type submit), tous les post seront transmis au scripte PHP nommé recup\_offre.php. La balise input qui a un type file permet au navigateur d’ouvrir une fenêtre qui permet de choisir le fichier à envoyer. Une fois que l’utilisateur a choisis son fichier et l’a envoyé, il est transmis au script PHP :



La première ligne *if(!empty($\_FILES['doc']['tmp\_name']))* est un test qui permet de voir si il y a un fichier qui a été sélectionner par l’utilisateur. S’il y a un fichier le script continue sinon il s’arrête. Dans le cas où il y a bien un fichier, le scripte va vérifier si il y a un problème ou pas grâce au test if(is\_uploaded\_file($\_FILES['doc']['tmp\_name'])).

S’il n’y a pas de problème alors on va vérifier la nature du document c’est-à-dire son extension (l’extension d’un fichier se trouve après le point dans les fichiers par exemple un\_exemple.exe ou bien deuxieme\_exemple.doc). Il est impératif de faire cette vérification car si une personne mal intentionnée arrive à mettre du PHP par exemple elle pourrait détruire toute la base donnée par exemple. C’est une énorme fails de sécurité c’est pour cela que le système va seulement accepter les fichiers en .doc, .docx ou bien les PDF. Pour se faire on utilise la fonction PHP mime\_content\_type qui permet de récupérer l’extension du fichier. Ensuite on fait une vérification pour voir si l’extension correspond à .PDF, .doc ou bien .docx.

Ensuite on vérifie la taille du fichier. Si la taille du fichier est supérieure à 3MO alors le système va refuser le fichier. Cette partie est essentiel car si une personne mal intentionnée lui venait à l’idée de mettre un fichier de 2TO par exemple dans notre serveur, il risque de très vite saturer.

Enfin, si tous les tests au-dessus sont bon, alors on va entrer le fichier dans le serveur mais avant il faut formater le nom du fichier car si un fichier a le même nom qu’un autre, alors celui déjà présent dans le serveur seras supprimer, ce n’est pas l’idéale. Pour pouvoir palier a ce problème, je supprimer l’ancien nom puis j’écris à la place offre\_de suivi du nom de l’entreprise puis j’insère une série de chiffre et lettre. Pour terminer, j’enregistre le fichier dans un endroit confidentiel. Une fois que le logiciel a upload le fichier, ce dernier enregistre donc le fichier de cette manière :



## Test de fonctionnement

Une fois que l’application a été réaliser, il a fallu tester son bon fonctionnement. Pour ce faire, j’ai dû créer diffèrent scénario à réaliser. Je vais présenter ici quelque scénario que j’ai réalisé pour tester mon application web. Ces tests ont été réaliser pour toutes les fonctionnalités de l’application web, mais je ne les ai pas toutes montrée ici.

### Test de l’inscription/connexion d’un utilisateur

En tant qu’utilisateur, on peut :

|  |  |  |  |
| --- | --- | --- | --- |
| Objectif de test | Résultat attendu | Statut | Commentaire |
| S’inscrire comme candidat | Un utilisateur de type candidat est ajouté dans le système |  |  |
|  | Se rendre sur le site | OK |  |
|  | Cliquer sur "Inscription" | OK | L’utilisateur est redirigé vers le formulaire d’inscription |
|  | Remplir les champs obligatoire (champs marqués par un astérisque) | OK | Le mot de passe est marqué deux fois et doit être strictement identique |
|  | Cliquer sur "s’inscrire" et être rediriger vers la page de connexion | OK |  |
| Se connecter au système | Un utilisateur déjà connu du système peut se connecter |  |  |
|  | Se rendre sur la page de connexion | OK |  |
|  | Remplir le login (adresse mail) et le mot de passe | OK | Si le login ou le mot de passe sont incorrect l’utilisateur n’est pas connecter |
|  | Cliquer sur "connexion" | OK | L’utilisateur est redirigé vers son espace personnel |

### Test d’ajout d’une offre

Pour pouvoir ajouter une offre, un utilisateur interne ou administrateur doit :

|  |  |  |  |
| --- | --- | --- | --- |
| Objectif | Résultat attendu | Statut | Commentaire |
| Ajouter une entreprise | Un utilisateur de type entreprise est enregistrer dans le système |  |  |
|  | Se connecter en tant qu’utilisateur interne ou admin | OK |  |
|  | Cliquer sur "ajouter utilisateur" | OK | L’utilisateur est redirigé vers la page de création d’utilisateur |
|  | Remplir tous les champs | OK | Choisir de créer un utilisateur entreprise si l’utilisateur connecter est l’administrateur |
|  | Cliquer sur "ajouter" | OK |  |
| Créer une ou plusieurs catégories | Une ou plusieurs catégories est ajouté au système |  |  |
|  | Cliquer sur "ajouter une catégorie" | OK |  |
|  | Remplir un ou plusieurs champs afin d’ajouter des catégories | OK | L’utilisateur crée une catégorie en écrivant le nom de celle-ci |
|  | Cliquer sur "ajouter" | OK |  |
| Ajouter une offre | Une offre est ajoutée au système |  |  |
|  | Choisir un type d’offre | OK | Choisir entre un CDD, CDI, etc |
|  | Choisir une ou deux catégories | OK |  |
|  | Choisir une entreprise | OK |  |
|  | Choisir un type de rémunération | OK | Une rémunération par jour, par mois, par semaines |
|  | Remplir tous les champs restant | OK |  |
|  | Cliquer sur "crée une offre" | OK |  |

## Conclusion

Cette année en alternance au sein du CFI de Montigny-le-Bretonneux ainsi que dans mon entreprise Expressur, m’a permis de faire un premier pas dans la vie professionnel. J’ai pu découvrir un métier qui me passionne et acquérir de nouvelle compétence grâce au projet qui m’a été confier durant cette année.

Cette année m’a permis d’avoir plus de savoir-faire dans le domaine de l’informatique que se soit en théorie ou bien en pratique. Cette année m’a été bénéfique pour le travail d’équipe, entendre les différentes idées de mes collègues et mettre en pratique les meilleurs choix. J’ai pu m’améliorer dans le développement même s’il me reste encore du travail afin d’être plus à l’aise avec le langage PHP.

Je remercie mon tuteur ainsi que mes professeurs pour cette année qui ne peut être que bénéfique pour moi.