



Création d'une CVthèque

RÉSUMÉ

Dans le cadre du cours de Programmation WEB, nous avons développé une CVthèque. Dans ce rapport, nous explicitons notre démarche, modèle de données ainsi que les différents éléments qui composent notre site.

Abla MIKOU & Maïrâme NDIAYE

Responsable du cours : Mr AZOULAY Yohan

Plan du rapport

I.	INTRODUCTION.	3
II.	LES OBJECTIFS GENERAUX DU SITE.	3
III.	LES UTILISATEURS CIBLES.	3
IV.	DESCRIPTION DE L'ENVIRONNEMENT TECHNIQUE.	4
1.	TECHNOLOGIES UTILISEES.	4
2.	BASE DE DONNEES.	4
3.	L'API DE PAIEMENT PAYPAL.	4
V.	ARCHITECTURE TECHNIQUE DU SITE.	5
1.	ORGANISATION DU CODE SOURCE.	5
2.	DIAGRAMME DE BASE DE DONNEES.	6
3.	DIAGRAMMES DE CAS D'UTILISATION.	7
4.	DESCRIPTION DES FICHIERS PRINCIPAUX.	8
VI.	STRUCTURE DU SITE.	9
1.	ARBORESCENCE DES PAGES.	9
VII.	ERGONOMIE DU SITE.	10
1.	CHARTE GRAPHIQUE.	10
2.	PRESENTATION DES PAGES.	10
VIII.	PLAN DE TESTS EFFECTUE.	13
IX.	DIFFICULTES.	14
X.	LES EVOLUTIONS A MOYEN ET LONG TERME.	14
XI.	CONCLUSION	15

I. Introduction.

Dans le cadre du cours de Programmation WEB en M1 MIAGE en apprentissage au sein de l'Université Paris Dauphine, nous avons développé une CVthèque. Dans ce rapport, nous exposerons les différentes fonctionnalités de notre site, son architecture ainsi que les difficultés que nous avons rencontré pour sa réalisation.

II. Les objectifs généraux du site.

Dans le monde du travail, il est toujours du rôle du demandeur de chercher les offres des entreprises et y postuler pour espérer d'être contacté par ces derniers. Cette recherche d'offre prend énormément de temps pour les chercheurs d'emploi. Ayant nous-même vécues des situations pareilles, nous avons choisi de développer une CVthèque. L'objectif de ce site est avant tout de rééquilibrer les pouvoirs entre chercheurs d'emploi et entreprises, qui sont souvent en situation de force.

Le site permet de mettre en contact les entreprises qui disposent de poste à promouvoir et des chercheurs d'emploi. Seulement, ce sont uniquement les entreprises qui peuvent contacter les candidats qui les intéressent. Afin d'avoir accès à ce service, les entreprises devront bien évidemment payer un abonnement. Notre activité sera monétisable de la sorte, nous avons mis en place un système de paiement en ligne sécurisé avec Paypal.

Évidemment, notre objectif final est de faire connaître notre marque et avoir une image de marque forte en se distinguant de la concurrence pour pouvoir disposer d'un maximum de part de marché. Gagner la fidélité et la confiance des entreprises et des chercheurs d'emploi est donc enjeu important pour nous. Notre marque doit être facilement repérable.

III. Les utilisateurs ciblés.

Notre site *My Job* dispose d'un positionnement assez large car il cible d'un côté les chercheurs d'emploi et de l'autre côté les entreprises.

Pour les employés, il est à destination des personnes cherchant tous types de contrat, alternance, CDI, CDD, ... Nous disposons ainsi d'une stratégie multi segment. En effet, nous segmentons le marché par fonction (recruteur ou chercheur), par type de contrat et niveau d'expérience. Nous avons ciblé tous ces segments pour couvrir l'ensemble de la demande et l'offre du marché du travail. Notre entreprise cible donc les chercheurs d'emploi qui ont entre 18 ans, âge moyen pour intégrer le marché du travail et 65 ans, âge de départ à la retraite.

Concernant les entreprises, notre site est à destination de toutes celles qui disposent de postes à promouvoir. En effet, le prix d'un abonnement est de 19,99 euros par utilisateur/mois. Ce prix est assez bas, ce qui permet aux grands groupes ainsi qu'aux startups de souscrire à un abonnement. Bien évidemment, pour que notre activité soit rentable, l'abonnement est sur un engagement de 12 mois.

IV. Description de l'environnement technique.

1. Technologies utilisées.

Notre site Web a été développé en NodeJS. Le code source est composé de fichier « EJS » qui contiennent notre code HTML. Ce langage nous permet d'intégrer dans le HTML du JavaScript. Cela nous a évité de devoir créer des fichiers en HTML et d'autres en JavaScript pour gérer nos vues.

Nous avons aussi utilisé « Express », qui est un framework basé sur NodeJS. Il nous a été très utile pour la construction de notre application Web, d'une part pour la création du serveur et d'autre part pour la partie Backend avec nos différentes méthodes get, post ainsi que pour le routage.

Remarque : Pour la gestion du projet, nous avons utilisé GitHub pour pouvoir versionner le projet et pour pouvoir travailler dessus de façon parallèle.

2. Base de données.

Pour stocker et gérer notre base de données, nous utilisons MariaDB. La connexion à notre base est effectuée à l'aide du module *mysql* que nous avons installé avec la commande *npm install*. Le fichier « dbService.js » contient l'url, le login et le mot de passe pour la connexion à la base voulue. Les ressources sont récupérées à l'aide des différentes méthodes de cette classe.

3. L'API de paiement Paypal.

Pour monétiser notre site, nous avons décidé d'utiliser l'API de paiement Paypal pour sécuriser le paiement de nos utilisateurs. Pour connecter notre application à notre compte Paypal, nous avons suivi les instructions fournis sur leur site pour les développeurs. Nous avons récupéré les différents scripts permettant d'intégrer ce système de paiement sur notre site et l'avons adapté à notre application. Ainsi, lorsqu'une entreprise désire s'inscrire sur *My Job*, elle a la possibilité de payer soit via Paypal soit par carte bancaire et tout cela de façon totalement sécurisée.

V. Architecture technique du site.

1. Organisation du code source.

Nous avons organisé notre code en utilisant le modèle MVC, Model View Controller, qui est le plus répandue dans le monde du développement. L'objectif étant de bien séparer la logique métier et l'affichage de l'application. Le contrôleur demande les ressources au modèle, qui se charge d'interroger la base de données et de retourner les résultats. Le contrôleur analyse ensuite les résultats renvoyés, et si tout est correct, les transmets aux vues pour qu'elles affichent les informations.

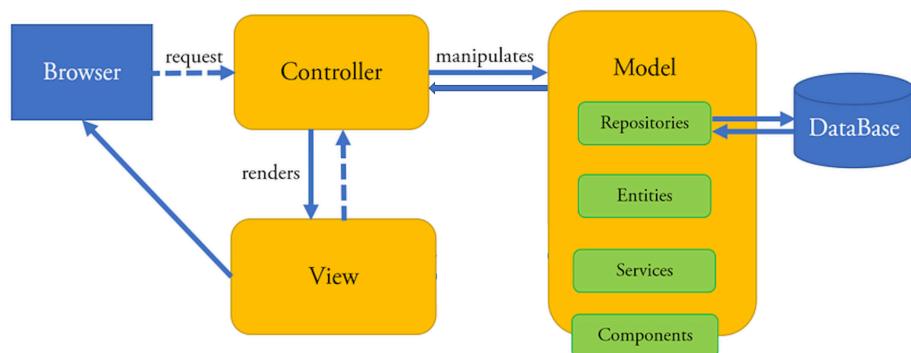


Figure 1 : Architecture utilisée pour le projet.

Dans notre projet, le **contrôleur** est composé du fichier « *app.js* » qui contient notre serveur. Lorsqu'une ressource lui est demandé ou transmise par les pages du site, il effectue des requêtes au modèle à l'aide de la classe *dbService*, qui permet d'interroger et d'alimenter la base de données, et renvoie les réponses retournées à view pour l'affichage des informations dans les pages du site Web.

Le **model** est donc composé du fichier « *dbService.js* » qui est le seul à être en contact avec la base de données. Toutes nos requêtes pour récupérer, mettre à jour, insérer et supprimer des données sont à l'intérieur de cette classe.

La **vue** de notre projet dispose de deux dossiers : *views* et *public*. Le dossier *views* contient tous les fichiers « **.ejs* » qui contiennent le code HTML des pages. Le dossier *public* est quant à lui composé de toutes les images, les fichiers PDF et les CSS's de notre site.

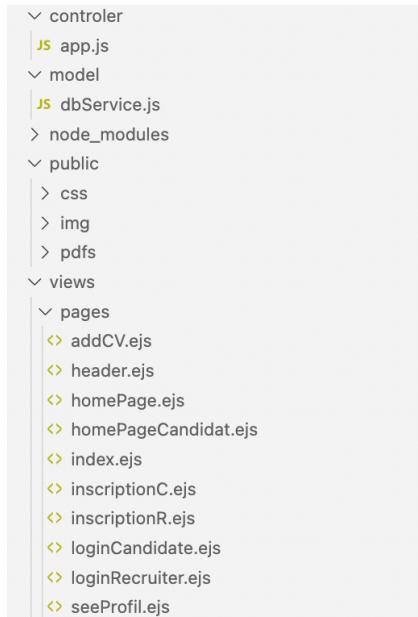


Figure 2 : Architecture du projet

2. Diagramme de base de données.

Nous utilisons le modèle de donnée suivant pour la collecte et le stockage des informations nécessaires au bon fonctionnement de notre site.

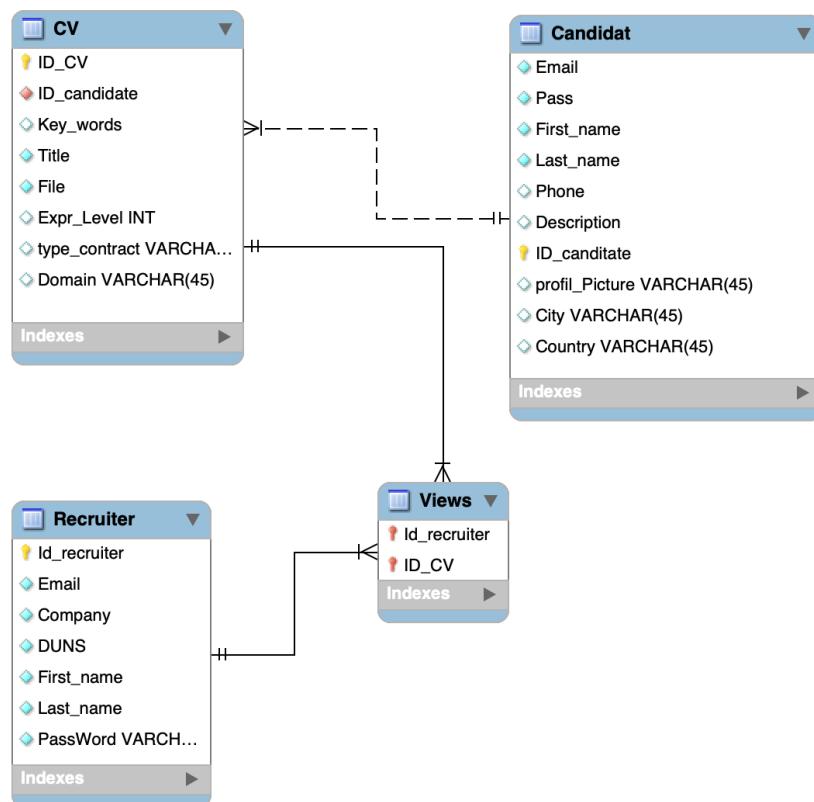


Figure 3 : Architecture de notre base de données.

Notre base de données se compose des quatre tables suivantes :

- ◊ La table « **Candidat** » contient toutes les informations concernant le profil créé par un chercheur d'emploi. Ces informations seront utiles pour adapter les pages du site à un candidat donné ainsi que pour permettre au recruteur de contacter le candidat.
- ◊ La table « **Recruiter** » dispose des données concernant le recruter. Des informations de création de compte assez classique.
- ◊ La table « **CV** » stocke tous les CVs de tous les candidats de notre base. Un candidat peut disposer de plusieurs CVs. Par exemple, une personne qui dispose de deux masters peut effectuer deux métiers différents. Sur notre site, le candidat a donc la possibilité de créer tous ses « sous-profil », un par métier recherché. Notre table « **CV** » est donc multi occurrencée.
- ◊ La table « **Views** » permet de récupérer les vues d'un CV. Ces données sont indispensables pour effectuer des statistiques. Ainsi, un candidat pourra, par exemple, voir lequel de ses CV's est le plus consulté par les recruteurs.

3. Diagrammes de cas d'utilisation.

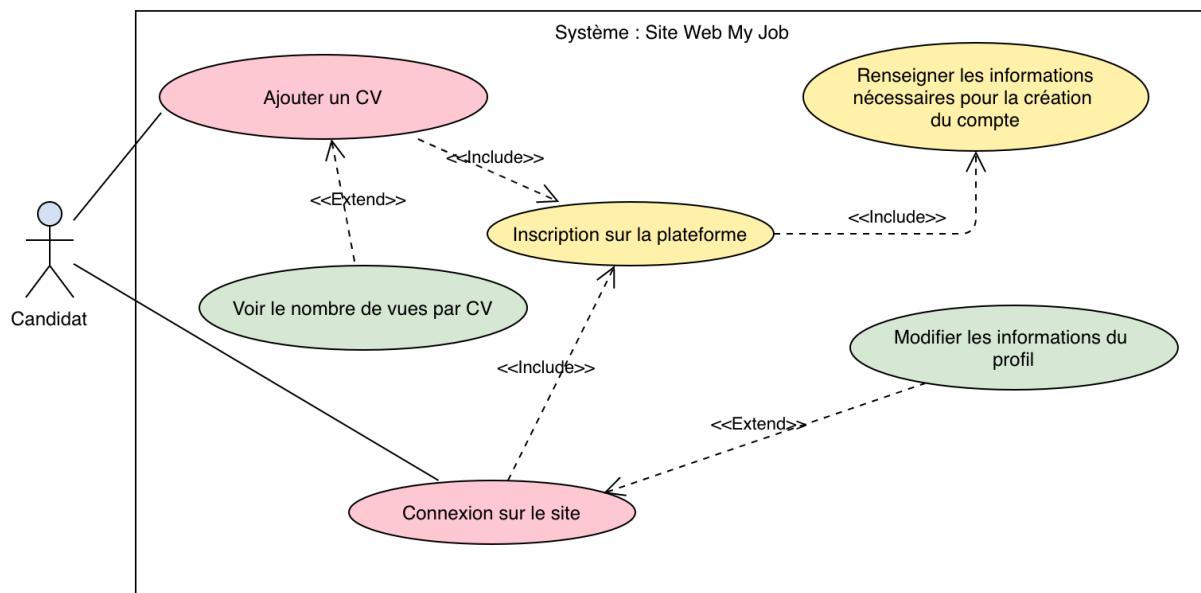


Figure 4 : Diagramme de cas d'utilisation pour les chercheurs d'emploi.

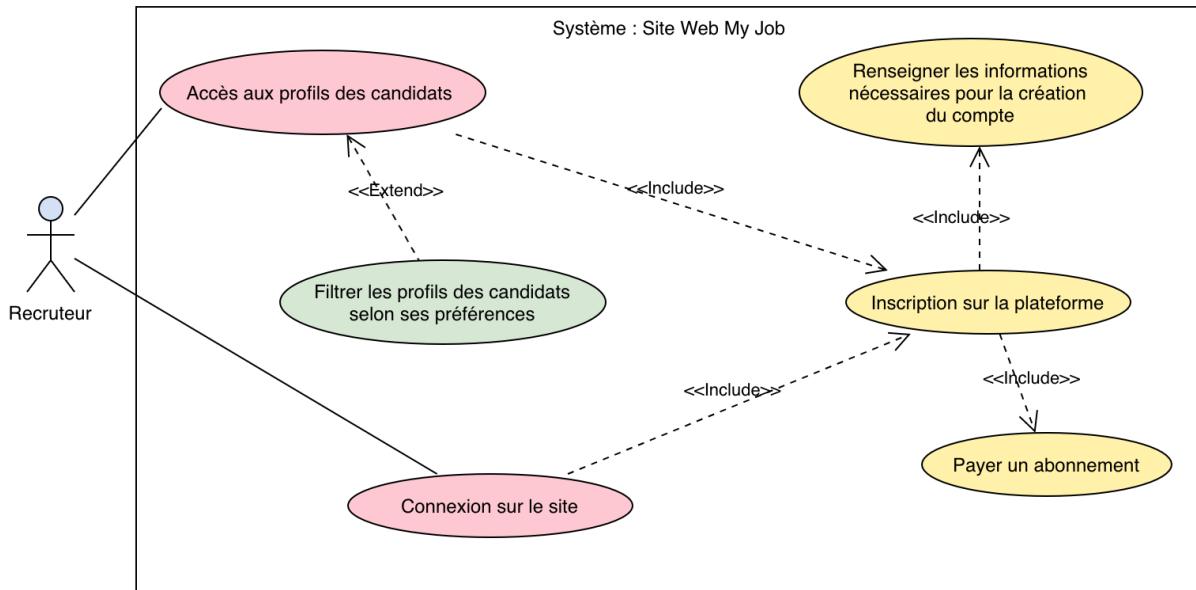


Figure 5 : Diagramme de cas d'utilisation pour les recruteurs.

4. Description des fichiers principaux.

Notre projet est constitué de plusieurs fichiers « .ejs » qui contiennent les pages à afficher. Un fichier correspond à une page WEB. Dans cette partie, nous allons brièvement présenter l'utilité de chacun des fichiers présents.

Ces fichiers sont les suivants :

- « **homePage.ejs** » : c'est la page d'accueil du site, c'est-à-dire la première page que l'utilisateur dispose quand il accède au site, avant même de se connecter. Sur cette page, il peut voir différentes informations sur notre entreprise. Il peut aussi cliquer sur les boutons pour se connecter ou s'inscrire.
- « **loginCandidate.ejs** » : Page d'authentification pour les chercheurs d'emploi. Après avoir renseigné les informations, une gestion d'authentification est réalisée à l'aide de requêtes vers la base de données vérifiant que l'utilisateur a bien un compte.
- « **loginRecruiter.ejs** » : Page d'authentification pour les entreprises. Après avoir renseigné les informations, une gestion d'authentification est réalisée à l'aide de requêtes vers la base de données vérifiant que l'utilisateur a bien un compte.
- « **inscriptionC.ejs** » : Page de création de compte pour les candidats. Lorsque le formulaire est correctement rempli, la table « Candidat » de la base de données est alimentée.
- « **inscriptionR.ejs** » : Page de création de compte pour les recruteurs. Lorsque le formulaire est correctement rempli et que le paiement est réalisé la table « Recruter » de la base de données est alimentée.

- « ***homePageRecruiter.ejs*** » : correspond à la page principale du recruteur une fois connecté à la plateforme. C'est sur cette page qu'apparaît les différents profils des candidats, qu'il effectue ses recherches et filtre les candidats en fonction de ses préférences. A partir de cette page, le recruteur a la possibilité de cliquer sur les profils qui l'intéressent pour avoir plus d'informations.
- « ***homePageCandidat.ejs*** » : ce fichier dispose du code pour l'affichage de la page principale du candidat après connexion. A partir de cette page, le candidat a la possibilité de voir ses informations et de les modifier, voir ses différents CVs triées dans l'ordre du plus vue au moins vue, gérer ces derniers et ajouter de nouveaux profils.
- « ***addCV.ejs*** » : Page permettant au candidat d'ajouter un nouveau CV à son profil. Lorsque le formulaire est correctement rempli, la table « CV » de la base de données est alimentée.
- « ***seeProfil.ejs*** » : c'est la page qui permet d'afficher les informations d'un profil. Sur cette page, il y a le CV, ainsi que des informations de contact pour permettre au recruteur de rentrer en contact avec le chercheur d'emploi.

VI. Structure du site.

1. Arborescence des pages.

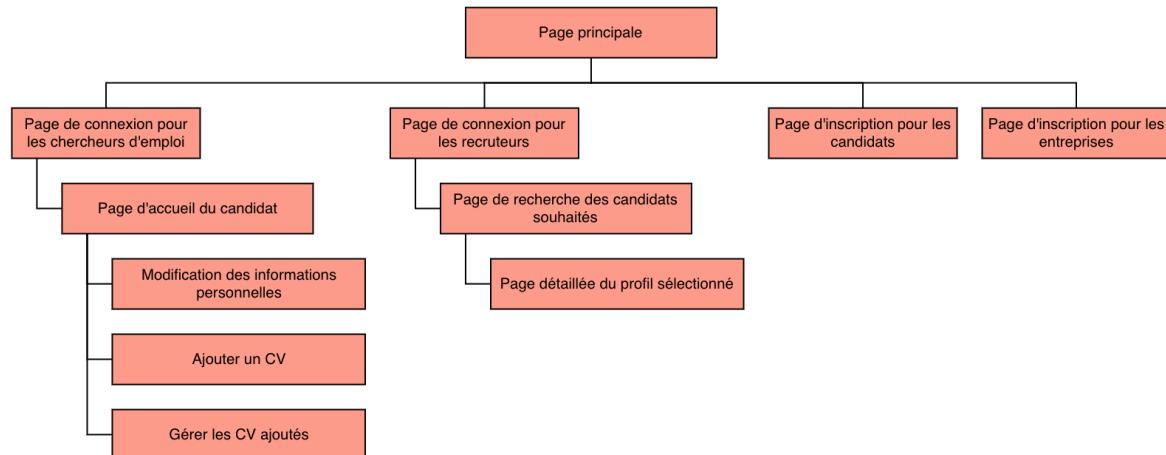


Figure 6 : Arborescence du site.

VII. Ergonomie du site.

1. Charte graphique.

Notre volonté était de réaliser un site web très professionnel au vue de son objectif mais nous souhaitions qu'il soit représentatif de nos deux personnalités. C'est ainsi que nous sommes parties sur des tendances de rose pour la charte graphique. Notre couleur principale est un rose pâle très tendance. Nous avons mélangé ce rose à du noir et du blanc qui restent des couleurs professionnelles pour redonner un équilibre à notre site. Cela permet d'avoir un site classe, professionnel et efficace.

Toujours dans l'idée de simplicité et de modernité, nous avons mis des radius à tous nos boutons pour fournir un coté moins rigide au site et donner une meilleure expérience utilisateur. Nous avons choisi « *Raleway, sans serif* » comme police principale pour notre site. Elle se présente comme ceci.

Raleway

Figure 7 : Prototype de la police utilisée.

Pour le logo, nous avons choisi un logo très simple en reprenant les couleurs de notre site : du rose et du noir. La main dans la main représente l'accord entre un candidat et une entreprise qui s'effectuera sur notre site « *My Job* ».



Figure 8 : Logo de notre marque

2. Présentation des pages.

Dans cette partie, nous montrons quelques visuels de notre site. Il s'agit de quelques captures d'écrans permettant d'illustrer ce dont nous parlons plus haut, lors de la soutenance, le site sera présenté de façon complète et dynamique.

Avec MyJob n'attendez plus que les candidats postulent à vos offres.

- ✓ Le pouvoir de construire une expérience de recrutement unique.
- ✓ Soyez le 1er à contacter les candidats qui vous intéressent.
- ✓ Des candidats motivées qui n'attendent que d'être contacté.

A Propos de My Job

My Job, c'est la rencontre de deux entrepreneuses dans l'âme, passionnées par les évolutions technologiques et numériques. Elles vont vite, peut-être trop vite dans un monde du travail qui a du mal à fournir les talents indispensables à cette innovation galopante.

MIKOU Abla, CEO de My Job, était créatrice de startups d'e-commerce. NDIAYE Mairame, CTO, était développeuse freelance. Tous deux se retrouvent en 2019 autour d'une problématique commune et bien connue des entreprises : la difficulté de trouver des candidats motivés et passionnés pour faire avancer leurs projets. L'année suivante, elles fondent ensemble My Job. Leur ambition : changer l'expérience de recrutement pour les entreprises et les chercheurs d'emploi.

My Job en chiffres

10 000 recrutements par an	+ de 1000 entreprises nous ont fait confiance	Présent dans les 7 continents	Recommandé à 94% par nos utilisateurs

Ils nous ont fait confiance et ils ont eu raison de le faire.

Tous nos utilisateurs sont interrogés par le biais d'enquêtes de satisfaction dans le but d'évaluer notre niveau de qualité et de savoir-faire. Les résultats nous permettent une amélioration continue de l'ensemble de nos services pour adapter notre offre aux besoins de nos utilisateurs. Que ce soit avec les particuliers, les grandes entreprises ou les startups, nous remplissons les objectifs qu'ils nous ont fixés avec la même exigence ... Alors pourquoi pas vous !

Rejoignez nous sur nos réseaux

f t G+ ln

Copyright ©2022 Designed By
MAÏRAME NDIAYE & ABLA MIKOU

Figure 8 : Page d'accueil du site.

The screenshot shows the 'My Job' recruitment registration page. At the top left is the logo 'My Job' with a stylized bird icon. At the top right is a 'Accueil' button. The main heading 'Bonjour ! Bienvenue sur My Job' is centered above a form for creating a user account. The form fields include: 'Nom' (Name), 'Prenom' (First Name), 'Nom de votre entreprise' (Name of your company), 'Numéro DUNS' (DUNS number), 'Email', and 'Saisissez un mot de passe' (Enter a password). To the right of the form, a price '19,99 € TTC utilisateur/mois' is displayed, along with payment method options: 'PayPal' (yellow button) and 'Carte bancaire' (black button). Below these buttons is the text 'Optimisé par PayPal'. A 'Let's Go !' button is located at the bottom center of the form area.

Figure 9 : Page d'inscription recruteur

The screenshot shows the 'My Job' candidate profile page. At the top left is the logo 'My Job' with 'Mes informations'. At the top right is a 'Accueil' button. The main section features a circular profile picture of a woman and her personal details: 'Paris, France', 'Email: mairame.ndiaye@dauphine.eu', and 'Téléphone: 0782891297'. Below this is a 'Description' field containing the text: 'Etudiante passionnée de mathématiques et informatique. Bienvenue sur mon profil :)'. To the right of the profile details is a 'Modifier mes informations' button. Below this section is a heading 'Mes profils' followed by three cards representing different CV profiles:

- CV 1:** Titre CV: Alternance IASD, Domaine: Intelligence Artificielle, Contrat: Alternance, Expérience: 1 ans, Nombre de vues: 2. Buttons: Gérer (orange), Ouvrir (orange).
- CV 2:** Titre CV: Recherche Alternance IA, Domaine: Informatique, Contrat: Alternance, Expérience: 2 ans, Nombre de vues: 2. Button: Gérer (orange).
- CV 3:** Titre CV: Recherche Stage Python 3, Domaine: Informatique, Contrat: Stage, Expérience: 0 ans, Nombre de vues: 1. Buttons: Gérer (orange), Ouvrir (orange).

Figure 10 : Page principale candidat

Remarque : L'ensemble des pages du site sont responsives, elles s'adaptent toutes à des tailles d'écrans plus petits. La responsivité sera démontrée au cours de la soutenance.

VIII. Plan de tests effectué.

Pour voir si notre site répond à nos exigences, nous avons réalisé plusieurs tests fonctionnels que nous allons décrire dans cette partie.

Premièrement, depuis la page principale du site, nous avons tester les boutons de connexions et inscriptions pour les chercheurs d'emploi et les recruteurs.

Pour l'inscription, nous disposons de deux boutons différents pour les candidats et les entreprises. En fonction du statut de la personne, le formulaire d'inscription ne demande pas les mêmes informations. De plus, pour les entreprises, il y a la partie pour choisir leur mode de paiement et payer leur abonnement.

Sur clic du bouton connexion, nous sommes redirigés vers une page pour renseigner le login et le mot de passe. En fonction du statut de la personne qui se connecte, le site redirige soit vers l'accueil du candidat ou l'accueil du recruteur. Si c'est une session candidat, la page contenant ses informations personnelles et les CVs qu'il a ajouté s'affiche, sinon elle redirige vers la page de recherche des profils pour le recruteur. Cette partie répond donc bien à nos souhaits.

Le deuxième test concerne la connexion à la base de données et la récupération des informations qui nous intéressent. La partie la plus importante est l'affichage des profils qui constitue l'élément essentiel du site web. Pour tester cette partie, nous avons ajouté plusieurs nuplets à nos tables « Candidat » et « CV » puis récupérons ces éléments pour les afficher.

Après avoir tester la récupération des éléments de la base de données, il était évident de tester si les filtres que nous voulons appliquer lors de la recherche de profils étaient fonctionnels. Nous avons donc inséré des nuplets avec des champs de table qui contiennent des valeurs intéressantes pour le filtrage des profils. Sur clic du bouton « Filtrer », la page affichant les profils ne contenait plus que les candidats correspondants à nos critères de recherches.

Le troisième test est pour le stockage des fichiers. En effet, pour notre projet, nous devons stocker des fichiers PDF ainsi que des images qui représentent des informations importantes du chercheur d'emploi. Nous avons donc testé cette partie à plusieurs reprises : après remplissage du formulaire et validation, le fichier est bien déplacé dans le dossier en local et son nom est inséré comme valeur de champ dans la base.

Un autre test portait sur la modification des informations du candidat, la gestion et l'ajout d'un CV. En effet, il y a plusieurs formulaires différents à remplir pour effectuer ces actions. Pour la modification des coordonnées personnelles, nous avons modifié le profil d'un candidat et nous avons vérifié que le nuplet le concernant est bien mis à jour dans la base de données. La démarche pour le formulaire de gestion de CV a été la même. En ce qui concerne l'ajout du CV, quand nous cliquons sur le bouton « Ajouter » la table « CV » de la base est bien alimenté et le fichier PDF est bien stocké dans le bon dossier en local.

Le dernier test concerne la page détaillée du profil d'un candidat choisi. En effet, après avoir effectué sa recherche et appliqué les filtres qu'il souhaite, un recruteur dispose de la possibilité de consulter plus en détail le profil d'un candidat qui l'intéresse. Pour cela, nous avons cliqué sur le bouton « *Voir profil* » et la redirection vers la page du candidat concerné s'affiche. Les informations sont bien récupérées.

IX. Difficultés.

Le début du projet a été particulièrement difficile pour nous, en effet, nous n'avions jamais codé en « *NodeJS* », il s'agissait d'un langage totalement nouveau pour nous. Nous avons eu beaucoup de mal à savoir comment commencer le projet et le mettre en place. Nous avons dû nous former un minimum au langage avant de démarrer le projet, ce qui nous a pris un temps non négligeable. Après avoir passé des journées à regarder des tutoriels sur YouTube et sur internet, nous avons finis par comprendre le langage et nous lancer sérieusement dans le développement du projet.

La deuxième difficulté a été sur le choix du projet en lui-même. En choisissant de faire notre projet sur la CVthèque ne nous nous sommes pas facilité la tâche. La construction d'une CVthèque requiert une partie back très importante, qui a été très longue à développer et difficile à comprendre au départ. Au total, à l'état dans lequel se trouve notre site, nous avons dû gérer un back pour plus de 10 formulaires différents et créer une vingtaine de requêtes vers la base de données, sans compter les différents tests qu'il a fallu réaliser à chaque fois pour s'assurer de la cohérence globale de tout ce qui a été réalisée.

Il s'agit des deux principales difficultés que nous avons rencontrés lors du projet, lorsque nous avons réussi à les surmonter, tout s'est relativement bien passé et nous avons réussi à réaliser une grande partie de ce que nous voulions au départ.

X. Les évolutions à moyen et long terme.

Pour ce projet, nous avons voulu implémenter plusieurs fonctionnalités mais malheureusement, nous n'avons pas pu tous les mettre en place par manque de temps.

À moyen terme, nous implémenterons un espace de favoris pour que le recruteur puisse retrouver ses candidats préférés. Pour réaliser cela, nous construirons une nouvelle table dans la base de données qui concernera les CVs aimés par un recruteur donné. De son côté, le candidat pourra voir qui a aimé ses CVs.

À l'ajout de cela, nous souhaitons mettre en place un espace dédié au contact du candidat, un système de messagerie instantanée interne à My Job qui facilitera la prise de contact du recruteur et du chercheur d'emploi. Avec notre version actuelle, le recruteur a la possibilité de contacter le candidat avec une redirection vers Outlook.

À long terme, nous souhaitons mettre en place un système de suivi de contact. Un recruteur aura un espace dans lequel il pourra mentionner s'il a contacté le candidat, s'il est en attente de réponse de la part de ce dernier, etc. Notre site ne sera plus qu'un simple site de recrutement mais un système de gestion des candidatures. Nous pouvons aussi ajouter un espace pour mettre des annonces auxquelles le candidat peut postuler directement et rentrer en contact avec le recruteur.

XI. Conclusion

Ainsi, nous avons pu mettre en avant dans ce rapport l'ensemble du travail que nous avons réalisé pour ce projet. Nous avons pu exposer l'environnement et l'architecture technique du site. A travers différentes illustrations nous avons présenté la structure et l'ergonomie du site. Dans la partie difficulté, nous avons exposé le démarrage compliqué du projet. Nous espérons vous avoir convaincu avec notre CVthèque et espérons que *My Job* sera leader dans le monde du recrutement et qu'elle sera utilisé par plusieurs entreprises et particuliers, concurrençant « *Welcome To The Jungle* », « *Indeed* », ...