

ÉCOLE NATIONALE SUPÉRIEURE D'INFORMATIQUE ET D'ANALYSE DES SYSTÈMES

RAPPORT DE PROJET JEE

Job Finder

Réalisé par :

AIT LAHCEN AHMED

AIT-SSI MOUSLIM

CHATOUANE OMAR

CHICHI HAMZA

Filière :

Génie logiciel

Sous la direction de :

Pr.Mahmoud EL HAMLAOUI

Résumé

Comme nous savons tous, trouver un stage ou un embauche pose toujours problème pour les candidats fraîchement diplômés ou pas encore, de plus, il ya des entreprises qui ont besoin de recruter un profil ayant des compétences spécifiques mais ne savent pas ou chercher.

Donc, c'est ici d'où vient le role de notre application qui répond aux candidats ayants besoin d'un stage ou d'embauche, c'est une interface qui assure la communication entre les candidats et les entreprises, les entreprises s'inscrivent et posent leurs offres et les stagiaires ou les employés s'inscrivent et naviguent les offres et postulent aux offres qui les conviennent, donc c'est à l'entreprise de choisir si elle veut recruter ce dernier ou non.

Lien de repo

Voici le lien de notre repo GitHub :

<https://github.com/MousslimAitssi/jee-s4/tree/master>

Cette repo contient deux projets, le premier (qui 1er essai) contient notre premier projet spring boot qu'on n'a pas pu le finir grâce à des problèmes d'intégration avec jsf, et le deuxième projet (essai2) et qui contient notre projet jsf pris comme projet final

Table des figures

2.1	Diagramme de cas d'utilisation	12
2.2	Diagramme de séquences	13
2.3	Diagramme de classes	14
3.1	Logo de JAVA EE	15
3.2	Logo de JSF	16
3.3	Logo de PrimeFaces	16
3.4	Logo de JUnit 5	16
3.5	La pyramide des tests	17
3.6	Logo de Eclipse IDE	19
3.7	Logo de Apache Tomcat	19
3.8	Logo de WampServer	19
3.9	Logo de MySQL	20
3.10	Logo de HTML	21
3.11	Logo de CSS	21
3.12	Logo de JavaScript	22
3.13	La page d'accueil	22
3.14	Targeted Fields	23
3.15	Le dépôt des offres	23
3.16	La page d'inscription	24

3.17	La page de connexion	24
3.18	Liste des offres	25
3.19	Une page qui contient les informations d'un offre	25
3.20	L'envoi de candidature	26
3.21	la class de test unitaire	26
3.22	test unitaire d'authentification	27

Table des matières

Table des figures	4
Introduction générale	7
1 Présentation du projet	9
1.1 Etude de l'existant	9
1.2 Problématique	10
1.3 Solution proposée	10
2 Etude conceptuelle du projet	11
2.1 Le diagramme de cas d'utilisation	11
2.2 Le diagramme de séquences	12
2.3 Le diagramme de classe	13
3 Réalisation	15
3.1 Les outils de développement	15
3.1.1 JAVA Entreprise Edition	15
3.1.2 JSF	15
3.1.3 Primefaces	16
3.1.4 JUnit 5	16
3.1.5 Les tests	17

3.1.5.1	Définition	17
3.1.5.2	Les différents types de tests automatisés	17
3.1.6	Eclipse IDE	18
3.1.7	Apache Tomcat	19
3.1.8	WampServer	19
3.1.9	MySQL	20
3.1.10	HTML 5	20
3.1.11	CSS 3	21
3.1.12	JavaScript	21
3.2	Scénario d'exécution	22
3.2.1	La page d'accueil	22
3.2.2	Le dépôt des offres	23
3.2.3	L'inscription et la connexion des candidats	24
3.2.4	La consultation des offres	25
3.2.5	L'envoi de candidature	26
3.3	Les tests	26
3.3.1	Structure de la class de test unitaire generer automatiquement	26
3.3.2	Structure de notre class AuthentificationTest	27

Conclusion	27
-------------------	-----------

Bibliographie et Webographie	28
-------------------------------------	-----------

Introduction générale

Dans nos jours, le problème de trouver un stage ou un embauche est devenu très fréquent, puisque le nombre des diplômés s'incrémente, d'autre part, on sait bien que plusieurs entreprises ont besoin d'un profil spécifique, donc on a besoin d'un moyen qui fait apparier les entreprises ayant un manque avec les candidats convenables. Nous, comme étants des étudiants de 2A GL, on est supposés de réaliser un site web dynamique en JEE qui a pour but de répondre à cette problématique.

Notre rapport contiendra les chapitres suivants :

Tout d'abord, nous aborderons la présentation du projet où nous analyserons notre sujet et citerons les différentes problématiques puis les solutions proposés. De plus, une partie conception qui contiendra les diagrammes UML resumants notre site web. Enfin une partie réalisation où on va présenter les outils qu'on a utilisé lors du déroulement de notre projet et un scénario d'exécution de notre application.

Présentation du projet

1.1 Etude de l'existant

La recherche d'emploi a pour but de permettre aux personnes qui souhaitent changer d'emploi ou qui se trouvent au chômage de bâtir un projet professionnel et de trouver un nouvel emploi, ainsi elle peut prendre des formes différentes, selon la cible (fonction publique, grande entreprise, PME). Mais certains points restent communs.

Pour obtenir un résultat satisfaisant, cette recherche est une activité qui doit être structurée car on est amené à respecter des aspects formels, à faire un bilan de ses compétences, à élaborer un plan d'intervention et à réaliser un suivi de ses démarches.

Rechercher un emploi est beaucoup plus efficace lorsque cela se fait avec des organismes ou des associations spécialisés dans ce domaine, mais ces derniers restent limités en ce qui concerne les emplois dans le monde de l'informatique, c'est pour cela que recherche en ligne passant par les réseaux sociaux ou les sites web peut être favorable et peut créer de bonne opportunité soit pour un lauréat ou un stagiaire.

Cette recherche par internet permet la découverte du monde de travail informatique et rester à jour devant les différents profils demandés tels que le : "Data Analyste", "Information Security Analyst", "Web Developer", "Software Developer"...

1.2 Problématique

Trouver un stage ou un emploi n'est pas toujours facile, loin de là. Sans contacts et sans expérience, le stagiaire ou les jeunes diplômé en informatique peut avoir la sensation que la tâche est longue, pénible et complexe. Cela peut être le cas car pour beaucoup de personnes, des difficultés apparaissent en chemin, et se révèlent parfois vraiment pénibles. Absence de réponses, délais interminables, promesses non tenues...

Parmi les erreurs les plus communes est de penser que plus grand sera le nombre d'entreprises auxquelles postuler, plus il y aura un retour. C'est l'erreur la plus répandue mais surtout la plus pénalisante.

Et les problèmes ne manquent pas, ce qui amène à lâcher prise et céder pour n'importe quel emploi.

1.3 Solution proposée

Vu les différents problèmes rencontrés, nous nous sommes mis dans la création un site web dynamique sous le nom de "Job Finder" que nous avons associé à notre projet JEE qui a pour but de mettre en relation les différentes entreprises ou start-up dans le domaine de l'informatique avec les stagiaires et diplômés en recherche d'emploi.

Le site "Job Finder" permettra d'offrir beaucoup des services à savoir :

- Rassembler les différentes informations sur les candidats tels que leurs compétences et leurs domaines ciblés.
- Permettre aux entreprises de mettre leurs annonces ainsi que leurs contacts.
- Permettre au candidat de parcourir les différentes annonces et d'appliquer à un emploi en envoyant son CV, Port-Folio... par email à l'entreprise souhaité.
- Assurer la communication entre les deux partis ceci en restant notifié par email des différentes nouvelles.

CHAPITRE 2

Etude conceptuelle du projet

2.1 Le diagramme de cas d'utilisation

Use Case Diagram

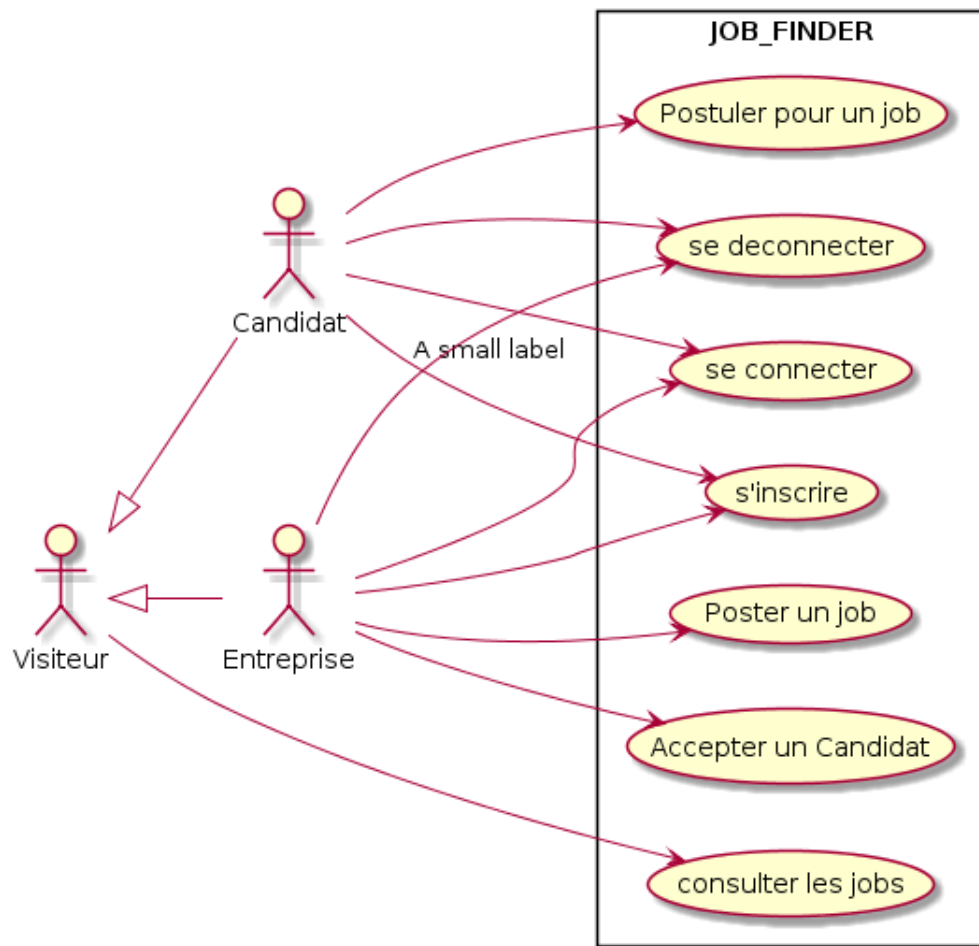


FIGURE 2.1 – Diagramme de cas d'utilisation

2.2 Le diagramme de séquences

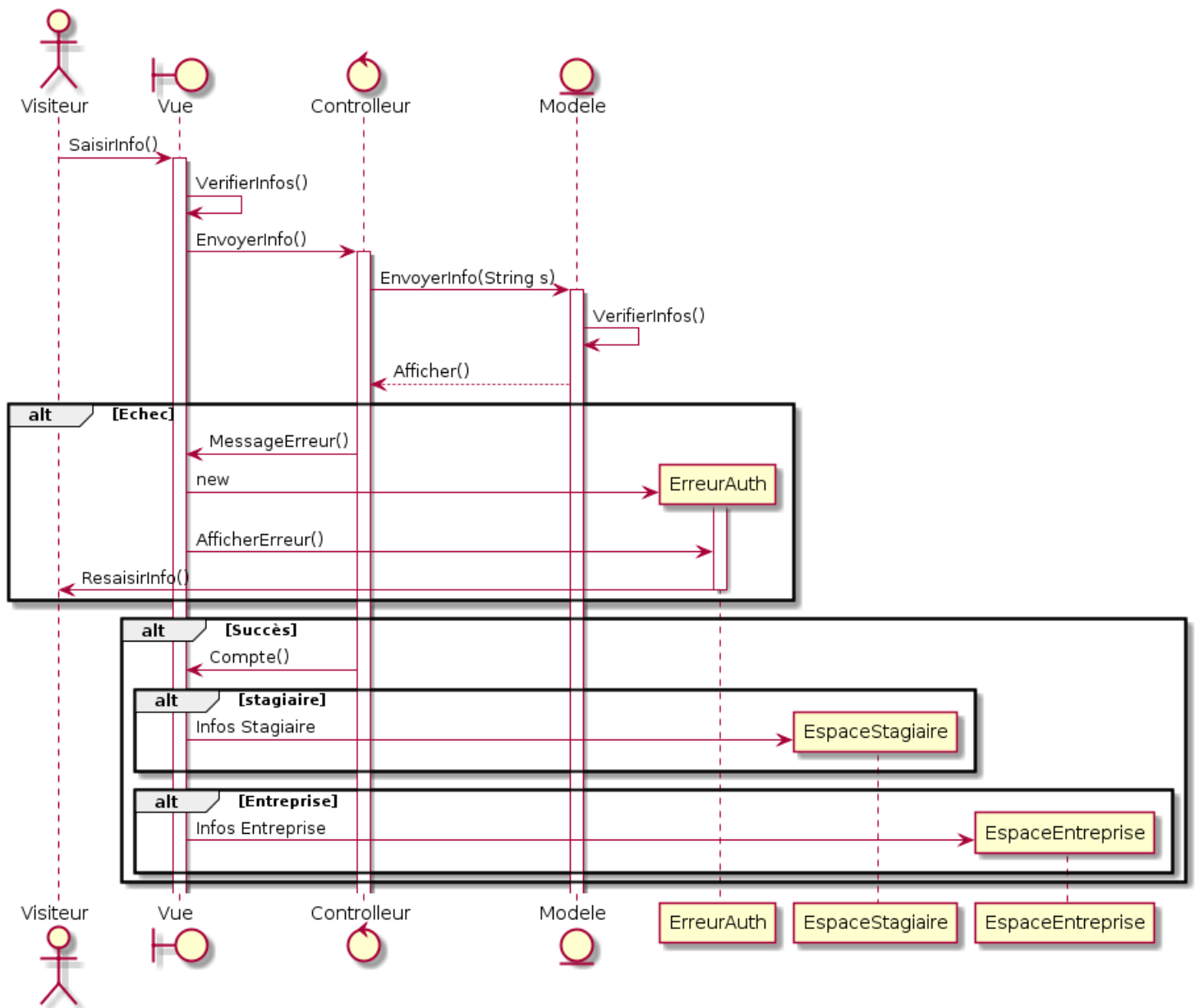


FIGURE 2.2 – Diagramme de séquences

2.3 Le diagramme de classe

Classes - Class Diagram

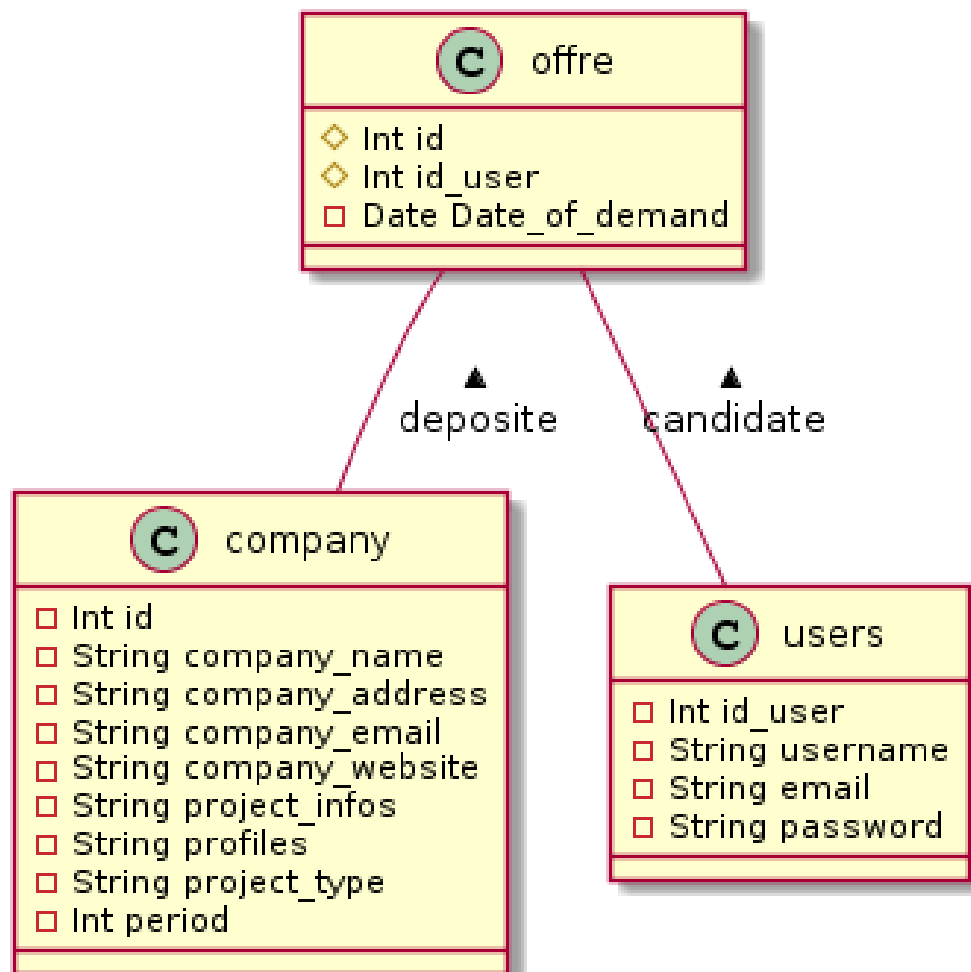


FIGURE 2.3 – Diagramme de classes

3.1 Les outils de développement

3.1.1 JAVA Entreprise Edition



FIGURE 3.1 – Logo de JAVA EE

La plateforme Java Entreprise (Java EE) est un ensemble de spécifications coordonnées et pratiques qui permettent des solutions pour le développement, le déploiement, et de la gestion des applications multi-tiers centralisées sur un serveur. Construite sur la plateforme de Java 2 édition standard (Java SE), la plateforme Java EE ajoute les possibilités nécessaires pour fournir une plateforme complète, stable, sécurisée, et rapide de Java au niveau entreprise. La plateforme Entreprise fournit un ensemble de services permettant aux composants de dialoguer entre eux.

3.1.2 JSF

JavaServer Faces (abrégié en JSF) est un framework Java, pour le développement d'applications Web. À l'inverse des autres frameworks MVC traditionnels à base d'actions, JSF est basé sur la notion



FIGURE 3.2 – Logo de JSF

de composants, comparable à celle de Swing ou SWT, où l'état d'un composant est enregistré lors du rendu de la page, pour être ensuite restauré au retour de la requête.

3.1.3 Primefaces



FIGURE 3.3 – Logo de PrimeFaces

Primefaces est une librairie open source de composants graphiques pour les applications JSF (Java Server Faces) créée par la compagnie Turque PrimeTek. Le développement de Primefaces remonte en 2008 et la première version sera publiée Janvier 2009. Il s'agit d'un ensemble de plus de 100 composants graphiques. A part leur beau look and feel, les composants intègrent pleinement les fonctionnalités AJAX et HTML5, sont responsives et compatibles avec la plupart des navigateurs modernes. Primefaces apporte aussi un framework PUSH (basé sur Atmosphere) pour réaliser des applications temps réel avec l'utilisation de l'API WebSocket. Primefaces existe aussi en version mobile, avec des composants qui s'intègrent bien aux navigateurs des différents systèmes d'exploitation mobile présents sur le marché.

3.1.4 JUnit 5



FIGURE 3.4 – Logo de JUnit 5

Pour effectuer les tests unitaire on a utilisé JUnit, qui est un framework de test unitaire pour le langage de programmation Java. Créé par Kent Beck et Erich Gamma, JUnit est certainement le projet de la série des xUnit connaissant le plus de succès. JUnit définit deux types de fichiers de tests. Les TestCase (cas de test) sont des classes contenant un certain nombre de méthodes de tests. Un TestCase sert généralement à tester le bon fonctionnement d'une classe. Une TestSuite permet

d'exécuter un certain nombre de TestCase déjà définis. Dans un TestCase il n'y a pas de main méthode, chaque test étant indépendant.

3.1.5 Les tests

3.1.5.1 Définition

En développement, les tests visent à vérifier que le produit codé fonctionne comme prévu selon des scénarios prédéfinis et représentatifs. Cela permet de garantir la qualité de ce qui est codé, malgré les contraintes du projet, comme les délais, par exemple.

3.1.5.2 Les différents types de tests automatisés

Les 3 types de tests automatisés les plus courants sont les tests unitaires, d'intégration et fonctionnels. Mike Cohn, l'un des fondateurs du mouvement Agile, présente ces types de tests sous forme de pyramide des tests, en particulier dans le livre *Succeeding with Agile* (en anglais). En utilisant cette métaphore, l'auteur préconise la répartition de ces types de tests dans un projet de développement agile selon la pyramide ci-dessous.

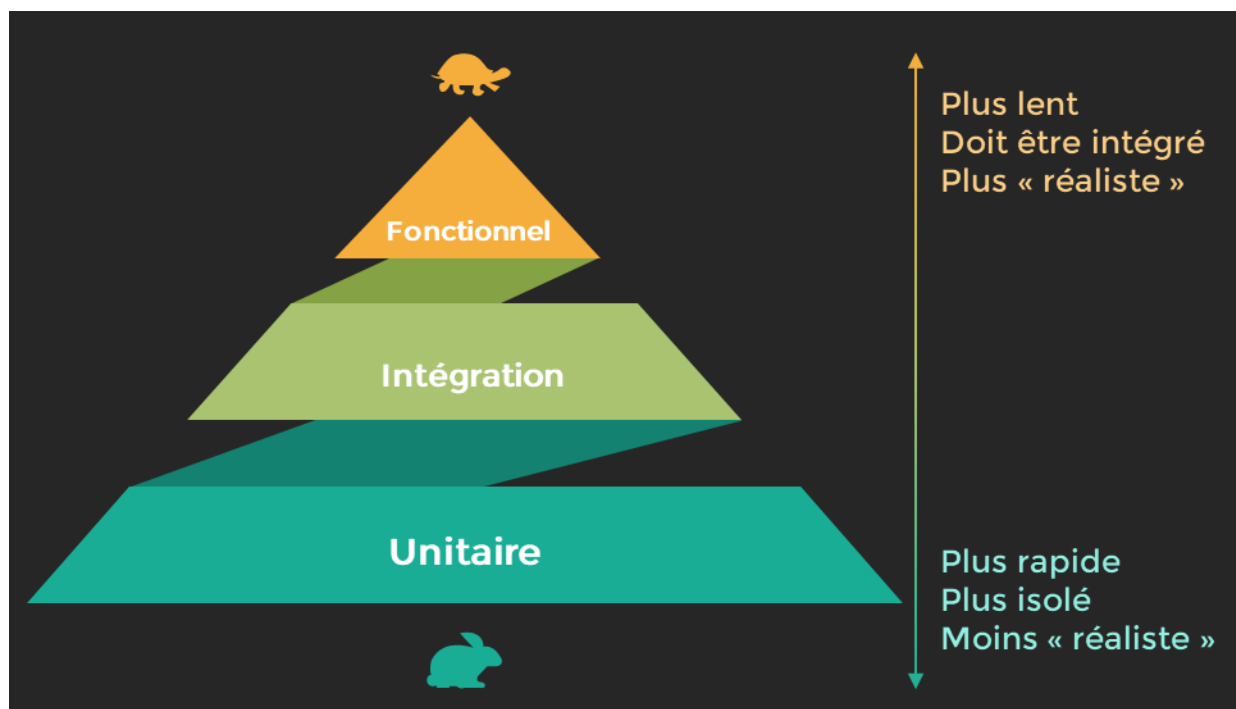


FIGURE 3.5 – La pyramide des tests

Tests unitaires

La plus grosse section, à la base de la pyramide, est constituée des tests unitaires qui testent de "petites" unités de code. Plus précisément, ils testent que chaque fonctionnalité extraite de manière isolée se comporte comme attendu. Ils sont très rapides et faciles à exécuter.

De bons tests unitaires sont stables, c'est-à-dire que le code de ces tests n'a pas besoin d'être modifié,

même si le code de l'application change pour des raisons purement techniques. Ils deviennent donc rentables, car ils ont été écrits une seule fois, mais exécutés de nombreuses fois. Cependant, seules des unités individuelles de code sont testées, vous avez donc besoin d'autres tests permettant de s'assurer que ces unités de code fonctionnent entre elles.

Tests d'intégration

Au milieu de la pyramide se trouvent les tests d'intégration. Ils vérifient si vos unités de code fonctionnent ensemble comme prévu – en présupposant que vos tests unitaires soient passés ! Comme les tests d'intégration vérifient les interactions entre les unités, vous avez plus de certitude concernant le bon fonctionnement de l'application finale.

Ils peuvent nécessiter l'exécution de composants extérieurs (base de données, services web externe, etc.). Le lancement de ces composants et l'interaction entre vos unités de code développées rendent ces types de test plus lents et potentiellement moins stables. Mais on simule des scénarios plus proches de l'utilisation finale de l'application.

Tests fonctionnels

Enfin, au sommet de la pyramide, les tests fonctionnels (appelés end-to-end en anglais), visent à simuler le comportement d'un utilisateur final sur l'application, depuis l'interface utilisateur. L'ensemble du code développé est pris comme une boîte noire à tester, sans connaissance des briques et des unités de code qui la composent. Les simulations obtenues sont donc les plus proches de l'application utilisée dans des conditions réelles.

Ces tests nécessitent toute l'infrastructure nécessaire à l'application. Ces types de tests sont les plus lents à exécuter, et testent une partie beaucoup plus grande du code développé.

Cela crée une plus forte dépendance qui rend les tests moins stables, donc moins rentables. Potentiellement, une modification simple de l'interface utilisateur (la couleur d'un bouton) pourrait nécessiter de recoder le test fonctionnel associé.

3.1.6 Eclipse IDE

Eclipse IDE est un environnement de développement intégré libre extensible, universel et polyvalent, permettant potentiellement de créer des projets de développement (application web dans notre cas) mettant en œuvre le langage Java. La spécificité d'Eclipse IDE vient du fait de son architecture totalement développée autour de la notion de plug-in : toutes les fonctionnalités de cet atelier logiciel sont développées en tant que plug-in.



FIGURE 3.6 – Logo de Eclipse IDE



FIGURE 3.7 – Logo de Apache Tomcat

3.1.7 Apache Tomcat

Apache Tomcat est un conteneur web libre de servlets et JSP Java EE. Issu du projet Jakarta, c'est un des nombreux projets de l'Apache Software Foundation. Il implémente les spécifications des servlets et des JSP du Java Community Process, est paramétrable par des fichiers XML et des propriétés, et inclut des outils pour la configuration et la gestion. Il comporte également un serveur HTTP.

3.1.8 WampServer

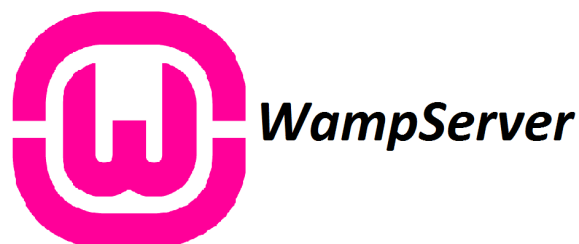


FIGURE 3.8 – Logo de WampServer

WampServer (anciennement WAMP5) est une plateforme de développement Web de type WAMP, permettant de faire fonctionner localement (sans avoir à se connecter à un serveur externe) des

scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant trois serveurs (Apache, MySQL et MariaDB), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL.

Il dispose d'une interface d'administration permettant de gérer et d'administrer ses serveurs au travers d'un tray icon (icône près de l'horloge de Windows).

La grande nouveauté de WampServer 3 réside dans la possibilité d'y installer et d'utiliser n'importe quelle version de PHP, Apache, MySQL ou MariaDB en un clic. Ainsi, chaque développeur peut reproduire fidèlement son serveur de production sur sa machine locale.

3.1.9 MySQL



FIGURE 3.9 – Logo de MySQL

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, PostgreSQL et Microsoft SQL Server.

Son nom vient du prénom de la fille du cocréateur Michael Widenius, "My". SQL fait référence au Structured Query Language, le langage de requête utilisé.

MySQL AB a été acheté le 16 janvier 2008 par Sun Microsystems pour un milliard de dollars américains. En 2009, Sun Microsystems a été acquis par Oracle Corporation, mettant entre les mains d'une même société les deux produits concurrents que sont Oracle Database et MySQL. Ce rachat a été autorisé par la Commission européenne le 21 janvier 2010.

3.1.10 HTML 5

L'HyperText Markup Language, généralement abrégé HTML, est le langage de balisage conçu pour représenter les pages web. C'est un langage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et logiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie et des programmes informatiques.



FIGURE 3.10 – Logo de HTML

3.1.11 CSS 3



FIGURE 3.11 – Logo de CSS

Les feuilles de style en cascade, généralement appelées CSS de l'anglais Cascading Style Sheets, forment un langage informatique qui décrit la présentation des documents HTML et XML. Les standards définissant CSS sont publiés par le World Wide Web Consortium (W3C). Introduit au milieu des années 1990, CSS devient couramment utilisé dans la conception de sites web et bien pris en charge par les navigateurs web dans les années 2000.

3.1.12 JavaScript

Le JavaScript est un langage de script basé sur la norme ECMAScript. Il s'insère dans le code (x)HTML d'une page web, et permet d'en augmenter le spectre des possibilités. Ce langage de POO, faiblement typé, est exécuté côté client.



FIGURE 3.12 – Logo de JavaScript

3.2 Scénario d'exécution

3.2.1 La page d'accueil

La page d'accueil offre aux entreprises la possibilité d'ajouter les offres d'emploi ou de stage, et aussi la possibilité aux chercheur d'emploi/stage de s'inscrire et se connecter afin d'envoyer les candidatures à ses offres et aussi de consulter les domaines ciblés

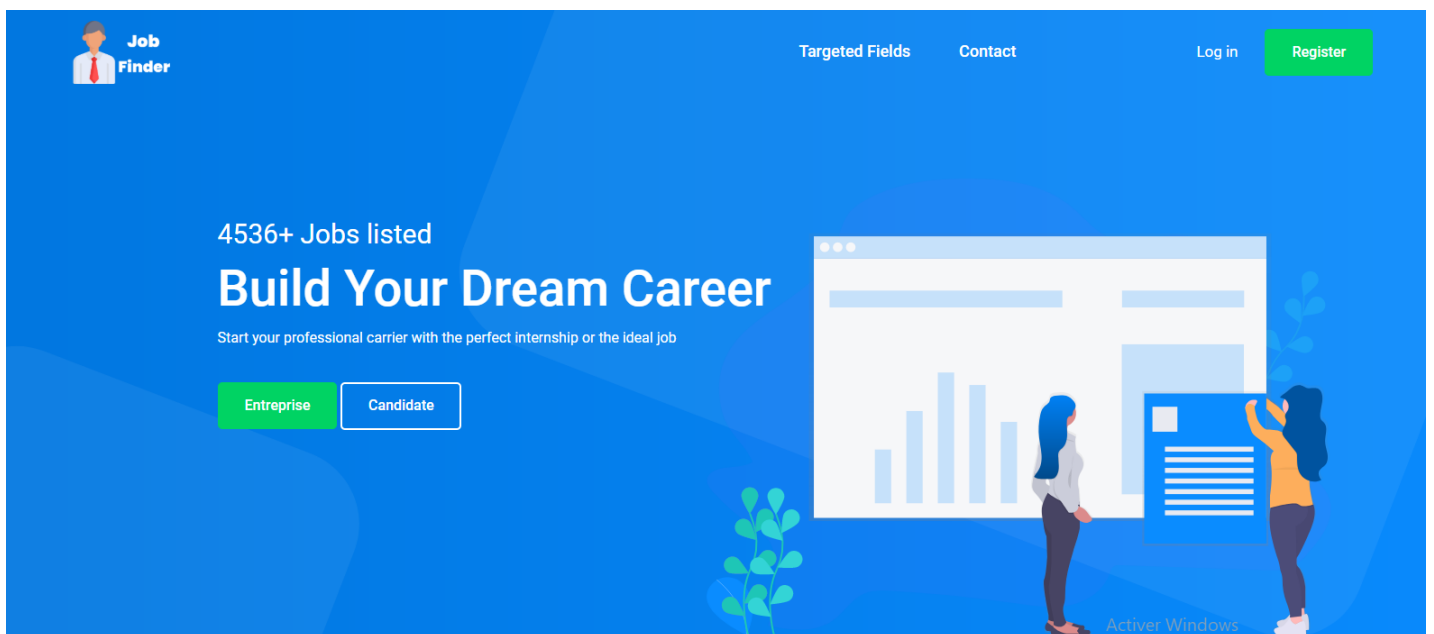


FIGURE 3.13 – La page d'accueil

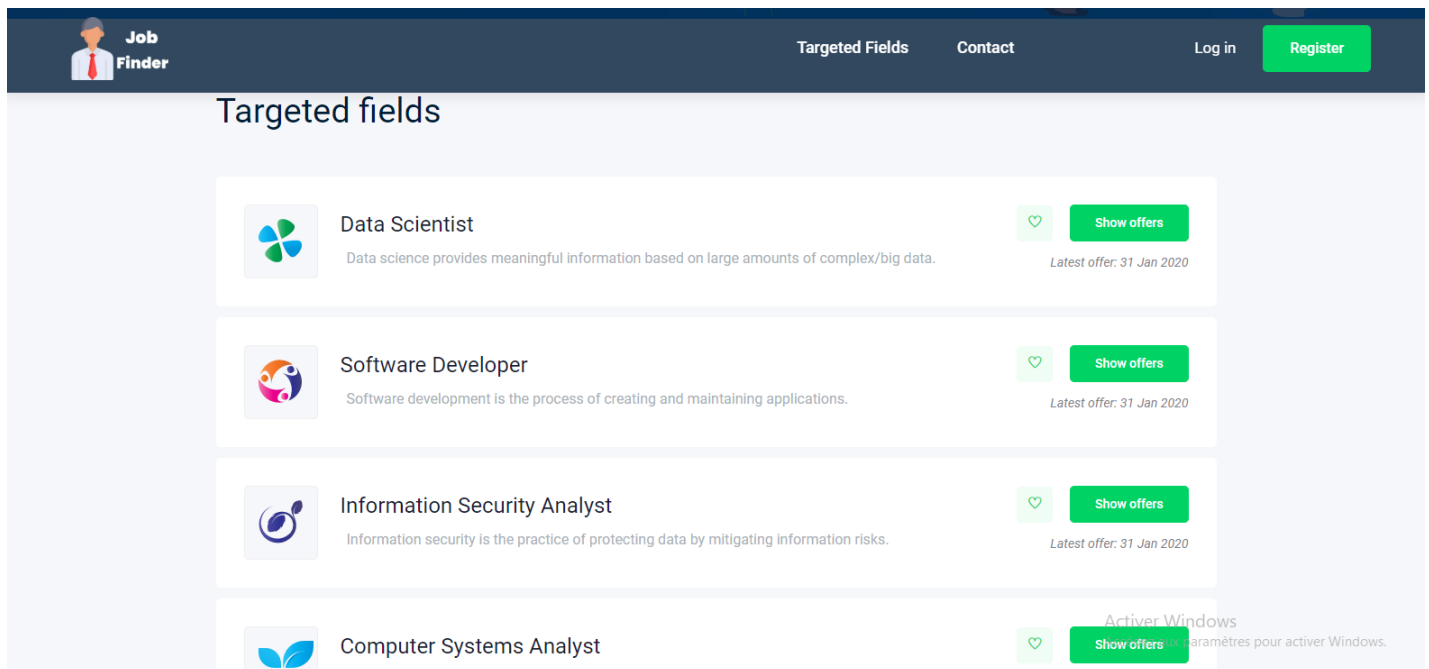


FIGURE 3.14 – Targeted Fields

3.2.2 Le dépôt des offres

Job Finder

Give us your informations

Company name
Your company

Company e-mail
Your company e-mail

Company address
Your company address

Website
Your Web Site (optional)

Tell us more about your project
Type your message here...

Select the wanted profile(s)

☐ Data Scientist
 ☐ Software Developer
 ☐ Information Security Analyst
 ☐ Computer Systems Analyst
 ☐ Web Developer
 ☐ Information Technology Manager

Select the type of project

☐ Job
 ☐ Internship

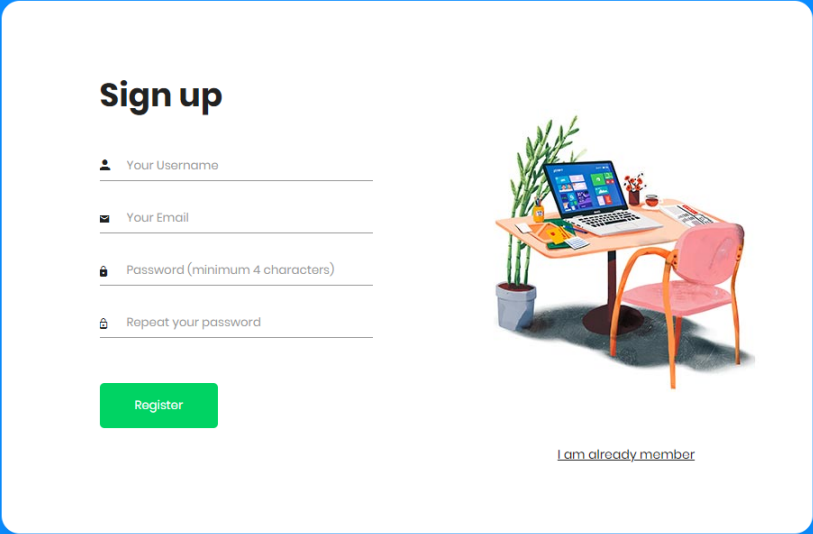
Salary/remuneration(per month)
Salary (optional)

Submit

Activer Windows
Accédez aux paramètres pour activer Windows.

FIGURE 3.15 – Le dépôt des offres

3.2.3 L'inscription et la connexion des candidats



The sign-up page features a white rounded rectangle on a blue background. On the left, the title "Sign up" is in bold. Below it are four input fields: "Your Username" (with a person icon), "Your Email" (with an envelope icon), "Password (minimum 4 characters)" (with a lock icon), and "Repeat your password" (with a lock icon). A green "Register" button is below these fields. On the right is an illustration of a desk with a laptop, a potted plant, and a pink chair. Below the illustration is a link "I am already member". At the bottom right of the blue background is the text "Activer Windows" and "Accédez aux paramètres pour activer Windows."

Sign up

Your Username

Your Email

Password (minimum 4 characters)

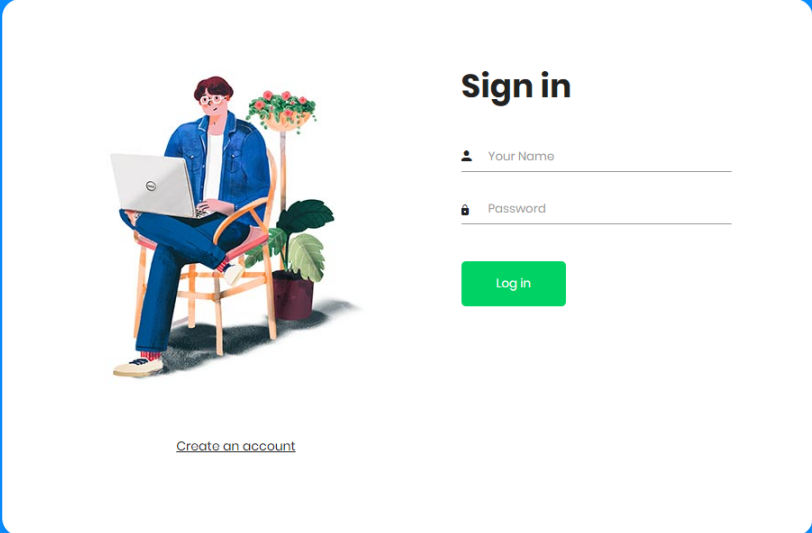
Repeat your password

Register

[I am already member](#)

Activer Windows
Accédez aux paramètres pour activer Windows.

FIGURE 3.16 – La page d'inscription



The sign-in page features a white rounded rectangle on a blue background. On the left is an illustration of a person sitting in a chair with a laptop and a potted plant. On the right, the title "Sign in" is in bold. Below it are two input fields: "Your Name" (with a person icon) and "Password" (with a lock icon). A green "Log in" button is below these fields. At the bottom left is a link "Create an account". At the bottom right of the blue background is the text "Activer Windows" and "Accédez aux paramètres pour activer Windows."

Sign in

Your Name

Password

Log in

[Create an account](#)

Activer Windows
Accédez aux paramètres pour activer Windows.

FIGURE 3.17 – La page de connexion

3.2.4 La consultation des offres

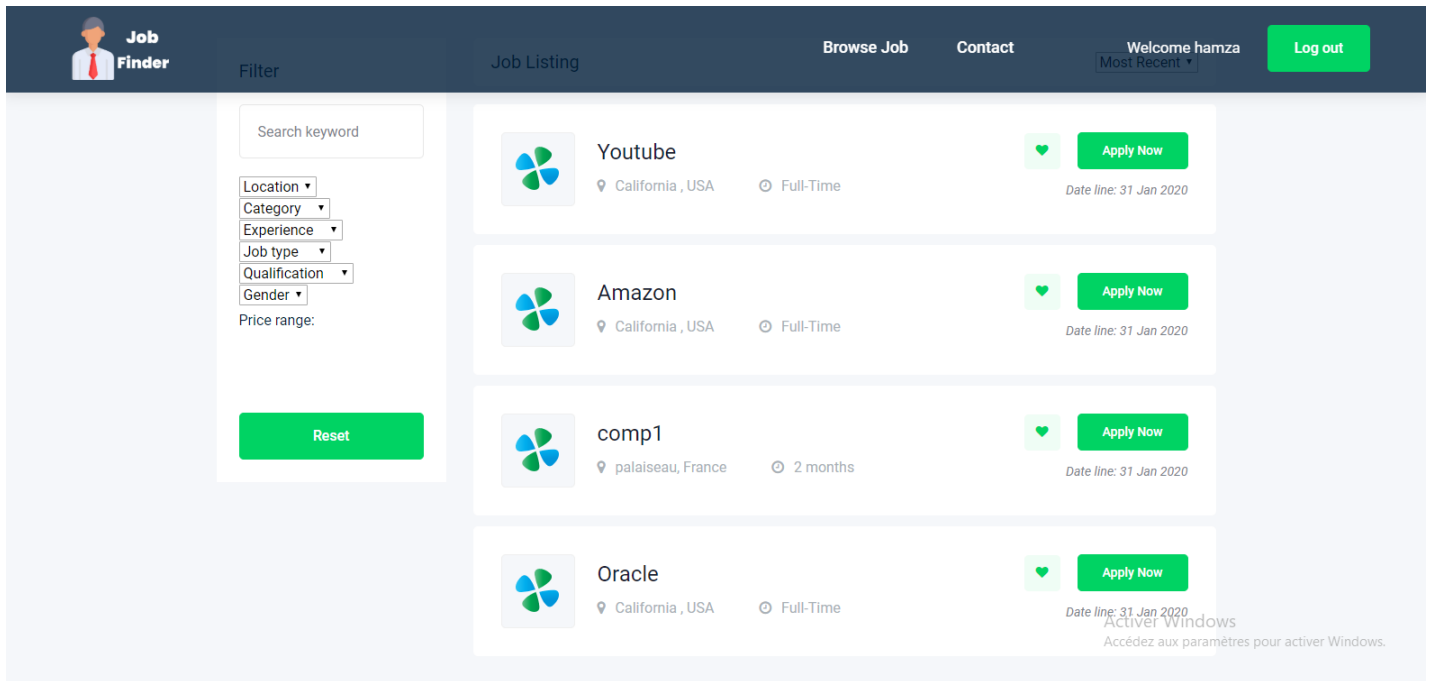


FIGURE 3.18 – Liste des offres

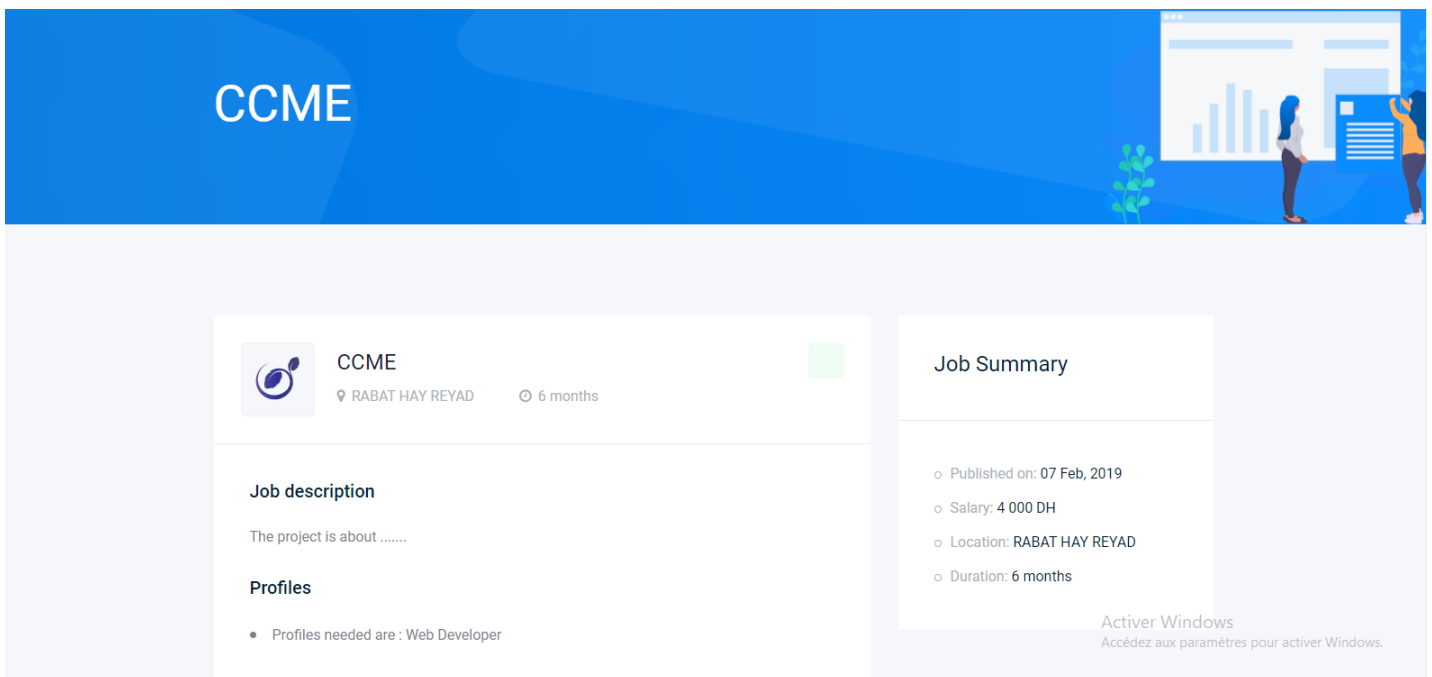
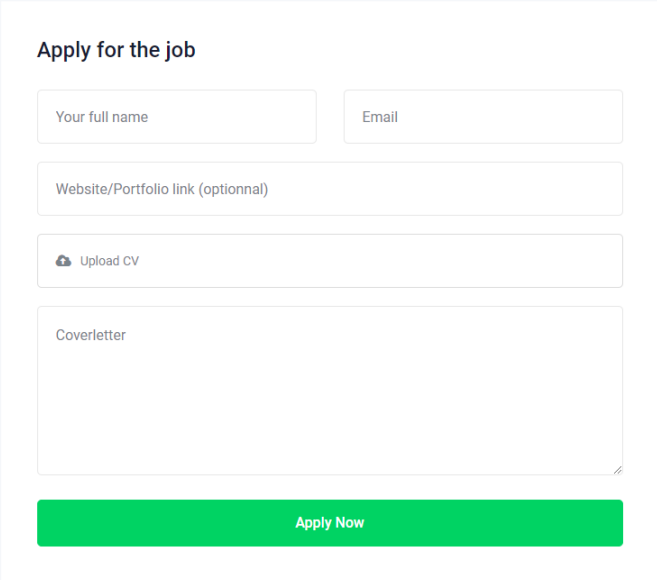


FIGURE 3.19 – Une page qui contient les informations d’une offre

3.2.5 L'envoi de candidature



The screenshot shows a web form for applying for a job. The form is titled "Apply for the job" and contains the following fields:

- A text input field for "Your full name".
- A text input field for "Email".
- A text input field for "Website/Portfolio link (optional)".
- A file upload button labeled "Upload CV" with a document icon.
- A large text area for a "Coverletter".
- A green "Apply Now" button at the bottom.

In the bottom right corner of the page, there is a Windows watermark that reads: "Activer Windows. Accédez aux paramètres pour activer Windows."

FIGURE 3.20 – L'envoi de candidature

3.3 Les tests

3.3.1 Structure de la class de test unitaire generer automatiquement

```
1 package managedBeans;
2
3 import static org.junit.jupiter.api.Assertions.*;
4
5 import org.junit.jupiter.api.Test;
6
7 class Testing {
8
9     @Test
10    void test() {
11        fail("Not yet implemented");
12    }
13
14 }
15
```

FIGURE 3.21 – la class de test unitaire

Ce fichier, généré par l'IDE, contient un squelette de test minimal :

- aux lignes 3 et 5, on importe les bibliothèques JUnit;
- à la ligne 9, l'annotation `@Test` indique que la méthode `test()` de `CalculatorTest` est un test;
- à la ligne 11, la méthode `fail()` fait échouer le test, en expliquant qu'il n'est pas implémenté.

Cette classe n'est qu'un squelette de tests, donc cette ligne permet de rappeler au développeur qu'il pense à le coder !

3.3.2 Structure de notre class AuthenticationTest

```
package managedBeans;

import static org.junit.jupiter.api.Assertions.*;
import org.junit.jupiter.api.Test;

class Testing {

    @Test
    public void testAuthenticationValide(){
        LoginController lg = new LoginController("valeurCorrect1","valeurCorrect2");
        String st = authenticateUser();
        assertEquals("SUCCESS", st);
    }

    @Test
    public void testAuthenticationInValide(){
        LoginController lg = new LoginController("valeurInCorrect1","valeurInCorrect2");
        String st = authenticateUser();
        assertEquals("FAILURE", st);
    }
}
```

FIGURE 3.22 – test unitaire d’authentification

dans cette classe on a essayé de faire tester le bon fonctionnement de l’authentification de l’utilisateur

- La premier fonction faire le test ou cas de succès par entrer des valeur qui figure dans notre bdd.
- La deuxième faire le test dans le cas d’échec par entrer des valeur invalide. on exécute cette classe en tant que jUnit Test afin de voir le résultat de notre test.

Conclusion

Notre application sert à minimiser le nombre des candidats ayants besoin d'un stage (ou d'un emploi) et des entreprises ayants besoin de recruter des profils spécifiques, nous espérons que notre application soit utile et répond réellement à notre problématique, il se peut qu'ils y auraient encore des erreurs à fixer ultérieurement, et on pense à ajouter des améliorations dans le futur comme par exemple la possibilité du chat entre l'entreprise et les candidats.

Bibliographie et Webographie

[1] : <https://openclassrooms.com/fr/courses/2434016-developpez-des-sites-web-avec-java-ee>

[2] : <https://stackoverflow.com/>

[3] : <https://www.w3schools.com/>

[4] : <https://www.planttext.com/>

[5] : <https://git-scm.com/docs/>