

الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

Université Dr. Tahar Moulay SAIDA

جامعة د. الطاهر مولاي سعيدة

Faculté : Technologie

كلية التكنولوجيا

Département : Informatique

قسم : الإعلام الآلي



MEMOIRE DE MASTER

Option : Sécurité informatique et Cryptographie

THEME

**Conception d'un Système d'aide pour la
gestion des offres d'emploi Sécurisé**

Présenté par :

Ouis Djamel Eddine

Encadré par :

Mme Meddah

Remerciements

Quelques lignes ne pourront jamais exprimer la reconnaissance que nous éprouvons envers tous ceux qui, de près ou de loin, ont contribué, par leurs conseils, leurs encouragements ou leurs amitiés à l'aboutissement de ce travail.

Mes vifs remerciements accompagnés de toute Ma gratitude vont tout d'abord à notre encadreur Madame Meddah. Pour nous avoir proposé ce sujet, pour les conseils qu'elle n'a cessé de nous prodiguer et surtout pour la confiance qu'elle m'a accordé pour la réalisation de ce projet. Ma reconnaissance va à tous nos enseignants à l'UTMS, en général.

Dédicace

Grace à Dieu voilà notre travail terminé et il est temps pour moi de partager ma joie avec tous ceux qui m'ont soutenu et encouragé. A vous, ma mère et mon père qui avez consacré votre vie à notre éducation et à faire notre bonheur. Sans oublier mes Grands Parents.

A mon frère Lahcen, A mes sœurs Imane et son marié khelífa, Soria , Dalíla.

A mes oncles fodil, mohamed , abdelkrim et sa femme .

A mes tantes yamina, fatíma, kheíra.

A l'ensemble des amis que j'ai connu pendant mes études et à ceux qui ont prodigué leurs vifs conseils, encouragements et témoigné de leur amitié .

Djamel eddine.

ملخص

في هذه الأيام، ارتفعت نسبة البطالة وعدم وجود مناصب شغل كافية، لها عدة أسباب، فبظهور تكنولوجيات الحديثة والانترنت، وجب توفير بيئة آمنة تساعد على حل هذه المشاكل.

في هذا المشروع، سنطرح حلول آمنة لتسهيل البحث عن وظيفة للمتقدمين للعمل وسماع لأرباب العمل بإيجاد موظفين في أقرب وقت ممكن.

Abstract

In this days, The unemployment rate has increased and the availability of insufficient job posts has several reasons. But, the emergence of various modern technologies and the Internet, can provide a secure environment, which can help solve these problems .

In this work, we propose secure solutions to facilitate the job search for candidates and enable recruiters to find the profiles sought in the shortest possible time.

Résumé

De nos jours, Le taux de chômage a augmenté Et la disponibilité de postes d'emplois insuffisants, a plusieurs raisons .Mais, l'émergence des différentes technologies modernes et l'Internet, peuvent fournir un environnement sûr, pouvant aider à résoudre ces problèmes .

Dans ce travail, nous proposons des solutions sécurisés pour faciliter la recherche d'emploi aux postulants et permettre aux recruteurs de trouver les profils recherchés dans les plus brefs délais.

Table des matières

Introduction Générale	1
1 Les applications WEB	3
1.1 Introduction	3
1.1.1 Définition d'une application WEB	3
1.2 PRESENTATION DE CLIENT /SERVEUR	3
1.2.1 Définition	3
1.2.2 Le client serveur pour web	3
1.2.3 Fonctionnement d'un système client/serveur	4
1.2.4 L'architecture client/serveur	4
1.2.5 Présentation de l'architecture à 2 niveaux	4
1.2.6 Présentation de l'architecture à 3 niveaux	5
1.2.7 Présentation de l'architecture à N niveaux	6
1.3 Internet :	6
1.3.1 Le réseau Internet et ses protocoles	6
1.3.2 Évolution des architectures applicatives	8
1.3.3 Web 2.0	9
1.4 Composants du client Web	9
1.4.1 Le navigateur	9
1.5 Composants serveur	10
1.5.1 Serveurs Web et serveurs d'application	10
1.6 CONCLUSION	10
2 Sécurité des applications WEB	11
2.1 Introduction	11
2.2 Attaques	11
2.2.1 Les attaques d'un système d'information visent :	11
2.3 Types d'Attaques	11
2.3.1 Authentification	12
2.3.2 autorisation	12
2.3.3 Attaques côté client	12
2.3.4 Exécution de commandes	12
2.3.5 Révélation d'informations	12
2.3.6 Logiques	12
2.4 Attaques liées à l'authentification	13
2.5 Attaques liées aux autorisations	14
2.6 Attaques côté client	15
2.7 Attaques par exécution de commandes ou de requêtes	15
2.8 Attaques liées à la révélation d'informations	16
2.9 Attaques logiques	16

2.10	Les solutions	17
2.10.1	Solution faille XSS	17
2.10.2	Solution Injections	17
2.10.3	Détournement de session	18
2.11	Conclusion	20
3	Analyse et Conception	21
3.1	Introduction	21
3.2	Présentation de L'UML :	21
3.2.1	Le langage de modélisation UML :	21
3.3	Définition des besoins :	22
3.3.1	Analyse fonctionnelle	22
	SPECIFICATIONS :	22
	Identification des acteurs du système :	23
3.3.2	DIAGRAMME DE CAS D'UTILISATION :	23
3.3.3	DIAGRAMME DE CLASSE :	25
4	Implémentation	27
4.1	Introduction	27
4.2	Outils de réalisation :	27
4.2.1	PHP :	27
	Qu'est ce PHP	27
	Les Bonnes raisons pour utiliser PHP	27
4.2.2	MySQL	28
	Les Bonnes raisons pour utiliser MySQL	28
4.2.3	XAMPP (MySQL, Apache PHP et Perl) :	31
4.2.4	Editeur NOTEPAD++ :	31
4.3	Réalisation de l'application Web :	32
4.3.1	Les interfaces de l'application :	32
4.4	Sécuriser l'application Web :	48
4.4.1	La sécurité de la base de données :	48
4.4.2	Le chiffrement des documents Cv :	49
	L'algorithme du DES (Data Encryption Standard) :	50
	Génération des clés :	52
	Triple DES (Triple Data Encryption Standard) :	52
	Modes de chiffrement :	53
4.4.3	L'application d'une attaque SQL injection sur notre systeme :	54
	L'attaque sur le formulaire d'une connexion :	54
4.4.4	L'application d'une attaque xss (Cross-Site Scripting) sur notre systeme :	55
4.4.5	L'attaque par force brute :	55
4.4.6	La sécurité contre les attaques :	56
4.5	Conclusion :	57
	Conclusion Générale	59
	Bibliographie	61

Table des figures

1.1	Fonctionnement de system client / serveur.	4
1.2	l'architecture à 2 niveaux.	5
1.3	l'architecture à 3 niveaux.	5
1.4	l'architecture à N niveaux.	6
1.5	Transfert des données à travers la pile de protocoles d'Internet. . .	7
1.6	Pile de protocoles d'Internet.	8
1.7	Mode de fonctionnement des applications Web.	9
3.1	Diagramme de cas d'utilisation «Administrateur ».	23
3.2	Diagramme de cas d'utilisation «Recruteur».	24
3.3	Diagramme de cas d'utilisation «Postulant».	25
3.4	Diagramme de «classe».	26
4.1	Interface Xampp.	31
4.2	Interface Notepad++.	32
4.3	Interface d'accueil.	33
4.4	Interface d'enregistrement.	34
4.5	Interface d'enregistrement postulant.	35
4.6	Interface d'enregistrement recruteur.	36
4.7	Interface de connexion.	37
4.8	Interface par domaine.	38
4.9	Interface géographique.	39
4.10	Interface offre.	40
4.11	Boutton postule.	40
4.12	Interface gestion d'utilisateurs.	41
4.13	Interface gestion des offres.	42
4.14	Interface gestion des CV.	42
4.15	Interface Ajouter des offres.	43
4.16	Interface d'Affichage des offres et les postulants.	44
4.17	Interface information.	45
4.18	Interface d'affichage les cv.	46
4.19	Interface choix de postulant.	47
4.20	Test temps d'exécution.	49
4.21	Algorithme de DES.	51
4.22	Génération des clés.	52
4.23	Triple DES.	53
4.24	Mode CBC.	54
4.25	L'application d'une attaque SQL.	55
4.26	L'application d'une attaque xss.	55
4.27	Le bannissement d'IP.	56
4.28	Message d'alerte correspond à la détection.	56

Introduction Générale

L'informatique et en particulier Internet représente la révolution la plus importante et la plus innovante qui a marqué la vie de l'humanité ces dernières décennies. En effet, Internet est devenu un outil indispensable pour le fonctionnement des entreprises à travers le monde, car il a accru les chances de communication à travers des sites web. Aucun domaine n'est resté étranger à cette stratégie qui offre tant de services aussi bien pour l'administration ou les autorités gouvernementales que pour les personnes.

Problématique

En général, les méthodes classiques de recrutement consistent à consulter les agences de recrutement locales. Le recrutement à travers ces méthodes est loin de satisfaire les besoins des employeurs et des employés. Les employés qualifiés n'étant pas toujours informés sur les offres disponibles sur le marché du travail. Il est nécessaire de trouver une méthode rapide et efficace d'où l'extension du recrutement à travers le net. Les acteurs du marché des offres d'emploi ont compris les avantages compétitifs qu'Internet pouvait leur apporter. Un site d'emploi, au même titre qu'un journal, permet d'afficher des offres d'emploi, mais en plus, il donne la possibilité de les actualiser en temps réel et d'en assurer le suivi. Un chercheur d'emploi peut consulter les offres en ligne et déposer immédiatement sa demande en ligne. Ces raisons ont conduit les premiers acteurs du recrutement en ligne à créer des sites d'emploi pour exposer le plus grand nombre d'offres possibles.

L'Internet est une amélioration des méthodes traditionnelles dans le sens, qu'il n'est pas géographiquement limité, son accès rapide et immédiat permet d'optimiser le processus de recrutement, la qualité des recherches est maximisée grâce à la diffusion via l'Internet, un médium de haute technologie.

Le recrutement à travers les sites d'emploi en ligne permet d'offrir simultanément des avantages aux employeurs et aux employés. Pour les Recruteurs, cette méthode garantit une visibilité optimale parmi les personnes correspondant au profil recherché par le recruteur, elle permet également une diffusion des annonces à moindre coût. Pour les chercheurs d'emploi, elle les oriente facilement vers les emplois répondant à leurs aspirations et à leurs attentes du fait que ces sites d'emplois regroupent sous un même toit des chercheurs et des pourvoyeurs d'emplois, ils constituent des outils de choix non seulement pour les recruteurs à la recherche des employés qualifiés, mais également pour les chercheurs d'emploi qui désirent être connus par ces employeurs.

D'un autre côté bien que les agences en ligne soient hautement performantes, des problèmes liés à la sécurité des données perturbent leur bon fonctionnement d'où la nécessité de mettre en place un système très sécurisé en utilisant des services, des mécanismes, des outils et des procédures qui présentent des solutions et des mesures de sécurité.

C'est dans ce cadre d'idées que s'inscrit notre projet de fin d'études. L'objectif ciblé est la conception et réalisation d'un système de gestion d'offres d'emploi sécurisé .

Plan du Mémoire :

Le mémoire est organisé comme suit :

- 1) Une introduction générale déjà défini ci-dessus .
 - 2) Chapitre I : Application web
 - 3) Chapitre II : Sécurité des applications web
 - 4) Chapitre III : Analyse et Conception
 - 5) Chapitre IIII : Implémentation
- Enfin, nous clôturons par une conclusion générale .

Chapitre 1

Les applications WEB

1.1 Introduction

Le développement des applications WEB présente certaines particularités, au niveau technique et ergonomique. Cette spécificité nous oblige, au moment de la conception, à préconiser des méthodes de conception et des méthodes de travail dédiées à ce genre d'applications.

1.1.1 Définition d'une application WEB

Une application Web est un ensemble de pages qui interagissent avec les utilisateurs, les unes avec les autres, ainsi qu'avec les différentes ressources d'un serveur Web, notamment les bases de données. Une application Web est un site Web qui contient des pages et dont le contenu est partiellement ou totalement indéterminé. Le contenu final d'une page est déterminé uniquement lorsque l'utilisateur requiert une page depuis le serveur Web. Il variant d'une requête à une autre en fonction des actions de l'utilisateur, ce type de page est appelé page dynamique. Les applications Web sont construites de manière à répondre à différents types de défis et de problèmes.

1.2 PRESENTATION DE CLIENT /SERVEUR

Le mode client/serveur est un mode de fonctionnement dissymétrique dans lequel deux logiciels différents sont nécessaires pour permettre les communications : un logiciel serveur et un logiciel client, nécessaires sur toute machine.

1.2.1 Définition

Un environnement client/serveur désigne un mode de communication à travers un réseau informatique entre plusieurs logiciels. Un client et un serveur sont reliés par un réseau informatique. Le client peut envoyer une requête au serveur.

1.2.2 Le client serveur pour web

Dans les applications web, le client est le navigateur, de tels logiciels existent pour tous les systèmes d'exploitation : Les applications web sont hébergées sur des serveurs dédiés qui sont nommés serveurs web.

Le navigateur émet une requête http vers un serveur web afin d'obtenir la page web désirée. Le serveur envoie les données demandées par le client ; si celui-ci est autorisé à accéder au document. Le navigateur interprète les instructions de mise en page contenus dans les données envoyées par le serveur.

1.2.3 Fonctionnement d'un système client/serveur

Un system client / serveur fonctionne selon le schéma suivant :

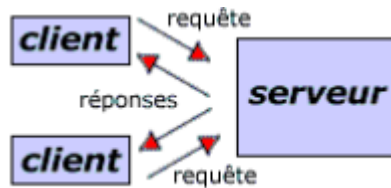


FIGURE 1.1 – Fonctionnement de system client / serveur.

Le client émet une requête vers le serveur grâce à son adresse IP et le port ; qui désigne un service particulier du serveur. Le serveur reçoit la demande et répond à l'aide de l'adresse de la machine client et son port

1.2.4 L'architecture client/serveur

De nombreuses applications fonctionnent selon un environnement client/serveur. Cela signifie que des machines clients (des machines faisant partie du réseau) contactent un serveur, une machine généralement très puissante en terme de capacité d'entrée-sortie, qui leur fournit des services. Ces services sont des programmes fournissant des données-t-elle que l'heure ; des fichiers, une connexion, etc. Les services sont exploités par des programmes, appelés programmes client, s'exécutant sur les machines clients. On parle ainsi de client (client FTP, client de messagerie, etc.) lorsque l'on désigne un programme tournants sur une machine cliente, capable de traiter des informations qu'il récupère auprès du serveur (dans le cas du client FTP il s'agit de fichier ; tandis que pour le client de messagerie il s'agit de courrier électronique).

1.2.5 Présentation de l'architecture à 2 niveaux

L'architecture à deux niveaux (aussi appelée architecture 2-tier, tier signifiant rangée en anglais) caractérise les systèmes clients/serveurs pour lesquels le client demande une ressource et le serveur la lui fournit directement, en utilisant ses propres ressources. Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir une partie du service.

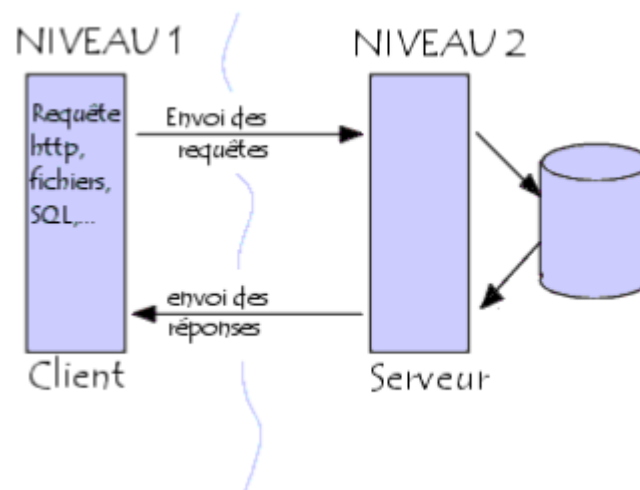


FIGURE 1.2 – l’architecture à 2 niveaux.

1.2.6 Présentation de l’architecture à 3 niveaux

Dans l’architecture à 3 niveaux (appelée architecture 3-tier), il existe un niveau intermédiaire, c’est-à-dire que l’on a généralement une architecture partagée entre : Un client, c’est-à-dire l’ordinateur demandeur des ressources, équipées d’une interface utilisateur (généralement un navigateur web) chargée de la présentation ; Le serveur d’application (appelé également middleware), chargé de fournir la ressource mais faisant appel à un autre serveur Le serveur des données, fournissant au serveur d’application les données dont il a besoin.

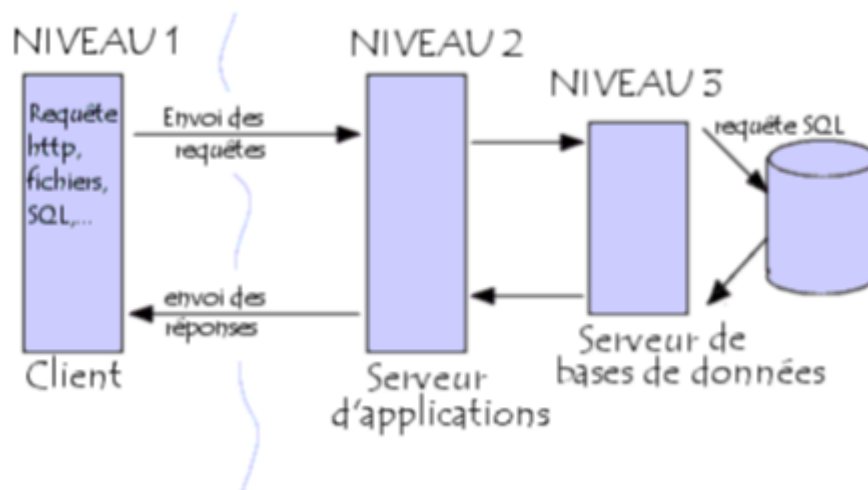


FIGURE 1.3 – l’architecture à 3 niveaux.

Etant donné l’emploi massif du terme d’architecture à 3 niveaux, celui-ci peut parfois désigner aussi les architectures suivantes : Partage l’application entre client,

serveur intermédiaire, et serveur d'entreprise ; Partage l'application entre client, serveur d'application, et serveur de données d'entreprise.

1.2.7 Présentation de l'architecture à N niveaux

L'architecture 3 niveaux permet de spécialiser les serveurs dans une tâche précise : avantage de flexibilité, de sécurité et de performance. L'architecture peut être étendue sur un nombre de niveaux plus important : on parle dans ce cas d'architecture à N niveaux (ou multi-tier).

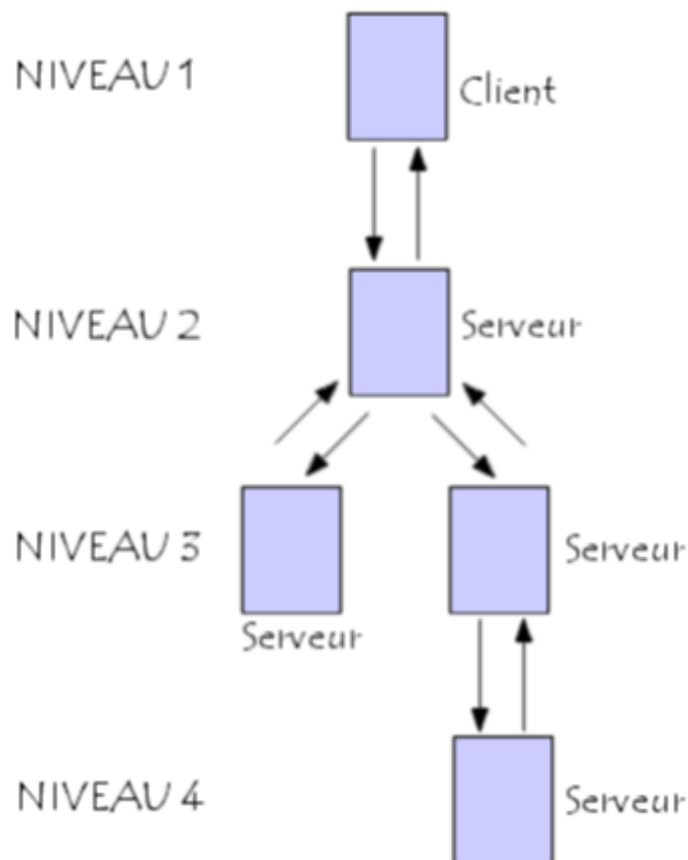


FIGURE 1.4 – l'architecture à N niveaux.

1.3 Internet :

1.3.1 Le réseau Internet et ses protocoles

Le Web repose sur le réseau Internet et comme tous les réseaux informatiques, celui-ci repose sur des couches de protocoles de communication. Le paquet est l'unité de base de la transmission de données sur Internet.[5] L. Shklar et R. Rosen [11] font la description suivante de la couche de protocoles pour Internet dont le couple TCP/IP est la fondation :

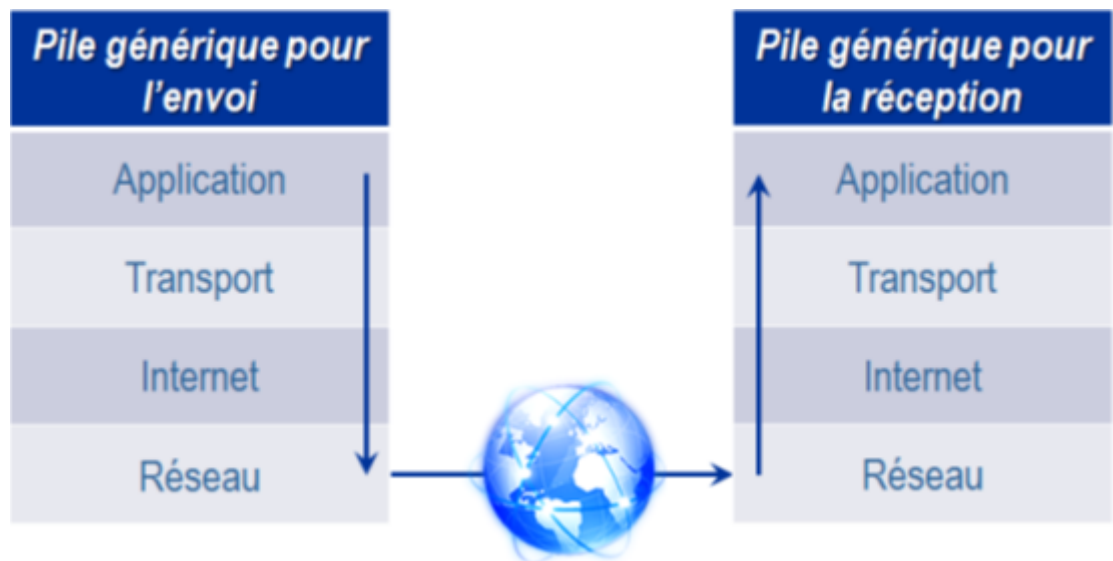


FIGURE 1.5 – Transfert des données à travers la pile de protocoles d'Internet.

-La couche « Réseau » est la couche responsable de la transmission physique des données. L'information peut cheminer sur différents supports avant d'atteindre la destination.

-La couche « Internet » est la couche qui indique où les données doivent être envoyées, sans garantie que la destination sera bien atteinte. Elle peut utiliser les protocoles IP (Internet Protocol) et ICMP (Internet Control Message Protocol).[4]

-La couche « Transport » repose sur deux protocoles : TCP et UDP (User Datagram Protocol). TCP s'assure que les paquets sont reçus dans le même ordre qu'ils ont été envoyés et que les paquets perdus sont à nouveau envoyés. TCP est donc un moyen de transmission fiable puisqu'il s'assure que les paquets sont arrivés. Comme indiqué par G.Florin et S.Natkin [13], UDP est un protocole simplifié. Cela permet de transmettre des informations plus rapidement qu'avec TCP puisqu'il y a finalement moins d'informations échangées. UDP est utilisé notamment par NFS (Network File System) et les applications de streaming audio et vidéo telles que la vidéoconférence et la téléphonie sur IP où la perte de paquets est acceptable et la vitesse de communication primordiale.

-La couche « Application » est celle qui permet aux utilisateurs finaux de communiquer sur Internet avec des protocoles tels que Telnet, pour agir sur un serveur : FTP (File Transfer Protocol) pour la transmission de fichiers, SMTP (Simple Mail Transfer Protocol) pour l'envoi de courrier électronique et HTTP pour le Web.

-HTTP est un protocole de messages de type texte, basé sur le paradigme « requête/réponse ». L'utilisateur envoie via son navigateur un message, la requête, au serveur HTTP. Chaque requête est traitée individuellement et de façon unique. Ensuite le serveur renvoie un message, la réponse, au navigateur. HTTP est un protocole déconnecté, c'est-à-dire que le protocole ne permet pas d'établir des communications entre requêtes pour partager des informations, alors qu'une application Web a besoin de conserver les réponses des différentes requêtes d'un

utilisateur pour avoir le même comportement qu'avec une application non Web. C'est pourquoi la plupart des navigateurs intègrent le système de « cookie » qui permet de conserver le résultat d'une requête.

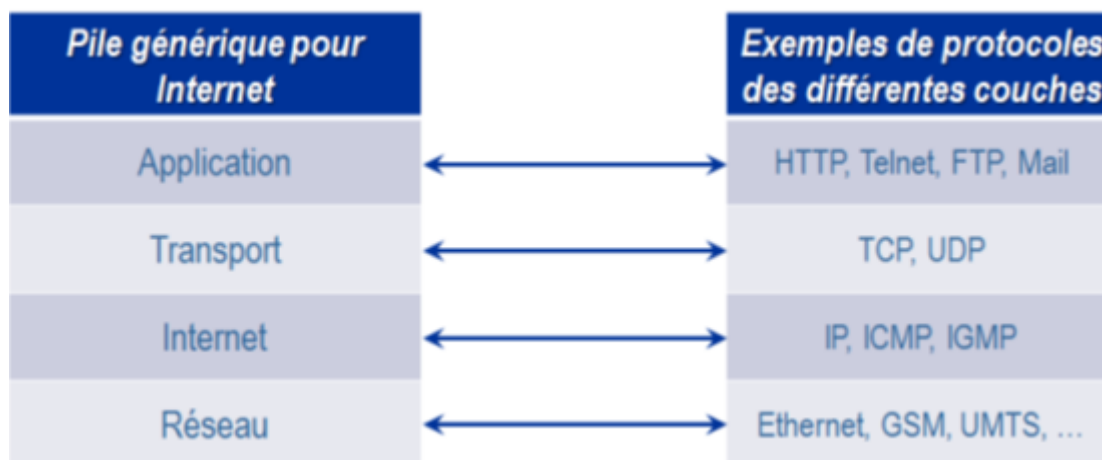


FIGURE 1.6 – Pile de protocoles d'Internet.

1.3.2 Évolution des architectures applicatives

- La décennie 1970-1980 était dominée par le système Mainframe. Un serveur centralisait l'ensemble des informations, exécutait les traitements, gérant les droits d'accès. Le client manipulé par l'utilisateur permettait d'envoyer des demandes de traitement au serveur et d'en afficher les résultats. La machine était passive. Ce mode de fonctionnement est le même que celui du protocole HTTP.

- La décennie suivante 1980-1990 a vu l'émergence du système client/serveur. Le client récupérait des données depuis le serveur de base de données, exécutait les traitements puis affichait les informations à l'écran et enfin mettait à jour les données sur le serveur si nécessaire. Le serveur ne servait plus qu'à stocker les informations et à exécuter éventuellement différées. Cette architecture posait des problèmes de maintenance des applications sur chacun des postes utilisateur concernés. Les applications Web ont suivi cette évolution avec les « applets » au début des années 90. Il s'agissait d'applications écrites dans un langage de développement, exécutées par le navigateur depuis un site Web.

- À partir des années 90, les architectures étaient composées de plusieurs tiers. L'application cliente présentait alors les informations à l'utilisateur et invoquait des services. Les services étaient responsables de l'exécution des processus. Les processus pouvaient être distribués sur plusieurs serveurs. Enfin des serveurs étaient responsables du stockage des données. Dans le milieu des années 90, les applications Web ont également intégré plusieurs composants. Le navigateur Web ne s'occupe plus que de l'affichage. Le serveur HTTP pour répondre aux requêtes génère dynamiquement l'interface graphique dans les pages HTML en faisant appel à des services ou en interrogeant les bases de données. Depuis les années 2000 les applications Web et les autres types d'applications clientes peuvent

utiliser les mêmes services, ce qui facilite la réutilisation des développements et évite la redondance des données.

1.3.3 Web 2.0

le Web 2.0 [3] n'est pas une mise à jour technique mais un changement de comportement des internautes. Comme évoqué précédemment, le Web avait pour but initial de mettre à disposition des informations. L'utilisateur était passif face aux sites Web. Puis le Web est devenu collaboratif, l'utilisateur est alors devenu créateur de contenu sans avoir à connaître les protocoles techniques sous-jacents. L'internaute ne consulte plus l'information, il publie du contenu quel que soit le média (texte, vidéo, musique). Internet a permis de mettre en relation des ordinateurs et est devenu le support du Web qui a permis d'y mettre à disposition des informations. À son tour le Web est devenu le support du Web 2.0 qui a permis de mettre en relation des personnes.

Ce nouveau comportement a pu naître grâce à la possibilité de modifier l'interface graphique sans recharger complètement la page Web. Elle devient en réalité un conteneur dans lequel il est possible de mettre à jour et de différencier le contenu, les fonctions, selon la zone de la page.

1.4 Composants du client Web

1.4.1 Le navigateur

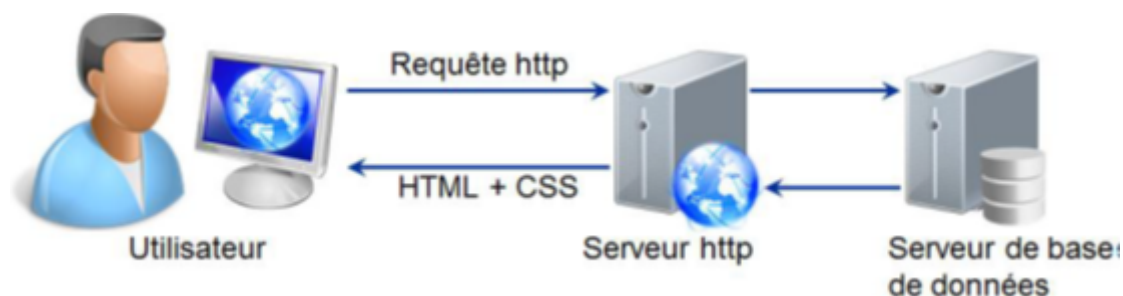


FIGURE 1.7 – Mode de fonctionnement des applications Web.

Dans les architectures citées précédemment, le navigateur est une application cliente. Il permet d'envoyer des requêtes http au serveur Web et d'en interpréter la réponse. Les navigateurs sont aujourd'hui capables de travailler également avec le protocole FTP et d'afficher d'autres formats tels que XML. Il existe plusieurs méthodes http pour envoyer des requêtes au serveur. Les plus répandues sont GET, HEAD et POST. GET permet de demander le téléchargement du contenu d'un document, HEAD permet de n'en récupérer que l'en-tête et POST permet d'envoyer des informations au serveur pour traitement. Lorsque l'utilisateur saisit une adresse ou clique sur un lien hypertexte, le navigateur envoie une requête GET au serveur qui ne comprend qu'un en-tête. Les requêtes POST ont un corps de message qui comporte l'ensemble des informations saisies dans un

formulaire, alors qu'avec GET, ces informations sont transmises en ajoutant des paramètres à l'adresse.

1.5 Composants serveur

1.5.1 Serveurs Web et serveurs d'application

Comme évoqué précédemment, le navigateur et le serveur communiquent en utilisant le protocole http. Les serveurs ne sont pas obligés d'implémenter toutes les méthodes http, seulement GET et HEAD. Bien qu'optionnelle, peu de serveurs actuels n'implémentent pas la méthode POST.

Sur Internet le navigateur et le serveur http communiquent rarement directement. Le plus souvent un serveur intermédiaire est présent : le serveur proxy. Les requêtes à destination du serveur sont interceptées par le serveur proxy qui peut leur faire subir un traitement, avant de les retransmettre au serveur. Ce principe est également appliqué aux réponses. Le serveur proxy peut servir de cache pour moins solliciter le serveur http. Il est possible de faire agir plusieurs serveurs proxy en cascade.

La seule fonction du serveur Web étant d'envoyer le contenu des fichiers au client, des extensions peuvent y être ajoutées, permettant de faire appel à des services pour générer dynamiquement les informations à transmettre. Ils traitent les requêtes http que le serveur http leur a fait suivre, interprètent et exécutent le code de l'application, puis génèrent une réponse qu'ils renvoient au serveur http qui l'envoiera au navigateur de l'utilisateur. Si ces services fonctionnent indépendamment du serveur http, ils sont appelés serveurs d'applications. Les services Web sont des applications Web dont le but est de fournir des données selon une structure prédéfinie et des services à une autre application en utilisant les protocoles standards d'Internet.

1.6 CONCLUSION

Dans ce chapitre nous avons présenté l'Internet et le Web sont deux concepts à tel point liée que la confusion règne parfois chez les nouveaux utilisateurs. L'Internet a pris l'ampleur et relie aujourd'hui plusieurs millions de machines fonctionnent sur une architecture client-serveur qui doit utiliser le même protocole de communication TCP/IP.

Chapitre 2

Sécurité des applications WEB

2.1 Introduction

La sécurité des applications informatique et des bases de données en particulier est devenue une priorité pour les citoyens ainsi que pour les administrations. Le besoin de partager et d'analyser des données personnelles est multiple : pour rendre plus simples et efficaces les procédures administratives et pour personnaliser les services rendus par une grande quantité d'objets électroniques dans un environnement d'intelligence ambiante.

2.2 Attaques

2.2.1 Les attaques d'un système d'information visent :

- **L'intégrité des données** : modification des données publiées sur le site (défaucement : nuit à l'image de marque); modification ou suppression d'informations confidentielles.

- **La confidentialité** : Obtention de données sur :

- le client (n sécurité sociale, carte bancaire, coordonnées, ...);
- les visiteurs du serveur Web (logs);
- l'organisation (accès à des données sensibles, au réseau local de l'organisation);
- le serveur (accès aux mots de passes, aux fichiers de configuration en vue d'une attaque).

- **La disponibilité des données** :

- bloquer l'accès au site Web (DoS);ou à un utilisateur en particulier.
- L'attaque d'un serveur web peut également être destinée à prendre le contrôle du serveur pour attaquer d'autres sites ou installer des services.

2.3 Types d'Attaques

Nous reprenons dans ce chapitre la classification des types de menaces définie par le WASC (Web Application Security Consortium) [15].

2.3.1 Authentification

- a. force brute
- b. authentification insuffisante
- c. mauvais traitement des recouvrements de mot de passe

2.3.2 autorisation

- a. prédiction de session
- b. autorisation insuffisante
- c. expiration de session insuffisante
- d. fixation d'identifiant de session

2.3.3 Attaques côté client

- a. usurpation de contenu (content spoofing)
- b. XSS

2.3.4 Exécution de commandes

- a. buffer overflow
- b. format string
- c. injection LDAP
- d. OS Commanding
- e. injection SQL
- f. injection SSI
- g. injection XPath

2.3.5 Révélation d'informations

- a. listing de répertoires
- b. fuite d'informations
- c. traversée de chemin
- d. prédiction de localisation de ressources

2.3.6 Logiques

- a. abus de fonctionnalité
- b. déni de service
- c. anti-automatisation insuffisante
- d. validation insuffisante du flux logique de l'application

2.4 Attaques liées à l'authentification

Le but de ces attaques est l'accès à une application Web protégée.

a) Force brute

Emploi d'un processus automatique pour trouver les informations protégeant un système (login, mot de passe). Généralement le pirate cherche un mot de passe pour un login fixé.

Protection :

- limiter le nombre d'essais lors de l'authentification (bloquer l'accès pendant 15mn après 3 échecs) et répondre après un délai de 3 secondes lors du 1er essai, 15 lors du 2d et 30 pour le 3ème;
- conserver des traces de toutes les tentatives de connexion (aide à la détection de ce type d'attaque);
- imposer une taille minimale pour le login et le mot de passe; en interne :
- imposer une complexité minimale (nombre de chiffres, car spéciaux, ...);
- ne jamais indiquer si c'est le login ou le mot de passe qui est erroné;
- ne jamais conserver les comptes avec un login et mot de passe par défaut.

b) Authentification insuffisante

Certaines applications insuffisamment protégées permettent l'accès à des ressources à des personnes non authentifiées :

- intranet protégé par l'obscurité, il est possible d'accéder à l'intranet :
- depuis un listing de répertoire s'il n'y a pas de fichier index.html;
- par une attaque de force brute recherchant les noms de répertoires d'administration les plus courants (/admin, /administrateur, /intranet, /backoffice, ...); en interne :
- depuis un bookmark ou l'historique de navigation sur un ordinateur en accès libre; en notant l'URL affiché dans le navigateur lorsque l'administrateur est connecté.
- intranet dont seule la page principale (index.php) demande le login/mot de passe .

Protection :

Utiliser un .htaccess (protection du répertoire et de ses sous-répertoires) ou des sessions.

c) Mauvais traitement des recouvrements de mot de passe

Une application Web doit gérer le recouvrement des mots de passe des utilisateurs.

La méthode la plus simple et la plus sûre serait de réaliser une nouvelle inscription mais les données de l'ancien login seraient perdues. Pour recouvrer le mot de passe plusieurs méthodes sont utilisées :

- Utilisation d'un deuxième moyen d'authentification pour retrouver ou recréer

un mot de passe (aller voir l'administrateur ou faxer un document). C'est la méthode la plus sûre de recouvrement de mot de passe mais elle est difficile à mettre en place pour des applications web non commerciales.

- Partage de secret : lors de la création du compte l'application pose plusieurs questions personnelles à l'utilisateur, lors de la procédure de recouvrement l'utilisateur doit répondre à une ou plusieurs des questions posées lors de l'inscription. Cette méthode nécessite le stockage d'informations personnelles sur le serveur. Ces questions ne doivent pas porter sur des données qui peuvent être obtenues par un pirate (adresse mail, adresse personnelle, numéro de tel. portable, numéro de sécurité sociale, ...). Plus le pirate connaît personnellement la victime plus il a de chance de pouvoir répondre aux questions.

Bonnes pratiques :

- toujours fournir un nouveau mot de passe quand il a été perdu (si on est capable de fournir l'ancien c'est qu'il a été stocké en clair ou crypté de manière réversible);
- conserver toutes les demandes de recouvrement de mot de passe;
- limiter la durée de validité du nouveau mot de passe envoyé à 24h;
- limiter le mot de passe à une utilisation unique, l'utilisateur devra obligatoirement le changer lorsqu'il se connectera avec son nouveau mot de passe.

2.5 Attaques liées aux autorisations

Le but de ces attaques est d'accroître le niveau de privilège dans une application Web protégée.

a) Prédiction de session

Méthode de détournement de session qui repose sur la prédiction d'un identifiant de session valide

b) Autorisations insuffisantes

Le site web permet un accès à du contenu sensible qui devrait demander des restrictions d'accès accrues. Par exemple dans un menu utilisateur (pour un utilisateur authentifié du site) on ne prévoit pas les liens du menu admin mais l'utilisateur s'il connaît le lien peut accéder à la ressource.

c) Fixation d'identifiant de session

Méthode de détournement de session qui impose à un utilisateur légitime d'un site un identifiant de session.

Une fois la victime authentifiée le pirate peut se loguer sur le serveur avec l'identifiant qu'il avait fixé.

d) Expiration de session

Plus la validité d'une session est courte dans le temps plus il sera difficile pour un pirate de détourner la session. Beaucoup d'attaques sont possibles car les données de sessions restent présentes sur le serveur web (pas supprimées après la session). La session d'un utilisateur ne devrait plus être valide au bout d'une durée fixée selon le type d'application.

2.6 Attaques côté client

a) Usurpation de contenu (content spoofing)

Attaques consistant à faire croire à un utilisateur que le contenu apparaissant sur le site Web est légitime et ne vient pas d'une source extérieure (peut utiliser des frames, des XSS).

b) XSS

Attaque qui a pour but de faire exécuter un code malveillant par le navigateur du client .

2.7 Attaques par exécution de commandes ou de requêtes

a) Injection LDAP

Attaque concernant les applications qui construisent dynamiquement des requêtes LDAP.

b) Injection SQL

Attaque concernant les applications qui construisent dynamiquement des requêtes SQL.

c) Injection SSI

Attaque concernant les serveurs web qui gèrent les Server Side Include dans le HTML avant envoi.

d) Injection XPath

Attaque qui concerne les applications qui construisent dynamiquement des requêtes XPath.

2.8 Attaques liées à la révélation d'informations

Attaques qui permettent d'obtenir des informations systèmes spécifiques au site web : software, numéro de version, niveau de patch, localisation des fichiers temporaires, de sauvegardes, ...

a) Listing des répertoires

Affichage du contenu d'un répertoire qui n'a pas de fichier par défaut.

b) Fuite d'informations (information leakage)

Le site web révèle des données sensibles :

- données permettant de trouver des failles de sécurité (messages d'erreurs, commentaires, informations sur la version des logiciels, ...);
- données permettant d'accéder au système (fichiers de mots de passe);
- données confidentielles (numéros de cartes de crédits, numéros de comptes bancaires, adresses, ...).

La fuite d'information est généralement liée à une authentification ou une autorisation insuffisante ou à l'interception d'informations non cryptées sur le réseau (numéro de compte, ...).

Dans certains cas elle est due à un niveau d'information trop élevé des programmes (erreurs données par le serveur web par ex.).

c) Traversée de chemin (path traversal)

Attaque qui consiste à modifier le chemin de l'arborescence afin d'accéder à des fichiers ou répertoires qui seraient interdits d'accès si on les demandait directement.

Ces attaques utilisent généralement des adresses relatives avec ../ pour remonter jusqu'au répertoire d'intérêt (parfois ../ est codé en hexadécimal lorsque l'attaque est réalisée en passant l'argument par la méthode GET). Lorsque l'accès à une ressource est réalisé à partir d'une donnée qui peut être manipulée par l'internaute (nom dans l'URL par exemple) il faut :

- utiliser realpath pour obtenir le chemin après développement des liens symboliques, de ... et suppression des séparateurs doubles de répertoires ../.
- utiliser basename lorsque la ressource à ouvrir ou inclure est située dans le répertoire courant, cette fonction ne conserve que le nom du fichier d'un chemin (évite la prise en compte d'un chemin si un utilisateur en ajoute un).

2.9 Attaques logiques

Ces attaques concernent l'abus ou l'exploitation du flux logique de l'application Web (procédure de récupération d'un mot de passe oublié, enregistrement

de comptes,...)

a) Abus de fonctionnalité

Attaque qui utilise les caractéristiques et fonctionnalités du site web. Voici quelques exemples :

- utiliser une fonction de recherche du site Web pour accéder à des fichiers en dehors du répertoire ;
- remplacer un fichier de configuration du système en faisant un file upload ;
- déni de service en envoyant plusieurs mots de passe faux pour des utilisateurs existants afin de bloquer leurs comptes.

b) Déni de service (DoS)

Attaque qui a pour but d'empêcher le serveur de répondre aux clients.

Le déni de service est provoqué par la consommation excessive de ressource (CPU, mémoire, bande passante, espace disque, ...). Il est possible également d'obtenir un DoS par buffer overflow ou par abus de fonctionnalité.

L'attaque peut viser :

- un utilisateur en particulier (invalidation du mot de passe) ;
- le serveur de base de données (injection SQL pour amener le serveur à une charge maximale, grand nombre de requêtes à un site web qui utilise un SGBD pour produire les pages, ...) ;
- le serveur web.

- Utiliser un outil pour mesurer les performances lors de la montée en charge permet de donner une idée du nombre de requêtes qu'un pirate aura à générer pour obtenir un déni de service .

- Régler le nombre de clients et de connexions persistantes simultanés et les temps de connexion.

2.10 Les solutions

2.10.1 Solution faille XSS

La solution consiste plutôt à ne pas stocker immédiatement toutes données envoyées à partir des formulaires de votre base de données mais d'abord les analyser : voir s'il y a des caractères spécifiques et surtout s'il y a des balises script.

2.10.2 Solution Injections

Injections SQL :

Pour contrer ce type d'attaques, il est nécessaire d'effectuer un filtrage beaucoup plus précis du contenu des données saisies par les utilisateurs. Il faudra en particulier interdire ou « échapper » les mots clés comme SELECT, INSERT, UNION, LIKE, etc...

L'utilisation de fonctions de substitution et d'expressions régulières est ici très utile. Il est préférable d'utiliser des procédures stockées, moins sujettes à l'injection, et ne pas laisser de requêtes SQL dans les pages de script.

Il est nécessaire ensuite de sécuriser la configuration du service de base de données :

- Suppression des comptes inutiles créés par défaut et création de comptes avec des privilèges réduits (tous les utilisateurs authentifiés ne doivent pas utiliser le même compte pour effectuer toutes les transactions dans la base de données)
- Suppression des procédures stockées présentes par défaut
- Application de permissions d'accès en lecture, suppression, exécution sur les tables, les procédures stockées et les autres objets de la base de données.

2.10.3 Détournement de session

Principe :

Les sessions reposent sur un identifiant unique transmis par le client lors de chaque requête HTTP.

Cet identifiant peut être transmis par :

- URL (méthode GET);
- un champ caché de formulaire envoyé par la méthode POST;
- un cookie.

Renforcer la sécurité des sessions

a) Utiliser un moyen d'identification secondaire

L'identifiant de session est le premier moyen d'identification. Des programmeurs proposent d'utiliser un moyen secondaire. Certains programmeurs utilisent le champ User-Agent de l'en-tête HTTP. Ce champ n'est pas toujours disponible mais il est logique de supposer qu'un utilisateur ne changera pas de navigateur au cours d'une session. Le contenu de User-Agent est donc stocké dans une variable de session sur le serveur en l'encryptant (ceci permet d'éviter de vérifier la validité du contenu avant de l'utiliser). Lors de chaque requête le navigateur est comparé avec celui stocké en session. En cas de différence il faut redemander à l'utilisateur de s'authentifier. Il faut noter cependant que si le pirate a intercepté le cookie il peut aussi avoir intercepté le navigateur de l'utilisateur ou tout autre champ de l'en-tête. Une approche proposée par d'autres programmeurs est de propager une chaîne aléatoire cryptée dans l'URL (l'identifiant unique étant lui propagé par cookie).

b) Détruire les sessions

Il faut toujours proposer à un utilisateur la possibilité de fermer la session et mettre en place une durée limite pour la session (timeout en javascript sur le navigateur, date de validité dans les variables de session). Lorsqu'une session est détruite il faut :

- supprimer les données sur le serveur

- envoyer un cookie vide au client.

c) D'une manière générale

- Mot de passe de l'utilisateur :

- imposer une taille minimale;
- imposer une complexité minimale (nombre de chiffres, car spéciaux, ...) pour contrer les attaques
- de force brute qui utilisent des dictionnaires;
- imposer une modification périodique (sans possibilité de remettre l'ancien mot de passe);
- un mot de passe ne doit pas être stocké en clair ni crypté de manière réversible.

- Authentification :

- Limiter le nombre d'essais à 3 en cas d'échec (bloquer pendant une période de temps : 10 minutes par exemple);
- Utiliser des login comportant uniquement (A-Za-z0-9) (facile à filtrer);
- Ne pas indiquer si c'est le login ou le password qui est faux;
- Indiquer à un utilisateur la date de sa dernière connexion et le nombre de tentatives d'accès qui ont échoué depuis cette dernière connexion;
- Ne jamais soumettre les données par GET;
- Utiliser SSL pour transmettre les identifiants;
- Mettre un no-cache pour la page de login;
- Ne pas protéger uniquement la page d'accès au site mais toutes les pages (vérifier dans chaque page du site que l'utilisateur s'est authentifié);
- Redemander le mot de passe lors d'un changement de niveau de privilège (utilisateur qui a des privilèges de consultation et d'administration doit redonner son mot de passe quand il passe de la partie consultation à la partie administration).

- Modification du compte utilisateur :

- mot de passe : demander l'ancien mot de passe (utile en cas de détournement de session)
- mail : demander la saisie du mot de passe (important pour les sites qui effectuent un recouvrement de mot de passe par mail car en cas de détournement de session le pirate qui ne connaît pas le mot de passe ne pourra pas changer le mail)

- Outils d'administration : utiliser SSL, changer les mots de passe et le login donnés par défaut à l'administrateur.

2.11 Conclusion

La sécurité du Web est une des grandes problématiques actuelles. La difficulté est d'avoir la capacité de protéger automatiquement une application Web, c'est-à-dire être capable de filtrer les données entrantes (en ne laissant que les caractères attendus) tout en garantissant l'intégrité des données envoyées par l'internaute. En effet, par défaut il ne faut faire aucunement confiance aux données reçues et ne déléguer aucun traitement critique au niveau du client. Ainsi, de nombreux produits de sécurité font surface afin de jouer un rôle de reverse proxy filtrant avec des règles de filtrage très précises.

Chapitre 3

Analyse et Conception

3.1 Introduction

Dans un projet de développement d'un système d'information, il y a une démarche qui garantira la réussite du résultat final. En effet, à la suite de notre synthèse bibliographique qui nous a permis de comprendre les procédures de travail, l'étape de conception est la phase dans laquelle les contours du système seront définis. C'est pour cette raison que cette partie du développement d'un système d'information est jugée délicate et déterminante pour l'obtention d'un résultat fiable et qui répond aux besoins des utilisateurs. Car, une simple erreur commise dans la conception d'un Système d'information pourrait avoir des conséquences sur la suite du projet. Pour mener à bien cette étape nous avons choisi l'utilisation de UML (Unified Modeling Language) qui offre une riche palette de diagrammes qui permettent de modéliser tous les aspects d'un système d'information.

3.2 Présentation de L'UML :

Face à la diversité des méthodes d'analyse et de conception objet, en particulier aux différentes notations des mêmes concepts, UML (Unified Modeling Language) représente un réel facteur de progrès par l'effort de normalisation réalisé. En effet, UML constitue une étape importante dans la convergence des notations utilisées dans le domaine de l'analyse et la conception objet puisqu'il représente une synthèse des trois méthodes OMT, BOOCH, et OOSE. Ces trois méthodes couvrent environ la moitié du marché des méthodes objet.

3.2.1 Le langage de modélisation UML :

L'UML est un langage de modélisation graphique, il permet de représenter sous forme de diagrammes différents aspects d'un logiciel dans le cadre de ses phases d'analyse et de conception. UML propose de décrire un système à l'aide de neuf (9) diagrammes :

1. Diagrammes de cas d'utilisation : représentation des fonctions du système du point de vue de l'utilisateur.
2. Diagrammes d'objets : représentation des objets et de leurs relations, correspond à un diagramme de collaboration simplifié, sans représentation des envois de messages.

3. Diagrammes de classes : représentation de la structure statique en termes de classes et de relations.
4. Diagrammes de composants : représentation du code en termes de modules, de composants et surtout des concepts du langage ou de l'environnement d'implémentation.
5. Diagrammes de déploiement : représentation du déploiement des composants sur les dispositifs matériels.
6. Diagrammes de collaboration : représentation spatiale des objets, des liens et des interactions.
7. Diagrammes de séquence : représentation temporelle des objets et de leurs interactions.
8. Diagrammes d'états-transitions : représentation du comportement d'une classe en terme d'état.
9. Diagrammes d'activités : représentation du comportement d'une opération en terme d'actions.

Ces diagrammes, d'une utilité variable selon les cas, ne sont pas nécessairement tous produits à chaque modélisation. Pour le cas de notre application, Les diagrammes utilisés sont les diagrammes de classes et de cas d'utilisation.

3.3 Définition des besoins :

La définition des besoins est la première étape dans le cycle de développement d'un logiciel. Elle doit traduire ce que le futur système est susceptible d'apporter aux utilisateurs, en faisant abstraction de la manière dont il sera construit. Elle définit les fonctionnalités du système et surtout la façon de l'utiliser. Cette première phase, se focalise donc sur les propriétés externes du logiciel, à savoir :

- Ce que le système peut apporter à l'utilisateur.
- Comment le système se comporte face à l'utilisateur.

L'emploi du modèle de « **cas d'utilisation** » est une bonne approche pour définir les besoins des utilisateurs. Il permet la plupart du temps de compléter les besoins déjà élaborés, de les améliorer, de les corriger et également de les valider.

3.3.1 Analyse fonctionnelle

SPECIFICATIONS :

- Un Postulant peut déposer un seul CV.
- Un Postulant peut postuler sur un ou plusieurs offres d'emploi.
- Un Recruteur offre un ou plusieurs offres d'emploi.
- Un Recruteur recrute un ou plusieurs Postulant.

Identification des acteurs du système :

La liste des acteurs : nous avons identifié les acteurs suivants :

- **Administrateur :** C'est le Gérant du site, c'est lui le responsable principal de Tout les publications.
- **Recruteur :** Dépôt des offres d'emploi en ligne
- **Postulant :** Dépôt de CV en ligne

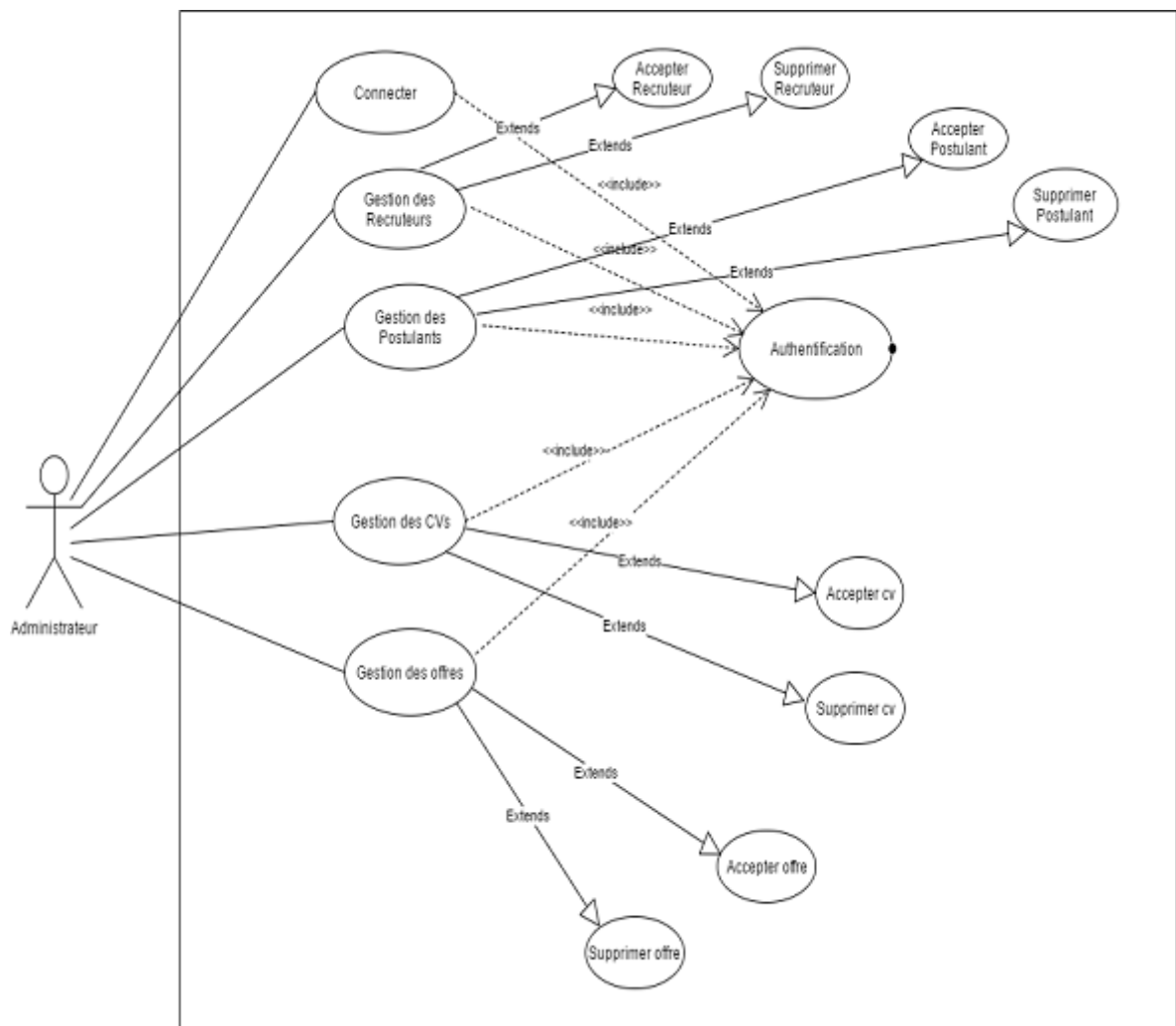
3.3.2 DIAGRAMME DE CAS D'UTILISATION :

FIGURE 3.1 – Diagramme de cas d'utilisation «Administrateur ».

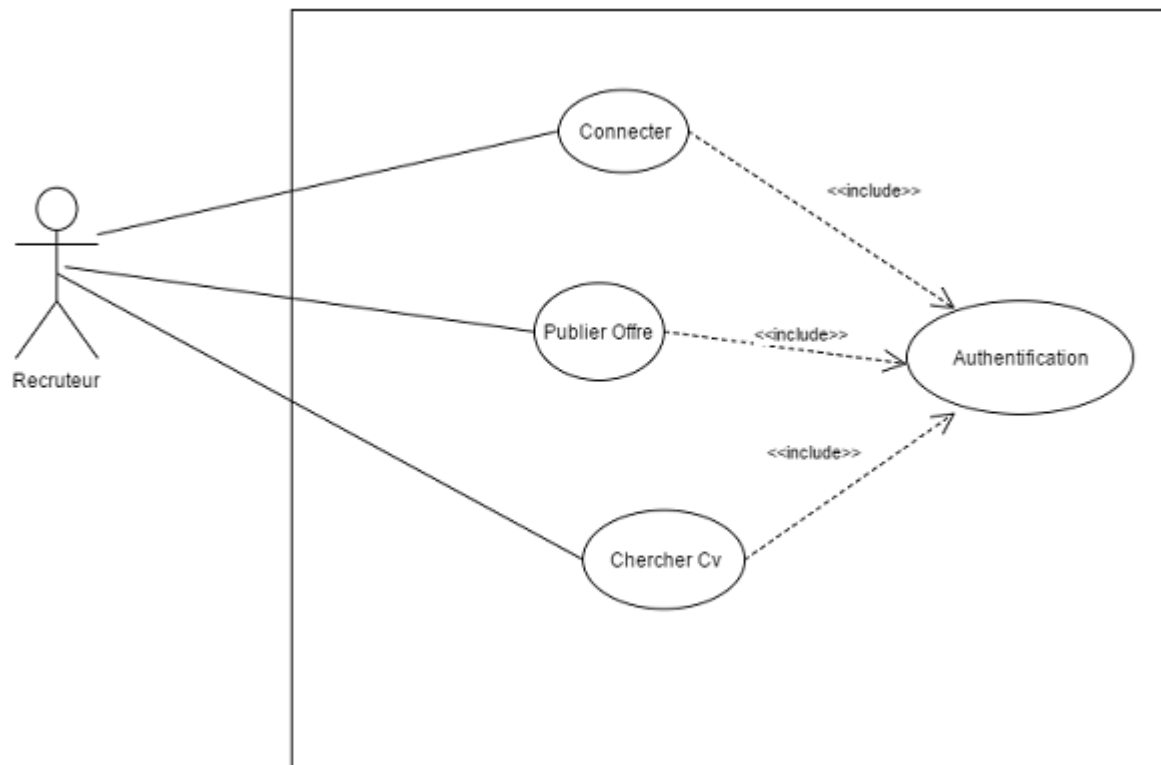


FIGURE 3.2 – Diagramme de cas d'utilisation «Recruteur».

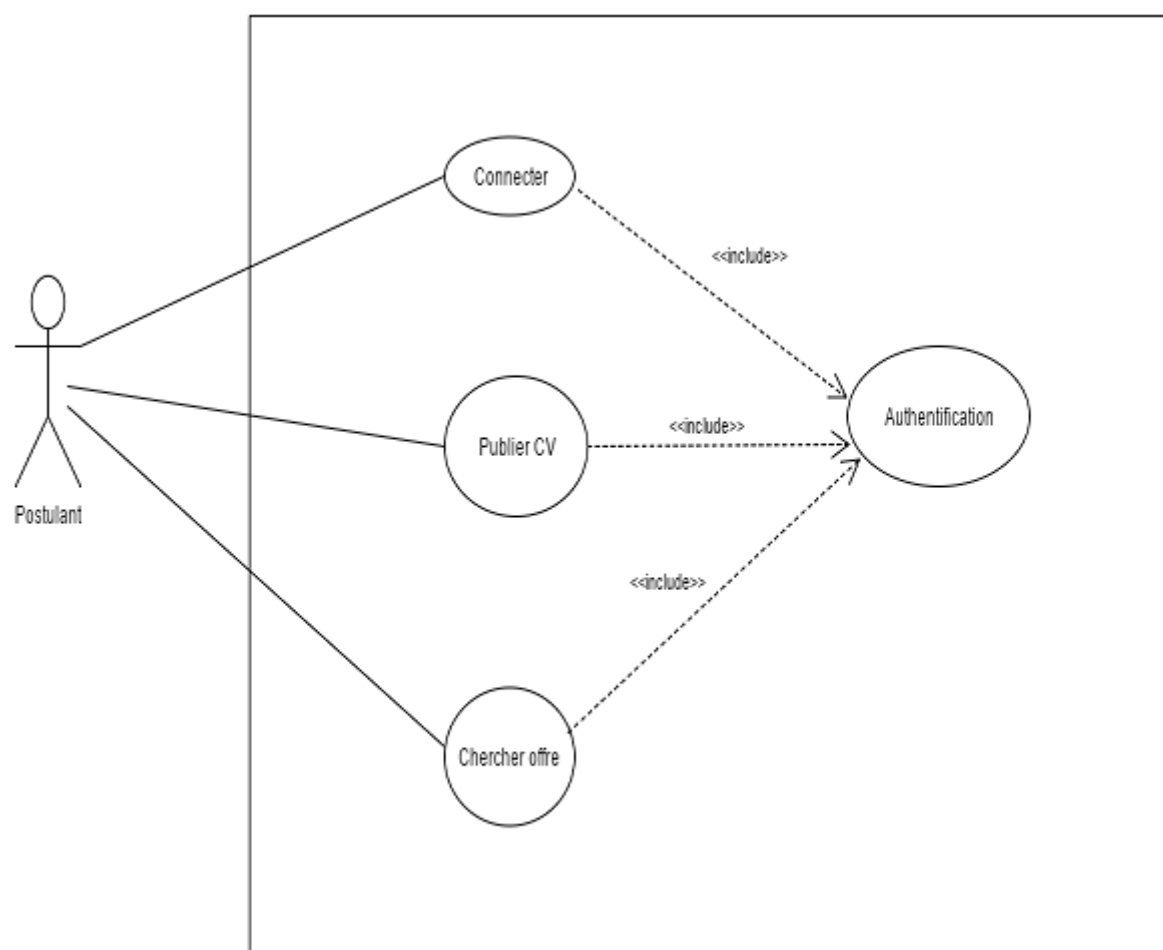


FIGURE 3.3 – Diagramme de cas d'utilisation «Postulant».

3.3.3 DIAGRAMME DE CLASSE :

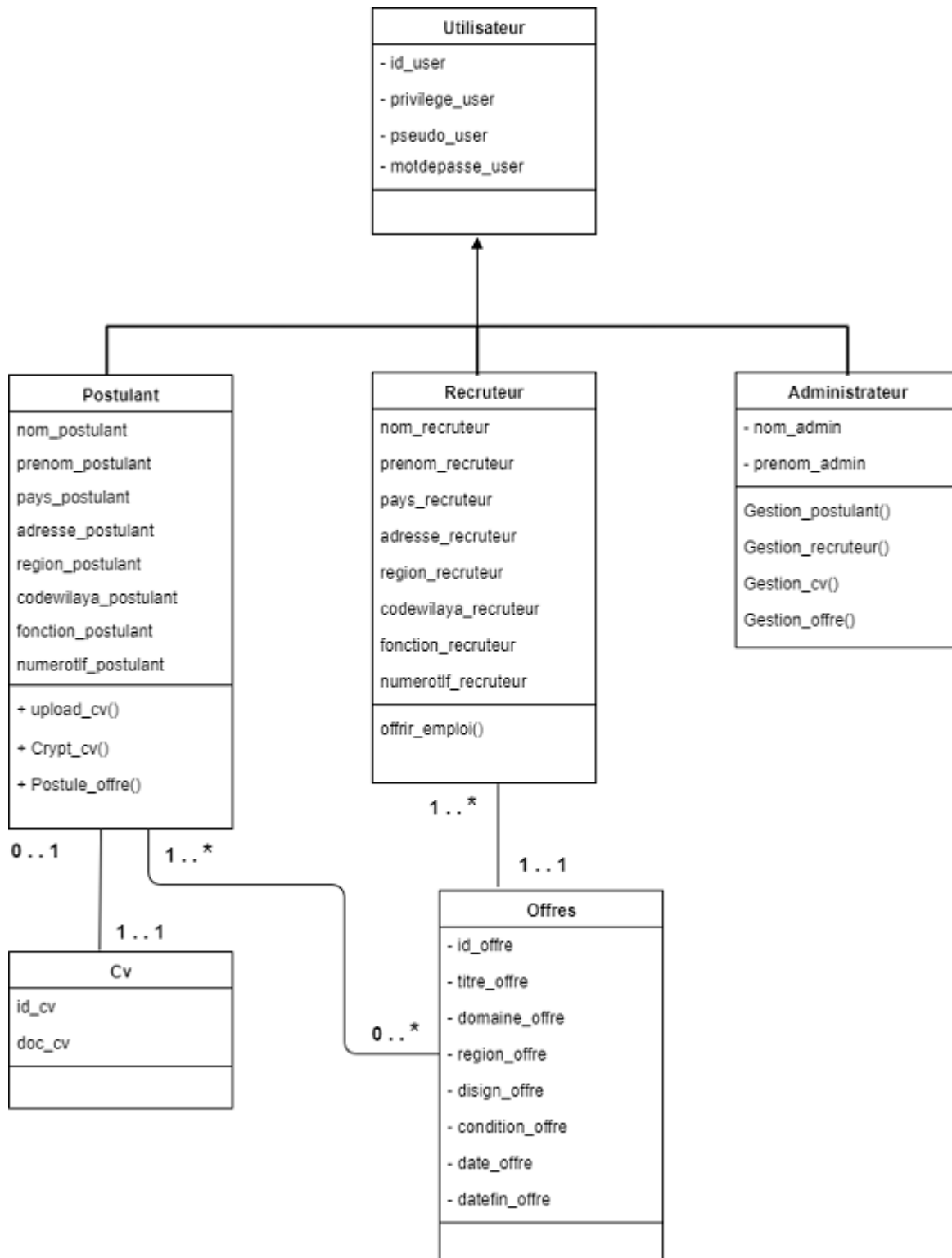


FIGURE 3.4 – Diagramme de «classe».

Chapitre 4

Implémentation

4.1 Introduction

La phase d'implémentation donne une description technique détaillée du système conçu. Elle permet de présenter l'architecture logique et physique (matérielle) du système, ainsi de décrire les techniques utilisées dans l'implémentation (réalisation). Nous allons donc examiner d'abord les différents schémas de déploiement des applications pour prendre les décisions adaptées au projet.

4.2 Outils de réalisation :

Dans cette partie nous allons présenter les principaux outils utilisés pour la mise en place de notre application. Nous avons décidé d'utiliser le langage PHP et le système de la base de données MySQL qui sont décrits en détail dans la section suivante.

4.2.1 PHP :

Qu'est ce PHP

PHP a une définition réursive : HyperText Processor. Mais, en réalité, son premier nom est : Personal Home Page Tools. PHP est un langage de script côté serveur inclus dans HTML. PHP est créé par Rasmus Lerdorf, un ingénieur qui fait partie de l'équipe développement d'Apache. La première partie de PHP est construite en 1994. En 1997, il y avait 50.000 sites utilisant PHP. En 2000, ce chiffre était 1.000.000. En 2005, il était 22.000.000. Et maintenant, il est environ 244 Millions.

Les Bonnes raisons pour utiliser PHP

- **PHP est gratuit :** PHP ne coûte rien. Pas un dinar, pas un euro. Rien au départ, rien pendant la durée de la vie de l'application, et rien à la fin. Le développement, le serveur, la gestion de la base de données, le support, tous sont gratuits.

- **PHP est simple :** Le syntaxe de PHP est simple, PHP est donc facile à apprendre. Pourtant, on ne peut pas utiliser les outils pour générer le code source de PHP, ils sont écrits à la main.
- **PHP est incorporé :** PHP vient s'incorporer dans HTML. L'incorporation de PHP dans HTML a plusieurs conséquences utiles comme : PHP peut être rapidement ajouté à du code produit par un éditeur HTML graphique ; PHP se prête de lui-même à une division du travail entre concepteurs graphiques et développeur de scripts ; PHP peut réduire les coûts de développement et améliorer son efficacité.
- **PHP est disponible sur plusieurs plates-formes :** PHP est disponible en natif pour Unix et pour Windows (la plupart des serveurs HTTP fonctionne sous l'un de ces types de système d'exploitation). PHP est aussi compatible avec les serveurs Web populaires : Apache HTTP Server, Microsoft Internet Information Server et Netscape Entreprise Server.
- **PHP est de plus en plus populaire :** PHP devient rapidement l'une des solutions de Développement dite «à deux étages» (Web et données).

4.2.2 MySQL

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

Les Bonnes raisons pour utiliser MySQL

- **Montée en charge et flexibilité :** Le serveur de base de données MySQL offre les meilleures performances en termes de montée en charge. Il est capable de gérer des applications embarquées n'utilisant qu'1 Mo de mémoire comme des entrepôts de données de grande taille contenant plusieurs téraoctets d'information. La polyvalence des plates-formes est l'un des points forts de MySQL, qui fonctionne sur toutes les déclinaisons de Linux, UNIX ou Windows. Et, bien sûr, sa nature open source autorise une personnalisation complète pour les utilisateurs désirant ajouter des fonctionnalités spécifiques au serveur de base de données.
- **Des performances élevées :** Une architecture unique de moteur de stockage Permet aux professionnels des bases de données de configurer le serveur MySQL de façon spécifique pour certaines applications, avec pour résultat des performances stupéfiantes. Que l'application envisagée soit un système de traitement de transactions à haut débit ou un site Web à fort volume servant un milliard de requêtes par jour, MySQL peut répondre aux

demandes de performance les plus exigeantes. Grâce à ses utilitaires de charge à haute vitesse, son indexation en texte intégral et à d'autres mécanismes d'amélioration des performances.

- **Haute disponibilité :** Une fiabilité à toute épreuve et une disponibilité constante est la marque de fabrique de MySQL. C'est pourquoi ses utilisateurs lui font confiance pour garantir un fonctionnement sans faille de leurs systèmes. MySQL offre une grande diversité d'options de haute disponibilité, depuis des configurations de réplication maître/esclave à haut débit jusqu'aux serveurs spécialisés en clusters offrant des fonctions de basculement instantané, en passant par des solutions de haute disponibilité proposées par nos partenaires.
- **Un support transactionnel solide :** MySQL offre l'un des moteurs de bases de données transactionnelles les plus puissants du marché. Il est pourvu de fonctionnalités complètes de support de transaction ACID (atomique, constant, isolé, durable), d'une fonction de verrouillage de ligne illimitée, de capacités de transactions distribuées et d'un support de transactions multi-version dans lequel les opérations de lecture ne bloquent jamais celles d'écriture et vice-versa. L'intégrité complète des données est également assurée par une fonction d'intégrité référentielle via le serveur, par des niveaux spécialisés d'isolation de transactions et par la détection instantanée des blocages.
- **De puissantes fonctionnalités Web et d'entreposage de données :** MySQL est le standard lorsqu'il s'agit de sites web à fort trafic en raison de son moteur de requêtes à hautes performances, de ses capacités d'insertion de données phénoménalement rapides et de ses fonctions web spécialisées telles que la recherche rapide en texte intégral. Ces mêmes capacités s'appliquent également aux environnements d'entrepôts de données, dans lesquels MySQL peut gérer de nombreux téraoctet qu'il s'agisse de serveurs simples ou d'architectures en déploiement horizontal (scale-out). Ses autres caractéristiques, par exemple les tables de mémoire principale, l'indexation du B-tree et du hachage ou les tables d'archives comprimées, qui ont pour effet de réduire les besoins de stockage jusqu'à 80
- **Une forte protection des données :** La protection des données névralgiques d'une entreprise étant la tâche prioritaire des professionnels des bases de données, MySQL offre des fonctions de sécurité exceptionnelles qui garantissent une protection des données absolue. En matière d'authentification des bases de données, MySQL dispose de mécanismes puissants visant à s'assurer que seuls les utilisateurs autorisés ont accès au serveur de la base de données, avec la possibilité de bloquer les utilisateurs au niveau de la machine client. La prise en charge des protocoles SSH et SSL est également fournie afin de garantir des connexions sûres et sécurisées. Une infrastructure de privilèges d'objets granulaires a été intégrée, de façon à ce que les utilisateurs ne puissent voir que les données pour lesquelles ils disposent d'une autorisation. Par ailleurs de puissantes fonctions de chiffrement et de déchiffrement des données assurent la protection des données sensibles

contre les accès non autorisés. Enfin, des utilitaires de sauvegarde et de récupération fournis par MySQL et par des éditeurs de logiciels tiers permettent d'effectuer des sauvegardes logiques et physiques ainsi que des opérations de récupération complètes ou ponctuelles.

- **Des fonctions complètes de développement d'applications :** L'une des raisons pour lesquelles MySQL est la base de données open source la plus populaire au monde est qu'elle est adaptée à tous les besoins de développement d'applications. Au sein de la base de données, on pourra bénéficier de procédures stockées, de déclencheurs, de fonctions, de vues, de curseurs, d'un SQL à la norme ANSI, etc. Pour les applications embarquées, des bibliothèques de plug-ins sont disponibles pour intégrer la prise en charge des bases de données MySQL dans la quasi-totalité des applications. MySQL fournit également des pilotes (ODBC, JDBC, etc.) qui permettent à toutes les formes d'applications d'utiliser MySQL comme serveur préférentiel de gestion des données. MySQL offre aux développeurs d'applications, qu'ils travaillent en PHP, Perl, Java, Visual Basic ou .NET, tout ce dont ils ont besoin pour réussir le développement de leurs systèmes d'information pilotés par bases de données.
- **Facilité d'utilisation et d'administration** MySQL possède d'exceptionnelles capacités de démarrage rapide, le temps nécessaire pour installer le logiciel après l'avoir téléchargé n'excédant pas 15 minutes. Cette règle vaut aussi bien pour la plate-forme Microsoft Windows que pour Linux, Macintosh ou UNIX. Une fois l'installation terminée, les fonctions de gestion automatique, telles que l'extension d'espace automatique, le redémarrage automatique et les changements de configuration dynamiques, soulagent d'une grande partie du travail des administrateurs de bases de données déjà surchargés.

4.2.3 XAMPP (MySQL, Apache PHP et Perl) :

XAMPP [16] est un ensemble de logiciels permettant de mettre en place facilement un serveur Web et un serveur FTP. Il s'agit d'une distribution de logiciels libres (X Apache MySQL Perl PHP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. Ainsi,

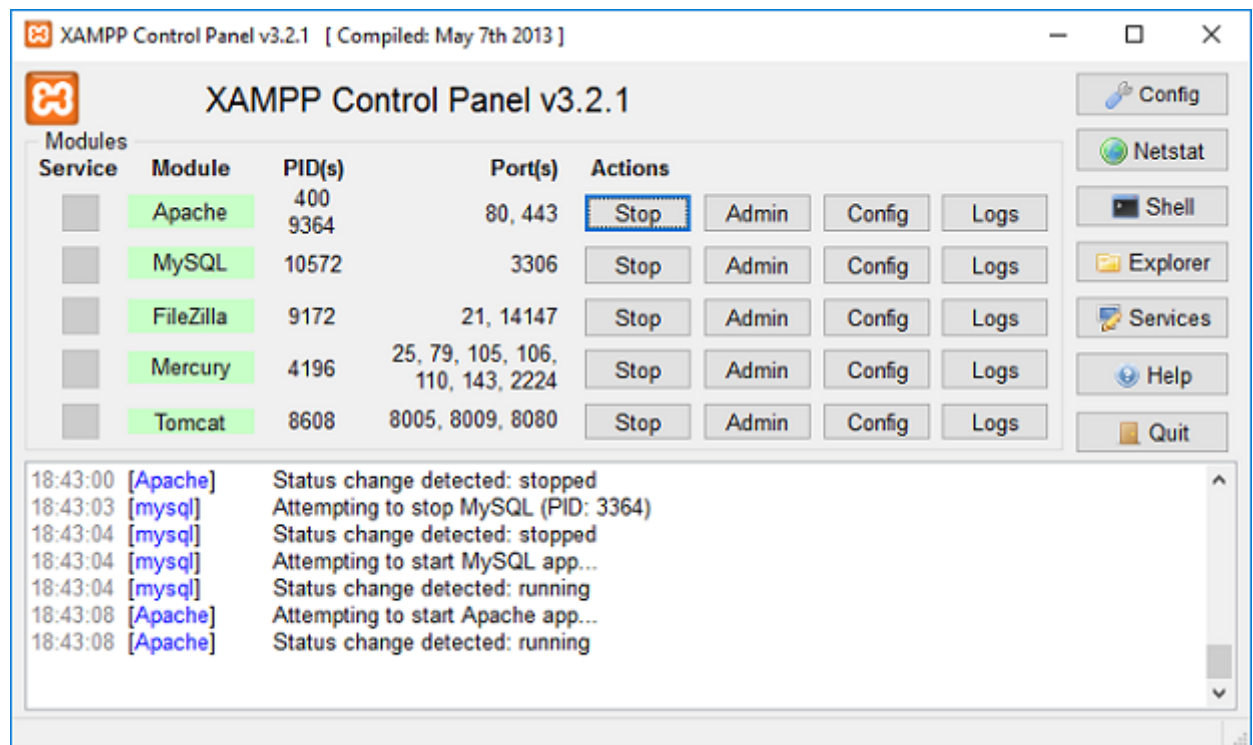


FIGURE 4.1 – Interface Xampp.

4.2.4 Editeur NOTEPAD++ :

Notepad++ [9] est un éditeur de code source qui prend en charge plusieurs langages. Ce programme, codé en C++ avec STL et win32 api, a pour vocation de fournir un éditeur de code source de taille réduite mais très performant. En optimisant de nombreuses fonctions tout en conservant une facilité d'utilisation et une certaine convivialité, Notepad++ contribue à la limitation des émissions de dioxyde de carbone dans le monde : en effet, en réduisant l'utilisation de CPU, la consommation d'énergie des ordinateurs chute considérablement, en conséquence de quoi, la terre est plus verte

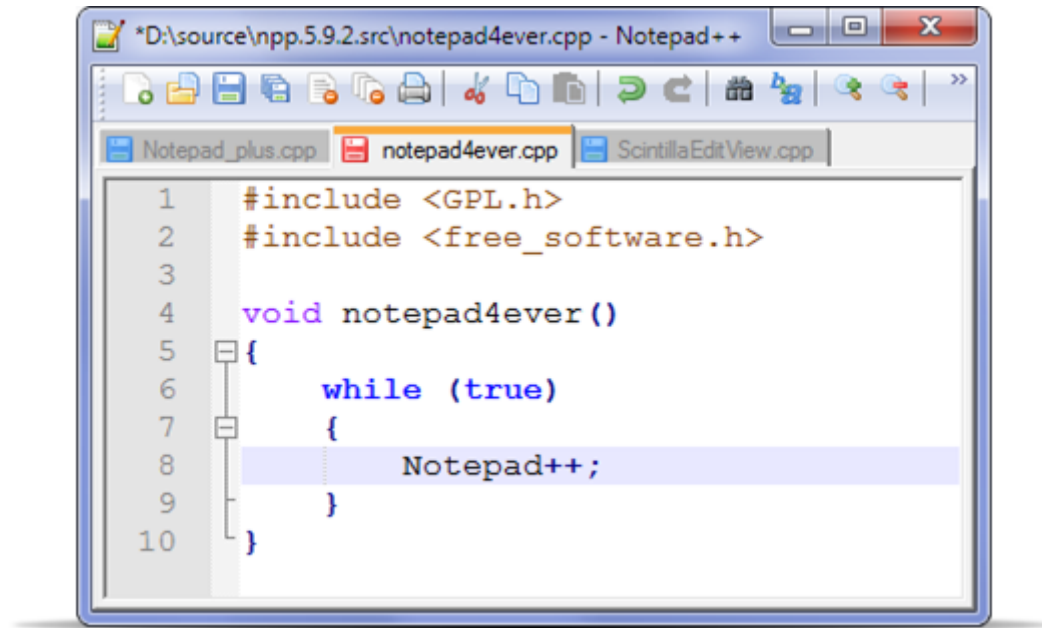


FIGURE 4.2 – Interface Notepad++.

4.3 Réalisation de l'application Web :

4.3.1 Les interfaces de l'application :

La page d'accueil de l'application contient le menu principal du site Web. Des liens donnent la possibilité à l'utilisateur d'accéder directement aux rubriques qui l'intéressent. Le menu principal est composé des pages Accueil, Offres, Contact, Connexion, Enregistrement

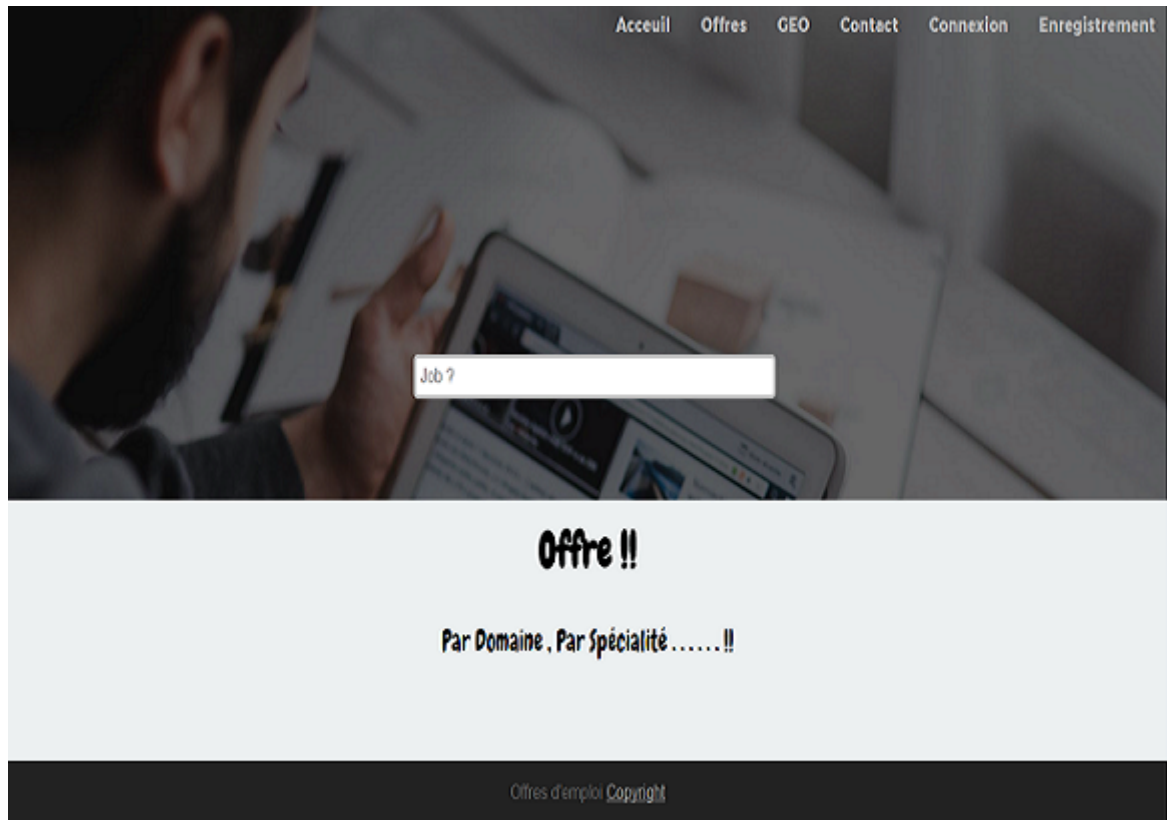


FIGURE 4.3 – Interface d'accueil.

Pour s'inscrire, l'utilisateur (Recruteur ou Postulant) clique sur le lien Enregistrement. Il remplit un formulaire et soumet l'information au serveur.

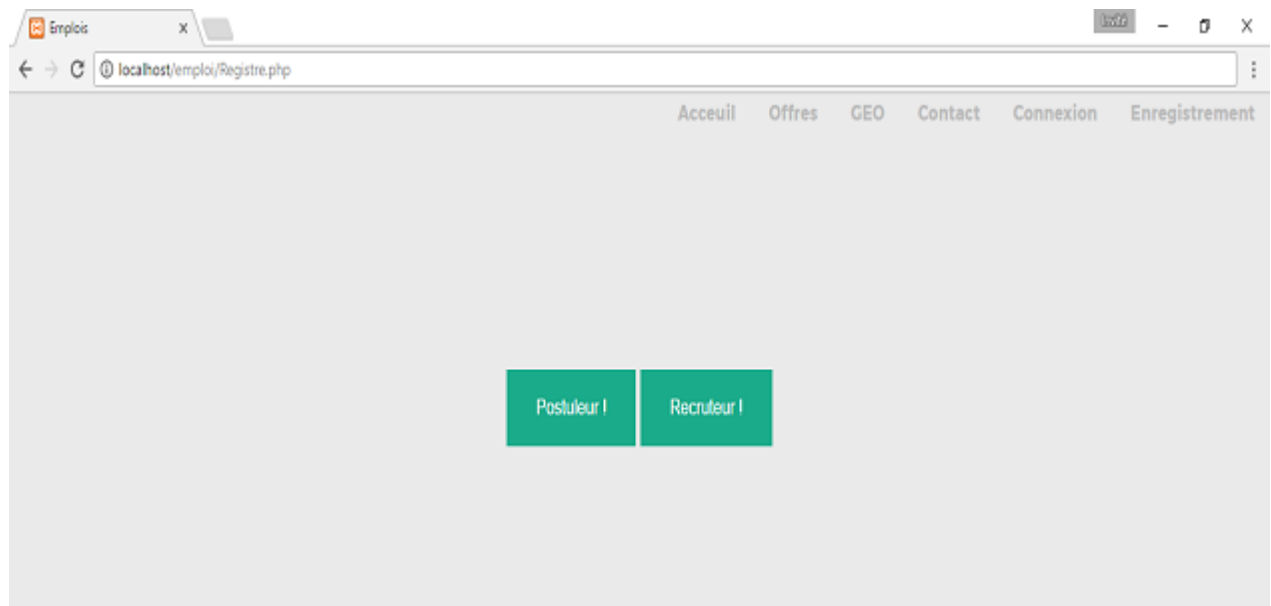
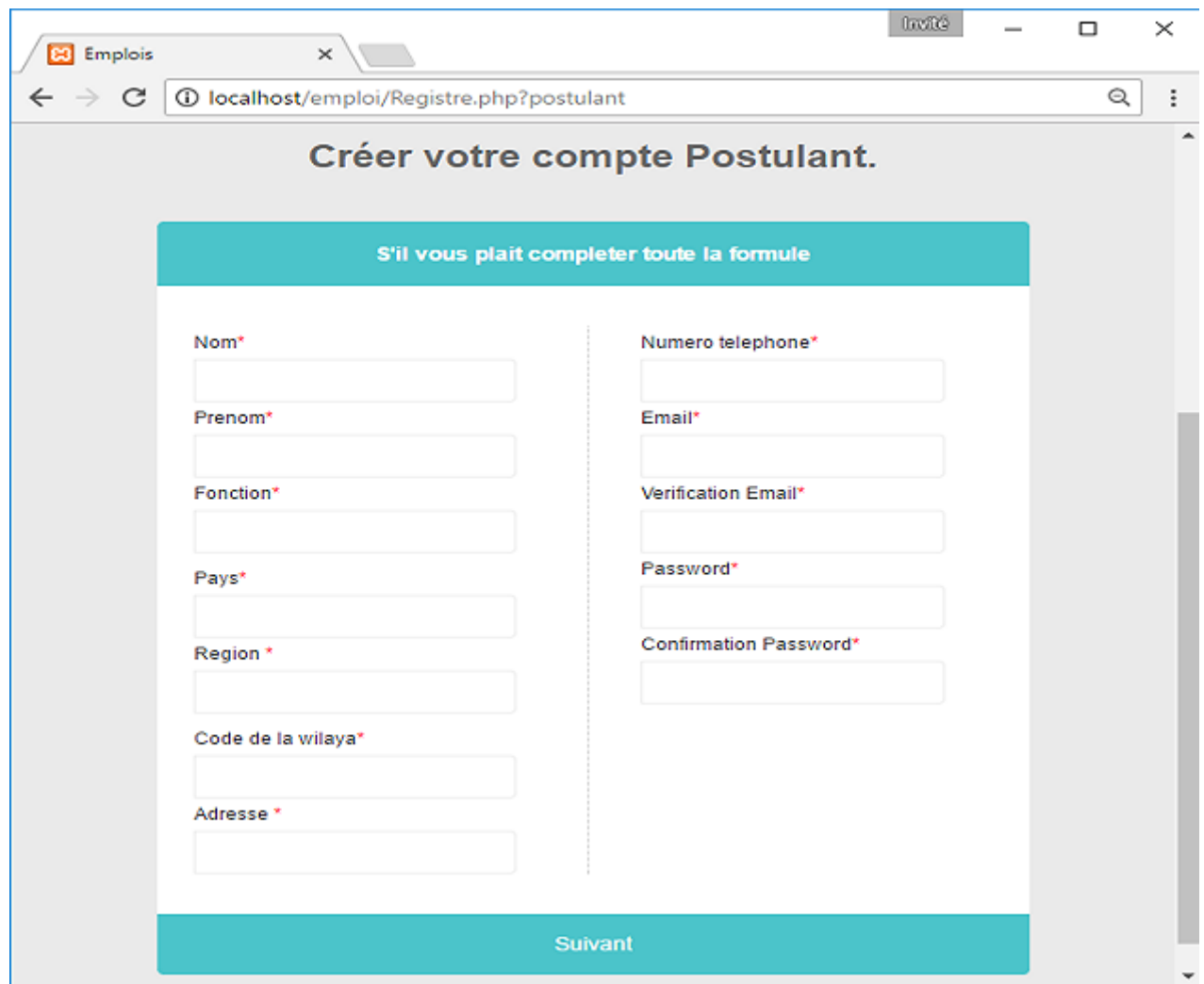


FIGURE 4.4 – Interface d'enregistrement.

- Cet écran affiche le formulaire d'enregistrement pour postulant



The screenshot shows a web browser window with the title 'Emplois' and the URL 'localhost/emploi/Registre.php?postulant'. The page has a header 'Créer votre compte Postulant.' and a teal banner with the text 'S'il vous plaît compléter toute la formule'. Below this is a registration form with two columns of input fields, each with a red asterisk indicating it is required. The left column contains: 'Nom*', 'Prenom*', 'Fonction*', 'Pays*', 'Region *', 'Code de la wilaya*', and 'Adresse *'. The right column contains: 'Numero telephone*', 'Email*', 'Verification Email*', 'Password*', and 'Confirmation Password*'. At the bottom of the form is a teal button labeled 'Suivant'.

FIGURE 4.5 – Interface d'enregistrement postulant.

- Cet écran affiche le formulaire d'enregistrement pour recruteur



The screenshot shows a web browser window with the title 'Emplois' and the URL 'localhost/emploi/Registre.php?recruteur'. The page content is titled 'Créer votre compte Recruteur.' and includes a teal header bar with the text 'S'il vous plaît compléter toute la formule'. The form is divided into two columns by a vertical dashed line. The left column contains fields for 'Nom*', 'Prenom*', 'Fonction*', 'Pays*', 'Region*', 'Code de la wilaya*', and 'Adresse*'. The right column contains fields for 'Numero telephone*', 'Email*', 'Verification Email*', 'Password*', and 'Confirmation Password*'. A teal button labeled 'Suivant' is located at the bottom of the form.

FIGURE 4.6 – Interface d'enregistrement recruteur.

Pour accéder à son compte, l'utilisateur (Recruteur ou Postulant) doit fournir à l'application une combinaison valide de mot de passe et de l'email.

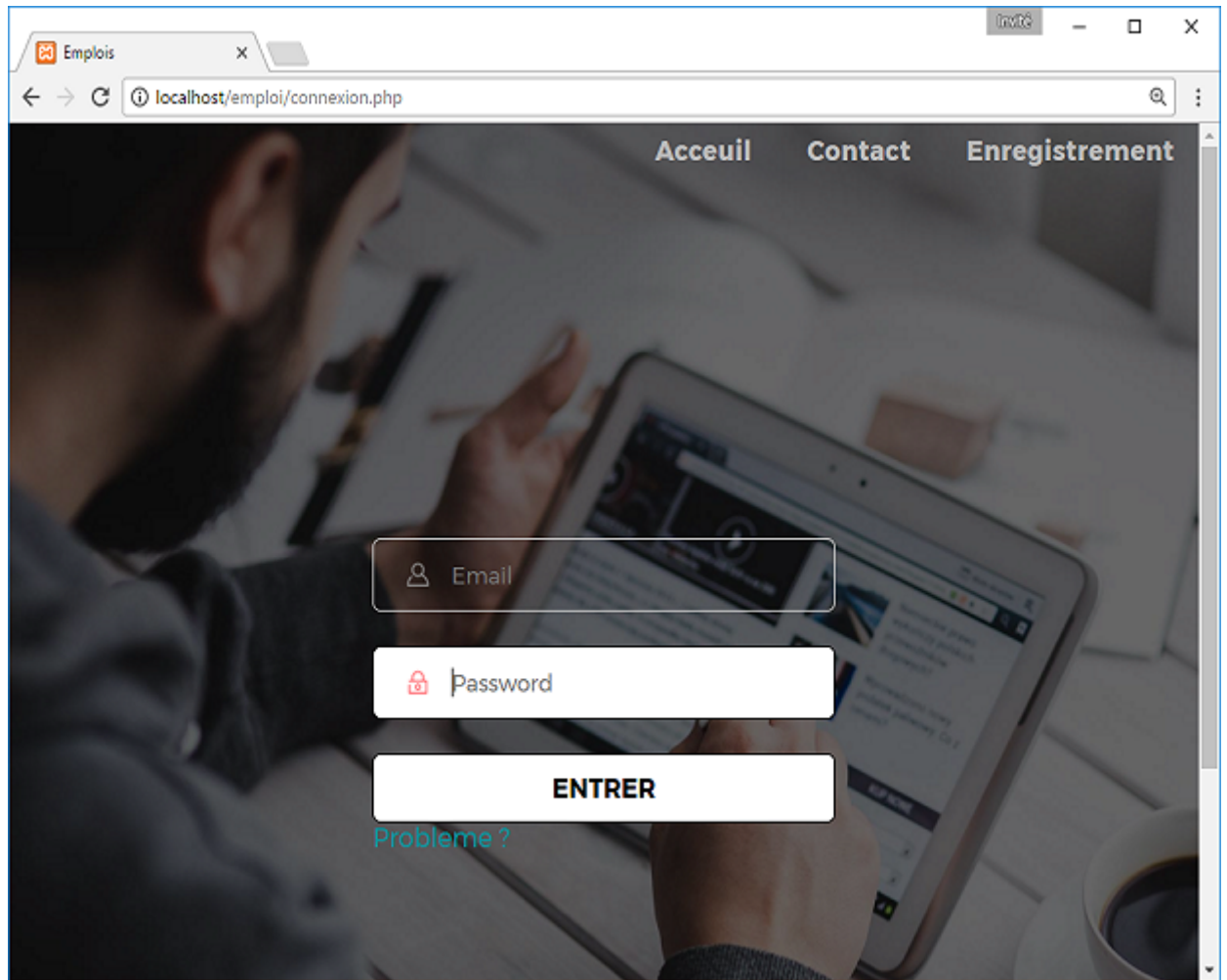


FIGURE 4.7 – Interface de connexion.

Pour consulter les offres d'emploi, l'utilisateur clique sur le bouton **offres**. En cliquant sur une offre choisie, le détail s'affiche. Il a la possibilité de postuler directement après la consultation d'une offre, Seuls les postulants inscrits peuvent postuler de manière interactive. Pour pouvoir postuler, il faut cliquer sur le bouton **Postuler** après avoir joint un CV sur le compte personnel.

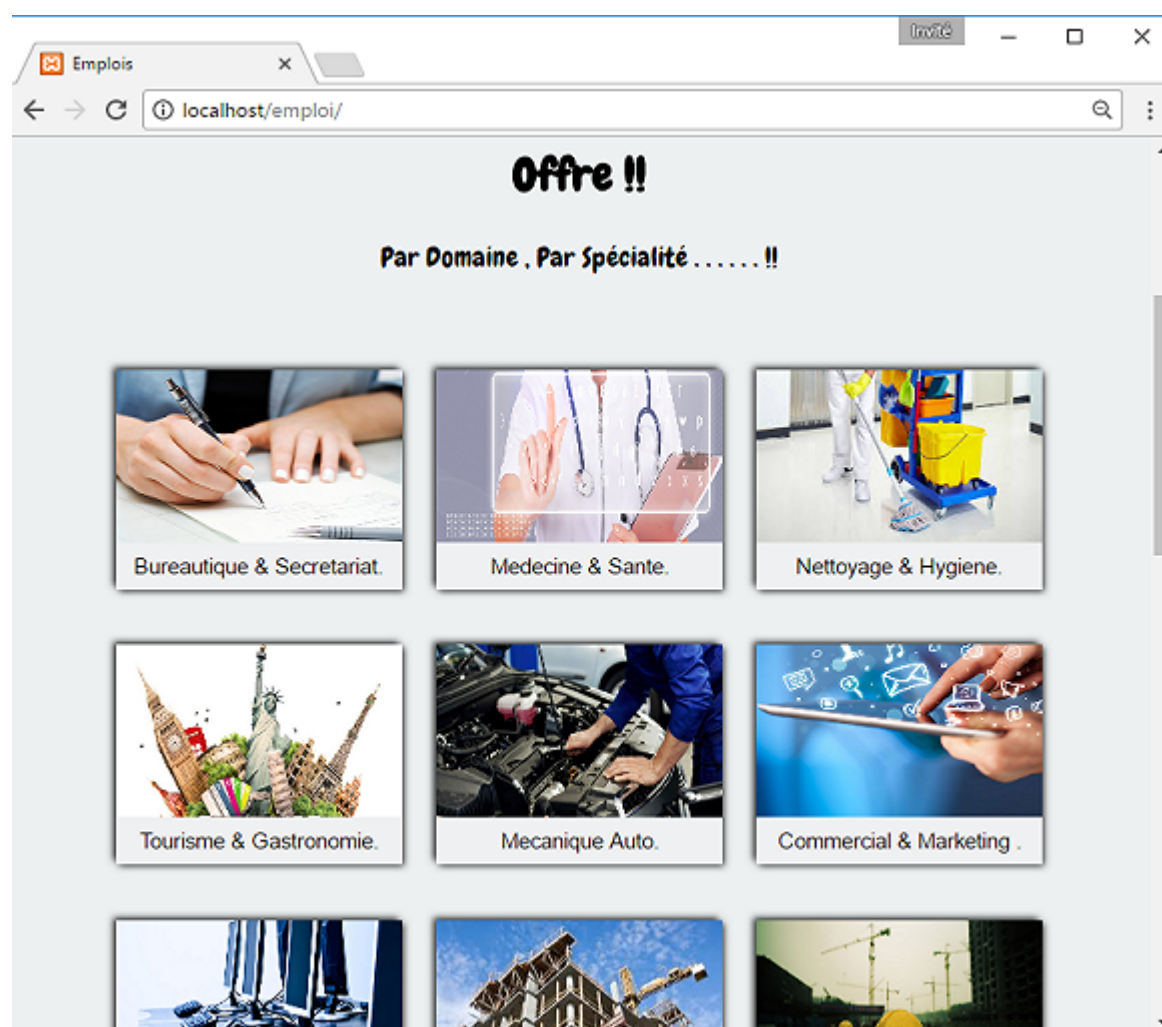


FIGURE 4.8 – Interface par domaine.

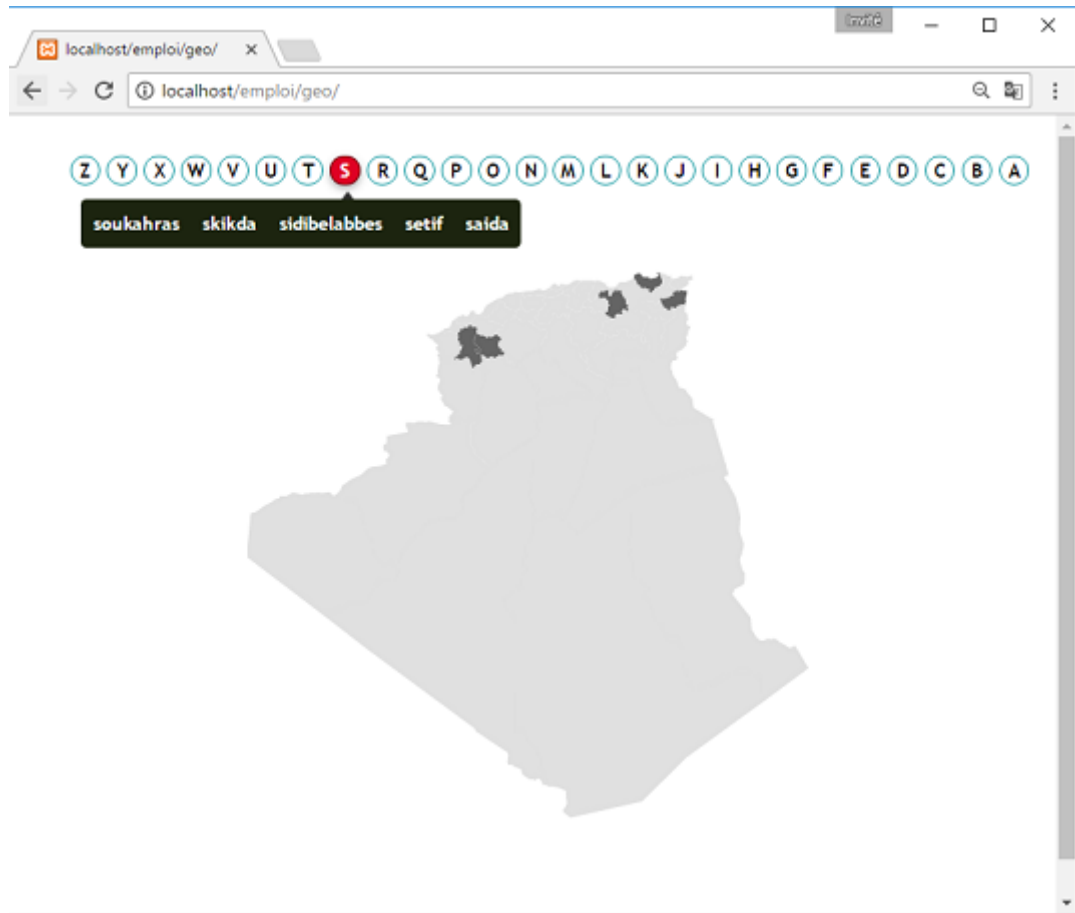


FIGURE 4.9 – Interface géographique.

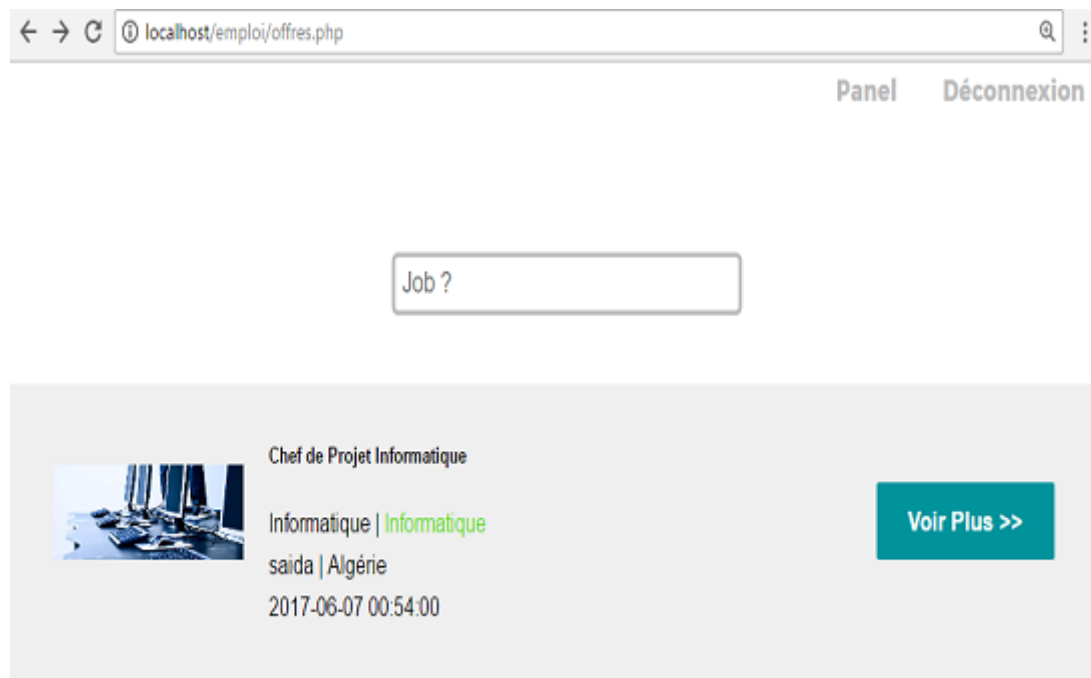


FIGURE 4.10 – Interface offre.

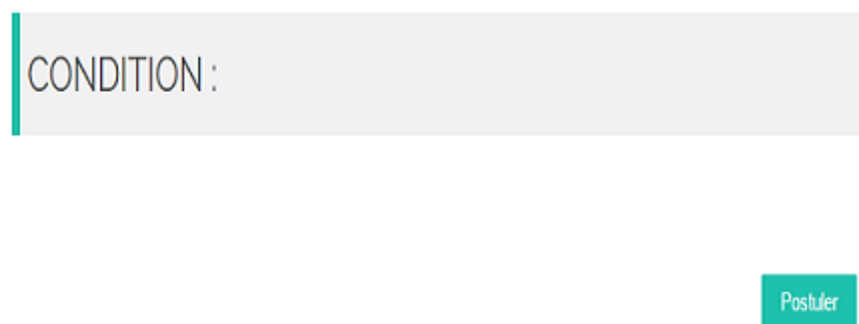


FIGURE 4.11 – Boutton postuler.

Le recruteur doit se connecter pour gérer les offres d'emploi et consulter la liste des postulant, le recruteur clique sur le lien des postulant. En cliquant sur un CV choisi, le détail s'affiche.

Interface administration :

- Cet écran affiche la gestion des utilisateurs "recruteurs et postulants".


The screenshot displays a web application interface for user management. On the left is a dark sidebar with navigation links: General, Messages, Gestion Des Utilisateurs (highlighted), Offres, and CV. The main content area is titled 'Gestion :'. It contains two tables. The first table, 'Listes des recruteurs', has one row with data for a recruiter named 'ouis'. The second table, 'Listes des Postulant', has three rows with data for applicants named 'postulant1', 'postulant2', and 'sicsic'. Each row in both tables includes columns for ID, Name & Surname, Email, Function, Region, Address, and Phone Number, along with 'Activer' and 'ANNULER' links.

#	Nom & Prenom	Email	Fonction	Région	Adresse	Numero telephone		
1	ouis recruteur	ouis@ouis.com	Entreprise	saida	rue arbi ben mhidi	123456789	Activer	ANNULER

#	Nom & Prenom	Email	Fonction	Région	Adresse	Numero telephone		
1	postulant1 postulant1	postulant@postulant	informaticien	saida	Allel medaghri	987654321	Activer	ANNULER
1	postulant2 postulant2	pos@pos	postulant2	postulant2	postulant2	987654321	Activer	ANNULER
1	sic sicsic	sic@sic	sicsic	sic	sicsic	1097654456	Activer	ANNULER

FIGURE 4.12 – Interface gestion d'utilisateurs.

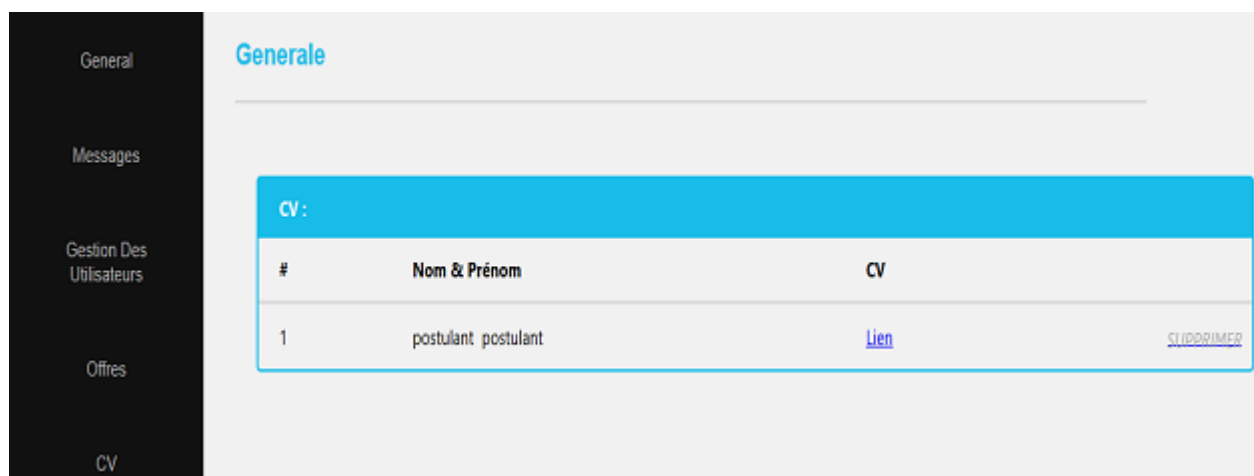
- Cet écran affiche la gestion des offres



#	Titre	Domaine	Pays	Wilaya	Date Création	Date fin	
1	Chef de Projet Informatique	Informatique	Algérie	saida	2017-06-07 00:54:00	2017-06-26 00:00:00	ANNULER

FIGURE 4.13 – Interface gestion des offres.

- Cet écran affiche la gestion des CV.



#	Nom & Prénom	CV
1	postulant postulant	Lien

[AJOUTER](#)

FIGURE 4.14 – Interface gestion des CV.

Interface Recruteur :

-Cet écran affiche un formulaire pour ajouter une nouvelle offre



The screenshot shows a web form titled "Ajouter une nouvelle offre" in a bold, dark grey font. The form is set against a light grey background. It contains several input fields: a single-line text box for "Titre de l'offre", a single-line text box for "Domaine / Spécialité", two side-by-side single-line text boxes for "Pays" and "Région / wilaya", a date selection field for "Date fin de l'offre :" with a placeholder "jj/mm/aaaa --:--", a large multi-line text box for "Disignation", and another large multi-line text box for "Condition de cette offre". At the bottom of the form is a prominent blue button with the white text "Ajouter".

FIGURE 4.15 – Interface Ajouter des offres.

-Cet écran affiche la liste des offres avec les postulants Correspondant

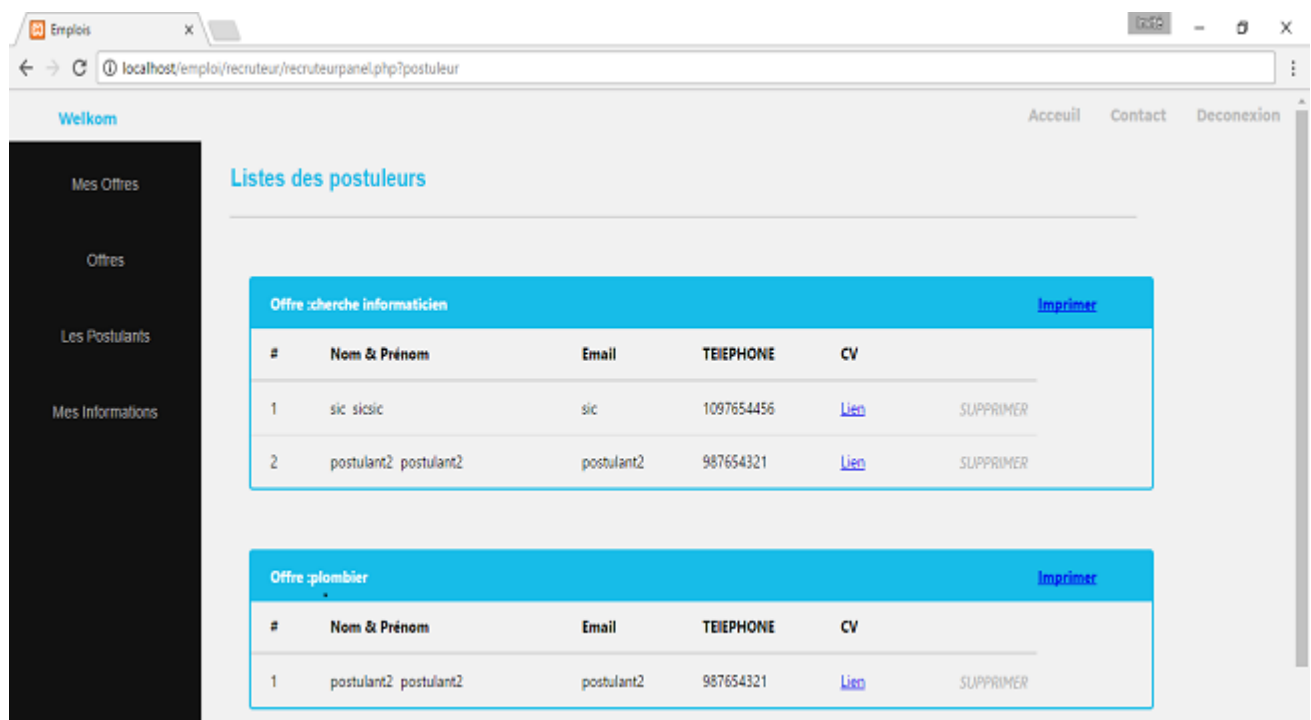


FIGURE 4.16 – Interface d’Affichage des offres et les postulants.

-Cet écran affiche les informations de Recruteur.

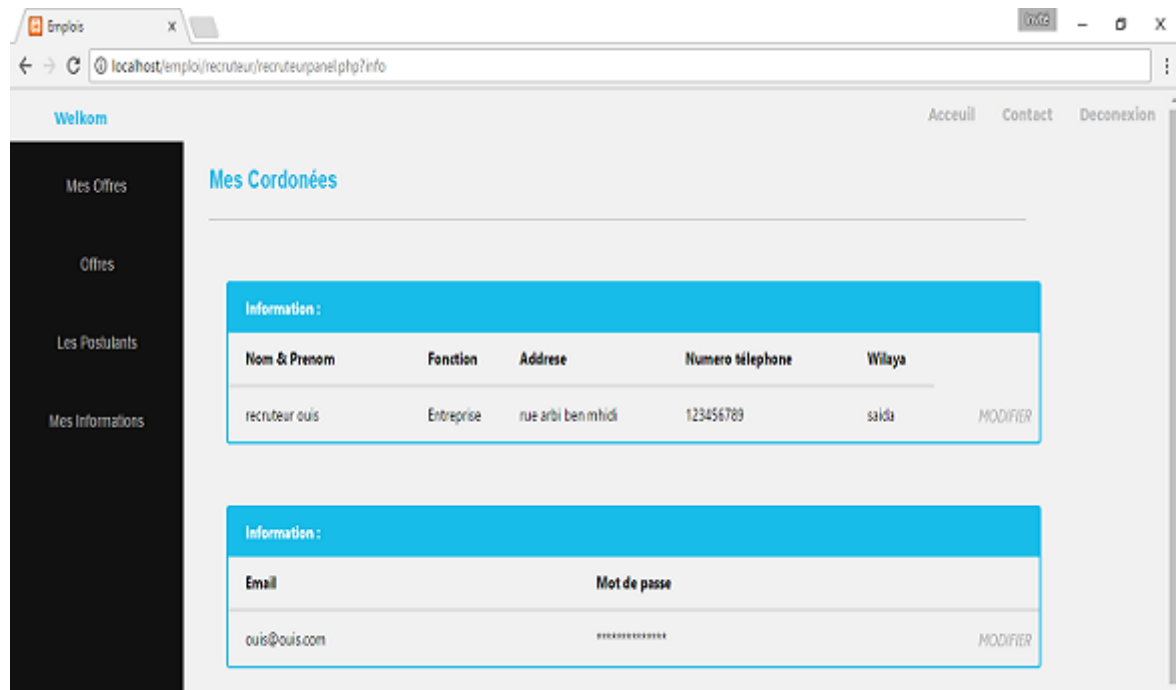


FIGURE 4.17 – Interface information.

Interface postulant :

-Cet écran affiche le cv de postulant

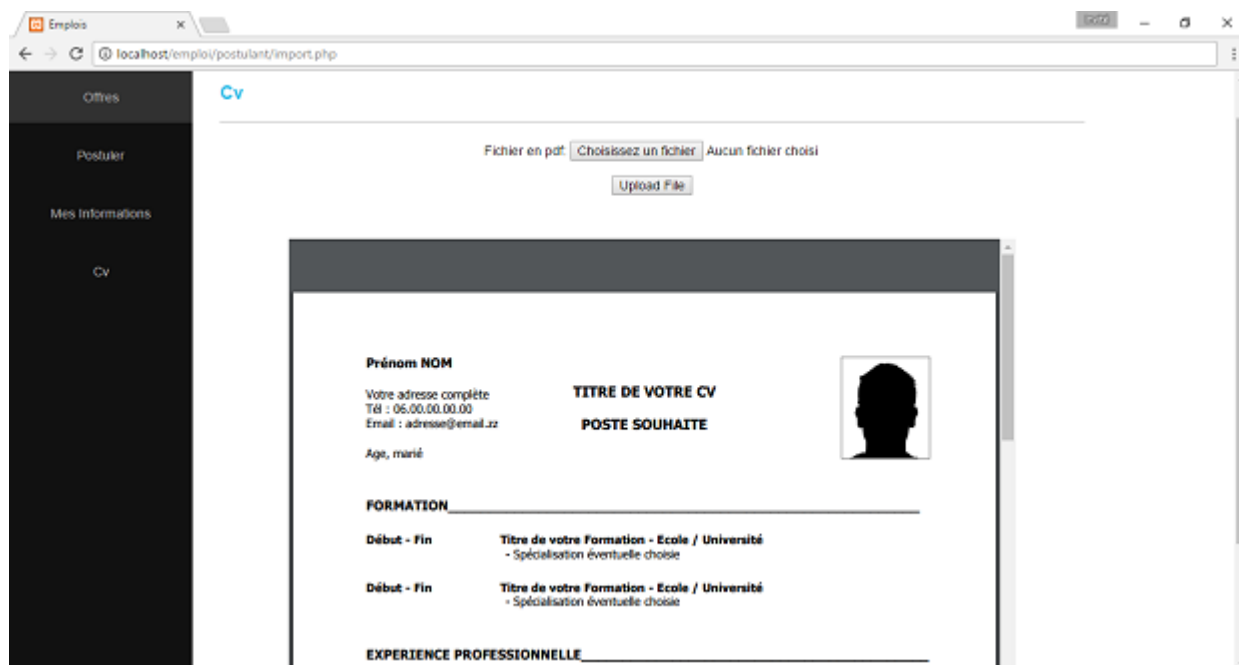


FIGURE 4.18 – Interface d’affichage les cv.

-Cet écran affiche les choix de postulant

The screenshot shows a web application interface. At the top left, there is a blue header with the text 'Welkom'. To the right of this header, there are three links: 'Accueil', 'Contact', and 'Deconnexion'. On the left side, there is a dark vertical sidebar with four menu items: 'Offres', 'Postuler', 'Mes Informations', and 'Cv'. The main content area has a title 'Mes Postules' in blue. Below this title, there is a table with a blue header row labeled 'Offres :'. The table has four columns: '#', 'Titre de l'offre', 'Recruteur', and 'Détail'. There is one data row with the following values: '# 1', 'Titre de l'offre Chef de Projet Informatique', 'Recruteur recruteur recruteur', and 'Détail Voir Plus >>'. To the right of the 'Détail' cell, there is a link 'ANNULER'.

#	Titre de l'offre	Recruteur	Détail
1	Chef de Projet Informatique	recruteur recruteur	Voir Plus >> ANNULER

FIGURE 4.19 – Interface choix de postulant.

4.4 Sécuriser l'application Web :

4.4.1 La sécurité de la base de données :

Pour assurer la sécurité des mots de passe nous avons appliquées la fonction De hachage, car notre requête devra faire la comparaison entre le mot de passe tapé par l'utilisateur et l'empreinte du bon mot de passe qui lui se trouve dans notre base de données.

Il existe plein de fonctions de hachage (MD5, SHA-1, ...), mais certaines d'entre elles ne doivent plus être utilisées même avec une valeur de sel (Le sel est une chaîne aléatoire) .

Le souci est que les fonctions de hachage disponibles sont très rapides .

Les fonctions de hachage ont généralement comme objectif d'être rapide (notamment pour générer des signatures), mais dans le cas d'une authentification, la vitesse est notre ennemi !

Si l'algorithme est rapide, le craquage est rapide ; même si on rajoute un sel, une attaque par dictionnaire peut donc suffire à craquer notre mot de passe (la puissance de calcul actuelle pouvez génère 2,5 millions de SHA-1 en une seconde),

Si nous voulons empêcher l'Attaques de force brute (ou dictionnaire), nous devons utiliser des algorithmes Qui sont lentement calculés.

La solution est avec L'algorithme Bcrypt

Bcrypt [2] est une fonction de hachage basée sur l'algorithme de chiffrement Blowfish a été créé par Niels Provos et David Mazières en 99. Bcrypt est une fonction adaptative (où l'on peut augmenter par exemple les itérations pour augmenter la complexité). Combiné à l'utilisation d'un sel.

Cost	Times / Secondes
10	0.1
11	0.2
12	0.4
13	0.7
14	1.5
15	3.0
16	6.0
17	12
18	24.3
19	48.7
20	97.3
21	194.3
...	...

FIGURE 4.20 – Test temps d'exécution.

On choisira en général un temps d'exécution entre 0,1 s et 0,7 s ; c'est le bon compromis entre sécurité et attente de l'utilisateur.

bcrypt est moins vulnérable aux attaques par recherche exhaustive et cela malgré l'augmentation rapide des puissances de machines.

4.4.2 Le chiffrement des documents Cv :

Le chiffrement est une méthode qui consiste à protéger les documents en les rendant illisibles par toute personne n'ayant pas accès à une clé dite de déchiffrement.

La protection des informations confidentielles a principalement pour but de minimiser les risques pour notre système.

Chiffrement avec l'algorithme Triple DES et le mode CBC

L'algorithme du DES (Data Encryption Standard) :

Le DES a été conçu et développé par IBM avec la participation de la NSA (National Security Agency). Il a été développé par le gouvernement américain comme standard officiel

Cet algorithme consiste à d'utiliser une clé de 64 bits mais qui satisfait la condition suivante : Dans chaque octet, le huitième bit doit être un bit de parité pour rendre le nombre des 1 dans cet octet un nombre impaire, ce qui rend le nombre des clés possible égal à 2^{56}

Les grandes lignes de l'algorithme sont les suivantes :

- Fractionnement du texte en blocs de 64 bits (8 octets);
- Permutation initiale des blocs;
- Découpage des blocs en deux parties : gauche et droite, nommées G et D;
- Etapes de permutation et de substitution répétées 16 fois (appelées rondes);
- Recollement des parties gauche et droite puis permutation initiale inverse.

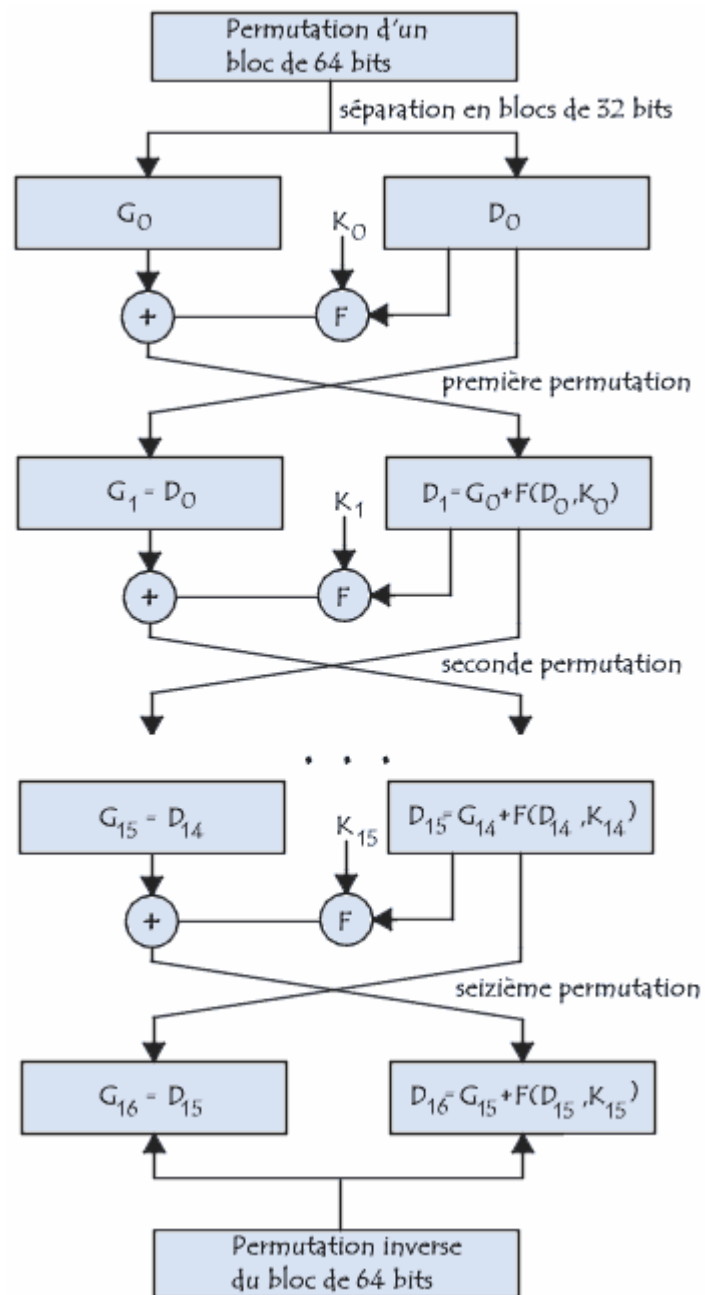


FIGURE 4.21 – Algorithme du DES.

Génération des clés :

Toute la sécurité de l'algorithme du DES repose sur la complexité des clés de chiffrement.

L'algorithme ci-dessous montre comment obtenir à partir d'une clé de 64 bits (composé de 64 caractères alphanumériques quelconques) 16 clés diversifiées de 48 bits chacune servant dans l'algorithme du DES :

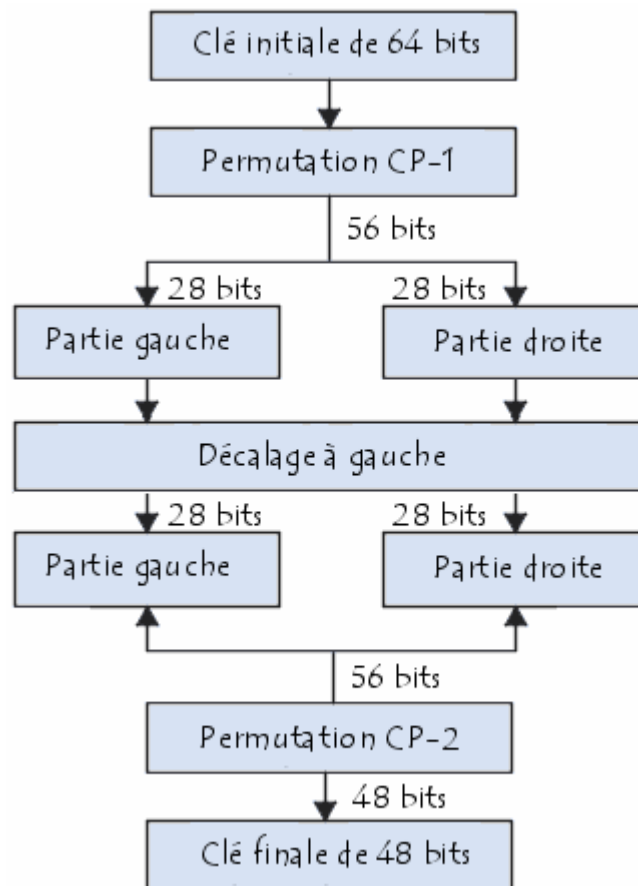


FIGURE 4.22 – Génération des clés.

Triple DES (Triple Data Encryption Standard) :

IBM a conçu un moyen pour augmenter la taille du DES car elle a été considérée courte. La méthode utilise 3 étapes et 3 clés.

Le Triple DES [6] permet d'augmenter significativement la sécurité du DES

Le fonctionnement est le suivant :

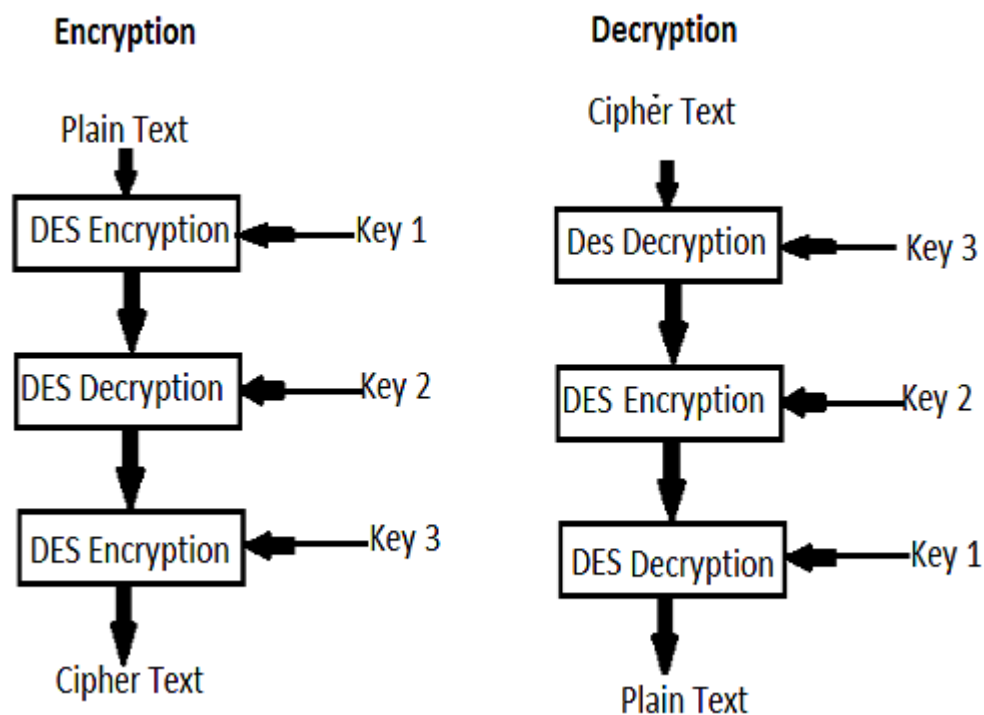


FIGURE 4.23 – Triple DES.

Modes de chiffrement :

Une donnée traitée par le Triple-DES est une donnée de 64 bits, pour chiffrer une Donnée de plus grande taille, on se mène à le décomposer en plusieurs messages successives chacun de 64bits. Il existe divers modes de décomposition disponibles (ECB ,OFB ,CFB ,CBC,.....).

Cipher-Block Chaining (CBC) :

CBC est le mode le plus utilise.Invente par IBM en 1976, où chaque bloc du message en clair est XORé avec le bloc chiffré précédent avant d'être chiffré. Pour cela, chaque bloc chiffré dépend de tous les blocs du message en clair qui le précèdent. Pour rendre chaque bloc unique, on se mène d'un vecteur d'initialisation qui sera utilisé avec le premier bloc.

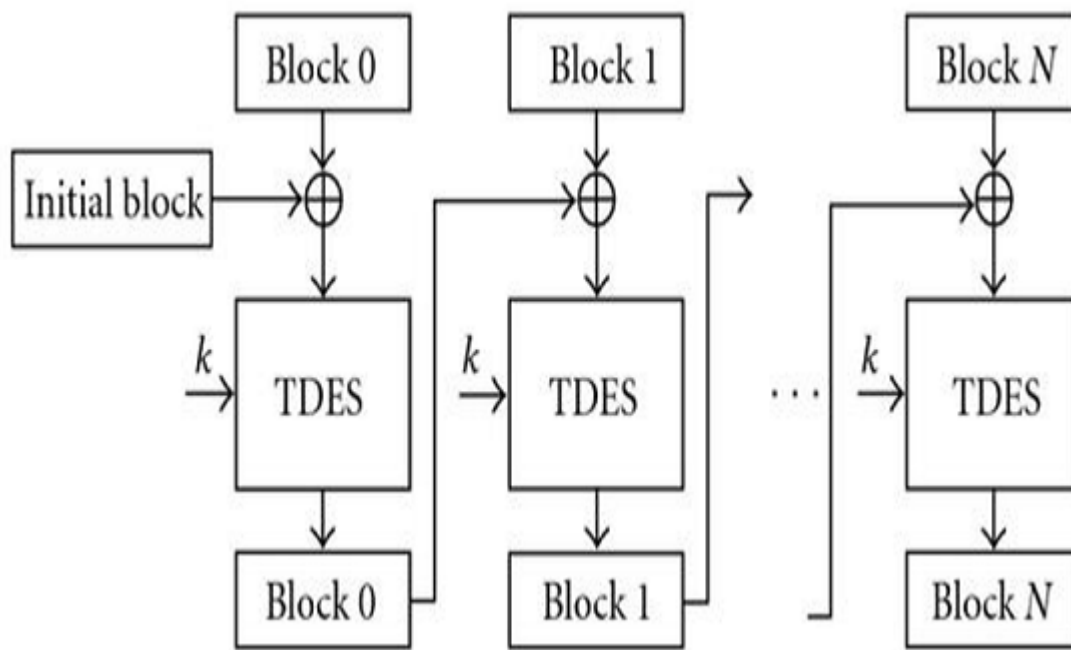


FIGURE 4.24 – Mode CBC.

Pour le chiffrement :

$$C_i = E_K(P_i \oplus C_{i-1}),$$

– $C_0 = IV$

Pour le déchiffrement :

$$P_i = D_K(C_i) \oplus C_{i-1},$$

– $C_0 = IV$.

4.4.3 L'application d'une attaque SQL injection sur notre système :

L'attaque sur le formulaire d'une connexion :

Nous allons montrer comment appliquer l'attaque SQL injection sur le formulaire de connexion.

Exemple1 : Si nous rentrons ceci : `1' OR '1' = '1`, la requête devient :

Query="SELECT *FROM utilisateurs WHERE Pseudo = '1' OR '1'='1' AND password = '1' OR '1'='1'";

Ainsi, on peut se loguer sans connaître le pseudo et le mot de passe, puisque la condition `'1'='1'` est toujours vérifiée.

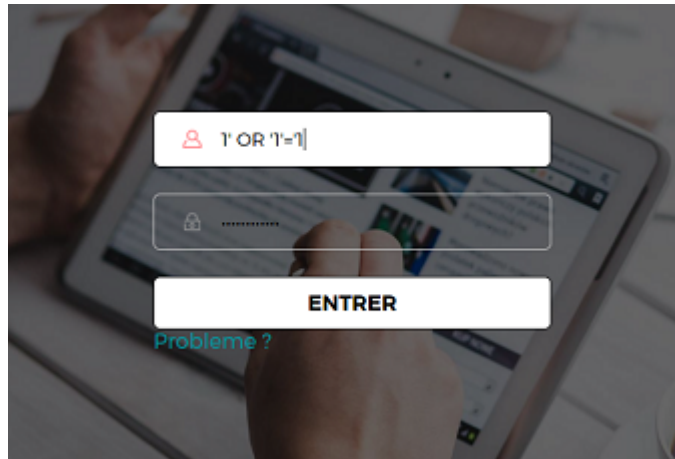


FIGURE 4.25 – L'application d'une attaque SQL.

Je vous ai parlé ici des injections par formulaire, mais le principe est le même pour toutes requête SQL. Ainsi une requête qui récupère un paramètre dans l'URL avec **GET** sera également vulnérable.

4.4.4 L'application d'une attaque xss (Cross-Site Scripting) sur notre systeme :

Une alerte JS par exemple :

```
<script>alert('Il y a une faille XSS')</script>
```

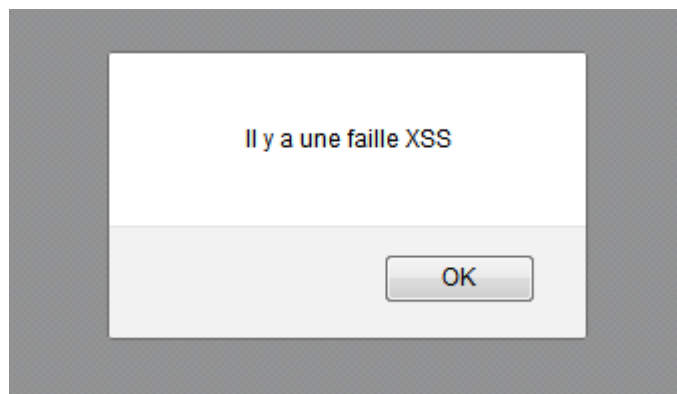


FIGURE 4.26 – L'application d'une attaque xss.

4.4.5 L'attaque par force brute :

Le bannissement d'IP Elle consiste à limiter le nombre de tentatives par personne et par jour. Bien évidemment, il ne faut pas bloquer le compte visé par l'attaque, mais empêcher le hacker de continuer à brutaliser notre formulaire.

L'astuce consiste donc à enregistrer toutes les tentatives ratées de connexions au site. On y enregistrera simplement l'IP de la personne. Au-delà d'un certain nombre de tentatives, l'accès au compte avec cet IP devient impossible pendant un certain temps.

```
public function get_ip()
{
    if(isset($_SERVER['HTTP_X_FORWARDED_FOR']))
    {
        $ip = $_SERVER['HTTP_X_FORWARDED_FOR'];
    }
    elseif(isset($_SERVER['HTTP_CLIENT_IP']))
    {
        $ip = $_SERVER['HTTP_CLIENT_IP'];
    }
    else
    {
        $ip = $_SERVER['REMOTE_ADDR'];
    }

    return $ip;
}
```

FIGURE 4.27 – Le bannissement d'IP.

4.4.6 La sécurité contre les attaques :

Après la détection d'attaque notre système va bloquer l'utilisateur c'est-à-dire que ce dernier ne peut pas poursuivre ses recherches ou et finir son rôle par l'affichage d'un message d'alerte comme le montre la fenêtre de la Figure

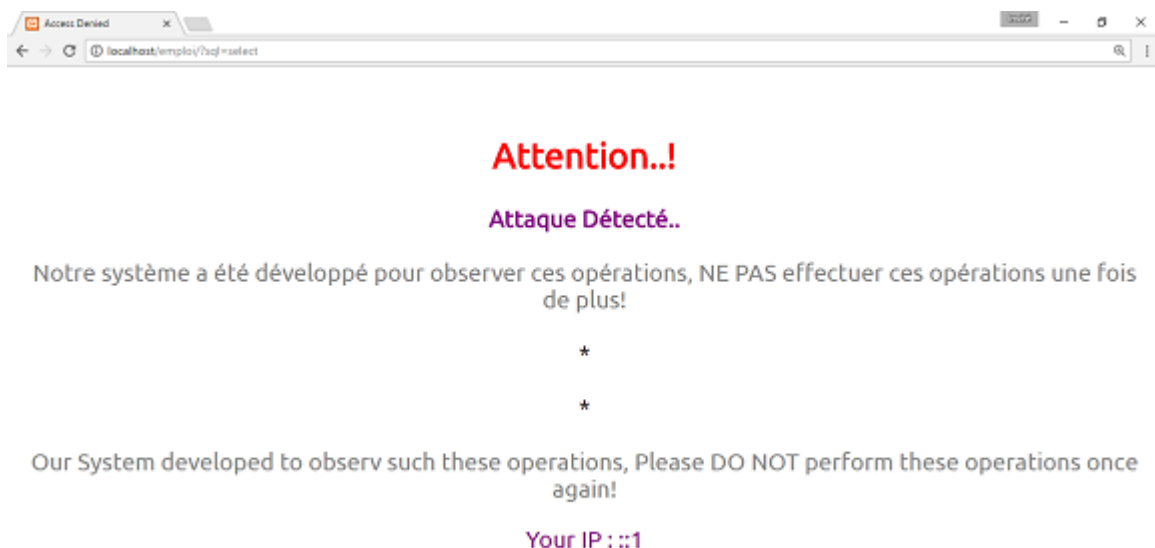


FIGURE 4.28 – Message d'alerte corresponds à la détection.

4.5 Conclusion :

Dans ce chapitre, nous avons décrit le processus de réalisation de notre application en spécifiant les outils de développement, l'implémentation de la base des données et la démarche suivie pour la réalisation.

En effet, nous avons implémenté et testé les techniques de sécurité avec les différents aspects et nous avons réalisé la sécurité sur notre application contre les types d'attaque utilisés.

Conclusion Générale

Durant ce projet, Nous étions chargés de la conception et la réalisation d'un système de gestion d'offres d'emploi sécurisé. Comme nous venons de le voir, la mise en place de ce système n'est pas forcément complexe, mais, elle exige tout de même qu'on suive une démarche structurée et rigoureuse.

L'utilisation du système d'offres d'emploi apporte des solutions nouvelles aux recruteurs comme aux postulants. Les premiers peuvent aujourd'hui recevoir des cv sans avoir à ouvrir les enveloppes. Par la suite, l'envoi des réponses, et le dialogue avec le postulant s'effectuent aussi électroniquement. Ceci peut diversifier leurs choix, Les postulants quant à eux, ont la possibilité de chercher un emploi quand et où ils le désirent.

Pour assurer la confidentialité et l'intégrité des informations des différents utilisateurs (recruteur et postulant), nous avons eu recours aux différentes méthodes de sécurité que nous avons vues durant notre formation ce qui nous a aidé à mieux comprendre l'importance de ce domaine. Enfin l'application reste ouverte à toute évolution ou proposition pour son amélioration.

[5] [4] [13] [3] [11] [15] [7] [Reference8] [16] [9]
[2] [1] [6] [10] [14] [8] [12]

Bibliographie

- [1] « bcrypt ». In : <https://www.bcrypt.fr/explications>.
- [2] « bcrypt algorithm ». In : <http://static.usenix.org/events/usenix99/provos.html>.
- [3] D. Nickull J. Governor et D. HINCHCLIFFE. In : *Web 2.0 Architectures*. 2009, 248p.
- [4] S. GULATI. In : *Under the hood of the Internet : Connector : the Internet protocol*.
- [5] Guillaume HARRY. In : *Faibles de sécurité des applications Web Principes, parades et bonnes pratiques de développement*. 2012.
- [6] H. KUMMERT. *The PPP Triple-DES Encryption Protocol*.
- [7] « Les dix risques de sécurité applicatifs web les plus critiques 2017, OWASP. » In : https://www.owasp.org/index.php/Main_page.
- [8] Josh LOCKHART. *PHP : The Right Way*.
- [9] « Notepad ». In : <https://notepad-plus-plus.org/fr/>.
- [10] « Protégez-vous efficacement contre les failles web ». In : <https://openclassrooms.com>.
- [11] L. Shklar et R. ROSEN. In : *Web Application Architecture : Principles, Protocols and Practices*. John Wiley Sons Ltd. 2003, 372p.
- [12] Chris SHIFLETT. *Essential PHP Security*.
- [13] G.Florin et S.NATKIN. In : *Support de cours RESEAUX ET PROTOCOLES*. 2007, 884p.
- [14] UML. <https://www.draw.io/>.
- [15] « WASC (Web Application Security Consortium) est un groupe international d'experts, de spécialistes de l'industrie et de représentants d'entreprises, qui élabore des normes de sécurité source ouverte et des bonnes pratiques largement reconnues pour le World Wide Web. » In : <http://www.webappsec.org/>.
- [16] « XAMPP ». In : <https://www.apachefriends.org/fr/index.html>.