



Département : Mathématiques & Informatique

Master Cybersécurité & cybercriminalité

Compte Rendu

Mini-Projet : Réalisation d'une application Python pour la gestion d'une « Agence de recrutement virtuel »

Par : Nihal Akerdad

Zineb Louzi

Achraf Iabiten

Encadré par : Mr m.alami

Mme j.jebrane

Fais le 25/02/2020

Index

I-INTRODUCTION ET PROLEMATIQUE

1-INTRODUCTION

2-PROBLEMATIQUE

II-MISE EN PLACE DE L'INFRASTRUCTURE

1-SERVEURS WEB

APACHE

PHPMyADMIN

2-LANGAGE DE PROGRAMMATION

PYTHON

HTML

3-FRAMEWORK DJANGO

III-TUTORIEL PROJET

BACK-END

FRONT-END

IV-CONCLUSION

I. Introduction et problématique

1- Introduction

Le recrutement est une étape fondamentale et incontournable pour chaque organisme professionnel, il s'agit d'un ensemble d'actions mises en œuvre pour trouver un candidat correspondant aux besoins d'une organisation pour un poste donné.

De nos jours, les recruteurs professionnels sont, depuis le début des années 2000, confrontés à un bouleversement sans précédent des moyens à leur disposition pour recruter leurs cadres.

Ces mutations ont des causes auto-entretenu : la pénurie de cadres et les potentialités de l'internet ont favorisé l'émergence d'une nouvelle offre de services, que les acteurs nouveaux et traditionnels se sont efforcés de porter en direction des entreprises.

Le e-recrutement est aujourd'hui assez économe en moyens, il permet avec ciblage, souplesse et réactivité de conquérir et préqualifier des profils en offrant de bons rendements quantitatifs.

2- Problématique

Le recrutement est bien plus qu'un process, c'est devenu aussi un métier de relations, tel que l'activité commerciale il y a quelques années. Ce métier pourra donc, lui aussi, bénéficier de l'utilisation du digital pour optimiser les démarches de recrutement et mieux gérer la mobilité interne.

Afin de répondre à cette problématique, nous avons choisi d'implémenter une application Web à l'aide du framework DJANGO, pour la gestion d'un « Agence de Recrutement Virtuel » en se basant sur le langage de programmation PYTHON.

II. Mise en place de l'infrastructure

1- Serveur web

i) Apache

Le logiciel libre Apache HTTP Server est un serveur http créé et maintenu au sein de la fondation Apache. C'est le serveur HTTP le plus populaire du World Wide Web. Il est distribué selon les termes de la licence Apache.



ii) Phpmyadmin

PhpMyAdmin est une application Web de gestion pour les systèmes de gestion de base de données MySQL réalisée principalement en PHP et distribuée sous licence GNU GPL.



2- Langage de programmation

i) Python

Python est un langage de programmation interprété, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet.



ii) HTML

Le HyperText Markup Language, généralement abrégé **HTML** ou dans sa dernière version HTML5, est le langage de balisage conçu pour représenter les pages web.



3- Framework : Django



Django est un framework web Python de haut niveau qui encourage le développement rapide et propre. Gratuit et open source, Django vous permet d'éviter de réinventer la roue grâce à toutes les libraires disponibles en Python, mais aussi de tout ce que ce framework offre dès son installation.

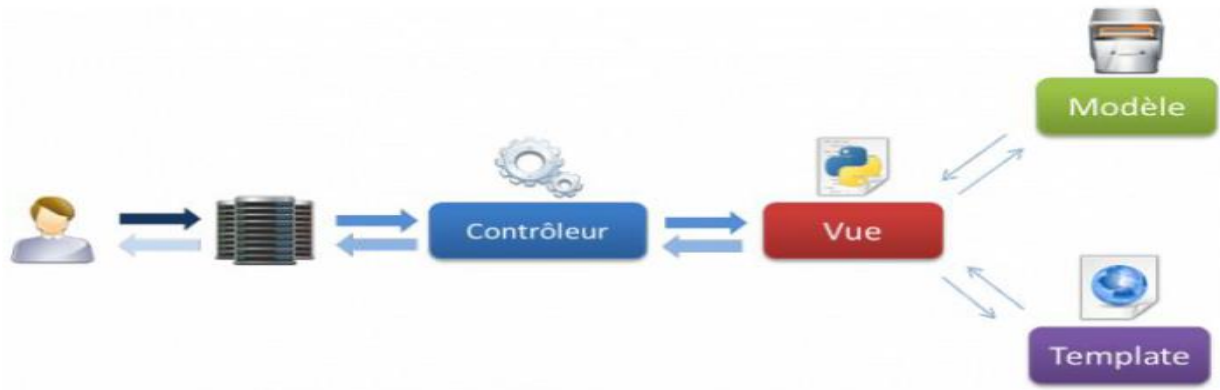
Le fonctionnement d'une application Django se divise en 2 parties.

La première section a pour but de préparer l'étape entre l'utilisateur et l'application en elle même. Cette section gère la relation entre les bases de données et les applications Django, mais aussi s'occupe du routage via des règles URL.

La seconde section est l'application. Organisée en modèles, vues et templates, l'application se trouve au cœur du projet Django.

Utilisé par la NASA ou le Washington Post, Django prouve sa fiabilité et sa stabilité à travers ce type d'organisme.

Django se base sur le modèle MVT qui représente une **architecture** orientée autour de trois pôles : le **modèle**, la **vue** et le **template**. Elle s'inspire de l'architecture MVC, très répandue dans les frameworks web. Son objectif est de séparer les responsabilités de chaque pôle afin que chacun se concentre sur ses tâches.



- Le cycle de vie d'une requête traitée par Django

Django est un framework Web, le serveur lui fait passer chaque requête GET, POST (ou autres) envoyée par les clients. Django, lui, génère un contenu pour chacune d'elle (souvent du HTML), et le renvoie au serveur qui renvoie la réponse au client. En détail, ça donne la figure ci-dessous.

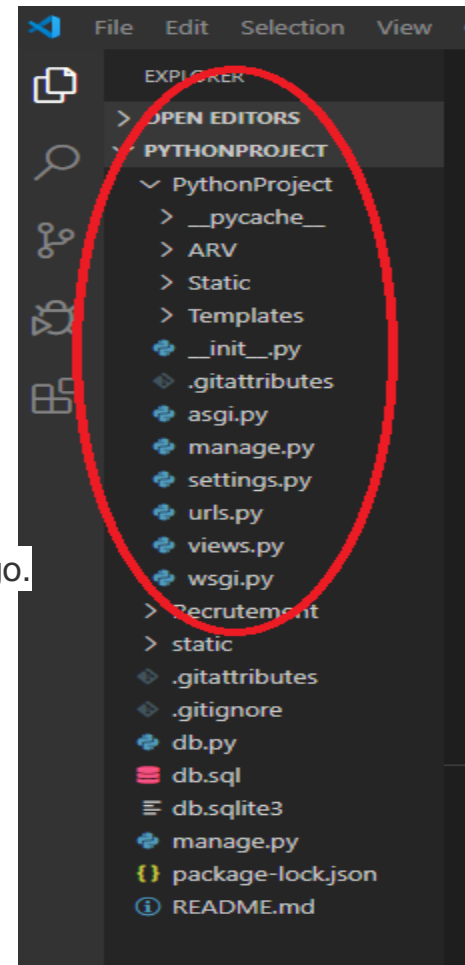
La requête arrive, on prend un template, on dit à Django de mettre des données récupérées depuis la base de données dans le template, on génère du HTML, on retourne la réponse.

Le reste, ce sont juste des outils pour automatiser toutes les variantes de ce cycle: formulaire, flux RSS, gestion de droits, etc. Mais tout tourne TOUJOURS autour de la logique requête/réponse car le but est de faire du Web, et que le Web marche comme cela.


```
Administrateur : Invite de commandes
C:\xampp\htdocs>django-admin startproject PythonProject
```

Ce qui va nous générer les fichiers suivants :

- **__init__.py** : permet au dossier qui le contient de pouvoir être traité comme un packet python.
- **manage.py** : un utilitaire en ligne de commande qui vous permet d'interagir avec ce projet Django de différentes façons. .
- **urls.py** : contient les routes du projet: il dispatche les requêtes vers les contrôleurs adéquats.
- **settings.py** : réglages et configuration de ce projet Django.



Etape 2 : Dans « SETTINGS.PY »

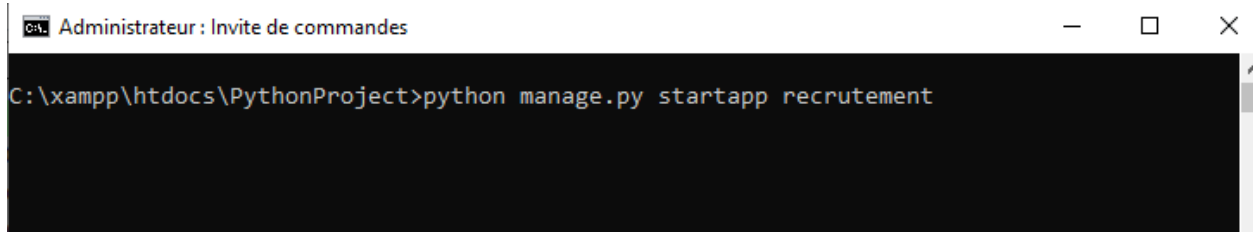
➔ Configurer l'accès à sa base de données

```
settings.py X
PythonProject > settings.py > ...
/4
75 WSGI_APPLICATION = 'PythonProject.wsgi.application'
76
77
78 # Database
79 # https://docs.djangoproject.com/en/3.0/ref/settings/#databases
80
81 DATABASES = {
82     'default': {
83         'ENGINE': 'django.db.backends.mysql',
84         'NAME': 'recrutement',
85         'USER': 'root',
86         'PASSWORD': '',
87         'HOST': '127.0.0.1',
88         'PORT': '3306',
89     }
90 }
91
92 # Password validation
93 # https://docs.djangoproject.com/en/3.0/ref/settings/#auth-password-validators
94
95 AUTH_PASSWORD_VALIDATORS = [
96     r
```

➔ Indiquer le chemin de votre dossier templates dans le fichier DIRS

```
settings.py X
PythonProject > settings.py > ...
55 # Template directory setting
56 TEMPLATE_DIRS = (
57     os.path.join(os.path.dirname(__file__), 'Templates'),
58 )
59 TEMPLATES = [
60     {
61         'BACKEND': 'django.template.backends.django.DjangoTemplates',
62         'DIRS': ['C:\\xampp\\htdocs\\PythonProject\\PythonProject\\Templates'],
63         'APP_DIRS': True,
64         'OPTIONS': {
65             'context_processors': [
66                 'django.template.context_processors.debug',
67                 'django.template.context_processors.request',
68                 'django.contrib.auth.context_processors.auth',
69                 'django.contrib.messages.context_processors.messages',
70             ],
71         },
72     },
73 ]
74
```

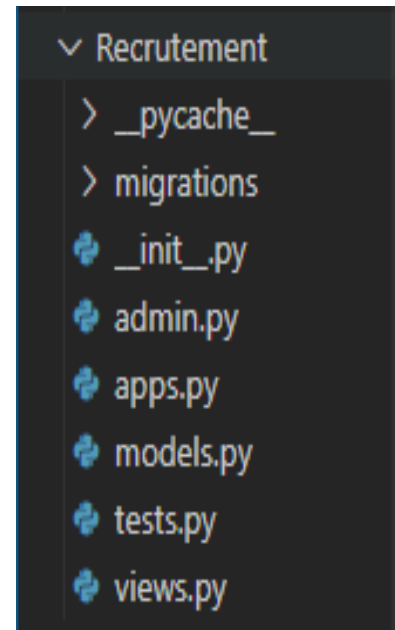
Etape 3 : création de l'application « recrutement »



```
Administrateur : Invite de commandes
C:\xampp\htdocs\PythonProject>python manage.py startapp recrutement
```

Ce qui va nous générer, à son tour, un ensemble de fichiers :

- **admin.py** : va permettre de définir ce que souhaitez afficher et modifier dans l'administration de l'application.
- **Models.py** : contiendra vos modèles
- **Tests.py** : permet la création de tests unitaires.
- **Views.py** : contiendra toutes les vues de votre application.

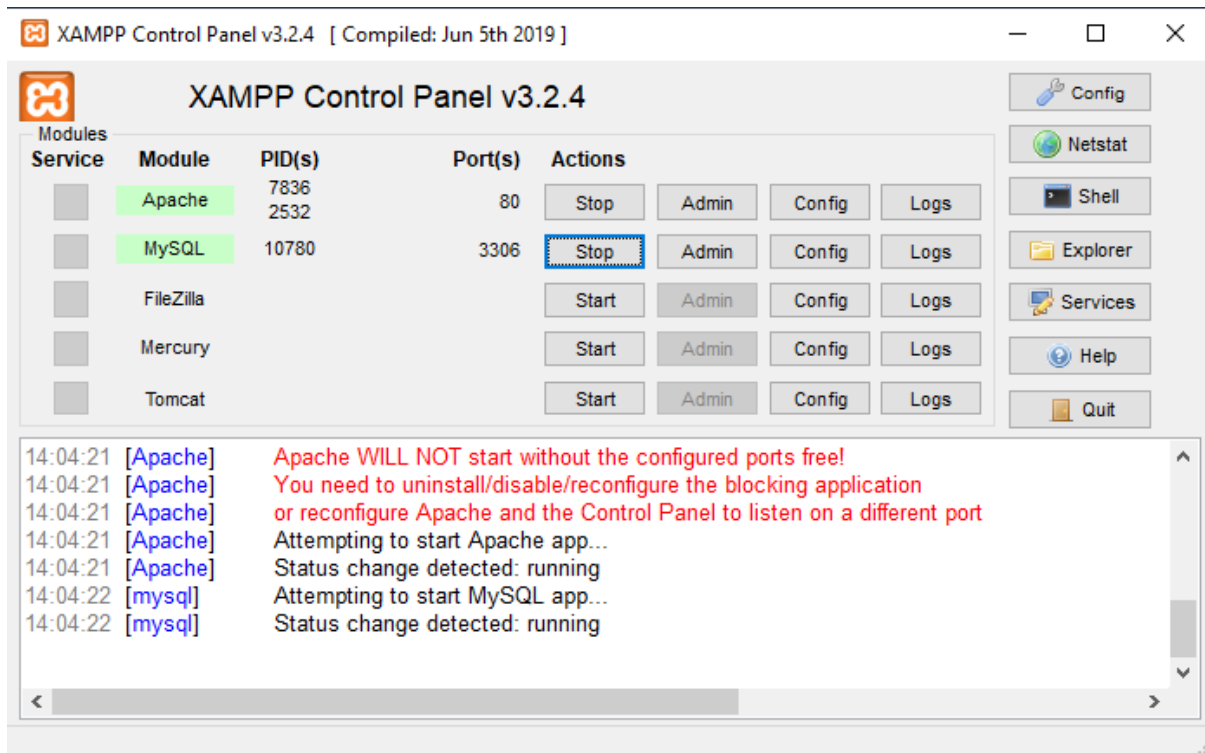


2- Front-end

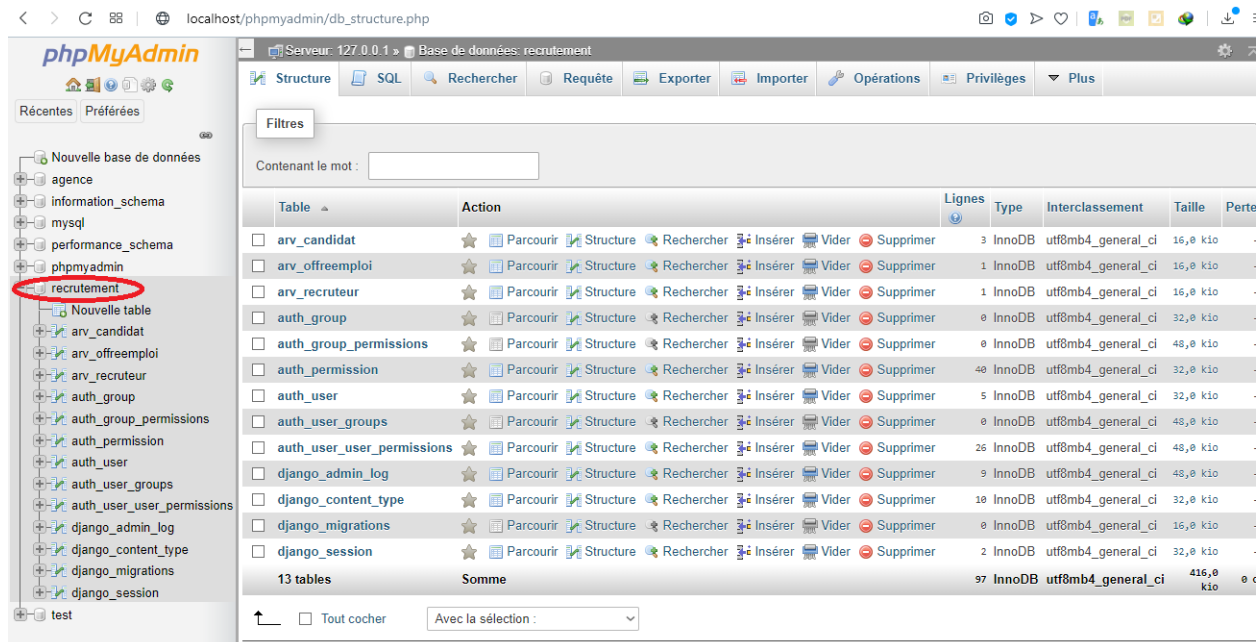
i) Lancement du serveur

Etape 1

→ Commençons par ouvrir XAMPP et lancer APACHE et MYSQL

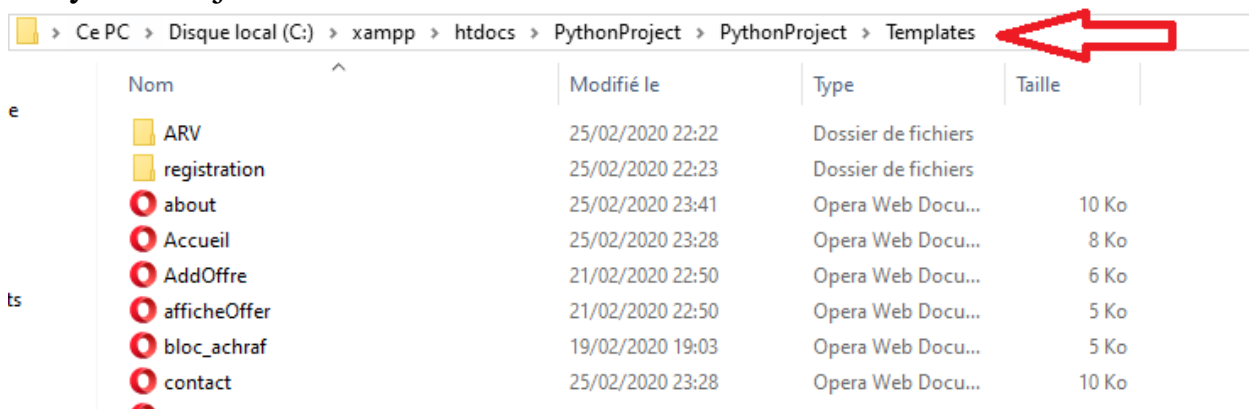


→ Ensuite, ouvrir PHPMYADMIN sur notre navigateur avec l'adresse <http://localhost:8080/phpmyadmin>, puis, créer une base de donnée intitulée « recrutement », et importer le fichier recrutement.sql .



Etape 2

➔ Copier le lien du dossier « Templates » qui se trouve dans le dossier « PythonProject »



➔ Le mettre dans le Script « setting » qui se trouve dans le dossier C:\xampp\htdocs\PythonProject\PythonProject

```

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': ['C:\xampp\htdocs\PythonProject\PythonProject\Templates'],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

```

→ Ouvrir la CMD et accéder au dossier « PythonProject »

```

Administrateur : Invite de commandes
Microsoft Windows [version 10.0.18362.657]
(c) 2019 Microsoft Corporation. Tous droits réservés.

C:\Windows\system32> cd C:\xampp\htdocs\PythonProject

C:\xampp\htdocs\PythonProject>

```

→ Lancer le serveur émulé dans notre machine locale avec la commande

`.\manage.py runserver`

```

Administrateur : Invite de commandes - .\manage.py runserver

C:\xampp\htdocs\PythonProject>.\manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

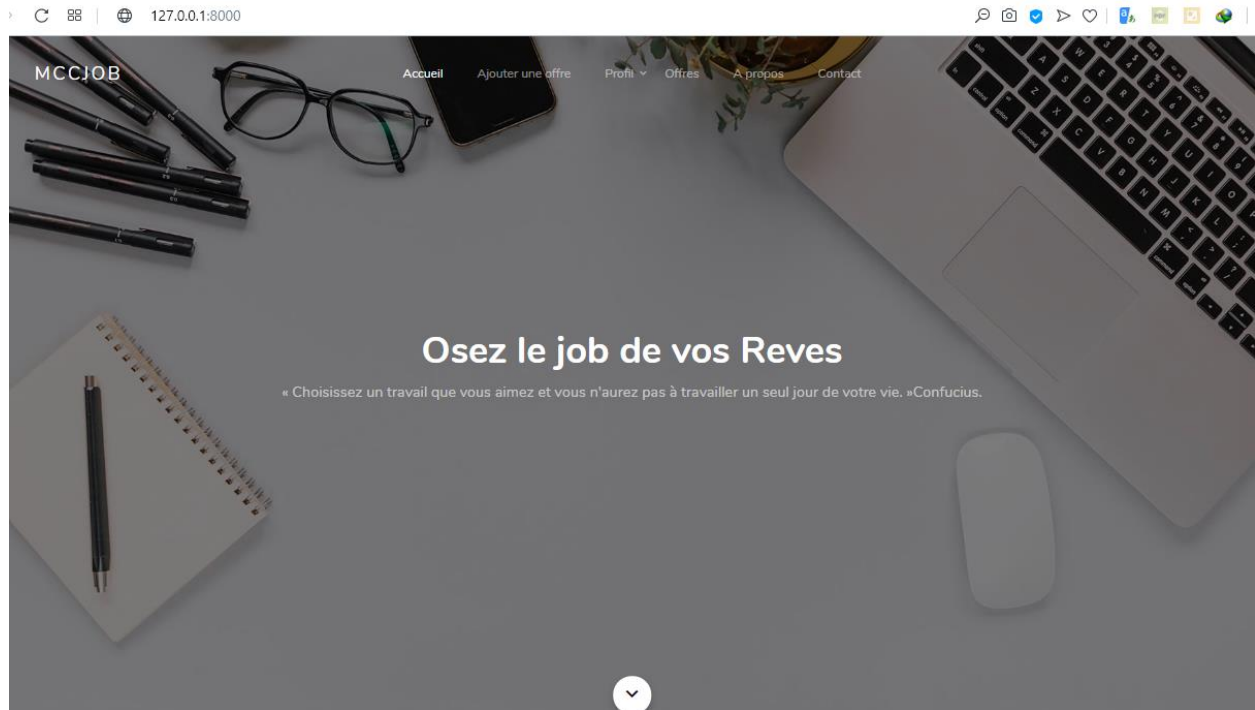
You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): Recrutement, admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
February 26, 2020 - 21:41:29
Django version 3.0.3, using settings 'PythonProject.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

```

EH VOILA ! maintenant Django est lancé, à ce stade vous pouvez s'amuser à naviguer avec toute liberté sur notre application, via l'adresse <http://127.0.0.1:8000>

ii) Application Web

➔ Page « Accueil » :



➔ Page « Login » : vous pouvez vous connecter en utilisant

Id : niha1

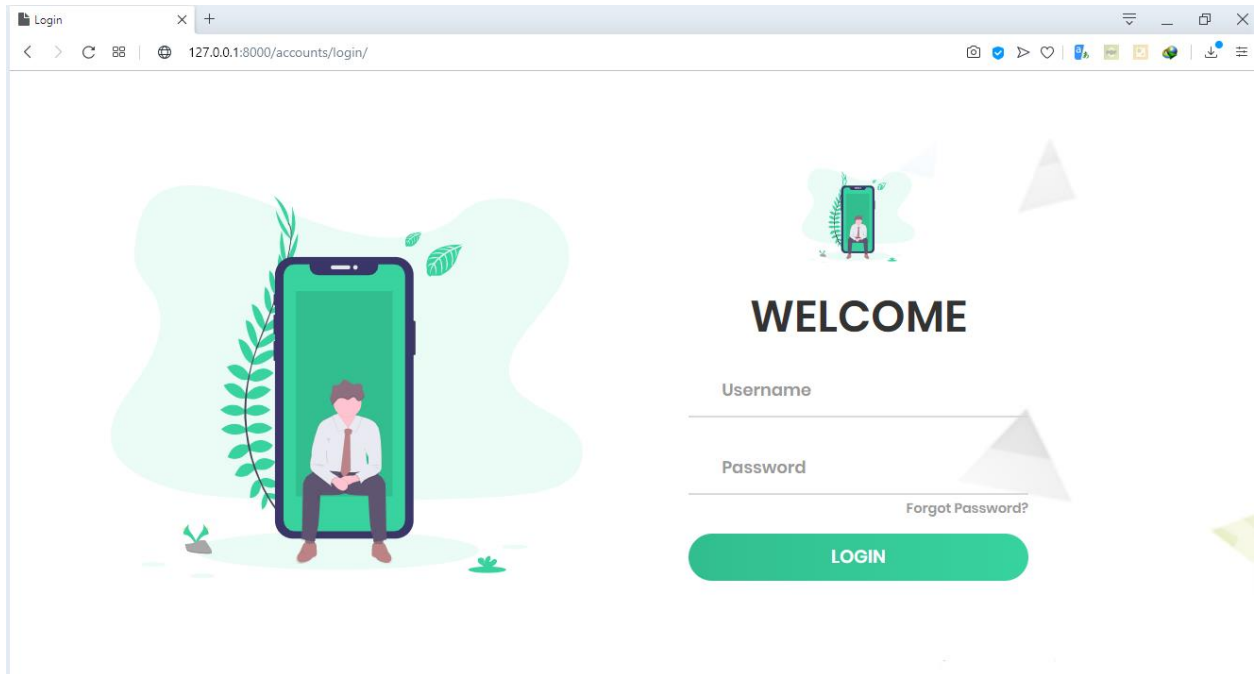
password : candidat1

Id : zineb

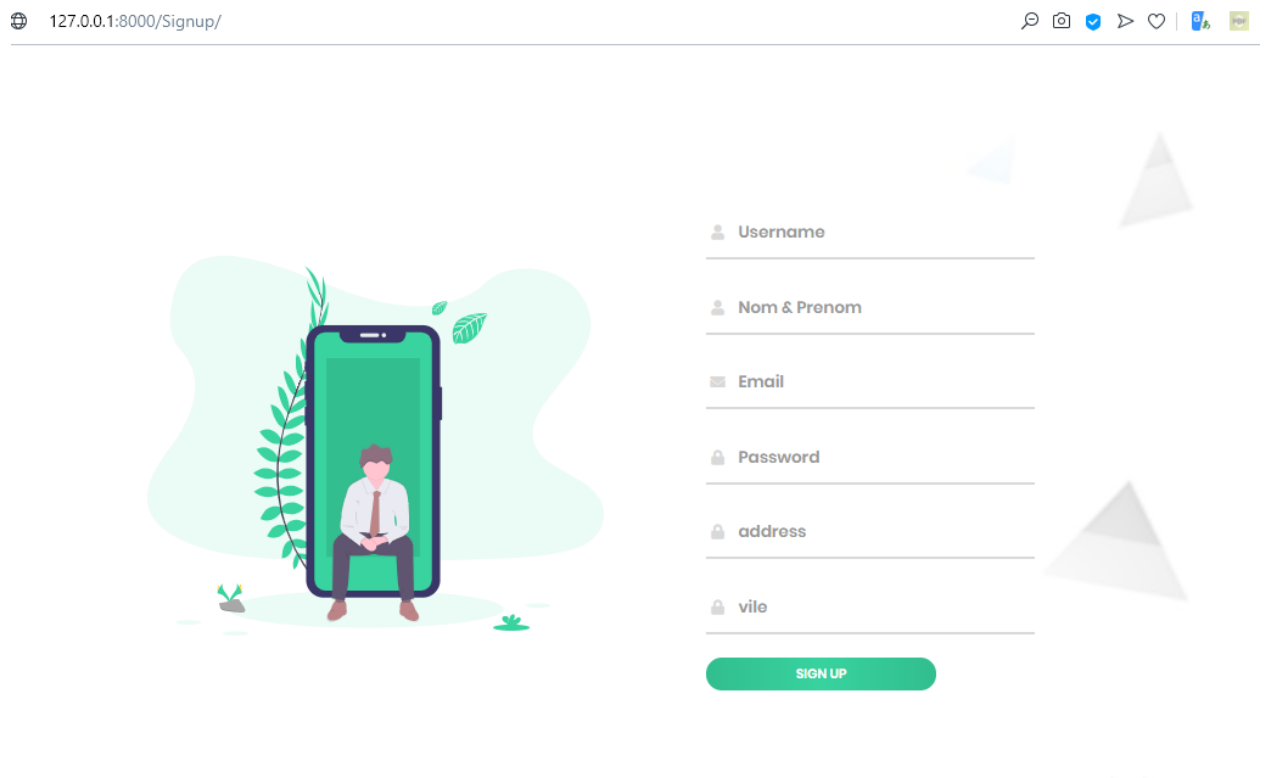
password : candidat2

Id : achraf

password candidat3

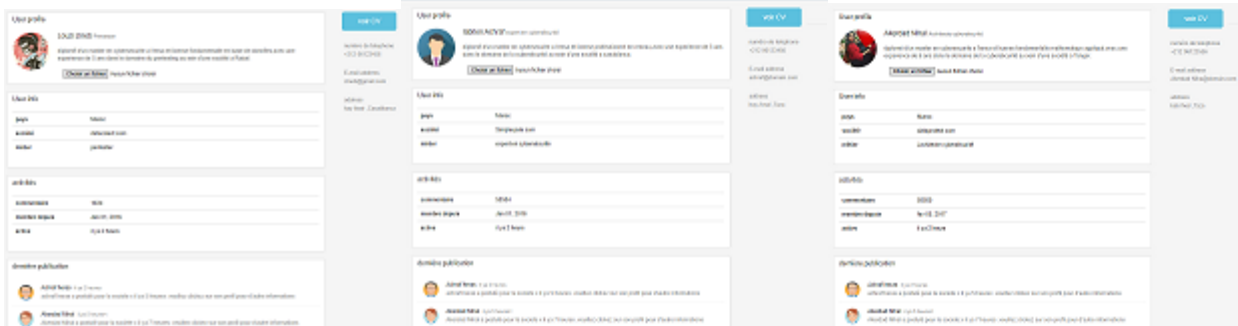


➔ Page « SignUp » : Inscrivez vous



➔ Page « Profil » :

- Afin de consulter un profil de nos candidats, vous pouvez accéder à cette page

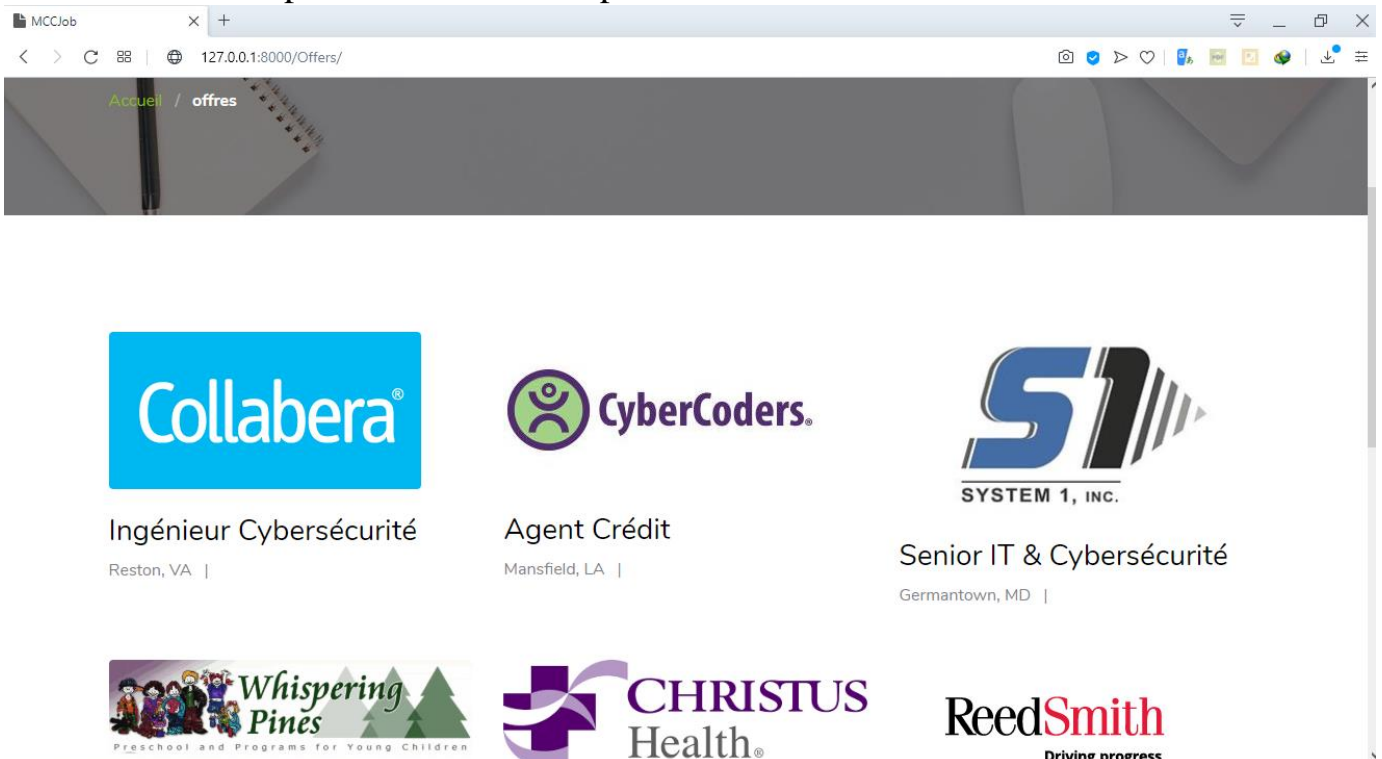


- Vous pourrez même voir leurs CVs, en cliquant sur « voir cv »




➔ Page « Offres » :

- ❖ Vous pouvez voir les multiples offres dans notre site



- ❖ Vous pouvez consulter le contenu de chaque offre, en cliquant sur l'offre, vous serez capable de voir : la description de l'entreprise, la localisation de l'emploi, les responsabilités, la formation et l'expérience requise...

Exemple :

 **CyberCoders.**

**Ingénieur Audit
Cybersécurité**
CyberCoders Rabat, Technopolis

[Enregistrer
Emploi](#)

[Postuler](#)



Description de l'Entreprise

CyberCoders Group est un concepteur, intégrateur et opérateur de systèmes critiques intelligents cyberprotégés. En proposant des solutions innovantes dans les domaines de la cyber sécurité, des systèmes sûrs industriels et embarqués, de la continuité numérique et de la gestion de crise et de la

Responsabilités


- Analyse des besoins,
- Participation à la conception,
- Spécifications techniques et/ou fonctionnelles,
- Développement de nouvelles solutions complexes en JAVA/J2EE,
- Evolution et maintenabilité du code,
- tests.

Education + Experience

- De formation Bac+4/5 en informatique.
- Une expérience de 4 ans minimum sur un environnement JAVA/J2EE..
- Expérience d'au moins 6 mois sur le Framework JS (AngularJS, React, Ember).
- Frameworks associés du type Struts, Spring, Hibernate,Maven.

Autres

- souple
- en capacité d'identifier les solutions techniques appropriées.
- créatif(ve)
- Polyvalent
- bonnes capacités d'adaptation pour répondre aux demandes des clients dans le respect des normes de développement.

 Enregistrer

Postuler

➔ Page « A Propos »

Notre Equipe



Nihal Akerdad

Architecte Cybersécurité

Architecte cybersécurité et cybercriminalité depuis 2015, lauréate de l'ENSAT, master CyberSécurité et Cybercriminalité



Zineb Louzi

Expert Cybersécurité

Experte en cybersécurité et cybercriminalité depuis 2015, lauréate de l'ENSAT, master CyberSécurité et Cybercriminalité



Achraf labiten

Mobile Developer

Développeur applications mobile depuis 2015, lauréat de l'ENSAT, master CyberSécurité et Cybercriminalité



➔ Page « Contact »

Vous pouvez contacter notre équipe via cette page

127.0.0.1:8000/ContactUs/

MCCJOB

Home Ajouter une offre Profil Offres A propos Contact

Contact Us

Accueil / Contact Us

Prénom Nom

Email

Sujet

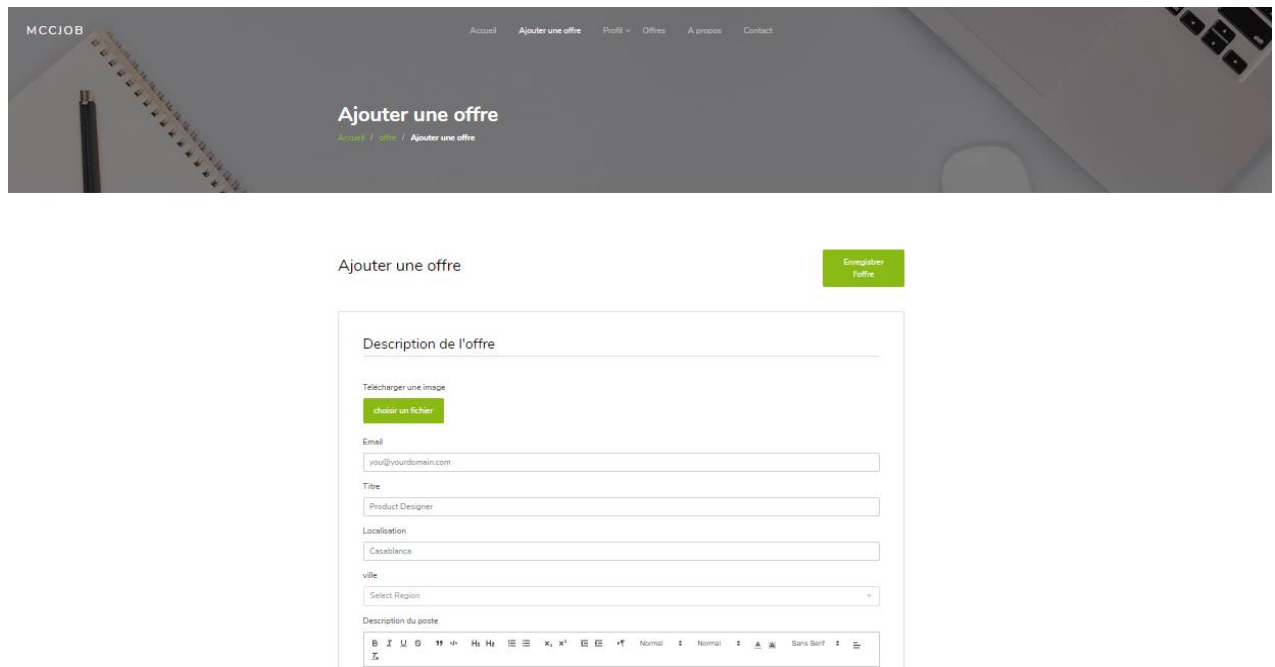
Message
Write your notes or questions here...

Adresse
23 Rue ifane 2, Boukhalaf Tanger

Téléphone
+212 6 88115046

Adresse Email
MCSC@Ensa.com

➔ Page « Ajouter une Offre »



The screenshot shows the 'Ajouter une offre' page on the MCCJOB website. The header includes the MCCJOB logo and navigation links: Accueil, Ajouter une offre, Profil, Offres, A propos, and Contact. The main heading is 'Ajouter une offre' with a breadcrumb trail: Accueil / offre / Ajouter une offre. Below the heading, there is a green button labeled 'Enregistrer l'offre'. The form itself is titled 'Description de l'offre' and contains the following fields: 'Télécharger une image' with a 'Choisir un fichier' button, an 'Email' field with the placeholder 'you@yourdomain.com', a 'Titre' field with the placeholder 'Product Designer', a 'Localisation' field with the placeholder 'Casablanca', a 'ville' dropdown menu with 'Select Region' selected, and a 'Description du poste' field with a rich text editor toolbar. The toolbar includes buttons for bold, italic, underline, link, unlink, list, and text color, along with a font family dropdown set to 'Sans Serif'.

En remplissant le formulaire ci-dessus, vous serez capable d'ajouter une offre d'emploi.

III. Conclusion

Ce projet qui consistait à gérer une agence de recrutement virtuelle, commençant par le login et finissant par déposer des offres a été une nouvelle expérience, puisque c'était la première fois que nous travaillons avec le Framework Django, nous avons la chance de bien appliquer nos connaissances en python, html et Bootstrap.

En effet, c'était pas du tout facile mais grâce à notre endurance, notre travail dur et notre dévouement pour donner le meilleur, nous avons pu bien présenter ce projet.

Dans les chapitres précédents, nous avons essayé d'entamer et de bien expliquer le fonctionnement de l'application, dans le but d'avoir une vue globale et simplifiée du système.

Finalement, nous avons détaillé les différents outils avec lesquels nous avons travaillé et qui nous ont permis d'organiser le travail, bien évidemment les tâches

étaient partagées d'une façon équivalente, résultant un travail de groupe bien homogène et compatible.