

Rapport du Projet

Réalisé par :

AIT BRIK Sara – ARI Chaymaa

ASSAG Khadija – ISLAH Zineb

Application des recommandations des offres d'emplois

Encadré par :

Prof LAMGHARI Nidal

2023/2024

Résumé

Notre application web innovante révolutionne la recherche d'emploi en utilisant des algorithmes avancés de machine Learning pour fournir des recommandations personnalisées et pertinentes. Les utilisateurs peuvent s'inscrire facilement, avec une option de reconnaissance faciale, et se connecter à leur compte. Une fois connectés, ils peuvent gérer leurs comptes, ils reçoivent des recommandations d'emploi adaptées à leur profil, peuvent marquer leurs offres préférées comme favoris, utiliser une fonction de recherche avancée avec des filtres spécifiques pour trouver des offres correspondant à leurs critères, et postuler directement aux offres d'emploi. Du côté administratif, un tableau de bord complet permet de visualiser les données telles que le nombre d'offres et les postulations, ainsi que de gérer les offres et les utilisateurs. Grâce à ces fonctionnalités, notre application nommée JobMatch+ simplifie et optimise la recherche d'emploi, renforçant ainsi l'engagement et la satisfaction des utilisateurs.

Remerciement

Cher professeur,

Nous tenons à vous exprimer notre profonde gratitude pour l'encadrement exceptionnel que vous nous avez offert tout au long de ce projet. Votre dévouement, vos conseils avisés et votre soutien constant ont été inestimables pour la réussite de notre travail.

Dès le début, vous avez su instaurer un climat de confiance et d'encouragement qui nous a permis de nous investir pleinement. Votre expertise et votre disponibilité ont été des atouts précieux qui nous ont guidés à chaque étape du projet. vous avez toujours été là pour répondre à nos questions, dissiper nos doutes et nous orienter dans la bonne direction.

Votre rigueur intellectuelle et votre exigence nous ont poussés à donner le meilleur de nous-mêmes. Grâce à vos retours constructifs, nous avons pu affiner notre réflexion et améliorer la qualité de notre travail. Chaque suggestion de votre part nous a permis de progresser et de gagner en confiance.

En plus des compétences académiques, vous nous avez transmis des valeurs humaines essentielles : l'esprit de collaboration, le respect des idées de chacun et l'importance de la persévérance. Votre passion pour l'enseignement et votre engagement envers vos étudiants sont une source d'inspiration pour nous tous. Vous avez su rendre ce projet non seulement formateur, mais aussi particulièrement enrichissant sur le plan personnel.

Nous avons énormément appris grâce à vous et nous vous en sommes infiniment reconnaissants. Ce projet a été une expérience marquante de notre parcours académique, et cela n'aurait pas été possible sans votre encadrement et votre soutien indéfectible.

Encore une fois, merci pour votre investissement et votre générosité.

Table des matières

Introduction	8
Chapitre 1 : Présentation du Projet.....	9
Objectifs	9
Description	9
Méthodologies	9
Technologies	10
Chapitre 2 : Modélisation avec UML.....	13
Introduction	13
1-Diagramme de Cas d'utilisation.....	13
2-Diagramme de classe	15
3-Diagramme de Séquences.....	17
4-Diagramme d'activité :.....	21
Chapitre 3: Prétraitement	23
Description de Dataset.....	23
Pré-traitement	23
Chapitre 4 : Réalisation du projet	33
1-Home:.....	33
2-Inscription	35
3-Login	35
4-Espace Utilisateur	36
4-1-Recommandations	36
4-2-Search	37
4-3-See more	39
4-4-Postuler	39
4-5-Ajouter Favoris	40
4-6-Profil Utilisateur	40
4-7-Liste postulation.....	42
4-8-Liste Favoris	42
5-Espace Admin	43
5-1-Dashboards	43
.....	43

5-2-Gestion Offre	44
5-3-Ajout	44
5-4-Suppression.....	45
5-5-Update.....	45
.....	45
Conclusion	46

Liste de figures

Figure 1: diagramme cas d'utilisation	13
Figure 2: Diagramme de classe.....	15
Figure 3: diagramme de séquence pour Inscription d'utilisateur	17
Figure 4: diagramme de séquence pour authentification par reconnaissance faciale	18
Figure 5: diagramme de séquence pour Authentification d'utilisateur	18
Figure 6: diagramme de séquence pour Postulation d'offre	20
Figure 7: Diagramme d'activité	21
Figure 8: importation des bibliothèques	23
Figure 9: Importation de Dataset , affichage des colonnes et calcul de taille	25
Figure 10: détermination de nombre de valeurs nulles dans chaque colonne	25
Figure 11: création de diagramme	26
Figure 12: visualisation graphique du dataset	26
Figure 13: code de visualisation du nombre des offres d'emplois en fonction de l'année, le mois et IT	26
Figure 14: visualisation du nombre des offres d'emplois en fonction de l'année, le mois et IT.....	27
Figure 15: code d'affichage de Dataset sous forme de Dataframe.....	27
Figure 16: affichage de Dataset sous forme de Dataframe.....	27
Figure 17: fonction remove_newline ().....	28
Figure 18: affichage de dataset	28
Figure 19: code d'initialisation du vectorizer TF-IDF et affichage de matrice.....	29
Figure 20: matrice TF-IDF	29
Figure 21: code de sauvegarde du vectorizer et la matrice TF-IDF et les datasets après traitement...	29
Figure 22: code permettant de lister les fichiers dans le repertoire de base	30
Figure 23: de liste des fichiers dans le repertoire de base.....	30
Figure 24: fonction de recommandation	31
Figure 25: Test de fonction recommend_jobs()	31
Figure 26: vérification pour le user_job_title = "Data Analyst" que les 5 recommandations proposées sont dans son domaine	31
Figure 27: Enregistrement de dataset sous forme JSON.....	32
Figure 28: page Home	33
Figure 29: page Home- pied	34
Figure 30: page Home - Team	34
Figure 31: page Home - services	34
Figure 32: page inscription	35
Figure 33: page login	35
Figure 34:page de recommandation en mode sombre.....	36
Figure 35: page de recommandation en mode éclairé	36
Figure 36: recherche des emplois en fonction de la localisation	38
Figure 37: résultat de recherche des emplois en fonction de la localisation.....	38
Figure 38: affichage des informations détaillées sur l'offre recommandée	39
Figure 39: formulaire de postulation d'offre.....	40
Figure 40: profil utilisateur	41

Figure 41: affichage d' alerte de succès	41
Figure 42: liste de postulation	42
Figure 43: liste favoris	42
Figure 44: Dashboards.....	43
Figure 45: liste des offres	44
Figure 46: ajout d'une offre.....	44
Figure 47 : formulaire de modification d'offre.....	45

Introduction

Dans un marché de l'emploi de plus en plus compétitif et en constante évolution, la quête d'opportunités professionnelles peut rapidement devenir une tâche complexe et chronophage. L'émergence d'applications capables de personnaliser les recommandations d'offres d'emploi révolutionne cette recherche. En offrant des suggestions d'emploi finement adaptées aux compétences, à l'expérience et aux aspirations des utilisateurs, ces outils numériques rehaussent non seulement la satisfaction des chercheurs d'emploi et des recruteurs, mais renforcent également leur engagement et leur fidélité envers la plateforme. C'est dans ce cadre que JobMatch+ a été développée. Cette application web innovante, basée sur des algorithmes d'apprentissage automatique, aspire à transformer radicalement la manière dont les utilisateurs découvrent leur emploi idéal, en rendant le processus plus fluide, plus rapide et infiniment plus efficace.

Chapitre 1 : Présentation du Projet

Objectifs

Réalisation d'une application web développée en Python basé sur un système de recommandation des emplois personnalisées et pertinentes à ses utilisateurs suivant leurs intérêts tout en contribuant à l'engagement, à la fidélité et à la satisfaction des utilisateurs.

Description

JobMatch+ est une application innovante conçue pour simplifier la recherche d'emploi en proposant des opportunités professionnelles personnalisées. Grâce à des algorithmes avancés d'apprentissage automatique, JobMatch+ analyse les compétences, l'expérience professionnelle, les intérêts et les objectifs de carrière des utilisateurs pour recommander les emplois les plus pertinents et adaptés à leur profil. JobMatch+ guide vers les opportunités qui correspondent le mieux aux aspirations. Les offres d'emploi recommandées sont facilement accessibles, avec des notifications instantanées pour les nouvelles opportunités et un tableau de bord intuitif pour suivre la progression. Avec JobMatch+, l'utilisateur peut trouver son emploi idéal plus rapidement et efficacement que jamais

Méthodologies

Pour le développement de l'application JobMatch+, nous avons utilisé les méthodologies suivantes :

- Reconnaissance faciale

Objectif : Assurer une sécurité accrue et améliorer l'expérience utilisateur lors de la Connexion.

Technologie : Utilisation de la bibliothèque de reconnaissance faciale face_recognition
Pour authentifier les utilisateurs de manière rapide et sécurisée.

- Filtrage par contenu

Objectif : Fournir des recommandations d'offres d'emploi personnalisées et pertinentes.

Technologie : Algorithmes de machine learning basés sur le filtrage par contenu, qui analysent les compétences, l'expérience professionnelle, les intérêts et les objectifs de carrière des utilisateurs.

Processus :

1. Collecte de Données : Récupération des informations utilisateur et des descriptions d'offres d'emploi.
2. Prétraitement : Nettoyage et transformation des données pour les rendre utilisables par l'algorithme.
3. Comparaison : Identification des correspondances entre les profils utilisateurs et les offres d'emploi.

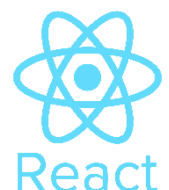
Technologies

Pour créer une application avec des interfaces en React (frontend) et un backend en Django, on suit généralement un modèle de développement où React gère l'affichage et l'interaction avec l'utilisateur, tandis que Django s'occupe de la logique métier et du traitement des données:

React (frontend) :

C'est une bibliothèque JavaScript frontale à code source ouvert permettant de créer des interfaces utilisateur ou des composants d'interface utilisateur.

React peut être utilisé pour effectuer des requêtes HTTP et manipuler les données. Les composants React envoient des requêtes à l'API pour récupérer, envoyer ou



mettre à jour des données. Les réponses JSON de l'API sont utilisées pour mettre à jour l'état des composants React, offrant ainsi une expérience utilisateur dynamique et réactive.

Cette séparation claire entre le frontend et le backend permet une meilleure organisation du code, une maintenance plus facile et la possibilité de faire évoluer chaque partie de l'application indépendamment, c'est pour cela on a choisi de travailler avec Django.

Django (backend):

Django est un framework web open source en Python qui simplifie le développement d'applications web en promouvant la réutilisation du code et en suivant le principe "ne pas se répéter" (DRY). En utilisant :



- Django REST Framework (DRF), pour faciliter la création des API RESTful robustes en définissant des modèles de données, des vues et des sérialiseurs. Les modèles structurent les données, les vues gèrent la logique de traitement des requêtes HTTP et les sérialiseurs convertissent les données en formats compatibles avec JSON.
- Django-cors-headers Pour permettre la communication entre le frontend et le backend, est utilisé pour gérer les Cross-Origin Resource Sharing (CORS), facilitant ainsi les requêtes provenant de domaines différents, comme ceux d'applications React.

Cette approche modulaire et organisée permet de développer des applications web puissantes et maintenables, tout en assurant une intégration fluide entre le frontend et le backend.

Mongo db (Base de donnees) :

MongoDB est une base de données orientée documents qui permet de stocker des données en utilisant un format flexible et JSON-like, appelé BSON. Pour connecter Django à MongoDB, on utilise généralement une bibliothèque tierce comme Djongo , qui permet à Django de communiquer avec MongoDB de manière transparente. En configurant les paramètres de la base de données dans `settings.py`, nous spécifions le moteur de base de données, le nom de la base, et les détails de



connexion. Ensuite, les modèles Django peuvent être définis et manipulés de la même manière que pour une base de données relationnelle traditionnelle, et ces modèles sont ensuite migrés dans la base de données MongoDB. Cela permet de profiter de la flexibilité et de la performance de MongoDB tout en utilisant le puissant ORM de Django. Cette intégration simplifie le développement en combinant les avantages de Django pour la gestion des données avec la robustesse de MongoDB pour le stockage des documents, offrant ainsi une solution efficace pour manipuler et stocker des données de manière flexible et performante.

scikit-learn :

L'utilisation de scikit-learn (souvent abrégé en sklearn) dans le contexte de l'extraction de texte et de la similarité cosinus offre plusieurs avantages, voici deux outils majeurs :



TfidfVectorizer : est un outil puissant pour la conversion de textes en matrices numériques, où chaque colonne représente une caractéristique unique (un mot, par exemple) et chaque ligne représente un document. L'acronyme TF-IDF signifie Term Frequency-Inverse Document Frequency.

cosine_similarity : mesure l'angle entre deux vecteurs. Deux vecteurs identiques auront une similarité cosinus de 1, deux vecteurs orthogonaux (totalement différents) auront une similarité de 0, et deux vecteurs diamétralement opposés auront une similarité de -1 .

modelio:

Modelio est un outil de modélisation UML open source qui permet aux développeurs et architectes de logiciels de concevoir et de documenter des systèmes en utilisant divers diagrammes de modélisation. Cet outil prend en charge une gamme complète de diagrammes UML (Unified Modeling Language), y compris les diagrammes de classes et les diagrammes de cas d'utilisation, et il est également compatible avec les modèles BPMN (Business Process Model and Notation). Voici une description de son utilisation pour réaliser des diagrammes de classe et des diagrammes de cas d'utilisation



Chapitre 2 : Modélisation avec UML

Introduction

La modélisation constitue un pilier essentiel dans le processus de développement logiciel, offrant une représentation organisée et cohérente du fonctionnement et de l'architecture d'une application. Dans le cadre de notre projet visant à créer une application web dédiée à la recommandation d'offres d'emploi, la modélisation joue un rôle central dans la conception de cette plateforme. Pour ce faire, une diversité de diagrammes est utilisée pour saisir les différentes composantes de l'application, couvrant diagramme de cas d'utilisation, de classes, de séquences et d'activités.

1-Diagramme de Cas d'utilisation

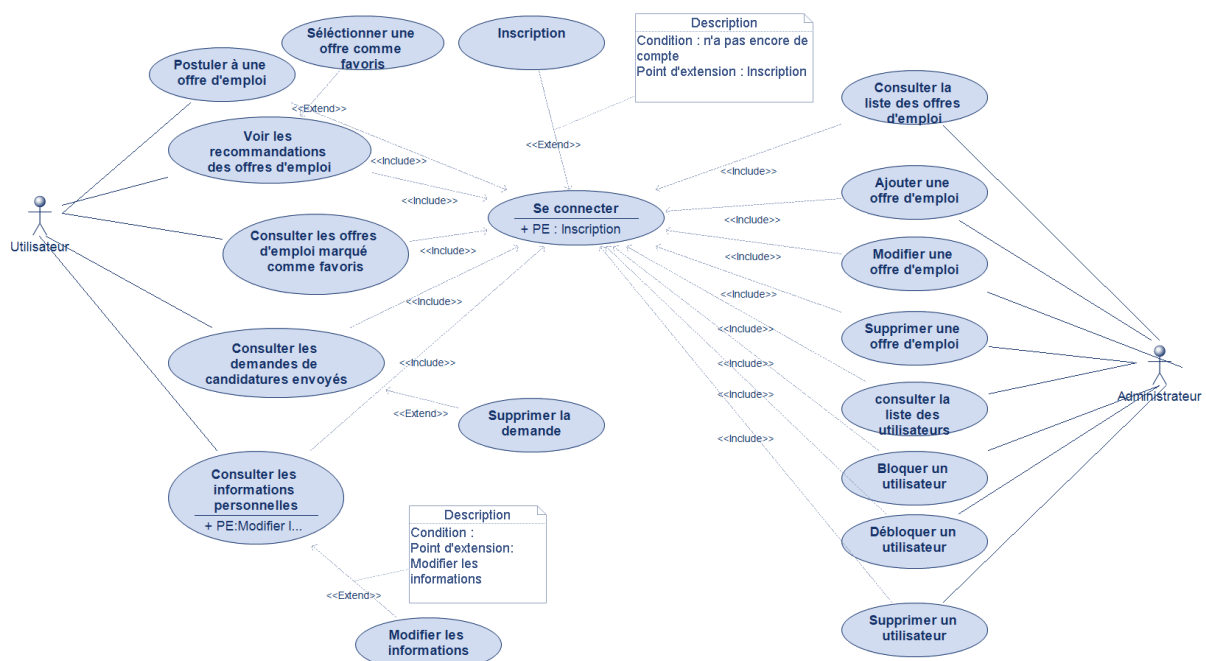


Figure 1: diagramme cas d'utilisation

Le diagramme de cas d'utilisation pour notre application web des recommandations des offres d'emplois offre une vue globale des différentes fonctionnalités et interactions au sein

du système. Deux acteurs majeurs sont en interaction avec notre système. les 'Utilisateurs' et les 'Administrateurs' .

'L'utilisateur' doit s'authentifier ou s'inscrire s'il s'agit de sa première utilisation. Après s'être authentifié, il a la possibilité

de visualiser la liste des recommandations d'offres d'emploi adaptées à ses informations (secteur d'activité, diplôme, etc.)

Et postuler à celles qui l'intéressent. Il peut aussi examiner ultérieurement la liste de ses candidatures avec option de suppression de ces demandes.

Il a également la possibilité de marquer une offre comme préférée et de consulter ultérieurement sa liste d'offres préférées.

Il est possible pour l'utilisateur de visualiser et de mettre à jour ses informations personnelles.

Les 'Administrateurs' jouent un rôle central dans le fonctionnement fluide de notre Application. Avant de pouvoir accéder à ses fonctionnalités, il doit s'authentifier pour garantir la sécurité du système. Une fois authentifié, il a la possibilité de gérer les offres d'emploi (Ajout, Modification, Suppression, consultation), les utilisateurs (Bloquer, Débloquer, Supprimer, consultation).

2-Diagramme de classe

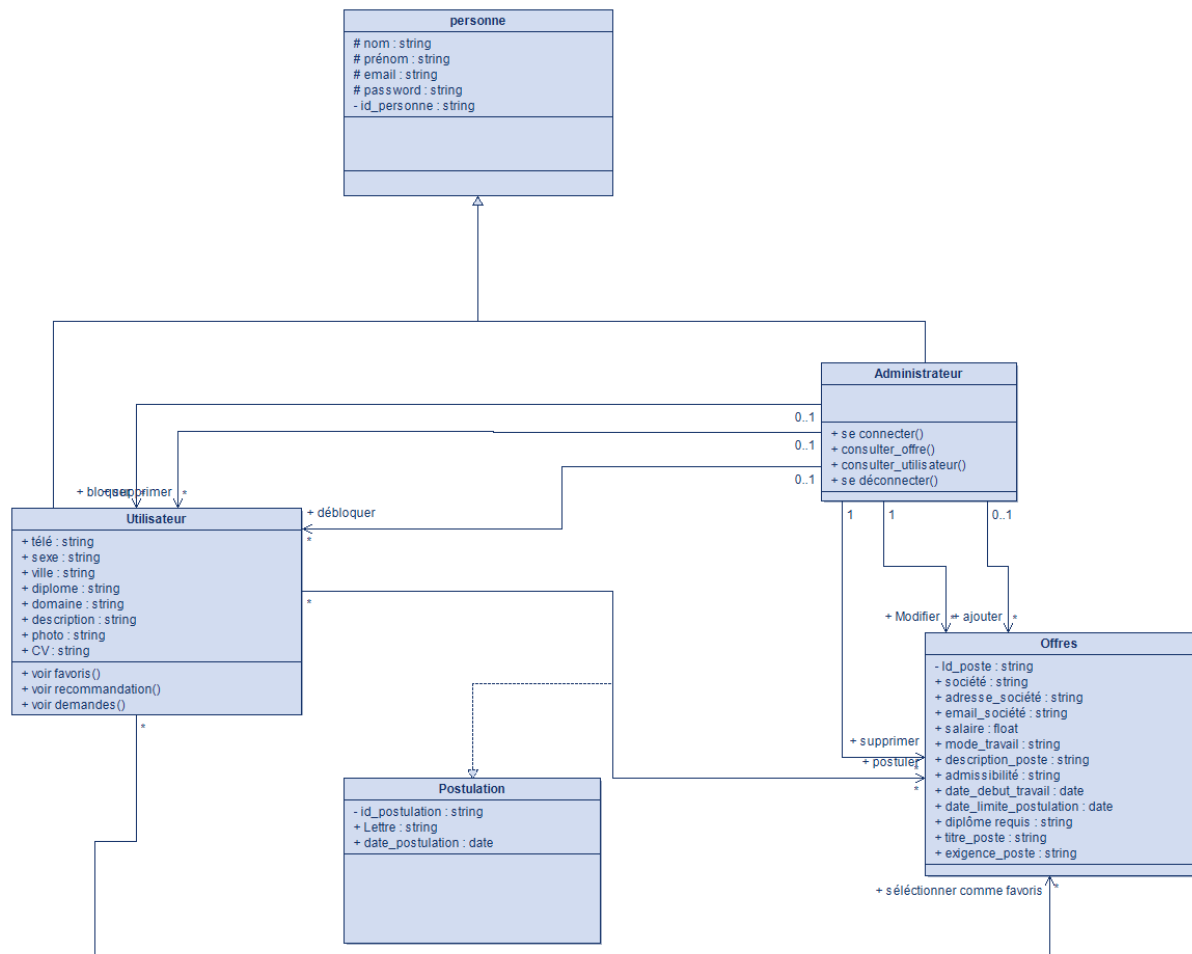


Figure 2: Diagramme de classe

Ce Diagramme de classe représente la structure statique de l'application, en mettant en évidence les classes du système, leurs attributs, leurs méthodes et les relations entre elles.

Classes :

Personne : Cette classe représente une personne générique et contient des attributs communs à tous les utilisateurs du système.

Administrateur : Une sous-classe de **Personne**, qui hérite des attributs de base de la classe **Personne** et possède des fonctionnalités spécifiques à l'administration du système.

Utilisateur : Une sous-classe de Personne, qui hérite également des attributs de base de la classe Personne mais possède des fonctionnalités d'utilisation normale du système.

Offre : Cette classe représente l'offre dans le système. Elle contient les détails de l'offre tels que le titre du poste, la société, etc.

Postulation : Une classe représentant la relation entre un Utilisateur et une Offre, indiquant que l'utilisateur a postulé pour cette offre.

Relations :

Relation d'héritage :

- Administrateur et Utilisateur héritent de la classe Personne. Cela signifie qu'ils héritent de tous les attributs et méthodes définis dans la classe Personne.

Relation d'association :

- Administrateur a une relation avec Offre. Cela signifie qu'un administrateur peut ajouter, modifier et supprimer des offres dans le système.
- L'utilisateur a une relation avec Offre. Cela signifie qu'un utilisateur peut consulter les offres disponibles dans le système et les sélectionner comme favoris.
- Utilisateur peut postuler à plusieurs Offres, une Offre peut être postulée par plusieurs Utilisateurs.

3-Diagramme de Séquences

Inscription d'utilisateur :

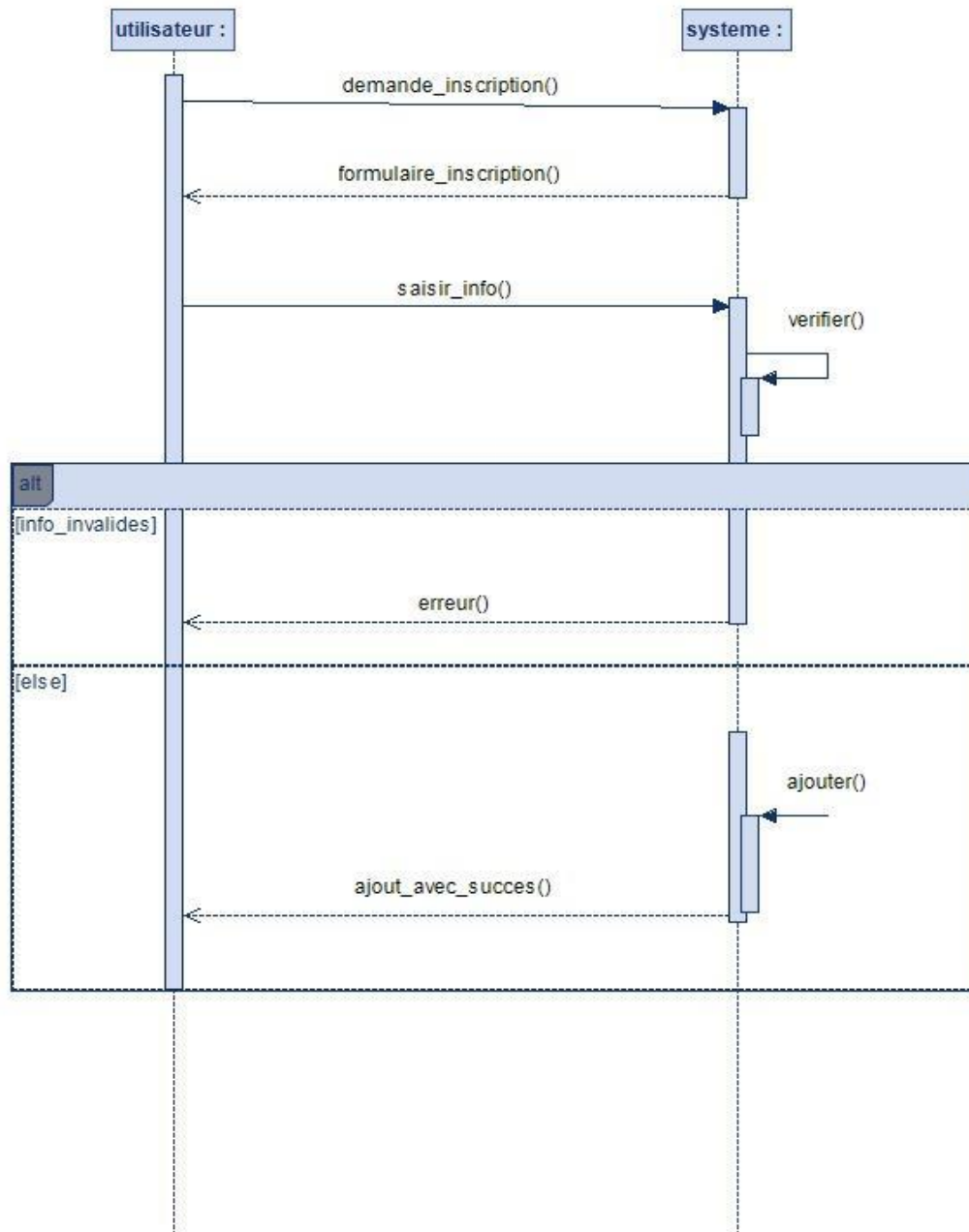


Figure 3: diagramme de séquence pour Inscription d'utilisateur

Le diagramme de séquence pour l'inscription d'utilisateur décrit un processus dynamique entre l'utilisateur et le système. Lorsque l'utilisateur demande l'inscription, le système affiche le formulaire, puis l'utilisateur saisit ses informations. Le système vérifie ensuite les données, bifurquant en deux scénarios : si les informations sont invalides, un message d'erreur est affiché, sinon, l'utilisateur est ajouté à la base de données avec succès.

Authentification d'utilisateur :

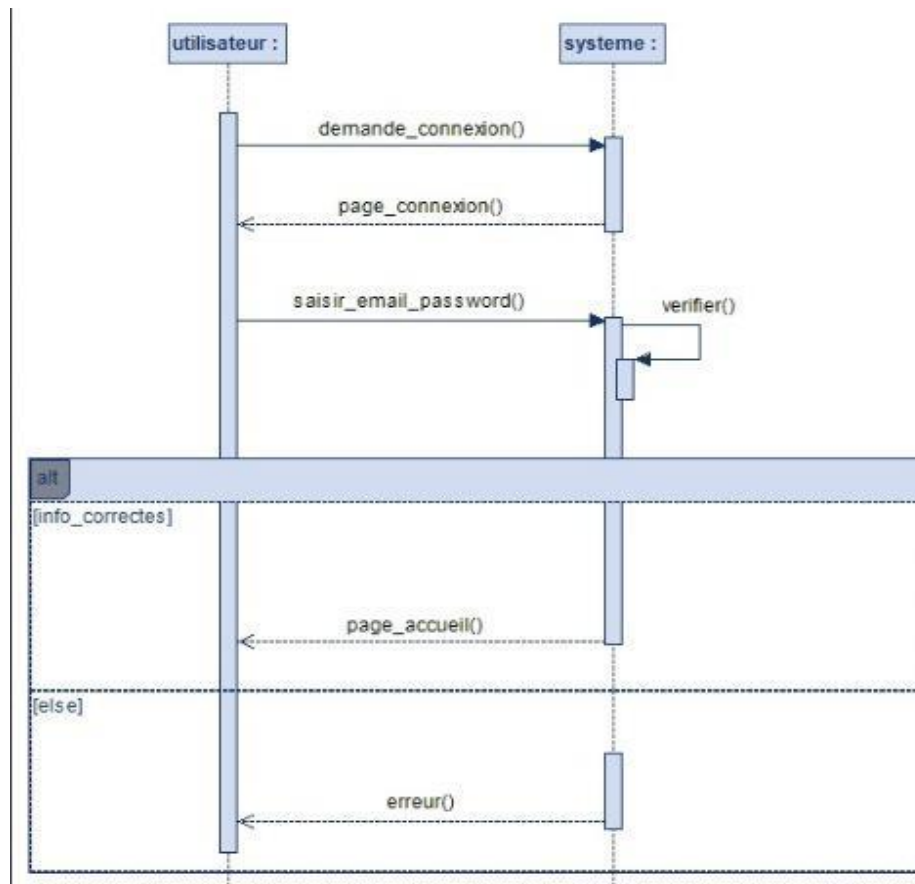


Figure 5: diagramme de séquence pour Authentification d'utilisateur

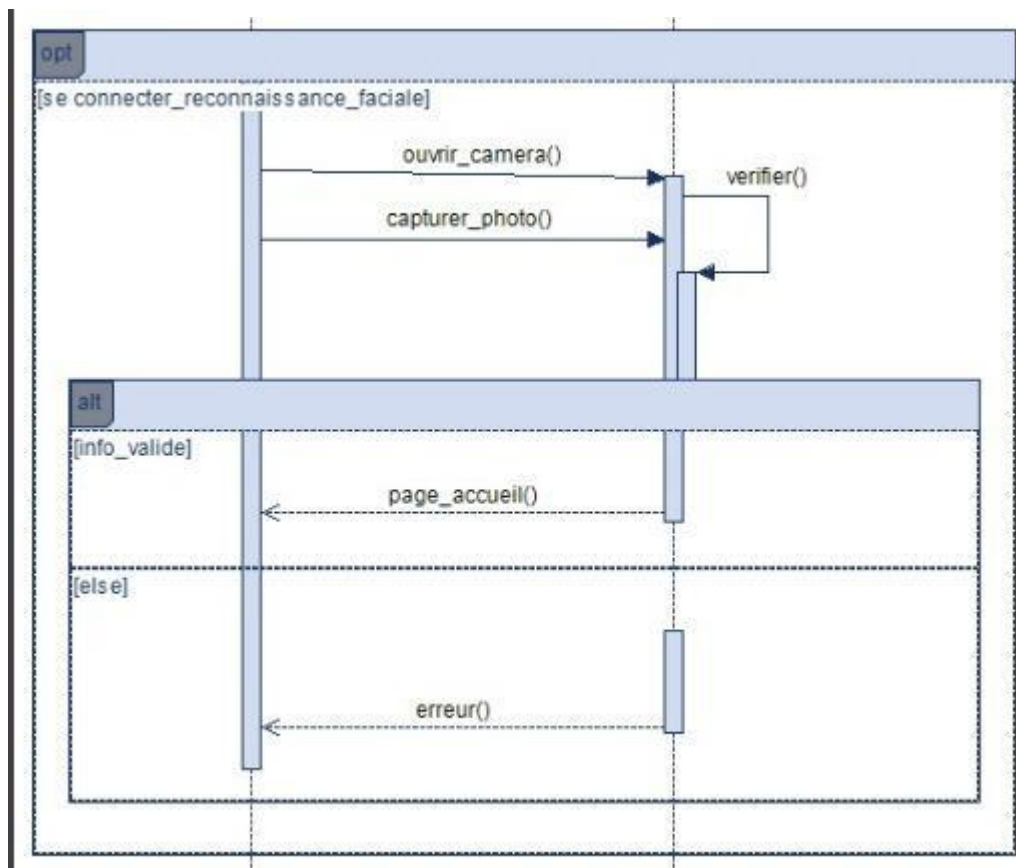


Figure 4: diagramme de séquence pour authentification par reconnaissance faciale

Le diagramme de séquence pour l'authentification d'utilisateur dépeint un processus dynamique entre l'utilisateur et le système. Lorsque l'utilisateur souhaite se connecter, il initie le processus en faisant une demande de connexion, ce à quoi le système répond en affichant la page de connexion.

L'utilisateur saisit ensuite son email et son mot de passe, et le système procède à la vérification de ces informations. À ce stade, deux scénarios se présentent : si les informations sont correctes, le système affiche la page d'accueil, sinon un message d'erreur est affiché, signalant à l'utilisateur que les informations sont incorrectes. En outre, le diagramme inclut un fragment optionnel où l'utilisateur a la possibilité de se connecter via la reconnaissance faciale. Si cette option est choisie, le processus commence par l'ouverture de la caméra par l'utilisateur, la capture d'une photo, puis le système procède à la vérification de cette dernière. Encore une fois, deux scénarios sont envisagés : si les informations sont valides, l'utilisateur est redirigé vers la page d'accueil, sinon un message d'erreur est affiché, informant que la reconnaissance faciale a échoué.

Postulation d'offre :

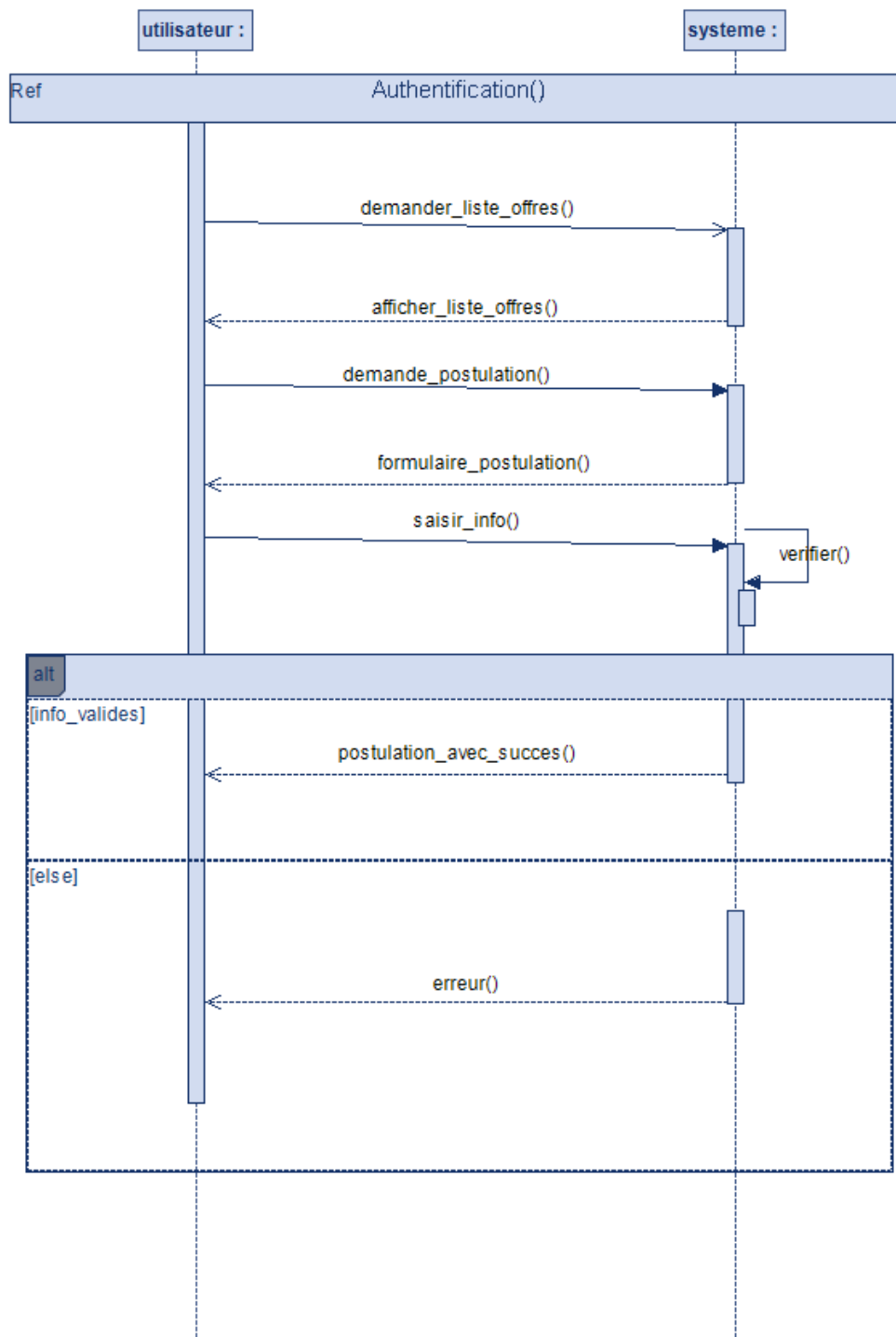


Figure 6: diagramme de séquence pour Postulation d'offre

L'interaction entre l'utilisateur et le système lors de la postulation est un processus dynamique. Tout commence par la connexion de l'utilisateur, une étape essentielle avant de pouvoir accéder aux offres disponibles. Une fois connecté, l'utilisateur demande à visualiser les offres disponibles, et le système répond en affichant la liste correspondante. Après avoir choisi l'offre à laquelle il souhaite postuler, l'utilisateur envoie une demande de postulation, et remplit le formulaire et le système effectue une vérification minutieuse de leur exactitude. Ensuite, deux scénarios se dessinent : si les informations sont valides, la postulation s'effectue, sinon si des informations incorrectes sont détectées, le système réagit en affichant un message d'erreur.

4-Diagramme d'activité :

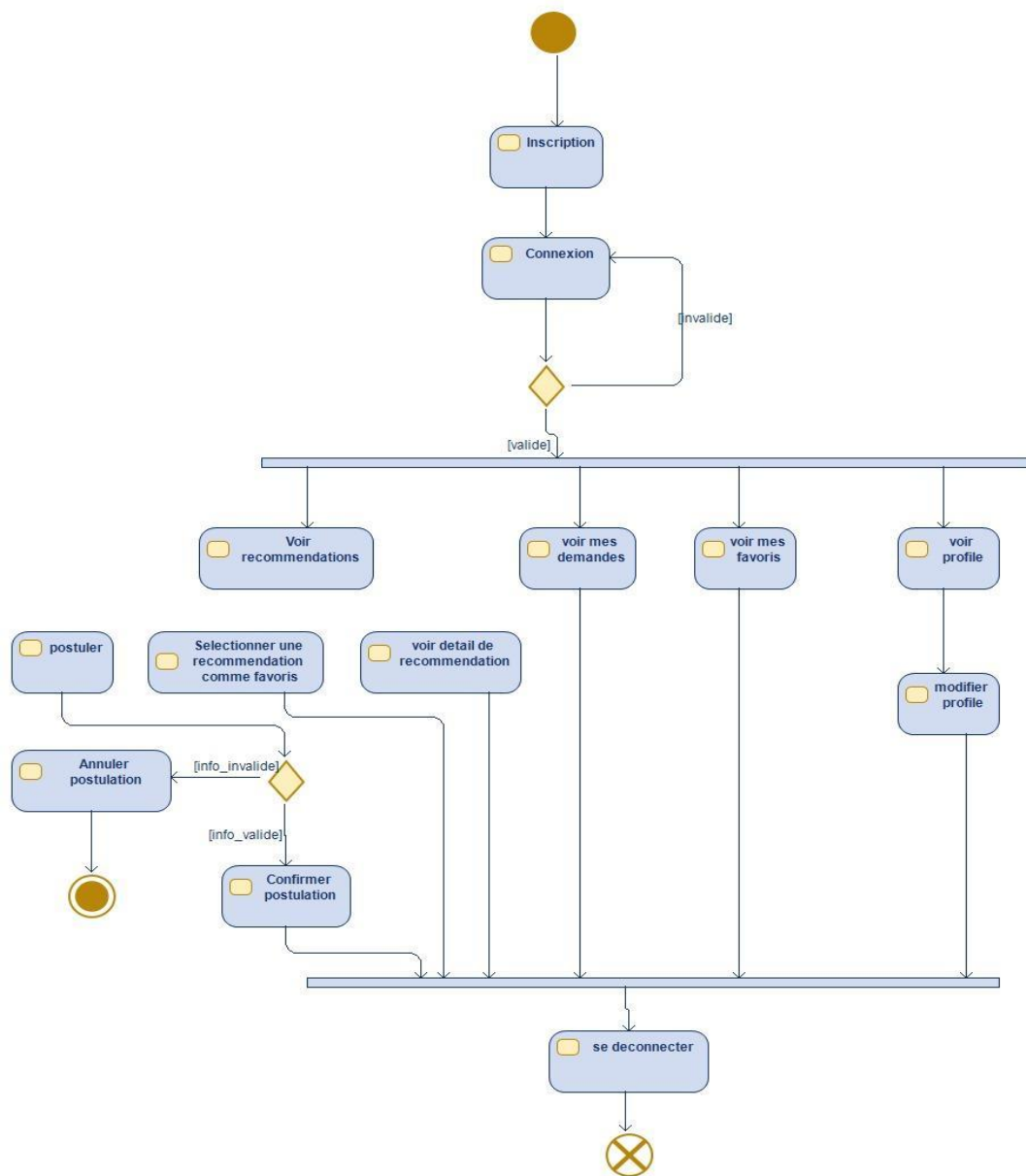


Figure 7: Diagramme d'activité

Le diagramme d'activité offre une vision fluide du parcours au sein de notre application. Tout commence par la création du compte, un processus intégrant l'inscription initiale, suivie de la connexion. C'est à ce point que l'utilisateur saisit les informations nécessaires, lesquelles seront ensuite vérifiées. En cas d'erreur, l'utilisateur demeure sur la page de connexion pour effectuer les corrections requises. En revanche, si les informations sont valides, un éventail d'options s'ouvre à lui. Il peut explorer différents axes, tels que la consultation et la modification de son profil, l'examen des demandes

précédemment effectuées, l'accès aux offres marquées comme favoris, ou encore l'exploration de la liste des offres recommandées. Au sein de cette dernière, l'utilisateur peut choisir de marquer une offre comme favorite, examiner en détail une offre donnée, ou bien postuler à une offre spécifique. Lorsqu'une demande de postulation est initiée, une vérification est effectuée pour garantir l'exactitude des informations fournies. En cas de données invalides, la postulation est annulée, tandis que leur validation entraîne la confirmation de la postulation. Enfin, une fois les interactions terminées, l'utilisateur peut choisir de se déconnecter pour clore sa session.

Chapitre 3: Prétraitement

Description de Dataset

Pour réaliser notre système de recommandation d'offres d'emploi, nous avons utilisé une base de données particulièrement riche intitulée "data_job_post.csv", contenant 19 001 lignes (offres) et 24 colonnes. Parmi les colonnes principales, on trouve "jobpost" qui englobe l'ensemble des autres colonnes, y compris "Title" qui présente le domaine de l'offre, "date", "company" qui contient l'entreprise associée à l'offre, "RequiredQual" pour identifier les qualifications requises pour l'offre, ainsi que "Location" qui indique le lieu de l'entreprise. La base de données inclut également la colonne "jobdescription" pour les descriptions des postes, une colonne indiquant si l'offre est dans le domaine de l'IT ou non, et une colonne "ApplicationP" contenant des indications pour postuler à l'offre.

Grâce à cette base de données extrêmement détaillée, notre système est en mesure d'effectuer un traitement approfondi et sophistiqué pour offrir des recommandations d'offres d'emploi pertinentes et personnalisées. En analysant ces multiples aspects des offres d'emploi, nous pouvons mieux comprendre les exigences des postes, les préférences des utilisateurs et les tendances du marché, permettant ainsi une expérience utilisateur optimisée et efficace.

Pré-traitement

- **Les Bibliothèques**

```
import missingno as mno
from mpl_toolkits.mplot3d import Axes3D
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt # plotting
import numpy as np # linear algebra
import os # accessing directory structure
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
import pickle
from sklearn.metrics.pairwise import cosine_similarity
```

Figure 8: importation des bibliothèques

Pandas : Une bibliothèque pour la manipulation et l'analyse de données. Elle offre des structures de données et des fonctions pour manipuler des tableaux de données hétérogènes (DataFrames) de manière efficace.

numpy: Une bibliothèque fondamentale pour le calcul scientifique en Python. Elle prend en charge des tableaux et des matrices multidimensionnelles, ainsi que des fonctions mathématiques de haut niveau.

matplotlib.pyplot: Une bibliothèque Python pour la création de graphiques 2D. Elle est couramment utilisée pour la visualisation de données.

os: Une bibliothèque standard de Python qui fournit une façon d'interagir avec le système d'exploitation.

missingno : C'est une bibliothèque Python utilisée pour la visualisation des valeurs manquantes dans les ensembles de données. Elle permet de visualiser facilement les données manquantes dans un DataFrame, ce qui aide à diagnostiquer des problèmes de données.

mpl_toolkits.mplot3d (Axes3D): Un sous-module de Matplotlib qui fournit des outils pour la visualisation de données en 3D.

pickle : Une bibliothèque standard de Python pour la sérialisation et la désérialisation des objets Python.

sklearn.preprocessing (StandardScaler) : Une partie de la bibliothèque scikit-learn utilisée pour normaliser ou standardiser les caractéristiques des données.

sklearn.feature_extraction.text : Une partie de scikit-learn utilisée pour convertir une collection de documents texte en une matrice de caractéristiques TF-IDF (Term Frequency-Inverse Document Frequency).

sklearn.metrics.pairwise : Une fonction de scikit-learn pour calculer la similarité cosinus entre des vecteurs. La similarité cosinus est une mesure de similarité entre deux vecteurs de produit scalaire, souvent utilisée pour la mesure de similarité entre documents textuels.

- **Importer la Dataset , afficher les colonnes et calculer la taille**


```
df=pd.read_csv('data job posts.csv')
```

```
df.columns
```

```
Index(['jobpost', 'date', 'Title', 'Company', 'AnnouncementCode', 'Term',  
      'Eligibility', 'Audience', 'StartDate', 'Duration', 'Location',  
      'JobDescription', 'JobRequirment', 'RequiredQual', 'Salary',  
      'ApplicationP', 'OpeningDate', 'Deadline', 'Notes', 'AboutC', 'Attach',  
      'Year', 'Month', 'IT'],  
      dtype='object')
```

```
df.shape
```

```
(19001, 24)
```

Figure 9: Importation de Dataset , affichage des colonnes et calcul de taille

- Visualiser le nombre de valeurs nulles dans chaque colonne:

```
#check the missing values in a given DataFrame
```

```
def showme_all_mv(df):  
    return df.isna().sum()
```

```
showme_all_mv(df)
```

jobpost	0
date	0
Title	28
Company	7
AnnouncementCode	17793
Term	11325
Eligibility	14071
Audience	18361
StartDate	9326
Duration	8203
Location	32
JobDescription	3892
JobRequirment	2522
RequiredQual	484
Salary	9379
ApplicationP	60
OpeningDate	706
Deadline	65
Notes	16790
AboutC	6531
Attach	17442
Year	0
Month	0
IT	0
dtype:	int64

Figure 10: détermination de nombre de valeurs nulles dans chaque colonne

```
mno.bar(df)
```

Figure 11: création de diagramme

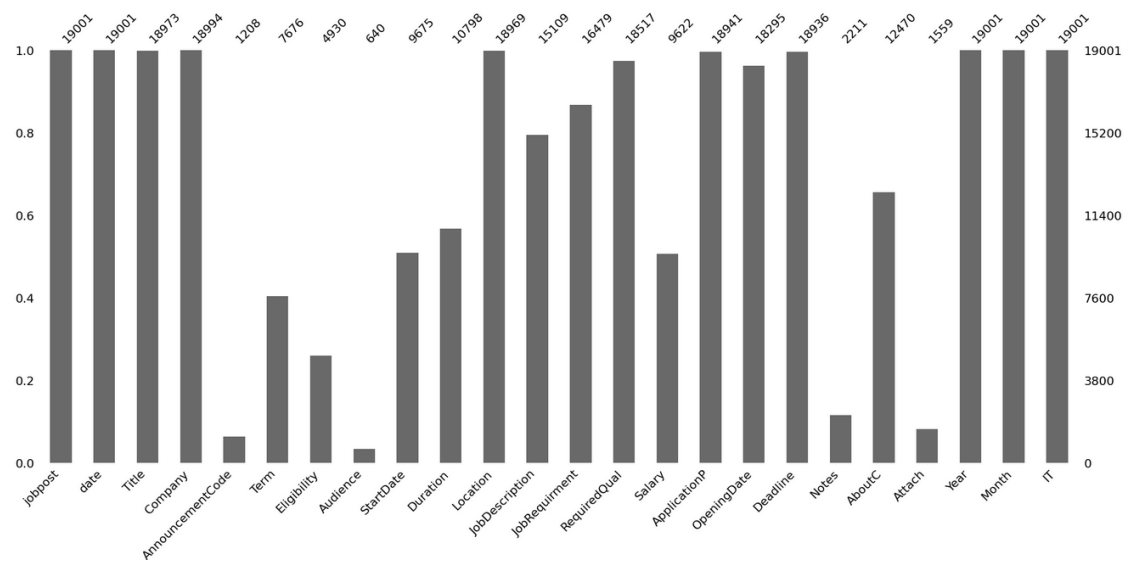


Figure 12: visualisation graphique du dataset

- Visualiser le nombre des offres d'emplois en fonction de l'année, le mois et aussi IT

```
def plotPerColumnDistribution(df, nGraphShown, nGraphPerRow):
    nunique = df.nunique()
    df = df[[col for col in df if nunique[col] > 1 and nunique[col] < 50]] # For displaying purposes, pick columns that have between 1 and 50
    nRow, nCol = df.shape
    columnNames = list(df)
    nGraphRow = int(np.ceil(nCol / nGraphPerRow)) # Calculate the number of rows as a positive integer
    plt.figure(num=None, figsize=(6 * nGraphPerRow, 8 * nGraphRow), dpi=80, facecolor='w', edgecolor='k')
    for i in range(min(nCol, nGraphShown)):
        plt.subplot(nGraphRow, nGraphPerRow, i + 1)
        columnDf = df.iloc[:, i]
        if (not np.issubdtype(type(columnDf.iloc[0]), np.number)):
            valueCounts = columnDf.value_counts()
            valueCounts.plot.bar()
        else:
            columnDf.hist()
            plt.ylabel('counts')
            plt.xticks(rotation=90)
            plt.title(f'{columnNames[i]} (column {i})')
    plt.tight_layout(pad=1.0, w_pad=1.0, h_pad=1.0)
    plt.show()

# Example usage:
plotPerColumnDistribution(df, 10, 5)
```

Figure 13: code de visualisation du nombre des offres d'emplois en fonction de l'année, le mois et IT

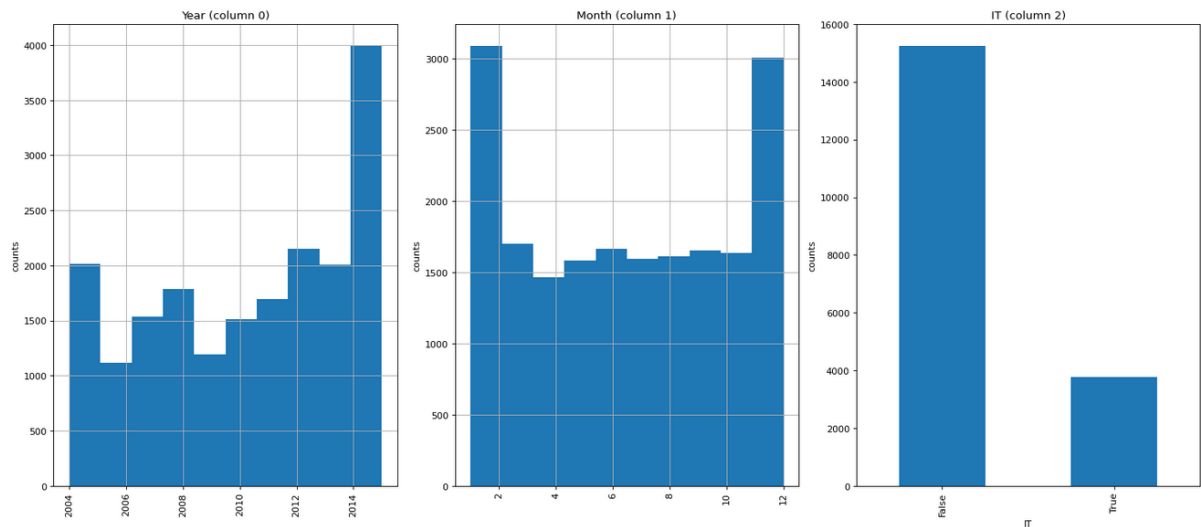


Figure 14: visualisation du nombre des offres d'emplois en fonction de l'année, le mois et IT

● Afficher la Dataset sous forme de DataFrame :

```
# Assuming df is your DataFrame
for i in range(0, df.shape[0]):
    for j in range(0, df.shape[1]):
        if pd.isna(df.iloc[i, j]):
            df.iloc[i, j] = ''
# Display the DataFrame after replacement
df
```

Figure 15: code d'affichage de Dataset sous forme de DataFrame

	jobpost	date	Title	Company	AnnouncementCode	Term	Eligibility	Audience	StartDate	Duration	...	Salary
0	AMERIA Investment Consulting Company\r\nJOB TL...	Jan 5, 2004	Chief Financial Officer	AMERIA Investment Consulting Company							...	
1	International Research & Exchanges Board (IREX...	Jan 7, 2004	Full-time Community Connections Intern (paid i...	International Research & Exchanges Board (IREX)						3 months	...	
2	Caucasus Environmental NGO Network (CENN)\r\n...	Jan 7, 2004	Country Coordinator	Caucasus Environmental NGO Network (CENN)						Renewable annual contract\r\nPOSITION	...	
3	Manoff Group\r\nJOB TITLE: BCC Specialist\r\n...	Jan 7, 2004	BCC Specialist	Manoff Group							...	
4	Yerevan Brandy Company\r\nJOB TITLE: Software...	Jan 10, 2004	Software Developer	Yerevan Brandy Company							...	
...

Figure 16: affichage de Dataset sous forme de DataFrame

- Appliquer la fonction `remove_newline ()` pour chaque valeur de chaque colonne afin de supprimer les caractères `\r\n` en affichant la Dataset après la suppression :

```
# Fonction pour supprimer le caractère \r\n d'une chaîne
def remove_newline(text):
    if isinstance(text, str): # Vérifier si la valeur est une chaîne de caractères
        return text.replace('\r\n', '')
    else:
        return text # Retourner la valeur inchangée si ce n'est pas une chaîne de caractères

# Appliquer la fonction remove_newline à chaque valeur de chaque colonne
df = df.applymap(remove_newline)

# Afficher le DataFrame mis à jour
df
```

Figure 17: fonction `remove_newline ()`

	jobpost	date	Title	Company	AnnouncementCode	Term	Eligibility	Audience	StartDate	Duration	...	Salary
0	AMERIA Investment Consulting CompanyJOB TITLE:...	Jan 5, 2004	Chief Financial Officer	AMERIA Investment Consulting Company							...	
1	International Research & Exchanges Board (IREX...	Jan 7, 2004	Full-time Community Connections Intern (paid i...	International Research & Exchanges Board (IREX)						3 months	...	
2	Caucasus Environmental NGO Network (CENN)JOB T...	Jan 7, 2004	Country Coordinator	Caucasus Environmental NGO Network (CENN)						Renewable annual contractPOSITION	...	
3	Manoff GroupJOB TITLE: BCC SpecialistPOSITION...	Jan 7, 2004	BCC Specialist	Manoff Group							...	
4	Yerevan Brandy CompanyJOB TITLE: Software Dev...	Jan 10, 2004	Software Developer	Yerevan Brandy Company							...	

Figure 18: affichage de dataset

● Vectorisation

Initialiser le vectorizer TF-IDF (Term Frequency-Inverse Document Frequency)

La bibliothèque “`sklearn.feature_extraction.text.TfidfVectorizer`” transforme les titres de l’offre en une matrice TF-IDF, puis affiche cette matrice sous forme de tableau.

```
# Initialiser le vectorizer TF-IDF
vectorizer = TfidfVectorizer()

# Transformer les titres de poste en matrice TF-IDF
tfidf_matrix = vectorizer.fit_transform(df['Title'])

# Afficher la matrice TF-IDF
print(tfidf_matrix.toarray())
```

Figure 19: code d'initialisation du vectorizer TF-IDF et affichage de matrice

```
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
```

Figure 20: matrice TF-IDF

- **Sauvegarder du vectorizer et la matrice TF-IDF et les datasets après traitement:**

Dans cette partie du code on va enregistrer le vectorizer et la matrice TF-IDF et notre dataset après traitement pour les utiliser dans notre modèle de recommandations

```
# Sauvegarder le vectorizer
BASE_DIR = os.getcwd()
with open('tfidf_vectorizer.pkl', 'wb') as f:
    pickle.dump(vectorizer, f)

# Sauvegarder la matrice TF-IDF
with open('tfidf_matrix.pkl', 'wb') as f:
    pickle.dump(tfidf_matrix, f)

# Sauvegarder les données des offres d'emploi
df.to_csv('job_data.csv', index=False)
```

Figure 21: code de sauvegarde du vectorizer et la matrice TF-IDF et les datasets après traitement

- **L'enregistrement est effectué avec succès:**

```
# Lister les fichiers dans le répertoire de base
for file in os.listdir(BASE_DIR):
    if file.endswith('.pkl') or file.endswith('.csv'):
        print(file)
```

Figure 22: code permettant de lister les fichiers dans le repertoire de base

```
data job posts.csv
fake_or_real_news.csv
job_data.csv
tfidf_matrix.pkl
tfidf_vectorizer.pkl
```

Figure 23: de liste des fichiers dans le repertoire de base

- **Recommandation basé sur le contenu:**

Dans ce code on charge un `tfidf_vectorizer` et une `tfidf_matrix` à partir de fichiers `.pkl` (fichiers de sauvegarde pickle). Le `tfidf_vectorizer` est utilisé pour transformer le texte en vecteurs TF-IDF. La `tfidf_matrix` contient les vecteurs TF-IDF de tous les titres de poste dans la base de données des offres d'emploi.

on va appliquer ce vectorizer sur `user_job_title` entré puis on calcule la similarité cosinus entre le vecteur TF-IDF du titre de poste de l'utilisateur et tous les vecteurs dans la `tfidf_matrix`.

Le résultat est une matrice de similarité où chaque valeur représente la similarité entre le domaine de l'utilisateur et un titre de poste dans la base de données.

```
def recommend_jobs(user_job_title, top_n=5):
    # Charger le vectorizer et la matrice TF-IDF
    with open('tfidf_vectorizer.pkl', 'rb') as f:
        vectorizer = pickle.load(f)
    with open('tfidf_matrix.pkl', 'rb') as f:
        tfidf_matrix = pickle.load(f)

    # Transformer Le titre de poste de l'utilisateur en vecteur TF-IDF
    user_tfidf = vectorizer.transform([user_job_title])

    # Calculer la similarité cosinus entre le titre de poste de l'utilisateur et les titres dans la base de données
    similarities = cosine_similarity(user_tfidf, tfidf_matrix)

    # Obtenir les indices des offres les plus similaires
    similar_indices = similarities.argsort()[0][-top_n:][::-1]

    # Charger les données des offres d'emploi
    job_data = pd.read_csv('job_data.csv')

    # Retourner les offres les plus similaires
    return job_data.iloc[similar_indices]
```

Figure 24: fonction de recommandation

- **Tester la fonction recommend_jobs()**

On a bien vérifié pour le user_job_title = “Data Analyst” que les 5 recommandations proposées sont dans son domaine.

```
# Tester la fonction de recommandation
user_job_title = 'Data Analyst'
recommended_jobs = recommend_jobs(user_job_title)
recommended_jobs
```

Figure 25: Test de fonction recommend_jobs()

	jobpost	date	Title	Company	AnnouncementCode	Term	Eligibility	Audience	StartDate	Duration	...	Salary	Applica
2192	SCDM LLC TITLE: Data Analyst TERM: Full time	Feb 24, 2006	Data Analyst	SCDM LLC		NaN	Full time	NaN	NaN	ASAP	Permanent	Competitive + paid overtime + bonuses.	Please email your Cover Letter to:
5140	Synergy International Systems, Inc./ Armenia TITLE: Data Analyst	Apr 8, 2008	Data Analyst	Synergy International Systems, Inc./ Armenia		NaN	Full time	NaN	NaN	Immediate	Long Term	NaN	Candidates who these qualifications:
11620	National Instruments TITLE: Data Analyst TERM: ...	May 15, 2012	Data Analyst	National Instruments		NaN	Full time	NaN	NaN	Immediate	NaN	NaN	Please send re: to:employment.armer
1284	Synergy International Systems, Inc./ Armenia TITLE: Data Analyst	May 2, 2005	Data Analyst	Synergy International Systems, Inc./ Armenia		NaN	Full time	NaN	NaN	Immediate	NaN	NaN	If interested, please your resume wit
4842	"K-Telecom" CJSC/VivaCell/ TITLE: Data Analyst	Feb 7, 2008	Data Analyst	"K-Telecom" CJSC / VivaCell/		NaN	Full time	All interested candidates	NaN	01 March 2008	Permanent with three months probation period	VivaCell offers a competitive compensationincl...	Please send yc to:dataanalyst@...Plk

Figure 26: vérification pour le user_job_title = “Data Analyst” que les 5 recommandations proposées sont dans son domaine

- **Enregistrer dataset sous forme JSON :**

```
output_path = 'datasetjob1.json'  
df.to_json(output_path, orient='records')  
print("Fichier JSON créé avec succès.")
```

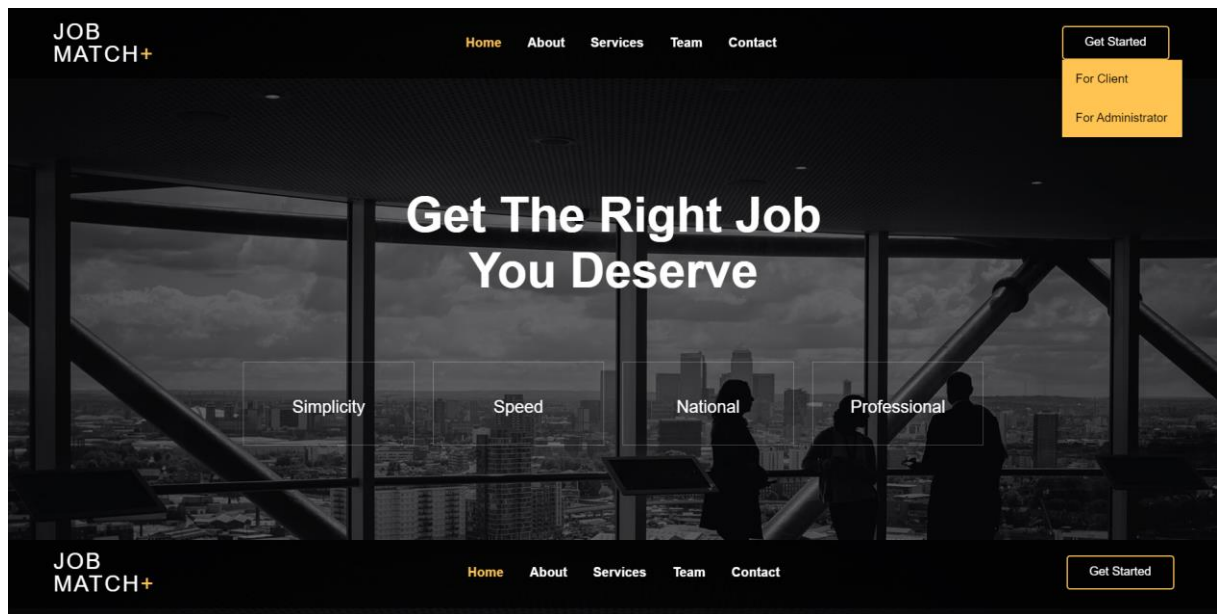
Fichier JSON créé avec succès.

Figure 27: Enregistrement de dataset sous forme JSON

Chapitre 4 : Réalisation du projet

1-Home:

la page d'accueil (Home) de notre site web de recommandation d'offres d'emploi "JobMatch+" offre une vue d'ensemble attrayante et informative des services proposés tel que l'inscription ,les recommandations personnalisées ...



Who We Are ?

JobMatch is an innovative online platform dedicated to simplifying the job search process by providing personalized job recommendations. Designed for job seekers of all backgrounds, from beginners to experienced professionals, JobMatch analyzes each user's skills, experience and preferences, providing them with a targeted and relevant list of job opportunities.



Figure 28: page Home

SERVICES

CHECK OUR SERVICES

1

Registration

create a detailed profile by providing information such as their professional experience, skills, employment. A complete profile helps the site recommend relevant job openings.

2

Personalized Recommendations

A job recommendation system to recommend positions that best match skills, experience and professional preferences of each user

3


Advanced Search

filter job offers based on specific criteria such as industry, location, etc. This helps users quickly find the offers that best suit their needs.


Figure 31: page Home - services

TEAM


CHECK OUR TEAM




Chaymaa Ari
IT & Data Student



Assag Khadija
IT & Data Student



Zineb Islah
IT & Data Student



Sara Ait Brik
IT & Data Student

Figure 30: page Home - Team

Figure 29: page Home- pied

2-Inscription

La page d'inscription d'un site web de recommandation des offres d'emploi offre une interface conviviale et intuitive pour les utilisateurs désireux de créer un compte. Il contient les champs requis : Nom complet, Adresse e-mail, Mot de passe , Confirmation du mot de passe , et autres champs comme domaine , le diplôme ,cv... Bouton "Register" pour valider le formulaire .

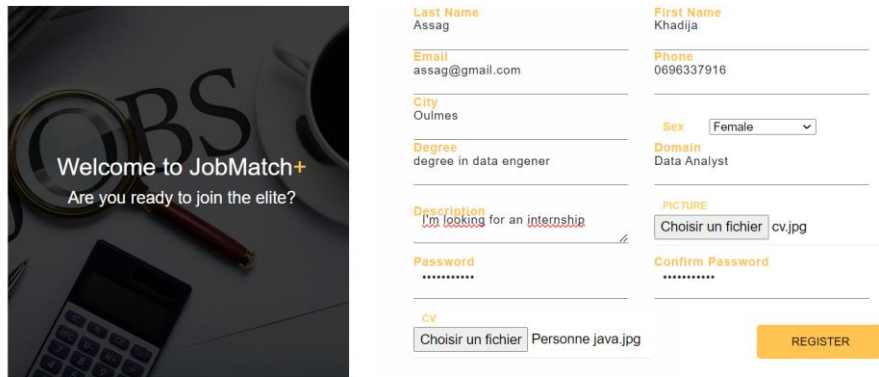


Figure 32: page inscription

3-Login

Cette interface graphique offre aux utilisateurs la possibilité de se connecter rapidement et en toute sécurité. Un formulaire de connexion qui contient les champs requis : Adresse e-mail et mot de passe avec 2 boutons l'une pour la connexion et l'autre pour l'inscription si l'utilisateur n'a pas encore de compte. L'administrateur peut aussi se connecter en entrant son email et son mot de passe pour visualiser des informations sur la base de données et aussi bien faire la gestion des offres d'emploi .

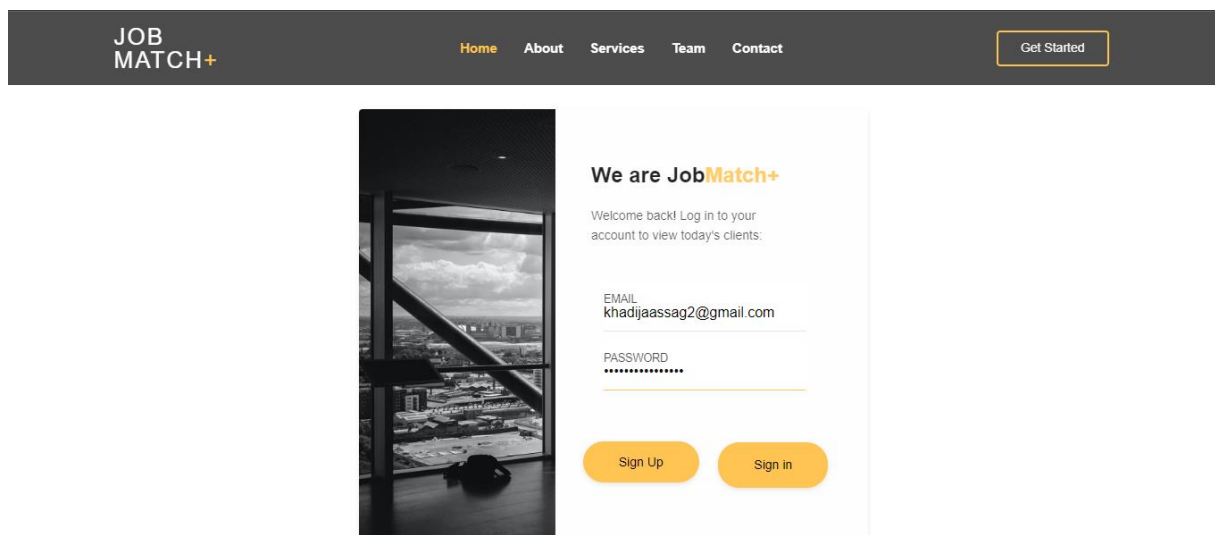


Figure 33: page login

4-Espace Utilisateur

4-1-Recommandations

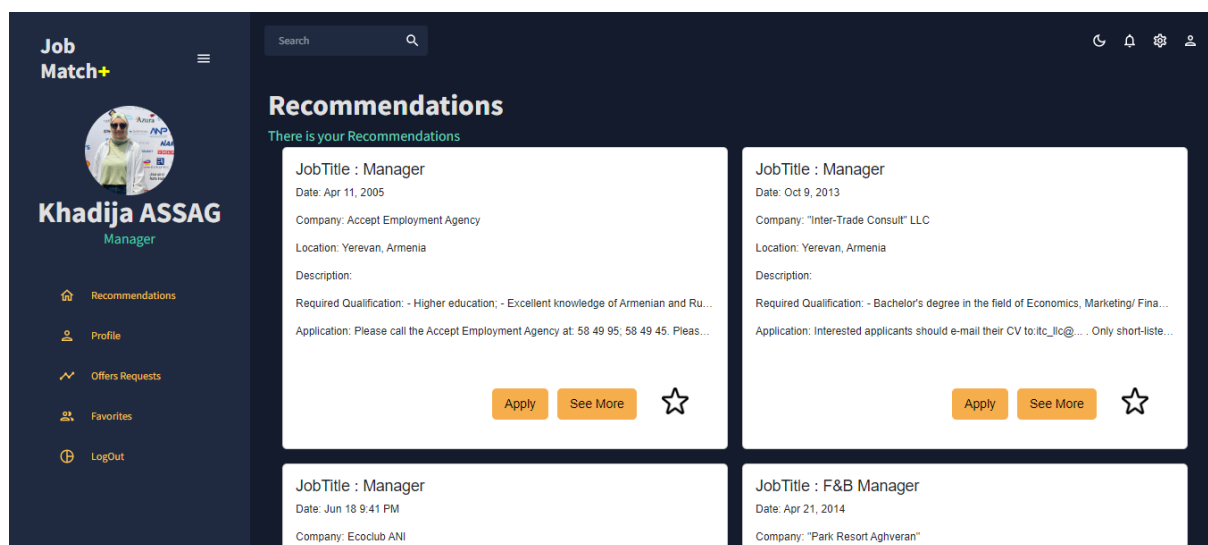


Figure 34: page de recommandation en mode sombre

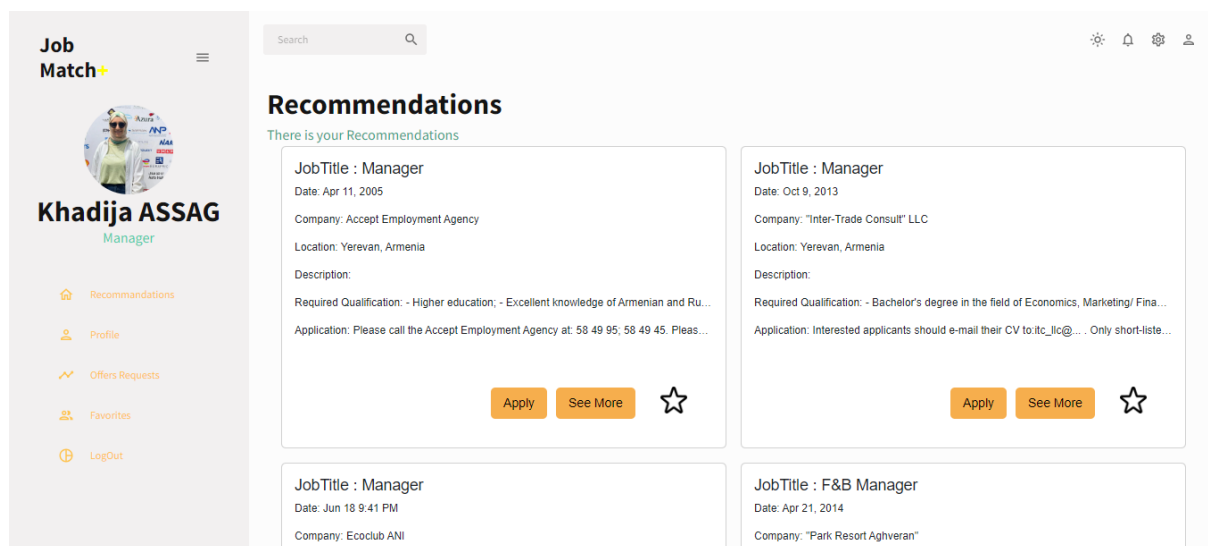


Figure 35: page de recommandation en mode éclairé

L'interface "Recommendations" appartient à l'espace utilisateur qui contient les éléments suivants :

- ❖ **Menu bar** qui contient les liens des autres interfaces Profile, Offers Requests, Favorites et Logout pour quitter l'espace utilisateur et se rediriger vers la page Home.
- ❖ **Top Bar** : contient un espace de recherche , **par ville** du poste de travail intéressé par l'utilisateur.
- ❖ **Cards** désignent les offres d'emploi recommandées pour l'utilisateur concerné, avec la possibilité de voir plus pour afficher toutes les informations à propos de l'offre via le bouton "see more" ainsi la possibilité de postuler à l'offre via le bouton "Apply" .

La recommandation qu'on a utilisé est basée sur le contenu ; c'est-à-dire examiner les caractéristiques intrinsèques du contenu lui-même.

Pour appliquer cette approche à notre application web on doit utiliser les métadonnées des offres d'emploi ainsi que les profils des candidats pour générer des recommandations pertinentes.

L'analyse des métadonnées des offres d'emploi s'agit de recueillir des informations détaillées sur chaque offre y compris le titre, description du poste, le lieu et le salaire, ainsi que pour l'utilisateur en analysant son domaine et son centre d'intérêt.

Cette analyse est basée sur l'extraction des caractéristiques du contenu en utilisant des techniques d'analyse de texte pour extraire des caractéristiques importantes (Tokenization, Stop words removal et Stemming & Lemmatisation).

Pour comparer les caractéristiques des offres d'emploi avec les profils des candidats, nous utilisons des algorithmes de similarité. Parmi les mesures que nous avons adoptées, **la similarité cosinus** joue un rôle clé. Cette mesure calcule l'angle cosinus entre deux vecteurs dans un espace multidimensionnel, ce qui permet de déterminer à quel point les offres d'emploi et les profils des candidats sont similaires. Concrètement, la similarité cosinus permet de quantifier la proximité entre les descriptions de postes et les compétences ou qualifications des candidats, en attribuant une valeur comprise entre -1 et 1. Une valeur proche de 1 indique une forte similitude, tandis qu'une valeur proche de -1 indique une dissimilarité. En utilisant cette technique, nous pouvons efficacement faire correspondre les candidats aux offres qui correspondent le mieux à leurs domaines.

4-2-Search

Dans cette partie on fait la recherche des emplois en fonction de la localisation spécifiée par l'utilisateur. Lorsqu'une requête GET est reçue, le code extrait le paramètre de localisation des paramètres de la requête. Si une localisation est fournie, le code parcourt l'ensemble des données d'emplois, vérifie que chaque emploi a une localisation valide, puis compare les localisations de manière insensible à la casse. Les emplois correspondant à la localisation spécifiée sont collectés et leurs détails sont ajoutés à une liste de recommandations. Cette liste

est ensuite renvoyée à l'utilisateur . En cas d'erreur, comme l'absence de localisation ou une exception non prévue, des messages d'erreur appropriés sont retournés.

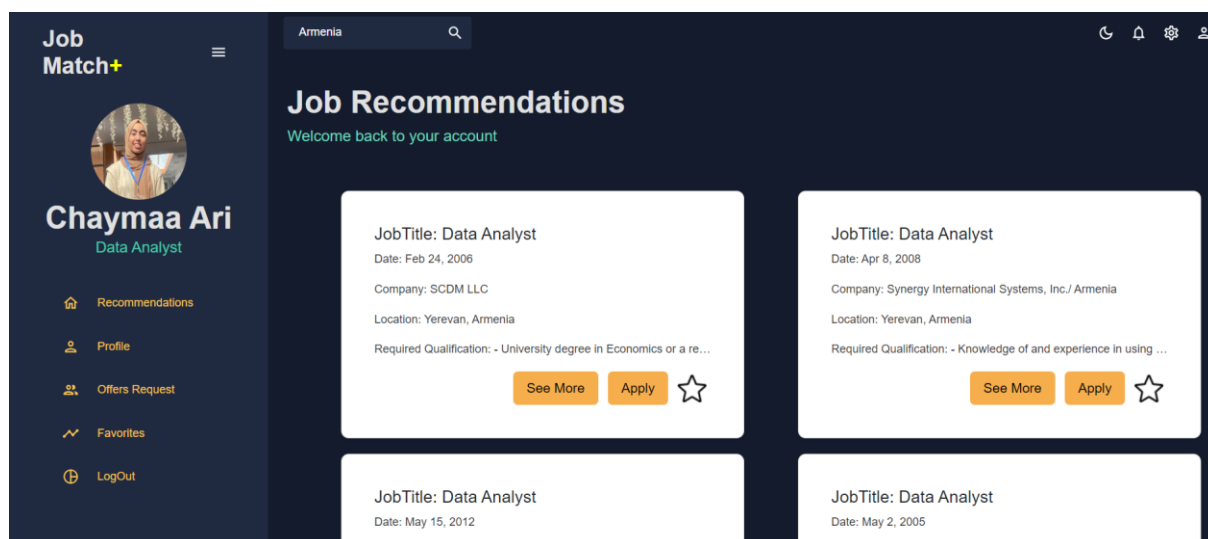


Figure 36: recherche des emplois en fonction de la localisation

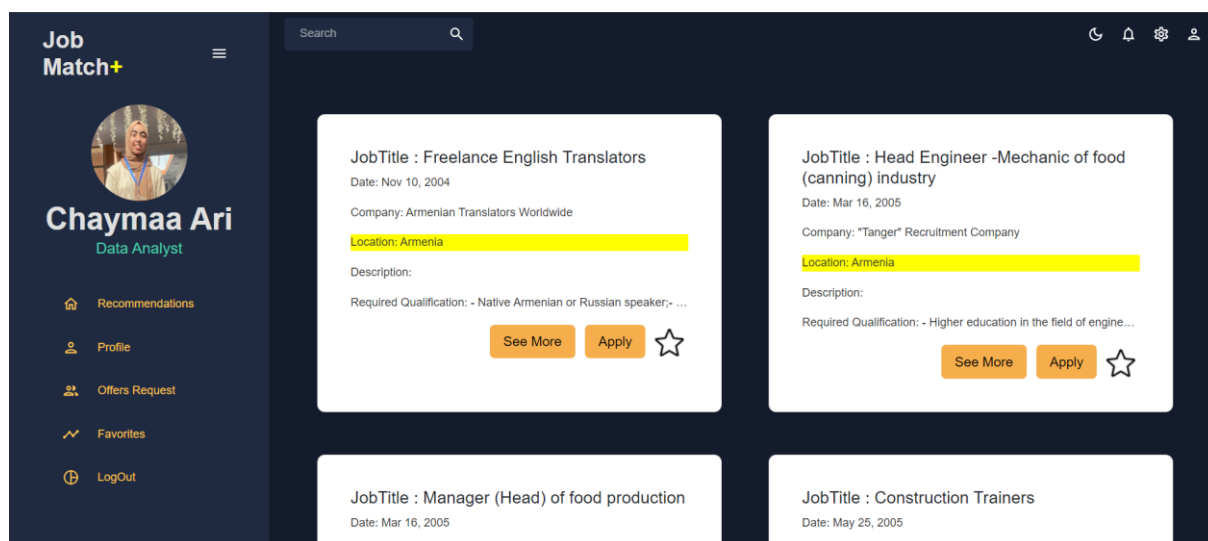


Figure 37: résultat de recherche des emplois en fonction de la localisation

4-3-See more

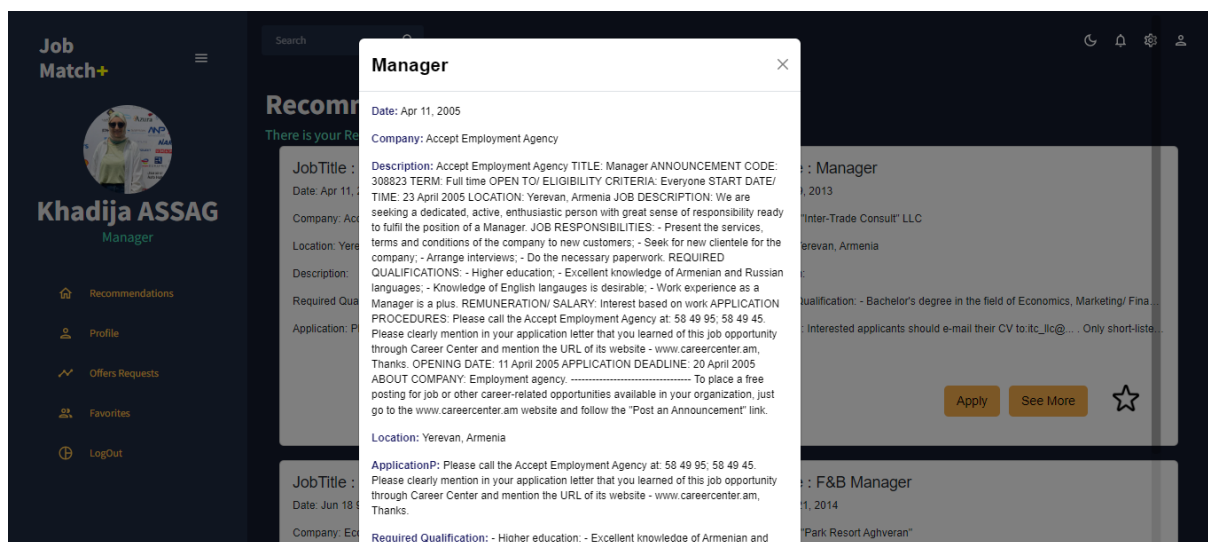


Figure 38: affichage des informations détaillées sur l'offre recommandée

Le bouton “See More” affiche des informations détaillées sur l'offre recommandée, offrant ainsi à l'utilisateur une présentation simplifiée des détails de l'offre.

4-4-Postuler

Job Match+

Search

Khadija ASSAG
Manager

Recommendations

Profile

Offers Requests

Favorites

LogOut

Apply

NOM
ASSAG

PRÉNOM
Khadija

EMAIL
khadijaassag2@gmail.com

TÉLÉPHONE
0653440613

DOMAINE
Manager

CV
Choose File No file chosen

LETTRE DE MOTIVATION
Choose File No file chosen

Envoyer

Figure 39: formulaire de postulation d'offre

Après avoir cliquer sur le bouton “Apply” le système se redirige vers cette interface de postulation sur notre site web “JobMatch+” .

Elle offre aux utilisateurs la possibilité de soumettre leur candidature pour les postes qui les intéressent en affichant **automatiquement** les informations par défaut de l'utilisateur connecté et le domaine de l'offre recommandée en ajoutant aussi des informations supplémentaires tel que la lettre de motivation.

4-5-Ajouter Favoris

L'étoile dans le card de la recommandation permet de sauvegarder l'offre et de l'ajouter à une liste de favoris, afin que l'utilisateur puisse la consulter ultérieurement ou l'utiliser dans d'autres fonctionnalités.

4-6-Profil Utilisateur

Chaymaa Ari
Data Analyst

Recommendations
Profile
Offers Request
Favorites
LogOut

FIRST NAME: Chaymaa
LAST NAME: Ari
MOBILE: 0762626939
EMAIL: chaymaa@gmail.com
CITY: Casablanca
DOMAINE: Data Analyst
DEGREE: Data Engineer
DESCRIPTION: Python, Java, Machine learning, Deep learning, IA

Selected photo : undefined
See your resume
Selected resume : undefined
UPDATE

Figure 40: profil utilisateur

l'interface profil présente les informations générales de l'utilisateur (son nom , son email , son numéro de téléphone , sa ville ,son domaine ...)il peut aussi visualiser son cv en cliquant sur “see your resume” ainsi elle lui offre la possibilité de modifier sa photo de profil , son CV ou tout autre champ rapidement en cliquant sur le bouton update en affichant une alert de succès.

localhost:3000 indique
Profile updated successfully!
OK

0762626939
CITY: Armenia
DOMAINE: Data Analyst
DEGREE: Data Engineer
DESCRIPTION: Python, Java, Machine learning, Deep learning, IA, BI

Selected photo : undefined
See your resume
Selected resume : undefined
UPDATE

Figure 41: affichage d' alerte de succès

4-7-Liste postulation

Job Match+

Khadija ASSAG
Manager

Offers Request
There is your offers Done

Name	Username	Email	Phone	Job Title	Date Offer	Action
ASSAG	Khadija	khadijaassag2@gmail.com	0653440613	Manager	2024-06-01T19:59:32.234Z	
ASSAG	Khadija	khadijaassag2@gmail.com	0653440613	Manager	2024-06-01T20:09:03.865Z	
ASSAG	Khadija	khadijaassag2@gmail.com	0653440613	Manager	2024-06-01T20:09:58.993Z	
ASSAG	Khadija	khadijaassag2@gmail.com	0653440613	Freelance English Translators	2024-06-01T20:24:24.410Z	

Figure 42: liste de postulation

La page " Offers Requests" offre une interface claire et concise pour consulter toutes les demandes de postulation soumises par l'utilisateur **connecté**.

Chaque élément de la liste présente de manière organisée les informations cruciales de l'utilisateur incluant son nom, son adresse e-mail, son numéro de téléphone, ainsi que le **titre du poste** pour lequel il a exprimé son intérêt.

L'action contient un bouton pour supprimer la demande d'offre définitivement de la base de données.

4-8-Liste Favoris

Job Match+

Chaymaa Ari
Data Analyst

Favorite Jobs
Your saved job listings

JobTitle: Data Manager
Date: Mar 11, 2013
Company: IPSC - Institute for Political and Sociological Consulti...
Location: Yerevan, Armenia
Required Qualification: - At least MA degree in Social Sciences ...
[See More](#) [Apply](#)

JobTitle: Data Coordinator
Date: Jul 11, 2006
Company: Firmplice Corporation Yerevan Branch
Location: Yerevan, Armenia
Required Qualification: - BS degree;- Fluency in English langua...
[See More](#) [Apply](#)

JobTitle: Finance Assistant

Figure 43: liste favoris

Cette interface est conçue pour permettre à un utilisateur spécifique de lister les offres d'emploi qu'il a ajoutées à ses favoris, afin qu'il puisse les consulter ultérieurement et postuler. Elle comprend également des boutons **See More** et **Apply**, permettant respectivement d'afficher plus d'informations sur les offres d'emploi et de soumettre une candidature.

5-Espace Admin

5-1-Dashboards

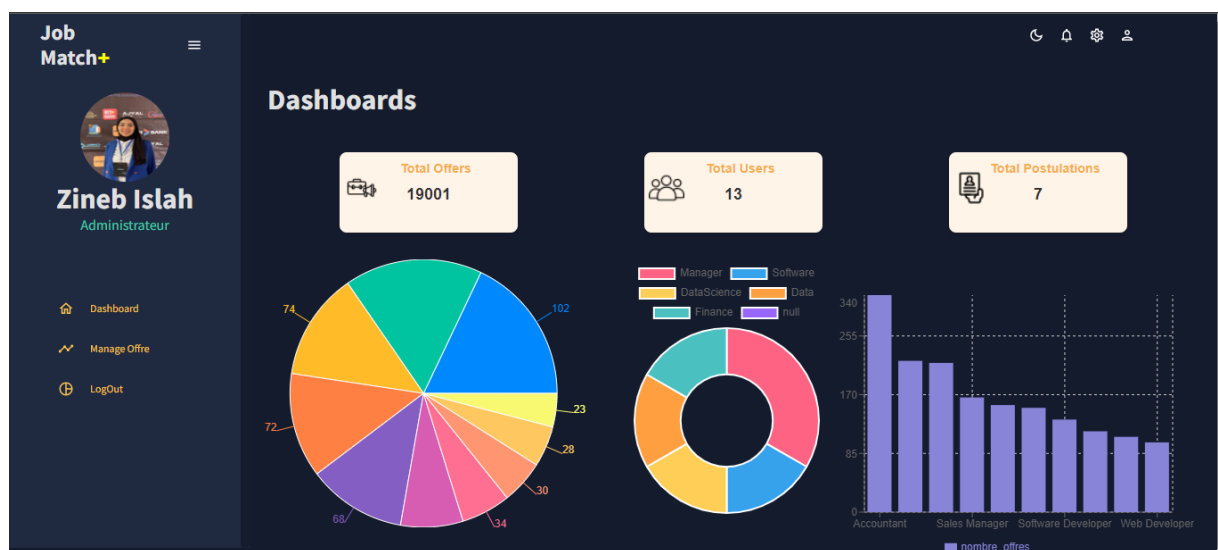


Figure 44: Dashboards

- L'interface Dashboards dans l'espace administrateur présente des statistiques et des tableaux de bord permettant de présenter visuellement des données importantes sous forme de graphiques, de diagrammes, de jauges,. Cela facilite la compréhension et l'interprétation des données par les administrateurs.
- Dans notre application, nous affichons le nombre total d'offres disponibles dans notre base de données, qui s'élève à 19 000 offres, le nombre total des utilisateurs de notre application, ainsi que le nombre total de postulations effectuées.
- Le **BarChart** est un Dashboard qui présente le nombre d'offres d'emploi par domaine. Par exemple, dans le domaine "Accountant", le nombre d'offres est de 328.
- Le **Doughnut** est un Dashboard qui affiche le nombre des utilisateurs par domaine
- Le **PieChart** affiche le nombre d'offres par ville. Par exemple dans la ville "Tbilisi Georgia" on trouve 74 offres.

5-2-Gestion Offre



Figure 45: liste des offres

La page "Gérer les Offres" est une interface conviviale conçue pour la gestion efficace des offres d'emploi. Elle présente une liste des offres disponibles, incluant le titre du poste, le nom de l'entreprise, l'emplacement et le diplôme requis, permettant ainsi aux administrateurs de visualiser rapidement les offres et leurs exigences. Ils peuvent également supprimer ou modifier une offre en un clic. De plus, un bouton d'ajout permet aux administrateurs d'ajouter facilement de nouvelles opportunités à la liste existante.

5-3-Ajout

ADD OFFER

Job Title: Manager

Company Name: Jesa

START DATE: 06/02/2024

APPLICATION DEADLINE: 06/02/2024

Company Address: CasaBlanca

Company Email: jesa@gmail.com

Salary: 2400

Required Degree:

Figure 46: ajout d'une offre

Notre interface permet d'ajouter facilement de nouvelles offres d'emploi en quelques clics. Il vous suffit de sélectionner l'option "add offre" pour être dirigé vers la page de création d'offre. Sur cette page, un formulaire clair et organisé vous permet de saisir toutes les informations pertinentes pour votre offre d'emploi. Des champs tels que le titre du poste, le nom et l'adresse de l'entreprise, les détails de contact, la date de début et la date limite de candidature sont clairement indiqués. Vous pouvez également spécifier le salaire, les exigences en matière de diplôme, les qualifications requises, les critères d'éligibilité, le mode de travail et fournir une description détaillée du poste. Cette interface intuitive simplifie le processus d'ajout d'offres d'emploi, vous permettant de publier rapidement et efficacement de nouvelles opportunités professionnelles.

5-4-Suppression

Le bouton supprimer dans l'action de la liste des offres permet de supprimer définitivement l'offre sélectionnée de notre base de données.

5-5-Update

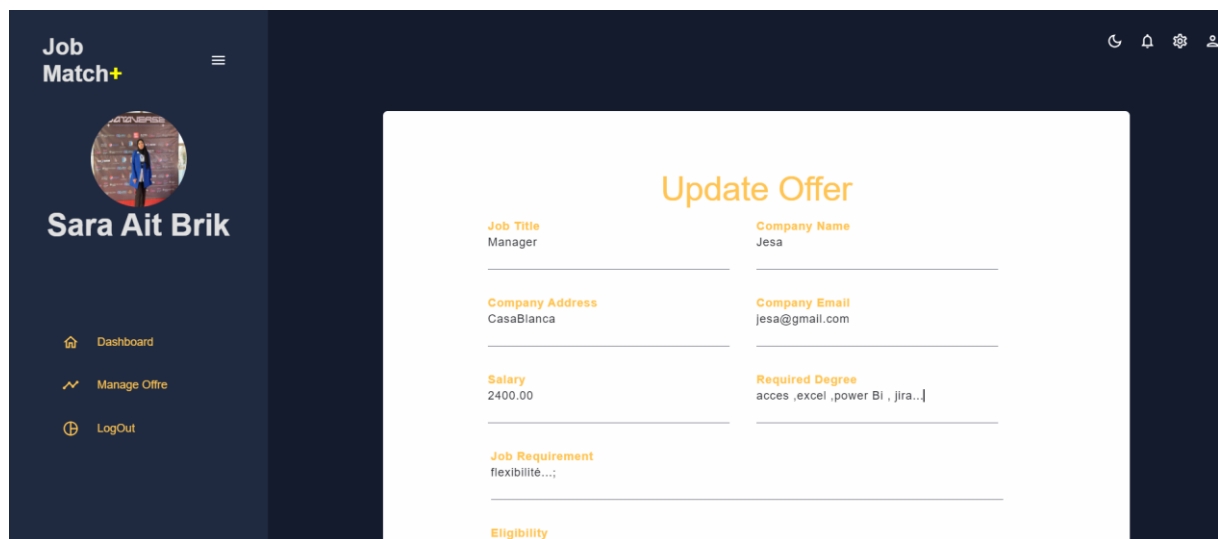


Figure 47 : formulaire de modification d'offre

En cliquant sur le bouton de la modification, on se redirige vers cette interface présentant tous les détails de l'offre actuelle. Les champs, tels que le titre du poste, le nom et l'adresse de l'entreprise, les coordonnées de contact, la date de début et la date limite de candidature, sont automatiquement remplis avec les informations précédemment saisies.

Ensuite on peut apporter les modifications nécessaires, que ce soit pour mettre à jour le salaire, les exigences en matière de diplôme, les qualifications requises, les critères d'éligibilité, le mode de travail ou la description du poste.

Conclusion

Notre application web innovante JobMatch+ a permis de transformer la recherche d'emploi en une expérience personnalisée et efficace. En intégrant des technologies avancées et des algorithmes de machine Learning, nous avons réussi à offrir des recommandations d'emploi pertinentes et sécurisées. JobMatch+ se distingue par sa capacité à simplifier et améliorer le processus de recherche d'emploi, posant ainsi les bases pour de futures améliorations et extensions.