



École Polytechnique Sousse

Filière : 5ème génie informatique

Sujet du projet

**Rapport de Projet : Assistant de Recommandation d'Emplois et
Stages Basé sur les Compétences**

Élaboré par : Sana Gouisseem Lakhal et Fedi Ben Haj Sassi

Année universitaire : 2024/2025

SOMMAIRE

1	INTRODUCTION	4
2	OBJECTIFS DU PROJET.....	4
3	TECHNOLOGIES UTILISEES.....	4
4	STRUCTURE DU PROJET	5
5	FONCTIONNALITES	5
5.1	PAGE D'ACCUEIL (INDEX.HTML)	5
5.2	DONNER COMPETENCES	6
5.3	TYPE DE RECOMMANDATIONS	7
5.4	PAGE DE RECOMMANDATION (RECOMANDATIONS.HTML)	8
6	CODE	8
6.1	APP .PY (BACKEND FLASK)	8
7	DEVELOPPEMENT ET TESTS.....	9
7.1	TEST DU STAGE	9
7.2	TEST D'EMPLOI.....	9
8	CONCLUSION	10
8.1	PERSPECTIVES D'AMELIORATION	10
9	ANNEXE.....	10
9.1	EXTRAIT DE CODE	10

Liste des figures

Figure 1 : Structure du projet	5
Figure 2 : Index.html	6
Figure 3 :Recommandations de Stage.....	7
Figure 4 :Recommandations d Emploi.....	7
Figure 5 :Recommandation donnée	8

1 INTRODUCTION

Ce projet consiste en la création d'un assistant virtuel intelligent capable de recommander des offres d'emplois et de stages basées sur les compétences saisies par l'utilisateur. En utilisant le traitement du langage naturel (NLP) avec la bibliothèque spaCy, l'application extrait les compétences spécifiques du texte fourni par l'utilisateur. Ensuite, en fonction de ces compétences, elle propose des opportunités de stages ou d'emplois, en tenant compte du type de demande (stage ou emploi) et, pour les stages, du type spécifique (été ou PFE). Ce système vise à simplifier la recherche de postes pertinents en offrant des recommandations personnalisées, rapidement accessibles via une interface web développée avec Flask.

2 OBJECTIFS DU PROJET

- **Recommandation automatisée** : Proposer des offres de stages et d'emplois en fonction des compétences des utilisateurs.
- **Extraction des compétences** : Utiliser le traitement du langage naturel pour extraire les compétences clés du texte soumis.
- **Personnalisation des offres** : Offrir des recommandations adaptées selon le type de demande (stage ou emploi) et la spécificité des stages (été ou PFE).
- **Interface intuitive** : Fournir une interface web simple et accessible via Flask pour améliorer l'expérience utilisateur.
- **Actualisation continue** : Permettre l'ajout de nouvelles compétences et la mise à jour régulière des offres.

Simplification de la recherche : Réduire le temps et les efforts nécessaires pour trouver des opportunités professionnelles pertinentes.

3 TECHNOLOGIES UTILISEES

- **Python 3.x** : Le langage de programmation principal utilisé pour le développement du backend.
- **Flask** : Framework Python léger utilisé pour construire l'API web.
- **spaCy** : Bibliothèque NLP utilisée
- **HTML/CSS** : Utilisé pour construire l'interface utilisateur.
- **Bootstrap** : Framework CSS pour rendre l'interface responsive et bien structurée.

4 STRUCTURE DU PROJET

Le projet est structuré comme suit :

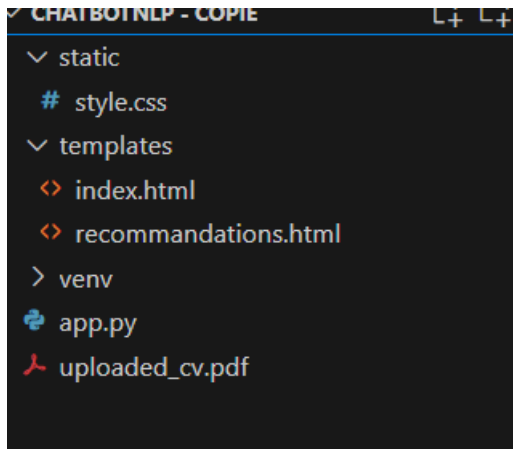


Figure 1 : Structure du projet

5 FONCTIONNALITES

5.1 PAGE D'ACCUEIL (INDEX.HTML)

La page `index.html` permet à l'utilisateur de sélectionner ses compétences et de spécifier le type de recommandation (stage ou emploi). Elle comprend :

1. Un champ de saisie pour entrer des compétences.
2. Un champ pour indiquer le type de recommandation ("stage" ou "emploi").
3. Si un stage est sélectionné, des options de type de stage (été ou PFE) sont affichées.
4. Un bouton permet d'envoyer le formulaire pour obtenir des recommandations basées sur les compétences et la demande spécifiée.

L'objectif est de guider l'utilisateur dans la sélection de ses compétences et préférences pour obtenir des offres adaptées.



Assistants d'emplois et stages

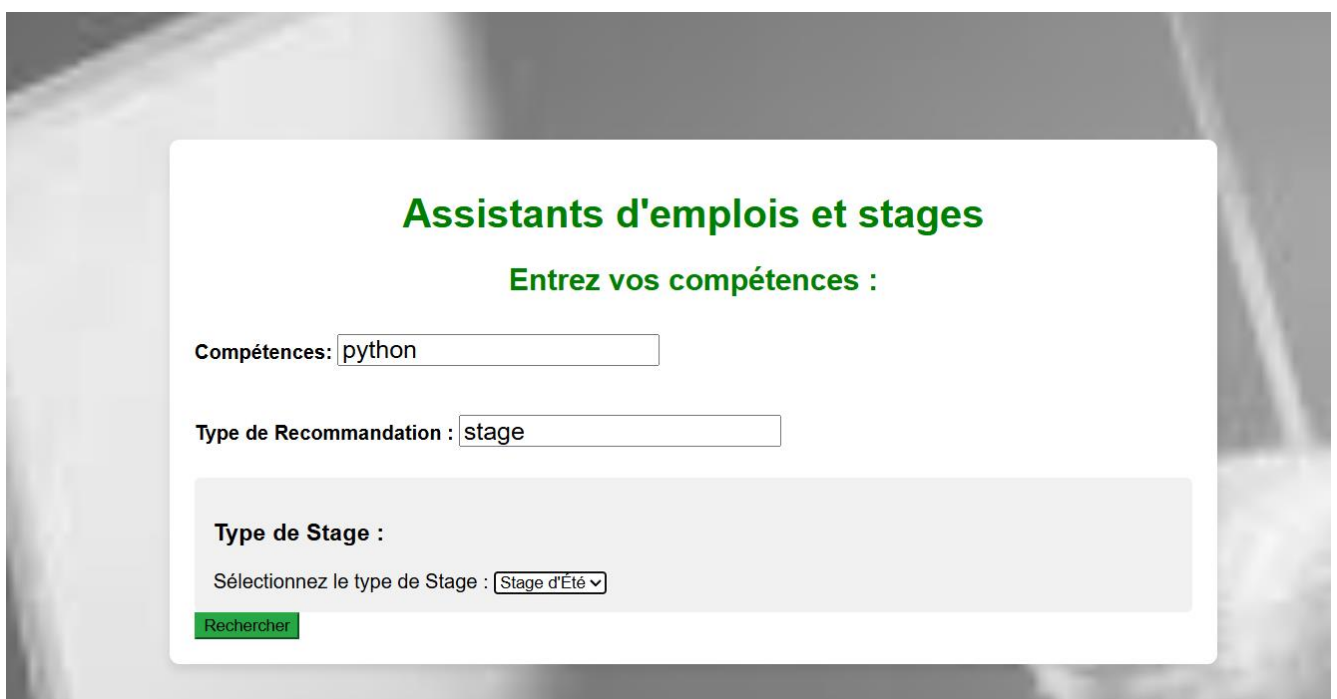
Entrez vos compétences :

Compétences:

Type de Recommandation :

Figure 2 : Index.html

5.2 Recommandation De Stage



Assistants d'emplois et stages

Entrez vos compétences :

Compétences:

Type de Recommandation :

Type de Stage :

Sélectionnez le type de Stage :

Figure 3 :Recomandations De Stage

5.3 Recommandation D Emploi



The screenshot shows a web interface for job recommendations. It features a white form box on a blurred background. The form has a title 'Assistant d'emplois et stages' in green, followed by the instruction 'Entrez vos compétences :'. There are two input fields: 'Compétences:' with the text 'python' and 'Type de Recommandation :' with the text 'emploi'. A green 'Rechercher' button is at the bottom left of the form.

Assistant d'emplois et stages

Entrez vos compétences :

Compétences:

Type de Recommandation :

Figure 4 :Recommandation D Emploi

5.4 Recommandation donné D Emploi

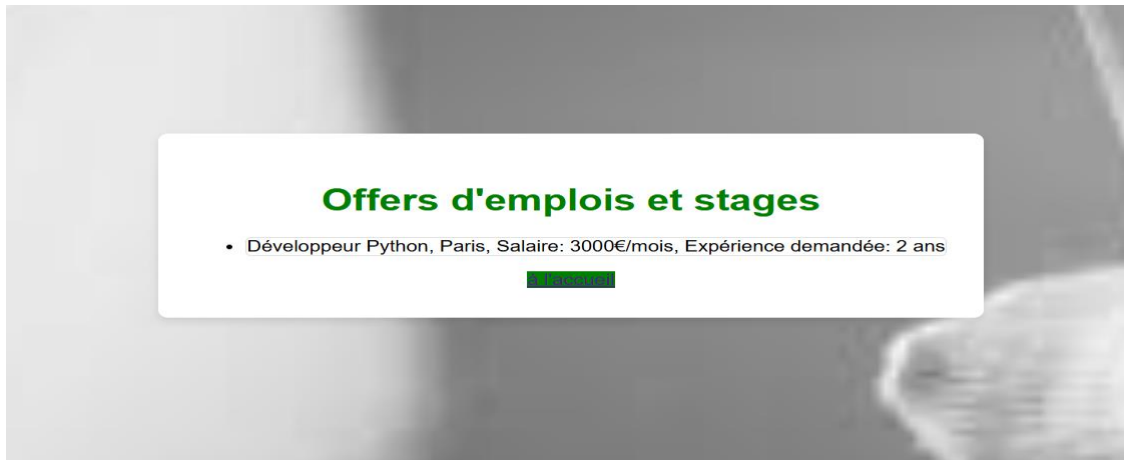


Figure 5 : Recommandation donnée de emplpoi

5.5 Recommandation donné De Stage



Figure 6 : Recommandation donnée de Stage

6.1 APP.PY (BACKEND FLASK)

- `spacy` : Utilisé pour le traitement du langage naturel (NLP), notamment pour extraire des compétences du texte.
- `Flask` : Framework web utilisé pour créer l'application web.
- `render_template`, `request`, `jsonify` : Fonctionnalités Flask pour rendre des templates HTML, récupérer des données envoyées par le formulaire, et retourner des réponses JSON en cas d'erreur.
- `nlp = spacy.load('fr_core_news_sm')` : Charge un modèle de langue français de spaCy, utilisé pour l'analyse linguistique.
- Un dictionnaire est défini avec des offres de stage et d'emploi pour différentes spécialités (comme `python`, `data scientist`, `javascript`, etc.).
- Une liste de compétences prédéfinies est créée, qui correspond aux spécialités présentes dans le dictionnaire `job_offers`
- `@app.route('/')` : Déclare la route d'accueil qui rend le template `index.html`, la page d'accueil où l'utilisateur entre ses compétences et son type de recommandation.
- `@app.route('/process_skills', methods=['POST'])` : Cette route reçoit les données soumises par l'utilisateur via un formulaire PO

7 DEVELOPPEMENT ET TESTS

7.1 TEST DU Recommandation Du stage

le processus de test consiste à vérifier manuellement que les recommandations de stage fonctionnent comme prévu et à utiliser des tests automatisés pour garantir la stabilité et la fiabilité de l'application..

7.2 TEST DU Recommandation D'Emploi

pour tester la fonctionnalité de recommandation d'emplois, vous devez tester manuellement le formulaire en saisissant des compétences et en vérifiant les résultats, tout en utilisant des tests automatisés pour garantir la validité des recommandations d'emplois basées sur les compétences spécifiées.

8 CONCLUSION

Ce projet permet de recommander des offres de stage et d'emploi en fonction des compétences des utilisateurs. L'application utilise des techniques avancées de traitement du langage naturel pour extraire les compétences à partir d'un texte. Elle fournit ensuite des suggestions personnalisées de stages ou d'emplois, en fonction du type de demande et des compétences détectées. L'interface utilisateur est intuitive et facile à utiliser, offrant une expérience fluide pour les chercheurs d'emploi et de stage. Ce système est un outil pratique pour aider les utilisateurs à trouver des opportunités adaptées à leurs compétences.

PERSPECTIVES D'AMELIORATION

améliorations rendraient le système plus performant, plus adapté aux besoins des utilisateurs, et capable de s'adapter à un environnement professionnel en constante évolution.

9 ANNEXE

9.1 EXTRAIT DE CODE

```
predefined_skills:
    skills.append(token.text.lower())
    return skills

# Fonction de recommandation d'emplois/stages
def recommend_jobs(skills, request_type, stage_type=None):
    recommendations = []
    for skill in skills:
        if skill in job_offers:
            if request_type == "stage" and stage_type:
                recommendations.append(job_offers[skill]["stage"].get(stage_type,
"Aucune offre de stage disponible."))
            elif request_type == "emploi":
                recommendations.append(job_offers[skill]["emploi"])
    if not recommendations:
        return ["Aucune offre disponible pour les compétences détectées."]
    return recommendations

# Initialiser Flask
app = Flask(__name__)

# Route pour afficher la page d'accueil
@app.route('/')

```

```

def index():
    return render_template('index.html')

# Route pour traiter la soumission des compétences et recommander des emplois/stages
@app.route('/process_skills', methods=['POST'])
def process_skills():
    try:
        # Récupérer les compétences sélectionnées
        skills_input = request.form['skills']
        skills = skills_input.split(',') # Séparer les compétences par des virgules
        skills = [skill.strip().lower() for skill in skills] # Nettoyer les espaces
        et rendre en minuscules

        # Récupérer le type de demande (stage ou emploi)
        request_type = request.form.get('request_type').lower()

        # Si le type est "stage", récupérer le champ "stage_type". Sinon, l'ignorer.
        stage_type = request.form.get('stage_type') if request_type == "stage" else
None

        # Recommander les emplois ou stages en fonction des compétences et du type de
demande
        recommended_jobs = recommend_jobs(skills, request_type, stage_type)

        # Rendre la page des recommandations
        return render_template('recommandations.html',
recommendations=recommended_jobs)
    except Exception as e:
        return jsonify({"error": str(e)}), 500

# Lancer l'application Flask
if __name__ == '__main__':
    app.run(debug=True)

```

