Cahier des Charges

Projet : Site Web de Scraping et Recommandation d'Offres d'Emploi

Introduction

Ce projet consiste à développer un site web dynamique permettant de présenter des offres d'emploi en temps réel. Le site utilisera le web scraping pour collecter les offres d'emploi depuis divers sites, les stocker dans une base de données MySQL et les afficher aux utilisateurs avec un système de recommandation basé sur l'intelligence artificielle. Le backend sera développé avec Flask.

Objectifs

Scraper les offres d'emploi en temps réel depuis plusieurs sites de recherche d'emploi.

Stocker les données d'offres d'emploi de manière structurée dans une base de données MySQL.

Fournir une interface utilisateur permettant la recherche et la visualisation des offres d'emploi.

Implémenter un système de recommandation personnalisée d'offres d'emploi.

Créer des visualisations des tendances du marché du travail.

Fonctionnalités

1. Scraping des Offres d'Emploi

Sources de données : Identifiez les principaux sites d'offres d'emploi (ex. : Indeed, Monster, LinkedIn).

Fréquence de scraping : Définir une fréquence (ex. : toutes les heures) pour le scraping des nouvelles offres.

Extraction des données : Collecter les informations suivantes :

Titre du poste

Nom de l'entreprise

Lieu

Date de publication

Description du poste

Exigences

Type de contrat

Salaire (si disponible)

2. Base de Données

Structure de la base de données :

Table Offres : ID, titre, entreprise, lieu, date de publication, description, exigences, type de contrat, salaire, source.

Table Utilisateurs : ID utilisateur, nom, email, mot de passe (haché), préférences de recherche.

Table Recommandations : ID utilisateur, ID offre, score de recommandation.

Optimisation :

Indices sur les colonnes fréquemment utilisées (ex. : titre, entreprise, lieu).

Normalisation pour réduire la redondance.

3. Interface Utilisateur

Accueil : Présentation des nouvelles offres d'emploi.

Recherche : Formulaire de recherche permettant de filtrer les offres par mots-clés, lieu, type de contrat, entreprise.

Détails de l'offre : Page de détail pour chaque offre d'emploi avec toutes les informations collectées.

Recommandations : Page de recommandations personnalisées basées sur les préférences et l'historique de l'utilisateur.

Compte utilisateur :

Inscription et connexion.

Gestion du profil et des préférences de recherche.

Historique de recherche et d'offres sauvegardées.

4. Système de Recommandation

Modèle de recommandation :

Entrée : Historique de recherche et interactions de l'utilisateur.

Sortie : Liste d'offres d'emploi recommandées avec un score de pertinence.

Algorithmes : Utiliser des techniques de machine learning (collaborative filtering, content-based filtering).

Évaluation des recommandations :

Mesurer la précision, le rappel et la satisfaction utilisateur.

Ajuster les algorithmes en fonction des retours utilisateurs.

5. Visualisations et Rapports

Tableaux de bord :

Tendances des offres d'emploi par secteur, lieu, type de contrat.

Statistiques sur les entreprises les plus actives.

Graphiques :

Distribution des types de contrat.

Moyennes salariales par secteur et par région.

Accessibilité des visualisations :

Graphiques interactifs sur le site.

Rapports téléchargeables en PDF.

Contraintes Techniques

Technologies :

Backend : Flask

Base de données : MySQL

Scraping : BeautifulSoup, Scrapy ou Selenium

Frontend : HTML, CSS, JavaScript, Bootstrap

Recommandation : Scikit-Learn, TensorFlow ou PyTorch

Sécurité :

Protection des données utilisateurs.

Sécurisation des communications (HTTPS).

Stockage sécurisé des mots de passe (hachage avec bcrypt).

Performances :

Optimisation des requêtes SQL.

Mise en cache des résultats fréquents.

Scalabilité pour gérer une grande quantité de données.

Planning Prévisionnel

Phase de Conception (2 semaines) :

Définition des exigences détaillées.

Conception de la base de données et des schémas.

Conception de l'architecture du système de recommandation.

Phase de Développement (6 semaines) :

Développement du scraping.

Implémentation du backend et de la base de données.

Développement de l'interface utilisateur.

Phase de Tests (3 semaines) :

Tests unitaires et d'intégration.

Tests de performance et de sécurité.

Tests utilisateurs et ajustements.

Phase de Déploiement (2 semaines) :

Déploiement sur un serveur de production.

Mise en place de la surveillance et de la maintenance.

Phase de Suivi et Amélioration Continue (Continu) :

Collecte des feedbacks utilisateurs.

Amélioration des fonctionnalités existantes.

Ajout de nouvelles fonctionnalités selon les besoins.

Livrables

Code source complet du projet.

Documentation technique et utilisateur.

Base de données configurée avec des exemples de données.

Rapports de tests et d'évaluation des performances.

Rapports de visualisations en format PDF.

Conclusion

Ce projet vise à démontrer la capacité à intégrer diverses compétences en développement data et intelligence artificielle. En suivant ce cahier des charges, le projet répondra aux exigences de la certification et fournira une solution fonctionnelle et efficace pour la recherche d'emploi.

Si vous avez besoin de plus de détails ou de modifications spécifiques, n'hésitez pas à me le faire savoir !