Résumé du Projet: Web crawler et Indexeur

Ce projet consiste à développer un crawler web et un système d'indexation pour collecter, organiser et rechercher des données à partir de pages web. Le crawler est un outil qui parcourt automatiquement le web (bot) pour extraire du contenu brut (fichier html par example). Ensuite, un l'indexeur traite ces données pour créer un index consultable qui permet de récupérer rapidement et efficacement les info recherchées. L'objectif principal est de permettre aux utilisateurs de rechercher des contenus spécifiques à travers une interface simple (le plus gros du travail ne sera pas le frontend, une UI très simple sera suffisante), tout en optimisant les performances de l'indexation et de la recherche. Ce projet est une introduction pratique aux concepts de base de l'indexation web et à la construction d'un système de recherche.

Crawler Web et Indexeur :

- **Crawler Web** : Un programme conçu pour parcourir systématiquement le web et collecter des données à partir de pages web. Il ne rend pas les pages web comme le fait un navigateur ; à la place, il récupère le contenu HTML pour analyse et stockage.
- **Indexeur** : Un système qui traite et organise les données collectées par le crawler. L'indexeur crée un index consultable du contenu, permettant une récupération efficace des données.
- **Interface de Recherche** : Une interface simple qui permet aux utilisateurs de faire des requêtes sur les données indexées.

Composants Clés du Crawler Web et de l'Indexeur Personnalisé :

1. **Crawler Web** :
    - **Objectif** : Collecter des données brutes (HTML, JSON, XML) à partir de pages web.
    - **Technologies** : Scrapy, BeautifulSoup…
    - **Fonctionnalité** : Récupère et analyse les pages web, suit les liens présents dans la page et répète l'opération.
2. **Stockage des Données** :
    - **Objectif** : Stocker les données collectées de manière efficace.
    - **Technologies** : Elasticsearch
    - **Fonctionnalité** : Organise les données de manière à permettre un indexage et une récupération efficaces.
3. **Indexation** :
    - **Objectif** : Créer un index consultable des données collectées.
    - **Technologies** : Elasticsearch
    - **Fonctionnalité** : Traite les données brutes, extrait les informations pertinentes (par exemple, mots-clés, métadonnées), et construit un index.
4. **Interface du Moteur de Recherche** :
    - **Objectif** : Fournir une interface conviviale pour interroger les données indexées.
    - **Technologies** : Flask/Django (pour l'interface web), Elasticsearch (pour la fonctionnalité de recherche).
    - **Fonctionnalité** : Permet aux utilisateurs de saisir des requêtes de recherche, récupère et affiche les résultats pertinents de l'index.

Exemple de workflow:

1. **Crawling** :
    - Commencer par une liste d'URLs.
    - Récupérer le contenu HTML de chaque URL.
    - Analyser le contenu pour extraire des liens vers d'autres pages.
    - Les deux dernières étapes seront répétées (automatiquement)
2. **Stockage des Données** :
    - Stocker le contenu HTML brut et les métadonnées.
    - Stocker les données pour un accès et un traitement efficaces.
3. **Indexation** :
    - Traiter les données stockées pour extraire les mots-clés et autres informations pertinentes.
    - Construire un index qui associe les mots-clés aux documents (pages web).
4. **Interface de Recherche** :
    - Fournir une interface web où les utilisateurs peuvent saisir des requêtes (interface simple type google).
    - Interroger l'index pour trouver les documents pertinents.
    - Afficher les résultats de la recherche à l'utilisateur.

Livrables attendus :

- **Code du Crawler et Documentation** : Code source du web crawler
- **Schéma de la Base de Données** : Conception du schéma de la base de données.
- **Implémentation de l'Indexation**: Code pour la fonctionnalité d'indexation et de recherche, ainsi qu'une interface utilisateur simple.
- **Rapport Analytique** : Un rapport sur les données collectées, les performances d'indexation, et la précision des recherches.
- PS : Le rendu est attendu sous forme de répertoire GitHub + Rapport (peut-être réalisé sous forme de documentation technique