Project Documentation: Job Search AI

Problem Statement

Job seekers often struggle to identify roles that align with their skill sets. While job portals and search engines provide a plethora of options, these tools often lack personalization, leading to inefficient job searches. This project aims to address this issue by analyzing a user's CV to extract relevant skills, suggest suitable job titles, and optionally scrape job listings from popular portals to provide targeted recommendations.

Requirements

Functional Requirements:

1. Users should be able to upload their CV in PDF format.
2. The application should extract text from the uploaded CV.
3. The system should analyze the CV content for predefined job-relevant skills and their associated weights.
4. Based on the identified skills, the application should suggest job titles.
5. Optionally, the application should scrape job listings from online portals (e.g., Naukri.com) based on the extracted skills.
6. Results, including extracted skills, weights, suggested job titles, and scraped job listings, should be displayed to the user.

Non-Functional Requirements:

1. The system should be user-friendly and provide clear feedback during each step.
2. The application should ensure data privacy by deleting uploaded CVs after processing.
3. The system should be scalable to handle multiple concurrent users.
4. The scraping functionality should operate efficiently without causing delays.

Technology Requirements:

- Backend: Python (Flask framework)
- Frontend: HTML with embedded Flask templates
- PDF Parsing: PyPDF2
- Web Scraping: Selenium with Chrome WebDriver
- File Handling: Werkzueg utilities
- Deployment: Localhost or cloud-based server

Approach

1. CV Upload and Text Extraction:
- A simple HTML form allows users to upload their CV in PDF format.
- The server uses PyPDF2 to extract textual content from the uploaded PDF.

2. Skill Analysis:
- A predefined dictionary maps job-relevant skills to their respective weights.

- The extracted text is analyzed to identify skills present in the CV.
- A list of detected skills and their associated weights is prepared for further processing.

3. Job Title Suggestion:
- Each detected skill is mapped to potential job titles using a predefined dictionary.
- A comprehensive list of suitable job titles is generated based on the user's skills.

4. Job Scraping:
- Selenium is used to scrape job listings from Naukri.com.
- For each skill, the scraper fetches the top 5 job titles and their links.
- These listings are added to the results for the user.

5. User Interface and Display:
- The results, including extracted skills, weights, suggested job titles, and scraped job listings, are displayed on a responsive web page.
- Clean and minimalistic UI ensures an intuitive user experience.

Results

The application processes user-uploaded CVs to:

1. Identify relevant skills and their importance (weights).
2. Suggest job titles tailored to the user's skillset.
3. Provide links to top job postings for further exploration.