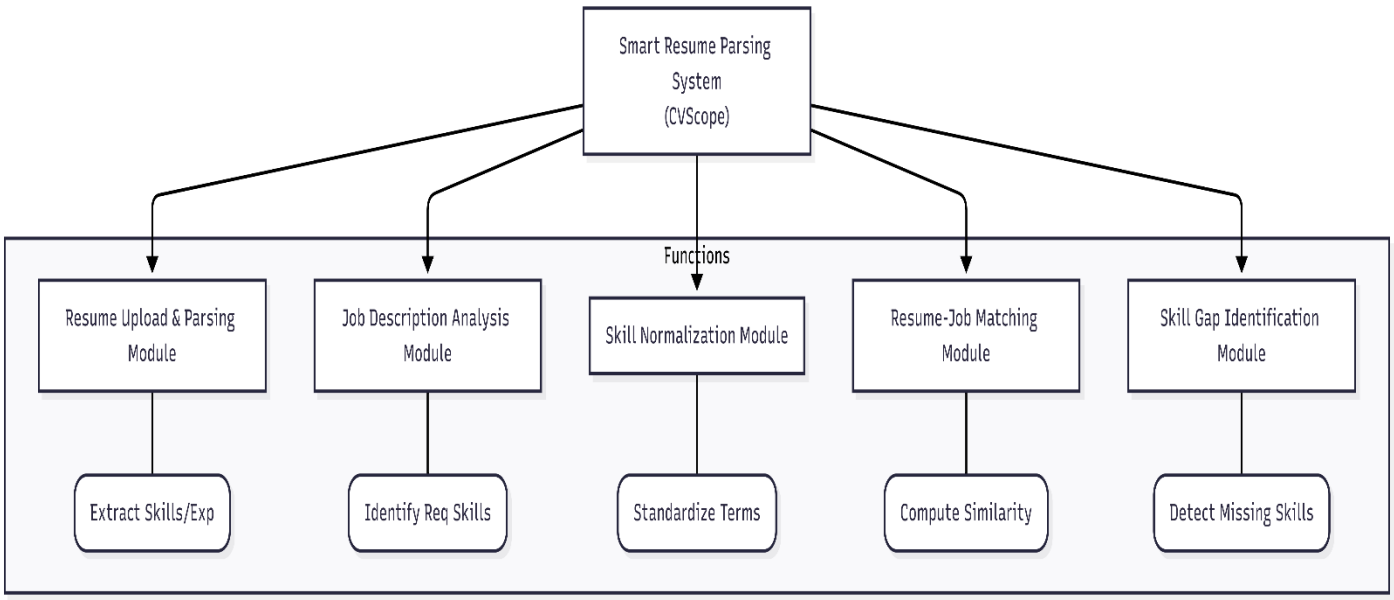
1. Introduction:

In today's competitive job market, organizations receive a large number of resumes for a single job opening. Manual screening of resumes is time-consuming, inefficient, and prone to human bias. Similarly, job seekers often struggle to understand how well their resumes match specific job requirements and which skills they lack.

The proposed system aims to automate the process of resume analysis and job matching using Natural Language Processing (NLP) and Machine Learning (ML) techniques. The system accepts resumes and job descriptions as inputs, extracts relevant information such as skills, experience, and projects, and computes a match percentage between the candidate profile and the job requirements. It also identifies missing skills, helping candidates understand gaps in their profiles.

This system improves efficiency, accuracy, and transparency in the recruitment process while providing valuable insights to job seekers.



1.1 Block Diagram of CVScope

2. An overview of the relevant literature:

Several existing systems, such as Applicant Tracking Systems (ATS), perform resume screening using keyword matching and rule-based approaches. However, traditional systems often fail to handle variations in skill names, contextual relevance, and semantic meaning.

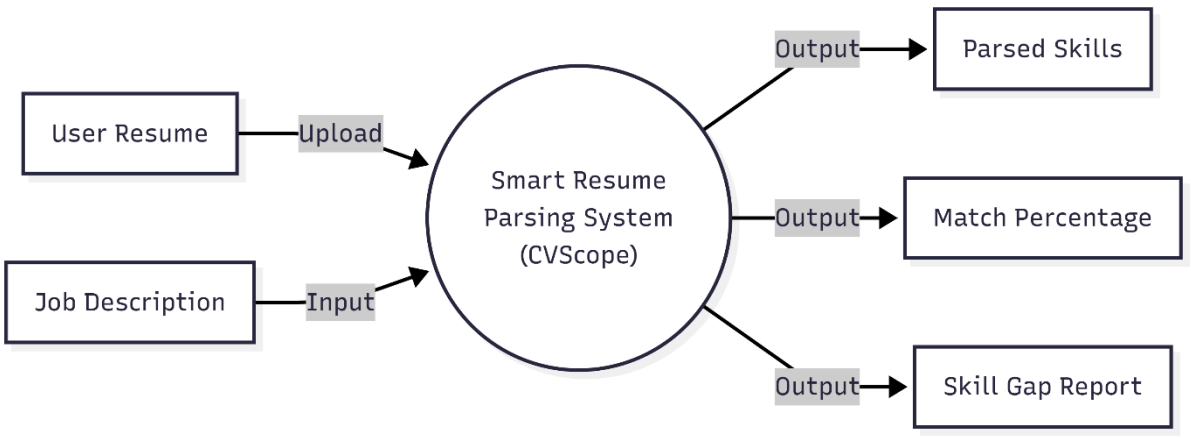
Recent studies show that NLP-based approaches using techniques like tokenization, Named Entity Recognition (NER), TF-IDF, and cosine similarity provide better accuracy in extracting and matching resume data. Research also highlights the importance of skill gap analysis for personalized career development.

Despite advancements, many existing solutions depend on expensive third-party APIs or lack transparency. This project focuses on implementing an efficient, cost-effective, and offline NLP-based system without dependency on external AI APIs.

3. Outline of the proposed work:

The proposed system consists of the following modules:

Module Name	Description
Resume Upload and Parsing Module	Extracts skills, experience, education, and projects from resumes.
Job Description Analysis Module	Identifies required and preferred skills from job descriptions.
Skill Normalization Module	Standardizes different representations of the same skill.
Resume–Job Matching Module	Computes similarity scores and match percentage.
Skill Gap Identification Module	Detects missing or partially matched skills.
Recommendation Module (Optional Future Scope)	Suggests learning resources for missing skills.



4.1 Data Flow Diagram (Level 0)

4. Experimental set up (H/W & S/W Requirements):

- Hardware Requirements
  - Processor: Intel i3 and later versions.
  - RAM: Minimum 4 GB
  - Storage: Minimum 4 GB free space
  - System: Laptop/Desktop Computer

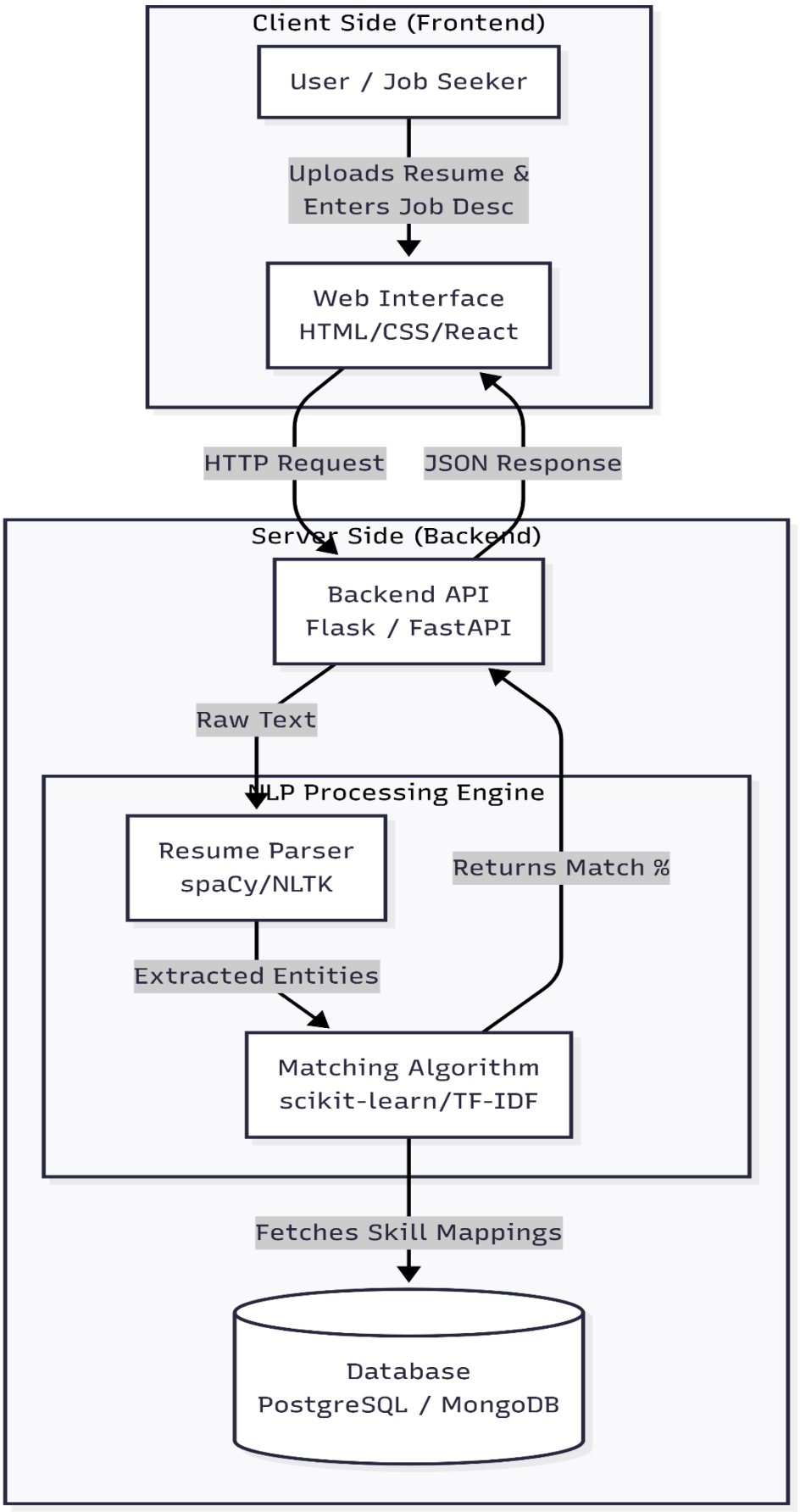
- Software Requirements
  - Operating System: Windows / Linux
  - Programming Language: Python
  - Backend Framework: Flask / FastAPI
  - Frontend: HTML, CSS, JavaScript / React
  - NLP Libraries: spaCy, NLTK
  - ML Libraries: scikit-learn
  - Database: PostgreSQL / MongoDB
  - Tools: VS Code, Git

5. Methodology / Architecture to be used:

The system follows a client–server architecture.

1. The user uploads a resume and enters a job description through the web interface.
2. The backend processes the documents using NLP techniques such as tokenization, entity recognition, and keyword extraction.
3. Extracted skills are normalized using predefined mappings and similarity measures.
4. Machine learning techniques like TF-IDF and cosine similarity are used to calculate the match percentage.
5. The system displays matched skills, missing skills, and overall job compatibility.
6. All processing is performed locally without using external AI APIs.

This modular architecture ensures scalability, maintainability, and future extensibility.



6.1 System Architecture

6. References:

1) Jurafsky, D., & Martin, J. H., *Speech and Language Processing*, Pearson.
2) Manning, C. D., Raghavan, P., & Schütze, H., *Introduction to Information Retrieval*.
3) spaCy Documentation – https://spacy.io
4) scikit-learn Documentation – https://scikit-learn.org
5) Research papers on resume parsing and ATS systems.