

## Overview

This project is designed to streamline the recruitment process by providing an automated system for job-resume matching and video interview analysis. The application focuses on two primary modules:

1. **Job Matching System:** Automates the alignment of resumes with job descriptions.
2. **Interview Analysis Module:** Evaluates interview content to assess communication style, active listening, and engagement using advanced NLP techniques.

## Design and Implementation Choices

### 1. User Interface Design

- **Objective:** Provide an intuitive, user-friendly experience.
- **Framework:** Streamlit was chosen for its simplicity and ability to rapidly create interactive web applications.
- **Design Choices:**
  - A **home page** with two buttons to navigate to either the Job Matching System or the Interview Analysis Module, simplifying navigation.
  - Each module has dedicated pages with clear sections for uploading files, processing, and displaying results.

### 2. Job Matching System

- **Objective:** Match resumes to job descriptions based on skills, experience, education, and tools/technologies.
- **Implementation:**
  - **Resume Parsing:**
    - Used python-docx for .docx resumes.
    - Used PyPDF2 for .pdf resumes.
    - Extracted relevant sections such as skills, experience, education, and tools using regex-based keyword matching.
  - **Job Description Parsing:**
    - Processed text and JSON files to extract job requirements, including skills, experience, education, and tools.
  - **Matching Algorithm:**
    - Calculated matching scores for:
      - **Skill Match:** Intersection of required and possessed skills.

- **Experience Match:** Proportional comparison of required and actual experience.
  - **Education Match:** Direct comparison of educational qualifications.
  - **Tool Match:** Intersection of required and known tools.
- The final match score is the average of the four metrics.
- **Assumptions:**
  - Job descriptions contain explicit mentions of required skills, experience, and education.
  - Resumes are structured and contain necessary information.

### 3. Interview Analysis Module

- **Objective:** Analyze the quality of interview answers from video uploads.
- **Implementation:**
  - **Video Processing:**
    - Used OpenCV to extract video frames.
    - Audio extracted using ffmpeg for transcription purposes.
  - **Transcription Analysis:**
    - Pre-trained language models from the transformers library (Hugging Face) were used to classify communication style, active listening, and engagement:
      - **Communication Style:** Multi-label classification using facebook/bart-large-mnli.
      - **Active Listening:** Analyzed engagement cues with similar BART model.
      - **Engagement:** Evaluated overall interactivity with the same architecture.
      - **Sentiment Analysis:** Used Hugging Face's sentiment-analysis pipeline for assessing sentiment polarity.
- **Assumptions:**
  - Clear audio is available in the video for transcription.
  - NLP models perform well for generic classifications without domain-specific fine-tuning.

#### 4. Vector Database for Interview Analysis

- **Objective:** Efficiently store and retrieve interview content embeddings.
- **Implementation:**
  - Used FAISS for vector similarity search.
  - Embeddings generated from pre-trained models were stored and queried for similarity comparisons.
- **Reasoning:**
  - FAISS enables fast and scalable nearest-neighbor searches, essential for handling large datasets of interview embeddings.
- **Assumptions:**
  - Embedding space is sufficiently representative of interview semantics.

#### Key Design Decisions

1. **Streamlit for UI:**
  - Simplifies deployment and allows non-technical users to interact with the system effectively.
2. **Pre-Trained Models:**
  - Chose facebook/bart-large-mnli and Hugging Face pipelines to minimize training time and resource usage while leveraging state-of-the-art performance.
3. **File Parsing:**
  - Libraries like python-docx and PyPDF2 were selected for their robust handling of common resume formats.
4. **Matching Algorithm:**
  - Used a modular approach to calculate individual match components (skills, experience, education, tools) for interpretability and scalability.
5. **Scalability:**
  - FAISS and modular pipelines ensure the system can handle large datasets of resumes, job descriptions, and interview videos.

## Challenges and Solutions

1. **Challenge:** Parsing unstructured resumes and job descriptions.
  - **Solution:** Keyword-based regex extraction and modular design to accommodate variable formats.
2. **Challenge:** Transcription quality dependent on audio clarity.
  - **Solution:** Ensure video uploads are in supported formats with audible responses.
3. **Challenge:** Balancing generality and specificity in NLP models.
  - **Solution:** Fine-tune tasks using pre-trained models and well-defined label sets.