A screenshot of a project

Description automatically generated

**IMPORTANCE :** 30% Final project (Project Proposal and Implementation 10%, Presentation 10%)!!!!!

**DEADLINE:** 18th and 19th of December!!!!

A diagram of a team of agents

Description automatically generated Draft – Plan

1. Build functionality for uploading **input files** such as job descriptions, CVs, and cover letter descriptions. As one of the system requirements is to implement **User interface** as a solution design for it could be an HTML page (use FLASK to connect with backend) **or** we can do it using terminal through the datafile folder.
   1. Develop a drag-and-drop or upload feature for users to upload:
      1. Job descriptions in PDF, DOCX, or plain text.
      2. CVs in DOCX or PDF formats.
      3. Additional descriptions or cover letter instructions.
   2. Implement a file preprocessing system to convert uploaded documents into plain text for easier analysis.
   3. Extract relevant sections from each input:
      1. For job descriptions: extract required skills, job roles, and key responsibilities.
      2. For CVs: extract skills, work experience, and education.
2. Prepare or Build Dataset for Training.
   1. Datasets:
      1. Kaggle’s job listings dataset, OpenResume, or scraping job boards.
         1. <https://www.kaggle.com/datasets/madhab/jobposts?utm_source=chatgpt.com>
         2. <https://www.kaggle.com/datasets/moyukhbiswas/job-postings-dataset?utm_source=chatgpt.com>
         3. <https://www.open-resume.com/?utm_source=chatgpt.com>
         4. <https://github.com/xitanggg/open-resume?utm_source=chatgpt.com>
         5. <https://www.kaggle.com/datasets/ravindrasinghrana/job-description-dataset?utm_source=chatgpt.com>
   2. Preprocess the Data.
      1. Normalize text by removing unnecessary formatting.
      2. Tokenize data for NLP tasks.
      3. Annotate datasets for relevant features
   3. Split Data (20/80)
3. **Develop Skill Matching System**. Create an NLP-based system that identifies skill gaps between the job description and the user’s skills.
   1. **Extract Key Information**:
      1. Use NLP libraries (e.g., spaCy, NLTK) to identify and extract job requirements and skills from job descriptions.
      2. Process CVs to extract skills, work experience, and qualifications.
   2. **Match Skills**:
      1. Use similarity measures (e.g., cosine similarity, BERT embeddings) to compare job description skills and user skills.
      2. Highlight missing or unmatched skills.
   3. **Score the Match**:
      1. Generate a matching score (0-100%) for CV-job alignment.
   4. **Integrate Feedback**:
      1. Provide suggestions to improve the match (e.g., add specific skills to the CV, adjust keywords).
4. **Create Content Generation System.** Automatically generate personalized resumes and cover letters tailored to the job description (Can be adjusted steps, cos I use CHAT4o to generate this section).
   1. **Fine-Tune Generative AI Models:**
      1. Use GPT-4 or similar models fine-tuned with your datasets for resume and cover letter generation.
   2. **Resume Generation:**
      1. Rearrange user-provided CV content to emphasize relevant experience and skills.
      2. Use a template-driven approach for formatting.
   3. **Cover Letter Generation:**
      1. Generate tailored cover letters using user-provided descriptions and job information.
      2. Structure user strengths and align them with the job description.
   4. **Iterate:**
      1. Incorporate feedback from the skill matching and feedback agent for further refinements.
5. **Feedback and Refinement System.** Build a feedback loop to improve generated documents.
   1. **Integrate a Feedback Agent:**
      1. Use rule-based or machine learning models to evaluate document quality.
   2. **Score the Output:**
      1. Provide a score (0-100%) for the generated CV and cover letter. Genereted cv or ect has to go through the **test**.
   3. **Suggestions for Improvement (Optional, can generate already improved version of cv):**
      1. Suggest adding specific skills, rephrasing sentences, or restructuring the format.
   4. **User Iteration:**
      1. Allow users to incorporate feedback and regenerate documents.
6. **Develop the User Interface.(Under discussion).**
   1. If we have time it could be HTML page with frontend if not can be through terminal\*
7. **Environment.** 
   1. Clone GitHub Link : <https://github.com/Yehormann/NLPFinalProject.git>
   2. Deploy project on **Master** branch only not **Main**.

Reference to Read before begin:

* <https://towardsdatascience.com/a-comprehensive-guide-to-collaborative-ai-agents-in-practice-1f4048947d9c>
* <https://www.deeplearning.ai/short-courses/multi-ai-agent-systems-with-crewai/>