PROJECT AND TEAM INFORMATION

Project Title

(Try to choose a catchy title. Max 20 words).

AI-Based Resume Shortlisting System

Student / Team Information

Team Name: *Team #*	*PYTHON.AI*
Team member 1 (Team Lead) *(Last Name, name: student ID: email, picture):*	*DWIVEDI, SANU KUMAR – 24391228* *sanukumardwivedi2515@gmail.com*
Team member 2 *(Last Name, name: student ID: email, picture):*	*SUNIL KUMAR – 24391314* *sy2996597@gmail.com*

Team member 2 *(Last Name, name: student ID: email, picture):*	*RISHU – 24391028* *rishuKT3107@gmail.com*

PROPOSAL DESCRIPTION (10 pts)

Motivation (1 pt)

(Describe the problem you want to solve and why it is important. Max 300 words).

With the increasing number of job applications, HR professionals face difficulties in efficiently shortlisting candidates. An AI-based system can automate the resume screening process, ensuring fairness, accuracy, and efficiency.

State of the Art / Current solution (1 pt)

(Describe how the problem is solved today (if it is). Max 200 words).

Currently, resume shortlisting is done manually, which is time-consuming and prone to bias. Some companies use keyword-based filtering, but it often lacks contextual understanding. AI-based solutions, utilizing NLP, can enhance the accuracy of resume parsing.

Project Goals and Milestones (2 pts)

(Describe the project general goals. Include initial milestones as well any other milestones. Max 300 words).

- ***Goal:*** *Develop an AI-powered resume shortlisting system.*
- ***Milestones:***
  - *Week 1-2: Research and data collection*
  - *Week 3-4: Implement resume parsing using NLP*
  - *Week 5-6: Develop a web-based interface for resume upload*
  - *Week 7-8: Test and optimize system performance*

Project Approach (3 pts)

(Describe how you plan to articulate and design a solution. Including platforms and technologies that you will use. Max 300 words).

We will use Python, Flask/Django for the web interface, and NLP models (spaCy/BERT) for skill extraction. The system will analyze resumes, match them against job descriptions, and rank candidates based on relevancy. SQLite/PostgreSQL will be used for data storage.

System Architecture (High Level Diagram)(2 pts)

(Provide an overview of the system, identifying its main components and interfaces in the form of a diagram using a tool of your choice).

The system consists of the following main components:

- ***Frontend (Flask/Django Web Interface):*** *Allows users to upload resumes.*
- ***Backend (Python, NLP Models):*** *Processes resumes, extracts skills, and matches them with job requirements.*
- ***Database (SQLite/PostgreSQL):*** *Stores candidate details, extracted skills, and scores.*
- ***AI Processing Layer (spaCy/BERT):*** *Enhances skill extraction and relevance scoring.*
- ***Ranking Module:*** *Assigns a score to candidates based on skill-job match.*

Project Outcome / Deliverables (1 pts)

(Describe what are the outcomes / deliverables of the project. Max 200 words).

- ***A web-based AI resume screening system***
- ***Automated resume parsing and ranking mechanism***
- ***User-friendly interface for resume upload***
- ***Database for storing and retrieving candidate scores***

Assumptions

(Describe the assumptions (if any) you are making to solve the problem. Max 100 words)

- ***Resumes are in structured formats (PDF, DOCX, etc.).***
- ***Job descriptions will be provided for comparison.***
- ***AI model training data is sufficient for accurate predictions.***

References

(Provide a list of resources or references you utilised for the completion of this deliverable. You may provide links).

- ***spaCy NLP Documentation***
- ***Flask/Django Documentation***
- ***AI Resume Parsing Research Papers***