Project Report: Resume Parser

1. Introduction: The Resume Parser project aims to develop a tool that helps Human Resources (HR) professionals quickly evaluate job applicants' resumes based on a provided Job Description (JD). The tool extracts information from resumes, matches it against the JD, and assigns a star rating to each resume, indicating the degree of compatibility.

2. Objectives:

- Create a Flask web application to upload resumes and a basic JD.
- Extract candidate information, including name and email, from resumes.
- Compare extracted skills with the JD and calculate a star rating.
- Display the results in a user-friendly table.

3. Technologies Used:

- Python (Flask framework)
- HTML/CSS
- PyMuPDF for PDF processing
- Regular expressions for text extraction

4. Implementation:

4.1 Resume Processing: Resumes are uploaded and processed using PyMuPDF to extract text content. The candidate's name, email, education, are extracted using custom functions.

We can use **pyresparser**, instead of custom functions.

4.2 Star Rating: The match ratio is used to calculate a star rating, capped at a maximum of 5 stars. The higher the match ratio, the higher the rating.

4.3 User Interface: A simple Flask web interface allows HR professionals to upload resumes and a JD. The results are displayed in a table, showing the candidate's name, email, and rating.

5. Results and Discussion: The Resume Parser provides a quick and efficient way for HR professionals to assess multiple resumes. The star rating offers a visual representation of the match between candidate skills and job requirements.

6. Future Improvements:

- Enhanced skill extraction using more advanced natural language processing techniques.
- Incorporate user feedback for continuous improvement.
- Integration with external databases for long-term storage and analysis.

7. Conclusion: The Resume Parser project successfully creates a practical tool for HR professionals to streamline the resume screening process. By automating the initial screening, it allows HR teams to focus on more strategic aspects of candidate selection.

8. Acknowledgments: We express our gratitude to the Flask and PyMuPDF open-source communities for providing the tools and resources necessary for the successful implementation of this project.

9. References:

- Flask Documentation: https://flask.palletsprojects.com/
- PyMuPDF Documentation: https://pymupdf.readthedocs.io/
- pyresparser Documentation: https://github.com/OmkarPathak/pyresparser [If you want to use it.]