AI-Powered Resume Ranker using NLP & Flask

📌 Introduction:

Hiring processes involve screening numerous resumes to shortlist suitable candidates. Manual shortlisting is time-consuming, inconsistent, and often biased. This project automates the resume ranking process using Natural Language Processing (NLP) to evaluate how well each resume matches a given job description.

📄 Abstract:

The AI-Powered Resume Ranker reads multiple PDF resumes and a job description input by HR. It applies NLP techniques (like TF-IDF vectorization and SpaCy-based preprocessing) to compare the similarity of resumes to the job requirements. Candidates are scored and ranked accordingly. The application features a simple web interface using Flask, where HR can upload resumes and download ranked results in a CSV format.

⚙️ Tools & Libraries Used:

Category	Tools & Libraries
Language	Python
Web Framework	Flask
NLP	SpaCy, Scikit-learn (TF-IDF)
File Handling	PDFMiner
Data Handling	Pandas, NumPy
Frontend	HTML/CSS (inside Flask templates)

🔨 Steps Involved in Building the Project:

✅ Step 1: PDF Resume Text Extraction

- Used `pdfminer.six` to extract raw text from uploaded `.pdf` resumes.

✅ Step 2: NLP Preprocessing

- Cleaned and tokenized text using `SpaCy` (stopword removal, lowercasing, lemmatization).

✅ Step 3: Vectorization

- Applied TF-IDF to convert resume and job description into comparable vectors.

✅ Step 4: Similarity Scoring

- Used cosine similarity to measure how closely each resume matches the job description.

✅ Step 5: Ranking & CSV Export

- Sorted resumes by relevance score and exported the results to `ranked_resumes.csv`.

✅ Step 6: Web Interface using Flask

- Developed a form to upload multiple resumes and input job description.
- After submission, ranked results are displayed and downloadable as CSV.

📊 Sample Output:

- A table showing candidate resume filenames, relevance scores, and ranks.
- Option to download full ranked list.

✅ Final Deliverables:

- `app.py`, `extract_text.py`, `preprocess.py`, `rank_resumes.py`
- HTML template in `templates/index.html`
- `requirements.txt` with pinned versions
- `README.md` for GitHub
- `ranked_resumes.csv` as output

✅ Conclusion:

The project successfully demonstrates how NLP and basic machine learning techniques can streamline HR operations. This tool reduces the time required to shortlist resumes and provides consistent, unbiased ranking based on job relevance. It is scalable, interpretable, and can be further enhanced with Named Entity Recognition (NER)