REPORT ON

SMART RESUME PARSER

INTRODUCTION: Recruitment processes involve screening large volumes of resumes, which is often a time-consuming and error-prone task when done manually. Traditional methods of reviewing resumes lack consistency and scalability, especially when dealing with unstructured data across different resume formats. The Smart Resume Parser project addresses this challenge by providing an automated solution to extract relevant candidate information.This system employs **Python-based tools** such as PyMuPDF and python-docx for text extraction, and **spaCy NLP models** combined with regular expressions to identify and segment critical information like skills, education, and experience. The processed data is structured into user-friendly formats, enabling faster decision-making for recruiters. A lightweight **Streamlit web application** is integrated, offering an intuitive interface for uploading resumes and visualizing parsed results. By exporting structured data into CSV or JSON files, the parser also supports further integration with Applicant Tracking Systems (ATS).The Smart Resume Parser demonstrates how combining text-processing techniques with NLP can automate repetitive HR tasks, minimize bias, and streamline candidate shortlisting for modern recruitment needs.

ABSTRACT: The Smart Resume Parser is an intelligent system designed to automatically extract structured information such as skills, education, and professional experience from resumes in PDF and DOCX formats. By leveraging Natural Language Processing (NLP) with spaCy, along with text extraction libraries like PyMuPDF and python-docx, the parser transforms unstructured text into machine-readable formats. The extracted data is organized into JSON or tabular outputs, making it easier for recruiters and HR teams to analyze candidate profiles efficiently. The system also provides an interactive Streamlit-based user interface for uploading resumes, viewing extracted results, and exporting them into CSV or JSON files. This automation reduces manual effort in resume screening, improves accuracy, and enhances recruitment workflows.

TOOLS USED: **Python** – Core programming language for implementing text extraction, preprocessing, and NLP tasks.

- **PyMuPDF (fitz)** – To extract text content from resumes in **PDF** format.
- **python-docx** – To read and extract text from resumes in **DOCX** format.
- **spaCy** – For **Natural Language Processing (NLP)** tasks such as named entity recognition, tokenization, and extracting skills, education, and experience.
- **Regular Expressions (Regex)** – To identify and extract structured patterns like emails, phone numbers, dates, etc.
- **Streamlit** – To build a simple and interactive **user interface** for uploading resumes, viewing parsed results, and exporting outputs.
- **Pandas** – For organizing extracted information into **tables (DataFrames)** and enabling easy export to CSV/JSON.
- **JSON / CSV** – For storing and exporting structured parsed data.

STEPS INVOLVED IN BULIDING THE PROJECT:

1. **Requirement Analysis**
    - Identify the objectives of the parser (extract skills, education, and experience).
    - Select suitable libraries and frameworks (PyMuPDF, docx, spaCy, Streamlit).
2. **Data Collection**
    - Gather sample resumes in **PDF** and **DOCX** formats.
    - Prepare at least 5 test resumes for evaluation.
3. **Text Extraction**
    - Use **PyMuPDF** for extracting text from PDF files.
    - Use **python-docx** for extracting text from DOCX files.
4. **Text Preprocessing**
    - Clean extracted text (remove special characters, unwanted line breaks, extra spaces).
    - Normalize text (lowercasing, standard formatting).
5. **Information Extraction (NLP + Regex)**
    - Apply **spaCy NLP models** to detect entities like names, organizations, and dates.
    - Use **regex patterns** for structured elements (emails, phone numbers, dates).
    - Extract and classify sections: **Skills, Education, Experience**.
    - r accuracy.
6. **Output and Export**
    - Provide final outputs in both **CSV** and **JSON** formats.
    - Ensure compatibility with further recruitment/ATS systems.

CONCLUSION:

The Smart Resume Parser successfully automates the process of extracting structured information such as **skills, education, and experience** from unstructured resumes in PDF and DOCX formats. By integrating **PyMuPDF**, **python-docx**, and **spaCy NLP**, the system ensures accurate text extraction and classification. The addition of **regex patterns** enhances the detection of contact details and other structured fields.