**RESUMEPARSER**

A MINI-PROJECT REPORT

*Submitted by*

**AKSHAY KARTHICK 2116210701322**

**SELESTIN VINCENT RAJ 2116210701236**

# KAJA MOHIDEEN 2116210701514

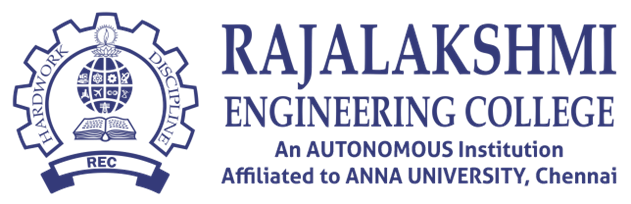
*In partial fulfillment of the award of the degree*

*of*

BACHELOR OF ENGINEERING

**IN**

# COMPUTER SCIENCE AND ENGINEERING



**RAJALAKSHMI ENGINEERING COLLEGE**

# AUTONOMOUS, CHENNAI JAN – MAY 2024

## ABSTRACT

The resume parser project aims to develop an advanced tool for automating the extraction and analysis of information from resumes or CVs submitted by job applicants. Leveraging state-of-the-art Natural Language Processing (NLP) techniques, the parser efficiently parses various sections of resumes, including personal details, work experience, education, skills, and certifications. The parsed data is then structured and organized for further processing, facilitating seamless integration with applicant tracking systems (ATS) or other recruitment platforms. By reducing manual effort and improving efficiency in the hiring process, the resume parser enhances productivity for recruiters and hiring managers. Machine learning algorithms are employed to continuously refine parsing accuracy and adapt to diverse resume formats and languages. Challenges such as handling complex layouts and non-standardized data are addressed through robust parsing algorithms. Additionally, privacy and security standards are prioritized to ensure the confidentiality of applicant information. Overall, the resume parser project represents a pivotal advancement in modern recruitment strategies, empowering organizations to make informed and data-driven hiring decisions while streamlining the candidate selection process.

**INTRODUCTION**

Automating the extraction and analysis of crucial information from resumes The resume parser project signifies a pioneering effort aimed at or CVs. It underscores the fusion of cutting-edge technology and digital innovation to streamline the recruitment process effectively.

In response to the challenges encountered in traditional resume screening methods, the project leverages advanced Natural Language Processing (NLP) techniques to parse and categorize diverse resume sections comprehensively. This includes personal details, work history, educational qualifications, skills, and certifications.

By incorporating machine learning algorithms, the project ensures continuous enhancement in parsing accuracy and adaptability to various resume formats and languages. Moreover,

privacy and security measures are prioritized to safeguard applicant data confidentiality.

In summary, the resume parser project represents a significant advancement in recruitment strategies, offering recruiters and hiring managers a sophisticated tool to expedite candidate selection while ensuring consistency and efficiency in the evaluation process.

## LITERATURE SURVEY

The literature survey for eResume parser project encompasses an in-depth exploration of existing research, methodologies, and technologies relevant to automating the parsing and analysis of electronic resumes CVs. This survey serves as a foundation for understanding the state-of-the-art approaches and identifying potential avenues for innovation and improvement within the field.

1. **Natural Language Processing (NLP) Techniques:**

Numerous studies have explored the application of NLP techniques for parsing and extracting information from unstructured text, including resumes. Techniques such as named entity recognition, part-of-speech tagging, and semantic analysis are commonly employed to identify and categorize key resume components.

**2.Machine Learning Algorithms for Resume Parsing:** Research has demonstrated the efficacy of machine learning algorithms, including support vector machines, decision trees, and deep learning models, in automating the parsing process. These algorithms can learn to recognize patterns and structures within resumes, facilitating accurate extraction of relevant information.

3.**Integration with Applicant Tracking Systems (ATS**): Integration with ATS platforms is crucial for seamless recruitment workflow. Research has focused on developing parsers compatible with popular ATS systems, enabling automatic transfer of parsed resume data for further processing and analysis.

4**.Privacy and Security Considerations**: With the increasing emphasis on data privacy and security, studies have explored methods for ensuring the confidentiality of applicant information during the parsing process. Techniques such as data anonymization, encryption, and access control mechanisms are essential for protecting sensitive data.

5**.Cross-lingual and Multimodal Resume Parsing**: As globalization continues to influence the job market, there is growing interest in cross-lingual and multimodal resume parsing. Research in this area explores techniques for parsing resumes written in multiple languages and incorporating additional modalities such as images and videos.

**6.Evaluation Metrics and Benchmark Datasets:** Evaluating the performance of resume parsers requires standardized metrics and benchmark datasets. Studies have proposed various evaluation metrics, including precision, recall, and F1-score, and curated datasets for benchmarking parser performance across different domains and languages.

In summary, the literature survey highlights the diverse range of research efforts aimed at automating the parsing and analysis of electronic resumes. By synthesizing insights from existing studies, the eResume parser project can leverage established methodologies and technologies to develop a robust and efficient parsing system tailored to the needs of modern recruitment processes.

**EXISTING SYSTEM**

The existing system refers to the current state of technology or methodology that is currently in use for parsing and analyzing resumes or CVs. Here's an overview of the existing system:

The existing system for resume parsing typically involves manual or semi-automated methods for extracting information from resumes submitted by job applicants. In many cases, human recruiters manually review each resume, extract relevant details such as work experience, education, skills, and certifications, and enter them into an applicant tracking system (ATS) or database for further processing.

Some organizations may use semi-automated tools or software to assist in resume parsing. These tools often rely on keyword-based approaches or simple pattern matching algorithms to identify and extract specific information from resumes. However, these methods are limited in their ability to accurately parse resumes with diverse formats, languages, or layouts.

Additionally, the existing system may lack integration with advanced technologies such as natural language processing (NLP) and machine learning (ML), which can significantly improve the accuracy and efficiency of resume parsing. Without these technologies, the existing system may struggle to handle large volumes of resumes, leading to delays in the recruitment process and potential errors in candidate evaluation.

Overall, while the existing system may suffice for small-scale recruitment operations or organizations with limited resources, it often falls short in

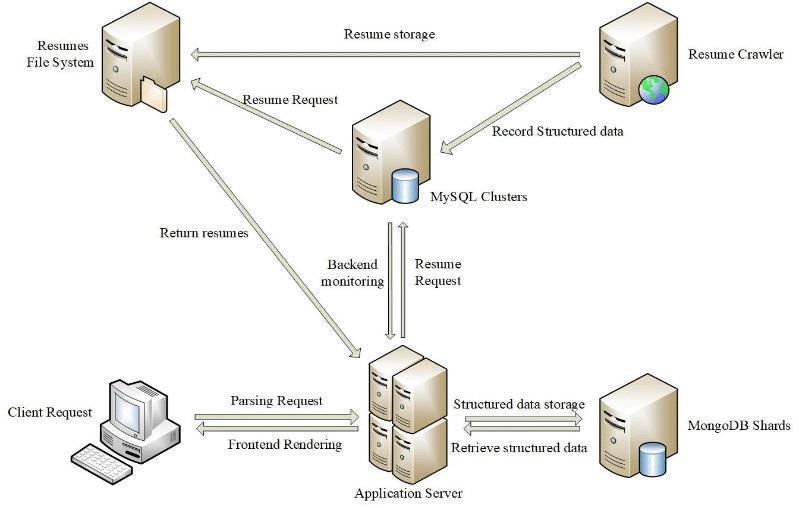
**PROPOSED SYSTEM**

The proposed system aims to revolutionize the process of resume parsing by leveraging advanced technologies such as natural language processing (NLP), machine learning (ML), and artificial intelligence (AI) to automate and optimize the extraction and analysis of information from resumes or CVs. Here's an overview of the proposed system:

1. **Advanced NLP Techniques**: The proposed system will utilize advanced NLP techniques to accurately parse and extract information from resumes, including personal details, work experience, education, skills, and certifications. These techniques will enable the system to understand the context and semantics of resume content, leading to more accurate parsing results.
2. **Machine Learning Algorithms:** Machine learning algorithms will be employed to continuously improve the parsing accuracy and adaptability of the system. These algorithms will learn from past parsing experiences and user feedback to enhance performance over time, ensuring that the system can handle diverse resume formats, languages, and layouts effectively.
3. **Semantic Analysis and Contextual Understanding:** The proposed system will go beyond simple keyword matching and incorporate semantic analysis and contextual understanding to extract meaning and insights from resume content. This will enable the system to identify relevant skills, experiences, and qualifications, even in resumes with complex or non-standard structures.
4. **Integration with Applicant Tracking Systems (ATS):** Seamless integration with ATS platforms will be a key feature of the proposed system, enabling automatic transfer of parsed resume data for further processing and analysis. This integration will streamline the recruitment workflow, improve data consistency, and enhance collaboration among recruiters and hiring managers.
5. **Scalability and Efficiency:** The proposed system will be designed to handle large volumes of resumes efficiently, ensuring fast processing times and minimal manual intervention. This scalability will enable organizations to manage recruitment processes more effectively, even during periods of high candidate influx.

Overall, the proposed system represents a significant advancement in resume parsing technology, offering organizations a sophisticated and efficient solution for candidate selection and recruitmen

**SYSTEM ARCHITECTURE**



## MODULES

Modules for building a resume parser typically involve a combination of natural language processing (NLP), machine learning, and data extraction techniques. Here are some key modules commonly used in building a resume parser:

**Text Preprocessing:** Module for cleaning and normalizing text data, including tasks such as removing stop words, punctuation, and special characters, as well as tokenization and stemming.

**Information Extraction**: Module for extracting relevant information from resumes, such as contact details (name, address, phone number, email), education history, work experience, skills, certifications, etc. This often involves techniques like named entity recognition (NER), part-of-speech (POS) tagging, and dependency parsing.

**Entity Recognition:** Module for identifying entities mentioned in the resume, such as person names, company names, job titles, universities, and dates. This is often achieved using NER models trained on annotated data.

**Resume Structure Analysis**: Module for analyzing the structure of the resume, including sections such as education, experience, skills, and projects. This may involve rule-based methods or machine learning models.

**Keyword Extraction:** Module for extracting keywords or key phrases from the resume that indicate important skills, technologies, or experiences. This can help in matching resumes to job descriptions or in categorizing resumes.

## IMPLEMENTATION AND RESULTS

**To implement a resume parser, you can follow these steps without diving into specific code details:**

**### 1. Define Required Information**

**Identify the key details you need to extract from resumes, such as:**

* **Personal information (name, contact details)**
* **Professional summary**
* **Work experience (job titles, companies, dates, responsibilities)**
* **Education (degrees, institutions, dates)**
* **Skills**
* **Certifications**
* **Languages**
* **Projects**

**### 2. Choose Tools and Libraries**

**Select appropriate tools and libraries for handling different resume formats and processing text:**

* **\*\*PDFs\*\*: Use libraries like `PyMuPDF` or `pdfminer.six`.**
* **\*\*DOCX\*\*: Use `python-docx`.**
* **\*\*TXT\*\*: Use Python's built-in file handling.**

**### 3. Extract Text from Resumes**

**Develop functions to read and convert different file formats into plain text:**

* **For PDFs, use a PDF parsing library to extract text.**
* **For DOCX files, use a DOCX library to read the text.**
* **For TXT files, read the content directly.**

**### 4. Process Text with NLP**

**Utilize an NLP library to parse the extracted text and identify key entities:**

* **Use a library like `spaCy` to process the text and extract entities such as names, organizations, dates, etc.**

**### 5. Post-process Extracted Data**

**Refine the extracted data to ensure it meets your structured format:**

* **Validate and normalize dates.**
* **Standardize job titles and company names.**
* **Normalize skill names and other relevant fields.**

**### 6. Store or Use Parsed Data**

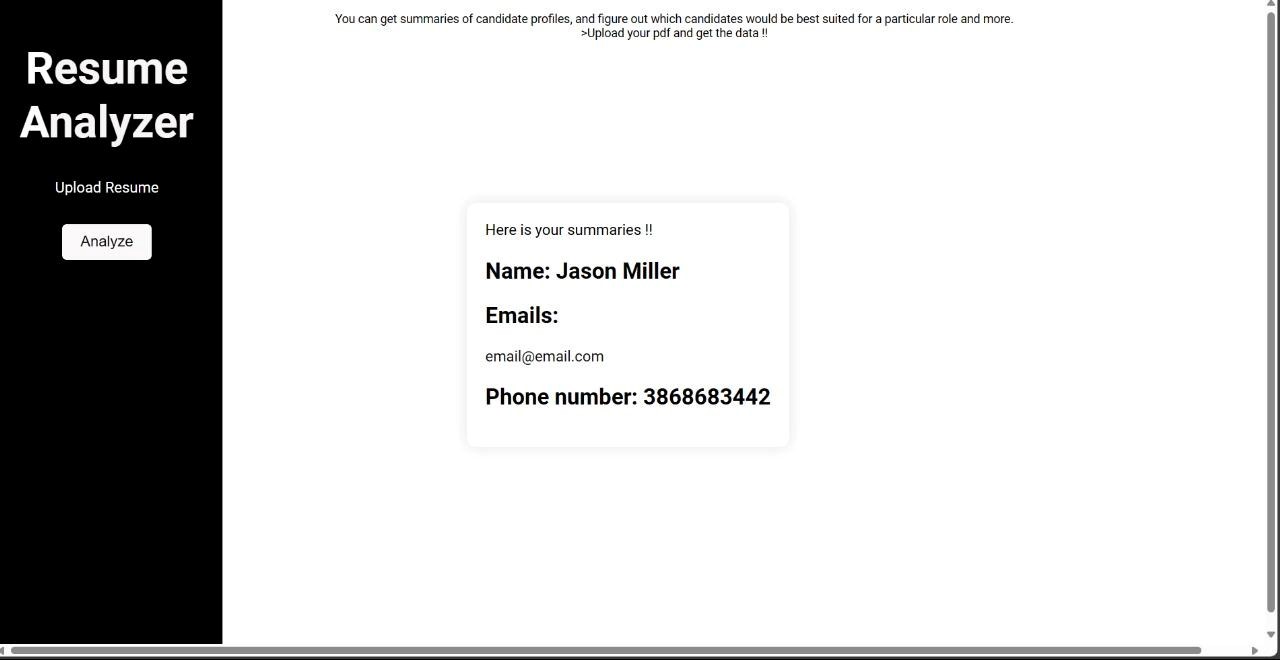
**Convert the parsed information into a structured format such as JSON or a database schema:**

* **Store the data in a database if needed, or use it directly in your application.**

**. Error Handling and Improvements. Implement robust error handling to manage different resume formats and inconsistencies in data**

**### Summary**

* **\*\*Define the information\*\* you need.**
* **\*\*Choose libraries\*\* for handling PDFs, DOCX, and text files.**
* **\*\*Extract text\*\* from resumes using these libraries.**
* **\*\*Parse the text\*\* using an NLP library to identify key details.**
* **\*\*Refine and validate\*\* the extracted data.**
* **\*\*Convert and store\*\* the data in a structured format.**
* **\*\*Improve the parser\*\* continuously based on real-world data and feedback**



## SAMPLE CODING

**import spacy import docx import fitz # PyMuPDF**

**# Function to extract text from DOCX**

**def extract\_text\_from\_docx(file\_path): doc = docx.Document(file\_path) return ' '.join([para.text for para in doc.paragraphs])**

**# Function to extract text from PDF**

**def extract\_text\_from\_pdf(file\_path):**

**doc = fitz.open(file\_path)**

**text = "" for page in doc:**

**text += page.get\_text() return text**

**# Load spaCy NLP model**

**nlp = spacy.load("en\_core\_web\_sm")**

**# Function to parse resume text**

**def parse\_resume(text):**

**doc = nlp(text) entities = {ent.label\_: ent.text for ent in doc.ents} return entities**

**# Main function to process a resume file**

**def process\_resume(file\_path, file\_type):**

**if file\_type == 'pdf': resume\_text = extract\_text\_from\_pdf(file\_path) elif file\_type == 'docx':**

**resume\_text = extract\_text\_from\_docx(file\_path)**

**else:**

**raise ValueError("Unsupported file type") parsed\_data = parse\_resume(resume\_text) return parsed\_data**

**# Example usage**

**file\_path = 'resume.pdf' file\_type = 'pdf' # Change to 'docx' for DOCX files**

**parsed\_data = process\_resume(file\_path, file\_type)**

**print(parsed\_data)**

## REFERENCES

Pandey, Abhishek, Anirudh Kaushik, Amit Kumar Jha, and Girish Kapse. "A technological survey on autonomous home cleaning robots." *International Journal of Scientific and Research Publications* 4, no. 4 (2014): 1-7.

Kaur, Manreet, and Preeti Abrol. "Design and development of floor cleaner robot (automatic and manual)." *International Journal of Computer Applications* 97, no. 19 (2014).

Khalid, Uman, Muhammad Faizan Baloch, Haseeb Haider, Muhammad Usman Sardar, Muhammad

Faisal Khan, Abdul Basit Zia, and Tahseen Amin Khan Qasuria. "Smart floor cleaning robot (CLEAR)." (2015).

Prabakaran, Veerajagadheswar, Mohan Rajesh Elara, Thejus Pathmakumar, and Shunsuke Nansai. "Floor cleaning robot with reconfigurable mechanism." *Automation in Construction* 91 (2018): 155-165.

Vishaal, Raj, P. Raghavan, R. Rajesh, Sachin Michael, and Mohan Rajesh Elara. "Design of dual purpose cleaning robot." *Procedia computer science* 133 (2018): 518-525.

Saleem, Adeel, Atif Iqbal, and Adnan Sabir. "Design and implementation of an intelligent dust cleaner robot for uneven and nonstructural environment." In *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, pp. 1-6. IEEE, 2019.

Yatmono, S., M. Khairudin, H. S. Pramono, and A. Asmara. "Development of intelligent floor cleaning robot." In *Journal of Physics: Conference Series*, vol. 1413, no. 1, p. 012014. IOP Publishing, 2019.

Parween, Rizuwana, Manuel Vega Heredia, Madan Mohan Rayguru, Raihan Enjikalayil Abdulkader, and Mohan Rajesh Elara. "Autonomous self-reconfigurable floor cleaning robot." *IEEE Access* 8 (2020): 114433-114442.

Irawan, Yuda, Muhardi Muhardi, Rian Ordila, and Roni Diandra. "Automatic floor cleaning robot using arduino and ultrasonic sensor." *Journal of Robotics and Control (JRC)* 2, no. 4 (2021): 240-243.

Patil, Virta Banduji. "Design For Dust Cleaning Robot Using Embedded System." In *2023 3rd International Conference on Smart Data Intelligence (ICSMDI)*, pp. 579-583. IEEE, 2023

Kachhela, Heena B., Ms Mamta Shyam Khadse, Sneha R. Bhange, Mr Tarun Prashant Ramteke, Ms Achal Ravindra Meshram, and Mr Manav Ravi Kamble. "DESIGN AND IMPLEMENTATION OF FLOOR CLEANING ROBOT USING IOT.".