An Intelligent System for Automated Resume Screening and Resume Building Using NLP and Machine Learning

Mukunda Rudrakar[1], Swakshan Tayade[2], Dhiraj Wankhede[3,] Umar Khan[4], Prof. Gajanan Gaikwad[5]
[1,2,3,4]B. E Student, Dept. of CSE, Prof Ram Meghe College of Engg & Management, Maharashtra, India.
[5]Assistant Professor, Dept. of CSE, Prof Ram Meghe College of Engg & Management, Maharashtra, India.

***Abstract*—AI Resume Analyzer is an artificial intelligence-based system that analyzes candidates' CVs and makes recommendations to improve them. The system takes the candidate's resume as input and processes it using machine learning algorithms to identify opportunities for improvement. The main goal of the system is to help job seekers create resumes that will successfully pass the interview selection process. To extract useful information from a candidate's resume, the system uses various approaches such as natural language processing, text mining and sentiment analysis. Review the candidate's experience, skills, education, and achievements to get feedback to improve their resume. The algorithm also evaluates your resume's layout, language, and spelling to determine whether it meets industry standards. job seekers and recruiters can use the intelligent resume parser. Recruiters can use the system to quickly find the best candidates for a specific position, and job seekers can use it to create resumes that are more likely to pass the selection process at interviews. Job seekers can increase their chances of getting hired in a selected position by using a smart resume analyzer.**

***Index Terms*—Natural Language Processing (NLP), Resume Parser, Resume Analysis, Part-of-speech tagging, Named Entity Recognition (NER).**

I. INTRODUCTION

A resume is a formal document that highlights a candidate's professional background, skills, and experience**,** primarily used for job applications. Typically, it includes sections such as a professional summary, work history, and education**,** serving as a marketing tool for job seekers. The main objective of a resume is to introduce the candidate to potential employers, showcase qualifications, and secure an interview**.**

Resume analysis is a multi-step process that involves both automated and manual reviews to extract relevant applicant details efficiently. Large organizations often receive thousands of resumes each month and use resume management software to streamline the screening process. These systems scan resumes for keywords, work history, education, and experience**,** ranking candidates based on job criteria. As a result, resumes that do not meet specific requirements may never reach a human reviewer.

After automated screening, manual review is conducted to ensure the authenticity of the candidates, as some resumes may be professionally crafted to enhance a candidate's profile. Various techniques are employed to identify the most suitable candidates, ensuring that only the best applicants move forward in the hiring process.

II. LITERATURE REVIEW

The increasing demand for efficient and effective recruitment processes has led to significant research and development in the field of automated resume analysis. This literature review examines key studies and technological advancements that have contributed to the development of AI-based resume analyzers. The literature indicates significant progress in the development of AI-based resume analyzers. From early rule-based systems to sophisticated NLP and machine learning models, the evolution of resume parsing and candidate evaluation techniques has improved the efficiency and accuracy of recruitment processes.

2.1 Automated Resume Parsing: Resume parsing, the process of converting unstructured resume data into structured information, is a critical component of AI-based resume analyzers. Early work in this field

focused on keyword matching and rule-based systems. Grosky et al. (1997) pioneered methods for extracting personal information, skills, and work experience using pattern matching and heuristic rules. However, these systems were limited by their reliance on predefined patterns and struggled with variations in resume formats.

2.2 Natural Language Processing: (NLP) in Resume Analysis With advancements in NLP, more sophisticated techniques have emerged. Sarwar et al. (2013) utilized NLP to enhance resume parsing accuracy. Their system employed part-of-speech tagging and named entity recognition (NER) to identify key information in resumes. The study highlighted the importance of understanding the context and semantics of text, moving beyond simple keyword matching. Recent advancements in NLP, particularly with the advent of deep learning models like BERT (Bidirectional Encoder Representations from Transformers), have further improved resume parsing. Devlin et al. (2018) demonstrated that BERT, pre-trained on large text corpora, could be fine-tuned for specific tasks such as resume parsing, significantly improving accuracy. Liu et al. (2019) applied BERT to resume data and achieved state-of-the-art results in extracting entities like skills and job titles.

2.3 Machine Learning for Candidate Evaluation: Machine learning algorithms have been employed to evaluate and rank candidates based on resume data. Zhao et al. (2015) developed a machine learning model that scored resumes based on predefined criteria such as education, experience, and skills. Their system used a combination of supervised learning techniques, including support vector machines (SVM) and decision trees, to predict the suitability of candidates for specific job roles. Huang et al. (2017) introduced a deep learning-based approach for resume evaluation. They proposed a neural network model that learned to rank resumes by comparing them to job descriptions. Their experiments showed that deep learning models outperformed traditional machine learning algorithms in understanding the nuances of resume data and job requirements.

2.4 User Interface and Experience: The importance of a user-friendly interface in recruitment technology cannot be overstated. Johnson et al. (2014) emphasized the need for intuitive and interactive user interfaces to enhance the user experience for both recruiters and job seekers. Their study highlighted the benefits of visualizing resume data and providing actionable insights through easy-to-use dashboards.

2.5 Security and Data Privacy: With the increasing use of AI in handling sensitive personal data, security and data privacy have become paramount. Rastogi et al. (2018) reviewed security measures in AI systems, stressing the importance of encryption, secure authentication, and compliance with data privacy regulations like GDPR. Their work provides a comprehensive overview of best practices for ensuring the security and privacy of user data in AI applications.

2.6 Summary: User interface design, security, and scalability remain critical areas for ensuring the practical deployment of these systems. The Resume Analyzer and Recommender System Using Python project builds on these advancements, aiming to provide a comprehensive solution that automates resume evaluation, enhances accuracy, ensures user-friendly interaction, and maintains robust security and scalability.

III. RESEARCH OBJECTIVE

The primary objective of this study is to analyze applicant resumes using the resume parser method along with various computational algorithms. This tool focuses on the resume screening process, determining whether a candidate is qualified for a specific job role based on education, skills, experience, and other details present in their resume.

The foundation of a resume analyzer is to help applicants create a well-structured, error-free, and job-specific resume. The system utilizes AI-driven techniques to scan resumes and compare them against job descriptions, industry standards, and best practices.

AI Resume Analyzer Capabilities:

- Grammar & Typo Detection – Identifies spelling mistakes and grammatical errors.
- ATS-Friendly Formatting – Ensures compatibility with Applicant Tracking Systems (ATS).
- Video Recommendations – Suggests relevant videos for resume writing and interview preparation.

- Keyword Recognition – Extracts and highlights key industry-specific terms.
- Professional Layout – Enhances resume presentation for better readability.
- Impactful Language – Encourages result-driven phrasing to make resumes more compelling.
- Resume Completeness Check – Ensures all necessary sections are included for a strong resume.

AI Resume Builder Integration: In addition to resume analysis, the system includes an AI Resume Builder that helps users generate a professional resume from scratch. The AI Resume Builder guides candidates in structuring their resumes effectively, ensuring they meet industry expectations.

AI Resume Builder Features:

- Template Selection – Offers various professional resume templates.
- Personalized Sections – Allows users to input experience, education, skills, and achievements.
- Auto-Formatting – Ensures a clean and well-organized resume layout.
- Keyword Optimization – Integrates job-specific keywords for better visibility.
- Downloadable Resume – Generates a final resume in PDF or DOCX format for easy sharing.

By integrating both AI Resume Analyzer and AI Resume Builder, the system provides a comprehensive solution for job seekers, ensuring they create resumes that are well-structured, error-free, and tailored to their target job roles.

IV. METHODOLOGY

Various techniques and algorithms exist to address NLP-based challenges. Since these problems rely on deep learning concepts, Python is the preferred programming language. Python provides several libraries, such as nltk and spacy, which facilitate text extraction and information retrieval from documents. For text cleaning, the re (Regular Expression) library is utilized. nltk and spacy play a crucial role in NLP tasks, including removing stop words, stemming, part-of-speech (POS) tagging, and named entity recognition (NER).

Text preprocessing is a critical and complex phase in any NLP project, serving as the foundation for accurate analysis.

The preprocessing stage includes:

- Removing stop words
- Converting text to lowercase
- Tokenization
- Lemmatization

Parsing Steps:

a] Extracting text from PDF files.
b] Extracting text from DOC files.
c] Identifying file extensions.
d] Extracting named entities.
e] Extracting email addresses.
f] Identifying candidate names from resumes.
g] Extracting mobile numbers.
h] Identifying skills listed in the resume.
i] Recognizing technical skills.
j] Extracting education details along with graduation year.
k] Identifying work experience.

Once the parsing process is completed, the extracted data undergoes analysis and is stored in a database. This data assists recruiters in visualization and decision-making while also providing personalized job recommendations to job seekers.

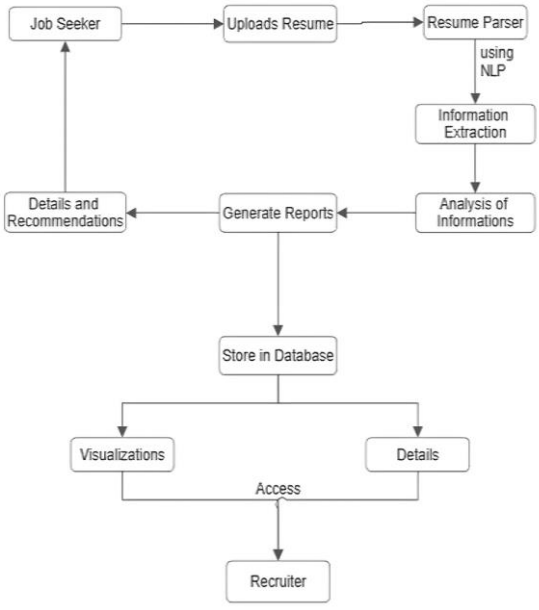

Fig. System Architecture

4.1 Models & Libraries Used:

1. NLP: Natural Language Processing (NLP) is a field of computer science and artificial intelligence (AI) that focuses on enabling computers to understand, interpret, and generate human language. It involves developing algorithms and models that can process and analyze natural language data, such as text and speech, in order to perform various tasks, including sentiment analysis, machine translation, text summarization, and chatbot development, among others. NLP combines techniques from computer science, linguistics, and mathematics to enable computers to "understand" human language and communicate with humans in a more natural way.

2. Sklearn: Scikit-learn (sklearn) is a popular open-source Python library that provides a range of tools for machine learning and data analysis tasks. It includes a variety of machine learning algorithms, such as classification, regression, clustering, and dimensionality reduction, as well as tools for preprocessing, model selection, and evaluation. Scikit-learn is designed to be user-friendly and efficient, making it a valuable tool for both beginners and experienced practitioners. It also offers integration with other Python libraries, such as NumPy, Pandas, and Matplotlib, for data manipulation and visualization. Scikit-learn is widely used in industry and academia for various applications, such as predictive modeling, image recognition, and natural language processing.

3. NumPy: NumPy is a Python library for scientific computing and numerical analysis that provides a powerful array data structure, which enables efficient manipulation of large multi-dimensional arrays and matrices. NumPy also provides a variety of mathematical functions and tools for working with these arrays, such as linear algebra operations, Fourier transforms, and random number generation. NumPy arrays are homogeneous, meaning all elements of the array must be of the same data type. This allows for fast, memory efficient operations and makes NumPy a popular choice for data analysis, machine learning, and scientific computing tasks. In addition, NumPy is also used as a foundation for many other scientific Python libraries, such as Pandas and Matplotlib. Overall, NumPy is a fundamental library in the Python data science ecosystem and is widely used in academia and industry.

4. Lightcast API: The Lightcast API is an AI-powered tool that provides skill recommendations by analyzing industry trends and job market demands. It enhances resume analysis by identifying missing or relevant skills, ensuring candidates align their resumes with current hiring requirements. By comparing an applicant's skill set with real-world job postings, the API suggests in-demand skills that improve job prospects. This integration helps job seekers optimize their resumes, making them more competitive and better suited for career growth opportunities.

Libraries Used:

1. Streamlit – A web application framework used to create an interactive user interface. It enables users to upload resumes, view analysis results, and provides an admin dashboard for managing resume data.
2. Pandas – A data processing library used to store, manage, and analyze resume data. It helps in organizing extracted information and generating structured reports.
3. PyResParser – A resume parsing library that extracts key details such as name, email, phone number, skills, and experience from resumes using NLP techniques.
4. PDFMiner3 – A PDF text extraction library that converts resume files into plain text. This extracted text is further processed to identify useful information.
5. NLTK – A natural language processing (NLP) library used for text preprocessing, such as removing stopwords. It helps refine extracted text for accurate analysis.
6. SpaCy – An advanced NLP library used for named entity recognition (NER) and keyword extraction. It helps identify job roles, skills, and structured data in resumes.
7. PyMySQL – A database connectivity library that enables storing and retrieving resume analysis data in an SQL database. It ensures efficient management of user information.
8. Plotly – A data visualization library used to create interactive charts and graphs. It helps display insights such as skill trends and user experience distribution.
9. YouTube-DLP – A video extraction library that fetches relevant YouTube videos. It is used to

suggest resume-building and interview preparation resources to users.

10. Resume Builder – A custom-built module that allows users to create professional resumes from templates. It structures user input into a polished document format.

4.2 Frontend Development

The frontend of the Resume Analyzer and Recommender System Using Python is built using Streamlit, an open-source app framework designed for creating data-driven web applications with Python. Streamlit provides a simple yet powerful way to build interactive user interfaces.

4.2.1 Design Principles: The design of the frontend is focused on usability and accessibility, ensuring that both recruiters and job seekers can easily interact with the system. Key design principles include:

a) Intuitive Navigation: Ensuring that users can easily navigate through different sections of the application.

b) Clear Visualization: Providing clear and meaningful visual representations of analyzed data.

c) Responsiveness: Ensuring the application performs well on various devices and screen sizes.

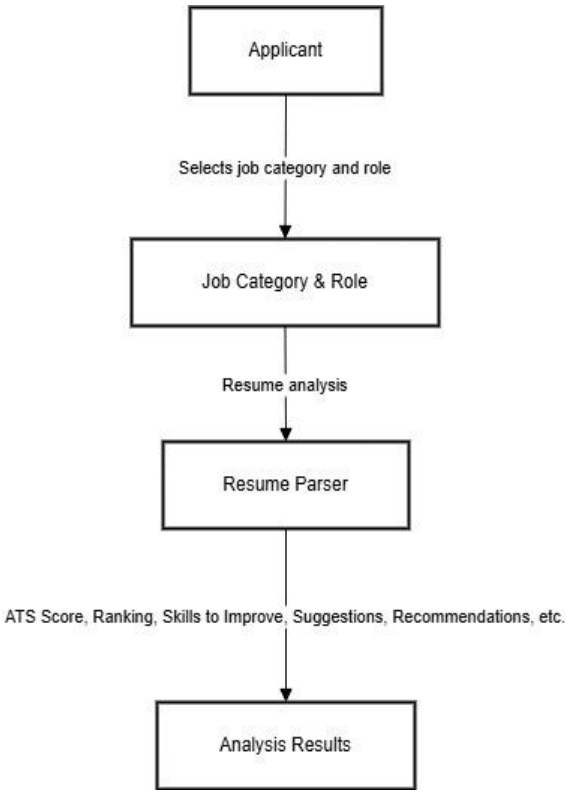

Fig. Overview of system

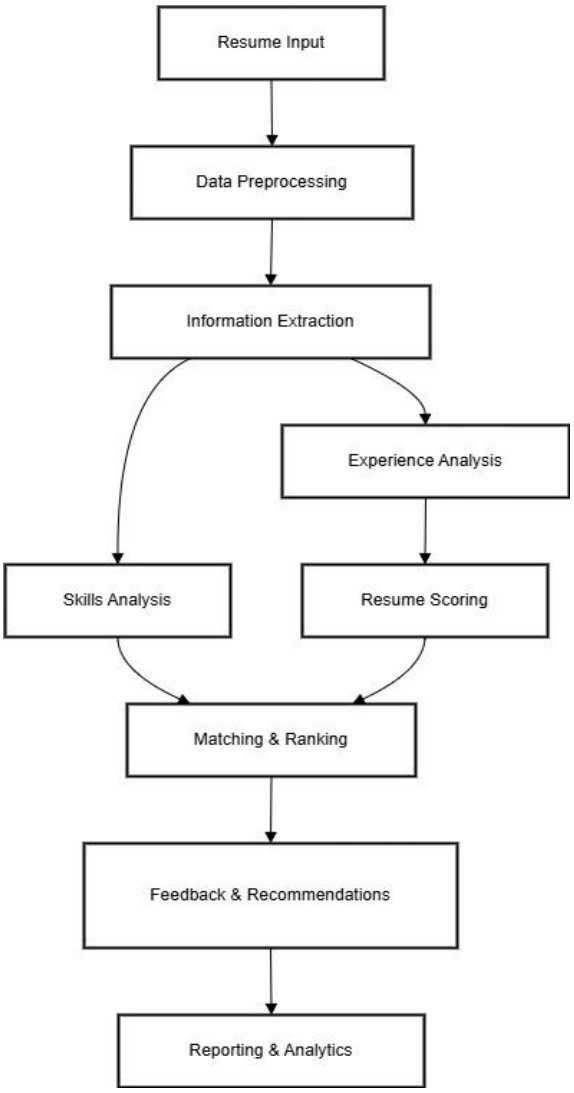

Fig. Workflow of the system

4.2.2 Implementation: The implementation involves creating several interactive components using Streamlit:

a) Upload Interface: Allows users to upload resumes in various formats (PDF, DOCX).

b) Criteria Configuration: Enables recruiters to set evaluation criteria, such as key skills, experience levels, and educational qualifications.

c) Results Display: Visualizes the parsed and analyzed data, highlighting key information and scoring candidates based on the set criteria.

4.3 Backend Development

The backend is the core of the Resume Analyzer and Recommender System Using Python, responsible for parsing resumes, extracting relevant information, and analyzing the data based on predefined criteria.

Python and the Pyresparser library are used extensively for these tasks.

4.3.1 Resume Parsing: The Pyresparser library is utilized to parse resumes and extract structured information from unstructured text. The process involves:

a) Tokenization: Breaking down the resume text into individual tokens (words, phrases).
b) Part-of-Speech Tagging: Identifying the grammatical structure of the text to understand the role of each token.
c) Named Entity Recognition (NER): Extracting entities such as names, dates, and job titles.
d) Custom Extraction Rules: Defining rules for extracting specific information relevant to the job criteria.

4.3.2 Data Analysis: Once the data is extracted, it is analyzed to evaluate the suitability of candidates. This involves: a. Criteria Matching: Comparing extracted information with the predefined criteria set by recruiters. b. Scoring Algorithms: Assigning scores to resumes based on the degree of match with the criteria. c. Ranking: Sorting candidates based on their scores to highlight the most suitable candidates.

4.4 Database Management

MySQL is used for storing parsed resume data securely and efficiently. The database schema is designed to handle various components of a resume, including personal information, skills, work experience, and education.

4.4.1 Database Schema Design: The database schema is designed to ensure efficient storage and retrieval of data. Key tables include:

a) Users: Stores user information (recruiters and job seekers).
b) Resumes: Stores raw resume data and parsed information.
c) Criteria: Stores evaluation criteria set by recruiters.
d) Results: Stores analysis results and scores for each resume.

4.4.2 Implementation: The implementation involves:

a) Database Setup: Installing MySQL, configuring the server, and creating the necessary databases and tables.
b) Data Insertion: Developing scripts to insert parsed data into the database.
c) Query Optimization: Ensuring queries are optimized for fast retrieval and processing of data.

4.5 Dataflow

The dataflow within the Resume Analyzer and Recommender System Using Python system is carefully designed to ensure seamless interaction between the frontend, backend, and database.

4.5.1 Upload and Parse

- Step 1: User uploads a resume through the Streamlit interface.
- Step 2: The resume file is sent to the backend for parsing.
- Step 3: Pyresparser processes the resume and extracts relevant information.

4.5.2 Analyze and Store

- Step 4: Extracted data is analyzed based on predefined criteria.
- Step 5: Analysis results and scores are stored in the MySQL database.

4.5.3 Retrieve and Display

- Step 6: The frontend retrieves analysis results from the database.
- Step 7: Results are visualized on the Streamlit interface for the user to review.

4.6 Security Measures

Security is a paramount concern, given the sensitive nature of the data being handled. Various measures are implemented to ensure data security and privacy.

4.6.1 Data Encryption :

- In-Transit Encryption: Using HTTPS to encrypt data transmitted between the user and the server.
- At-Rest Encryption: Encrypting sensitive data stored in the database.

4.6.2 Secure Authentication:

- User Authentication: Implementing secure login mechanisms, including password hashing and multi-factor authentication (MFA).
- Role-Based Access Control: Ensuring that different user roles (recruiters, job seekers) have appropriate access levels.

4.6.3 Data Privacy Compliance:

- GDPR Compliance: Implementing features such as data anonymization and user consent management to comply with data privacy regulations.

4.7 Server Setup

Setting up a reliable server environment is crucial for hosting the Resume Analyzer and Recommender

System Using Python application and ensuring its availability and performance.

4.7.1 Server Environment:

a) Choice of Server: Selecting a cloud-based solution (e.g., AWS, Google Cloud, Azure) for scalability and reliability.
b) Configuration: Setting up the server with necessary software packages (Python, MySQL, web server).

4.7.2 Installation of Dependencies:

a) Python and Libraries: Installing Python and required libraries (Streamlit, Pyresparser, MySQL connector).
b) Web Server: Configuring Nginx or Apache to serve the Streamlit application.

4.8 Database Deployment

Deploying the MySQL database involves setting up the database server, configuring security settings, and creating the database schema.

4.8.1 Setup

a) Installation: Installing MySQL on the server.
b) Configuration: Configuring MySQL for secure access and optimal performance.

4.8.2 Schema Creation

a) Tables: Creating tables for users, resumes, criteria, and results.
b) Indexes: Implementing indexes on key columns for faster query performance.

4.9 Application Deployment

Deploying the Streamlit application involves transferring application files to the server, configuring the web server, and setting up domain and SSL certificates.

4.9.1 Local Development

a) Development: Building and testing the application locally.
b) Version Control: Using Git for version control and collaboration.

4.9.2 Server Deployment

a) Transfer: Transferring application files to the server.
b) Configuration: Configuring Nginx/Apache to serve the application, setting up SSL certificates for HTTPS.

4.9.3 Load Balancing

Implementation: Setting up load balancers to distribute incoming traffic across multiple instances of the application, ensuring reliability and performance.

V. RESULTS AND DISCUSSION

The Intelligent Resume Analyzer system relies heavily on modeling and analysis. Technological models and analyzes resumes using powerful natural language processing and machine learning techniques and delivers results valuable insights for job seekers to improve their resume.

Step 1**:** Applicant needs to choose from two options i.e Resume Builder & Resume Analyzer. (Fig 1)

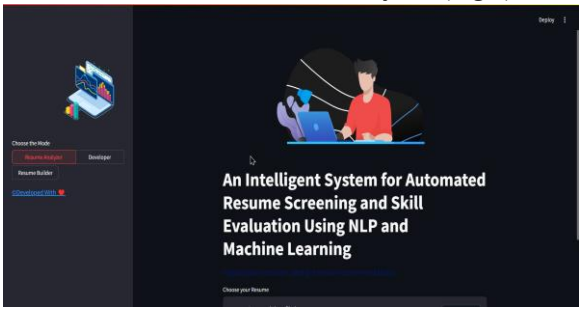

Fig 1. Frontend Interface

Step2: Applicant needs to upload his/her resume to the portal for the analysis. (Fig 2)

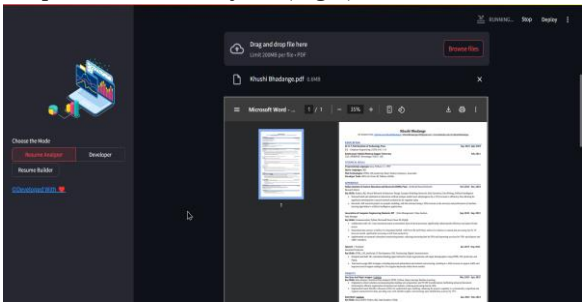

Fig 2. Resume Upload Interface

Step 3: Resume parser will extract basic information from applicant's resume. Fig. 3 shows result of current step.

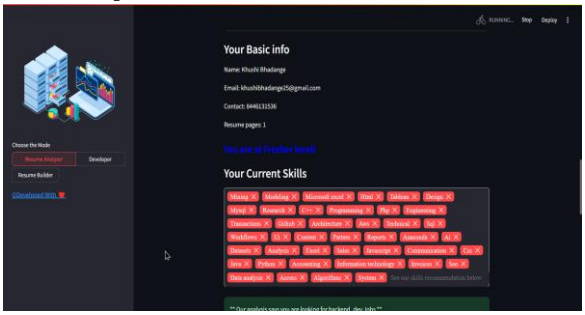

Fig 3. Parsed Resume Data Extraction

Step 4: Fig 4,5 & 6 shows detailed analysis & pictorial representation of applicant's resume after parsing.

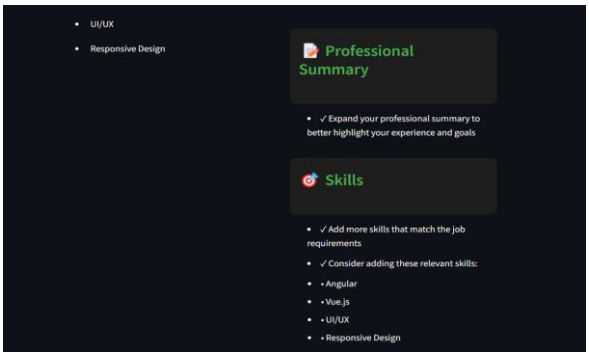

Fig 4. Summary of Uploaded Resume

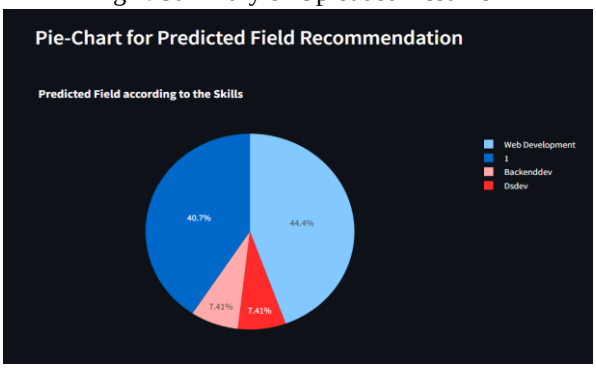

Fig 5. Pie Chart for Predicted Field Recommendation

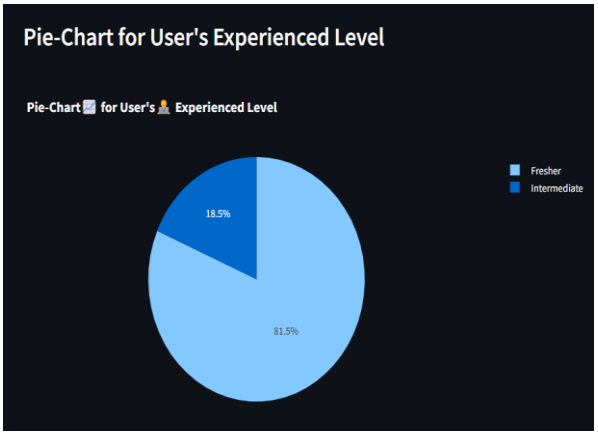

Fig 6. Pie Chart for User's Experienced Level

Step 5: Fig 7 & 8 shows the recommendations & suggests helpful videos for improvement.

Fig 7. AI Powered Recommendations

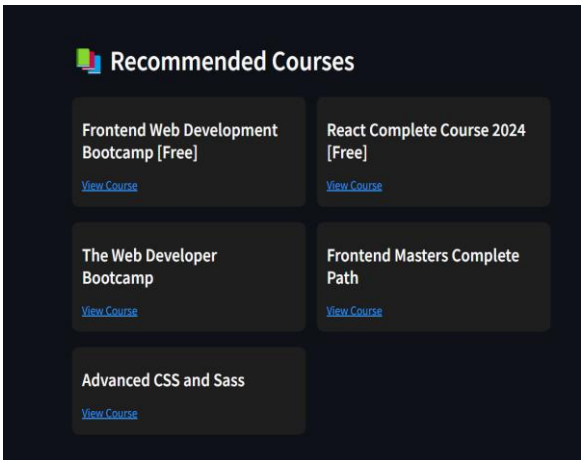

Fig 8. AI Powered Recommendations

For Resume Builder:

Step 1: Applicant needs to choose a template from drop down menu and fill the remaining details.

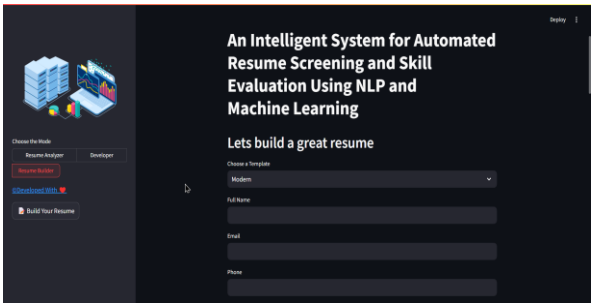

Fig 9. Resume Builder Template Selection Interface

Step 2: After this, the system generates a resume (downloadable) based on the information you have provided.

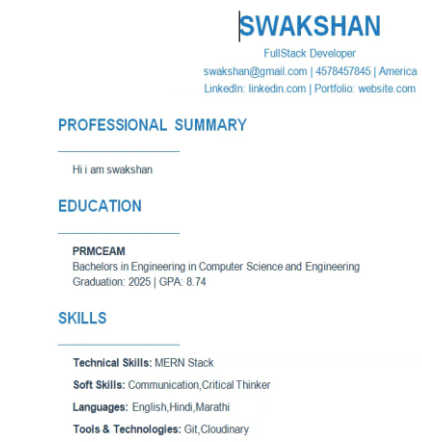

Fig 10. Generated Resume (Downloadable Format)

Admin Side:

- Admin can login to check the user data (uploaded resumes) [Fig 13].

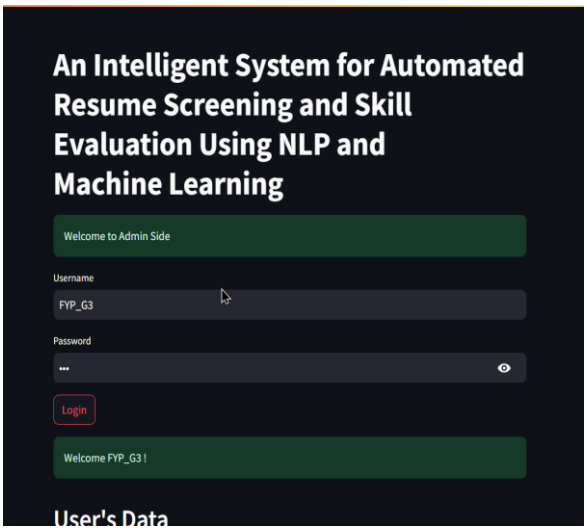

Fig 11. Admin Panel for Managing user data

- Admin can download the user data report if needed. (Fig 12)

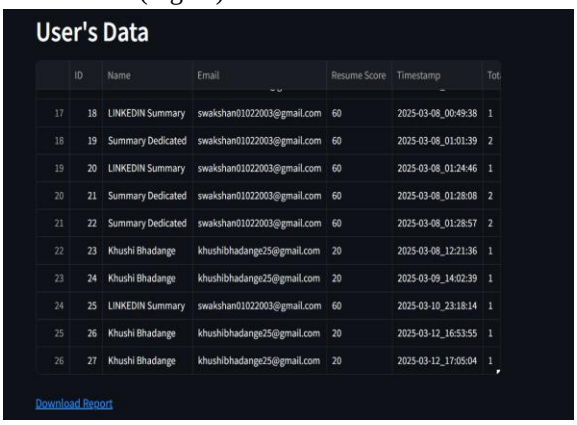

Fig 12. Resume Data History Interface

VI. FUTURE WORK

In the future, we can enhance our model by providing a more comprehensive analysis that includes an evaluation of a candidate's facial expressions and tonal quality of voice to assess their level of confidence. Video content analysis (VCA) or video analytics (VA) is a technique that automatically analyzes video to detect temporal and spatial events. For degree verification, we cannot rely solely on ground-level checks. Therefore, we plan to use third-party verification services like DigiLocker, which is an initiative under the Digital India program aimed at providing citizens with authentic digital documents. We aim to develop a module within our system that allows users to assess their resume skills and receive an evaluation after submitting their CV. Our future plans also include offering better pay scales and commission opportunities for our users, as well as implementing a detailed employee evaluation system to improve the efficiency of the hiring process for employers and clients seeking to hire freelance employees.

VII. CONCLUSION

The Intelligent Resume Analyzer system leverages machine learning and artificial intelligence to streamline the resume evaluation process, offering job seekers valuable recommendations for improvement. By incorporating advanced natural language processing (NLP) techniques, the system provides personalized feedback that enhances the quality of resumes, ultimately increasing candidates' chances of securing job opportunities.

The methodology of the system encompasses study design, data collection, and analytical approaches, ensuring a robust and efficient framework for resume parsing and analysis. The Intelligent Resume Analyzer has the potential to revolutionize the job application process by enabling job seekers to create well-structured, compelling resumes that align with employer expectations.

As automation continues to expand across industries, the need for intelligent recruitment solutions is more evident than ever. Our system addresses this demand by providing an efficient, accurate, and scalable solution for resume parsing and analysis. In an era where thousands of candidates apply for the same role, such systems play a crucial role in optimizing hiring efficiency and ensuring a fair, data-driven selection process.

REFERENCES

[1] Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit" by Steven Bird, Ewan Klein, and Edward Loper.

[2] Ronan Collobert, Jason Weston L´eon Bottou, Michael Karlen, Koray Kavukcuoglu, Pavel Kuksa “Natural Language Processing (almost) from Scratch” arxiv.

[3] Deepak, Gerard, Varun Teja, and A. Santhanavijayan. "A novel firefly driven scheme for resume parsing and matching based on entity

linking paradigm." Journal of Discrete Mathematical Sciences and Cryptography 23.1 (2020): 157- 165.

[4] A Review Paper on Resume scanning using python, Panil Jain, Nandkishor Kamble, Risha Nadar, Shaikh Luqman, International Research Journal of Engineering and Technology (IRJET), Volume: 09 Issue: 04 — Apr 2022.

[5] Resume Recommendation using Machine Learning, Ramba S Naik, Shrinivas R Dhotre, International Journal of Creative Research Thoughts (IJCRT), Volume 10, Issue 7, July 2022.

[6] Resume Analyzer and Recommender System Using Python Pratik G. Raut 1, Prof. Rajesh D. Wagh International Journal of Research Publication and Reviews, Vol 5, no 6, pp 6245-6253 June 2024

[7] RESUME PARSER AND ANALYZER USING NLP Nimish Patil*1, Shubham Yadav*2, Vikas Biradar*3 IRJETS Volume:05/Issue:04/April-2023