iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** III **Month of publication:** March 2026

DOI: https://doi.org/10.22214/ijraset.2026.77954

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

Automated ATS-Optimized Resume Builder with Dynamic Templates

Yelubolu Surya Chakri[1], Telugu Vinay[2], Darapureddy Yeswanth Surya Rama Mahesh[3], Polamuri Lukeshbabu[4], Donga Naga Venkata Sri Surya[5], Onteddu Siva Rama Krishna[6]

[1, 2, 3, 4, 5]*Department of Computer Science and Engineering (Artificial Intelligence and Data Science), Bonam Venkata Chalamayya Engineering College, Affiliated to JNTU Kakinada, Andhra Pradesh, India*

[6]*Project Guide, Department of Computer Science and Engineering (Artificial Intelligence and Data Science), Bonam Venkata Chalamayya Engineering College, Affiliated to JNTU Kakinada, Andhra Pradesh, India*

Abstract: ***The recruitment sector relies heavily on Applicant Tracking Systems (ATS) to manage the sheer volume of daily job applications. While these software platforms streamline operations for hiring managers, they unintentionally erect a massive barrier for qualified professionals. Industry observations indicate that a substantial majority of resumes are filtered out before human review, often due to structural formatting errors, unreadable layouts, or keyword mismatches rather than actual skill deficits. Despite this reality, many candidates continue to rely on traditional word processors, leading to customized documents that ultimately fail algorithmic parsing. This paper introduces an Automated ATS-Optimized Resume Builder, a dedicated web application designed to bridge the gap between human readability and machine parsing. Architected with a strict focus on the job seeker, the platform bypasses complex administrative layers to offer a direct, interactive user environment. Candidates can select dynamic templates and input their professional history into an editor that provides a real-time visual preview alongside their text. The backend infrastructure is powered by the Python Flask framework and a PostgreSQL database for secure data management. To evaluate document content, the system avoids computationally expensive machine learning models in favor of a fast, deterministic Natural Language Processing (NLP) pipeline. By applying text tokenization, regular expressions, and predefined keyword dictionaries, the application instantly calculates an ATS compatibility score and pushes actionable, section-by-section recommendations to the user. Once the data is finalized, the platform leverages the python-docx library to dynamically generate and export structurally rigid, ATS-compliant documents in both PDF and DOCX formats.***

Testing demonstrates that this lightweight, rule-based approach significantly improves formatting compliance and keyword density. The results indicate that the system provides job seekers with a highly accessible, efficient, and reliable tool to successfully navigate modern algorithmic hiring filters. Future scope includes the potential integration of automated job-board scraping and expanded template libraries.

Index Terms: ***Applicant Tracking System (ATS), Flask Web Architecture, Resume Parsing, Tokenization, Heuristic Scoring, Dynamic Templating, Automated Document Generation.***

I. INTRODUCTION

Securing employment in the modern job market heavily depends on passing software filters long before a human hiring manager ever looks at an application. Applicant Tracking Systems (ATS) now act as the primary gatekeepers in the talent acquisition pipeline. The financial footprint of these platforms reflects their dominance, with the global ATS software market valued at roughly USD 2.41 billion in 2023 and steadily expanding. Because recruiters are routinely overwhelmed by hundreds of applications for a single open role, relying on software to parse and rank candidates has become a standard corporate necessity. However, this heavy reliance on automation creates a frustrating and largely invisible barrier for job seekers.

A. Background and Motivation

When we look closely at the current recruitment ecosystem, a clear disconnect exists between how candidates present their professional histories and how machines are programmed to read them. Many applicants naturally want to stand out, so they use traditional word processors to build visually creative documents. Unfortunately, elements like multi-column layouts, embedded graphics, and unconventional fonts frequently cause parsing failures when processed by an ATS. Even highly qualified professionals routinely find their applications ignored simply because their resumes lack specific industry keywords or fail to match a rigid

structural format. Studies and industry observations routinely highlight high rejection rates during this initial automated screening phase. While awareness of ATS filters is slowly growing, the average job seeker still struggles to manually optimize their documents to meet these strict machine-reading standards. They require accessible digital tools that actively guide them to align their wording and layout with algorithmic expectations.

B. Problem Statement

Despite the obvious need for ATS-friendly resumes, a significant portion of the existing resume-building platforms fail to address the core mechanical issues. Many online builders still prioritize flashy aesthetics that actively harm a candidate's chances of passing a text-extraction scan. On the other end of the spectrum, platforms that do prioritize ATS compliance often operate as opaque black boxes. They frequently rely on heavy, computationally expensive deep learning models that generate a score but fail to explain the clear, mechanical reasoning behind it. Furthermore, these platforms often lack a seamless editing environment. Users cannot immediately see how tweaking a single sentence impacts their document's layout or their overall compatibility score. There is a noticeable void for a lightweight, transparent web application that uses clear, rule-based logic to evaluate text and instantly generate structurally sound documents without the overhead of complex artificial intelligence.

C. Objectives

The primary aim of this research is to design, develop, and evaluate an accessible platform that directly tackles the resume parsing problem. The specific objectives of this project are:

1) To develop a streamlined, user-centric web application using a Python Flask backend and a PostgreSQL database. This architecture deliberately bypasses complex admin- istrative dashboards to focus entirely on the candidate's immediate needs.
2) To implement a deterministic, rule-based Natural Lan- guage Processing (NLP) pipeline. Instead of relying on unpredictable machine learning, this system utilizes text tokenization, regular expressions, and targeted keyword matching to calculate a reliable ATS compatibility score.
3) To engineer an interactive editor equipped with a live- preview interface, allowing users to draft their profiles while receiving instant heuristic feedback and structural updates.
4) To integrate a robust document export engine using the python-docx library, ensuring the final dynamic templates are strictly formatted and fully downloadable in ATS-compliant PDF and DOCX formats.

D. Contributions and Paper Organization

This paper provides a comprehensive software engineering blueprint for a functional, heuristic-based resume builder. The main contributions include the design of a lightweight text- evaluation algorithm that operates efficiently without advanced hardware, the integration of a real-time rendering interface, and the deployment of dynamic templates that guarantee optical character recognition (OCR) compliance. The remainder of this paper is structured as follows. Sec- tion II reviews the related literature concerning traditional screening limitations and rule-based text processing. Section III details the proposed system methodology, outlining the Flask architecture and the tokenization pipeline. Section IV discusses the system evaluation, focusing on user interface performance and document output fidelity. Section V presents a discussion of our design validations and constraints. Finally, Section VI concludes the study and offers directions for future enhancements.

II. LITERATURE SURVEY

The intersection of recruitment technology and automated document processing has been extensively studied over the past few years. As hiring practices shift entirely online, researchers have heavily scrutinized both the software that filters candidates and the digital tools designed to help them. A review of existing literature highlights several core challenges in modern recruitment and validates the foundational technologies required to build effective resume optimization platforms.

A. Traditional Applicant Tracking Systems and Screening Limitations

The rapid expansion of digital recruitment has forced organizations to rely heavily on Applicant Tracking Systems (ATS) to manage the sheer volume of incoming job applications. While these systems are incredibly effective at organizing candidate databases, researchers frequently point out their rigid operational limitations. A major flaw in traditional ATS architecture is its heavy dependence on exact-match string processing and basic lexical heuristics.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Studies indicate that if a candidate possesses the correct experience but uses slightly different terminology or synonyms not explicitly programmed into the ATS search query, their application is often automatically discarded. Furthermore, document formatting plays a disproportionate role in screening success. Literature analyzing ATS extraction algorithms shows that visually complex resumes—particularly those utilizing embedded graphics, multi-column layouts, or non-standard typography—frequently cause catastrophic parsing failures. Consequently, a significant percentage of highly qualified applicants are routinely filtered out not due to a lack of merit, but simply because their documents are structurally incompatible with legacy machine-reading standards.

B. Rule-Based Text Processing and Tokenization Techniques

To address the disconnect between human applicants and machine parsers, numerous studies have explored automated resume evaluation techniques. While recent academic trends often lean toward computationally expensive deep learning models, foundational Natural Language Processing (NLP) techniques remain highly effective and much more transparent. A well-documented, highly practical approach involves rule- based text processing.

Researchers have successfully demonstrated that breaking down unstructured resume text into smaller, manageable units—a process known as tokenization—allows systems to reliably isolate core competencies and experiences. When combined with regular expressions (Regex) and the systematic removal of stop-words, tokenization effectively cleans the raw data, stripping away irrelevant grammatical noise. Several aca- demic papers validate that comparing these cleaned text tokens against predefined skill dictionaries yields highly accurate compatibility scores. This deterministic, rule-based method provides a lightweight alternative to ”black box” machine learning models. It ensures that the scoring logic remains clear, predictable, and highly scalable without requiring extensive computational overhead.

C. Web-Based Document Generation and Dynamic Templating

Beyond just analyzing text, recent literature emphasizes the necessity of platforms that actively help candidates generate structurally compliant documents from scratch. Developing web-based tools using frameworks like Python’s Flask has become a standard industry approach due to its modularity and seamless integration with backend NLP libraries. However, a recurring engineering challenge discussed in the literature is the actual document rendering phase. Relying on basic browser-side printing functions often leads to inconsistent margins, broken tables, and hidden HTML artifacts, all of which are notorious for triggering ATS rejection protocols. To counter this, recent software architecture studies advocate for server-side document generation using program- matic libraries, such as python-docx. This approach enables the implementation of ”dynamic templates”—programmatic layouts that automatically adjust their structure based on the volume of the user’s input without breaking the underlying hierarchy. By hardcoding the document structure on the backend rather than the frontend view layer, these systems guarantee that the exported PDF or DOCX files maintain strict typographical simplicity and optical character recognition (OCR) compliance.

III. PROPOSED METHODOLOGY

The proposed application is engineered to provide a seamless, highly responsive environment for job seekers to draft, evaluate, and export their resumes. Unlike complex enterprise platforms that cater to both recruiters and candidates, this project intentionally strips away the administrative dashboard to focus entirely on a candidate-centric workflow.

A. System Architecture (Flask Backend and PostgreSQL Database Integration)

The platform is built on a robust, three-tier web architecture that separates the user interface from the underlying data processing logic.

1) Presentation Layer: The client-side interface handles user inputs and dynamic visual updates.
2) Application Layer: The server-side operations are pow- ered by the Python Flask framework (Flask==2.3.3). Flask acts as the central orchestrator, managing secure routing, handling API requests from the editor, and execut- ing the text-evaluation algorithms. The application server is deployed using Gunicorn (gunicorn==21.2.0) to ensure stable concurrent request handling.
3) Data Layer**:** Persistent data storage is managed by a PostgreSQL database (psycopg2-binary==2.9.9). To facilitate smooth communication between the Python backend and the database, the system utilizes SQLAlchemy (Flask-SQLAlchemy==3.0.5) as an Object-Relational Mapper (ORM). This prevents SQL injection vulnerabilities while organizing user profiles and resume drafts into highly structured relational tables.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

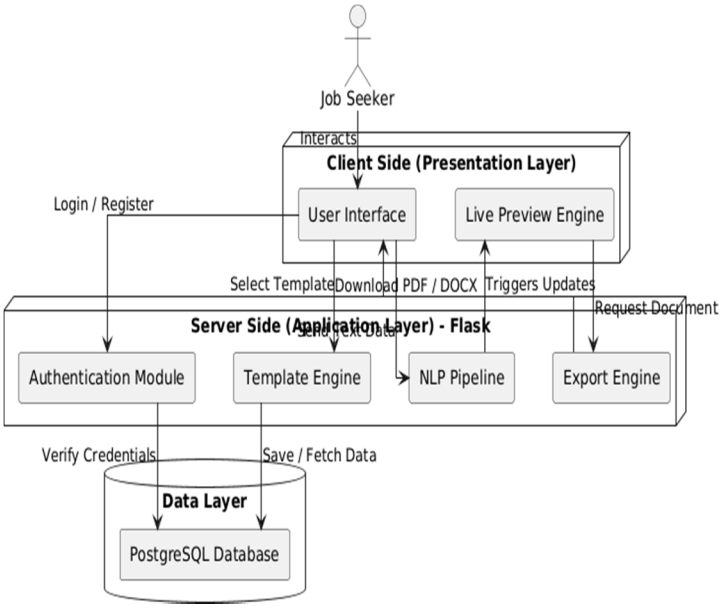


Fig. 1. System architecture of the proposed Flask-based ATS Resume Builder.

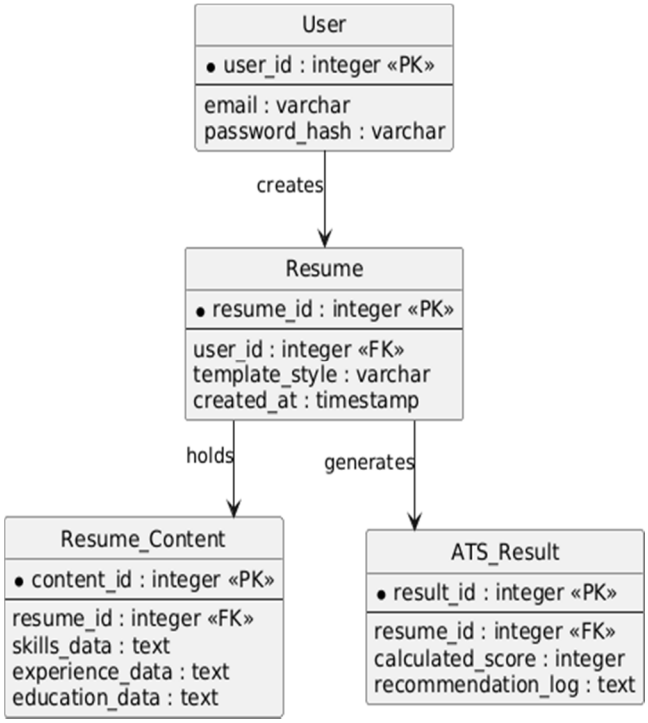


Fig. 2. Entity Relationship diagram detailing the database schema for users, resume content, and calculated ATS scores.

B. Functional Modules (User Authentication, Dashboard, and Template Selection)

To maintain an intuitive user journey, the application is divided into straightforward functional modules. The journey begins with the Authentication Module, which securely man- ages user registration and login sessions using Flask-Login and bcrypt for password hashing. Once authenticated, the user is directed to their personal Dashboard. This space serves as a dedicated repository where candidates can view, edit, or delete their previously saved resumes. When a user initiates a new document, they enter the Template Selection module. Here, they choose a baseline structural layout designed explicitly to meet standard parsing criteria—avoiding complex columns or unreadable graphics.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

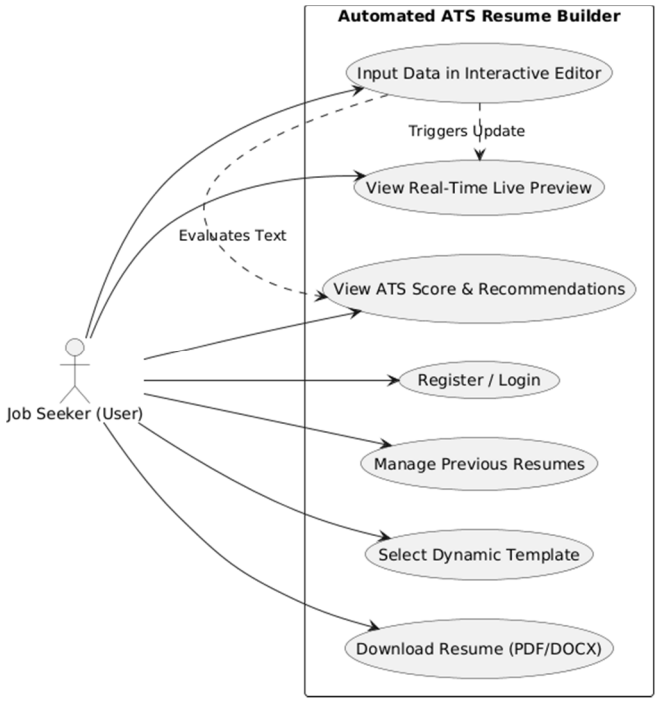


Fig. 3. Use case diagram illustrating the streamlined, candidate-focused system interactions.

C. Interactive Editor and Real-Time Resume Preview Render- ing

A major technical feature of the platform is the interactive dual-pane editor. Traditional resume builders often force users to fill out lengthy forms and navigate to a separate page just to view the result. To solve this, the proposed system implements a real-time rendering engine. As the candidate types their educational background, professional experience, and skills into the input fields on the left side of the screen, the data payload is instantly processed. The right side of the interface dynamically updates to reflect the compiled visual structure. This immediate feedback loop allows job seekers to see exactly how their phrasing impacts the document's spatial layout before they finalize the text.

D. Heuristic ATS Scoring and Keyword Matching Pipeline

Instead of relying on heavy, unpredictable machine learning models, the core analytical engine utilizes a highly efficient, deterministic Natural Language Processing (NLP) pipeline. When the user saves their content, the backend Python script executes a heuristic evaluation to calculate an ATS compatibility score. The algorithm evaluates the text across three weighted dimensions:

1) Completeness (40% weight): The system checks for the presence of essential data fields (e.g., contact information, professional summary, structured experience).

2) Text Quality (35% weight): The algorithm tokenizes the input strings and utilizes predefined arrays to search for high-impact action verbs (e.g., "achieved," "optimized," "led"). It also deploys regular expressions (re.findall) to count numerical figures, actively rewarding candidates who quantify their professional achievements.

3) Keyword Matching (25% weight): The text is scanned against comprehensive dictionaries of technical and soft skills.

Based on this mathematical evaluation, the system navigates the user to a summary page displaying their final score out of 100, alongside specific, AI-driven recommendations (e.g., "Expand your summary with more impact-driven metrics" or "Include more technical skills").

E. Document Export Engine (Dynamic PDF and DOCX Generation)

The final phase of the methodology involves converting the user's web-based data into a standalone, downloadable file. Relying on basic browser printing functions frequently corrupts document margins and breaks underlying text struc- tures. To guarantee structural rigidity, the backend utilizes the python-docx library. When a user requests a download, the server programmatically injects their parsed JSON data directly into an invisible, heavily formatted document template.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

This process ensures that the resulting PDF or DOCX file maintains a pristine typographical hierarchy, making it completely compliant with the optical character recognition (OCR) scanners used by legacy enterprise ATS platforms.

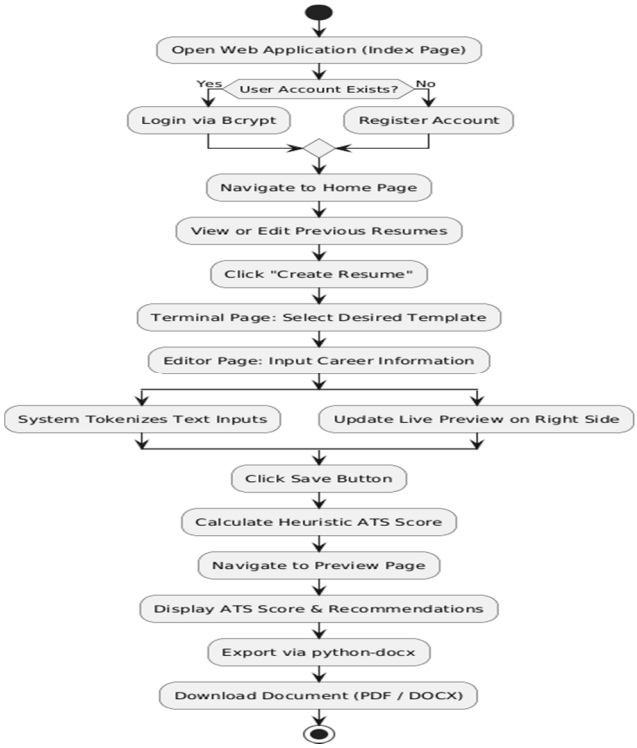


Fig. 4. System flow chart detailing the sequential execution from data entry to document generation.

IV. EVALUATION AND RESULTS

To determine the practical viability of the proposed web application, we conducted a series of functional tests focusing on usability, scoring accuracy, and the structural integrity of the final exported documents. The evaluation was carried out using simulated candidate profiles to observe how the heuristic algorithm handled varying levels of data input.

A. User Interface and Workflow Efficiency

A primary goal of this project was to eliminate the steep learning curve associated with complex formatting software. During functional testing, the interface proved to be highly intuitive. Because the system deliberately avoids a cluttered administrative panel, users are immediately greeted by a central- ized personal dashboard upon logging in. This dashboard acts as a dedicated repository, prominently displaying previously created documents alongside their respective template styles, last updated dates, and baseline ATS scores. This allows candidates to track their progress over time effortlessly.

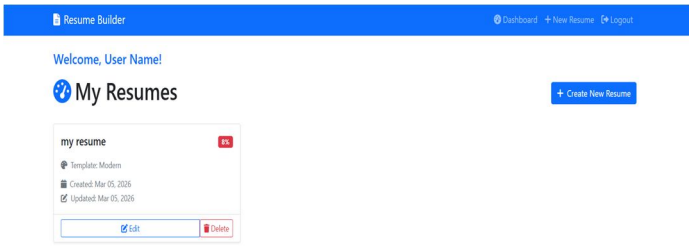


Fig. 5. The centralized user dashboard displaying saved resume drafts and their corresponding baseline ATS scores.

When a user clicks ”Create New Resume,” the system navigates to a dedicated template selection terminal. Here, the user provides a baseline title and chooses from a selection of predefined structural layouts (such as Classic, Modern, or Creative). These layouts were intentionally engineered to avoid unconventional graphics or multi-column barriers that typically trigger parsing failures in real-world recruitment software.

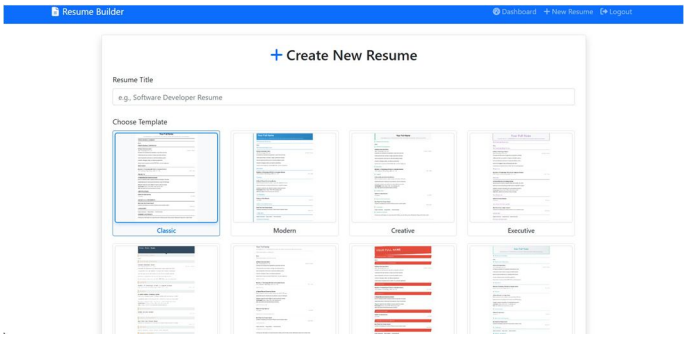


Fig. 6. The template selection module featuring clean, ATS-compliant layout options.

B. Heuristic Parsing and ATS Score Accuracy

Evaluating the text processing algorithm was a critical phase of the testing process. Because the application relies on a deterministic NLP pipeline rather than unpredictable machine learning models, the scoring mechanism proved to be exceptionally fast and transparent. We implemented a dual- pane interactive editor to facilitate this. As the candidate inputs their details—such as languages, hobbies, and professional experience—in the left-hand form fields, the system tokenizes the text. The right side of the interface continuously updates a live visual representation of the resume, complete with a dynamic ATS score badge displayed prominently at the top.

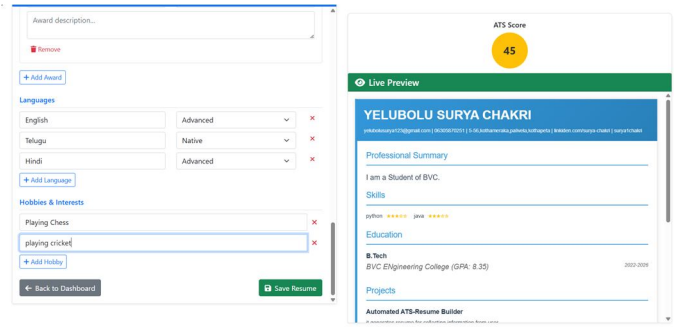


Fig. 7. The interactive dual-pane editor demonstrating real-time visual updates and dynamic ATS scoring.

Once the user finalizes their input and clicks save, the ap- plication processes the complete textual payload and navigates to a dedicated analysis page. Here, the heuristic algorithm successfully flags missing data points based on completeness and keyword density. During testing, if a profile lacked suffi- cient technical depth, the system generated specific, actionable prompts on the screen, such as ”Expand your summary with more details” or ”Add 5-8 relevant skills.” This clear, rule-based feedback mechanism effectively guides users to improve their document without confusing them with obscure algorithmic reasoning.

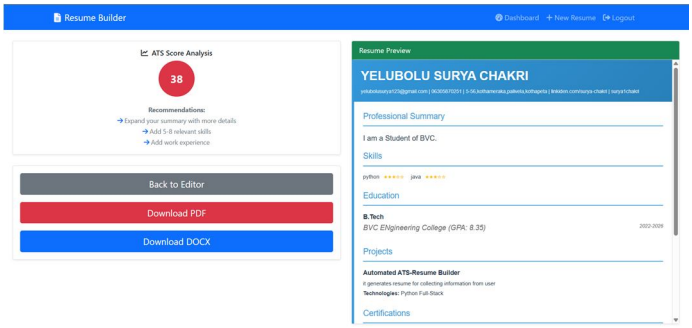


Fig. 8. The final analysis interface detailing the computed ATS score, actionable recommendations, and document export options.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

C. Output Fidelity (Document Structuring and Formatting)

The ultimate success of any resume builder relies entirely on the technical quality of the file it exports. A visually appealing web interface is useless if the downloaded document breaks when processed by an enterprise recruitment scanner. To evaluate output fidelity, we tested the server-side rendering and export capabilities of the application.

The results confirmed that the application successfully translates the web-based inputs into strictly formatted files. When users select the PDF export option, the system generates a clean, highly structured document that perfectly mirrors the live preview. Visual inspections of the generated PDFs showed perfect alignment, consistent margins, and no hidden HTML artifacts. Most importantly, the text layers remained completely intact and selectable, ensuring that optical character recognition (OCR) tools can easily scrape the candidate's core competencies without encountering any graphical interference.

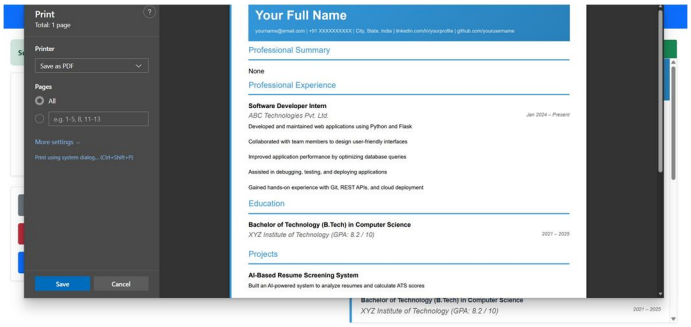


Fig. 9. The final document generation and print view, demonstrating high structural fidelity and clean text rendering for export.

V. DISCUSSION

A. Design Validation and Key Observations

Evaluating the implemented architecture reveals several practical advantages to using a deterministic web framework. The decision to construct the backend with Python and Flask, paired with a PostgreSQL database, provided a highly stable environment for concurrent data processing.

A major observation during testing was the value of algorith- mic transparency. Platforms utilizing complex deep learning models often act as black boxes, giving users a score without a clear explanation. By contrast, our rule-based Natural Language Processing (NLP) pipeline—relying strictly on text tokenization, keyword matching, and regular expressions—offered immedi- ate, predictable feedback. When a user received an ATS score of 45, the system could point exactly to the missing entities, such as a lack of quantifiable metrics or missing technical skills. Furthermore, routing the document generation through the python-docx library successfully bypassed the margins and artifact errors commonly associated with browser-based DOM printing. The exported files maintained strict typographical hierarchy, proving highly compatible with standard optical character recognition (OCR) scanners.

B. Limitations and Constraints

While the heuristic approach proved efficient, the system does operate within certain technical boundaries. Because the text analysis relies on exact keyword matching and predefined skill dictionaries, it cannot interpret deep semantic nuances. For example, the algorithm might easily detect the token ”Python,” but it struggles to differentiate between ”I am learning Python” and ”Led a team of Python developers.”

Additionally, the platform was explicitly designed as a candidate-centric tool. It features a single user dashboard for job seekers to manage their drafts but entirely omits an administrative or recruiter-facing dashboard. Consequently, the application cannot facilitate direct job-matching or allow HR departments to actively filter through the database. The scope is strictly limited to personal document optimization.

C. Alignment with Literature Findings

The outcomes of this project closely align with recent academic observations regarding recruitment software. Cur- rent literature frequently highlights that the vast majority of Applicant Tracking Systems still rely heavily on rigid keyword extraction rather than advanced semantic understanding. By forcing users to adopt clean, single-column templates and actively prompting them to inject industry-specific terminology, this application directly mitigates the primary causes of algorithmic rejection documented in contemporary research.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

VI. CONCLUSION

Bridging the gap between a candidate's actual qualifications and the rigid parsing logic of automated recruitment software remains a critical challenge. This research successfully demon- strates the design and deployment of a lightweight, highly responsive ATS-Optimized Resume Builder. By prioritizing a seamless user interface over complex administrative features, the platform empowers job seekers to draft, evaluate, and export professional profiles in real time.

The integration of a deterministic, rule-based NLP pipeline ensures that text evaluation is both fast and entirely transparent. Instead of guessing what an opaque AI model wants to see, users receive clear, token-driven recommendations to improve their content density. Coupled with a backend rendering engine that guarantees structural fidelity in PDF and DOCX exports, this system provides a practical, accessible solution for navigating modern algorithmic hiring filters. Future iterations could explore the integration of lightweight semantic similarity models to better evaluate the context of the user's work experience without sacrificing the platform's processing speed.

VII. ACKNOWLEDGMENT

The authors would like to thank Bonam Venkata Chalamayya Engineering College and the Department of Computer Science and Engineering(Artificial Intelligence and Data Science) for their support in this research work.

REFERENCES

[1] S. Washimkar, K. Gour, K. Gour, H. Bokade, S. Utane, and P. S. Prasad, "AI-Powered Resume Analyzer," Iconic Research And Engineering Journals (IRE), vol. 9, no. 8, pp. 322–328, 2026.
[2] A. Sable, A. Macharekar, S. Wagh, P. Udbatte, and J. Nanajkar, "AI Powered Application Tracking System: With NLP Based Resume Scoring," International Journal of Creative Research Thoughts (IJCRT), vol. 13, no. 6, pp. c594–c598, Jun. 2025.
[3] S. Yadav, S. Ursal, S. Tate, and A. Thade, "Resume Analysis Using NLP and ATS Algorithm," International Journal of Latest Technology in Engineering, Management & Applied Science (IJLTEMAS), vol. 14, no. 4, pp. 761–767, Apr. 2025.
[4] Y. G. Deepa, A. Sindhu, A. Shruthi, and B. Neha, "Automated Resume Parsing: A Review of Techniques, Challenges and Future Directions," In- ternational Journal of Multidisciplinary Research and Growth Evaluation, vol. 6, no. 2, pp. 1065–1069, Mar. 2025.
[5] G. Raj, "Resume Parser and Auto-Formatter Using NLP," Scientific Journal of Artificial Intelligence and Blockchain Technologies, vol. 2, no. 3, pp. 64–71, Aug. 2025.
[6] L. Satpute, P. Narnaware, and U. Kosarkar, "Crafting the Perfect Resume: Insights, Strategies, and Trends for Effective Career Building," International Journal of Trend in Scientific Research and Development (IJTSRD), pp. 93–97, 2024.
[7] M. Vivek, V. Dhivya, R. B. Asangi, and S. L. Reddy, "SMART ATS: Intelligent Resume Screening Using Machine Learning," International Journal of Innovative Research in Technology (IJIRT), vol. 11, no. 8, pp. 9217–9221, Jan. 2025.
[8] H. Kazi, P. Muthekar, U. Chavan, and J. Jagtap, "AI-Powered Resume Generator," International Journal for Research Publication and Reviews (IJRPR), vol. 6, no. 6, pp. 261–267, Jun. 2025.
[9] M. Saatc¸ı, R. Kaya, and R. U¨ nlu¨ , "Resume Screening with Natural Language Processing (NLP)," Alphanumeric Journal, vol. 12, no. 2, pp. 121–140, Dec. 2024.
[10] N. Sasikumar, S. Baskar, S. Harishkumar, G. Arunkumar, and A. N. Vinodhini, "AI-Powered Opportunity Crafter: Unlocking Smart Career Possibilities," International Journal for Research in Applied Science & Engineering Technology (IJRASET), vol. 13, no. 4, pp. 5032–5036, Apr. 2025.
[11] A. Karunamurthy and M. Barath, "AI-Powered Resume Analysis Using SpaCy for Skill Extraction and Job Matching," International Scientific Journal of Engineering and Management (ISJEM), May 2025.
[12] S. M. S. Sheikh, P. Adep, N. Aidasani, S. Chavan, and V. Darade, "AI- Powered Resume Ranking System: Enhancing Recruitment Efficiency through Natural Language Processing," International Journal for Re- search Trends and Innovation (IJRTI), vol. 10, no. 5, pp. 112–118, May 2025.

BIOGRAPHIES OF AUTHORS



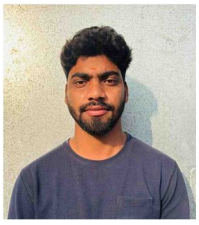
Yelubolu Surya Chakri is currently resid- ing at Odalarevu, East Godavari, Andhra Pradesh–533210. He is a B.Tech student specializing in Computer Science and En- gineering (Artificial Intelligence and Data Science) at Bonam Venkata Chalamayya Engineering College, Odalarevu, affiliated to JNTU Kakinada, with an expected grad- uation in May 2026. He aims to secure a position that leverages his strong organiza- tional skills, educational background, and ability to work effectively in professional environments. He possesses technical skills in Python full stack development, React, and MySQL.

Phone No.: 6305870251

Email: yelubolusurya123@gmail.com **ORCID:** https://orcid.org/0009-0000-948 4-8106



Telugu Vinay is currently residing at Odalarevu, East Godavari, Andhra Pradesh– 533210. He is a B.Tech student specializing in Computer Science and Engineering (Ar- tificial Intelligence and Data Science) at Bonam Venkata Chalamayya Engineering College, Odalarevu, affiliated to JNTU Kakinada, with an expected graduation in May 2026. He aims to build a successful career in the IT industry by applying his strong educational foundation and problem- solving skills. He possesses technical exper- tise in HTML, CSS, JavaScript, MySQL, and Python programming.
Phone No.: 7569140663
Email: vinaytelugu2004@gmail.com **ORCID:** https://orcid.org/0009-0007-848 5-9224



Darapureddy Yeswanth Surya Rama Mahesh is currently residing at Odalarevu, East Godavari, Andhra Pradesh–533210. He is a B.Tech student specializing in Computer Science and Engineering (Ar- tificial Intelligence and Data Science) at Bonam Venkata Chalamayya Engineering College, Odalarevu, affiliated to JNTU Kakinada, with an expected graduation in May 2026. He aims to pursue a career in data analytics and intelligent systems by applying analytical thinking and data- driven methodologies. He possesses skills in Python, MySQL, data analytics, and machine learning concepts.
Phone No.: 9010702589
Email: yeswanthdarapureddy666@gmail.c om
ORCID: https://orcid.org/0009-0006-486 1-8629



Polamuri Lukeshbabu is currently resid- ing at Odalarevu, East Godavari, Andhra Pradesh–533210. He is a B.Tech student specializing in Computer Science and En- gineering (Artificial Intelligence and Data Science) at Bonam Venkata Chalamayya Engineering College, Odalarevu, affiliated to JNTU Kakinada, with an expected grad- uation in May 2026. He aims to build a successful career in Web Development by applying his strong educational foundation. He possesses technical expertise in HTML, CSS, JavaScript, React, and Python pro- gramming.
Phone No.: 7995922501
Email: siddulukesh29@gmail.com **ORCID:** https://orcid.org/0009-0007-077 1-7943

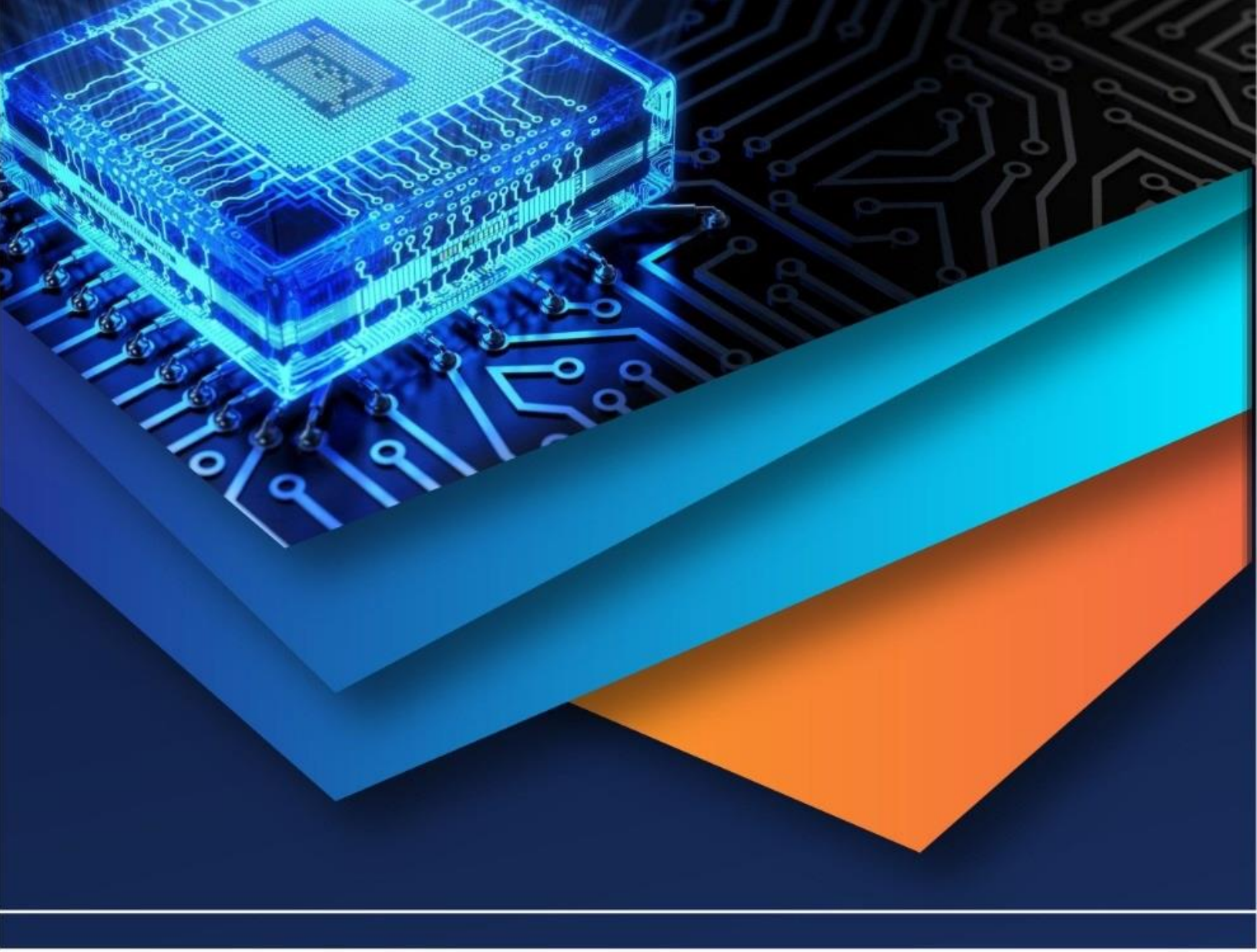


Donga Naga Venkata Sri Surya is cur- rently residing at Odalarevu, East Godavari, Andhra Pradesh–533210. He is a B.Tech student specializing in Computer Science and Engineering (Artificial Intelligence and Data Science) at Bonam Venkata Chala- mayya Engineering College, Odalarevu, affiliated to JNTU Kakinada, with an ex- pected graduation in May 2026. He aims to build a successful career in the IT industry by applying his problem-solving skills. He possesses technical expertise in Python full stack development, MySQL, and Java programming.
Phone No.: 6300978008
Email: suryadonga123@gmail.com **ORCID:** https://orcid.org/0009-0006-486 1-8629



Onteddu Siva Rama Krishna is an aca- demic professional associated with Bonam Venkata Chalamayya Engineering College, Odalarevu, Andhra Pradesh. He holds both B.Tech and M.Tech degrees and actively contributes to research and academic in- novation. He is an author of a published research paper titled ”Secure and Efficient method for Hidden Objects Crawling in Web Search Interface by KNN Queries”, published in the International Journal of Research, Vol. 5, No. 15 (2018). His areas of interest include artificial intelligence, data science, machine learning.
Phone No.: 9381093119
Email: onteddusiva2@gmail.com **ORCID:** https://orcid.org/0009-0005-517 9-5311

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)