RESUMIND: AN AI-POWERED RESUME ANALYZER, APPLICATION TRACKING SYSTEM AND JOB RECOMMENDATION SYSTEM

Anubhav Raj Singh , Mudita Rastogi , Shashank Shukla
B.Tech Student, Department of Computer Science and Engineering, Shri Ramswaroop Memorial University, Lucknow, India

***Abstract*:** In the era of a fast-paced world, organizations increasingly require people who work efficiently and are highly skilled in their areas of expertise. This paper presents Resumind — an AI-powered Resume Analyzer, Application Tracking System (ATS), and Job Recommendation System. The system leverages Natural Language Processing (NLP) and Google Gemini Pro to extract key resume information such as skills, experience, and qualifications, and compares it against job descriptions to generate an ATS match score along with targeted improvement recommendations. Resumind evaluates resumes across four independent dimensions: Tone & Style, Content, Structure, and Skills, providing multi-dimensional feedback that goes beyond conventional keyword-matching approaches. Tested on five diverse resume samples, the system demonstrated consistent processing within 3–5 seconds per submission, achieving ATS scores ranging from 69 to 88 out of 100 across varied professional domains and resume formats.

***Index Term* - Resume Analyzer, Artificial Intelligence, Natural Language Processing, Applicant Tracking System, Job Recommendation System, Resume Parsing, Google Gemini Pro.**

1. INTRODUCTION

As the world moves rapidly toward more advanced technological systems, companies increasingly require highly skilled individuals. Large organizations such as Fortune 500 companies and unicorn startups receive over thousands of applications on a daily basis. Processing this volume of applications demands significant human effort, as recruiters must carefully review each resume to identify suitable candidates. This manual process is both time-consuming and expensive for organizations, often introducing inconsistencies and cognitive biases in the screening process.

Conventional Applicant Tracking Systems (ATS) partially address this challenge by automating resume filtering through keyword matching. However, these systems are limited by their rigid rule-based nature — they frequently reject qualified candidates whose resumes are not perfectly formatted or do not use exact terminology from the job description. Furthermore, standard ATS tools provide no feedback to candidates, leaving job-seekers unaware of why their applications were rejected or how they might improve.

Resumind addresses both limitations by integrating NLP-based information extraction with Google Gemini Pro's semantic understanding to deliver structured, multi-dimensional, and job-context-sensitive resume evaluations. In addition to scoring, the system provides actionable recommendations for resume improvement and job-specific positioning. A supplementary job recommendation module further extends its utility by suggesting suitable roles based on the candidate's detected skill profile, making Resumind a comprehensive end-to-end career preparation platform.

1.1. Objectives

The primary objectives of the Resumind system are to: (i) build a resume analyzer that evaluates resumes against the desired job position, job description, and target company; (ii) recommend suitable job positions based on the user's resume profile; (iii) classify key resume entities such as name, email, skills, qualification, and experience; (iv) support multiple input file formats including PDF and DOCX with real-time parsing via the Gemini API; and (v) provide structured, dimension-specific feedback and ATS improvement guidance to help candidates optimize their resumes.

1.2. Literature Survey

Samsun A. and Dr. A. Angel Cerli [1] proposed a Resume Parser using Natural Language Processing and Machine Learning techniques for automated extraction of key resume sections, noting limitations in handling diverse formats and ambiguous overlapping information. Kokade and Nikam [2] developed a Resume Analyser with ATS Score using NLP, providing match scoring against job descriptions, but the system lacked AI-driven improvement feedback and predictive features for job seekers. Salitri et al. [3] combined a resume builder with an NLP-based analyzer, while Abhishek et al. [4] introduced an intelligent screening tool incorporating AI-driven analysis and recommendation capabilities. Ali et al. [9] proposed a resume classification system using NLP and ML to categorize resumes by domain. The proposed Resumind system advances these works by integrating large language model (LLM)-based semantic scoring, persistent per-user analysis history, and a job recommendation engine into a unified platform.

1.3. Proposed System

The proposed system presents a resume analyzer and job recommendation platform using NLP and a large language model (Google Gemini Pro). The system pipeline includes the following stages: (i) Document Reading — supporting PDF and DOCX input formats; (ii) Text Extraction — using optical and text parsing libraries (PDF.js, mammoth); (iii) Text Preprocessing — comprising tokenization, stop word removal, stemming, and normalization; (iv) Resume Content Structuring — categorizing content into Skills, Experience, Education, and Qualification; (v) AI-Based Resume Evaluation and Job Recommendation — leveraging the Gemini Pro API for semantic assessment and role matching; and (vi) Structured Output Generation — producing a JSON response consumed by the React UI for real-time display of scores, recommendations, and ATS flags. The system is designed to be stateless, scalable, and extensible, with MongoDB providing persistent storage for per-user analysis history.

2. METHODOLOGY

Resumind is a comprehensive full-stack web application employing a React frontend, an Express.js/Node.js backend, a MongoDB database for persistent storage, and Google Gemini Pro as the core AI analysis engine. The system pipeline comprises five sequential stages: (1) document ingestion, (2) text extraction, (3) preprocessing, (4) AI-based evaluation, and (5) structured output generation. Each stage is described in detail below.

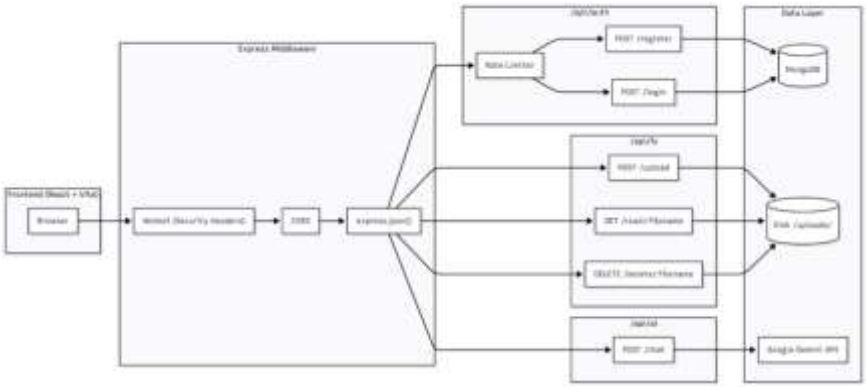

Fig. 1: System Backend Architecture and API Routing Flow

2.1. User Authentication and Access Control

Users access the system by registering and signing in through email and password credentials. The authentication subsystem employs three complementary security mechanisms. Helmet middleware is applied at the Express.js layer to protect against Cross-Site Scripting (XSS), Clickjacking, MIME-type sniffing, and information leakage attacks [7]. Passwords are hashed using BCrypt, a key derivation function based on the Blowfish cipher, which incorporates an adaptive salt to resist rainbow table and brute-force attacks [5]. Authenticated sessions are managed using JSON Web Tokens (JWT), which encode claims as digitally signed JSON entities, enabling stateless and verifiable authorization across requests [6]. Login and Signup API endpoints are rate-limited to 10 requests per 15-minute window per IP address to prevent credential-stuffing attacks.

2.2. Document Processing and Parsing

Resume acquisition begins at the React frontend, which provides an upload interface for users to select a resume file (PDF or DOCX) from their local system. Upon selection, React captures the file through an input component and transmits it to the backend via an HTTP multipart/form-data API request. Express.js routes the request to the upload controller, where Multer middleware handles the multipart payload and retains the file in server memory for processing. PDF content is subsequently extracted using PDF.js [8], which provides flexible API interfaces for interacting with PDF documents. DOCX content is parsed using the mammoth library, converting Word document content to raw plaintext. Both extraction paths yield unstructured text ready for the preprocessing stage.

2.3. Text Preprocessing

Following document parsing, the extracted resume text is passed through a preprocessing pipeline to transform raw content into structured, analysis-ready data. The pipeline applies three normalization steps: (i) stop word removal, which eliminates low-information words (e.g., "a", "an", "the", "is") to reduce dataset size and improve computational efficiency; (ii) stemming, a fast technique that strips suffixes and prefixes to reduce words to a common root form; and (iii) lemmatization, a context-aware method that uses vocabulary and morphological analysis to return words to their dictionary base form. Normalized tokens are then classified into named entity categories — Name, Email, Skills, Education, Qualification, and Experience — using NLP-based entity recognition. This structured representation enables the AI model to interpret resume content consistently across varying formats and writing styles.

2.4. AI-Based Resume Analysis

The structured resume content, together with the user-supplied target job title, company name, and optional job description, is transmitted to Google Gemini Pro via Express.js backend API calls. Gemini Pro performs semantic evaluation across four independent dimensions: Tone & Style (appropriateness of language and professionalism), Content (depth of role-specific achievements and action-oriented language), Structure (logical organization of resume sections), and Skills (alignment of detected skills with the target role). For each dimension, the model returns a numerical score (0–100) and a qualitative label (e.g., "Good Start", "Needs Work", "Strong"). An overall ATS compatibility score and a criterion-level ATS flag checklist with pass/fail status per criterion are also generated. The system's stateless architecture — each analysis conducted via a transient API payload with no persistent model state — ensures independent per-submission evaluation and supports concurrent multi-user usage without performance degradation.

2.5. Resume Enhancement and Job Recommendation

Based on the Gemini Pro evaluation, Resumind generates targeted enhancement recommendations including: identification of missing or weak skills relevant to the job role; suggestions for improving resume content depth and structural clarity; keyword optimization guidance to enhance ATS compatibility; and overall quality feedback aligned with recruiter expectations. The job recommendation module analyzes the candidate's detected skill and domain profile to match suitable job roles using keyword and semantic relevance analysis, generating a ranked list of personalized job suggestions. All analysis results — scores, recommendations, and ATS flags — are stored per-user in MongoDB, enabling candidates to track the longitudinal impact of successive resume revisions across multiple target roles over time.

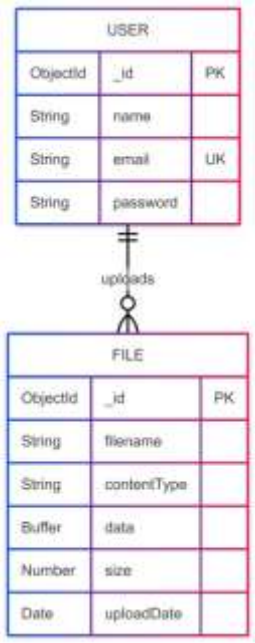

Fig. 2: Database Schema -- USER and FILE Entity Structure

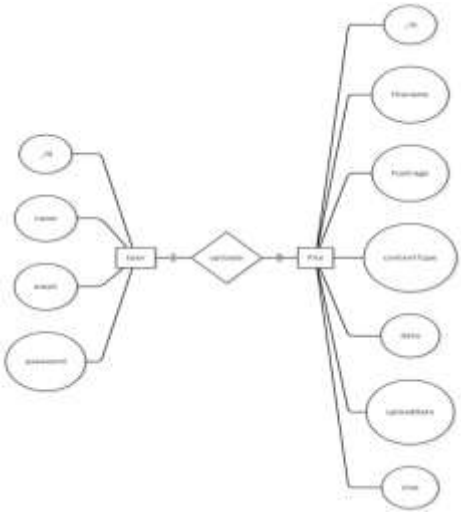

Fig. 3: Entity-Relationship Diagram (Chen Notation)

3. RESULTS AND DISCUSSION

This section presents the experimental evaluation of Resumind. The system was tested by submitting five resume samples (R1–R5) with varying structural formats, target job roles, and target organisations. All submissions included a PDF-format resume alongside user-provided contextual inputs: the target job title, company name, and an optional job description. Results are analysed at both the overall and category level, followed by a comparative discussion against conventional manual screening practices.

3.1. Experimental Setup and Overall ATS Scores

The five samples spanned frontend development, DevOps engineering, data analytics, ML engineering, and project management domains, in single-column, two-column, and hybrid layout formats. Table 1 summarises each sample's characteristics and the corresponding overall ATS score generated by Resumind.

Table 1: Resume Sample Characteristics and Overall ATS Compatibility Scores

Resume ID	Format Type	Target Job Title	Target Company	ATS Score (/100)
R1	Single-column	Frontend Developer	JavaScript Mastery	82
R2	Two-column	DevOps Engineer	Columbia Tech Group	88
R3	Single-column	Data Analyst	MNC – Finance Sector	74
R4	Hybrid Layout	ML Engineer	AI Research Firm	79
R5	Two-column	Project Manager	Government Agency	69

ATS scores ranged from 69 (R5, project management) to 88 (R2, DevOps). Resume R2 achieved the highest ATS score yet the lowest Content score (25/100) across all samples, demonstrating a well-documented gap between ATS compatibility and content quality. Resume R5 recorded the lowest ATS score (69), attributable to weaker keyword alignment and structural inconsistencies that impede automated parsing. Resumind's multi-dimensional scoring framework makes such divergences explicitly visible to candidates.

3.2. Category-Wise Scoring Analysis

Resumind evaluates each resume across four diagnostic dimensions. Table 2 presents the complete category-wise scores for all five samples alongside mean scores computed per dimension.

Table 2: Category-Wise Scores for Resume Samples R1–R5 (Scores out of 100)

Resume	Tone & Style (/100)	Content (/100)	Structure (/100)	Skills (/100)	ATS Score (/100)
R1	72	78	80	75	82
R2	55	25	70	32	88
R3	68	71	65	80	74
R4	80	76	78	82	79
R5	58	62	60	70	69
Mean	66.6	62.4	70.6	67.8	78.4

Structure recorded the highest mean score (70.6), indicating that the LLM-based evaluation consistently recognised logical section organisation across most submissions. Skills (mean 67.8) and Tone & Style (mean 66.6) followed, while Content recorded the lowest mean (62.4), reflecting that most resumes lacked depth of role-specific achievements, quantifiable outcomes, and action-oriented language expected by recruiters. The R2 case is the most notable: despite the highest ATS score, it recorded the lowest Content (25) and Skills (32) scores, confirming that ATS-optimised resumes can pass automated screening while remaining substantively weak. Table 3 presents the granular ATS flag output for R2.

Table 3: ATS Flag Output for Resume R2 (DevOps Engineer – Columbia Tech Group)

ATS Check Criterion	Status	Notes
Clear formatting, readable by ATS	✓ Pass	Two-column layout parsed correctly
Keywords relevant to the job	✓ Pass	Job title & description matched
Skills section detected	✗ Fail	No dedicated skills section found

While R2 passed ATS checks for formatting readability and keyword relevance, the absence of a clearly delineated skills section was flagged as a failure criterion. This criterion-level feedback directs the candidate's attention to specific, correctable deficiencies rather than providing only an aggregate score — a key practical differentiator of the system.

3.3. System Performance

Resumind demonstrated consistent and responsive performance across all five submissions. Resume analysis — encompassing PDF upload via Multer, Gemini Pro API invocation, and structured result rendering on the React frontend — was completed within 3–5 seconds per submission in all cases. No degradation in output completeness or scoring consistency was observed when varying resume length, structural complexity, or the specificity of the accompanying job description.

A notable behavioural property is context-sensitivity: resumes evaluated against technically precise roles — such as ML Engineer at an AI Research Firm (R4) — received more discriminative Skills-dimension feedback compared to resumes submitted against broader roles, confirming that Gemini Pro correctly conditions its evaluation on the user-supplied job context. The system's stateless architecture further ensures that each analysis is independent, eliminating cross-contamination between successive submissions.

3.4. Comparative Analysis: Resumind vs. Manual Resume Screening

Table 4 presents a structured comparison between Resumind and conventional manual resume screening across eight operationally relevant dimensions.

Table 4: Comparative Analysis – Resumind vs. Manual Resume Screening

Feature	Resumind (Proposed System)	Manual Screening
AI Engine	Google Gemini Pro (LLM)	Human recruiter judgement
Evaluation Speed	< 5 seconds per resume	15–30 minutes per resume
Scoring Dimensions	4 categories + ATS score	Unstructured / subjective
Job-Specific Analysis	Title, company & description	Depends on recruiter expertise
Feedback Type	Structured checklist per category	Verbal / inconsistent
History Tracking	MongoDB – per-user history	Not available
Bias Potential	Reduced via LLM prompting	High (cognitive / affinity bias)
Scalability	High (stateless API calls)	Low (human bottleneck)

The most operationally significant advantage is the reduction in evaluation time from 15–30 minutes per resume to under five seconds — a substantial improvement in recruiter productivity in high-volume hiring environments. The system produces reproducible, multi-dimensional scoring free from the cognitive biases — including affinity bias, presentation bias, and recency effects — that commonly affect human screening decisions. MongoDB-backed per-user analysis history enables candidates to longitudinally track the impact of successive revisions across multiple target roles, a capability entirely absent from manual workflows.

It is acknowledged that manual screening retains advantages in areas requiring deep contextual understanding — such as evaluating cultural fit, interpreting non-linear career trajectories, or assessing soft skills from written communication style. Resumind is therefore positioned as a complementary candidate preparation tool rather than a replacement for human recruitment judgment.

3.5. Discussion

The experimental results collectively demonstrate that Resumind successfully fulfils its core design objectives: generating rapid, structured, multi-dimensional, and job-context-sensitive resume evaluations using a large language model backend. The scoring outputs are internally consistent and behaviourally meaningful, evidenced by the observed divergence between ATS and content quality scores for R2. Limitations of the current system include: (i) scoring variability contingent on the quality of prompt engineering used to instruct Gemini Pro; (ii) dependence on Google Generative AI infrastructure for availability and latency; and (iii) an evaluation dataset limited to five samples. Future work should address these through prompt calibration

experiments, introduction of a human expert benchmark for ground-truth score validation, and expanded testing across a broader corpus of resume formats, industries, and experience levels.

4. CONCLUSION

This paper presented Resumind, an AI-powered resume analysis and job recommendation platform that integrates NLP-based text extraction with Google Gemini Pro's large language model capabilities. The system delivers sub-5-second, four-dimensional resume evaluations tailored to specific job contexts, with ATS compatibility scores ranging from 69 to 88 across five professional domains. The multi-dimensional scoring framework — encompassing Tone & Style, Content, Structure, and Skills — bridges the gap between ATS compatibility and genuine resume quality, as evidenced by the R2 case study where a high ATS score (88) coexisted with a low Content score (25). MongoDB-backed per-user history enables longitudinal candidate self-improvement tracking across successive revisions and target roles.

Resumind offers a scalable, bias-reduced, and practically viable complement to traditional recruitment screening workflows. The system's stateless API architecture supports concurrent multi-user operation without performance degradation, while its context-sensitive LLM evaluation delivers more discriminative feedback than conventional rule-based ATS tools. Future work is planned around prompt calibration for improved scoring consistency, human expert benchmark validation, expansion of the evaluation corpus across broader industries and experience levels, and development of a dedicated job recommendation engine that leverages stored MongoDB analysis history to suggest alternative roles aligned with the candidate's detected skill profile.

5. REFERENCES

[1] Samsun A and Dr. A. Angel Cerli, "Resume Parser Using Natural Language Processing and Machine Learning," International Journal of Multidisciplinary Research, vol. 3, no. 2, pp. 357–361, 2025.

[2] S. A. Kokade and Y. Nikam, "Resume Analyzer with ATS Score Using AI and ML," Journal of Emerging Technologies and Innovative Research, vol. 12, no. 4, pp. 3, Apr. 2025.

[3] A. Salitri, S. Tanpure, S. Pujari, P. Salunkhe, S. Sakore, and S. Salunkhe, "Resume builder website and AI resume analyzer using NLP," AIP Conference Proceedings, vol. 3325, p. 030031, 2025, doi:10.1063/5.0291770.

[4] K. L. Abhishek et al., "Developing an intelligent resume screening tool with AI-driven analysis and recommendation features," AI and Ethics, May 2025, doi:10.1002/ail2.116.

[5] P. Sriramya and R. A. Karthika, "Providing password security by salted password hashing using bcrypt algorithm," International Journal of Scientific & Engineering Research, vol. 6, no. 5, pp. 109–112, May 2015.

[6] P. Mahindrakar and U. Pujeri, "Insights of JSON Web Token," International Journal of Recent Technology and Engineering (IJRTE), vol. 8, no. 6, pp. 1707–1711, Mar. 2020, doi:10.35940/ijrte.F7689.038620.

[7] W. Uriawan et al., "Authenticate and verification source files using SHA256 and HMAC algorithms," Preprints, Jul. 2024, doi:10.20944/preprints202407.0075.v1.

[8] Y. Qian and Y. Li, "Web-based PDF Document Display Research and Implementation," Scientific Journal of Intelligent Systems Research, vol. 7, no. 7, pp. 28–37, 2025, doi:10.54691/r3n8m337.

[9] I. Ali, N. Mughal, Z. H. Khand, J. Ahmed, and G. Mujtaba, "Resume classification system using natural language processing and machine learning techniques," Mehran University Research Journal of Engineering & Technology, vol. 41, no. 1, pp. 65–79, 2022.