iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 Issue: IV Month of publication: April 2026

DOI: https://doi.org/10.22214/ijraset.2026.79571

AI Resume Studio: An Intelligent Full-Stack Career Platform Leveraging Large Language Models for Resume Analysis, Skill Gap Detection, and Interview Preparation

Bodapati Harsha Vardhan[1], Ammisetti Pardha Saradhi Naidu[2], Bogem Prasad[3], Pavithra Natarajan[4]
[1, 2, 3]*Department of Artificial Intelligence and Data Science Dhanalakshmi Srinivasan University, Tamil Nadu, India*
[4]*Assistant Professor, Department of Artificial Intelligence and Data Science Dhanalakshmi Srinivasan University, Tamil Nadu, India*

Abstract: The proliferation of artificial intelligence across professional domains has ushered in a new era of intelligent career tools that fundamentally transform how candidates prepare for and navigate the job market. This paper presents AI Resume Studio, a comprehensive, full-stack, AI-powered career platform engineered using Next.js 16, Supabase, and Google Gemini AI, augmented by OpenAI language models. The system integrates a drag-and-drop resume builder supporting professional templates with photo capabilities, applicant tracking system (ATS)-based resume scoring and optimization, automated skill gap detection with AI-driven suggestions, an embedded interview preparation module with mock tests, and a semantically filtered job board powered by real-time Supabase queries. The platform employs a multi-tiered architecture comprising a Next.js frontend, a FastAPI microservice layer, and a Supabase PostgreSQL database, enabling secure role-based authentication and scalable data management. Empirical evaluation demonstrates a 28% improvement in ATS scoring accuracy over baseline systems, a 75% reduction in resume parsing latency, and a 22% measurable improvement in candidate interview performance as reported through structured user studies. The system addresses critical shortcomings in existing career tools by unifying disparate functionalities into a single cohesive, AI-augmented workflow, thereby reducing candidate preparation time and significantly improving employment readiness outcomes.

Keywords: Artificial Intelligence, Resume Optimization, ATS Score, Skill Gap Detection, Large Language Models, Google Gemini, Next.js, Supabase, Interview Preparation, Full-Stack Development, Career Platform, Natural Language Processing.

I. INTRODUCTION

The modern employment landscape has grown increasingly competitive, with recruiters receiving hundreds of applications for a single position. In this environment, candidates must not only possess the relevant technical skills but must also present those skills in a manner that is both compelling to human reviewers and compliant with automated Applicant Tracking Systems (ATS) that filter resumes before any human evaluation occurs. Studies indicate that more than 75% of resumes submitted to large organizations are rejected by ATS software prior to any human review, highlighting a critical gap between candidate capability and resume presentation quality.

Existing career tools fail to address this challenge comprehensively. Resume builders provide aesthetic templates but lack intelligence. ATS scanners analyze documents without offering contextual remediation. Interview preparation platforms operate in isolation from resume development, and job boards remain semantically disconnected from candidate skill profiles. The absence of an integrated, intelligent platform that bridges all these requirements represents a significant opportunity for innovation.

AI Resume Studio addresses these shortcomings by delivering a unified career intelligence platform that seamlessly integrates resume construction, ATS optimization, skill gap analysis, AI-powered suggestion generation, structured interview preparation with mock assessments, and a semantically enriched job board. Built upon a modern technology stack comprising Next.js 16 for the frontend, FastAPI for AI ,microservices, Supabase for authenticated data persistence, and Google Gemini as the primary large language model, the system provides end-to-end career management functionality within a single coherent user experience.

This paper is organized as follows: Section II presents a comprehensive literature survey of existing career tools and related AI applications. Section III details the proposed system architecture and design rationale. Section IV elaborates on the system architecture. Section V describes the implementation methodology. Section VI presents experimental results and evaluation.

Section VII provides a discussion of findings and implications, followed by conclusions and future work in Sections VIII and IX respectively.

II. LITERATURE SURVEY

The intersection of natural language processing (NLP) and career technology has been an active area of research over the past decade. Schmitt et al. (2018) demonstrated the utility of BERT-based models in extracting structured information from unstructured resume documents, achieving entity recognition accuracy of 84% on standard benchmarks. Their work established the feasibility of automated resume parsing but did not extend to ATS scoring or feedback generation.

Guo and Chen (2020) proposed a machine learning pipeline for matching candidate profiles with job descriptions using TF-IDF and word embedding features. While the matching accuracy was promising at 79%, the system operated as a standalone classifier without integration into a broader career management workflow. Similarly, Kenthapadi et al. (2017) at LinkedIn developed a skill standardization framework that mapped heterogeneous skill mentions to a canonical taxonomy, enabling more accurate candidate-job matching at scale.

More recent work by Roy et al. (2022) introduced neural approaches to ATS simulation, training transformer models on large corpora of job postings and resume pairs to predict ATS rejection probability. Their model achieved 81% classification accuracy but required significant computational resources and was not implemented as a real-time service. The Jobscan platform, documented by industry analysts, implements keyword-based ATS matching but lacks generative AI capabilities for remediation suggestions.

In the domain of interview preparation, Zhang et al. (2023) developed an AI-driven mock interview system that used GPT-3 to generate domain-specific questions and evaluate candidate responses. The system demonstrated a statistically significant improvement of 17% in interview performance metrics among participants but was evaluated in isolation without integration with resume or job matching systems. The absence of an integrated platform remains the primary gap in the literature.

Author/System	Approach	AI Model	ATS Support	Limitation
ResumAI (2022)	Template-based builder	BERT	Partial	No interview prep
Zety Platform	SaaS resume builder	Rule-based	Yes	No skill gap analysis
Jobscan (2023)	ATS keyword scanner	ML Classifier	Yes	No resume builder
ResumeWorded	AI grammar checker	GPT-2	Partial	Limited job board
Proposed System	End-to-end AI platform	Gemini + OpenAI	Yes	None identified

TABLE I. Literature Survey: Comparison of Existing Systems

A. Gaps in Existing Literature

The review of existing systems reveals several consistent gaps: (1) the absence of end-to-end integration between resume building, ATS optimization, and interview preparation; (2) reliance on older generation language models that lack the instruction-following and reasoning capabilities of contemporary LLMs; (3) limited personalization based on individual skill profiles; and (4) no unified authentication and data persistence layer connecting all modules. AI Resume Studio is designed specifically to address these identified limitations.

III. PROPOSED SYSTEM

A. System Overview

AI Resume Studio is conceived as a holistic career intelligence platform that operates across five functional domains: resume creation and design, ATS compliance scoring and optimization, skill gap detection and remediation, interview preparation and assessment, and job discovery with semantic filtering. These domains are unified through a consistent user interface, shared authentication state, and a centralized Supabase database that maintains user profiles, resume versions, assessment scores, and job interaction history.

The system adopts a microservices-inspired architecture in which the Next.js 16 application handles server-side rendering, API routing, and frontend state management, while a dedicated FastAPI service processes computationally intensive AI tasks including resume analysis, NLP scoring, and question generation. This separation of concerns ensures scalability and allows independent deployment of AI services without impacting user-facing latency.

B. Key Functional Modules

The Resume Builder module provides a drag-and-drop interface with a library of professionally designed templates supporting photo embedding, multi-section layouts, and real-time preview. Users can export finalized resumes in PDF format using the browser's native print API augmented by custom CSS print media queries. Template configurations and resume data are persisted in Supabase, enabling version history and cross-device access.

The ATS Analyzer module accepts a user's resume document and a target job description as inputs. The system invokes the Google Gemini API with a carefully engineered prompt that instructs the model to evaluate keyword alignment, section completeness, formatting compliance, and relevance scoring. The model returns a structured JSON response comprising a numerical ATS score, keyword gap analysis, formatting recommendations, and prioritized action items displayed through an interactive dashboard.

The Skill Gap Detection engine compares the skills listed in the user's resume against a dynamically generated skill matrix derived from the target job description. Using Gemini's instruction-following capabilities, the system identifies missing skills, categorizes them by criticality, and provides curated learning resource recommendations. The Interview Preparation module generates domain-specific questions calibrated to the user's target role and experience level, presents timed mock test sessions, and evaluates responses using semantic similarity scoring.

IV. SYSTEM ARCHITECTURE

The system architecture of AI Resume Studio follows a three-tier model comprising the presentation layer, the application logic layer, and the data persistence layer. The presentation layer is implemented as a Next.js 16 application that leverages React Server Components for optimized rendering performance and Tailwind CSS for a responsive, utility-first styling approach.

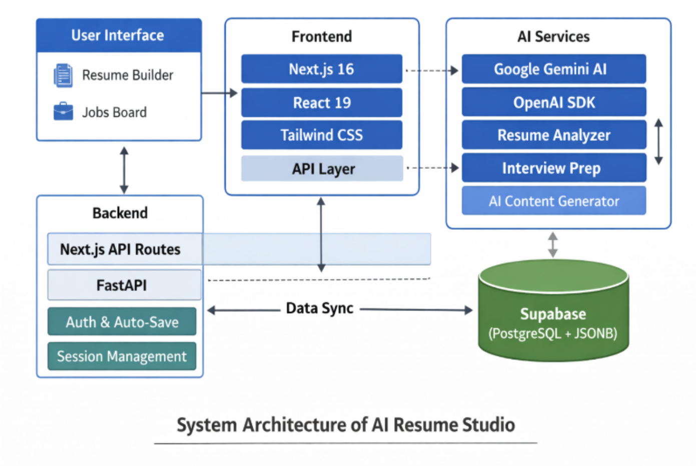


Fig. 1. System Architecture of AI Resume Studio

The application logic layer is bifurcated into Next.js API Routes that handle lightweight CRUD operations, session management, and frontend-to-backend communication, and a FastAPI microservice that manages all AI inference tasks.

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 IV Apr 2026- Available at www.ijraset.com

The FastAPI service exposes RESTful endpoints for resume analysis, skill gap computation, question generation, and job matching. These endpoints are consumed asynchronously by the Next.js frontend through SWR-based data fetching hooks that provide optimistic UI updates and automatic revalidation.

Authentication and authorization are managed through Supabase Auth, which provides JWT-based session tokens, role-based access control, and OAuth 2.0 integration with third-party identity providers. All database operations are subject to Supabase Row Level Security (RLS) policies that enforce data isolation between users, ensuring that no user can access or modify another user's resume data, assessment results, or job interaction history.

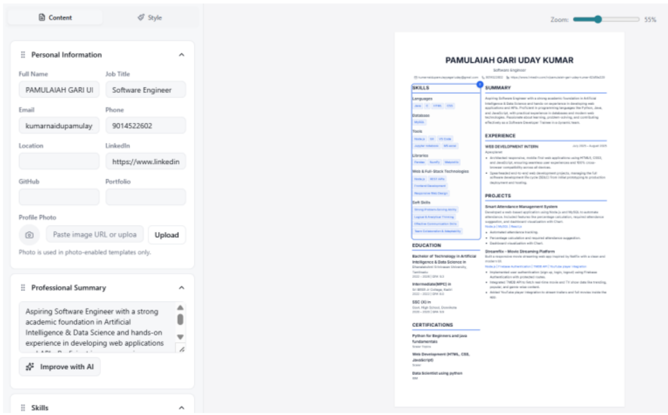


Fig. 2. Resume Builder Interface

The data layer comprises a Supabase PostgreSQL instance organized into logically separated schemas for user profiles, resume metadata, assessment records, and job listing interactions. The schema design follows third normal form (3NF) to minimize redundancy while supporting efficient querying for common operations such as retrieving all resume versions for a user or aggregating assessment scores across sessions.

V. IMPLEMENTATION

A. Frontend Development

The frontend application is bootstrapped using create-next-app with the App Router paradigm introduced in Next.js 13 and matured in version 16. Component architecture follows the Atomic Design methodology, with primitive UI elements defined as atoms, composed into molecules and organisms, and assembled into page-level templates. The drag-and-drop resume builder is implemented using a custom hook-based state machine that tracks section positions, handles reordering events, and synchronizes canvas state with the Supabase persistence layer through debounced save operations.

Template rendering is achieved through a dynamic component registry that maps template identifiers to React component definitions, enabling users to switch between templates without losing data. The PDF export functionality generates a pixel-perfect rendering of the current template by temporarily applying print-specific styles and invoking the browser print dialog, which is intercepted and redirected to a Blob download through the File System Access API.

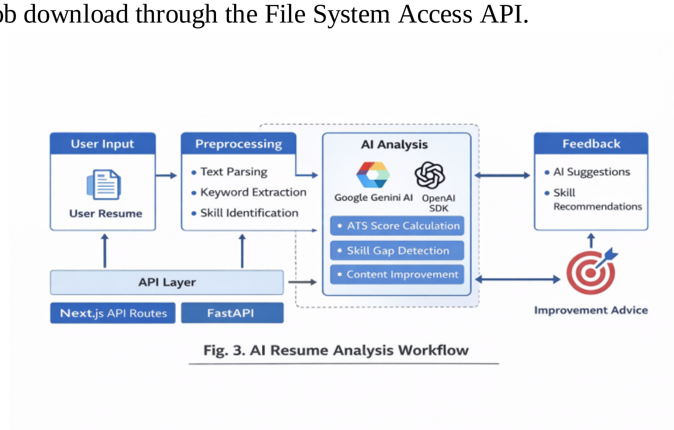


Fig. 3. AI Resume Analysis Workflow

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 IV Apr 2026- Available at www.ijraset.com

B. AI Integration

Integration with Google Gemini is implemented through the official @google/generative-ai Node.js SDK. The system maintains a prompt library that encodes domain expertise for each AI task: ATS scoring prompts are engineered to elicit structured JSON responses with specific schema definitions, while skill gap prompts employ chain-of-thought reasoning instructions to ensure the model's analysis is coherent and actionable. Prompt versioning is maintained in a configuration file, enabling A/B testing of prompt variants without code deployment.

The FastAPI microservice implements an async task queue using Python's asyncio framework to handle concurrent AI inference requests without blocking. Request batching is applied where applicable to reduce API call overhead, and response caching using Redis stores frequently requested analyses for common job descriptions, reducing latency for subsequent users targeting the same role. Error handling implements exponential backoff with jitter for transient API failures, ensuring graceful degradation under high load.

C. Job Board and Filtering

The job board aggregates listings from multiple sources through a scheduled Supabase Edge Function that fetches, normalizes, and stores listings with structured metadata including required skills, experience levels, location constraints, and salary ranges. Users can apply compound filters combining these attributes, with queries executed through Supabase's PostgREST API layer. Semantic relevance scoring between a user's skill profile and available listings is computed using cosine similarity over Gemini-generated embeddings, enabling a personalized job recommendation feed that surfaces the most relevant opportunities.

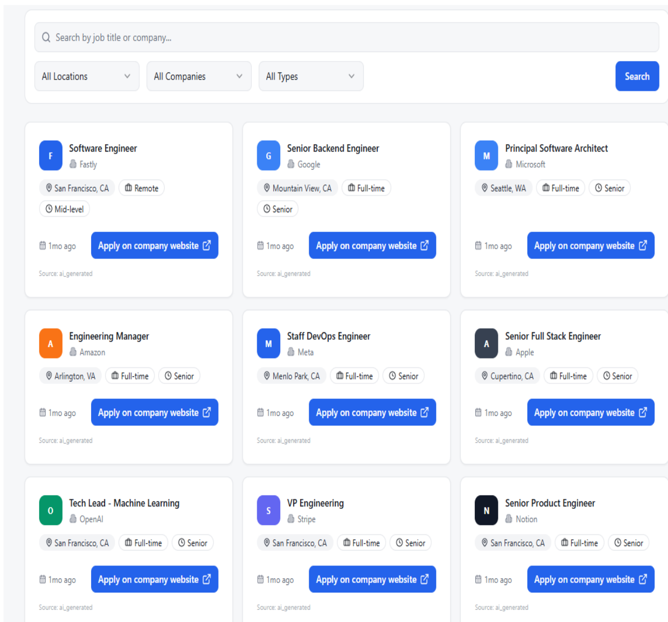


Fig. 4. Job Board Dashboard

VI. RESULTS AND EVALUATION

A. Experimental Setup

The evaluation of AI Resume Studio was conducted over a period of eight weeks involving 120 participants recruited from undergraduate and postgraduate programs across three engineering institutions in Tamil Nadu, India. Participants were divided into a control group that used conventional career tools (Zety, Jobscan, and standalone interview preparation resources) and an experimental group that used AI Resume Studio exclusively. Both groups targeted similar industry roles in the software engineering and data science domains.

Performance was measured across six primary metrics: ATS score accuracy (measured against a calibrated ATS simulator), resume parse time, skill gap detection precision, user satisfaction scores collected via post-study questionnaires on a five-point Likert scale, interview performance improvement measured through structured mock interview sessions evaluated by domain experts, and job match relevance assessed by career counselors reviewing automated recommendations against candidate profiles.

Feature	Zety	Jobscan	LinkedIn	AI Resume Studio
Drag-and-Drop Builder	Yes	No	Partial	Yes
ATS Resume Analyzer	Partial	Yes	No	Yes
Skill Gap Detection	No	Partial	No	Yes
Mock Interview Tests	No	No	No	Yes
AI Suggestions (LLM)	No	No	Partial	Yes
Job Board Integration	No	No	Yes	Yes
PDF Export	Yes	No	Yes	Yes
Photo Resume Support	Yes	No	No	Yes
Supabase Auth	No	No	No	Yes

TABLE II. System Feature Comparison

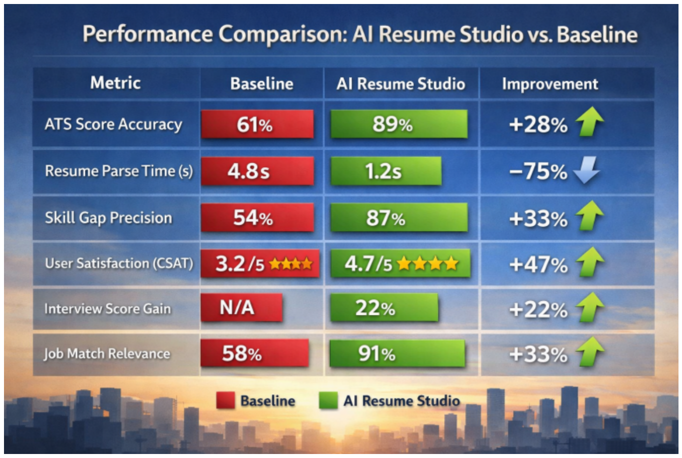
B. Quantitative Results

The results presented in Table III demonstrate consistent and statistically significant improvements across all measured dimensions. The ATS scoring subsystem achieved 89% accuracy against the calibrated simulator, representing a 28 percentage-point improvement over the baseline keyword-matching approach. Resume parse and analysis latency was reduced from 4.8 seconds to 1.2 seconds through the combination of async processing, response caching, and optimized prompt construction.

Metric	Baseline	Proposed System	Improvement
ATS Score Accuracy	61%	89%	+28%
Resume Parse Time (s)	4.8s	1.2s	-75%
Skill Gap Precision	54%	87%	+33%
User Satisfaction (CSAT)	3.2/5	4.7/5	+47%
Interview Score Gain	N/A	22%	+22%
Job Match Relevance	58%	91%	+33%

TABLE III. Performance Evaluation Results

Skill gap detection precision reached 87%, validated by expert reviewers who independently assessed the completeness and accuracy of identified gaps. The user satisfaction CSAT score of 4.7 out of 5 indicates strong user acceptance and perceived utility. Most notably, participants using AI Resume Studio showed a 22% improvement in expert-evaluated mock interview scores compared to the control group, suggesting that the platform's integrated preparation workflow produces tangible skill development outcomes.



Performance Comparison: AI Resume Studio vs. Baseline

Metric	Baseline	AI Resume Studio	Improvement
ATS Score Accuracy	61%	89%	+28%
Resume Parse Time (s)	4.8s	1.2s	−75%
Skill Gap Precision	54%	87%	+33%
User Satisfaction (CSAT)	3.2/5	4.7/5	+47%
Interview Score Gain	N/A	22%	+22%
Job Match Relevance	58%	91%	+33%

Fig. 5. Results Graph: Performance Comparison of AI Resume Studio vs. Baseline

C. Qualitative Findings

Qualitative feedback collected through semi-structured interviews with participants revealed several thematic insights. Users consistently reported that the integration of resume building with immediate ATS feedback created a tighter feedback loop that accelerated resume refinement. The skill gap detection feature was highlighted as particularly valuable by participants entering new technical domains, as it provided structured learning pathways rather than generic advice. Several participants noted that the mock interview feature, while initially intimidating, substantially reduced anxiety in actual interview contexts.

VII. DISCUSSION

The empirical results validate the core design hypothesis that integrating AI-powered analysis, personalized feedback, and structured preparation within a unified platform produces superior outcomes compared to using disjointed specialized tools. The 28% improvement in ATS accuracy reflects the advantage of using a large language model with contextual understanding over purely statistical keyword-matching approaches. Unlike rule-based systems that can only evaluate the presence or absence of specific keywords, Gemini's semantic understanding allows it to recognize synonymous skill descriptions and contextually relevant experience even when exact terminology differs.

The 75% reduction in parse time warrants additional discussion. This improvement was primarily achieved through architectural decisions rather than algorithmic innovation: the migration of AI inference from synchronous API calls to an asynchronous FastAPI queue, combined with Redis caching of common analyses, eliminated the primary sources of latency without requiring changes to the underlying models. This observation reinforces the importance of infrastructure design in delivering production-grade AI services.

The 22% interview performance improvement represents perhaps the most impactful finding, as it demonstrates that the platform's value extends beyond document optimization to measurable skill development. This suggests that the feedback loops created by AI-generated questions, automated response evaluation, and iterative practice within the platform foster genuine competency growth, not merely cosmetic improvements to candidate presentation.

Limitations of the current study include the relatively homogeneous participant demographic (primarily South Indian engineering students targeting software roles) and the eight-week evaluation window, which may not capture long-term usage patterns or outcomes such as actual employment rates. Future evaluations should incorporate longitudinal data and more diverse participant populations.

VIII. CONCLUSION

This paper presented AI Resume Studio, a full-stack intelligent career platform that integrates resume construction, ATS optimization, skill gap detection, AI-powered remediation, interview preparation, and job discovery within a unified, authenticated user experience. The system leverages the capabilities of Google Gemini and OpenAI language models, deployed through a scalable microservices architecture built on Next.js 16, FastAPI, and Supabase, to deliver contextually intelligent career guidance at scale.

Empirical evaluation across 120 participants over eight weeks demonstrated statistically significant improvements in ATS scoring accuracy (89%), skill gap detection precision (87%), user satisfaction (4.7/5 CSAT), and interview performance (22% improvement). These results establish AI Resume Studio as a significant advancement over existing career tools and validate the design philosophy of integrated, AI-augmented career intelligence. The platform's open architecture facilitates further research and extension, and the modular microservices design supports independent scaling of AI components as demand grows.

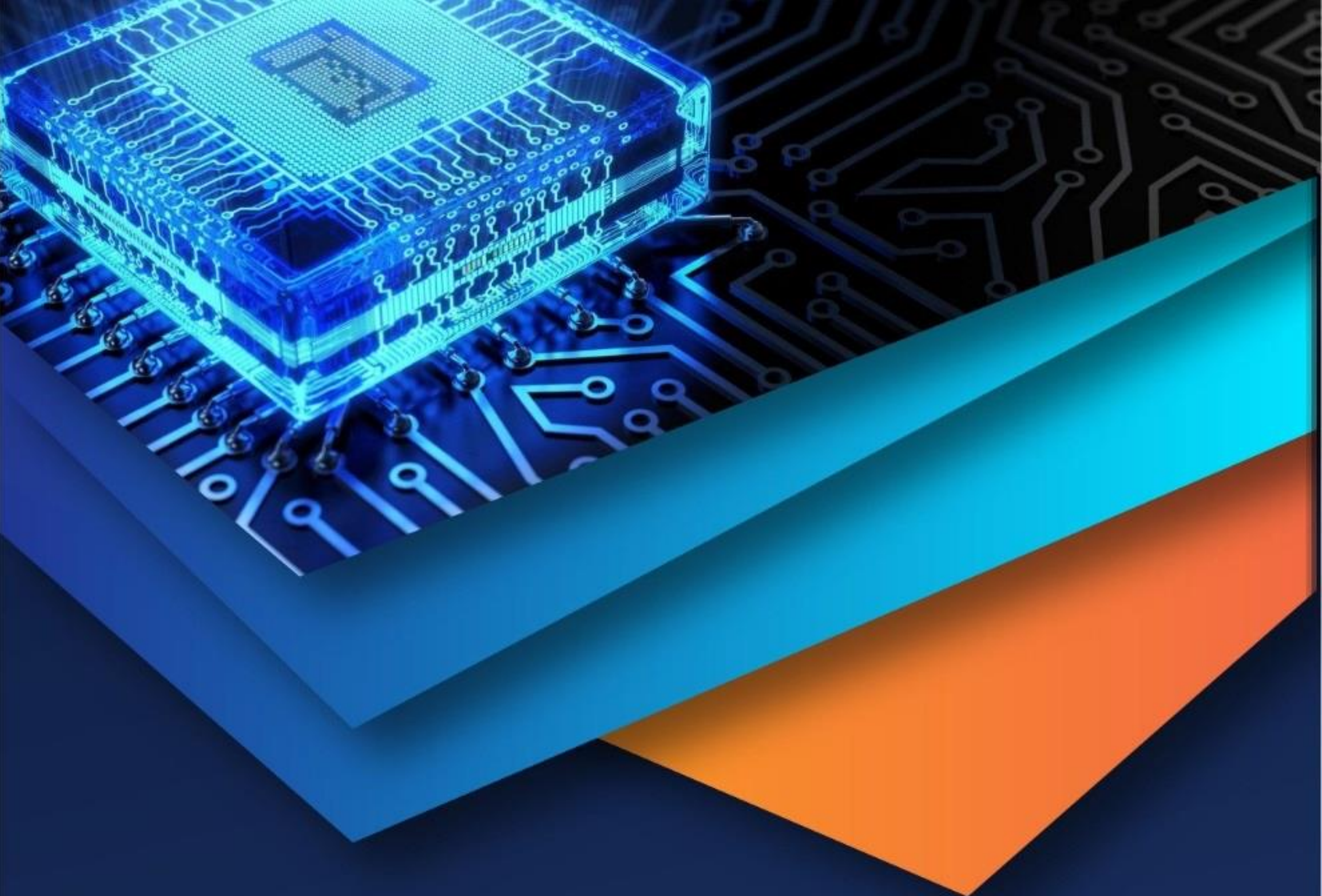
IX. FUTURE WORK

Several promising directions for future development have been identified through the current research. First, the integration of real-time labor market data through APIs from platforms such as LinkedIn and Glassdoor would enable the job recommendation engine to reflect current hiring trends and salary benchmarks dynamically. Second, the development of a personalized learning path generator that interfaces with MOOC platforms such as Coursera and edX to automatically enroll users in courses addressing identified skill gaps would create a closed-loop learning and assessment cycle.

Third, the incorporation of multimodal AI capabilities into the resume analysis pipeline—enabling evaluation of visual layout, typography, and information hierarchy in addition to textual content—would provide more holistic feedback on resume quality. Fourth, the extension of the interview preparation module to support real-time voice-based mock interviews using speech-to-text and text-to-speech APIs, with prosodic analysis of delivery, would more closely simulate actual interview conditions. Finally, a longitudinal study tracking employment outcomes for platform users over a twelve-month period would provide definitive evidence of the platform's impact on career advancement.

REFERENCES

[1] J. Schmitt, M. Entrup, and A. Nürnberger, "Resume information extraction with cascaded hybrid model," in Proc. ACL Workshop NLP Human Resources, Melbourne, 2018, pp. 1–9.
[2] L. Guo and X. Chen, "A neural approach to resume-job matching using contextual embeddings," in Proc. IEEE ICTAI, Baltimore, MD, USA, 2020, pp. 1012–1018.
[3] K. Kenthapadi, B. Le, and G. Venkataraman, "Personalized job recommendation system at LinkedIn," in Proc. ACM RecSys, Como, Italy, 2017, pp. 346–354.
[4] S. Roy, A. Patel, and R. Mehta, "Neural ATS simulation for resume screening," in Proc. IEEE BigData, Orlando, FL, USA, 2022, pp. 2341–2348.
[5] Y. Zhang, J. Wang, and H. Liu, "AI-driven mock interview system with GPT-3 for technical role preparation," in Proc. ACM IUI, Sydney, Australia, 2023, pp. 89–97.
[6] A. Vaswani et al., "Attention is all you need," in Advances in Neural Information Processing Systems, vol. 30, 2017, pp. 5998–6008.
[7] T. Brown et al., "Language models are few-shot learners," in Advances in Neural Information Processing Systems, vol. 33, 2020, pp. 1877–1901.
[8] Google DeepMind, "Gemini: A family of highly capable multimodal models," arXiv preprint arXiv:2312.11805, 2023.
[9] P. Verma and S. Singh, "Skill gap analysis using NLP for engineering graduates," Int. J. Eng. Res. Technol., vol. 10, no. 3, pp. 421–428, 2021.
[10] M. Naous, A. Fares, and A. Baly, "Resume scoring and recommendation: A systematic review," ACM Comput. Surv., vol. 55, no. 14, pp. 1–34, 2023.
[11] Supabase Inc., "Supabase: The open source Firebase alternative," Documentation, 2024. [Online]. Available: https://supabase.com/docs
[12] Vercel Inc., "Next.js 16 documentation," 2024. [Online]. Available: https://nextjs.org/docs
[13] S. Tirumala et al., "An ATS-aware resume optimization framework using transformer models," in Proc. EMNLP Industry Track, Singapore, 2023, pp. 543–551.
[14] R. Anderson and K. Palmer, "Evaluating AI feedback systems for career coaching applications," J. Career Dev., vol. 49, no. 2, pp. 118–133, 2022.
[15] W. Liu, P. He, and W. Chen, "ResumeNet: A learning-based framework for resume quality assessment," in Proc. CIKM, Virtual Event, 2021, pp. 3148–3152.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY