



SkillMatch: An Integrated Platform for AI Resume Analysis, Job Description Matching, and Interview Preparation

Prathamesh Nistane, Shejal Jibhakate, Harsh Gour, Akash Harne

¹ Department of Information Technology, Prof. Ram Meghe Institute of Technology & Research,
Badnera, India, Shejaljibhka27@gmail.com

How to Cite this Article:

Nistane, P., Jibhakate, S., Gour, H. & Harne, A. (2026). SkillMatch: An Integrated Platform for AI Resume Analysis, Job Description Matching, and Interview Preparation. International Journal of Creative and Open Research in Engineering and Management, <i>02</i>(10), 1-9.
<https://doi.org/10.55041/ijcope.v2i10.109>

License:

This article is published under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

© The Author(s). Published by International Journal of Creative and Open Research in Engineering and Management.



<https://doi.org/10.55041/ijcope.v2i10.109>

Abstract— SkillMatch is a full-stack, AI-powered web application that unifies resume analysis, resume enhancement, job-description matching, cover-letter generation, and interview preparation in a single platform. It is designed to help students and job seekers understand exactly why their resumes may be rejected by Applicant Tracking Systems (ATS) and what they can do to improve them. The work addresses several gaps in existing tools: most candidates do not know why their resume fails at the ATS screening stage; existing tools either give an opaque score with no explanation or solve only one part of the problem, such as a resume builder or a job board; interview preparation is typically disconnected from the candidate's actual resume content; and premium tools that offer detailed feedback are expensive and inaccessible to students. The system combines a React frontend, a Node.js/Express backend, and a PostgreSQL database, using deterministic processing for parsing, grading, and score tallying, and generative AI for rubric-based scoring, skill-gap detection, and interview-question generation. The cover-letter module saves job seekers from writer's block by automatically turning their resume and job-match data into a customized, ready-to-send pitch. The implemented platform completes the main workflow from resume submission to analysis, feedback, and career-preparation support, although AI-based scores vary slightly between runs and scanned resumes are not yet supported. Further testing with larger numbers of users and resumes is required to evaluate accuracy and effectiveness.

Keywords— Applicant Tracking System; Resume Analysis; Job Description Matching; Interview Preparation; Generative AI; Cover Letter Generation

I. INTRODUCTION

Finding a job involves more than having the right skills; candidates also need to present their skills effectively. Many companies now use Applicant Tracking Systems (ATS) to screen resumes, where issues such as missing keywords, poor formatting, or unclear skills can reduce a candidate's chances of being shortlisted. However, candidates often receive little feedback on how to improve their resumes.

Job seekers, especially students and fresh graduates, also need to prepare for interviews and create customized cover letters, often using separate tools. To address these challenges, **SkillMatch** is developed as an integrated full-stack application that combines ATS-based resume analysis, personalized interview-question generation, and customized cover-letter creation in one platform. The system uses traditional text-processing techniques such as regular expressions, pdf-parse, and Mammoth, along with *generative AI* for intelligent analysis and personalized feedback. By bringing these features together, SkillMatch aims to make the job-application process simpler, faster, and more accessible.

II. LITERATURE SURVEY

Resume screening has been studied piece by piece for years, and SkillMatch mostly combines those pieces rather than inventing a new one.



Early work like Kopparapu (2010) parsed resumes with hand-written rules and found formatting inconsistency, not the logic, was the main source of error — something we ran into with our own regex parser. Çelik and Elçi (2012) added a skills ontology for semantic matching, but it needed manual upkeep, a cost we avoided.

Roy, Chowdhary and Bhatia (2020) trained classifiers for resume-job recommendation, performing well in-sample but poorly outside their training categories — a key reason we used a general LLM instead of a trained model. Reimers and Gurevych (2019) showed sentence embeddings could compare meaning rather than exact words, which is conceptually what our JD matcher does, except we let the LLM judge similarity inside a prompt rather than computing embeddings ourselves — simpler to build, less reproducible.

Qin et al. (2018) and Bian et al. (2020) learned job-matching from historical hiring outcomes, which needs years of platform data we don't have, so SkillMatch relies on the LLM's general language understanding instead.

Interview-assessment research is more ambitious than our module: Naim et al. (2018) and Hemamou et al. (2019) judged speech, expression and language to predict interviewer ratings. We deliberately scoped our mock interview down to graded multiple-choice, leaving communication judgment to the user.

III. SYSTEM ARCHITECTURE

SkillMatch uses a layered architecture in which the React frontend communicates with the Node.js/Express backend through REST APIs. The backend acts as the main layer, handling authentication, business logic, resume processing, database operations, and AI integration.

Overall Flow: *User → React 18 + Tailwind CSS → REST APIs (Axios) → Node.js + Express.js → PostgreSQL / AI Service → Results*

The frontend provides features such as registration, resume upload, resume analysis, and interview preparation. The backend validates requests, manages authentication using JWT and bcrypt, processes uploaded PDF/DOCX resumes, and communicates with the database and AI service.

For resume analysis, the uploaded file is converted into text and parsed before being analyzed by the AI service. The AI generates resume feedback, skill-gap suggestions, interview questions, mock-interview feedback, self-introductions, and cover letters. PostgreSQL stores user, resume, interview, score, and feedback data.

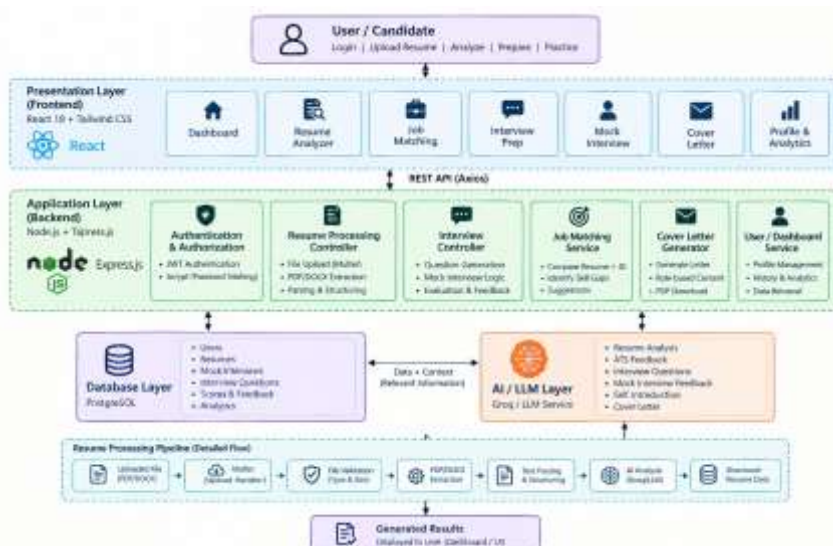


Fig. 1: System Architecture and Flow Diagram of the AI Resume Analyzer & Interview Preparation System

IV. METHODOLOGY

The SkillMatch system combines web development, resume processing, AI-based analysis, and database management to support users in improving their resumes and preparing for interviews. The methodology consists of the following stages:



1.Requirement Analysis: The system requirements are identified based on common job-seeker needs, including resume analysis, ATS feedback, skill-gap detection, job matching, interview preparation, mock interviews, self-introduction, and cover-letter generation.

2.System Design: The system architecture, database, APIs, and component interactions are designed using React, Node.js/Express, PostgreSQL, and AI services.

3.Application Development: React 18 and Tailwind CSS are used for the frontend, while Node.js and Express.js handle APIs, authentication, business logic, file processing, database operations, and AI communication.

4.Resume Processing: Uploaded PDF/DOCX resumes are validated, converted into text, and parsed to extract relevant information such as skills, education, experience, and projects.

Resume → File Validation → Text Extraction → Resume Parsing → Structured Information

5.AI-Based Analysis: The processed resume data is analyzed by the AI layer to provide resume suggestions, skill-gap feedback, interview questions, mock-interview feedback, self-introductions, and cover letters.

6.Data Management: PostgreSQL stores user details, resumes, interview sessions, scores, and feedback, allowing users to access their previous results.

7.Integration and Response: The backend connects the frontend, database, resume-processing modules, and AI service. The processed results are then returned to the frontend and displayed to the user.

Overall Workflow:

Login → Upload Resume → Resume Processing → AI Analysis → ATS & Skill-Gap Feedback → Interview Preparation → Mock Interview → AI Feedback → Self-Introduction/Cover Letter → Dashboard & Results

V. IMPLEMENTATION DETAILS

The SkillMatch system combines deterministic programming techniques with AI-based processing. Conventional application logic is used for tasks such as resume parsing, data storage, scoring, authentication, and security, while AI services are used for tasks that require natural-language understanding and generation.

A. Resume Parsing

Uploaded PDF and DOCX resumes are processed using the pdf-parse and mammoth libraries. The extracted text is processed line by line using regular expressions and rule-based heuristics. Section headers such as Education, Experience, and Projects are identified, while patterns are used to extract information such as email addresses, phone numbers, GPA, and date ranges. This approach reduces unnecessary AI usage and provides consistent parsing results.

B. ATS Scoring

An additive ATS scoring rubric is used to evaluate resume quality. The rubric considers factors such as the presence of a professional summary, quantified achievements, action verbs, education details, and certifications. The AI service applies this predefined rubric to the resume and generates a score along with strengths, weaknesses, and improvement suggestions.

C. AI-Based Fallback Mechanism

AI-dependent features include fallback mechanisms to handle API failures, network errors, rate limits, or invalid responses. When an AI request fails, the system uses the already-parsed resume information and rule-based checks to generate basic results. This improves the reliability of the application and prevents users from being left without a response.

D. Cover Letter Generation

The cover-letter module uses the user's saved resume and optional job information to generate personalized content. Users can provide details such as company name, hiring manager, and job description. When a job description is available, the AI generates a more targeted letter based on the required role and skills. If it is not provided, a general cover letter is generated based on the user's resume and detected role.



E. Dashboard Analytics

The dashboard provides an overview of the user's resume-related data. Skills and detected job roles are collected from the user's resumes and organized using frequency-based analysis. The most frequently identified skills and role distribution are then displayed through charts using the Recharts library.

F. Interview Management and Grading

The interview module generates 20 questions across seven categories using the AI service. The questions and answer data are stored in a PostgreSQL JSONB structure. Once the questions are generated, answers are evaluated through direct comparison with the correct options, eliminating the need for additional AI processing during grading.

G. PDF Export

The enhanced resume can be exported as a PDF directly from the browser using the jsPDF library. A custom layout process manages text wrapping, page margins, vertical positioning, and automatic page breaks to ensure that the generated document remains properly formatted.

H. Job Search

Job postings are retrieved through the **JSearch API via RapidAPI** using a backend proxy. The backend processes the API response and provides a consistent format to the frontend. Application security is implemented using JWT-based authentication and bcrypt password hashing. Google authentication is verified on the server side. Additional security measures include input validation, Helmet security headers, CORS configuration, rate limiting, and resource-ownership checks for protected routes.

VI. CHALLENGES AND LIMITATIONS

- **JSONB Data Handling:** An early issue occurred when Sequelize did not detect changes inside the JSONB interview data. This was resolved by explicitly updating the modified array before saving.
- **Score Variability:** ATS and job-match scores depend on AI-based evaluation, so the same resume may receive slightly different scores across runs. Therefore, these scores should be treated as guidance rather than exact measurements.
- **Resume Parsing Limitations:** The parser works with text-based resumes but may not handle scanned or image-based resumes because OCR is not currently supported. Unusual headings and complex layouts may also affect parsing accuracy.
- **Interview Scoring:** All mock-interview questions currently have equal weight, meaning technical and HR questions contribute equally to the final score.
- **Job Description Transfer:** Text-based job descriptions can be transferred between the Job Matcher and Enhancer, while uploaded JD files cannot be transferred automatically due to browser file-handling limitations.

VII. RESULTS AND DISCUSSION

The developed SkillMatch platform successfully integrates resume analysis, job matching, interview preparation, and career-support features into a single application. The system can process uploaded resumes, extract relevant information, evaluate resume quality, identify skill gaps, and provide AI-generated suggestions for improvement.

The platform also supports interview preparation through question generation and mock-interview feedback. Additional features such as self-introduction and cover-letter generation help users prepare application materials based on their profile.

The developed dashboard brings these results together, allowing users to view their resume performance, detected skills, interview progress, and previous activities in one place (Fig. 2). The combination of rule-based processing and AI-based analysis helps the system provide personalized feedback while keeping structured operations consistent.



Fig. 2: SkillMatch user dashboard showing resume count, average ATS score, detected skills, role distribution, and recent resumes

Overall, the implementation demonstrates that SkillMatch can provide an integrated and practical platform for resume improvement and interview preparation. The current version successfully completes the main workflow from **resume submission to analysis, feedback, and career-preparation support**. However, further testing with a larger number of users and resumes would be required to formally evaluate the accuracy, consistency, and effectiveness of the system.

VIII. CONCLUSION

SkillMatch brings resume analysis, enhancement, cover-letter generation, JD matching and mock interviews into one platform, so feedback from one stage can actually inform the next instead of living in separate disconnected tools. The core design decision throughout was knowing where the LLM belongs and where it doesn't: mechanical tasks like parsing, grading, and score tallying were handled with deterministic code we wrote ourselves, while the Groq-hosted model was reserved for genuine judgment — scoring against our rubric, recognizing equivalent skills worded differently, and generating interview questions. This kept the system fast, cheap, and still usable through fallbacks when the AI service itself failed.

We've tried to be honest about what isn't solved yet — scores aren't perfectly reproducible, scanned resumes don't parse, interview categories are weighted equally, and we haven't audited for bias. The more lasting contribution here isn't a single algorithm but the underlying approach: a small team can build something genuinely useful by being deliberate about which parts of a problem actually need AI, and keeping everything else honest and simple.

REFERENCES

- [1] J. Fuller, M. Raman, E. Sage-Gavin and K. Hines, *Hidden Workers: Untapped Talent*, Harvard Business School Project on Managing the Future of Work and Accenture, Sept. 2021.
- [2] J. Dastin, "Amazon scraps secret AI recruiting tool that showed bias against women," *Reuters*, Oct. 10, 2018.
- [3] M. Raghavan, S. Barocas, J. Kleinberg and K. Levy, "Mitigating bias in algorithmic hiring: Evaluating claims and practices," in *Proc. ACM Conf. on Fairness, Accountability, and Transparency (FAT*)*, 2020.
- [4] L. Hemamou, G. Felhi, V. Vandenbussche, J.-C. Martin and C. Clavel, "HireNet: A hierarchical attention model for the automatic analysis of asynchronous video job interviews," *AAAI Conf. on Artificial Intelligence*, 2019, pp. 573–581.
- [5] M. Bogen and A. Rieke, *Help Wanted: An Examination of Hiring Algorithms, Equity, and Bias*, Upturn, Dec. 2018, 72 pp.