



International Journal of Research Publication and Reviews

Journal Homepage: www.ijrpr.com ISSN 2582-7421

Smart Resume Screening Platform: An Intelligent System for Automated Resume Analysis and Candidate Matching

Saranya, Ajay P, Barath R, Deepak RG, Kavinraj S

Department of Information Technology, Sri Shakthi Institute of Engineering and Technology, Coimbatore, Tamil Nadu, India.

| saranyait@siet.ac.in | ajayajax02@gmail.com | barathramesh26@gmail.com | rgdeepak91@gmail.com | kavinrajsaravanakumar@gmail.com

ABSTRACT

The Smart Resume Screening Platform is a web-based application designed to support recruiters in reviewing job applications and identifying candidates whose skills and experience align with a job description. Manual resume screening can be time-consuming when organizations receive large volumes of applications. Differences in resume formats, inconsistent terminology, and reliance on manual review may also make candidate comparison difficult. The platform provides an interface for uploading resumes, extracting relevant information, comparing candidate profiles with job requirements, and viewing structured screening results. A text-processing module identifies skills and other relevant information from supported resume files. A matching component can calculate a relevance score based on configurable criteria and present the factors contributing to that score. The system is intended to assist human decision-makers rather than independently make hiring decisions. Candidate data should be handled securely, and screening criteria should be transparent, job-related, and reviewed for potential bias. The platform aims to reduce repetitive administrative work, improve consistency in initial screening, and provide a more organized recruitment workflow.

Keywords: Smart Resume Screening; Resume Parsing; Applicant Tracking System; Candidate Matching; Natural Language Processing; Recruitment Automation; Skill Extraction; Job Description Analysis; Resume Ranking; Human-in-the-Loop; Web Application; Data Privacy; Explainable Screening.

1. Introduction

Recruitment teams need to review applications, identify relevant qualifications, and shortlist candidates for available roles. When resumes arrive in different file formats and use different terms to describe similar skills, comparing applicants can require substantial manual effort. Manual screening may also lead to inconsistent evaluation when reviewers interpret requirements differently or must process a large number of applications within a limited period.

The Smart Resume Screening Platform provides a centralized interface for managing job descriptions and submitted resumes. A recruiter can define role requirements, upload resumes, and review extracted information in a structured format. The system compares resume content with role requirements and displays a match score together with matched and missing criteria. This makes the initial review process easier to navigate while preserving the recruiter's responsibility for evaluating candidates.

Objectives

- Provide a centralized interface for uploading and organizing resumes.
- Extract relevant candidate information from supported resume formats.
- Compare resume content with role-specific job requirements.
- Present matching factors, including relevant skills and unmet criteria.
- Reduce repetitive manual work while retaining human oversight.
- Support secure handling of candidate information and consistent screening workflows.

2. Analysis of Existing Methodology

2.1 Manual Resume Screening

In manual screening, recruiters read resumes individually and compare them with the job description. This approach allows contextual judgment, but processing many applications can be slow and reviewers may apply criteria inconsistently. Candidate information may also be recorded in separate spreadsheets or folders, making it harder to track screening progress.

2.2 Basic Keyword-Based Screening

Keyword searches or simple filters can identify resumes containing specified terms. These methods are easy to understand and can quickly narrow a large set of applications. However, exact keyword matching may miss equivalent terms, relevant context, transferable skills, or differently formatted information. It may also overvalue keyword repetition without confirming the depth of a candidate's experience.

2.3 Limitations of Existing Approaches

Manual review can consume significant time when application volumes are high.

Keyword-only filters may overlook synonyms, context, and transferable skills.

Separate files and tools can make candidate status and review notes difficult to track.

An unexplained score can hide why a candidate was ranked highly or poorly.

Poorly selected criteria or historical data can reproduce unfair patterns.

Resume parsing may be inaccurate when documents contain tables, columns, images, or unusual formatting.

3. Research on the Proposed Methodology

The proposed platform brings resume intake, job requirement management, candidate comparison, and screening review into a single web application. Recruiters create or select a job description and define required skills, preferred skills, qualifications, and experience criteria. Resumes are uploaded in supported formats and processed into text. Extracted details are normalized and compared with the selected job criteria. Results are displayed in a structured candidate list with a match score and evidence that can be checked by a recruiter.

3.1 Resume Upload and Text Extraction

The upload module accepts supported resume files and associates each document with the relevant job application. The interface shown in Figure 1 supports PDF resume upload and describes AI-powered parsing and automatic ranking. A parsing component can extract fields such as contact details, education, work experience, projects, certifications, and technical skills. Extracted fields should be displayed for verification because file layout and parsing errors can affect accuracy.

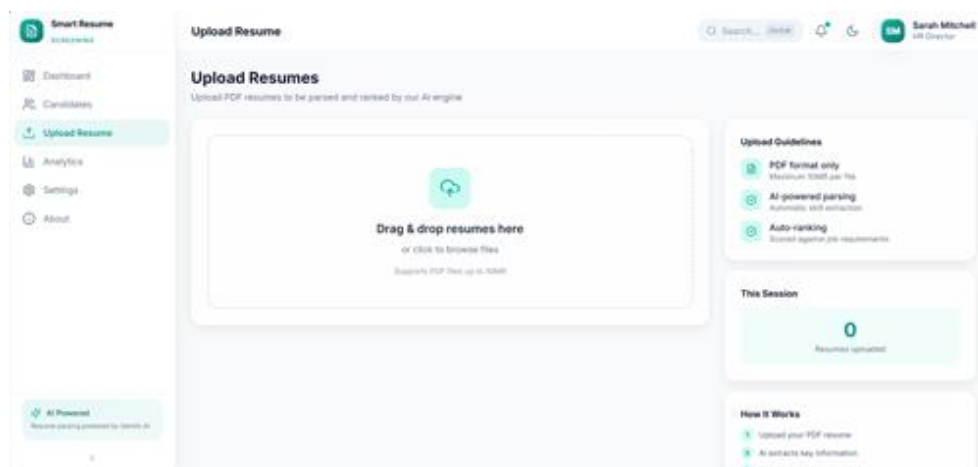


Figure 1. Resume upload interface of the Smart Resume Screening Platform.

3.2 Job Description Analysis and Candidate Matching

The job description module stores role requirements, essential skills, preferred skills, qualifications, and experience expectations. A matching module compares extracted resume content with these requirements. A simple implementation may use weighted skill and qualification matching; an advanced implementation may use natural language processing or semantic similarity to recognize related terminology. The platform can display a score alongside matched skills, missing requirements, and relevant resume excerpts. The score should be treated as a screening aid rather than a validated prediction of job performance.

3.3 Recruiter Dashboard and Candidate Management

The dashboard summarizes the number of resumes received, shortlisted candidates, candidates awaiting review, and rejected applications. Candidate records can display names, email addresses, skills, scores, application dates, and status. Recruiters should be able to inspect the original resume and correct extraction errors before taking action.

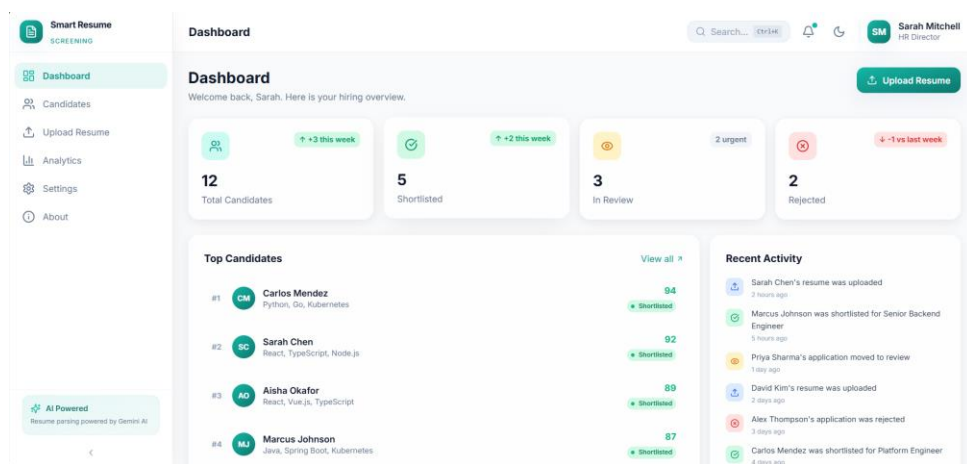


Figure 2. Dashboard showing candidate summary, top candidates, and recent activity.

3.4 Candidate List and Screening Status

The candidate management page organizes applications in a tabular format. Search and status filters help recruiters locate candidates by name, email, skill, or screening stage. A score column provides a quick view of the matching result, while status labels indicate whether a candidate has been shortlisted, is under review, is pending, or has been rejected. Recruiters should use these indicators alongside the resume and job requirements rather than relying on score alone.

The candidate list table displays 12 total candidates. It includes a search bar and filters for All (12), Shortlisted (5), Review (3), Pending (2), and Rejected (2). The table columns are Candidate, Email, Score, Status, and Applied. The data is as follows:

CANDIDATE	EMAIL	SCORE	STATUS	APPLIED
SC Sarah Chen React, TypeScript, Node.js	sarah.chen@email.com	92	Shortlisted	Jul 10, 2026
AO Aisha Okafor React, Vue.js, TypeScript	aisha.okafor@email.com	89	Shortlisted	Jul 6, 2026
MJ Marcus Johnson Java, Spring Boot, Kubernetes	marcus.j@email.com	87	Shortlisted	Jul 9, 2026
PS Priya Sharma Python, Django, React	priya.sharma@email.com	78	In Review	Jul 8, 2026
ER Elena Rodriguez Python, TensorFlow, PyTorch	elena.r@email.com	74	In Review	Jul 4, 2026
DK David Kim Go, Rust, C++	david.kim@email.com	71	Pending	Jul 5, 2026
BO Ryan O'Brien React, JavaScript, Node.js	ryan.obrien@email.com	65	Pending	Jul 3, 2026

Figure 3. Candidate list with search, match scores, and screening status.

3.5 Analytics and Skills Distribution

The analytics page presents recruitment pipeline summaries, average matching score, shortlisted count, candidate status distribution, and skill distribution. These views can help recruiters understand the current applicant pool and identify commonly represented skills. Any displayed values should be based on the application's actual candidate records and should be interpreted according to the dataset used.

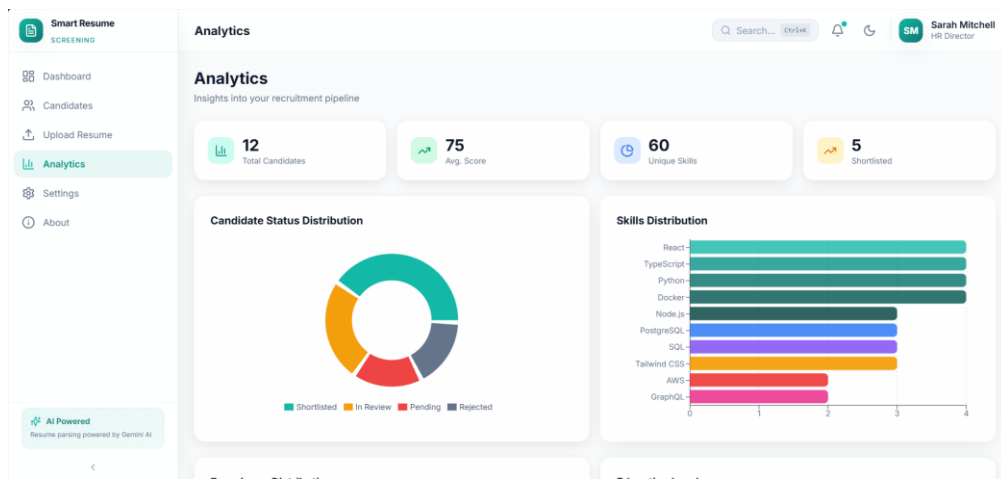


Figure 4. Analytics page showing candidate metrics, status distribution, and skill distribution.

3.6 Proposed System Workflow

The recruiter creates or selects a job description.

The recruiter uploads resumes for the role.

The system validates each file and extracts its text.

The parser identifies candidate information and relevant skills.

The matching module compares extracted details with job criteria.

The platform displays scores and supporting match evidence.

The recruiter reviews the resume and makes the shortlisting decision.

3.7 Security, Privacy, and Responsible Use

Resume files contain personal information. The implementation should use authentication, controlled file storage, secure transport, and a defined retention and deletion policy. Access to resumes and screening results should be limited to authorized users. Screening criteria and score explanations should be available for review, and the system should be tested for parsing errors and unfair differences in outcomes. Protected characteristics should not be used as ranking features.

4. Software and Technologies

The following table presents a suggested stack for a web-based implementation. Confirm and retain only the technologies actually used in the project.

Technology / Component	Purpose in the Platform
React.js	Builds the recruiter dashboard, resume upload interface, and candidate review pages.
Node.js and Express.js	Provides backend APIs for resume uploads, job descriptions, candidate records, and screening workflows.
MongoDB	Stores job descriptions, candidate metadata, screening status, and review records, if used.
Resume parsing / NLP library	Extracts text and identifies skills or other relevant information from supported documents.
Matching module	Compares resume information with job criteria and generates matching indicators.
Authentication and file storage	Controls access and protects resume documents and candidate information.

5. Results and Discussions

This section presents the platform through process and architecture diagrams rather than unsupported numerical claims. The figures summarize the intended workflow and the information available to recruiters. The screenshots elsewhere in the journal show the dashboard, candidate list, resume upload page, and analytics interface.

5.1 End-to-End Screening Workflow

Smart Resume Screening Workflow



Figure 5. End-to-end workflow from job setup to recruiter decision.

The workflow begins when a recruiter defines job-related criteria and uploads a resume. The system extracts available text, identifies relevant information, and compares it with the selected role requirements. The resulting score and supporting criteria are presented for review. The recruiter remains responsible for checking the resume and making the final decision.

5.2 Logical System Architecture

Logical Architecture of the Platform

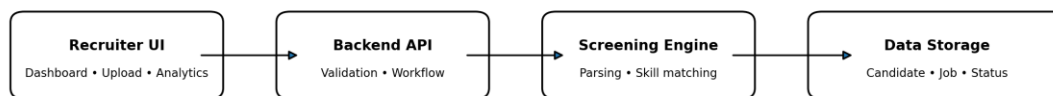


Figure 6. Logical components of the Smart Resume Screening Platform.

The interface communicates with backend services that validate requests and coordinate screening tasks. The screening component performs document parsing and matching, while the data layer stores candidate metadata, job criteria, and application status where supported by the implementation. The exact technologies and deployment details should match the actual project configuration.

5.3 Result Analysis and Recruiter Evidence

Result Analysis: Evidence Presented to the Recruiter

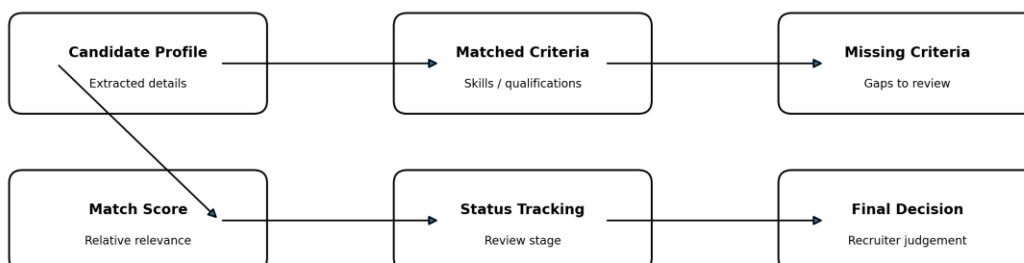


Figure 7. Screening evidence displayed to support recruiter review.

A screening result is more useful when it includes evidence instead of a score alone. Extracted candidate details should be checked for parsing errors; matched criteria show where the resume aligns with the role; and missing criteria identify points requiring review. Status tracking helps organize the recruitment pipeline. These indicators support a structured review but do not establish a candidate's future job performance.

5.4 Evaluation Plan

The current screenshots demonstrate interface elements, but they do not by themselves prove model accuracy or processing performance. A future evaluation should use a labelled set of resumes and job descriptions, then record field-extraction accuracy, skill-match precision and recall, processing time, and recruiter agreement with the ranking. Results should be reported only after running the tests, with the dataset size, criteria, and evaluation procedure documented.

5.5 Limitations

Resume layouts vary and may lead to missing or incorrectly extracted information. Keyword matching can overlook synonyms or context, while a single score may conceal differences between essential and preferred requirements. The system should therefore show understandable evidence, allow corrections, protect personal information, and retain human oversight. Candidate data in the supplied screenshots should be treated as illustrative unless confirmed as real test data.

6. Conclusion and Future Scope

The Smart Resume Screening Platform provides a structured approach to organizing resumes and comparing candidate information with job-specific requirements. By combining document intake, text extraction, criteria management, matching indicators, and recruiter review, the platform can help reduce repetitive administrative effort and make initial screening easier to manage. The usefulness of the system depends on parsing reliability, relevant criteria, and clear explanations. Human review remains essential because resumes may contain incomplete information and automated methods can produce errors.

Future Scope

Future improvements may include support for additional resume formats and languages, semantic skill matching, configurable criteria weights, integration with job portals, recruiter collaboration, interview scheduling, candidate communication, analytics, and audit logs. Further work can investigate bias evaluation, explainability, secure document retention, and accessibility.

References

- [1] National Institute of Standards and Technology, Artificial Intelligence Risk Management Framework (AI RMF 1.0), NIST AI 100-1, 2023.
- [2] National Institute of Standards and Technology, AI Risk Management Framework: Generative Artificial Intelligence Profile, NIST AI 600-1, 2024, if generative AI is used.
- [3] OWASP Foundation, Application Security Verification Standard (ASVS), current version used during implementation.
- [4] MongoDB, "MongoDB Manual," official documentation, <https://www.mongodb.com/docs/>.
- [5] Meta Open Source, "React Documentation," official documentation, <https://react.dev/>.
- [6] OpenJS Foundation, "Node.js Documentation," official documentation, <https://nodejs.org/docs/>.
- [7] Express.js, "Express Web Framework Documentation," official documentation, <https://expressjs.com/>.
- [8] Add the exact documentation or research paper for the resume parsing library used in the implementation.
- [9] Add the exact research papers, datasets, and evaluation references used for the candidate-matching method.

Submission Note

Replace any placeholder references with the exact sources used, confirm the technology stack, and ensure all claims match the implemented application. The candidate names and email addresses in this journal are the supplied author details, not a claim that these people are candidates in the example screenshots. The displayed candidate data in screenshots should be treated as illustrative unless it represents actual test data.