International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Aegis Recruit: AI-Powered Recruitment CRM and Applicant Tracking System

Pradeepkumar R[1], Sachin S[1], Shanmugam S[1], VishnuRam V[1],RajiniGanth R[2]

[1]Department of Computer Science and Engineering, SNS College of Engineering, Coimbatore, India
[2]Assistant Professor, CSE Department, SNS College of Engineering, Coimbatore, India

---

Abstract - Recruitment teams often handle resumes, assessments, interviews, and candidate records across separate tools, creating repeated administrative work and fragmented evidence. Aegis Recruit is an AI-assisted recruitment customer relationship management (CRM) and applicant tracking system (ATS) designed to connect these activities within a structured workflow. The proposed platform combines resume parsing and skill extraction, job-candidate comparison, aptitude testing, coding assessment, AI-assisted technical interviewing, and a final human-led human resources (HR) interview. Candidates must meet the configured 75% threshold in the aptitude and coding rounds to be eligible for the next stage; eligibility is intended to be validated by backend rules rather than by the browser interface alone. The design also includes centralized candidate records, stage tracking, notifications, and consolidated evaluation reports. This paper describes the system scope, workflow, architecture, prototype screens, and responsible-use considerations. The available project artifacts show the intended design and illustrative interfaces, but verified runtime accuracy, security-test results, load performance, and end-to-end test evidence have not been provided. Accordingly, the reported outcomes are design- and prototype-level observations, not claims of measured production performance. A validation plan is outlined for assessing accuracy, reliability, security, usability, and fairness before deployment.

Key Words: AI-assisted recruitment, applicant tracking system, candidate assessment, human oversight, recruitment CRM, resume screening.

I. INTRODUCTION

Artificial intelligence (AI) and cloud-based applications are changing how organizations manage talent acquisition. Recruitment teams may receive large numbers of applications and must compare qualifications, identify job-related skills, coordinate assessments, schedule interviews, and consolidate feedback. When each activity is handled in a separate tool or spreadsheet, information can be duplicated, candidate status can become difficult to trace, and recruiters may spend considerable time on administration rather than evaluation.

Aegis Recruit is proposed as a centralized recruitment CRM and ATS that connects job requisitions, candidate profiles, resume-derived information, assessment attempts, interview evidence, and final HR decisions. Its purpose is not to automate hiring without oversight; rather, it is to organize evidence and provide assistance at selected stages. Resume extraction and AI-generated summaries are intended to support a recruiter's review. Authorized HR personnel retain responsibility for final selection.

The workflow contains four selection rounds: aptitude assessment, coding assessment, AI-assisted technical interview, and human HR interview. The stated qualification rule is a score of at least 75% in the aptitude round to become eligible for coding, and at least 75% in coding to become eligible for the technical interview. These values represent the project's proposed policy. The server must calculate or verify the scores and persist the progression decision; a score shown by a client-side screen is not sufficient evidence of eligibility.

This paper makes three design-level contributions. First, it consolidates the recruitment lifecycle into a single candidate journey. Second, it describes how stage gates, evidence records, and human review can be integrated into a modular architecture. Third, it identifies implementation and evaluation requirements needed to avoid unsupported claims about AI accuracy, security, or production readiness. The work is presented as a system-design case study supported by project documentation and illustrative interface prototypes. Because reproducible runtime measurements were not supplied, the paper distinguishes specified features and prototype outputs from verified operational results.

II. LITERATURE REVIEW

Research on AI-enabled recruiting describes opportunities to assist candidate identification and reduce repetitive administrative effort, while cautioning that successful adoption depends on clear objectives, suitable workflows, and responsible management [1], [2]. Automated tools can help organize candidate information or support initial matching, but their outputs depend on the quality and relevance of the available data. In recruitment, a technically plausible match is not automatically a defensible hiring recommendation; job requirements, the evidence extracted from a resume, and the context of a role must remain visible to reviewers.

Research on algorithmic hiring has documented concerns about bias, explainability, and the validity of claims made for automated screening systems [3]. Systematic review work further shows that algorithmic decision-making in human resources can reproduce discrimination when data, target variables, or evaluation criteria are unsuitable [4]. These findings are especially relevant when a screening score becomes a gate to further assessment. A recruitment platform should therefore retain job-related evidence, expose uncertainty and missing data, and support correction of extraction errors. Performance should be evaluated across representative resume formats and candidate groups rather than summarized by one aggregate score alone.

Ethics-focused research on AI-enabled recruitment emphasizes transparency, accountability, privacy, and meaningful human supervision [5]. These principles affect both the interface and the underlying workflow. Candidates should be told when AI assistance is used and what information is processed; recruiters should understand how a recommendation was formed; and authorized staff should be able to correct errors or override a recommendation with a recorded reason. For interviews, structured questions and consistent evaluation criteria can improve comparability between candidates, provided the rubric measures role-related competencies [6].

Security frameworks provide additional guidance for systems that process personal information and control access to sensitive records. The NIST AI Risk Management Framework emphasizes managing AI-related risks across design, development, deployment, and use [7]. The OWASP Application Security Verification Standard provides a reference for examining application security controls [8]. For Aegis Recruit, these perspectives support a design that separates AI assistance from authoritative business rules, protects candidate documents, records important state changes, and requires testing before operational claims are made.

The design gap addressed here is integration. Resume screening, coding tests, video or text interviews, applicant tracking, and final decision records can exist as distinct functions. Aegis Recruit proposes to connect these functions through a shared candidate record and a defined progression policy. Its intended contribution is workflow continuity and traceability, not a claim that the proposed model or prototype outperforms existing commercial systems. Such a comparative claim would require a controlled evaluation with equivalent tasks, datasets, and performance criteria.

III. METHODOLOGY

A. Study Scope and Design Approach

The work is organized as a design-oriented case study of a recruitment platform. The source materials used for this manuscript describe the problem context, functional workflow, architecture, interface prototypes, limitations, and future validation needs. The analysis maps those artifacts into a common system model: actors and modules are defined first, the four-round workflow is then specified, and the architecture identifies where application rules, AI-assisted services, persistent records, and external integrations belong. This is not a controlled experiment. No new test dataset, recruitment cohort, or measured performance benchmark is claimed.

The design objective is to provide recruiters with one traceable view of candidate progress while keeping high-impact decisions reviewable. The primary actors are company administrators, recruiters, technical interviewers, candidates, and authorized HR decision-makers. Their permissions should be scoped by role and organization. A candidate should be able to access their assigned activities but not other candidates' documents or results; an interviewer should access only the sessions and records authorized for their role; and recruiters should not be able to bypass the server-side progression policy by editing browser state.

B. Functional Model and Candidate Data

The functional model contains several connected modules. Job management stores the position description, required skills, and assessment policy. Candidate management stores applicant details and links the candidate to a job or hiring campaign. Resume ingestion validates a supported file, stores it in a protected location, and invokes parsing to identify available education, experience, projects, skills, and certifications. Extracted fields are presented for human review because document structure, scanned pages, and incomplete information can produce errors. The original resume should remain available to authorized reviewers so that extracted values can be checked against the source.

Job-candidate comparison relates the extracted information to role requirements. A useful result should identify matching evidence and relevant gaps, not provide only an opaque percentage. Assessment management assigns questions, timing, attempts, scoring rules, and status. Coding assessment records source-code submissions and test outcomes when a secure execution service is integrated. Interview management records question prompts, submitted answers, notes, and scorecards. Reporting consolidates the evidence and the current stage, while an audit trail records significant changes to scores, access rights, stage status, and final decisions.



Fig. 1. Proposed system overview of Aegis Recruit.

C. Four-Round Recruitment Workflow

The workflow begins when a recruiter creates a job requisition and imports or registers candidates. Resume analysis supports the initial review, but it does not itself constitute a final hiring decision. Candidates who are invited to the process complete the configured rounds in the following order:

- **Round 1 - Aptitude assessment:** Candidates answer timed, job-relevant objective questions. The system records responses, the attempt status, timestamps, and the calculated score. A score of at least 75% makes the candidate eligible for Round 2 under the project policy.
- **Round 2 - Coding assessment:** Eligible candidates solve two programming problems. When integrated, an isolated execution service checks submissions against configured test cases. Follow-up prompts may ask candidates to explain their approach, complexity, edge cases, or design choices. A score of at least 75% makes the candidate eligible for Round 3.
- **Round 3 - AI-assisted technical interview:** The platform may provide role-related questions and follow-up prompts. Answers may be captured through text, voice, or video according to the implemented service. Any transcript or AI-generated summary is supporting evidence and should be available for reviewer correction.
- **Round 4 - Human HR interview:** An authorized HR interviewer reviews the candidate's profile, assessment evidence, interview notes, and relevant exceptions. HR records the final outcome, such as selected, on hold, or rejected, with an explanation where appropriate.

The two qualification gates are business rules, not visual labels. The backend should calculate or verify the score from persisted assessment data, apply the configured boundary rule consistently, and store the resulting eligibility state. A candidate with an incomplete attempt, invalid submission, or unavailable assessment service should not be silently advanced. The application should use explicit states for in-progress, completed, eligible, not eligible, and review-required outcomes, with retry and recovery behavior defined by the campaign policy.



Fig. 2. Proposed four-round recruitment workflow and qualification gates.

D. System Architecture and Data Flow

The proposed architecture separates the presentation layer, application workflow, AI-assisted processing, persistent data, file management, and external services. The presentation layer provides recruiter, interviewer, and candidate experiences. The application layer validates requests, checks permissions, coordinates candidate and job records, assigns assessments, calculates or verifies scores, applies stage transitions, and produces reports. These responsibilities should remain authoritative on the server so that users cannot alter eligibility by changing client-side state.

The AI service layer may support resume parsing, skill extraction, job matching, question generation, speech transcription, and interview-summary drafting. A replaceable provider interface can isolate the recruitment workflow from a specific model vendor. Before being stored or displayed, AI output should be checked for the expected format and reviewed for job relevance. Where practical, the system should retain enough context—such as the rubric and model or prompt version—to help an authorized reviewer understand the result. A prototype response or mock value must not be represented as a live inference result.

The data layer holds structured records such as organizations, users, jobs, candidates, assessment configurations, attempts, submissions, interviews, scorecards, stage transitions, decisions, and audit events. Resume files and authorized interview artifacts should be stored in protected file storage, with file metadata linked to database records. The server should check authorization before serving documents or reports. Long-running tasks such as bulk resume parsing and report generation may use background jobs with progress reporting, bounded retries, and clear failure states. A code-execution service, if provided, must be isolated from the main application and restricted by resource limits, timeouts, and network controls.



Fig. 3. Proposed system architecture and external-service boundaries.

E. Security, Privacy, and Responsible AI

Recruitment records include personal details, resumes, assessment responses, code submissions, interview content, and employment decisions. Access should be restricted by role and organization, connections should use secure transport, and stored files should not be publicly accessible by default. Input validation is needed for forms, file uploads, assessment submissions, and API calls. Audit events should record important status changes and decision updates with timestamps and authorized-user information. Retention and deletion rules should be defined for resumes, recordings, transcripts, and reports. These requirements should be validated against the selected application-security controls [8].

Candidates should receive clear notice before any interview recording or monitoring begins, including what is collected, why it is collected, how long it is retained, and who can view it. Accessible alternatives should be considered where voice or video interaction creates a barrier. AI-derived matches and interview summaries should be advisory, evidence-linked, and correctable. Facial-expression or emotion inference should not be treated as reliable evidence of job competence. The platform should be evaluated for errors and unintended disparities, and its AI-related risks should be managed throughout the lifecycle rather than only after deployment [5], [7].

IV. RESULTS

A. Prototype and Design Outcomes

The available project artifacts define the main journey across recruiter and candidate functions. The recruiter-facing prototype includes a dashboard, job and candidate navigation, resume analysis, assessment monitoring, interviews, reports, and settings. Candidate-facing activities are represented as a sequence of application, aptitude assessment, coding assessment, and technical interview. These screens communicate the intended module boundaries and the information a recruiter is expected to review; they are illustrative interface prototypes rather than evidence that all backend integrations are operational.

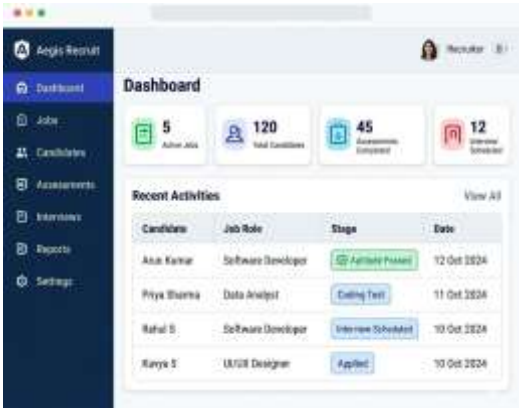


Fig. 4. Illustrative recruiter dashboard prototype.

The resume-analysis screen illustrates how extracted fields can be presented alongside a candidate document and a job-match indicator. In a live system, each extracted item should be traceable to the resume or marked as uncertain, and the recruiter should be able to correct parsing errors. The percentage shown in a mock interface must not be treated as an experimentally validated matching score. Candidate email values in the prototype image have been redacted for privacy.



Fig. 5. Illustrative resume-analysis interface with sample email redacted.

The candidate-evaluation screen brings assessment indicators and a final HR decision into one view. This design is intended to help a reviewer inspect aptitude and coding evidence alongside the technical-interview record before entering a decision. Displayed scores and candidate information are illustrative placeholders; they are not results from a verified participant study or evidence of model accuracy. The final decision remains with authorized human HR personnel.



Fig. 6. Illustrative candidate-evaluation interface with sample email redacted.

At the workflow level, the reported design outcomes are the explicit four-round process, the proposed 75% gates after aptitude and coding, the separation of application and AI-assisted responsibilities, and the intention to consolidate evidence in a candidate report. The documents do not provide verified automated-test reports, execution logs, load-test outputs, or validated resume-matching metrics. Consequently, no test pass rate, extraction accuracy, precision, recall, latency, throughput, uptime, or security assurance is reported as an observed result.

V. DISCUSSION

A. Design Implications

The principal design benefit is continuity across recruitment stages. A centralized candidate record can reduce repeated transfer of information and make the current stage easier to identify, provided that the resume, assessment attempts, interview evidence, and decision remain linked to persistent records. A consistent rubric can improve the comparability of assessments, while a consolidated report can make it easier for authorized reviewers to inspect the evidence behind a recommendation. These benefits are plausible design goals; their magnitude has not yet been measured in the available materials.

The distinction between AI assistance and authority is equally important. Resume parsing can fail on scanned or unusual layouts, while a matching model may overweight terms that appear frequently in a resume but are not sufficient proof of competence. Generated questions may be off-topic or inconsistent in difficulty. Aegis Recruit should therefore expose source evidence, allow corrections, and preserve the criteria used to score candidates. The 75% progression rule must be applied consistently and audited for whether the questions and rubric are appropriate to the role. No single automated score should replace the review of relevant evidence or the final human decision.

The security model is not an optional addition to the workflow. A platform that collects resumes, code, interview responses, and hiring decisions needs authorization checks at the data-access layer, not only protected navigation in the interface. Organization-level isolation is essential in a multi-company system. Coding submissions must be treated as untrusted input and executed only in a suitably isolated service. Interview recording and monitoring need explicit candidate notice, purpose limitation, retention controls, and access records. The project should document which of these controls are implemented and which remain planned.

B. Limitations and Validation Plan

The main limitation is the absence of verified empirical evidence. Prototype images do not demonstrate that authentication, resume parsing, question generation, score calculation, interview capture, file storage, reporting, or notifications work end to end. Likewise, diagrams of external services do not establish that those services are configured, secured, or available. The current paper therefore reports system design and prototype coverage, not production readiness or comparative superiority.

A reproducible evaluation should begin with unit tests for score calculation, pass/fail boundaries, and permitted stage transitions. API integration tests should verify that assessment attempts and decisions are persisted and that direct requests cannot bypass the 75% gates. Role- and organization-based access tests should cover candidate records, files, scorecards, and reports. File-handling tests should examine supported formats, size limits, malformed uploads, and unauthorized downloads. The coding runner should be tested for timeouts, memory limits, network restrictions, hidden-test handling, and containment of untrusted code.

Resume extraction should be evaluated against a consent-aware dataset with manually verified values; appropriate measures may include field-level precision, recall, and error rate. Candidate matching should be reviewed against role-related criteria by trained evaluators, with ranking relevance and agreement reported where appropriate. Reliability tests should specify workload and measure latency, throughput, failure rate, and recovery behavior. Usability evaluation should involve recruiters and candidates using realistic tasks and accessible interaction paths. Fairness review should examine error patterns across representative resume formats and candidate groups. Reports should disclose the test setup, sample size, metrics, failures, and unresolved limitations rather than presenting only favorable outcomes.

Until these checks are completed, Aegis Recruit should be described as a structured prototype and design proposal. This cautious reporting improves technical credibility and gives future implementation work a concrete acceptance checklist.

VI. CONCLUSION AND FUTURE WORK

Aegis Recruit proposes an integrated recruitment CRM and ATS that connects candidate records, resume analysis, aptitude and coding assessments, AI-assisted technical interviewing, and a final human-led HR interview. Its core design contribution is a single traceable workflow with explicit eligibility gates and consolidated candidate evidence. The accompanying prototypes illustrate how recruiters could review candidate information and stage outcomes in a centralized workspace.

The available artifacts establish the intended workflow and interface direction but do not establish verified runtime accuracy, security, scalability, or end-to-end reliability. Future work should implement and test server-side authorization and progression rules, benchmark resume extraction and candidate matching, integrate a secure code runner, validate AI outputs against role-related rubrics, and perform functional, security, usability, fairness, and performance testing. Candidate notices, accessible alternatives, controlled data retention, auditability, and human ownership of final hiring decisions should remain fundamental requirements. Quantitative performance claims should be added only when supported by reproducible evidence.

ACKNOWLEDGMENT

The authors acknowledge the Department of Computer Science and Engineering, SNS College of Engineering, Coimbatore, for supporting the academic project. The authors also thank the project guide for guidance during the design and documentation process.

REFERENCES

[1] A. Black and M. van Esch, “AI-enabled recruiting: What is it and how should a manager use it?” Business Horizons, vol. 63, no. 2, pp. 215–226, 2020, doi: 10.1016/j.bushor.2019.12.001.

[2] P. Tambe, P. Cappelli, and V. Yakubovich, “Artificial intelligence in human resources management: Challenges and a path forward,” California Management Review, vol. 61, no. 4, pp. 15–42, 2019, doi: 10.1177/0008125619867910.

[3] M. Raghavan, S. Barocas, J. Kleinberg, and K. Levy, “Mitigating bias in algorithmic hiring: Evaluating claims and practices,” in Proc. FAccT, 2020, pp. 469–481, doi: 10.1145/3351095.3372828.

[4] A. Köchling and M. C. Wehner, “Discriminated by an algorithm: A systematic review of discrimination and fairness by algorithmic decision-making in HR recruitment and development,” Business Research, vol. 13, pp. 795–848, 2020, doi: 10.1007/s40685-020-00134-w.

[5] A. L. Hunkenschroer and C. Luetge, “Ethics of AI-enabled recruiting and selection: A review and research agenda,” Journal of Business Ethics, vol. 178, pp. 977–1007, 2022, doi: 10.1007/s10551-022-05049-6.

[6] M. A. Campion, D. K. Palmer, and J. E. Campion, “A review of structure in the selection interview,” Personnel Psychology, vol. 50, no. 3, pp. 655–702, 1997, doi: 10.1111/j.1744-6570.1997.tb00709.x.

[7] E. Tabassi, Artificial Intelligence Risk Management Framework (AI RMF 1.0), NIST AI 100-1, National Institute of Standards and Technology, 2023, doi: 10.6028/NIST.AI.100-1.

[8] OWASP Foundation, OWASP Application Security Verification Standard (ASVS), Version 5.0.0, 2025. [Online]. Available: https://asvs.owasp.org.