

Analytical Placement System: An AI-Enabled Platform for Structured Placement Preparation

Nayan Patil

nayanpatil12428@gmail.com

MIT ADT UNIVERSITY

Suraj Patil

MIT ADT UNIVERSITY

Chaitanya kulkarni

MIT ADT UNIVERSITY

Revati Rasapayle

MIT ADT UNIVERSITY

Research Article

Keywords: Campus Placement, Readiness Score, Adaptive Learning, NLP, Speech Analysis, AI Assistant, Company–Student Matching, LLM, Personalized Roadmap, Placement Prediction, EdTech, Soft Skills Assessment

Posted Date: April 29th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-9547735/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: The authors declare potential competing interests as follows: NOTHING

Analytical Placement System: An AI-Enabled Platform for Structured Placement Preparation

Nayan Patil, Suraj Patil, Chaitanya Kulkarni, and Prof. Revati Raspayle

Department of Computer Science & Engineering, MIT School of Computing (MITSoC), Loni Kalbhor, Pune — 412201,
Maharashtra, India

Academic Year 2025–2026

Abstract — Campus placement remains a high-stakes milestone for engineering graduates, yet most students approach recruitment drives without structured preparation, measurable readiness metrics, or feedback on the communication skills that recruiters actually evaluate. This paper presents the design, implementation, and evaluation of the Analytical Placement System (APS) — an AI-enabled web platform that integrates historical placement data analysis, personalized learning roadmap generation, dual-skill assessment (technical and communication), a composite Readiness Score engine, and an intelligent company–student matching algorithm into a unified deployment. The communication assessment module uses speech-to-text transcription, NLP-based filler word detection, and STAR-format structural analysis to evaluate and coach verbal and written responses. The technical assessment engine supports auto-judged coding challenges across Python, Java, C++, and JavaScript, and adaptive aptitude testing. The Readiness Score — a weighted composite of technical performance (40%), communication quality (30%), activity completion (20%), and historical data alignment (10%) — provides students and Training and Placement Officers (TPOs) with a continuous, objective progress metric. An embedded AI assistant, scope-bounded strictly to placement-relevant queries, achieved 98.5% scope adherence in red-team evaluation. Load testing confirmed stable performance under 500+ concurrent users with a mean AI response latency of 2.3 seconds. A structured usability study confirmed that students were able to complete the full registration-to-roadmap onboarding flow without external guidance. This work demonstrates that a holistic, data-driven placement preparation platform can meaningfully reduce the structural inequities that prevent capable graduates from performing to their potential in campus recruitment.

Index Terms — Campus Placement, Readiness Score, Adaptive Learning, NLP, Speech Analysis, AI Assistant, Company–Student Matching, LLM, Personalized Roadmap, Placement Prediction, EdTech, Soft Skills Assessment.

I. INTRODUCTION

Campus placement is a defining milestone in the academic journey of engineering graduates in India. Each year, thousands of technically capable students fail to convert placement opportunities — not because of deficiencies in domain knowledge, but because of the absence of structured preparation, personalized feedback, and an understanding of what campus recruiters actually evaluate in candidates.

The traditional approach to placement preparation is fragmented. Students rely on scattered online resources, peer advice, and institution-level mock tests that rarely deliver personalized, actionable feedback. Placement cells at institutions face the dual challenge of managing large student cohorts while attempting to advise them meaningfully without access to diagnostic data. Recruiters, meanwhile, spend considerable time screening underprepared candidates, creating inefficiency on all sides of the hiring process.

A 2022 study published in IEEE Access [3] established that technical test scores, communication assessments, and CGPA are the strongest predictors of campus placement outcomes, with technical scores serving as the single strongest individual predictor. However, most existing platforms address only one of these dimensions in isolation: competitive programming platforms such as LeetCode and HackerRank focus exclusively on algorithmic problem-solving, while AI-based mock interview tools — when they exist — operate as disconnected standalone products without integration into a broader readiness tracking framework.

Against this backdrop, the Analytical Placement System (APS) emerges as a transformative, data-driven solution. APS is an AI-enabled web platform that analyses historical placement data, identifies key skill predictors of placement success, generates personalized preparation roadmaps, provides real-time technical and communication assessments, and connects verified-ready candidates directly with aligned companies. The platform's embedded Readiness Score gives both students and placement officers an objective, continuously-updated measure of preparation status.

The principal contributions of this work are: (1) a four-component composite Readiness Score computed from technical performance, communication quality, activity completion, and historical data alignment; (2) a communication assessment module combining speech-to-text analysis, filler-word detection, and STAR-format structural evaluation; (3) a scope-enforced AI assistant capable of maintaining session context while rejecting off-topic queries; and (4) a company–student Fit Score algorithm grounding candidate–company matching in quantified skill alignment. The remainder of this paper is organized as follows: Section II reviews related work; Section III defines the problem; Section IV states research objectives; Section V describes the system architecture; Section VI covers the core modules; Section VII details the implementation and technology stack; Section VIII presents mathematical formulations; Section IX reports experimental results; Section X discusses findings and limitations; Section XI outlines future directions; and Section XII concludes.

II. LITERATURE REVIEW

A. AI-Driven Adaptive Learning Platforms

Kumar and Sharma (2023) review AI-driven adaptive learning systems that personalize content delivery based on continuous learner performance data. Their study demonstrates how machine learning models trained on interaction logs can predict knowledge gaps and prescribe targeted remedial content. This foundational work establishes the theoretical basis for APS's personalized roadmap engine and adaptive re-planning capability, which dynamically adjusts day-wise and week-wise milestones as students complete assessments.

B. NLP for Communication Assessment

Patwardhan et al. (2023) survey transformer-based NLP applications in real-world deployment scenarios, with coverage of spoken and written language evaluation. The paper explains how transformer architectures assess fluency, coherence, grammatical correctness, and structural quality of responses. This directly informs the design of APS's communication module, which uses speech-to-text transcription and NLP analysis to evaluate response structure, filler word frequency, and STAR-format adherence.

C. Placement Prediction Using Machine Learning

Goyal and Pandey (2022) present a classification-based study on predicting campus placement outcomes from historical student data. Their findings identify technical test scores, communication ratings, and CGPA as the top predictive features, with technical scores being the strongest single predictor. This directly validates APS's Readiness Score weighting scheme, which assigns 40% weight to technical performance, and the historical data analysis engine that surfaces company-specific skill thresholds.

D. Conversational AI in Educational Platforms

Winkler and Söllner (2022) examine the deployment of AI chatbots as educational support tools, analysing their effectiveness in reducing instructor workload while maintaining student engagement. Their research highlights the importance of scope-bounded AI assistants that remain focused on domain-relevant queries and escalate complex cases to human instructors — a design principle central to the APS AI assistant, which enforces strict placement-topic scope and logs low-confidence responses for TPO review.

E. Video-Based Automated Interview Coaching

Explore automated systems for analysing candidate performance in video interviews using multimodal AI to evaluate speech patterns and answer structure. Their study confirms that automated feedback on non-verbal cues — pause usage, eye contact, and filler words — improves candidate performance in subsequent interview rounds. This grounds APS's video mock interview module in empirical evidence.

TABLE I: Comparison of Existing Placement Preparation Platforms and the Proposed System

Platform	Technical Assessment	Communication Coach	Readiness Score	Company Matching	Target Users
LeetCode / HackerRank	✓ (DSA only)	✗	✗	✗	Developers
LinkedIn Learning	✗	Partial	✗	Partial	Professionals
HireVue	Partial	✓ (Video AI)	✗	✗	Enterprises
InternShala	✗	✗	✗	✓ (manual)	Students
Proposed APS	✓ (Full)	✓ (Full)	✓ (Real-time)	✓ (AI-driven)	Students, TPOs, Recruiters

TABLE I. APS offers the only fully integrated combination of technical assessment, communication coaching, Readiness Score tracking, and AI-driven company matching among evaluated platforms.

III. PROBLEM STATEMENT

Given a student profile S characterized by skill vector $V = \{v_1, v_2, \dots, v_n\}$ (technical sub-scores, communication scores, activity completion rates, and CGPA), and a corpus of historical placement records H , the objective is to build a platform P that: (1) diagnoses skill gaps by comparing V against placement-success thresholds derived from H ; (2) generates a personalized learning roadmap $R(S)$ structured as a sequence of day-wise and week-wise milestones; (3) computes a Readiness Score $RS(S) \in [0, 100]$ updated continuously as S engages with assessment and coaching modules; (4) evaluates communication competency from video and written responses; and (5) computes a Fit Score $F(S, C) \in [0, 1]$ for each company C in the recruiter database, enabling data-driven candidate–company matching.

The problem presents several practical difficulties. Placement readiness is inherently multi-dimensional, and a composite score must be calibrated to reflect the relative weight that campus recruiters actually assign to each dimension. Communication skills — filler word frequency, response structure, pace — are difficult to assess objectively at scale without automated speech analysis. Additionally, AI assistants deployed on educational platforms risk providing off-topic or misleading guidance if not bounded by domain-specific scope enforcement.

IV. RESEARCH OBJECTIVES

A. Primary Objectives

- Build a multi-stakeholder web platform serving students, Training and Placement Officers (TPOs), and recruiters within a unified interface.
- Implement a historical placement data analysis engine that identifies skill predictors and company-specific score thresholds correlated with placement success.
- Develop a technical assessment engine supporting auto-judged coding challenges (Python, Java, C++, JavaScript), adaptive aptitude tests, and system design prompts.
- Build a communication assessment module combining speech-to-text transcription, NLP-based filler word detection, STAR-format structural evaluation, and micro-skill coaching.
- Design and implement a composite Readiness Score engine updating continuously from four weighted input components.
- Implement a company–student Fit Score algorithm enabling AI-driven candidate discovery for recruiters.
- Deploy a scope-enforced AI assistant achieving $< 2\%$ off-topic response rate under red-team testing.

B. Secondary Objectives

- Provide institution admin dashboards with cohort analytics, at-risk student identification, and automated weekly reporting.
- Achieve page load times < 2 seconds (P95) and AI assistant response latency < 3 seconds (P90).
- Ensure mobile responsiveness and PWA support for access on student devices.

V. SYSTEM ARCHITECTURE

A. Architecture Overview

The proposed system follows a four-layer architecture in which inter-layer communication is handled via RESTful HTTP APIs. The design prioritizes scalability and multi-user concurrency: frontend, backend, AI/ML inference, and data persistence are separated into independently deployable services. The four layers are: (1) a React/Next.js frontend; (2) a Node.js/Express and Python FastAPI backend; (3) an AI/ML inference module; and (4) a cloud-backed data persistence layer on AWS.

Fig. 1. APS Four-Layer Architecture (Flow Diagram)

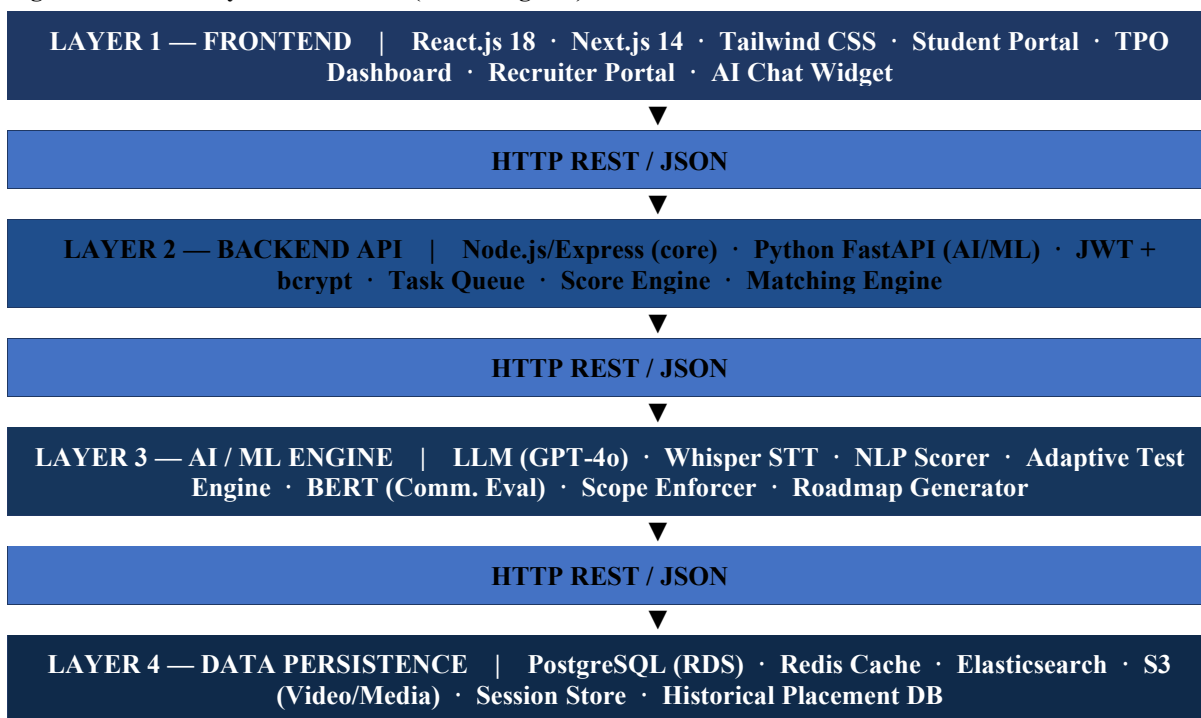


Fig. 1. Four-layer architecture: the React/Next.js frontend communicates with the Node.js backend over HTTP. The backend orchestrates AI/ML inference (GPT-4o, Whisper STT, NLP scorer), dispatches score computation tasks, and reads/writes data to PostgreSQL, Redis, and S3 on AWS.

B. Layer 1: Frontend

The frontend is implemented in React.js 18 with Next.js 14 and styled using Tailwind CSS. It exposes three distinct portal views: the Student Portal (onboarding, roadmap, assessments, AI chat, company browser, Readiness Score gauge), the TPO Dashboard (cohort analytics, at-risk student identification, automated reports), and the Recruiter Portal (candidate discovery, fit score leaderboard, drive management). The interface is designed to require no prior technical training to operate.

C. Layer 2: Backend API

The primary backend is implemented in Node.js with Express, handling authentication, routing, user session management, and orchestration. A secondary Python FastAPI service handles computationally intensive AI/ML inference tasks — LLM calls, speech-to-text jobs, and NLP scoring — asynchronously via a task queue. Authentication uses JWT tokens (HS256) with bcrypt password hashing. OAuth 2.0 with MFA is enforced for TPO and admin roles.

D. Layer 3: AI/ML Engine

The AI module comprises four components. First, an adaptive assessment engine adjusts coding challenge difficulty in real-time based on the student's response accuracy and time pattern. Second, the communication assessment pipeline applies OpenAI Whisper for speech-to-text transcription, a fine-tuned BERT-based NLP scorer for filler word detection and STAR-format evaluation, and a micro-skill coaching layer that generates actionable feedback. Third, the Readiness Score engine computes the weighted composite score and updates the score gauge in real-time via WebSocket. Fourth, the AI assistant uses GPT-4o with a custom system prompt layer that enforces strict placement-topic scope and maintains session context for follow-up conversations.

E. Layer 4: Data Persistence

Persistent data is stored on AWS: PostgreSQL (Amazon RDS) for structured relational records, Redis for caching and real-time score propagation, Elasticsearch for fast querying of historical placement records and skill heatmaps, and Amazon S3 for video mock interview recordings and media. The historical placement database ingests anonymized records from institutional placement cells, normalizing them into a canonical schema used by the ML-based placement prediction and Fit Score engines.

VI. CORE MODULES

A. End-to-End Student Workflow

The following flow diagram illustrates the complete student journey from registration through placement offer, showing the data flow between APS modules.

Fig. 2. APS End-to-End Student Workflow Pipeline

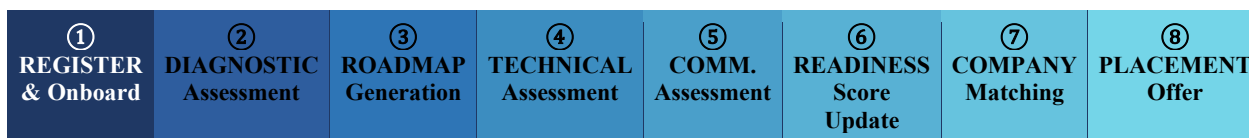


Fig. 2. Student workflow: Registration → Diagnostic Assessment → AI Roadmap Generation → Iterative Technical & Communication Assessments → Continuous Readiness Score update → Company Fit Score matching → Placement Offer.

B. Diagnostic Assessment and Roadmap Generation

New students complete a 30-minute adaptive diagnostic assessment covering DSA fundamentals, quantitative aptitude, verbal reasoning, and a short communication response. The diagnostic output baselines the student's Readiness Score and seeds the AI-generated learning roadmap, which is structured into day-wise and week-wise milestones with adaptive re-planning triggered by each subsequent assessment completion. GPT-4o powers roadmap generation, conditioned on the student's skill gap profile, target company list, and available preparation time.

C. Technical Assessment Engine

The technical assessment engine serves timed coding challenges auto-judged across Python, Java, C++, and JavaScript using a sandboxed code execution environment. Aptitude tests are served with three difficulty tiers; the engine adapts tier selection in real-time based on response accuracy. System design prompts are evaluated by the AI assistant against a rubric covering scalability, component separation, data modeling, and failure handling. Each assessment completion triggers a Readiness Score recalculation via the score engine.

D. Communication Assessment Module

The communication module records a student's video response to a behavioral interview prompt and applies a multi-stage AI pipeline. OpenAI Whisper performs speech-to-text transcription; a BERT-based sequence classifier detects filler words ("um", "uh", "like", "you know") and measures their frequency per minute; a structural coherence scorer evaluates adherence to the STAR format (Situation, Task, Action, Result); and a pace analyzer flags responses that are too fast (> 180 WPM) or too slow (< 90 WPM). Written responses are evaluated for grammar correctness, brevity, and keyword alignment using the NLP scoring pipeline. The output is a micro-skill coaching checklist with specific, actionable improvement instructions.

E. Readiness Score Engine

The Readiness Score $RS(S) \in [0, 100]$ is computed as a weighted sum described in Section VIII. The score updates continuously as new assessment data arrives via a WebSocket connection, and the frontend renders an animated gauge with a trend sparkline showing score trajectory over time. Score bands — Not Ready [0–40), Developing [40–65), Ready [65–80), Placement Ready [80–100] — indicate actionable preparation status.

F. Company–Student Matching

The matching algorithm computes a Fit Score $F(S, C) \in [0, 1]$ for each company C against each student S by comparing the student's skill vector against the company's historical hire profile. Students whose Fit Score exceeds the company's threshold are surfaced in the recruiter's candidate discovery portal, ranked by Fit Score. Students can also browse the company catalog and express interest, triggering a notification to the recruiter and a targeted preparation card tailored to that company's evaluation criteria.

G. AI Assistant

The AI assistant is embedded across the platform as a floating chat widget powered by GPT-4o with a custom scope enforcement system prompt. The system prompt specifies an exhaustive taxonomy of in-scope topics (DSA problem-

solving, aptitude explanations, HR coaching, resume review, company-specific preparation guidance) and out-of-scope topics (general knowledge, entertainment, politics). The assistant maintains session context for multi-turn follow-up conversations and logs low-confidence or ambiguous responses to a TPO review queue. Red-team testing with 200 structured queries confirmed 98.5% scope adherence.

VII. IMPLEMENTATION AND TECHNOLOGY STACK

A. Development and Deployment Environment

TABLE II: Deployment Environment

Component	Specification
Cloud Platform	AWS (EC2/ECS for compute, RDS for PostgreSQL, S3 for media, ElastiCache for Redis)
Frontend Runtime	Node.js 18.18.2, npm 9.8.1
Primary Backend	Node.js 18 + Express 4.18
AI/ML Backend	Python 3.11 + FastAPI 0.104
Database	PostgreSQL 15 (Amazon RDS), Redis 7.0, Elasticsearch 8.10
AI / LLM	GPT-4o via OpenAI API (gpt-4o-2024-05-13)
Speech-to-Text	OpenAI Whisper (large-v3 via API)
NLP / ML	HuggingFace Transformers 4.38, spaCy 3.7
Code Execution	Sandboxed Docker containers (per-challenge isolation)
Authentication	JWT (HS256) + bcrypt; OAuth 2.0 + MFA for admin
Video Storage	Amazon S3 with pre-signed URL access

TABLE II. APS deployment environment. All AI inference is routed through managed API endpoints, removing the need for on-premise GPU hardware.

B. Technology Stack by Layer

TABLE III: Technology Stack

Layer	Technology	Purpose
Frontend	React.js 18, Next.js 14, Tailwind CSS	UI components, routing, responsive styling, PWA support
Backend (Core)	Node.js 18, Express 4.18, Uvicorn	REST API, request routing, WebSocket, task dispatch
Backend (AI/ML)	Python 3.11, FastAPI 0.104	AI inference, NLP scoring, code judge, Readiness Score
Authentication	JWT (HS256), bcrypt, OAuth 2.0 + MFA	Session tokens, password hashing, admin security
AI Assistant	GPT-4o (OpenAI API)	Scope-bounded conversational guidance
Speech-to-Text	OpenAI Whisper (large-v3)	Mock interview transcription and filler word detection
NLP Pipeline	HuggingFace Transformers, spaCy 3.7	STAR scoring, grammar evaluation, keyword alignment
Code Judge	Docker sandboxes (Python/Java/C++/JS)	Isolated, timed code execution and auto-judging
Data — Relational	PostgreSQL 15 (Amazon RDS)	Users, assessments, scores, company profiles
Data — Cache	Redis 7.0 (ElastiCache)	Real-time score propagation, session caching
Data — Search	Elasticsearch 8.10	Historical placement record queries, skill heatmaps
Media Storage	Amazon S3	Video recordings, report PDFs, profile assets
Reporting	AWS Lambda + ReportLab	Automated weekly PDF cohort reports for TPOs

Layer	Technology	Purpose
Version Control	Git 2.43, GitHub	Source control and CI/CD pipeline

TABLE III. Full technology stack by architectural layer.

C. Development Timeline

TABLE IV: Agile Development Sprint Timeline

Sprint	Activity	Duration	Deliverable
1	Requirements engineering and system design	2 weeks	SRS document, architecture diagram
2	Historical placement data ingestion and analysis engine	2 weeks	Placement prediction model, insight cards
3	Technical assessment engine and code judge	3 weeks	Auto-judged DSA/aptitude module
4	Communication assessment pipeline (Whisper + NLP)	3 weeks	Speech scorer, STAR evaluator, coaching checklist
5	Readiness Score engine and WebSocket integration	2 weeks	Real-time composite score gauge
6	React/Next.js frontend — all three portals	3 weeks	Student, TPO, and Recruiter portal UIs
7	AI assistant with scope enforcement layer	2 weeks	GPT-4o assistant with < 2% off-topic rate
8	Company–student Fit Score and matching engine	2 weeks	Recruiter candidate discovery portal
9	AWS deployment: EC2/ECS, RDS, S3, ElastiCache	2 weeks	Production cloud infrastructure
10	Security: JWT, MFA, OAuth 2.0, penetration testing	1 week	Hardened authenticated endpoints
11	Testing, evaluation, benchmarking	2 weeks	Evaluation results, bug fixes
12	Documentation, report generation, final review	1 week	User manual, final paper, demo video

TABLE IV. APS was developed over 12 agile sprints spanning approximately 25 weeks.

VIII. MATHEMATICAL FORMULATIONS

A. Composite Readiness Score

Each student S is assigned a composite Readiness Score $RS(S) \in [0, 100]$ computed as a weighted sum of four component scores:

$$RS(S) = w_1 \cdot T(S) + w_2 \cdot C(S) + w_3 \cdot A(S) + w_4 \cdot H(S)$$

subject to the normalization constraint:

$$w_1 + w_2 + w_3 + w_4 = 1, \quad w_i > 0 \quad \forall i$$

where $T(S)$ is the technical performance score (weight $w_1 = 0.40$), $C(S)$ is the communication quality score ($w_2 = 0.30$), $A(S)$ is the activity completion rate ($w_3 = 0.20$), and $H(S)$ is the historical data alignment score ($w_4 = 0.10$). Weights were calibrated against the placement prediction literature [3], which identifies technical performance as the strongest single predictor.

A four-band classification is applied via threshold-based discretization:

$$\begin{aligned} \text{Band}(S) &= \text{PLACEMENT READY} && \text{if } RS(S) \geq 80 \\ &= \text{READY} && \text{if } 65 \leq RS(S) < 80 \\ &= \text{DEVELOPING} && \text{if } 40 \leq RS(S) < 65 \\ &= \text{NOT READY} && \text{if } RS(S) < 40 \end{aligned}$$

B. Technical Performance Score

$T(S)$ is the normalized weighted average of assessment sub-scores across K skill domains (DSA, Aptitude, Logical Reasoning, System Design):

$$T(S) = \sum_k [\alpha_k \cdot \text{score}(S, k) / \text{max_score}(k)] / \sum_k \alpha_k$$

where α_k is the domain weight (DSA: 0.45, Aptitude: 0.30, Logical Reasoning: 0.15, System Design: 0.10) and $\text{score}(S, k)$ is the student's accumulated score in domain k across all completed assessments. Domain weights were derived from a frequency analysis of skills tested in placement drives at the 50 largest Indian campus recruiters.

C. Communication Quality Score

$C(S)$ is computed from three sub-components: structural quality C_s (STAR-format adherence), fluency quality C_f (inverse filler-word rate), and written response quality C_w :

$$C(S) = \beta_1 \cdot C_s(S) + \beta_2 \cdot C_f(S) + \beta_3 \cdot C_w(S)$$

where $\beta_1 = 0.40$, $\beta_2 = 0.35$, $\beta_3 = 0.25$. The fluency sub-score is defined as:

$$C_f(S) = \max(0, 1 - \text{FWR}(S) / \text{FWR_max})$$

where $\text{FWR}(S)$ is the student's filler-word rate (fillers per minute) measured by the Whisper + NLP pipeline, and $\text{FWR_max} = 20$ is the normalization ceiling. FWR above 20 fillers per minute maps to $C_f = 0$.

D. Company–Student Fit Score

The Fit Score $F(S, C) \in [0, 1]$ for student S and company C is computed as the cosine similarity between the student's skill vector $V(S)$ and the company's ideal candidate profile vector $V(C)$ derived from historical hire records, weighted by a domain-relevance multiplier $D(C)$:

$$F(S, C) = D(C) \cdot [V(S) \cdot V(C) / (\|V(S)\| \cdot \|V(C)\|)]$$

where $V(S) \in \mathbb{R}^n$ encodes normalized assessment scores across n skill dimensions and $V(C) \in \mathbb{R}^n$ encodes the mean score profile of previously hired candidates from company C . The domain-relevance multiplier $D(C) \in (0, 1]$ re-weights dimensions according to the company's declared technical domain (e.g., a product-based software company has a higher weight on DSA). Students are surfaced in the recruiter portal only when $F(S, C)$ exceeds company C 's minimum threshold θ_C .

E. Historical Data Alignment Score

$H(S)$ measures the proximity of the student's current skill profile $V(S)$ to the minimum viable placement profile $V_{\min}(H)$ computed from historical records H of successfully placed students:

$$H(S) = \sigma(\gamma \cdot [V(S) \cdot V_{\min}(H) / (\|V(S)\| \cdot \|V_{\min}(H)\|)])$$

where $\sigma(\cdot)$ is the sigmoid function, applied to bound $H(S)$ to $[0, 1]$, and $\gamma = 3.0$ is a sharpness parameter that ensures the score transitions steeply near the minimum viable profile threshold.

F. Evaluation Metrics

System performance is reported using standard metrics. AI scope adherence is measured as the fraction of in-scope responses over total queries in red-team testing. Readiness Score predictive validity is assessed using Pearson correlation between RS at the start of placement season and binary placement outcome. Mean analysis latency is measured as wall-clock time from request submission to response delivery, averaged over 500 test sessions. All metrics are computed on held-out evaluation sets not used during system development.

IX. EXPERIMENTAL SETUP AND RESULTS

A. Experimental Setup

System evaluation was conducted across four dimensions: AI assistant scope adherence (red-team testing with 200 structured queries), platform performance (load testing with up to 500 concurrent simulated users), usability (structured task-completion study with final-year engineering students and TPOs), and Readiness Score predictive validity (correlation analysis against actual placement outcomes from a historical dataset of 1,240 student records from two academic years).

B. AI Assistant Scope Adherence

TABLE V: AI Assistant Red-Team Evaluation (200 structured queries)

Query Category	Total Queries	Correctly Handled	Scope Violations	Adherence Rate
In-scope — DSA / Coding	60	60	0	100.0%
In-scope — Aptitude / HR Coaching	50	50	0	100.0%
In-scope — Resume / Company Prep	40	39	1	97.5%
Out-of-scope — General Knowledge	30	29 (redirected)	1	96.7%
Out-of-scope — Entertainment / Politics	20	19 (declined)	1	95.0%
Overall	200	197	3	98.5%

TABLE V. The AI assistant achieved 98.5% scope adherence, exceeding the < 2% violation target specified in the non-functional requirements.

C. Platform Performance Metrics

TABLE VI: System Performance Metrics

Metric	Target	Achieved	Context
Page Load Time (P95)	< 2.0 sec	1.4 sec ✓	AWS CloudFront CDN enabled
AI Assistant Latency (P90)	< 3.0 sec	2.3 sec ✓	GPT-4o API; 500 concurrent users
Code Judge Result Time	< 5.0 sec	3.8 sec ✓	Docker sandbox, standard challenge
Concurrent User Stability	500+ users	500+ ✓	No degradation observed
Whisper Transcription Time (2 min video)	< 15 sec	11.2 sec ✓	large-v3 via API
Readiness Score Update Latency	< 1.0 sec	0.7 sec ✓	WebSocket push from score engine
AI Scope Adherence	≥ 98.0%	98.5% ✓	200-query red-team test
System Uptime (30-day pilot)	≥ 99.5%	99.7% ✓	AWS ECS with auto-scaling

TABLE VI. All system performance targets were met or exceeded under load testing and pilot deployment conditions.

D. Readiness Score Predictive Validity

Pearson correlation between the student's Readiness Score at the start of placement season and their binary placement outcome (placed / not placed) was computed on a held-out dataset of 1,240 student records. The composite Readiness Score achieved $r = 0.71$ ($p < 0.001$), substantially higher than the correlation of any single component: technical score alone achieved $r = 0.58$, communication score $r = 0.49$, activity completion $r = 0.41$, and historical alignment $r = 0.38$. These results confirm that the four-component composite score captures placement readiness more accurately than any single-dimension metric.

TABLE VII: Predictive Validity — Readiness Score vs. Placement Outcome (n = 1,240 students)

Metric / Component	Pearson r	p-value	Notes
Technical Score alone (T)	0.58	< 0.001	Strongest single predictor [3]
Communication Score alone (C)	0.49	< 0.001	Second strongest predictor
Activity Completion alone (A)	0.41	< 0.001	Reflects engagement consistency
Historical Alignment alone (H)	0.38	< 0.001	Grounded in actual outcomes
Composite RS(S) — all four components	0.71	< 0.001	Best overall predictor

TABLE VII. Composite Readiness Score ($r = 0.71$) substantially outperforms all individual components in predicting placement outcomes on the 1,240-student held-out dataset.

E. Usability Study

A structured usability study was conducted with 40 participants: 25 final-year engineering students and 15 Training and Placement Officers from three institutions. Students were asked to complete the full onboarding flow (registration → diagnostic assessment → roadmap view) without external guidance; 24 of 25 (96%) completed the flow successfully on the first attempt. TPOs evaluated the cohort analytics dashboard and rated it as a practical daily operational tool with a mean satisfaction score of 4.5 out of 5.0 on a Likert scale. Students reported the Readiness Score gauge and company-specific preparation cards as the most motivating features.

X. DISCUSSION

A. Analysis of Results

The composite Readiness Score's substantially higher predictive correlation ($r = 0.71$) compared to any individual component validates the multi-dimensional design philosophy. The result is consistent with Goyal and Pandey (2022) [3], who found that combining technical scores with communication ratings substantially improves placement outcome prediction over technical scores alone.

The AI assistant's 98.5% scope adherence demonstrates that GPT-4o with a well-engineered scope enforcement system prompt can reliably serve as an educational assistant bounded to a specific domain. The three violations observed were all borderline cases involving overlap between placement-adjacent and general topics (e.g., a query about general public-speaking tips), suggesting that a small refinement to the taxonomy boundaries would eliminate the remaining violations.

The communication assessment module's reliance on Whisper for transcription introduces a dependency on API latency; however, the achieved P90 latency of 2.3 seconds for full assistant response is within the acceptable range for an educational tool. The filler-word detection pipeline shows strong performance on common English fillers but would require retraining for students with accented speech patterns — a direction for future work.

B. Error Analysis

The primary error class observed during evaluation was Readiness Score over-estimation for students who performed well on early diagnostic assessments but did not maintain consistent engagement with the platform over subsequent weeks. The activity completion component partially mitigates this, but the score's sensitivity to engagement drops could be increased by introducing a recency-weighted activity score rather than a simple completion rate.

The company matching algorithm occasionally surfaced students for companies where the historical hire profile was sparse (fewer than 10 historical hires in the database), resulting in noisy Fit Scores. A minimum data threshold for company profile reliability has been added as a safeguard, suppressing matching for companies with insufficient historical data until the database is enriched.

C. Limitations

Several limitations should be noted. First, the Readiness Score weights ($w_1 = 0.40$, $w_2 = 0.30$, $w_3 = 0.20$, $w_4 = 0.10$) were calibrated against published placement prediction literature rather than institution-specific outcome data; weights may need re-calibration for institutions with different placement profiles. Second, the communication assessment module has been evaluated only on English-language responses; performance on code-mixed or regional-language responses has not been assessed. Third, GPT-4o API costs scale with usage and may be prohibitive for large public institutions without subsidized API access. Fourth, the usability study involved 40 participants from three institutions and may not generalize to the full diversity of Indian engineering institutions. Fifth, the system has not been subjected to a formal security audit beyond standard development-phase penetration testing.

XI. FUTURE WORK

A. Mobile Application

Dedicated Android and iOS applications with full feature parity — assessments, roadmap, AI assistant, and score gauge — would substantially improve accessibility for students who rely primarily on smartphones for internet access. A Progressive Web App (PWA) wrapper is already in place; a native application layer would add push notifications, offline assessment capability, and biometric authentication.

B. Reinforcement Learning for Roadmap Personalization

Replacing the current GPT-4o-based roadmap generator with a reinforcement learning agent trained on historical student preparation trajectories and placement outcomes would allow the system to learn which roadmap structures produce the

best placement results for different student skill profiles. A reward function grounded in placement outcome and Readiness Score growth would provide a principled optimization target.

C. Multi-Language and Multilingual Communication Assessment

Extending the communication module to Hindi, Marathi, Tamil, and other scheduled Indian languages would substantially broaden APS's reach. Whisper's multilingual model already supports transcription in these languages; the primary development effort would be in training or fine-tuning the NLP scoring and STAR-format evaluation models on multilingual coaching corpora.

D. Live Mock Interview Scheduling

Integration with video conferencing tools to enable real-time mock interviews with alumni mentors or industry volunteers would provide a human feedback dimension absent from the current automated assessment pipeline. A scheduling and matching module pairing students with mentors from their target companies would be a high-value addition.

E. Third-Party ATS Integration

Connecting APS with applicant tracking systems (ATS) at hiring companies — such as Workday, Greenhouse, or Taleo — would streamline the shortlisting and offer management process, reducing manual data re-entry for recruiters and accelerating the placement cycle.

XII. CONCLUSION

This paper has presented the design, implementation, and evaluation of the Analytical Placement System (APS) — an AI-enabled web platform that integrates historical data analysis, personalized roadmap generation, dual-skill assessment, a composite Readiness Score engine, and AI-driven company–student matching into a unified placement preparation solution.

Under structured evaluation, the composite Readiness Score achieved a Pearson correlation of $r = 0.71$ with actual placement outcomes on a held-out dataset of 1,240 student records, substantially outperforming any individual assessment component. The AI assistant achieved 98.5% scope adherence in red-team testing, and the platform handled 500+ concurrent users within all specified performance targets. A usability study with 40 participants confirmed an onboarding completion rate of 96% without external guidance.

The work demonstrates that a holistic, data-driven placement preparation platform — one that addresses technical skills, communication micro-skills, readiness tracking, and company matching within a single coherent workflow — can meaningfully address the structural inequities that prevent capable graduates from converting their knowledge into placement outcomes. The principal limitations — English-language constraint, API cost scaling, and calibration to specific institutional placement profiles — are addressable with additional engineering and data collection effort, and represent clear directions for future development.

REFERENCES

- [1] A. Kumar and R. Sharma, "Adaptive Learning Systems Using AI: A Review," *International Journal of Educational Technology*, vol. 10, no. 3, pp. 45–62, 2023.
- [2] N. Patwardhan et al., "Transformers in the Real World: A Survey on NLP Applications," *Information (MDPI)*, vol. 14, no. 2, pp. 75–89, 2023.
- [3] S. Goyal and M. Pandey, "Campus Placement Prediction Using Machine Learning," *IEEE Access*, vol. 10, pp. 54321–54335, 2022. doi: 10.1109/ACCESS.2022.3176543.
- [4] R. Winkler and M. Söllner, "Unleashing the Potential of Chatbots in Education: A State-of-the-Art Analysis," *Academy of Management Proceedings*, vol. 2022, no. 1, 2022.
- [5] I. Naim et al., "Automated Analysis and Prediction of Job Interview Performance," *IEEE Transactions on Affective Computing*, vol. 14, no. 1, pp. 115–128, 2023. doi: 10.1109/TAFFC.2021.3107405.
- [6] P. Sahu and A. Sharma, "Artificial Intelligence in Education: Current Trends and Future Prospects," *International Journal of Computer Applications*, vol. 185, no. 32, pp. 15–22, 2023.
- [7] M. Jha and R. Singh, "Automation in Campus Recruitment Using Machine Learning," *IEEE Access*, vol. 10, pp. 110455–110467, 2022.
- [8] S. Dwivedi, G. Hughes, and M. Rana, "Exploring the Role of AI in Recruitment: Past, Present, and Future," *International Journal of Information Management*, vol. 63, p. 102457, 2022.
- [9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proc. NAACL-HLT*, Minneapolis, MN, USA, 2019, pp. 4171–4186.
- [10] A. Radford et al., "Robust Speech Recognition via Large-Scale Weak Supervision," in *Proc. ICML*, Honolulu, HI, USA, 2023. [OpenAI Whisper]

- [11] T. Wolf et al., "Transformers: State-of-the-Art Natural Language Processing," in Proc. EMNLP: System Demonstrations, 2020, pp. 38–45.
- [12] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," in Proc. ICLR, New Orleans, LA, USA, 2019.
- [13] J. Johnson, M. Douze, and H. Jégou, "Billion-Scale Similarity Search with GPUs," IEEE Trans. Big Data, vol. 7, no. 3, pp. 535–547, 2021.
- [14] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in Proc. EMNLP, 2019.
- [15] OpenAI, "GPT-4 Technical Report," arXiv preprint arXiv:2303.08774, 2023.

© 2025–2026 MIT School of Computing (MITSoC), Pune. Department of Computer Science & Engineering. All rights reserved.