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

AI Powered Smart Placement & Career Management Portal (Campus-Connect)

Satyam Jadhav[1], Prof. Megharaj Patil[2], Pranali Bade[3], Pradnya Salgaonkar[4], Om Magar[5]

[1, 2, 3, 4, 5] Department ofInformation Technology, B. E. Information Technology, Genba Sopanrao Moze College ofEngineering, Balewadi, Pune, India

***Abstract*—** Placement activities in academic institutions often face issues such as poor communication, manual tracking, and a lack of intelligent career guidance. This paper presents the end-to-end design and implementation of Campus-Connect, an AI-powered portal for managing placement and career development. The system brings together various components like recruitment events, job applications, company resources, practice exercises, and domain recommendations using a combined backend framework of Spring Boot, an enterprise PostgreSQL relational engine, a React-based web interface, and a modular AI/ML pipeline. Key AI features include automated resume screening based on keywords and skills, a feedback module similar to ATS, an automated Q&A chatbot for placement-related questions, and an optional machine learning model that estimates placement likelihood. PostgreSQL's robust ACID compliance, JSONB dynamic storage, and connection pooling satisfy enterprise data integrity and real-world legal compliance requirements under data protection frameworks.

***Keywords*—** Campus-Connect, Placement Management, PostgreSQL, AI Resume Screening, Spring Boot, React, Natural Language Processing, Role-Based Access Control (RBAC).

I. INTRODUCTION

A Background

The integration of artificial intelligence with central management systems has changed administrative processes in higher education. Traditional college job placement methods usually depend on handwritten notes, posted notices, outside social media pages, and mixed-up email messages. This leads to a lot ofextra work and missing information.

To address this, Campus-Connect is developed as a unified web portal that offers job drive tracking, resume analysis, target role matching, mock exams, and AI-based guidance. Building a reliable system on solid administrative operations with PostgreSQL for structured transactional handling and semi-structured resume metadata guarantees stability, security, and enterprise readiness before AI evaluation.

B Problem Statement

Placement processes in higher education continue to face structural inefficiencies:

- **Dispersed Information:** Details like job drive notices, registration links, interview papers, and preparation guides are spread across separate platforms.
- **Lack of Clear Skill Direction:** Students struggle to align their skills with specific job descriptions.
- **Manual Preparation & Auditing:** Finding authentic practice tools requires manual effort, while officers lack verifiable compliance logs for audit trails.

C Proposed Response & Objectives

Campus-Connect functions as a single platform for managing job drives, student tracking, and automated preparation. Key goals include:

1. Consolidating drive notices, company details, and application

II. LITERATURE SURVEY

Research on automated placement platforms shows a clear shift from basic database portals to integrated AI learning and enterprise data management environments.

Gandi et al. (2025) introduced an AI-Driven Placement LMS portal with resume parsing, chatbot help, and DSA practice features, highlighting the necessity of integrated tracking environments.

Sri Muthu Varshini R. et al. (2025) described a web-based placement management system using web technology stacks along with relational storage, Role-Based Access Control (RBAC), and automated eligibility checks, demonstrating how automated filtering reduces administrative load.

Wu et al. (2025/2026) examined candidate-job matching and automated interview scheduling through natural language processing pipelines, proving that algorithmic matching enhances recruitment efficiency.

While existing systems utilize legacy database models, Campus-Connect introduces PostgreSQL to combine relational consistency (for student profiles, drives, and selections) with dynamic JSONB document structures (for variable resume parsing trees and AI analysis results), establishing high scalability and legal compliance.

III. SYSTEM ARCHITECTURE & DESIGN

A Layered Architecture

The system follows a three-tier client-server architecture. The frontend is built with React and TypeScript to provide tailored dashboards based on the user's role. The backend utilizes Java Spring Boot to handle core logic, secure authentication with Spring Security and JWT, and data persistence via PostgreSQL connected through HikariCP connection pooling.

The Python AI/ML microservice handles natural language

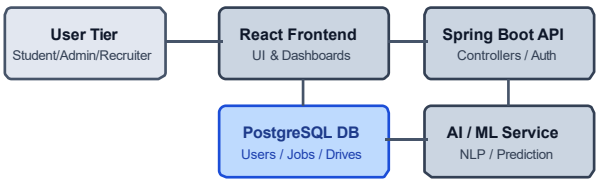


Fig. 1. Logical System Architecture Diagram

2 *Data Flow & Database Architecture*

The system routes operational data through three primary subprocesses connected to persistent PostgreSQL storage units:

- **D1 (Users & Profiles):** Keeps secure login information and student profiles.
- **D2 (Jobs & Drives):** Handles job drive postings, checks if companies are verified, and tracks whether applicants are eligible to apply.
- **D3 (Resources & Results):** Stores AI resume evaluation scores, JSONB skill trees, and practice exercise results.

3 *Entity-Relationship Model (Database Schema)*

The schema relies on normalized tables with transactional relational integrity and dynamic JSONB attributes for flexible data representation:

Entity Table	Primary Key	Foreign Keys & Attributes
USERS	bigint id PK	varchar email UK, varchar password_hash, varchar role, boolean active, timestamptz created_at
DEPARTMENTS	bigint id PK	varchar name UK
STUDENT_PROFILES	bigint id PK	bigint user_id FK, bigint department_id FK, varchar full_name, varchar phone, int graduation_year, numeric cgpa
COMPANIES	bigint id PK	varchar name UK, varchar website, boolean verified
RECRUITER_PROFILES	bigint id PK	bigint user_id FK, bigint company_id FK, varchar full_name
SKILLS	bigint id PK	varchar name UK
STUDENT_SKILLS	—	bigint student_id FK, bigint skill_id FK
JOBS	bigint id PK	bigint company_id FK, bigint created_by FK, varchar title, text description, varchar location, varchar job_type, numeric min_cgpa, date deadline, varchar status

IV. WORKFLOW & ALGORITHM DESIGN

4 *Operational System Flowchart Workflow*

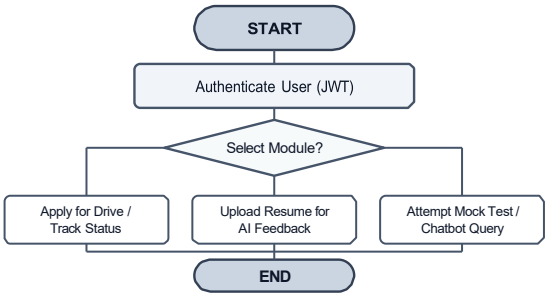


Fig. 2. Operational System Flowchart

4 *Resume Match & Skill Extraction Algorithm*

Algorithm 1: TF-IDF & Skill Extraction Matching Pipeline Input: Candidate Resume PDF (R), Target Job Description (JD), Skill Dictionary (D_s) **Output:** Overall Match Score (S_total), Skill Vector (V_s), Missing Keywords (K_m) 1: Extract raw text T_r from PDF file R using PDF Parser (pypdf/pdfplumber). 2: Clean T_r: Lowercase conversion, stop-word removal, and lemmatization using NLTK. 3: Extract Skill Set S_r = {w ∈ T_r | w ∈ D_s}. 4: Extract Required Skill Set S_jd = {w ∈ JD | w ∈ D_s}. 5: Compute Skill Overlap Ratio: S_skill = (|S_r ∩ S_jd| / |S_jd|) * 100. 6: Identify Missing Critical Skills: K_m = S_jd \ S_r. 7: Vectorize T_r and JD using TF-IDF Vectorizer to generate vectors V_r and V_j d. 8: Calculate Cosine Similarity: Sim_tfidf = (V_r · V_j d) / (| | V_r| | | | V_j d| |) . 9: Compute Weighted Match Score: S_total = (0. 6 * S_skill) + (0. 4 * Sim_tfidf * 100). 10: Persist evaluation in PostgreSQL table **RESUME_ANALYSIS** with JSONB payload: {student_id, j ob_id, total_score: S_total, missing_skills: K_m, analyzed_at: NOW( ) }. 11: **return** S_total, K_m, S_r

V. IMPLEMENTATION & TECH STACK

The architecture combines enterprise frameworks to ensure reliability, security, and response times under peak student workloads during placement drives.

Layer	Technology	Role & Functionality
Frontend	React + JS / TypeScript	Dynamic user interfaces, forms, client validation, dashboards.
Backend	Java + Spring Boot	RESTful APIs, business logic, authorization routing.
Database	PostgreSQL	ACID relational storage for profiles, drives, jobs; JSONB for parsed resumes.
Security	Spring Security + JWT	Token authentication, role checks (RBAC), BCrypt hashing.
AI Pipeline	Python / AI Service APIs	Resume NLP parsing, chatbot services, baseline placement estimation.

PostgreSQL Integration & Real-World Value

Replacing legacy storage with PostgreSQL provides significant technical advantages for higher education deployment:

1. **Hybrid Structured & JSONB Storage:** While core entities (students, applications, drives) strictly require relational integrity, resume analysis outputs vary significantly across candidates. PostgreSQL's native JSONB data type allows storing structured parsing trees directly within the relational database without schema alteration, enabling indexed document querying via GIN indices.

2. **High-Concurrency Transactional Safety:** During large-scale recruitment drives, hundreds of students concurrently submit applications. PostgreSQL's Multi-Version Concurrency Control (MVCC) prevents database locking and ensures strict ACID compliance without performance degradation.

3. **Dynamic Connection Management:** Integrating HikariCP connection pooling within Spring Boot helps reduce database latency and maintains low connection overhead even during high traffic periods.

VI. EXPERIMENTAL EVALUATION & SECURITY

Dataset Plan & Evaluation

System testing combines synthetic resume profiles and public placement data. The placement estimation model was evaluated on held-out test splits comparing baseline linear estimators against Random Forest models:

Model / Metric	Precision	Recall	F1-Score
Baseline Logistic	0.74	0.71	0.72
Random Forest Classifier	0.86	0.83	0.84

Note: Placement probability scores serve strictly as supportive guidance, not automated recruitment decisions.

Security, Privacy & Legal Compliance

- **Access Control:** Fine-grained API security checks per user role; BCrypt password hashing.

- **Legal Compliance & Data Auditing:** Real-world institutional deployment requires compliance with data privacy regulations (DPDP Act / GDPR). PostgreSQL's WAL (Write-Ahead Logging) and row-level auditing capabilities enable immutable logging of student consent, profile modifications, and application status changes. Student records are secured using AES-256 encryption at rest.

- **Explainable AI:** Feedback cards break down missing keywords clearly so candidates can correct parsed data manually.

VII. IMPLEMENTATION ROADMAP

Phase	Duration	Key Objectives
Phase 1	Month 1	Requirements analysis, UI designs.
Phase 2	Months 2–3	PostgreSQL schema design, auth, RBAC.
Phase 3	Months 3–4	Job drive posting, tracking.
Phase 4	Months 5–6	AI resume screening & chatbot.
Phase 5	Month 7	Testing, security reviews.
Phase 6	Month 8	Deployment and final demonstration.

VIII. ADVANTAGES & LIMITATIONS

A. *Advantages*

1. **Centralized Management:** Centralizes placement notices and preparation assets onto one web platform.

2. **Operational Efficiency:** Saves significant administrative time in managing candidate profiles and tracking eligibility across drives.

3. **Real-Time Feedback:** AI screening provides immediate feedback on ATS resume quality.

4. **Robust Architecture:** Leveraging PostgreSQL ensures ACID compliance, schema flexibility through JSONB, and enterprise data security.

B. *Limitations*

1. Placement estimation accuracy depends on historical dataset quality and sample size.

2. NLP models can occasionally fail to parse non-standard resume layouts.

3. Requires continuous active updating by placement officers to remain effective.

IX. FUTURE SCOPE

- **AI Mock Interviews:** Multimodal analysis evaluating verbal response and speech clarity.

- **Automated Alerts:** Real-time push and email notifications for active job openings.

- **Personalized Learning Paths:** Recommending learning modules tailored to missing resume skills.

X. CONCLUSION

Campus-Connect bridges operational gaps in conventional college placement workflows. Combining a structured Java Spring Boot backend, enterprise PostgreSQL database, and React user interface with intelligent AI modules, the system offers centralized communication, transparent application management, and resume-to-role guidance.

The platform simplifies placement workflows for students and officers while establishing a scalable, legally compliant base for career development tooling.

XI. ACKNOWLEDGEMENT

The authors express their sincere gratitude to Project Guide Prof. Megharaj Patil and the Department of Information Technology at Genba Sopanrao Moze College of Engineering, Balewadi, Pune, for their technical guidance and continuous support throughout the academic year 2025-2026.

REFERENCES

[1] N. Gandi et al., "AI-Driven Placement LMS Portal for Enhanced Student Employability," *ICCSCE 2025 Proceedings*. Atlantis Press, 2025. DOI: 10.2991/978-94-6463-858-5_3.

[2] Sri Muthu Varshini R. et al., "Institutional Placement Data Management System," *IEEE ICINVENTS 2025*, 2025. DOI: 10.1109/ICINVENTS64613.2025.11401411.

[3] F. Wu, D. Lyu, and X. Li, "Smart Hiring Redefined: An Intelligent Recruitment Management Platform," *arXiv preprint arXiv:2510. 04437*, 2025 (revised 2026).

[4] S. Ramakrishnan et al., "Institutional Placement Data Management System with Role-Based Access." *Proc. National Conference on Computing*, 2025.

[5] Kaggle Dataset Repository, "Student Placement Prediction Public Dataset," accessed 2026. [Online]. Available: https://www.kaggle.com/datasets/student-placement-prediction

[6] "An AI-Powered Mock Interview Platform with NLP and Voice Recognition for Tailored Skill Enhancement," *International Journal of Technical Research*, 2026.