© JUL 2026 | IRE Journals | Volume 10 Issue 1 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I1-1720166

Suprava: AI-Driven Automation for Campus Recruitment

SULTANA FIRDOUS[1], PREETHI P[2], VARUN KUMAR P[3], RAKESH R[4], VARSHA RANI[5], THAMEEMUNNISA R[6]

[1,2,3,4,5,6]*Department of Computer Applications, Nandi Institute of Management and Science, Affliated to Vijayanagara Sri krishnadevaraya University, Ballari, India*

Abstract- ***College placement processes are often stressful and messy, slowed down by manual paperwork and slow checks. We built Suprava to solve this problem by creating a single system that connects all stakeholders. The platform uses a secure five-tier system tailored for students, placement officers, department heads, corporate recruiters and college executives. To make the process more interactive, we integrated two helpful AI features. First, a friendly chatbot powered by Google Gemini helps students practice, prepare for, and build confidence in their upcoming interviews. Second, an automated resume scanner powered by Llama on Groq helps recruiters instantly match student profiles with open job roles. By moving everything online, Suprava makes college hiring faster, more transparent, and better organized. It simplifies and provides instant status updates and helps students land the right job while making the campus placement process a smooth, fast, and good experience for all involved.***

Keywords - Recruitment system, Placement officer, Candidate tracking, Automated ATS scanner, AI chatbot, College hiring, Workflow automation, Efficiency, Speed.

I. INTRODUCTION

The transition of final-year students into corporate employment is heavily influenced by the efficiency of campus placement operation. Traditional systems use static databases that perform basic CRUD operations, which fail to manage the unstructured text data inherent in student portfolios and in corporate job descriptions. A primary structural bottleneck in traditional pipelines is the separation between structural application screening and the optimization of professional documents. Placement software platforms traditionally focus on monitoring the application status after a submission has cleared the basic boundaries. However, these platforms do not provide students with immediate, practical feedback on how well their resumes align with modern applicant tracking systems (ATS). This information gap often leads to a high volume of incompatible job match submissions, creating extensive processing friction for institutional placement coordinators and corporate acquisition managers. To fundamentally resolve these systematic concerns, this study presents and evaluates Suprava, an intelligent web application ecosystem built on a high-concurrency Flask microframework consolidated into a single logical app.py runtime controller. By integrating Large Language Model (LLM) pipelines, Supravaupgrades the standard campus database into an interactive, context-aware resume screening system.

II. LITERATURE REVIEW

Institutional placement systems have evolved across several key technical phases, shifting from basic databases to automated workflows.

2.1 Foundational Transition to Digital Portals

Early foundational frameworks focused on replacing physical dossier file folders and localized spreadsheets with centralized web portals backed by relational database architectures [2], [3], [10]. These legacy models stabilize primary database entities across basic records. Although they minimize redundant processing steps, they function almost exclusively as passive storage units. They require extensive human effort to manually parse the candidate parameters during the selection cycles.

2.2 Specialization and Access Control Matrix Models

To partition the intensive data verification strain originally placed on individual Training and Placement Officers (TPOs), system engineers introduced role-based separation mechanisms. For example, the CABAL framework [9] allows individual HODs to manage their data by providing them with separate input forms and tracking

© JUL 2026 | IRE Journals | Volume 10 Issue 1 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I1-1720166

dashboards. Although this separation increased the profile integrity at the source, the data remained isolated. Recruiter interface synchronization remained unmanaged, necessitating secondary communication pipelines outside the main portal.

2.3 Automation of Recruitment Workflow Chains

Later systems automated student screening by running rigid, predefined filtering rules against groups of applicants [6], [7], [8]. These background handlers evaluate explicit boundaries based on rigid conditional statements:

$$S_{elig} = \{s \in S \mid CGPA(s) \geq T_{min} \wedge Blog(s) = 0\}$$

Although these systems reduce administrative workloads via automated email alerts, they lack the semantic flexibility to assess complex applicant profiles or to offer constructive qualitative suggestions to students during the preparation phase.

2.4 Monolithic AI Components and Screening Modules

The recent era has introduced modern conversational agents for general student query sorting, alongside independent visual monitoring scripts during aptitude examinations [1], [5]. However, existing academic models deploy AI almost entirely as a screening barrier for institutions to filter out applicants. This leaves an explicit structural gap: AI is rarely used to assist student document alignment prior to a program's application phase.

System Feature Metric	Traditional Portals [2, 3]	Specialized RBAC [9]	AI Modules [1, 5]	Novasphere Architecture
Centralized Normalized Database	Supported	Supported	Supported	**Supported**
Role Partitioning Views	Limited (2 Tiers)	Departmentalized	Segmented	**Full Ecosystem (5 Tiers)**
Generative Resume Building	Not Supported	Not Supported	Not Supported	**Integrated (Gemini Flash)**
Semantic ATS Analyzer Logic	Not Supported	Not Supported	Keyword-Only	**Context Aware (Groq Llama)**
Executive Strategy Forecasting	Not Supported	Not Supported	Limited Modules	**Active Predictive Engine**

JSON context for AI screening

III. SYSTEM ARCHITECTURE

Suprava routes all database operations, core business logic, and API calls through a centralized, single-file controller layout to maintain consistent execution.

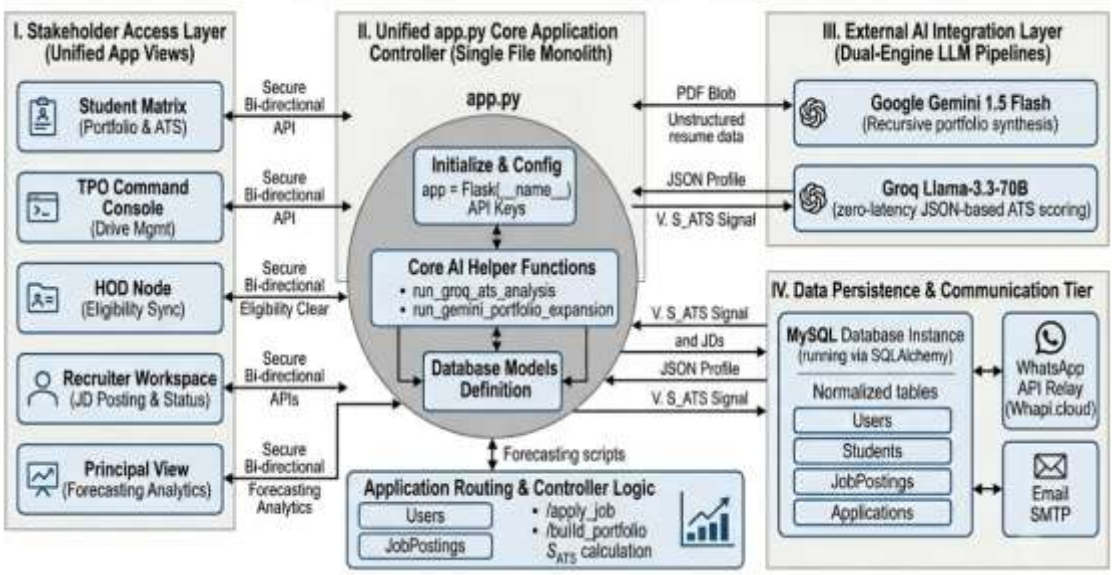
3.1 The 5-Tier Role Access Topology

To keep user sessions secure, the system tracks logins with the JSON Web Tokens (JWT) authentication method and splits access into five specific dashboards:

- Student Matrix: Wraps personal profile curation, automated résumé generation, and real-time ATS compatibility testing against active corporate listings.
- TPO Command: Grants absolute orchestration rights over ongoing recruitment drives, company clearance states, and institutional communication relays.
- HOD Node:Allows real-time checking of student eligibility records and localized whitelist controls to block entry fraud.
- Recruiter Space: Provides approved corporate entities with independent portals to publish job announcements, update candidate stages, and download candidate data.
- Principal View: Restricts operational modifications while exposing multi-

© JUL 2026 | IRE Journals | Volume 10 Issue 1 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I1-1720166

departmental metrics and skill-forecasting trends.



System Architecture

IV. IMPLEMENTATION

When a candidate targets an active job opening, Suprava extracts raw string representations from the candidate's resume. This data is fed inline to a low-latency Groq-hosted Llama-3.3-70B pipeline operating at deterministic sampling parameters (T = 0.0). The engine operates within the context of a strict JSON formatting system.

Using this JSON format prevents syntax errors that typically occur when parsing regular text with regex, allowing the admin controller script to calculate the matching scores and missing keywords cleanly. The score is calculated as follows:

$$S_{ATS} = 0.40(|K_m|/|K_t|) + 0.40S_{sem} + 0.20F_{str}$$

where K_m models matched technical phrases, K_t tracks overall targeted job keyword requirements, S_sem scores semantic alignment, and F_str gauges portfolio document style metrics.

4.1 Recruiter Quarantine and Security Verification Patches

To prevent unauthorized access to institutional data or fake company registrations, this project ensures a strict multi-step authentication processes. Upon initial registration, corporate accounts default to an unapproved state (approved=FALSE). Our Backend route blocks these unverified accounts from logging in until the central TPO explicitly confirms the corporate entity's credentials via their dashboard, updating the state to (approved=TRUE).

V. TECHNICAL IMPLEMENTATION

The backend of the Suprava platform is architected to prioritize responsiveness and scalability. To handle high-volume notification requests without affecting the performance of the main web interface, we utilize an Asynchronous Task Queue pattern.

5.1 Notification Architecture

The implementation relies on Celery as a distributed task queue, powered by Redis as a message broker. When a Training and Placement Officer (TPO) initiates a new job posting, the application flow is as follows:

© JUL 2026 | IRE Journals | Volume 10 Issue 1 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I1-1720166

1. Job Submission: The job data is persisted in the database.
2. Task Queuing: The server dispatches notification tasks to the message broker via the .delay() method, preventing thread blocking on the main web server.
3. Asynchronous Dispatch: Background worker nodes retrieve tasks from the queue and interface with the SMTP protocol for emails and the Whapi Cloud for real-time WhatsApp messaging.

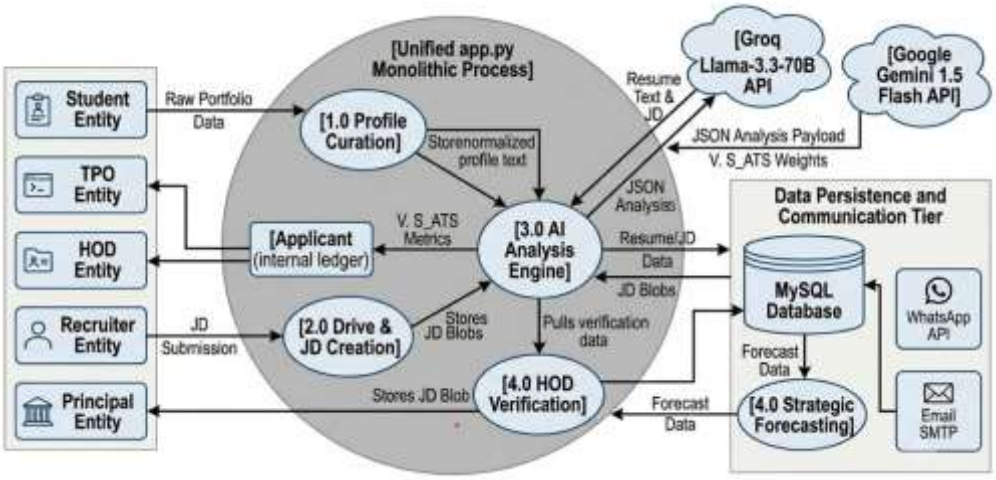
VI. EMPIRICAL RESULTS

The single-file unified system was evaluated through a pilot deployment. The multi-tier validation architecture completely eliminates database identity-forgery. By verifying the input credentials directly against the HOD registration whitelist, the platform prevented the data falsification of student logins. Compared to other platforms, this project ensures a reduction in the manual workload through AI automation. Machine learning algorithms were used to make predictions based on placement trends and to predict trends using a Linear Regression model. They also provided suggestions for improving institutional placement.

Empirical Capability Matrix and Architectural Comparison

Feature	Traditional	RBAC [9]	AI [1, 5]	Suprava
Normalized DB	Supported	Supported	Supported	Supported
Roles	2-Tier	Dept	Segmented	5-Tier
Resume Build	No	No	No	Yes
ATS Logic	No	No	Keyword	Context
Forecasting	No	No	Limited	Advanced

VII. DATA FLOW DIAGRAM



Data Flow Diagram of Suprava

VIII. CONCLUSION

Suprava is an AI-integrated campus placement system that digitizes and streamlines the entire recruitment workflow through a centralized single-file Flask architecture and a five-tier access model. By integrating ATS parsing and a chatbot, we offer real-time context-aware resume feedback and automated verification. Overall, Suprava enhances transparency, security, and efficiency in campus recruitment for all users.

IX. FUTURE ENHANCEMENT

Future enhancements will focus on strengthening communication and intelligence layers. Planned extensions include integrating the Twilio API to send SMS notifications and adding Socket.IO for instant and real-time notifications. Additionally, more advanced machine learning models can be implemented to recommend suitable jobs for students based on their skills and CGPA.

ACKNOWLEDGMENT

We would like to express our sincere gratitude to Prof. Thameemunnisa R for their invaluable mentorship, technical guidance, and constant encouragement throughout the development of this web application.

REFERENCES

[1] Sharma, M. Patel, and R. Kumar, "An AI-Powered Integrated Placement Module with Automated Proctoring and Chatbot-driven Interview Simulation for Higher Education," J. Educ. Technol. Syst., vol. 52, no. 3, pp. 312–329, 2024.

[2] S. Corporate and P. Software, "Training and Placement Web Portal Optimization Guidelines," Tech. Rep. TP-35436, Academic Informatics Association, 2018.

[3] K. R. Prasad and L. V. Reddy, "Design and Execution of Distributed Online Placement Portals," Int. J. Comput. Appl., vol. 89, no. 14, pp. 41–47, 2014.

[4] M. Singh and J. Malhotra, "Reviewing Vulnerabilities within Single Tenant Campus Portal Modules," ProQuest Core Technology Collection, Doc ID 7477cace, 2021.

[5] Shlok Vishwasrao et al., "A Review on Training and Placement Management System," Int. Res. J. Eng. Technol. (IRJET), vol. 09, pp. 544–547, 2022.

[6] Farheen Taqi Rizvi et al., "International Journal For research," IRJASET, vol. 09, 2021. P.

[7] Verma and S. Choksi, "Implementation Matrices of Automated Mail Triggers inside Academic Environments," IRJET, vol. 5, no. 5, pp. 189–194, 2012.

[8] N. Pathak, A. Joshi, and S. Deshpande, "CABAL: Training and Placement Departmental Portal utilizing Multi-Role Verification Patterns," ResearchGate Tech Report, Pub ID 335219239, 2019.

[9] D. H. Miller, "Database Optimization Techniques for Multi-Tenant Institutional Portals," Academic Web Architecture Journal, vol. 12, 2021.