International Journal of Scientific Research and Engineering Development-– Volume 9 Issue 3, May-June 2026
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Towards a Connected Campus: Design and Evaluation of an AI-Augmented Institutional Management Platform

[1]Vinay Babannavar, [2]Preetam Anil Kage, [3]Dilip T, [4]Prasad Reddy

[1,2,3,4](Dept. of Computer Science & Engineering, T. John Institute of Technology, Bangalore – 560083, India
Email: vinaybabannavar@gmail.com, kagepreetam@gmail.com, dilipt122006@gmail.com, prasadreddy.yerrasani@gmail.com

Abstract: **Managing a college is, at its core, an information problem. The institution already knows which students are missing lectures, which ones are slipping in their internal marks, which ones have not submitted assignments in three weeks. The trouble is that this information sits in different places, managed by different people, in formats that were never designed to be combined. Coordinators who want a complete picture of how a student is doing have to collect pieces from multiple sources manually, which takes time they rarely have, and often happens too late for a useful intervention.**

We built the Smart Campus Intelligence System (SCIS) to close this gap. The system unifies five operational areas—the student academic lifecycle, institutional event management, examination administration, campus identity services, and placement support—on a single shared data layer. Three components use machine learning where simpler methods would genuinely underperform: a face-recognition attendance pipeline that removes the roll-call burden from faculty, a grade-trajectory model that surfaces at-risk students before end-semester marks are finalised, and a semantic job-matching engine that ranks candidates against position requirements based on meaning rather than term overlap.

Keywords — **campus management, learning analytics, face-recognition attendance, semantic job matching, sentence transformers, QR authentication, evaluation monitoring, VTU, higher education technology**

I. Motivation and Problem Setting

Consider a straightforward scenario. A second-year student starts the semester normally but, around the eighth week, begins missing morning sessions. By week ten, her attendance in three subjects has fallen below the 75 % threshold required for examination eligibility under VTU rules. Her mid-semester scores in two of those subjects are also dropping. Two consecutive assignment submissions arrived late, and one did not arrive at all.

Taken together, these observations point clearly to someone who is struggling and needs support. But in the administrative reality of most affiliated colleges, no single person holds all these observations at the same time. Attendance records are maintained by individual faculty members. Mid-semester marks reach a central coordinator only after each unit's tabulation is complete. Assignment records live in the faculty member's own email. By the time a pattern becomes visible to someone in a position to help, the semester is usually too far gone for the most effective interventions.

This is not a technology failure in the narrow sense. Each system does what it was designed to do. The attendance software records attendance. The examination portal manages exam schedules. The placement cell spreadsheet tracks placement outcomes. What is missing is any connection between them. The institution's data cannot speak to itself.

SCIS was designed around a direct response to this diagnosis. Rather than adding another portal for another administrative function, we built a shared data layer first and then placed five functional modules on top of it. The modules serve the same operational areas that existing tools cover—admissions, attendance, examinations, campus access, placement—but because they share a single data backbone, information entered through any module is immediately visible to the others.

The decision to include machine learning in specific parts of the platform was deliberate, not reflexive. We applied it only where scale made manual approaches genuinely inadequate. Marking attendance from a camera image is faster and harder to game than a manual roll call when a class has sixty students and the session starts at 8 AM. Identifying which of three hun- dred students is on a failing trajectory requires simultaneously comparing patterns across five subjects that no single faculty member supervises. Matching candidate profiles to job descriptions benefits from understanding what two differently-phrased competencies mean, which term-overlap methods cannot do reliably. For everything else—scheduling logic, result computation, workflow routing, notification delivery—well-designed automation does the job without the additional complexity of a learned model.

Our contributions are:

- A five-module campus management platform sharing a unified relational and document data layer, designed around the operational norms of VTU-affiliated engineering colleges.
- An attendance pipeline using a MobileNetV2 face encoder fine-tuned on enrolled student images, achieving a 1.8 % false positive rate in live classroom conditions.
- A student grade-forecasting model (XGBoost with SHAP attribution) reducing prediction error by 46 % against a lin- ear baseline, with per-student feature explanations to support counsellor follow-up.
- A semantic placement-matching engine (all-MiniLM-L6-v2 sentence transformer) reaching MRR 0.74 and Hit@3 0.81 on a held-out institutional dataset, outperforming BM25 retrieval by 16 and 12 percentage points respectively.
- A 24-hour rotating HMAC-SHA256 QR credential replacing static institutional identity cards.
- Empirical results from a semester-long deployment covering model accuracy, system throughput, usability, and measured administrative workload change across five key tasks.

II. Literature Review

The research literature relevant to SCIS spans five distinct problem areas. We review each in turn, paying attention not just to what prior work achieved but to where it left gaps that shaped the decisions we made in this project.

A. How Campus Administration Software Got Here

Higher education institutions began adopting enterprise software in earnest during the 1990s and early 2000s, largely by adapting ERP packages originally built for manufacturing and logistics [1]. These systems solved a real problem—they gave large universities a single source of truth for student records, fee transactions, and course registrations—but the design assumption underneath all of them was that administrators needed to *record* what happened, not *understand* patterns across what happened. The result was software that was excellent at storing data and poor at doing anything with it analytically.

A second-generation of thinking, which became prominent in the 2010s, pushed for more modular and API-oriented designs that could be adapted more cheaply than monolithic ERP installations [2]. This is the tradition SCIS belongs to. What we added—and what remains uncommon even in the modular literature—is the decision to design the intelligence layer before building the administrative modules, rather than discovering partway through that the data schema did not support the queries the analytical components needed.

B. Predicting Student Outcomes Before It Is Too Late

The learning analytics movement began from a fairly blunt observation: universities were sitting on large quantities of digital records about student behaviour, and they were doing almost nothing with them analytically [3]. The follow-on question—whether these records could predict which students would struggle—attracted a great deal of empirical attention. Work from the mid-2010s showed that models trained on LMS log data (login frequency, resource access patterns, forum activity) could identify academically at-risk students with precision comfortably above chance [4]. Later work incorporated temporal structure, treating a student's semester as a sequence of observations rather than a bag of features, and found that models sensitive to change within a semester outperformed those trained on cumulative averages [5].

We read this literature carefully before settling on XGBoost with SHAP attributions as our forecasting approach. The accuracy gains from recurrent architectures are real but modest given our dataset size. More importantly, the counsellors who act on our model's output are not data scientists; they need to read a flag, understand the reason behind it in thirty seconds, and pick up a phone. A gradient-boosted model with feature attributions supports that workflow. A neural sequence model does not.

C. Taking Attendance Without Taking Up Class Time

Roll calls take time, they require faculty attention at the moment a session begins, and they are easy to game—a student who is not present can still have their name called by a friend. Hardware-based alternatives like RFID card readers [6] removed the faculty burden but simply shifted the fraud method: a card can be handed over just as easily as a voice can answer for someone.

Face recognition addressed the fraud problem by tying the credential to something that cannot be separated from the person carrying it. The accuracy figures reported in controlled-environment studies are encouraging [7], but they are measured under conditions that do not resemble a classroom: consistent frontal lighting, cooperative subjects, one face at a time. Real classrooms have students entering at angles, morning light coming from the wrong direction, and sometimes a group of four people walking through the door simultaneously. We found, during early prototyping, that a pretrained model applied off-the-shelf underperformed our requirements. Fine-tuning on images captured specifically in our classrooms, at the camera positions and lighting conditions that would be used in deployment, was necessary to close that gap.

D. Examination Systems and the Question of Evaluator Behaviour

Research on digital examination systems has covered scheduling algorithms, online delivery logistics, and remote proctoring approaches [8]. Automated scoring of written responses—essays, short answers—has also received sustained attention, and the accuracy of neural scoring models on constrained tasks now approaches human-level agreement [9].

Our examination module borrows from this literature for the administrative components but takes a different position on the evaluation side. Automating the marking of answer scripts in the context of a VTU-affiliated examination is not practically feasible—the regulatory, legal, and institutional trust questions are significant barriers that are unlikely to be resolved soon. What is feasible is monitoring the *process* of evaluation rather than its outputs. A human evaluator who works through forty scripts in twenty minutes is doing something different from one who takes three hours on the same set. Statistical process control, applied to evaluator timing distributions within a cycle, can surface this kind of anomaly for a coordinator to investigate without requiring the system to read or judge a single script.

E. Matching Candidates to Roles Using Meaning, Not Words

Automated support for recruitment has been an active research area since at least the early work on resume parsing and job categorisation [10]. The persistent core difficulty is that the same skill or experience can be described in many ways. A candidate who writes "built regression pipelines for churn prediction" and a job posting that asks for "experience in customer retention analytics using supervised learning" are describing overlapping competencies, but they share no useful terms. Keyword and term-frequency methods fail on pairs like this routinely.

The shift toward pre-trained language models that encode text by meaning rather than by surface form changed what was achievable [11]. Sentence transformers in particular—models trained on large corpora of paired sentences to produce embeddings where semantically similar inputs are geometrically close—have shown strong results on job-candidate matching benchmarks. Our deployment confirmed this on a real institutional dataset: replacing TF-IDF overlap with the all-MiniLM-L6-v2 sentence encoder improved mean reciprocal rank from 0.51 to 0.74 on the same 150 test pairs, a gap large enough to make a practical difference to which candidates a recruiter meets in a first-round interview.

III. System Architecture

A. Design Rationale

SCIS is structured as a set of functionally independent microservices sitting above a shared data layer, exposed through a single React.js web application and a mobile companion app. The separation serves two distinct purposes. Operationally, the five modules have very different load profiles. The gate-entry QR scanner may receive thirty scan events within ninety seconds at the start of a morning session; the course registration interface sees its peak demand spread across several days at the start of a semester. Separating these as services allows each to scale independently without the others being affected. Architecturally, the separation means that a change to one module's logic or schema does not require coordinated modification of unrelated code—a property whose value became evident when the placement cell asked for a schema change to the skill profile structure halfway through development.

A further isolated component hosts the three machine-learning inference pipelines. During load testing, saturating the attendance inference service with simultaneous classroom scans caused its response times to degrade predictably while the administrative services continued operating within bounds. Without isolation, a congested inference queue would have degraded the entire platform uniformly.

B. Technology Stack

Table **??** summarises the principal technology choices by layer.

C. Roles and Permissions

Four roles govern access: Student, Faculty, Administrator, Recruiter. Permission enforcement occurs at the API gateway before any service receives a request. Recruiters can browse shortlisted candidate profiles in a read-only portal but cannot access attendance records or full transcripts. Administrators can read and modify data across modules but cannot alter an examination mark after the evaluation audit trail for that cycle has been sealed. Students access only their own records. QR identity tokens carry an embedded expiry timestamp; a token presented after midnight on the day of issue fails signature verification regardless of whether the QR image is still intact.

IV. Module Descriptions

A. Student Lifecycle and Academic Intelligence

This module spans the period from a student's initial application to their graduation record, designed so that each transition—applying, enrolling, registering courses, submitting work—produces a structured record on the shared data layer without requiring staff to re-enter information that was already collected at a previous stage.

1) Admissions and Enrolment: Applications are received and tracked online. A configurable shortlisting engine ranks candidates by merit criteria set by the admissions office for each intake cycle. Acceptance letters, fee receipts, and enrolment documents are generated from the student record object, not prepared separately.

2) Course Registration: Students select courses for the coming semester through an interface that validates prerequisite completion and credit-load limits against their live transcript in real time, rejecting invalid combinations with an explanation before submission.

3) Attendance Capture: At session start, the system processes a frame from a webcam at the classroom entrance. A MobileNetV2 encoder fine-tuned on enrolment photographs taken under our classroom lighting conditions extracts face embeddings and compares them to the enrolled student gallery by cosine distance. Students above the matching threshold are marked present; unmatched faces create a review flag. Faculty review and correct the record before it is finalised. Across the pilot, the false positive rate was 1.8 % and the false negative rate was 3.4 %. Students whose cumulative attendance in any subject drops within five percentage points of the minimum threshold see an automatic alert on their dashboard.

4) Grade Forecasting: An XGBoost regressor trained on two years of internal assessment history—attendance rates, submission completion, unit-test scores, and mid-semester marks across all five subjects simultaneously—produces end-semester grade predictions at subject level. SHAP attribution values identify the three features contributing most to each student's predicted score, giving a counsellor specific talking points rather than an opaque risk flag. During the pilot, 17 students were referred to counsellors on the basis of model predictions; of these, 11 showed measurable improvement in subsequent internal assessments.

5) Assignment and Credit Tracking: Submission timestamps are recorded at upload; late penalties are applied automatically according to each faculty member's configured policy. A similarity check flags work that closely resembles prior-batch submissions. Credit accumulation is tracked against the applicable programme structure and GPA is computed following VTU credit-weighting rules.

B. Events and Institutional Calendar

Departmental workshops, seminars, cultural competitions, and inter-collegiate events are created and managed here. An organiser defines schedule, capacity, registration deadline, and eligibility criteria once; the system manages registration, waitlisting, reminder notifications, and post-event generation of digitally signed participation certificates. Co-curricular records from this module feed the placement skill profile in Module 5.

C. Examinations and Evaluation Oversight

1) Timetabling and Hall Tickets: Examination scheduling is formulated as a constraint-satisfaction problem. Hard constraints include no student facing simultaneous papers, no faculty member assigned to two venues at the same time, venue capacities not exceeded, and regulatory minimum gaps between consecutive papers for the same cohort. Once a feasible schedule is found, hall tickets are generated as signed PDFs each embedding the student's current QR token, allowing entry verification by phone scan without a database query at the hall.

2) Evaluation Workflow: Evaluators access scripts through a structured interface; every mark entry is time-stamped and linked to the authenticated evaluator session. Entries that fall outside the subject's historical distribution trigger a second-evaluator confirmation before acceptance. All mark modifications write to an audit log.

3) Results and Appeals: Semester results are computed automatically from finalised marks using the applicable credit weightings. Re-evaluation requests are submitted, paid, and tracked through the platform.

4) Evaluator Monitoring: Shewhart control charts are applied to the distribution of time-per-script across all evaluators in a single cycle. An evaluator whose average falls beyond 2.5 standard deviations from the cohort mean in either direction is flagged for coordinator review. The flag initiates a human review process; no mark is altered automatically.

D. Campus Identity and Access Services

1) Dynamic QR Credential: Each student's identity record encodes their university seat number, a biometric enrolment hash reference, current registration status, and institutional affiliation. A 24-hour HMAC-SHA256 signed token is derived from this record daily and displayed as a QR code in the student app. The signing key is held exclusively by the identity service; a photographed token from the previous day fails at the next rotation regardless of how well the image was preserved.

2) Facility Access Logging: Entry-point scan events for libraries and laboratories are written to the shared data layer in real time, providing utilisation data for resource planning without manual counting.

3) Movement Approval: A student needing to leave cam- pus during scheduled hours submits a request through the app specifying destination and expected return. Three sequential approvers—faculty advisor, department head, security post—each receive a notification and respond from their device. Both the student and the gate receive the outcome on final approval.

E. Placement and Career Services

1) Skill Profiles: Students maintain structured profiles listing technical skills, programming experience, certifications, completed courses, internship history, and project repository links. The placement cell configures drive-specific eligibility criteria—minimum CGPA, required skill tags, attendance floors—and the system identifies qualifying students without manual cross-checking.

2) Drive Administration: Company visits are managed end-to-end: JD upload, shortlist generation, pre-placement talk scheduling, interview slot booking, and offer archiving. Recruiters access a read-only portal for shortlisted profiles.

3) Semantic Matching Engine: Student skill profiles and job descriptions are encoded independently as 384-dimensional vectors by the all-MiniLM-L6-v2 sentence transformer. Cosine similarity between these vectors scores the relevance of each student-to-position pair, producing a ranked shortlist for each position and a ranked list of positions for each student. The semantic representation handles vocabulary mismatch that defeats keyword methods. In our dataset, a student who listed "feature engineering, data wrangling, and cross-validation pipelines" as core skills ranked in the top three for a role requiring "proficiency in production ML workflows" despite having no term overlap with the job description.

Table **??** compares the matching engine against two retrieval baselines on a held-out set of 150 student-JD pairs from the 2023–24 placement cycle, annotated by the placement cell.

V. Development Process

We ran two-week Agile sprints throughout, with sprint reviews attended by the faculty or staff who owned each module's domain. This made a practical difference. The examination office revised the anomaly-detection sensitivity threshold across three consecutive reviews before settling on a value that generated alerts at a rate their coordinators could absorb. The placement cell restructured the skill-profile schema mid-sprint when it emerged that the original design could not represent MOOC certifications with the specificity they needed. Both changes were isolated to a single service; in a monolith they would have required coordinated edits across the entire codebase.

GitHub Actions ran unit tests, API integration tests, and inter-service contract tests on every pull request before merge was permitted. Three categories of defect were caught before user-acceptance testing: a race condition in the concurrent QR scan endpoint that appeared only under burst load, an off-by-one error in VTU credit-weighting computation for a specific combination of elective credits, and a pagination boundary bug in the placement candidate listing that returned an extra record on the final page of large result sets.

VI. Evaluation

A. Pilot Scope

The pilot ran for one complete academic semester at T. John Institute of Technology. Participants included 320 undergraduate students from the Computer Science programme, 28 teaching staff, and six administrative officers. All five modules were active throughout. The examination scheduling and evaluation features were fully exercised only during the end-semester window, which required advance coordination with the university examination board.

B. Predictive Model Accuracy

The grade forecasting model produced an RMSE of 4.2 marks on a 100-point scale against actual end-semester results, compared with 7.8 marks for a ridge regression baseline trained on the same features — a 46 % reduction in prediction error. Most predictions fell within one letter-grade band of the final result, the resolution required for early-intervention decisions. Of the 17 students referred to counsellors on the basis of model flags, 11 showed improvement in subsequent internal assessments.

The evaluation anomaly detector was validated against synthetic timing irregularities injected into the evaluation records during a pre-deployment exercise. Detection sensitivity was 91 %. Missed cases clustered near the 2.5σ boundary, indicat- ing that a per-cycle adjustable threshold would outperform the current global setting.

C. System Throughput

Load testing simulated 500 concurrent users performing a realistic distribution of the most common operations. Mean API response time was 187 ms; the 95th-percentile response was 341 ms. The QR verification endpoint, which faces the steepest burst demand during morning campus entry, sustained 42 verifications per second without measurable degradation in the other services.

D. Usability

Forty-eight participants (32 students, 16 faculty) completed a System Usability Scale questionnaire at the pilot's conclusion. The mean score was 78.4 (range: 62–92), placing the platform in the "Good" adjective band [12]. Faculty responses were highest for the grade-risk dashboard and the evaluation monitoring view. Students rated the placement match recommendations and the movement approval workflow as the features with the most practical effect on their daily experience.

Two features consistently scored below average. The library access history display was described by multiple students as slow to scan visually; the information is accurate but the layout does not support fast reading. The QR scanning setup process required repeated guidance for gate security staff during the first two weeks. Both issues are queued for the next release.

E. Administrative Workload

Table **??** compares five administrative tasks before and after deployment. The before figures were collected by asking responsible staff to estimate time spent on each task during the equivalent period of the previous academic year. The after figures were drawn from activity logs and staff time reports gathered during the second half of the pilot semester, once familiarity with the platform had stabilised.

The largest savings in absolute time were in hall-ticket preparation and attendance consolidation, both driven by the elimination of manual eligibility cross-checking and register reconciliation respectively. A post-drive survey administered to 14 visiting recruiters asked what fraction of first-round interviews they had conducted with candidates they considered genuinely relevant to the role. Under prior manual shortlisting the figure was 61 %. Under SCIS it was 82 %.

VII. Limitations

Three limitations are worth naming directly.

The face-recognition component was calibrated for the camera hardware and lighting conditions in our classrooms. Its accuracy in an institution with different physical infrastructure has not been tested. Face coverings and non-frontal approach angles reduce recognition confidence; the next release will introduce a secondary QR check for frames below a confidence threshold.

The grade-forecasting model was trained and evaluated within a single department that uses a particular mix of assessment types. A department relying more heavily on project-based evaluation than unit tests may see different prediction behaviour. Cross-department validation is planned for the next academic year.

The pilot involved faculty who volunteered before the platform was institutionally mandated. Adoption metrics under mandatory rollout — particularly for features that alter established evaluation workflows — may look different.

VIII. Conclusion

The argument behind SCIS is not that Indian colleges lack data about their students. They have plenty of it. The problem is structural: the data is held in places that do not communicate, by administrative units that were never asked to share it, in formats that cannot be combined without someone doing it manually. Our contribution is a platform that addresses this at the architectural level by building the shared data layer first and placing functional modules on top of it, rather than integrating existing standalone tools after the fact.

The semester-long pilot at our home institution validated the approach across five dimensions: model accuracy for grade prediction and placement matching, system throughput under realistic concurrent load, user-perceived usability, and measured administrative workload. Each dimension produced results that meet the threshold for the system to be genuinely useful rather than merely functional.

The most important direction for future development is extending early-warning granularity from the semester level to the unit level, so that an intervention signal arrives while there is still time within the current instructional period to act on it. We also plan to investigate federated learning across institutions as a way to improve shared model quality without requiring any institution to expose individual student records. Connecting the platform to the Government of India's Academic Bank of Credits infrastructure would allow credit transfers and cross-institutional registrations to be handled through SCIS rather than through paper-based equivalency procedures that currently take weeks.

Acknowledgment

We thank the administration, faculty, students, and Training and Placement Cell of T. John Institute of Technology, Bangalore. Sprint reviews conducted with faculty coordinators from the examination office and placement cell shaped the platform in ways that would not have emerged from requirements gathering alone. We are particularly grateful to the staff who explained VTU evaluation procedures in enough operational detail that we could automate parts of them without making consequential mistakes.

References

[1] J. Pollock, "Enterprise resource planning in higher education: Lessons from early adopters," *Campus Technology*, vol. 14, no. 6, pp. 32–38, 2003.

[2] M. A. Kuhail, N. Alturki, S. Alramlawi, and K. Alhejori, "Interacting with educational chatbots: A systematic review," *Education and Information Technologies*, vol. 28, no. 2, pp. 973–1018, 2023.

[3] G. Siemens and P. Long, "Penetrating the fog: Analytics in learning and education," *EDUCAUSE Review*, vol. 46, no. 5, pp. 30–32, 2011.

[4] S. M. Jayaprakash, E. W. Moody, E. J. M. Lauria, J. R. Regan, and J. D. Baron, "Early alert of academically at-risk students: An open source analytics initiative," *Journal of Learning Analytics*, vol. 1, no. 1, pp. 6–47, 2014.

[5] S. Hussain, N. M. Gao, and M. Hussain, "Using machine learning to predict student difficulties from learning session data," *Artificial Intelligence Review*, vol. 54, pp. 6075–6113, 2021.

[6] R. Bhatt, C. Bhatt, and H. Patel, "RFID based attendance management system," *International Journal of Computer Applications*, vol. 59, no. 14, pp. 28–32, 2012.

[7] P. Gogoi, A. Bhattacharyya, B. Bhuyan, and D. K. Bora, "Deep convolutional transfer learning for multi-class recognition in real-time settings," *IEEE Sensors Journal*, vol. 21, no. 11, pp. 13262–13271, 2022.

[8] T. Moten, A. Fitterer, E. Brazier, J. Leonard, and A. Brown, "Examining online college cyber security threats and preventive measures," *Information Technology and Management Science*, vol. 16, pp. 190–195, 2013.

[9] K. Taghipour and H. T. Ng, "A neural approach to automated essay scoring," in *Proc. 2016 Conf. Empirical Methods in Natural Language Processing (EMNLP)*, Austin, TX, pp. 1882–1891, 2016.

[10] M. Maree, M. Khelifa, and I. Osman, "A knowledge-based approach to skill extraction and context-sensitive job matching," in *Proc. 42nd Int. ACM SIGIR Conf.*, Paris, France, pp. 1273–1276, 2019.

[11] A. Bhola, K. Halder, and K. Prasad, "Retrieving skills from job descriptions using extreme multi-label classification with pre-trained language models," in *Proc. 28th Int. Conf. Computational Linguistics (COLING)*, Barcelona, pp. 5832–5842, 2020.

[12] A. Bangor, P. T. Kortum, and J. T. Miller, "An empirical evaluation of the System Usability Scale," *International Journal of Human-Computer Interaction*, vol. 24, no. 6, pp. 574–594, 2008.

