

Campus360: An Integrated Academic Management System for Polytechnic Institutions

Dr. C. P. Divate¹, P. D. Honmore², Mandar Mahadev Umarani³, Rajat Abhijeet Devkar⁴, Peerzade Shafaqat⁵, Aditya Hanmant Bajbale⁶

¹ Dean, ATS. Sanjay Bhokare Group of Institute, Faculty of Polytechnic, Miraj, Maharashtra, India.

² HoD, Project Guide, Department of Electronics & Computer Engineering, ATS. Sanjay Bhokare Group of Institute, Faculty of Polytechnic, Miraj, Maharashtra, India.

^{3, 4, 5, 6} Department of Electronics & Computer Engineering, ATS. Sanjay Bhokare Group of Institute, Faculty of Polytechnic, Miraj, Maharashtra, India.

To Cite this Article: Dr. C. P. Divate¹, Mrs. P. D. Honmore², Mandar Mahadev Umarani³, Rajat Abhijeet Devkar⁴, Peerzade Shafaqat⁵, Aditya Hanmant Bajbale⁶, "Campus360: An Integrated Academic Management System for Polytechnic Institutions", Indian Journal of Computer Science and Technology, Volume 05, Issue 03 (September-December 2026), PP: 574-578.



Copyright: ©2026 This is an open access journal, and articles are distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by-nc-nd/4.0/); Which Permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Abstract: Academic administration in polytechnic institutions involves connected records for students, faculty, courses, semesters, timetables, attendance, examinations, fees and certificates. When these activities are maintained in separate files or processes, staff may face repeated data entry, delayed updates and difficulty detecting inconsistencies. This paper proposes Campus360, a role-based academic management system organized around a shared institutional data model. Its modules cover student and faculty profiles; courses and semesters; timetable scheduling with overlap checks; attendance and percentage reports; examinations, marks and grades; performance dashboards; fee records and receipts; document and certificate workflows; notices; and access control for administrators, faculty and students. The proposed architecture uses a web application, application services and a relational database, with authorization enforced for each protected operation. The system is intended for implementation and evaluation at a polytechnic institution. Since implementation measurements are not supplied, this paper presents the design and a planned evaluation rather than claiming tested outcomes.

Key Word: Academic Management System, Student Information System, Timetable Scheduling, Attendance Management, Role-Based Access Control, Polytechnic Education, Learning Analytics.

1. INTRODUCTION

Academic administration depends on accurate information about learners, staff, subjects and institutional processes. Student enrollment and semester assignment are connected to course offerings, faculty allocations, class schedules, attendance, assessment and official documents. When each process is handled through a separate register or spreadsheet, information can become duplicated, difficult to reconcile and slow to retrieve. A unified information system can establish consistent records and make routine academic work easier to coordinate.

Campus360 is proposed as an integrated platform for ATS. Sanjay Bhokare Group of Institute, Faculty of Polytechnic, Miraj.

It brings thirteen functional areas into one application: student and faculty management; courses and semesters; timetable scheduling and conflict detection; attendance; examinations and results; performance analytics; fee management; certificates and documents; notices and communications; role-based access; and data security. Rather than treating these as unrelated screens, the design connects them through shared identifiers for student, faculty, course, semester, class, and academic year.

The main contributions are: (i) a modular system design covering the stated academic and administrative requirements; (ii) rules for validating timetable, attendance, marks and fee records; and (iii) a security model that separates administrative, faculty and student capabilities. The proposal is intended as a practical project blueprint. Its operational benefits and quality must be confirmed through implementation and institutional testing.

A key design requirement is that a fact should be entered once and reused only where the user is authorized to see it. For example, a student's semester enrollment should determine the subjects available for attendance and marks, while the same semester context should constrain timetable and result reports. This reduces inconsistent copies of the same record and makes it easier to trace the source of a reported value. The design also separates routine data entry from approval-sensitive actions such as publishing results, adjusting a payment or issuing a certificate.

The proposed system is intended for the operational scale of a polytechnic department and can be introduced incrementally. A first release may provide master data, enrollment, roles and timetable validation. Subsequent releases can add class

attendance, assessment and results, then fees, certificates, communication and analytics. This staged approach allows institute staff to validate terminology and workflows before the full set of modules is put into service. The paper therefore emphasizes a clear data model, explicit business rules and an evaluation plan over unverified claims of efficiency improvement.

The remainder of this paper is organized as follows. Section II reviews related work. Section III describes the proposed architecture and modules. Section IV presents expected benefits, feasibility and the evaluation plan. Section V concludes the paper.

II. LITERATURE REVIEW

Student information systems consolidate academic and administrative records around a student lifecycle. For Campus360, this idea is extended into a connected workflow in which identity, enrollment, course allocation, attendance, examination and fee records refer to the same academic context. A shared data model is important because a result or attendance report is meaningful only when associated with the correct student, subject, semester and teaching assignment.

In a modular system, shared records need clear ownership and validation. Course and semester setup should precede enrollment; enrollment should precede attendance and mark entry; and publication should follow verification. This ordering is reflected in the proposed application workflow. It also means that reports should identify their source term and calculation rules, allowing a user to understand which underlying records produced a displayed total. Timetabling is a constrained allocation task involving events, rooms, teachers, students and time slots. Babaei et al. survey methods for university course timetabling and describe the role of hard constraints and optimization of softer preferences [1]. Davison et al. formulate timetable decisions with multiple stakeholders and resource requirements [2]. These studies support treating a schedule as validated data rather than an editable calendar alone. Campus360 therefore checks that a faculty member, classroom, cohort and subject are not allocated to incompatible simultaneous sessions. More advanced optimization can be added later; the initial project scope is deterministic conflict detection and administrative correction.

Academic dashboards can help staff review patterns across attendance and marks, but analytics must be interpreted responsibly. Pardo and Siemens discuss ethical and privacy principles for learning analytics, including transparency, student agency and responsible use of data [3]. Campus360 limits dashboards to academic indicators needed for institutional support and proposes access controls and purpose-aware data handling. The dashboard is descriptive; it is not designed to automatically make high-stakes decisions about students.

Access control is a core architectural concern. The NIST RBAC model associates permissions with roles and defines levels of role-based functionality [4]. OWASP recommends least privilege, deny-by-default behavior and permission checks on each request [5]. These principles inform the proposed Admin, Faculty and Student roles. Role checks are combined with record scope: for example, a faculty member should see the assigned classes and their students, while a student should see only their own academic record.

Research gap. The project requirement is not a new isolated algorithm; it is a cohesive, institution-specific workflow spanning academic, financial and document operations. The design addresses this integration need by defining one modular platform, common identifiers, validation rules, role-scoped access, and reporting across the listed features. The proposal remains to be tested against actual institute workflows and representative data.

III. METHODOLOGY

Campus360 is designed as a layered web application. Users access role-specific screens through a browser. Application services validate business rules and permissions before reading or changing records. A relational database stores linked academic entities, while reporting services prepare attendance, result, fee and administrative summaries. Fig. 1 shows the conceptual processing flow; Table I summarizes the modules and principal responsibilities.

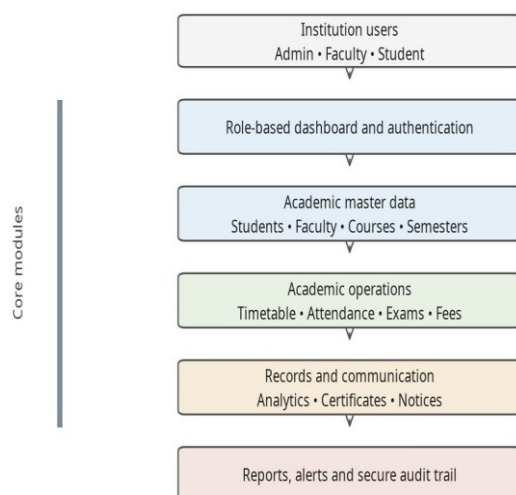


Fig. 1. Proposed Campus360 architecture and flow of academic information.

Module	Function	Key data / controls
1. Student & faculty	Profiles, enrollment, department, semester, subject allocation and class assignments.	Student ID, staff ID, course, cohort; role and scope checks
2. Courses & timetable	Manage courses, subjects and semesters; allocate rooms, faculty and time slots; detect overlaps.	Room/faculty/cohort/time-slot conflict validation
3. Attendance & exams	Record attendance, compute percentage; schedule exams, enter marks, calculate grades and publish results.	Session-wise presence; assessment and grade rules
4. Fees & documents	Maintain fee structures, payments, outstanding dues, receipts and certificate/document records.	Transaction history, document status, authorized issue
5. Notices & analytics	Publish notices and alerts; summarize attendance, marks and academic trends for authorized users.	Audience scope, aggregate indicators, audit events
6. Access & security	Authenticate users, enforce Admin/Faculty/Student permissions, log sensitive actions.	Least privilege, deny by default, request-level checks

TABLE I. Summary of the proposed Campus360 modules.

A. Module 1: Student and Faculty Management

The student record includes a unique institutional identifier, name, contact and enrollment information, department, programme, semester and active status. Faculty records contain staff identity, department, subjects and assigned classes. Enrollment and assignment changes should be effective-dated where required so reports can be interpreted in the correct academic year. Duplicate identifiers and invalid department or semester references are rejected before saving.

Administrative forms should distinguish required fields from optional profile information and provide clear validation messages. A student may have multiple enrollment records across academic years, but each active enrollment must be linked to a valid programme and semester. Faculty subject allocation should similarly be tied to a session or academic year. Changes to core identifiers or status should be restricted to authorized staff and retained in the audit history, helping the institution investigate corrections without exposing unnecessary information.

B. Module 2: Courses, Semesters and Timetable Scheduling

Course and subject records define the department, semester, subject code and academic year. A timetable entry links a subject to a faculty member, cohort, room, day and time interval. The first implementation checks hard conflicts: a faculty member cannot teach two classes at once; a room cannot host overlapping sessions; and a cohort cannot be assigned two simultaneous subjects. A conflict is reported with the conflicting records and is not silently overwritten. Optional preferences, such as preferred periods or balanced daily load, can be considered as a later scheduling enhancement [1], [2].

C. Module 3: Attendance, Examinations and Results

Attendance is recorded for a scheduled class session, subject and enrolled student.

The percentage for a student and subject is calculated as:

$$\text{Attendance percentage} = (\text{sessions attended} \div \text{sessions conducted}) \times 100 \quad (1)$$

The system should define how cancelled sessions, late entries and approved corrections are handled before deployment. Exam records connect an assessment to a course, semester and date. Faculty enter marks only for authorized subjects. Grade calculation follows the institute's approved grading rules, which must be configurable rather than assumed. Published results are versioned or audited so that corrections remain traceable.

Attendance reports should show the numerator and denominator behind each percentage, the date range and any excluded or cancelled sessions. Corrections should record the editor, timestamp and reason. Before marks are published, the system can check that required assessments have been entered and that each score lies within the configured maximum. A result summary should remain linked to the assessment scheme used for the semester; changing a grading rule should not silently rewrite previously approved results.

D. Module 4: Student Performance Analytics

Authorized dashboards summarize attendance percentages, assessment marks, grade distributions and semester-level trends. Students see their own results and attendance; faculty see assigned classes; administrators may access institution-level summaries according to policy. Where class-level summaries are shown, the interface should identify the reporting period and cohort. Indicators support review and communication but are not, by themselves, proof of a student's ability or cause of a performance change [3].

For transparency, dashboard labels should explain how measures are calculated and should avoid presenting incomplete data as a final judgment. Filters may include academic year, department, semester and subject, subject to the viewer's permissions. Individual-level indicators should be used for legitimate academic support and should be visible only to staff with an appropriate educational responsibility. A student-facing view can prioritize understandable feedback and access to their own source records.

E. Module 5: Fee Management and Certificates

Fee structures are associated with academic year, programme or semester, and applicable student category as defined by the institute. Payment entries record amount, date, mode, reference and receipt status. Outstanding dues are calculated from the approved fee schedule and posted payments; adjustments require an authorized reason and an audit entry. The certificate workflow stores request type, student identity, verification status, issue date and issuing authority. Bonafide and leaving certificates are generated only after required checks and approvals.

F. Module 6: Notices, Access and Data Protection

Notices may target all users or a defined department, semester, class or role. Timetable changes, examination updates and general announcements are dated and retained for reference. Access is denied unless a role and record-scope rule explicitly permits the requested action. Every protected operation is checked server-side, including downloads and API requests, following least-privilege and deny-by-default guidance [4], [5]. Sensitive data should be limited to legitimate institutional purposes, with secure authentication, backups, retention rules and audit logging included in deployment planning.

The permission matrix should be specified before implementation. Administrators manage configuration and authorized institutional records; faculty manage attendance and assessment for their assigned classes; students view permitted personal records and notices. No role should receive broad access simply because it is convenient for a screen. The system should also protect exported reports and certificates, since authorization risks apply to files as well as on-screen pages. Routine backups, tested restore procedures, secure session handling and periodic access review are necessary operational controls, not substitutes for correct application permissions.

G. Proposed data and application flow

A typical workflow begins when an administrator configures an academic year, departments, courses, semesters, users, rooms and fee structures. Faculty assignments and timetable entries are then created and checked for conflicts. Faculty take attendance and enter assessment marks for their assigned classes. The application calculates permitted summaries; students view their own academic information and notices. Fee receipts and certificate requests pass through their defined authorization and approval steps. A relational schema may include Student, Faculty, Department, Course, Subject, Semester, Enrollment, Timetable, AttendanceSession, AttendanceRecord, Exam, Mark, FeeStructure, Payment, Document, Notice, Role and AuditLog entities. Referential integrity and unique constraints protect essential relationships.

IV. EXPECTED BENEFITS AND EVALUATION PLAN

A. Expected Benefits

The proposed system is expected to reduce repeated entry by connecting records across academic modules. Automated timetable checks can expose resource clashes before schedules are published. Attendance and result summaries can be produced from recorded sessions and assessments rather than recomputed manually. Centralized notices and certificate workflows can make status and responsibility clearer. Role-scoped access is intended to reduce unnecessary exposure of student and faculty information. These are expected benefits, not measured outcomes; they depend on reliable data, user adoption and appropriate operational procedures.

B. Planned Evaluation

After implementation, evaluation should use synthetic or institution-approved test records and a defined pilot period.

The following measures are proposed:

- Functional correctness: percentage of agreed requirements that pass documented test cases, including enrollment, attendance, marks, fees and certificate workflows.
- Timetable validation: detection of deliberately inserted faculty, room and cohort overlaps, alongside verification that valid schedules are accepted.
- Calculation accuracy: agreement of attendance percentages, grades and outstanding balances with independently calculated test cases and
- institute-approved rules.
- Access-control effectiveness: role and record-scope tests for allowed and denied actions, including direct URL/API access and document downloads.
- Operational usability and performance: task completion time, user feedback, report generation time and response time under a stated test load.
- Data quality and reliability: duplicate/invalid record rejection, audit coverage, backup restoration checks and error recovery.
- Results should report the test dataset,

C. Feasibility and Implementation

The project can be developed incrementally: first establish authentication, roles and core academic records; then add course and timetable operations; next implement attendance, exams and results; and finally add fees, certificates, notices and dashboards. A conventional web stack with a relational database is sufficient for the stated workflows. Technology selection

should follow the team's available skills and institutional constraints. Deployment readiness requires validation of local grading and fee rules, migration plans, access policies, backups, and user training. Applicable Indian privacy and institutional policies should be reviewed before production use.

V. CONCLUSION

This paper presents Campus360, a proposed integrated academic management system for a polytechnic institution. The design connects student and faculty records with courses, semesters, timetables, attendance, examinations, performance summaries, fees, documents and notices. It includes deterministic timetable conflict checks and role-scoped access for administrators, faculty and students. The architecture is modular so that each feature can be implemented and verified in stages.

Future work is to confirm requirements with institute stakeholders, implement the database and interfaces, test the academic rules against approved procedures, and measure usability, correctness, security and performance. Only results from that evaluation should be used to claim effectiveness.

Acknowledgment

The authors acknowledge the guidance of Dr. C. P. Divate and Mrs. P. D. Honmore and the support of ATS. Sanjay Bhokare Group of Institute, Faculty of Polytechnic, Miraj, Maharashtra, India. This acknowledgment is based on the roles and institution details supplied for this manuscript.

REFERENCES

1. Babaei, J. Karimpour, and A. Hadidi, "A survey of approaches for university course timetabling problem," *Computers & Industrial Engineering*, vol. 86, pp. 43–59, 2015. doi: 10.1016/j.cie.2015.04.010.
2. M. Davison, A. Kheiri, and K. G. Zografos, "Modelling and solving the university course timetabling problem with hybrid teaching considerations," *Journal of Scheduling*, vol. 28, no. 2, pp. 195–215, 2025. doi: 10.1007/s10951-024-00817-w.
3. A Pardo and G. Siemens, "Ethical and privacy principles for learning analytics," *British Journal of Educational Technology*, vol. 45, no. 3, pp. 438–450, 2014. doi: 10.1111/bjet.12152.
4. R. Sandhu, D. Ferraiolo, and R. Kuhn, "The NIST model for role-based access control: Towards a unified standard," in *Proc. 5th ACM Workshop on Role-Based Access Control*, 2000. doi: 10.1145/344287.344301.
5. OWASP Cheat Sheet Series. Available: https://cheatsheetseries.owasp.org/cheatsheets/Authorization_Cheat_Sheet.html (accessed 27 Oct. 2026).