WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Academic Management and student Request Approval System

Ch. Rakesh[1], M. Bhargavi[2], Sk. Md. Afreen[2], Sk. Musharaf Ali[2], P. Sanjay Kumar[2]

[1] Supervisor, Department of CSE-AI&ML, GMR Institute of Technology, Andhra Pradesh, India

[2] Students, Department of CSE-AI&ML, GMR Institute of Technology, Srikakulam, Andhra Pradesh, India

A B S T R A C T :

Higher-education institutions still process routine academic services such as bonafide, provisional, conduct and original-degree certificates, transcripts, leave applications and ID-card requests through paper forms, physical signatures and scattered e-mails. This leads to long turnaround times, misplaced documents and no real-time visibility for students. This paper presents a centralized, web-based Academic Management and Request Approval System in which a student submits a request once and tracks it through a configurable multi-level approval chain involving Mentors, Heads of Department (HOD), Deans, the Examination Branch / Controller of Examinations (CoE) and Administrators. Approval is governed by Role-Based Access Control (RBAC): every action is authorized by the user's role, department scope and the request's current stage, so only the designated role-holder can approve, return or issue a request, and requests are handled in order of submission rather than ranked by urgency. The platform issues digitally generated, QR-verifiable certificates, records every approval and denied access attempt in an audit trail, sends automated e-mail and in-app notifications, and offers a central repository for semester-wise records, announcements and secure student–faculty messaging with CC-based communication. The system is implemented with React.js (front end), Django and Django Celery (back end and workflow automation), Redis (Celery broker and cache) and PostgreSQL (database). The paper details the architecture, workflow, RBAC authorization model and certificate-approval process, positions the work against fifteen recent studies, and defines an evaluation protocol based on turnaround time, return rate, access-control correctness and usability, and reports the implemented portal screens as preliminary results.

Keywords: *Academic Management System, Student Request Approval, Role-Based Access Control (RBAC), Workflow Automation, Digital Certificates, Audit Trail, Web-Based Application.*

1. INTRODUCTION

Colleges and universities handle a continuous stream of student-facing administrative requests: certificates (bonafide, study, conduct, provisional and original degree), consolidated and semester grade memos, no-dues and transfer certificates, leave applications and ID cards. In many institutions these requests still travel as paper forms that must be signed in sequence by a mentor, the Head of Department, the Dean and the Examination Branch. A student has no way of knowing where the form is, a missing signature stops the whole chain, and it is often unclear who is authorized to act on a request at each stage.

Digital portals have addressed parts of this problem, but, as the survey in Section 2 shows, most of them digitize a single process such as student clearance [3][6], thesis approval [2], work orders [1] or course registration [11]. Few combine several services, a configurable multi-level approval chain, stage-level role-based access control and verifiable document issue in one platform.

This work proposes a Centralized Academic Management and Request Approval System that closes this gap. The main contributions are:

- A unified web portal in which one request model covers certificates, transcripts, leave, ID cards and general enquiries, with real-time tracking and automated notifications.
- A configurable, role-based, multi-level approval workflow (Student → Mentor → HOD → Dean / Examination Branch → Administrator / Final Authority) in which every request type declares its own chain and any approver can return a request with comments.
- Role-Based Access Control (RBAC) that authorizes every approval action by the approver's role, department scope and the request's current stage, with least-privilege views, no stage skipping and no self-approval.
- Digital document management with e-document (PDF) generation, unique reference IDs, QR-based verification and a complete audit trail.
- Integrated secure messaging with CC-based communication between students and faculty inside the workflow.

The rest of the paper is organized as follows. Section 2 reviews related work and identifies the research gap. Section 3 describes the system architecture. Section 4 presents the methodology: workflow, RBAC authorization model, certificate approval and security. Section 5 gives the implementation, the evaluation plan and the results obtained so far. Section 6 discusses expected outcomes and limitations, and Section 7 concludes with future scope.

2. RELATED WORK

Fifteen recent studies on academic management, request approval and workflow automation were reviewed. They fall into four groups.

Approval and workflow systems. Pinto et al. [1] replace paper-based work orders with a role-based, multi-level approval workflow with e-mail alerts and report that processing time falls from about five days to under 24 hours. Grooms et al. [2] automate approval of graduate thesis and capstone projects, removing physical signatures and giving real-time visibility of pending approvals. Adedapo et al. [5] convert the academic and administrative workflows

of a university department into a digital workflow system that centralizes communication and task allocation. Sudharshan [12] provides a student request and approval platform with approval history, e-mail notification and role-based access, but its flow is mainly student-to-instructor.

Student clearance systems. Adegoke and Olalekan [3] and Eweoya et al. [6] automate multi-department clearance. Both offer real-time progress tracking and notifications, and [3] adds an audit trail and QR-based certificate verification. Their scope is limited to graduation clearance.

Academic management portals. Parameswari et al. [7], Hidayat and Sofyan [8], Shelke et al. [9], Rao et al. [10], Nagarjuna et al. [13], Reddii et al. [14] and Oliveros et al. [15] build centralized portals with role-based dashboards for students, faculty and administrators on stacks such as Django, MERN and Java. They manage records, courses, notices and enrolment. Reddii et al. [14] come closest to this work by automating leave, gate-pass, certificate and complaint workflows, but they depend on institutional system integration and do not describe a stage-level authorization model for approvals. Rajapandian and Kanchanashree [11] address online course registration only.

Mentoring and credential security. Tamilarasi et al. [4] propose an AI-driven mentor–mentee platform with matching, scheduling and chat, which motivates our mentor stage and messaging. The base paper for this work, BlockMEDC by Fartitchou et al. [16], secures Moroccan higher-education digital certificates using Ethereum Layer-2 (zk-Rollup) smart contracts and IPFS, addressing authenticity, manual verification effort and interoperability. We adopt its core idea, tamper-evident and instantly verifiable certificates, through lighter QR/hash-based verification, and treat blockchain anchoring as future work.

Table 1. Summary of the reviewed literature

Ref.	Year	System / focus	Key strength	Limitation
[1]	2026	Automated work-order management	5 user roles; real-time tracking; e-mail alerts	Limited to institutional work orders
[2]	2016	Thesis / capstone approval	Replaces paper forms; reporting	Specific to one process
[3]	2026	Workflow-driven student clearance	RBAC; audit trail; QR verification	Depends on many departments; clearance only
[4]	2025	Mentor Connect (AI mentorship)	Mentor matching; chat; scheduling	Needs sustained participation; no approvals
[5]	2023	Higher-education workflow system	Centralized tasks and communication	Single-department case study
[6]	2025	Web-based clearance	Real-time status; document upload	Graduation clearance only
[7]	2026	Virtual Campus Assistant	Role-based dashboards; communication	Not approval-centric
[8]	2026	Django academic and admission monitoring	Replaces spreadsheets; less redundancy	Admission-focused
[9]	2025	Student management system	Centralized records and profiles	No workflow automation
[10]	2025	MERN academic management	Scheduling; complaints; notices	No multi-level approvals
[11]	2025	Online course registration	Less paperwork; central data	Narrow scope
[12]	2025	Student request and approval	Request tracking; approval dashboard	Mainly student-to-instructor
[13]	2026	Smart academic management	Role-specific dashboards; secure access	Broad, not request-centric
[14]	2026	Unified digital student portal	Leave, certificates, complaints in one portal	Depends on institutional system integration
[15]	2026	Web-based student portal	Digital enrolment and grades	Small-school scope

2.1 Research gap

The survey shows five recurring gaps, which the proposed system targets directly (Table 2).

Table 2. Identified gaps and the corresponding design response

Finding in literature	Research gap	Proposed response
Services (clearance, registration, thesis, records) are digitized separately.	Academic services remain fragmented.	One centralized portal for all student services.
Approval workflows are built for one specific process.	Limited configurable multi-level approval across request types.	Per-request-type approval chain: Student → Mentor → HOD → Dean / Examination Branch → Final Authority.

Finding in literature	Research gap	Proposed response
Portals offer basic submission and status.	Little visibility of the full request journey.	Real-time tracking, approval history, timestamps, notifications.
Communication is handled outside the workflow.	No integrated student–faculty communication.	Messaging with CC-based communication inside each request.
Focus is on processing and records.	Role-based access is rarely specified at the level of each approval stage; limited document handling and auditability.	Stage-level RBAC with a permission matrix, QR-verifiable e-documents, and audit trail.

3. PROPOSED SYSTEM ARCHITECTURE

3.1 Overall architecture

The system follows a layered architecture (Fig. 1). Five user roles access the portal through a React.js single-page application. Requests enter the Request / Service Module, pass to the Workflow Engine for validation, routing and review, are decided by the multi-level Approval Authorities, and produce outcomes such as approval or rejection, status tracking, notifications, generated documents and audit logs. All layers read and write a centralized PostgreSQL database that holds user data, requests and status, approvals and workflow data, documents, and notifications and logs.

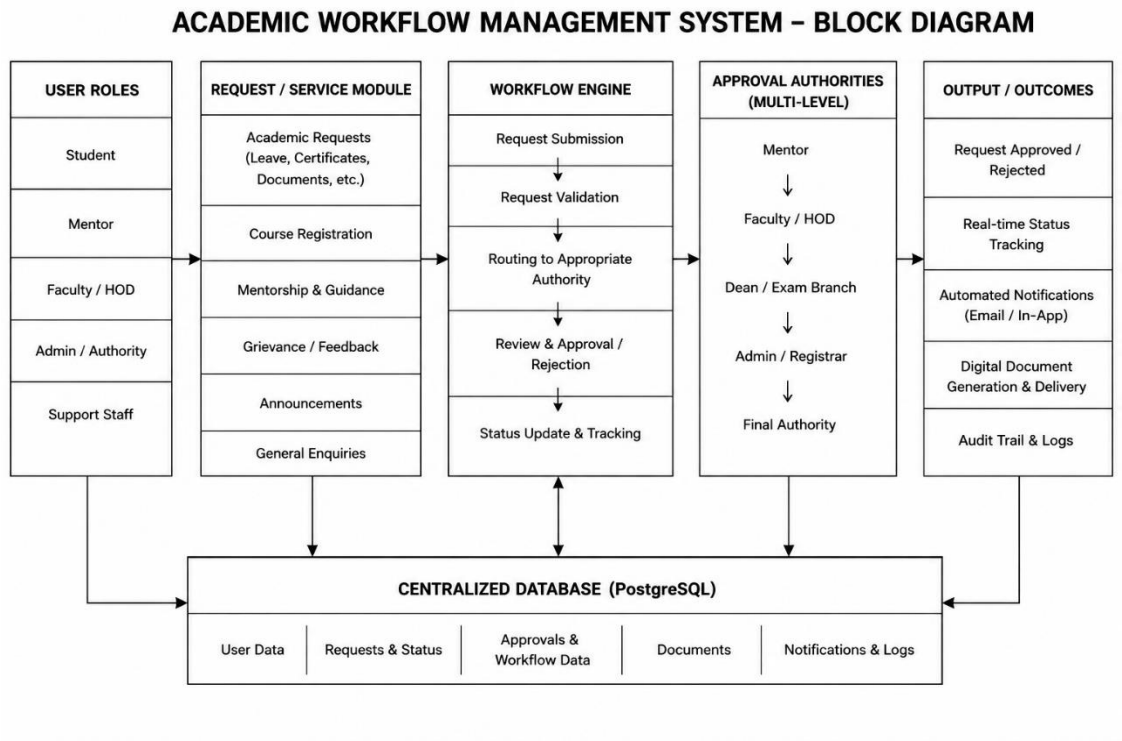


Fig. 1. Block diagram of the Academic Workflow Management System

3.2 User roles

Role	Responsibilities
Student	Logs in with roll number and password, submits requests with supporting documents, tracks status, corrects returned requests, downloads issued documents, messages faculty.
Mentor	First-level reviewer for requests that need mentor recommendation (for example leave); guides the student and can CC communications.
Faculty / HOD	Verifies department details, academic records and eligibility; approves, rejects or returns the request with comments.
Dean / Examination Branch (Coe)	Final academic approval, verification of documents and academic status, and issuance of examination-related certificates.

Role	Responsibilities
Admin / Authority and Support Staff	Configures request types, roles, permissions and approval chains; manages users but cannot approve on behalf of others; handles non-academic requests and system monitoring.

3.3 Technology stack

Layer	Technology	Purpose
Front end	React.js	Responsive role-based dashboards, request forms, live status view.
Back end / API	Django (Python), REST API	Business logic, validation, authentication, RBAC, document generation.
Access control	Django auth + custom RBAC	Role, permission and object-level (department, stage) checks on every API call.
Workflow automation	Django Celery	Asynchronous routing, reminders and notification dispatch.
Broker / cache	Redis	Celery message broker and cache for role and permission lookups.
Database	PostgreSQL	Relational storage of users, requests, approvals, documents and logs.
Notifications	E-mail and in-app	Status updates at every stage.

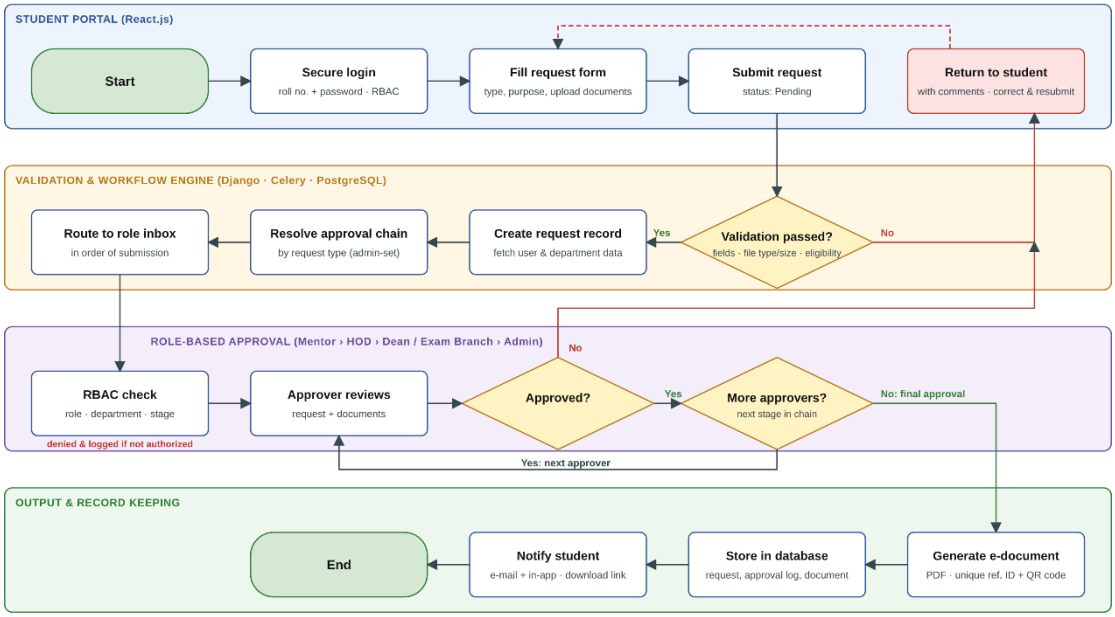
4. METHODOLOGY

4.1 Request lifecycle

Fig. 2 shows the end-to-end flow. The steps are:

1. The student fills the request form in the React client (for example a leave application or transcript request) and submits it with supporting documents.
2. The back end validates required fields, file type and size, and eligibility. Invalid or incomplete requests are returned to the student for correction.
3. The Workflow Manager (Django Celery) creates the request, authenticates the user, fetches the user and department data from PostgreSQL and resolves the approval chain defined for the request type.
4. The request is placed in the inbox of the role responsible for the current stage, in order of submission. Before any action, the RBAC check (Section 4.2) verifies the approver's role, department scope and the request's stage; unauthorized attempts are denied and logged.
5. The authorized approver reviews the request and documents and approves it or returns it with comments; a returned request goes back to the student for correction and resubmission. After each approval the engine checks for further stages (for example Mentor → HOD → Examination Branch → Admin) and routes the request to the next approver.
6. After final approval the system generates the e-document (PDF), updates the status to Completed and notifies the student, who downloads the output from the portal.

Fig. 2. Student request approval flowchart with RBAC-controlled approval stages



4.2 Role-based access control (RBAC) for approval

Approval authority is governed by role-based access control, not by how urgent a request is declared to be. Users are assigned roles, roles are granted permissions, and every action on a request is authorized against the user's role, department scope and the request's current stage. Each role works from its own inbox, in which requests appear in order of submission, so no request is advanced ahead of others because of its label.

Let each user u hold a role ρ(u) and a department scope D(u), let each role have a permission set Perm(r), and let each request type t have an administrator-defined approval chain $C(t) = (r_1, r_2, \ldots, r_n)$ of roles. User u may approve request q at stage k only if

$$\rho(u) = r_k \ \wedge \ approve \in Perm(\rho(u)) \ \wedge \ D(u) \supseteq dept(q) \ \wedge \ status(q) = Pending_k \qquad (1)$$

where r_k is the k-th role in C(type(q)). If any condition fails, the API returns HTTP 403 and the attempt is written to the audit log. The design follows these principles:

- **Least privilege.** Students see only their own requests; an approver sees only requests pending at their stage and within their department scope; the administrator configures the system but cannot approve on behalf of others (separation of duties).
- **No stage skipping, no self-approval.** A stage can be completed only by the role defined for it, and a user cannot approve a request that they submitted.
- **Server-side enforcement.** Role, permission and object-level checks run in Django on every call; the React interface merely hides actions the user cannot perform.
- **Configurable.** New request types and approval chains are created by assigning roles to stages, without code changes, and role changes take effect immediately.
- **Accountability.** Every approval, return and denied attempt is logged with user, role, timestamp and comments.

Table 3. Role–permission matrix

Role	Submit	View requests	Approve / return	Issue certificate	Configure system	Audit log
Student	Yes	Own only	No	No (downloads own)	No	No
Mentor	No	Assigned mentees	Mentor stage	No	No	No
Faculty / HOD	No	Own department	HOD stage (department)	No	No	Department
Dean	No	Dean scope	Dean stage	No	No	Read
Exam Branch / CoE	No	Assigned	CoE stage	Yes	No	Read
Admin / Authority	No	All (read)	No	No	Yes	Full
Support Staff	No	Non-academic requests	Non-academic stage	No	No	No

4.3 Certificate request approval workflow

The certificate service is the most regulated part of the platform and is modelled in Fig. 3. After login with roll number and password the student selects the certificate type, enters purpose and year, uploads supporting documents and submits. The system validates the form and eligibility (for example, a student must have passed out before an original degree or provisional certificate is issued) and sets the status to Pending with HOD. The request then passes the HOD (department verification), the Dean (final academic approval) and the Controller of Examinations (final verification and issuance). A rejection at any stage returns the request to the student with comments; the student corrects and resubmits. On approval the system generates the e-document with a unique reference ID, stores the request details, approval log, certificate copy and final status in the database, and notifies the student with a download link.

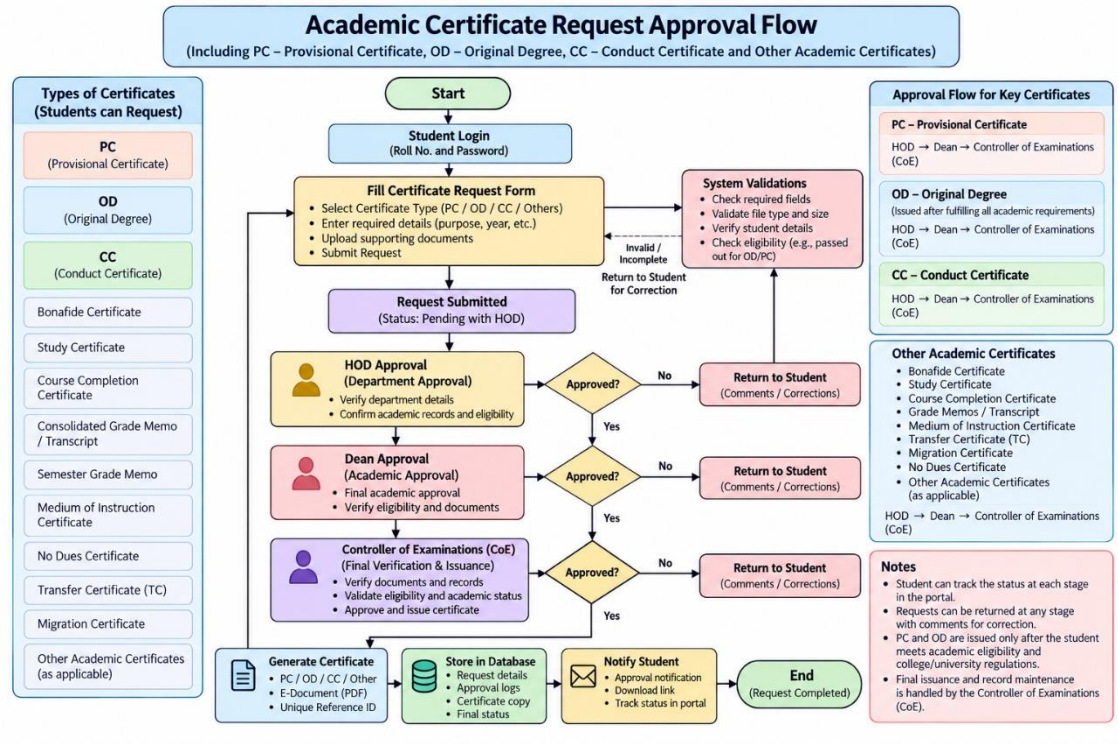


Fig. 3. Academic certificate request approval flow (PC, OD, CC and other academic certificates)

Table 4. Approval chain by certificate type

Certificate	Approval chain	Condition
PC – Provisional Certificate	HOD → Dean → CoE	Academic eligibility and regulations met
OD – Original Degree	HOD → Dean → CoE	All academic requirements fulfilled
CC – Conduct Certificate	HOD → Dean → CoE	Verified by department
Bonafide, Study, Course Completion, Medium of Instruction, No Dues, Transfer (TC), Migration, Grade Memo / Transcript	HOD → Dean → CoE	As applicable to the certificate

The two flows are two instances of one configurable engine. The general request flow in Fig. 2 uses the chain Mentor → HOD → Examination Branch → Admin, whereas certificates in Fig. 3 use HOD → Dean → CoE. The administrator defines the chain per request type, so a new service needs configuration, not new code.

4.4 Digital document management and verification

- Secure upload with file-type and size validation; storage of supporting documents and issued certificates.
- E-document (PDF) generation with a unique reference ID for every issued certificate.
- QR code on each document linking to a public verification page that returns the stored record and its hash, so employers and other institutions can confirm authenticity without contacting the college, following the QR-verification approach of [3].

4.5 Notifications, communication and audit

Every status change triggers an e-mail and an in-app notification. Students and approvers can exchange messages attached to a request, with CC to the mentor or HOD, so decisions and clarifications stay in one place. Each action (submit, approve, return, issue) is written to an append-only audit log with user, role, timestamp and comments, which gives accountability and a full approval history for any request.

4.6 Security

Authorization follows the RBAC model of Section 4.2; passwords are stored as salted hashes; sessions use token-based authentication over HTTPS; uploaded files are validated and stored outside the web root; and database access goes through the Django ORM to prevent SQL injection.

5. IMPLEMENTATION AND EVALUATION PLAN

5.1 Implementation status

The project is carried out in stages: literature survey, planning, requirement analysis and system design are complete; front-end, back-end and database development are in progress; testing and integration, deployment and documentation follow. The architecture in Fig. 1 and the workflows in Figs. 2 and 3 form the design baseline for the implementation.

5.2 Evaluation protocol

The system will be evaluated in a pilot with one department and compared against the existing manual process. No performance figures are claimed in this paper; the following measures will be reported after the pilot.

Table 5. Evaluation metrics

Metric	Definition	Baseline for comparison
Mean turnaround time	Submission to issue, per request type	Manual process records
Per-stage approval latency	Time spent at Mentor, HOD, Dean, CoE	Manual process
Access-control correctness	Test actions outside role, department or stage; % correctly denied and logged	Not applicable
Return / correction rate	% of requests returned for correction	Paper forms
Role-inbox waiting time	Time a request waits in each role inbox before action	Manual process
Scalability	Response time under concurrent users (load test)	Target set with administrators
Usability	System Usability Scale survey of students and staff	Not applicable
Verification correctness	Genuine and tampered documents correctly verified via QR	Not applicable

5.3 Results

The modules implemented so far were exercised with sample institutional data, and Figs. 4–8 show the resulting screens. Personal names have been removed from the screenshots and are replaced by the corresponding post or role. As stated in Section 5.2, no performance figures are claimed; the results below are functional outcomes, and the metrics in Table 5 will be reported after the pilot.

Library management module. Fig. 4 shows the Knowledge Resource Centre dashboard used by the Central Library. It summarizes the curated catalogue (70 titles across 9 academic branches), 939 available copies out of 1204 campus copies, 4 active borrowings, 3 overdue books and the outstanding overdue fines of ₹28. The module applies a 15-day borrowing cycle, a pre-due reminder on day 13 and an overdue fine of ₹1 per day, and its watchlist lists each overdue book with the number of days overdue, the fine accrued and a one-click “Collect & Renew” action.

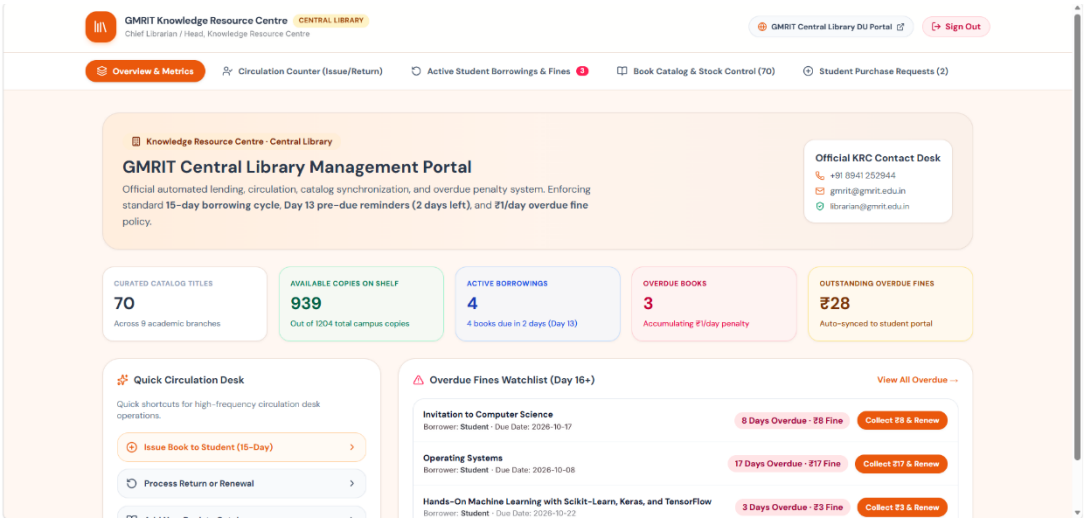


Fig. 4. Library management dashboard of the Knowledge Resource Centre with circulation metrics and the overdue-fines watchlist

Dean of Academics clearance panel. Fig. 5 shows the approver view for a Provisional Certificate (PC) request. The official three-tier chain (HOD endorsement, Dean clearance, Examination Cell verification and upload) shows the current stage, and the action area offers only the operations that the

Dean role is permitted to perform: grant clearance and forward to the Examination Cell, or reject / flag the request. Certificate upload is restricted to the Controller of Examinations, which demonstrates the stage-level RBAC rules of Section 4.2.

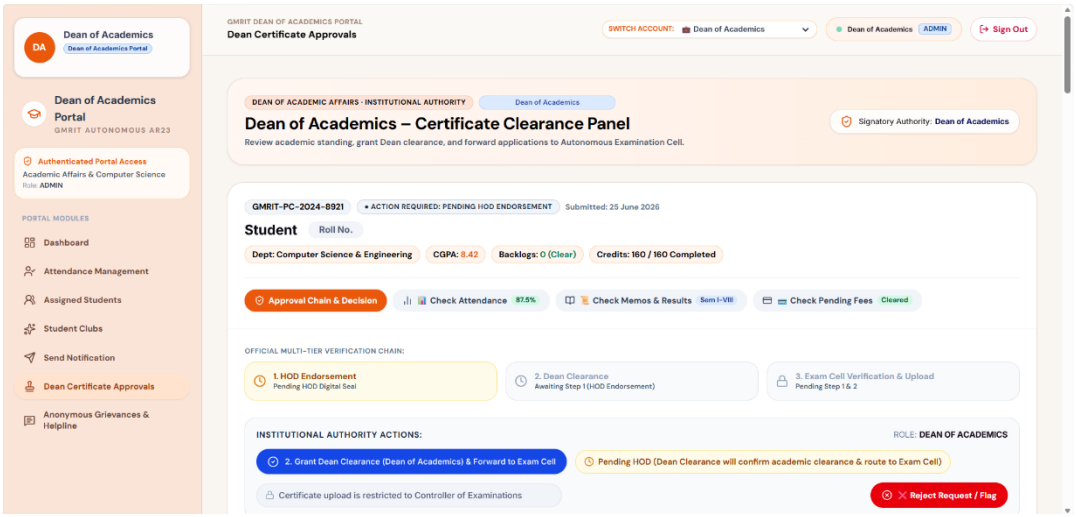


Fig. 5. Dean of Academics certificate clearance panel showing the multi-tier verification chain and role-restricted actions

Student certification tracker. Fig. 6 shows the student view of the same request. The live verification tracker reports each decision as soon as it is recorded, here "Dean Clearance Granted · Forwarded to Autonomous Examination Cell", and an in-app notification informs the student that the Examination Cell verification, performed by the Controller of Examinations, is complete and that the official certificate is being generated and uploaded. Until the upload is done the certificate remains locked, so a student cannot obtain a document before the final authority has issued it.

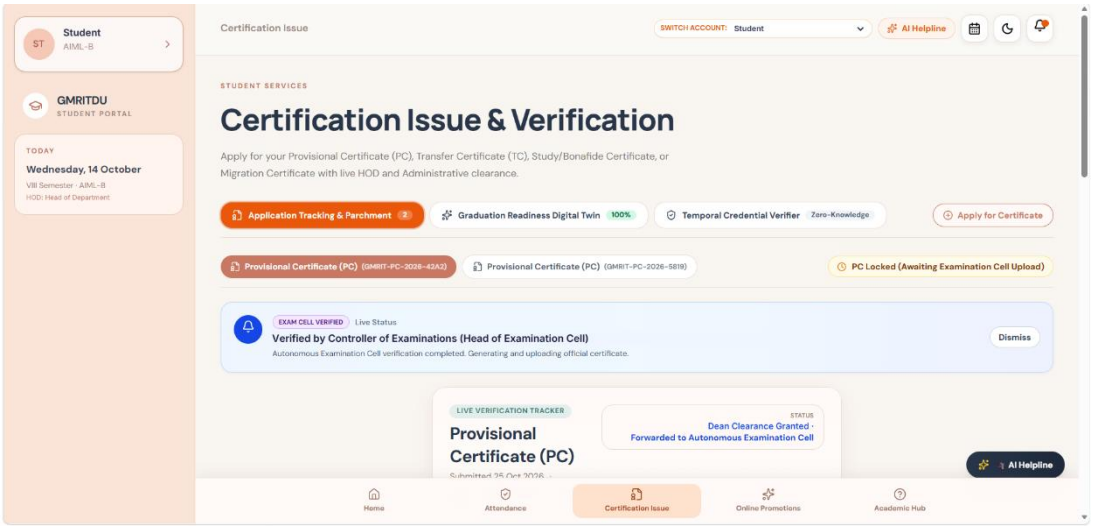


Fig. 6. Student certification issue and verification page with live status tracking and Examination Cell notification

Request tracking and document collection. Fig. 7 shows the continuation of the same Provisional Certificate request. The tracker header displays the application ID, the submission date and the current status, and lists each completed stage with its time stamp and the approver's remarks: the HOD endorsement records that academic records, credits, statutory attendance and laboratory clearance were verified, and the institutional clearance authority confirms that administration, accounts and library dues are cleared before forwarding the request. The Autonomous Examination Cell is then shown as the active stage, "In Progress", while the last stage, document collection at the Student Affairs counter, stays inactive until the certificate is uploaded. The page also shows the collection counter, its location and the documents to be brought. Names of the approvers are replaced by their roles in the figure.

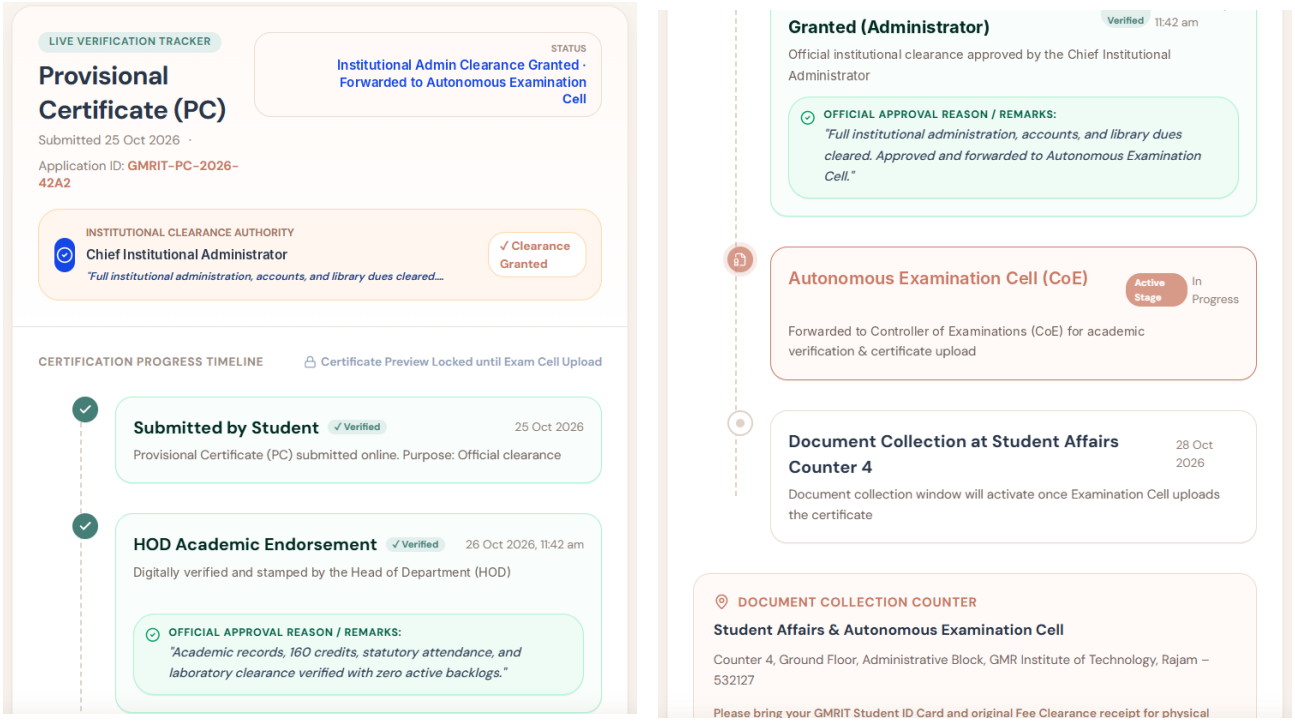


Fig. 7. PC request tracking: (a) request summary and completed approval stages; (b) active Examination Cell stage and document collection counter (approver names replaced by roles)

Provisional Certificate issue. Fig. 8 shows the Provisional Certificate (PC) generated once the Examination Cell verification is complete and the Controller of Examinations has uploaded the document. The e-document carries a unique serial number and PC number, the hall ticket number, the programme, the year of the examination and the class awarded, together with the institution seal and the signatures of the Controller of Examinations and the Chief Controller of Examinations. The student photograph, personal names and signatures are masked or replaced with sample values in the figure, in line with the anonymisation applied to Figs. 4–7. The issued certificate is then released to the student through the certification issue page of Figs. 6 and 7, completing the PC request lifecycle.

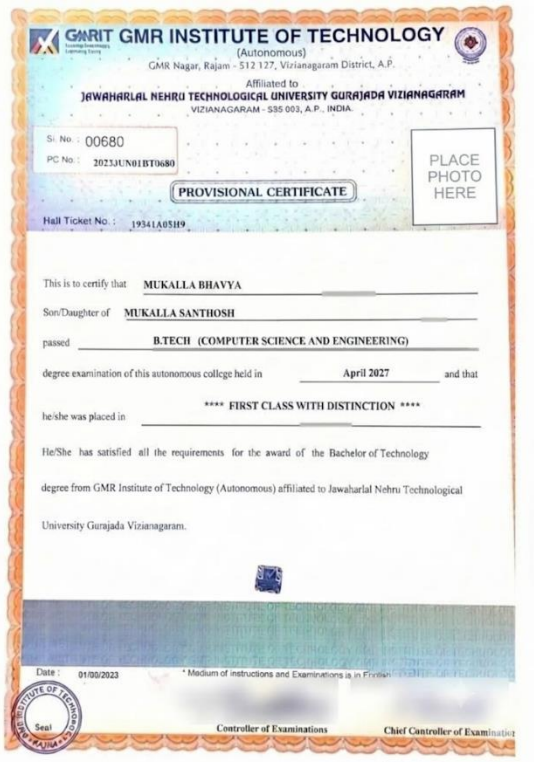


Fig. 8. Sample Provisional Certificate issued by the system after Examination Cell verification (personal details masked)

Together, these screens show that the request, approval and notification flow works end to end across three roles: the student submits and tracks, the approver acts only within the role's authority, and status changes appear to the student without manual follow-up.

6. DISCUSSION

Expected benefits. Related systems report large reductions in processing time when approvals are digitized, for example from about five days to under 24 hours in [1]. We expect a similar effect here, to be confirmed by the evaluation in Table 5. The additional benefits of the proposed design are role-scoped, accountable approvals, one platform for many services, and verifiable output documents.

Limitations. Sequential approval means one slow approver still delays a request, and because requests are handled in order of submission, time-critical requests rely on approvers acting promptly; automated reminders reduce but do not remove this. The permission matrix and approval chains must be validated with each department, and role assignments kept up to date. QR/hash verification proves that a record matches the portal but, unlike the blockchain approach of [16], relies on the institution hosting the verification service. Results will be limited to the pilot department until the system is rolled out more widely.

7. CONCLUSION AND FUTURE SCOPE

This paper presented a centralized web-based academic management and request approval system that unifies certificate, transcript, leave and other student services under one configurable multi-level approval workflow. By adding stage-level role-based access control, QR-verifiable e-documents, integrated messaging and an audit trail, it addresses the fragmentation, opacity and weak accountability that the surveyed systems leave open. The React.js, Django, Celery, Redis and PostgreSQL stack keeps the platform maintainable and scalable.

Future Scope:

Future enhancements to the proposed academic management and request approval system will focus on improving its scalability, security, accessibility, and operational efficiency. The following developments can be considered:

- Pilot Deployment and Performance Evaluation: Deploy the system in a real academic environment and evaluate its usability, reliability, scalability, security, and effectiveness in processing student requests.
- Secure Digital Document Verification: Enhance QR-code-based document verification mechanisms to support certificate authenticity checks, detect unauthorized modifications, and enable convenient verification of approved academic documents.
- Mobile Application Development: Develop dedicated Android and iOS applications with push notifications, real-time request status updates, and convenient access to academic services.
- Machine Learning for Delay Prediction: Incorporate machine-learning techniques to predict approval delays, identify workflow bottlenecks, estimate request processing times, and support what-if analysis for improving administrative efficiency.

REFERENCES

[1] Pinto, J., Priyadarshini, P., & Hareesh, B. (2026). Automated Work Order Management System for Educational Institutions. International Journal of Engineering Research & Technology (IJERT), 14(01). https://doi.org/10.17577/IJERTCONV14IS010028

[2] Grooms, J., Kline, D., & Cummings, J. (2016). Streamlining the Capstone Process: A Time-Saving Approval System for Graduate Theses/Projects. Information Systems Education Journal, 14(4), 81–96.

[3] Adegoke, T. B., & Olalekan, A. A. (2026). Design and Development of a Workflow-Driven Online Student Clearance System for Seamless Multi-Departmental Integration. ISA Journal of Engineering and Technology (ISAJET), 3(1), 62–74. https://doi.org/10.5281/zenodo.18506360

[4] Tamilarasi, K., Aishwarya, J., Chamini, G., & Betsee Natasha, A. (2025). Mentor Connect: AI-Driven Career Mentorship Platform for Personalized Growth and Skill Development. TIJER – International Research Journal, 12(5).

[5] Adedapo, O. A., Bello, J., Ogunrinde, M. A., Oyediran, M. O., Adio, M. O., & Oyekola, O. A. (2023). Development of a Workflow Management System for Higher Education: Case Study of a Department in a Nigerian University. FUOYE Journal of Engineering and Technology, 8(4), 456–462. https://doi.org/10.46792/fuoyejet.v8i4.1126

[6] Eweoya, I. O., Agbeyangi, A. O., & Lukose, J. M. (2025). Streamlining Student Clearance in Higher Education: A Web-Based Solution. Information Impact: Journal of Information and Knowledge Management, 16(1), 47–62. https://doi.org/10.4314/iijikm.v16i1.5

[7] Parameswari, K., Haritha, H., Ajith Kumar, B., Abinesh, M., & Gowtham, D. (2026). Virtual Campus Assistant (VCA): A Centralized Web-Based Academic Management System. International Journal of Science, Engineering and Technology, 14(2). https://doi.org/10.5281/zenodo.19919777

[8] Hidayat, M. N. F., & Sofyan, H. (2026). Design and Implementation of an Integrated Web-Based Academic Management and Student Admission Monitoring System Using Django Framework. CODEX: Journal of Software Engineering, 1(1), 20–29. https://doi.org/10.33650/codex.v1i01.14454

[9] Shelke, K., Khan, Y., & Kapse, A. S. (2025). Student Management System: A Web-Based Solution for Academic Administration. International Journal of Scientific Research in Science and Technology, 12(3), 50–54. https://doi.org/10.32628/IJSRST251222719

[10] Rao, D. S., Madhumala, P., Pravallika, N., Sridevi, K., Neha, C., Dinesh, M., & Kumar, M. V. (2025). MERN Stack-Powered Academic Management Solution. International Journal of Progressive Research in Engineering Management and Science (IJPREMS), 5(4), 2736–2750. https://doi.org/10.58257/IJPREMS40593

[11] Rajapandian, P., & Kanchanashree, R. (2025). Online Course Registration System. International Scientific Journal of Engineering and Management

(ISJEM), 4(6). https://doi.org/10.55041/ISJEM04012

[12] Sudharshan, K. T. (2025). Student Request and Approval Management System. International Research Journal of Education and Technology (IRJEdT), 7(3). https://doi.org/10.70127/irjedt.vol.7.issue03.1238

[13] Nagarjuna, E., Chandini Sree, K., Likhith, K. L., Yasodha, P., Satyakavya, V., & Tanuj Kumar, D. (2026). Smart Academic Management System. International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE), 14(3). https://doi.org/10.15680/IJIRCCE.2026.1403080

[14] Reddii, V., Padala Krishna Sai Moksha, P., Rani Sri Lakshmi Sravya, & Sankar, E. (2026). A Unified Digital Student Portal for Integrating Academic and Administrative Services in Higher Education Institutions. International Journal for Research in Applied Science & Engineering Technology (IJRASET), 14(4), 5563–5574. https://doi.org/10.22214/ijraset.2026.80420

[15] Oliveros, C. J. R., Salcedo, J. L. Q., Tobias, A. V., & Melendez, M. L. D. (2026). Development of a Web-Based Student Portal System for Dasol Catholic School: Enhancing Accessibility and Academic Management. International Journal of Research and Innovation in Social Science (IJRISS), 10(3), 6632–6636. https://doi.org/10.47772/IJRISS.2026.100300479

[16] Fartitchou, M., Lamaakal, I., El Makkaoui, K., El Allali, Z., & Maleh, Y. (2025). BlockMEDC: Blockchain Smart Contracts System for Securing Moroccan Higher Education Digital Certificates. IEEE Access, 13, 39152–39172. https://doi.org/10.1109/ACCESS.2025.3546177