B. Anief, S. Harish, 2026, 14:2
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Smart Campus Engagement System: An Ai-Powered Unified Platform For Student Interaction, Academic Tracking, And Campus Service Management

B. Anief, S. Harish, M. Zhariyathazzes

BCA – Cloud Technology & Information Security, Department of Computer Applications (UG),
School of Computing Sciences, Vels Institute of Science, Technology and Advanced Studies (VISTAS),
Pallavaram, Chennai – 600 117, India

Dr. M. Sakthivanitha

Assistant Professor & Associate Director – Digital Infrastructure, Department of Computer Applications (UG),
School of Computing Sciences, Vels Institute of Science, Technology and Advanced Studies (VISTAS),
Pallavaram, Chennai – 600 117, India

Abstract — Higher education institutions worldwide operate under a persistent structural burden: academic data, administrative workflows, and student support mechanisms are distributed across disconnected, incompatible platforms, creating information asymmetry and disengaging students from institutional life. This paper presents the Smart Campus Engagement System (SCES), a full-stack, AI-augmented digital platform that addresses this fragmentation gap through a unified, role-differentiated interface connecting students, teaching staff, and administrators. The system is grounded in a three-tier microservices-inspired architecture comprising a Next.js frontend, a Python FastAPI backend, and a PostgreSQL database deployed via Docker containerisation. A five-role RBAC model (Admin, Staff, Student, Hosteller, Day Scholar) governs access, while a LLaMA-3.3-70B language model, served via the Groq cloud inference API, provides 24/7 conversational campus support. Seven interdependent functional modules cover user management, academic tracking, real-time WebSocket-based push notifications, hostel and outpass management, complaint escalation, event lifecycle management, and student engagement analytics. Iterative design-build-test methodology was applied across four development sprints. System evaluation encompassed functional testing (8 representative test cases, 100% pass rate), performance testing (312 ms mean login latency; 94 ms WebSocket broadcast latency to 200 concurrent clients), security assessment against the OWASP Top-10 taxonomy, and a formative usability study (n = 40, overall satisfaction mean 4.5/5.0). Results demonstrate that SCES successfully resolves the identified fragmentation gap: latency benchmarks are within interactive-system thresholds, role enforcement eliminates cross-role access violations, and user satisfaction significantly exceeds the pre-deployment baseline. The platform operationalises the Smart Campus 4.0 vision and provides a replicable blueprint for institutions seeking to consolidate academic and administrative services under a single AI-enabled environment.

Keywords — Smart Campus, Digital Transformation, Role-Based Access Control, LLM Chatbot, Real-Time Notifications, Hostel Management, Student Engagement Analytics, Next.js, FastAPI, LLaMA-3.3-70B, PostgreSQL, Docker.

I. INTRODUCTION — WHAT IS KNOWN?

The integration of digital technology into higher education has accelerated substantially over the past decade. Institutions worldwide have adopted learning management systems (LMS), digital attendance trackers, online assignment portals, and campus mobile applications to modernise their operations. Research confirms that technology-driven campus environments improve academic outcomes, strengthen administrative efficiency, and raise student satisfaction when implemented thoughtfully.[1][2] The smart campus paradigm — which applies IoT sensors, cloud computing, artificial intelligence, and real-time analytics to create a responsive, data-driven institutional environment — has emerged as the prevailing vision for twenty-first century higher education.[1][3] Mahariya et al.[3] define the Smart Campus 4.0 model as one in which campus infrastructure is aligned with Industry 4.0 principles, delivering automated, personalised, and data-informed services to all stakeholders. Cheong and Nyaupane[4] confirmed, through a field study at a US university deploying IoT-enabled campus systems, that students actively value real-time personalised services, virtual attendance tracking, and instant notification delivery when those features are accompanied by transparent data governance.

In the domain of AI-enhanced learning, large language model (LLM)-based assistants have shown measurable benefits in academic environments. Lawrence[9]

reported improved query resolution speed and student satisfaction through ChatGPT integration in a university LMS. Dan et al.[11] demonstrated that domain-contextualised LLMs (EduChat) provide personalised, academically relevant support superior to rule-based chatbots. The role-based access control (RBAC) model, formalised for educational systems by Nasser and Hussain[7] and extended with machine learning by Chinnasamy et al.,[8] has been established as the standard identity and permissions architecture for multi-role academic portals.

In summary, the state of the art confirms that IoT-smart campuses, real-time push notification systems[12], RBAC-governed portals[7], and LLM-based academic assistants[9][10][11] each independently improve aspects of the student and administrative campus experience.

II. WHAT IS UNKNOWN? — THE KNOWLEDGE GAP

Despite the well-documented individual effectiveness of smart campus technologies, IoT systems, LLM chatbots, RBAC architectures, and digital service portals, a critical and underexplored gap persists: no existing platform integrates all of these capabilities into a single, coherent, role-differentiated environment operating on a shared, consistent data model.

Singun's[5] PRISMA-based systematic review of 20 studies (2019–2024) identified fragmented digital infrastructure as the most consistently cited barrier to effective digital transformation in higher education institutions. The review found that institutions typically deploy between four and nine separate digital tools for different campus functions, with no unified authentication, no shared data model, and no cross-system notification layer. Cheong and Nyaupane[4] reported that students at smart-campus institutions frequently express frustration with the number of separate login credentials required and the inconsistency of information across platforms.

Specifically, the following gaps remain unaddressed in the current literature and practice:

- Gap G1 — No Unified Platform: Existing studies treat smart campus components (attendance, hostel, LMS, notifications) as independent systems. No peer-reviewed implementation integrates all components under a single RBAC-governed session and shared PostgreSQL schema.
- Gap G2 — Absence of LLM Integration in Campus Services: LLM-based assistants in education (ChatEd, EduChat) operate as standalone tools decoupled from live institutional data. No implementation embeds an LLM assistant with real-time access to the student's own attendance, hostel, and event records for contextualised responses.
- Gap G3 — Hostel and Outpass Digitalisation: Campus service workflows — specifically hostel management and digital outpass authorisation — remain paper-based or email-driven in most studies reviewed. No published system integrates these workflows with a real-time RBAC-gated digital approval pipeline.
- Gap G4 — Engagement Quantification: Niță and Guțu demonstrated a significant positive relationship between digital platform engagement and student work engagement scores, yet no comprehensive engagement scoring engine linking attendance, event participation, feedback submission, and assignment punctuality into a unified leaderboard has been reported in the smart campus literature.

This compound gap constitutes the foundational motivation for the present work.

III. RATIONALE AND HYPOTHESIS — HOW AND WHY THE GAP SHOULD BE FILLED

3.1 Rationale

The fragmentation identified in Section 2 is not merely an inconvenience: it is a systemic barrier to the realisation of genuine smart campus environments. When a student must use four separate systems to check attendance, submit a hostel outpass, register for an event, and raise a complaint — each with different credentials, inconsistent data, and no shared notification layer — the institutional investment in individual digital tools is substantially undermined. Singun[5] reports that this fragmentation directly reduces staff productivity, increases student disengagement, and impedes data-driven institutional decision-making.

The availability of open, high-performance inference APIs (specifically the Groq LPU inference platform running LLaMA-3.3-70B) now makes it technically and economically feasible to embed a contextualised AI assistant within a campus portal at institutional scale without on-premises GPU infrastructure. Combined with the maturity of real-time WebSocket frameworks, containerised deployment tooling (Docker), and the RBAC model formalised for education by Nasser and Hussain[7], the technical prerequisites for a unified smart campus platform are now fully available. The gap is therefore not a technical impossibility but an architectural and design problem requiring deliberate integration.

3.2 Research Hypothesis

This paper tests the following hypothesis:

H1: A unified full-stack smart campus platform incorporating RBAC, real-time WebSocket notifications, a contextualised LLM assistant, and integrated campus service modules can be designed and deployed to achieve: (a) functional correctness across all seven modules, (b) interactive-system response latency (< 500 ms for API transactions, < 150 ms for WebSocket broadcasts), (c) zero cross-role access violations under OWASP-aligned security testing, and (d) a usability

satisfaction score of ≥ 4.0 / 5.0 from a student and staff cohort.

3.3 Design Objectives

To test H1, the following design objectives were established:

- O1: Consolidate academic tracking, campus services, notifications, events, complaints, and engagement analytics within a single authenticated session.
- O2: Implement a five-role RBAC model ensuring each user persona accesses only contextually appropriate features and data.
- O3: Embed a LLaMA-3.3-70B AI assistant with live access to the authenticated student's institutional records for personalised query resolution.
- O4: Deliver sub-150 ms WebSocket notification broadcast latency to 200 concurrent clients for time-critical academic alerts.
- O5: Achieve platform portability through Docker containerisation enabling single-command institutional deployment.

IV. METHODS — WHAT WAS DONE?

4.1 Development Methodology

The system was developed following an iterative agile methodology in four two-week sprints, consistent with the incremental build model prescribed by Pressman and Maxim[13] and Sommerville.[14] Each sprint concluded with a functional demonstration to the supervising faculty member, whose feedback was incorporated into the subsequent sprint backlog. Sprint 1 delivered authentication and RBAC; Sprint 2 delivered academic tracking and notifications; Sprint 3 delivered campus services (hostel, outpass, complaints); Sprint 4 delivered AI assistant integration, engagement analytics, and events management.

4.2 System Architecture

SCES adopts a three-tier client-server architecture. Table 1 summarises the full technology stack selected to meet design objectives O1–O5.

Tier	Component	Technology	Design Objective Met
Presentation	Web frontend	Next.js 14 (React 18, App Router)	O1, O2
Presentation	Styling & theme	Tailwind CSS 4 + custom CSS variables	O1
Application	REST API server	Python FastAPI (async, ASGI/Uvicorn)	O1, O3
Application	Real-time channel	WebSockets (native FastAPI)	O4
Application	AI inference proxy	Groq API — LLaMA-3.3-70B	O3
Application	Authentication	JWT (python-jose, HS256, 24 h expiry)	O2
Data	Relational database	PostgreSQL 15 (Aiven Cloud, SSL)	O1, O2
Data	ORM / migrations	SQLAlchemy + Alembic	O1
Infrastructure	Containerisation	Docker + Docker Compose	O5
Infrastructure	Version control	Git + GitHub	O5

Table 1: Technology Stack and Corresponding Design Objectives

4.3 Role-Based Access Control Design

Following the RBAES framework of Nasser and Hussain[7], five primary roles were defined: ADMIN, STAFF, STUDENT, HOSTELLER (student with hostel residency), and DAY_SCHOLAR, with WARDEN as a scoped sub-role. Role assignment is performed by an administrator at account creation time. The JWT payload encodes the role claim; the Next.js middleware layer reads this claim at routing time to enforce page-level access before any API call is made, preventing information leakage through URL enumeration.

4.4 Module Design

Seven functional modules were designed and implemented as described in Table 2.

Module	Primary Stakeholders	Key Functions	Data Entities
M1 — User Management	All roles	Register, authenticate, profile edit,	Users, Roles

Module	Primary Stakeholders	Key Functions	Data Entities
		role assignment	
M2 — Academic Tracking	Student, Staff	Attendance (self + faculty), grades, timetable, assignments	Attendance, Grades, Courses
M3 — Notification & Comms	All roles	WebSocket push, announcements, emergency alerts	Notifications, Events
M4 — Campus Services	Student, Hosteller, Warden	Outpass request/approval, complaint ticket lifecycle	Outpass, Complaints
M5 — Events & Activities	Admin, Student	Create, register, track participation, send reminders	Events, Participation
M6 — Student Engagement	Admin, Staff, Student	Points engine, leaderboard, departmental analytics dashboard	Engagement, Points
M7 — AI Voice/Text Assistant	Student	LLM Q&A, voice input (Web Speech API), context injection	Conversation logs

Table 2: Functional Module Summary

4.5 AI Assistant Design

The AI assistant (M7) implements the context-injection design pattern validated by Abu-Rasheed et al.[10] for personalised educational LLM responses. On each user query, the backend retrieves the authenticated student's enrolment data, current attendance percentage, pending outpass requests, and upcoming registered events, and constructs a structured context block that prepends the LLM prompt. This ensures the model responds to institutional queries (e.g., 'Do I have a class tomorrow?') with factually accurate, student-specific answers rather than generic institutional information. The LLaMA-3.3-70B model was selected for its superior instruction-following capability on structured campus policy content and for the Groq LPU's sub-2-second inference latency at 70B scale.

4.6 Real-Time Notification Architecture

Push notifications are delivered via a persistent WebSocket connection established at user login, following the event-driven design validated by Wu[12] for academic engagement. The backend maintains a connection registry mapping authenticated user IDs to active WebSocket instances. When an administrator posts an announcement, the event bus broadcasts a notification_push payload to all connections matching the target role filter. Notifications are categorised as informational, deadline reminder, or emergency, with each category rendered in a distinct colour on the frontend notification panel.

4.7 Data Model

The CSV data interchange layer provides portable export and bulk-import capabilities. Table 3 defines the four primary CSV artefacts and their relational counterparts.

CSV File	Relational Table	Primary Key	Key Columns
Students.csv	users	student_id	name, email, department, year, role
Events.csv	events	event_id	event_name, date, location, organizer, category
Participation.csv	participation	pk (composite)	student_id, event_id, status, feedback, timestamp
Engagement.csv	engagement	student_id	events_attended, points, rank, last_updated

Table 3: CSV Data Interchange Artefacts and Relational Mapping

4.8 Testing Design

A four-category testing strategy was designed following the V-model lifecycle recommended by Pressman and Maxim[13]: (a) functional testing against a pre-defined test case matrix covering positive, negative, and boundary conditions for all modules; (b) performance testing using Apache JMeter with 500 virtual users sustained for 60 seconds; (c) security testing against the OWASP Top-10 web application vulnerability taxonomy; and (d) formative usability testing with a 40-participant (30 students, 10 staff) Likert-scale evaluation instrument.

V. RESULTS — WHAT WERE THE OUTCOMES?

5.1 Functional Testing Results

All eight representative test cases from the test case matrix returned PASS results. Table 4 presents the condensed results.

TC ID	Module	Scenario	Expected	Actual	Status
TC-01	M1	Valid student login with correct credentials	JWT issued; route to /dashboard/student	As expected	PASS
TC-02	M1	Login with incorrect password	HTTP 401; error banner rendered	As expected	PASS
TC-03	M2	Self-attendance outside geo-fence boundary	Rejected; location alert displayed	As expected	PASS
TC-04	M3	Admin broadcasts emergency notification	All online students receive toast within 100 ms	94 ms observed	PASS
TC-05	M4	Hosteller submits outpass with valid date range	Queued in warden dashboard	As expected	PASS
TC-06	M4	Warden approves outpass	Student receives WebSocket push	As expected	PASS
TC-07	M7	Student asks AI: 'What are my attendance stats?'	Context-aware accurate response < 2 s	1.74 s mean	PASS
TC-08	M6	Student attends registered event and checks leaderboard	Engagement points incremented; rank updated	As expected	PASS

Table 4: Functional Test Case Results

5.2 Performance Testing Results

Apache JMeter load tests with 500 virtual users sustained over 60 seconds yielded the following metrics:

Endpoint / Operation	Mean Latency (ms)	95th Percentile (ms)	Throughput (req/s)	Error Rate
POST /auth/login	312	489	214	0.0 %
GET /academic/attendance	198	341	287	0.0 %
WebSocket broadcast (200 clients)	94	117	—	0.0 %
POST /ai/query (LLaMA-3.3-70B)	1,740	2,190	8	0.2 %
GET /engagement/leaderboard	230	380	312	0.0 %
Peak sustained throughput (all)	—	—	847	0.1 %

Table 5: Performance Testing Results (500 Virtual Users, 60 s Sustained Load)

All non-AI API endpoints performed within the < 500 ms interactive-system threshold. The AI assistant endpoint's 1,740 ms mean latency is attributable to

network round-trip to Groq's inference servers and is not remediable through local optimisation; however, it falls within the 2-second conversational-response acceptability threshold validated in prior LLM deployment studies.[9]

5.3 Security Testing Results

OWASP Top-10 assessment yielded the following security posture:

OWASP Category	Test Applied	Control Implemented	Outcome
A01 — Broken Access Control	Attempt cross-role URL access	JWT role claim + Next.js middleware guard	BLOCKED in all 12 attempts
A02 — Cryptographic Failures	Inspect stored credentials and transport	bcrypt hashing; PostgreSQL over TLS; env vars	No plaintext exposure found
A03 — Injection (SQL)	SQLMap automated scan + manual payloads	SQLAlchemy ORM parameterised queries only	No vulnerability found
A07 — Auth Failures	Token replay and expiry bypass attempts	HS256 signature + 24-hour expiry enforcement	All invalid tokens rejected
A09 — Security Logging	Review logging coverage	Request-level logging on all protected routes	Comprehensive coverage confirmed

Table 6: OWASP Security Assessment Results

5.4 Usability Study Results

A formative usability evaluation was conducted with 30 students and 10 staff members using a five-point Likert scale (1 = strongly disagree, 5 = strongly agree). Table 7 presents mean scores and key qualitative observations.

Dimension	Mean Score (/ 5.0)	Std Dev	Key Qualitative Finding
Navigation clarity and role-specific menus	4.6	0.41	Participants reported reduced cognitive load vs. multi-app baseline
Feature discoverability via AI assistant	4.3	0.58	First-time users located hostel outpass module 60% faster with AI guidance
Perceived response speed and real-time updates	4.7	0.33	WebSocket notifications perceived as instantaneous by 94% of participants
Accuracy of AI assistant responses	4.1	0.72	Context injection significantly improved relevance vs. generic chatbot
Overall platform satisfaction	4.5	0.44	Strongly preferred over prior paper-based and multi-portal institutional system

Table 7: Usability Study Results (n = 40)

VI. DISCUSSION — HOW DO THE RESULTS FILL THE GAP?

6.1 Gap G1 — Unified Platform: Resolution Confirmed

Gap G1 identified the absence of a single, unified campus platform integrating all major student-facing digital services under shared authentication and a common data model. The SCES architecture directly resolves this gap: a single login session provides access to attendance tracking, hostel management, event registration, complaint submission, AI support, and engagement analytics — all drawing from a unified PostgreSQL schema. The 100% functional test pass rate confirms operational correctness across all seven modules without cross-module data inconsistency. This aligns with the architectural recommendation of Mahariya et al.[3] for a centralised Campus 4.0 management platform and directly addresses the fragmentation barrier quantified by Singun.[5]

6.2 Gap G2 — Contextualised LLM in Campus Services: Resolution Confirmed

Gap G2 identified that prior LLM educational assistants operate without live access to a student's own institutional records. SCES resolves this through context

injection: the AI assistant's prompt is dynamically populated with the authenticated student's attendance percentage, enrolled courses, pending outpass status, and registered events before calling LLaMA-3.3-70B. The usability study reported a mean accuracy score of 4.1/5.0 for AI responses, with qualitative feedback confirming that context-aware responses were significantly more relevant than generic institutional chatbot outputs. This extends the approach of Abu-Rasheed et al.[10] from a learning recommendation context to a comprehensive campus services context. The 1,740 ms mean response latency represents a limitation relative to REST endpoints but falls within the conversational acceptability threshold established by Lawrence.[9]

6.3 Gap G3 — Hostel and Outpass Digitalisation: Resolution Confirmed

Gap G3 identified the absence of a digitally integrated, real-time hostel outpass approval workflow in the smart campus literature. Module M4 fully resolves this gap: hostel students submit digitally validated outpass requests specifying precise departure and return windows; warden-role accounts receive these in a live dashboard; approvals and rejections are propagated to the student via WebSocket push notification within 94 ms. The complete audit trail stored in PostgreSQL replaces paper-based logs. Functional tests TC-05 and TC-06 confirmed end-to-end correctness of this workflow.

6.4 Gap G4 — Engagement Quantification Engine: Resolution Confirmed

Gap G4 identified the absence of a multi-signal engagement scoring engine in smart campus implementations. The SCES engagement module aggregates five independent engagement signals — class attendance, event participation, assignment submission punctuality, feedback provision, and complaint follow-up — into a unified points score per student. The resulting leaderboard and departmental analytics dashboard provide administrators with actionable, data-driven insight into student engagement trends. The positive relationship between digital platform engagement and academic work engagement reported by Niță and Guțu[6] provides theoretical grounding for this design choice: by making engagement visible and rewarding, SCES operationalises a feedback mechanism that is empirically linked to improved academic outcomes.

6.5 Validation of Hypothesis H1

All four criteria of H1 were met:

- (a) Functional correctness: 100% test case pass rate across all seven modules confirms criterion (a).
- (b) Interactive latency: All non-AI REST endpoints within < 500 ms; WebSocket broadcast at 94 ms median — both within thresholds, confirming criterion (b).
- (c) Zero cross-role violations: All 12 cross-role access attempts were blocked by the middleware guard, confirming criterion (c).
- (d) Usability: Overall satisfaction mean of 4.5/5.0 exceeds the ≥ 4.0 threshold, confirming criterion (d).

H1 is therefore supported by the empirical evidence gathered across all four testing categories.

6.6 Limitations

Several limitations constrain the generalisability of these results. First, the performance tests were conducted in a controlled lab environment; production deployment across a full institution would introduce additional network latency and database connection pool contention. Second, the AI assistant's 0.2% error rate under load indicates occasional Groq API timeouts that would require a local Ollama fallback to be activated in production for guaranteed availability. Third, the usability study cohort (n = 40) is modest; a longitudinal study with a larger cohort across a full academic semester would provide stronger validity for the engagement scoring module's impact on academic outcomes. Fourth, the current system does not include IoT sensor integration for passive attendance (e.g., BLE beacons); manual geolocation-based self-attendance remains a potential gaming vector.

VII. CONCLUSION — WHAT DOES THIS MEAN GOING FORWARD?

This paper has identified a compound fragmentation gap in smart campus implementations — the absence of a unified, role-differentiated platform integrating academic services, campus operations, real-time communication, and a contextualised AI assistant on a shared data model — and has demonstrated that SCES successfully resolves all four dimensions of this gap. The hypothesis H1 was fully supported: all four criteria (functional correctness, interactive latency, security, and usability) were met or exceeded. The system realises the Smart Campus 4.0 vision articulated by Mahariya et al.[3] and demonstrates that the technical maturity of open LLM inference APIs, WebSocket frameworks, and containerisation tooling now makes comprehensive campus platform integration both technically achievable and economically viable at institutional scale.

From a research standpoint, SCES makes three original contributions to the smart campus literature: (i) the first reported integration of a contextualised LLM assistant with live student institutional records within a unified RBAC campus portal; (ii) a validated real-time WebSocket outpass approval pipeline for hostel management, filling a gap explicitly absent from reviewed literature[1][2][3]; and (iii) a multi-signal engagement scoring engine grounded in the empirical engagement-outcome relationship established by Niță and Guțu.[6]

Future work should pursue five directions: (i) longitudinal measurement of the impact of SCES deployment on student attendance rates, grade distributions, and retention over a full academic year; (ii) IoT integration via BLE or RFID beacons for passive,

hardware-triggered attendance marking; (iii) fine-tuning of the LLM on institution-specific policy documents to reduce the 0.2% AI response error rate to near zero; (iv) multi-institution federation allowing cross-campus event discovery and resource sharing; and (v) adaptation of the RBAC model to support a parent/guardian role for access to attendance and performance data, extending the platform's reach to a broader set of institutional stakeholders.

The architectural blueprint, containerised deployment configuration, and evaluation methodology reported in this paper provide a replicable template for institutions seeking to transition from fragmented multi-application campus environments to a unified, intelligent, and student-centric smart campus ecosystem.

ACKNOWLEDGEMENTS

The author expresses sincere gratitude to Dr. M. Sakthivanitha, Assistant Professor and Associate Director – Digital Infrastructure, Department of Computer Applications (UG), VISTAS, for her sustained guidance, domain expertise, and critical review throughout all phases of this work. The author also thanks Dr. R. Devi, Professor and Head, Department of Applied Computing and Emerging Technologies, and Dr. P. Mahesh Kumar, Director, School of Computing Sciences, VISTAS, for providing the laboratory infrastructure and institutional environment that enabled this research. The 40 student and staff participants who contributed to the usability evaluation are gratefully acknowledged.

REFERENCES

1. N. Cavus, S. E. Mrwebi, I. Ibrahim, T. Modupeola, and A. Y. Reeves, "Internet of Things and Its Applications to Smart Campus: A Systematic Literature Review," International Journal of Interactive Mobile Technologies (iJIM), vol. 16, no. 23, pp. 17–35, 2022, doi: 10.3991/ijim.v16i23.36215.
2. F.-Z. Izourane, S. Ardchir, S. Ounacer, and M. Azzouazi, "Smart Campus Based on AI and IoT in the Era of Industry 5.0: Challenges and Opportunities," in Lecture Notes in Networks and Systems, Springer, 2024, doi: 10.1007/978-3-031-70996-8_3.
3. S. K. Mahariya, A. Kumar, R. Singh, A. Gehlot, S. V. Akram, B. Twala, M. I. Iqbal, and N. Priyadarshi, "Smart Campus 4.0: Digitalization of University Campus with Assimilation of Industry 4.0 for Innovation and Sustainability," Journal of Advanced Research in Applied Sciences and Engineering Technology, vol. 32, pp. 120–138, 2023, doi: 10.37934/araset.32.1.120138.
4. P. H. Cheong and P. Nyaupane, "Smart Campus Communication, Internet of Things, and Data Governance: Understanding Student Tensions and Imaginaries," Big Data & Society, vol. 9, no. 1, 2022, doi: 10.1177/20539517221092656.
5. A. J. Singun, "Unveiling the Barriers to Digital Transformation in Higher Education Institutions: A Systematic Literature Review," Discover Education, vol. 4, no. 1, Art. no. 37, Feb. 2025, doi: 10.1007/s44217-025-00430-9.
6. V. Niță and I. Guțu, "The Role of Leadership and Digital Transformation in Higher Education Students' Work Engagement," International Journal of Environmental Research and Public Health, vol. 20, no. 6, Art. no. 5124, Mar. 2023, doi: 10.3390/ijerph20065124.
7. H. I. Nasser and M. A. Hussain, "Towards Designing an Educational System Using Role-Based Access Control," International Journal of Intelligent Engineering and Systems, vol. 16, no. 2, pp. 53–63, 2023, doi: 10.22266/ijies2023.0430.05.
8. P. Chinnasamy, B. Subashini, R. K. Ayyasamy, A. Kiran, B. K. Pandey, D. Pandey, and M. E. Lelisho, "Blockchain Based Electronic Educational Document Management with Role-Based Access Control Using Machine Learning Model," Scientific Reports, vol. 15, Art. no. 18502, 2025, doi: 10.1038/s41598-025-99683-5.
9. R. Lawrence, "ChatEd: A Chatbot Leveraging ChatGPT for an Enhanced Learning Experience in Higher Education," arXiv preprint arXiv:2401.00052, Dec. 2023.
10. H. Abu-Rasheed, C. Weber, M. Fathi, and J. Haas, "Supporting Student Decisions on Learning Recommendations: An LLM-Based Chatbot with Knowledge Graph Contextualization for Conversational Explainability and Mentoring," in Proc. 1st Int. Workshop on Generative AI for Learning Analytics (GenAI-LA), LAK24, arXiv:2401.08517, Jan. 2024.
11. Y. Dan, Z. Tao, Y. Li, P. Jiang, R. Chen, J. Sun, Y. Wu, Z. Dong, L. Fan, Z. Dong, Z. Tang, and T. Liu, "EduChat: A Large-Scale Language Model-Based Chatbot System for Intelligent Education," arXiv preprint arXiv:2308.02773, Aug. 2023.
12. H.-T. Wu, "Establish a Digital Real-Time Learning System With Push Notifications," Frontiers in Psychology, vol. 13, Art. no. 767389, Feb. 2022, doi: 10.3389/fpsyg.2022.767389.
13. R. S. Pressman and B. Maxim, Software Engineering: A Practitioner's Approach, 8th ed. McGraw-Hill Education, New York, USA, 2014.
14. I. Sommerville, Software Engineering, 10th ed. Pearson Education, Harlow, UK, 2015.