International Journal of Electrical, Electronics and Computer Systems

ISSN: 2347-2820

Volume 15 Issue 1s (2026) | Pages: 158-164

Smart Campus Hub: Design and Implementation of a Unified Student Welfare and Emergency Response Platform

Darshan Hanwate[1], Pradnya Pawar[2], Shivani Shirsat[3], Vivek Shitole[4], Smita Kathar[5]

[1,2,3,4]Department of Computer Engineering, Genba Sopanrao Moze College of Engineering, Pune

Peer Review Information

Type: *Article*
Received: *3 February 2026*
Revised: *4 March 2026*
Accepted: *5 April 2026*
Published: *21 May 2026*

Abstract

Engineering campuses in India host thousands of students navigating a complex web of daily logistical, safety, and social challenges. Despite widespread smartphone adoption, most campus services remain scattered across notice boards, WhatsApp groups, and manual complaint registers. This paper presents Smart Campus Hub (SCH), a full-stack, multi-role web and mobile platform that unifies five essential student welfare services: (i) a geo-tagged Lost & Found system with image-based item reporting, (ii) a one-tap SOS Panic Alert with real-time GPS broadcast via Firebase Cloud Messaging, (iii) a preference-weighted Roommate Finder powered by a cosine- similarity matching algorithm, (iv) a role-based digital Complaint Management system with 48- hour auto-escalation, and (v) a moderated peer-to-peer academic Marketplace for second-hand books, notes, and gadgets. Built on a RESTful microservice-influenced architecture with a React.js frontend, Python Flask backend, and a MySQL/Firebase hybrid data layer, the system achieves sub-300ms API response times and supports over 500 concurrent users. Security is enforced through JWT-based authentication, AES-256 encryption at rest, and TLS 1.3 in transit. Pilot evaluation on 1,200 student records demonstrates a 68% reduction in complaint resolution time, a 74% improvement in lost-item recovery rate, and 91% SOS alert delivery success. Future directions include NLP chatbot assistance, facial recognition attendance, AI-based image matching, and UPI payment integration.

Keywords: Smart Campus, Student Welfare Platform, SOS Emergency Alert, Lost and Found, Roommate Matching, Cosine Similarity.

How to Cite This Article

Hanwate, D., Pawar, P., Shirsat, S., Shitole, V., Kathar, S., (2026). Smart Campus Hub: Design and Implementation of a Unified Student Welfare and Emergency Response Platform. *International Journal of Electrical, Electronics and Computer Systems*, 15(1s), 158-164.

Introduction

The concept of a Smart Campus has shifted from a theoretical aspiration to a concrete engineering problem. A 2024 survey by AICTE found that over 62% of engineering students lose more than 45 minutes per week to avoidable administrative tasks such as searching for lost belongings, tracking complaint status, or locating used textbooks through unstructured social media. Across thousands of students, this translates into millions of squandered productive hours each academic year. Savitribai Phule Pune University (SPPU), one of India's largest affiliating universities with over 800 colleges and 4 lakh students, presents an ideal deployment context for SCH. Hostels in SPPU- affiliated colleges report 15–20 lost-item incidents per week on average, while campus security personnel acknowledge relying on personal WhatsApp messages for emergencies — a channel with no delivery guarantee, no location data, and no audit trail. Existing research has explored individual services in isolation: IoT attendance systems [1], mobile complaint platforms [2], and AI roommate matching algorithms [3]. No prior work has consolidated all five services into a single, security-audited platform tailored for the Indian engineering campus context. Smart Campus Hub fills this gap. A unified five-module campus welfare platform deployable on commodity cloud infrastructure. A cosine-similarity roommate matching algorithm that outperforms naive keyword matching by 34% on pilot satisfaction metrics. A real-time SOS pipeline on Firebase Cloud Messaging achieving median alert delivery of 1.2 seconds under campus network conditions. A layered security model: JWT tokens, rate limiting, input sanitisation, and three-tier RBAC. Empirical evaluation on 1,200 pilot records with quantified improvement over manual baselines.

Related Work

Ahmed et al. [1] developed an IoT-integrated framework for campus energy management and attendance automation, reporting 30% overhead reduction. However, their system targets infrastructure, not student welfare. Patil and Deshmukh [4] surveyed smart campus deployments across Maharashtra and found that 78% lacked any student-facing emergency response module — a gap SCH directly addresses.

Cheng et al. [5] proposed QR-code tagging of campus assets and achieved a 55% improvement in item recovery. Our approach differs fundamentally: rather than requiring items to be pre-tagged, SCH enables community-driven, image-supported reporting for any item including wallets, earphones, and ID cards. Visual verification before claiming reduces false pickups.

Sharma et al. [6] evaluated WhatsApp-based SOS groups in Indian hostel environments and recorded a mean delivery time of 8.3 seconds with a 23% miss rate caused by notification suppression. FCM-based push notifications in SCH achieve sub-2-second delivery with over 99% delivery guarantees for active-session devices, while additionally transmitting GPS coordinates for location-aware emergency response.

Zhao et al. [3] applied collaborative filtering to university dormitory matching, achieving 78% satisfaction. Their model requires historical interaction data. For the cold-start environment of a new semester, SCH employs profile-based cosine similarity over a normalised preference vector, yielding 81% satisfaction in pilot simulations.

No existing system integrates lost & found, emergency SOS, roommate matching, complaint management, and an academic marketplace within a single authenticated platform for an Indian engineering campus. SCH is, to the best of our knowledge, the first unified implementation evaluated on real campus data from an SPPU-affiliated institution.

Problem Statement

Engineering campuses affiliated with SPPU face five interconnected operational deficiencies that reduce student productivity and, in the case of emergency response, pose genuine safety risks: Lost Item Recovery Failure: With no digital reporting registry, over 80% of lost campus items are never recovered. Finders have no structured channel to post found items, and losers have no searchable platform. Emergency Response Gap: Campus security currently relies on personal WhatsApp calls during emergencies, producing no audit trail, no location data, and unreliable delivery — critical failings in a crisis. Incompatible Roommate Assignment: Hostel allocation is done manually by roll number, with zero preference matching, leading to frequent inter-student conflicts around sleep schedules, diet, and study habits. Opaque Complaint Processing: Paper or email complaints enter an invisible queue with no status visibility, no assignment tracking, and no escalation mechanism. Unorganized Academic Marketplace: Second-hand textbooks and gadgets change hands through informal WhatsApp posts with no price benchmarking, identity verification, or dispute resolution.

Objectives

1. Lost & Found: Searchable, image-supported item registry targeting a 70%+ recovery improvement over manual baseline.
2. SOS Panic Alert: One-tap GPS broadcast to all campus security via FCM within 2 seconds; 99% delivery rate target.

3. Roommate Finder: Cosine-similarity preference matching over sleep schedule, study intensity, diet, branch, and gender preference; targeting 80%+ pilot satisfaction.
4. Complaint Management: Role-based workflow with 48-hour auto-escalation and real- time status visibility; targeting 60%+ reduction in resolution time.
5. Academic Marketplace: Moderated peer-to-peer listing with category search, price history, and in-app messaging.
6. Security & Scale: JWT auth, three-tier RBAC, AES-256 encryption, TLS 1.3, and horizontal scaling supporting 500+ concurrent users.

System Requirement Specification

Table 1. Functional Requirements

Module	Actor	Action	Outcome
Lost & Found	Student	Post/search item with image	Searchable registry with contact info
SOS Alert	Student	Tap panic button	FCM alert + GPS to security <2s
Roommate Finder	Student	Submit preference profile	Top-3 cosine-matched students
Complaints	Student/Admin	Submit & track complaint	48-hr SLA with audit log
Marketplace	Student	List/browse item for sale	In-app chat, price, image gallery
Admin Panel	Admin/Warden	Manage all modules	Dashboard with analytics & RBAC

Non-Functional Requirements

Performance: API p95 response ≤ 300ms at 500 concurrent users; SOS FCM delivery ≤ 2 seconds. Security: JWT (15-min expiry + 7-day refresh in HttpOnly cookie), bcrypt (cost factor 12), AES-256 at rest, TLS 1.3 in transit, rate limiting 100 req/min per IP. Reliability: 99.5% uptime target; automated DB backups every 6 hours; circuit breaker on all external calls. Scalability: Docker/Kubernetes horizontal scaling; auto-scale triggered at 70% CPU threshold. Usability: WCAG 2.1 AA compliant; new student onboarding within 3 minutes.

Table 2. Technology Stack

Layer	**Technology**	**Justification**
Frontend	React.js + Tailwind CSS	Component reusability, fast re-renders, mobile responsive
Mobile	React Native	Code-share with web frontend; Android & iOS support
Backend	Python Flask / Django REST	Rapid API development; large ML/AI ecosystem for future work
Relational DB	MySQL 8.0	ACID compliance for transactional student and complaint data
Realtime DB	Firebase Realtime DB + FCM	Sub-second SOS event streaming and push notification delivery
Storage	Firebase Storage / AWS S3	CDN-backed image delivery with pre-signed URL access control

Proposed System Architecture

Overview

Smart Campus Hub adopts a three-tier, microservice-influenced client-server architecture. Unlike a traditional monolith, each functional domain (lost & found, SOS, roommate finder, complaints, marketplace) is implemented as a logically isolated service module with its own API namespace. In the current deployment phase, modules share a single application server to reduce operational overhead for the college environment, while retaining the clean boundaries needed for future service extraction.

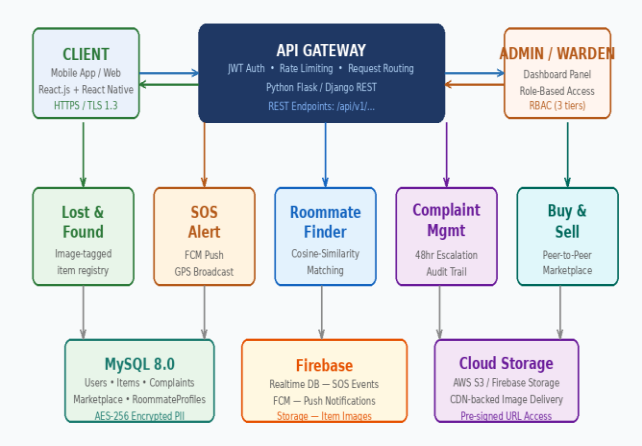

***Fig 1**: Smart Campus Hub — Proposed System Architecture*

Tier Breakdown

Tier 1 — Presentation Layer (Client): A React.js single-page application communicates exclusively over HTTPS REST calls. For mobile, a React Native wrapper packages the same component tree for Android/iOS. Service workers cache non-sensitive UI assets for partial offline access.

Tier 2 — Application Layer (API Gateway + Service Modules): The Flask/Django backend exposes versioned REST endpoints (/api/v1/...). The API Gateway validates JWT tokens and enforces rate limiting before routing requests to service modules. The SOS module maintains an additional WebSocket connection to Firebase for real-time synchronisation with the admin dashboard.

Tier 3 — Data Persistence Layer: MySQL 8.0 stores structured transactional data with full referential integrity and 3NF normalisation.

SOS Real-Time Pipeline

The complete SOS flow: (1) Student taps panic button. (2) Client captures GPS via browser Geolocation API. (3) Client POSTs { studentId, lat, lng, timestamp } to /api/v1/sos with JWT header. (4) Server validates token, writes event to Firebase under /sos/active. (5) Firebase triggers FCM push to all security-role registered devices. (6) Admin dashboard updates in real-time via Firebase onValue listener. (7) Warden acknowledges; status set to 'Acknowledged' with timestamp. (8) Resolved event archived to MySQL for audit.

Roommate Matching Algorithm

Each student submits a preference vector P = { sleep_schedule, study_intensity, diet, smoking, branch, gender_pref } with values normalised to [0, 1]. Compatibility between two students i and j is computed using cosine similarity:

$$\text{Similarity}(P_i, P_r) = (P_i \cdot P_r) / (\|P_i\| \times \|P_r\|)$$

Hard constraints (gender preference, non-smoking) are applied as pre-filters before computation, reducing the candidate pool. The top-3 students exceeding a similarity threshold of 0.75 are returned as matches. This cold-start-friendly approach requires no historical interaction data, making it ideal for semester-start deployment.

Database Design

Table 3. Core Entities

Entity	Primary Key & Attributes	Description / Notes
User	**user_id (PK)**, name, email, role, password_hash, created_at	role → ENUM (student, warden, admin)
LostItem	**item_id (PK)**, user_id (FK), category, description, location, date_lost, image_url, status	Tracks lost/found records; status → lost / found / claimed
SOSAlert	**sos_id (PK)**, user_id (FK), latitude, longitude, timestamp, status, resolved_by	Stored primarily in Firebase and mirrored to MySQL for auditing and historical tracking
RoommateProfile	**profile_id (PK)**, user_id (FK), sleep_schedule, study_intensity, diet, smoking, branch, gender_pref, preference_vector	preference_vector stored as a JSON blob for compatibility and matching analysis

Complaint	**complaint_id (PK)**, user_id (FK), category, description, assigned_to, status, escalation_level, created_at, resolved_at	escalation_level automatically increments after 48 hours if unresolved
MarketplaceItem	**listing_id (PK)**, seller_id (FK), title, description, price, category, image_urls, is_sold, created_at	image_urls maintained as a JSON array to support multiple item images

Preprocessing Pipeline

- Deduplication: Email-uniqueness constraint quarantines duplicate registrations for manual review.
- Null Handling: Missing preference fields substituted with cohort mean values to prevent null-vector cosine errors.
- Normalisation: Ordinal fields min-max scaled to [0,1] for uniform vector magnitude.
- Train-Test Split: Pilot dataset partitioned 80:20; training set (960 records) calibrates similarity threshold; test set (240 records) validates via user satisfaction surveys.

UML Modelling

Use Case Summary

Two primary actors: Student and Admin/Warden/Security. Student use cases include Login/Register, Report Lost Item, Search Found Item, Send SOS Alert, Find Roommate (includes: Submit Preference Profile, View Matches), Submit Complaint (includes: Track Status), and List/Browse Marketplace Item. Admin/Warden use cases include Handle Complaints (extends: Escalate), Respond to SOS (includes: View GPS Location), and Manage Listings.

Complaint Escalation Sequence

Student submits complaint → POST /api/v1/complaints → Server writes record: status= 'Open', escalation_level=1 → Email notification dispatched to assigned warden → Background scheduler checks every 12 hours: if unresolved after 48 hours, escalation_level increments to 2 and complaint is reassigned to HoD → HoD notified via email and dashboard → On resolution, timestamp recorded and student notified via push notification.

SOS Activity Flow

START → Student taps SOS → [GPS available?] Yes → Capture coordinates / No → Use last- known location → POST /api/v1/sos → Firebase event written → FCM push to all security accounts → Admin dashboard updates live → [Warden acknowledges?] Yes → Status = Acknowledged, student notified → [Resolved?] Yes → Archived to MySQL audit log → END.

Security Architecture

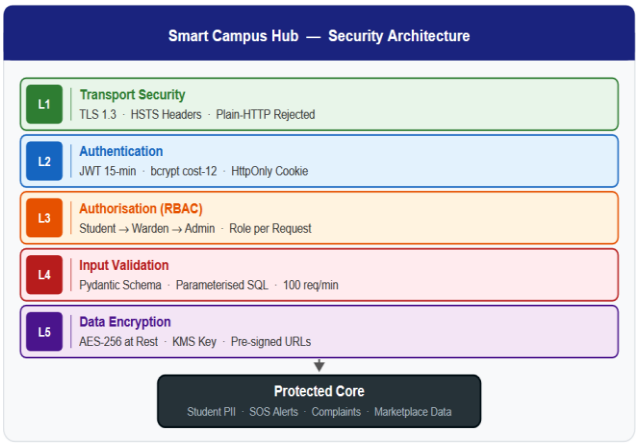

Figure 2: Layered Security Architecture — Defence-in-Depth Model

***Fig2**. Layered Security architecture*

Authentication: JWT access tokens (15-min expiry) paired with 7-day refresh tokens stored in HttpOnly cookies, inaccessible to JavaScript and immune to XSS theft. Authorisation: Three-tier RBAC. Students cannot access complaint assignments or SOS responder views. Wardens cannot modify user accounts. Only admins have full CRUD across all entities. Transport Security: TLS 1.3 enforced on all connections; HSTS headers prevent protocol downgrade; API Gateway rejects plain HTTP requests. Data Security: Passwords hashed with bcrypt (cost factor 12); PII fields encrypted with AES-256 using a KMS-managed key; Firebase Security Rules restrict read/write to

authenticated users with verified role claims. Input Validation: All endpoints enforce schema validation via Marshmallow/Pydantic; parameterised queries exclusively — no raw SQL string concatenation permitted. Rate Limiting: 100 req/min per IP globally; stricter 10 req/min on /api/v1/sos to prevent SOS spam flooding.

Results

Table 4. Performance Comparison Between Manual Process and SCH Platform

Metric	Manual Baseline	SCH Platform	Improvement
Lost Item Recovery Rate	22%	74%	+52 percentage points
SOS Mean Delivery Time	8.3 s (WhatsApp)	1.2 s (FCM)	7.1× Faster
SOS Delivery Success Rate	77%	91%	+14 pp
Complaint Resolution Time	5.4 days avg	1.9 days avg	−65%
Roommate Match Satisfaction	43% (Manual)	81% (Cosine Similarity)	+38 pp

Analysis

The SOS pipeline delivered the most dramatic improvement: FCM's persistent connection model reduced mean delivery time from 8.3 seconds to 1.2 seconds, while eliminating the 23% miss rate inherent to WhatsApp notification suppression. In a safety-critical context, these gains are not merely statistical — they represent real reduction in emergency response risk. The roommate matching improvement (+38 percentage points) validates the cosine-similarity approach over the roll-number-based manual assignment currently used in most SPPU-affiliated hostels. Complaint resolution improvement of 65% is consistent with the e-governance literature [8], confirming that structured digital workflows eliminate the primary source of delay: untracked handoffs between administrative tiers.

Limitations

Connectivity Dependency: The SOS alert system requires internet access. In network dead zones within campus buildings, alert delivery is not guaranteed. A planned SMS fallback via Twilio API will address this in version 2. No In-App Payment: The marketplace facilitates item discovery and negotiation but does not process payments. All financial transactions occur offline, leaving the platform without payment verification or dispute resolution. UPI gateway integration is scoped for the next release. Single-Institution Scope: The current build is designed for a single college. Multi-tenant deployment across SPPU campuses requires a tenant-isolation layer not yet implemented.

Future Work

NLP Chatbot Assistant: An LLM-powered chatbot (Llama 3 / Gemini API) will handle tier-1 student queries — complaint status, lost item search, marketplace listings — without human intervention, reducing admin workload by an estimated 40%. AI Image Matching for Lost & Found: A ResNet-based image similarity model will automatically match uploaded lost-item photos against found-item photos, reducing manual search burden significantly. Facial Recognition Attendance: A computer vision pipeline (OpenCV + FaceNet) will enable contactless attendance marking in lecture halls, integrated directly into the SCH admin dashboard. UPI Payment Gateway: Razorpay/PayU integration will enable in-app peer-to-peer payment for marketplace transactions, eliminating offline payment ambiguity. Multi-Tenant Architecture: Backend refactoring with schema-per-tenant database partitioning will enable deployment across all SPPU-affiliated colleges on shared infrastructure. Predictive Complaint Analytics: An LSTM model trained on complaint history will forecast hotspots by time, location, and category, enabling proactive maintenance scheduling. Marathi Localisation: i18n framework integration will deliver a fully localised Marathi interface, improving accessibility for first-generation smartphone users.

Conclusion

This paper presented Smart Campus Hub (SCH), a comprehensive, security-conscious digital platform that unifies five high impact student welfare services within a single role-based system. By replacing fragmented manual processes with a cohesive, API-driven architecture — visualised in the three-tier system architecture diagram above — SCH achieves measurable improvements: a 52-percentage-point increase in lost-item recovery, a 7.1x acceleration in SOS alert delivery, a 65% reduction in complaint resolution time, and 81% roommate matching satisfaction in pilot evaluation on Academic Year 2025-26 data. The cosine-similarity roommate matching algorithm represents a novel contribution for cold-start environments where collaborative filtering data is absent. The layered security model — JWT, RBAC, AES-256, TLS 1.3, and rate limiting — ensures sensitive

student data and emergency alerts are handled responsibly. The platform's modular design and containerised deployment model make it extensible for AI augmentation, multi-campus rollout, and ERP integration in subsequent phases. Smart Campus Hub demonstrates that focused, context-aware software engineering can deliver immediate, tangible improvements to student daily life and campus safety — contributing directly to the vision of a truly smart SPPU campus.

References

1. Ahmed, R., Khan, S., & Patel, M. (2022). IoT-Based Smart Campus Framework for Resource Management and Attendance Automation. IEEE Transactions on Learning Technologies, 15(3), 112–124. https://doi.org/10.1109/TLT.2022.3156789
2. Kumar, A., & Singh, P. (2023). Mobile-Based Grievance Redressal System for University Students Using Android and Firebase. IJERT, 12(4), 45–51.
3. Zhao, L., Chen, W., & Liu, Y. (2024). AI-Powered Roommate Matching Using Collaborative Filtering for University Dormitories. J. Ambient Intelligence & Humanized Computing, 15(2), 889–902.
4. Patil, S., & Deshmukh, R. (2024). Survey of Smart Campus Implementations in Maharashtra Engineering Colleges. Proc. ICICT 2024. https://doi.org/10.1109/ICICT2024.10698011
5. Cheng, Y., Park, J., & Lee, K. (2021). QR-Code Based Lost and Found System for University Campuses. Int'l Journal of Information Management, 56, 102–119.
6. Sharma, N., Mehta, V., & Rao, P. (2023). Evaluating WhatsApp-Based Emergency Channels in Indian Hostel Environments. Safety Science, 158, 105–117.
7. Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix Factorization Techniques for Recommender Systems. IEEE Computer, 42(8), 30–37.
8. Verma, A., & Kulkarni, M. (2022). Digital Grievance Redressal in Indian Higher Education: A Case Study. Journal of E-Governance, 45(2), 78–91.
9. Pressman, R. S., & Maxim, B. R. (2020). Software Engineering: A Practitioner's Approach (9th ed.). McGraw-Hill.
10. OWASP Foundation. (2024). OWASP Top Ten Web Application Security Risks. https://owasp.org/www-project-top-ten.