



iJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14

Issue: X

Month of publication: October 2026

DOI:

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com

NAKSHA: Networked Analytics for Knowledge and Spatial Hub for Academics

Himanshu Kumar¹, Abdul Rahil², Sahil Wani³, Harsh Kanchewar⁴, Ameya Fating⁵, Mayur Jadhao⁶, Sujal Salve⁷, Shyam Naik Mudu⁸

^{1, 2, 3, 4, 5, 6, 7, 8}Department of Computer Science and Engineering, School of Engineering and Technology, G H Rasoni University, Amravati, India

Abstract: Indian colleges still record attendance largely by roll call or paper sheets, which wastes lecture time, allows proxy marking and leaves the data unused until month-end, when struggling students are already far behind. This paper presents NAKSHA, a web system that needs no sensors or dedicated devices. A classroom screen shows a QR code that is replaced about every 20 seconds and expires after 90 seconds; students scan it with their own phones, and each phone is bound to a single roll number on first use. Identity, classroom and time are all decided by a database function, never by the phone. Each check-in updates one attendance table that drives both a live 2.5D map of classroom occupancy, which we describe as a digital shadow rather than a digital twin, and a mentor dashboard that groups students into support-framed bands from the last 14 class days. The system runs entirely on free hosting tiers using React and Supabase with row-level security, stores no protected attributes, and imports students only with recorded consent. We describe the design, the threats it addresses and our functional evaluation, and outline an explainable XGBoost and SHAP model as the next stage.

Keywords: QR code attendance, early warning system, digital shadow, learning analytics, row-level security, data privacy.

I. INTRODUCTION

Attendance is often the first visible sign that a student is in difficulty. A learner who stops coming to class may be facing financial pressure, illness, or a subject they can no longer follow, and in most Indian universities a fall below 75 % also blocks entry to the final examination. Yet the way attendance is taken has barely changed: a teacher calls sixty roll numbers, or a sheet travels around the room. Both methods cost several minutes per lecture, both are easy to game by answering or signing for a friend, and the resulting records are rarely examined before the monthly report.

Hardware solutions such as fingerprint readers, face recognition terminals and RFID gates reduce proxy marking but require a device in every room, ongoing maintenance and, in the biometric case, the collection of highly sensitive personal data. At the same time, nearly every student now carries a camera phone and nearly every classroom has a projector or laptop. This work asks whether those two existing resources are enough to produce attendance that is fast, hard to fake, visible in real time and useful to mentors for early support.

The work addresses Smart India Hackathon problem statement SIH25102 (AI-based dropout prediction and counselling), with SIH25016 (automated attendance monitoring) as a closely related target. Our contributions are: (i) a QR self check-in protocol that combines short-lived tokens with phone binding and moves every trust decision to the database; (ii) a single-source data design in which one check-in row simultaneously updates a live campus map and a mentor dashboard; (iii) a support-band rule that avoids two common counting errors; and (iv) a privacy-first deployment built only on permanent free tiers.

II. RELATED WORK

Macfadyen and Dawson [1] showed that plain LMS activity counts were enough to flag most students who later failed a course, suggesting that regular, ordinary data matters more than exotic data for early warning. Cortez and Silva [2] predicted secondary school grades from school records using tree ensembles and released the widely used UCI Student Performance dataset. Yao et al. [3] used smart card logs of more than 18,000 university students and found that regular daily routines were associated with better grades, while Das Swain et al. [4] linked WiFi-derived collocation to academic outcomes. These studies confirm that campus behaviour is informative, but they also show how easily such data turns into surveillance.

For explainable prediction, XGBoost [5] performs strongly on tabular records, and SHAP [6] attributes each individual prediction to its inputs, which is what a mentor needs in practice. On the infrastructure side, the West Cambridge project [7] and a Swedish university building twin [8] demonstrate rich live building views but rely on sensors whose installation and upkeep proved costly.

Kritzinger et al. [9] distinguish a digital model, a digital shadow (automatic data flow from physical to digital only) and a digital twin (flow in both directions); many systems labelled twins are in fact shadows. Finally, the 2021 investigation of a commercial risk-scoring product by The Markup [10] found that Black students were labelled high risk several times more often than white students, partly because race was a direct input. India's Digital Personal Data Protection Act, 2023 [11] adds duties of purpose limitation and parental consent for minors.

We found little work that joins a zero-hardware, tamper-resistant collection method to both a live campus view and an early warning list from one shared record, while treating privacy as a design constraint. NAKSHA targets that gap.

III. SYSTEM DESIGN

A. Architecture

Fig. 1 summarises the design. Three inputs — student phones, the teacher's QR screen and a mentor's CSV roster import — write to a single Supabase project that provides PostgreSQL with PostGIS, authentication, WebSocket change feeds and callable SQL functions. Three views read from it: a live campus map, a mentor dashboard and a personal student page. Because the browser talks to the database directly through row-level security policies, no custom application server is needed; the only planned custom service is the machine learning component.

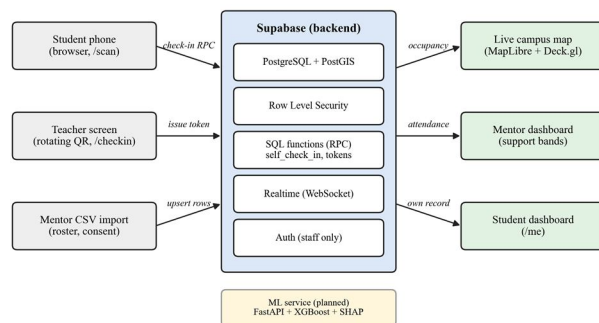


Fig. 1 Overall architecture of NAKSHA.

B. One Source of Truth

There is exactly one attendance table. A database trigger keeps an aggregate occupancy count on each classroom row in step with that day's check-ins, and the public map reads only that aggregate. The mentor dashboard reads the attendance rows themselves, under staff-only policies. Since both views are derived from the same insert, they cannot drift apart. The student table deliberately has no gender, caste, religion or race column, which guarantees that no future model can use these attributes. The schema evolved through nine ordered SQL migrations, so any instance can be rebuilt reproducibly.

IV. TAMPER-RESISTANT QR CHECK-IN

Our first prototype printed one fixed QR code per room. It failed in an obvious way: a photo of the code, shared in a class chat, let absent students mark themselves present. The final protocol, shown in Fig. 2, closes this and related gaps.

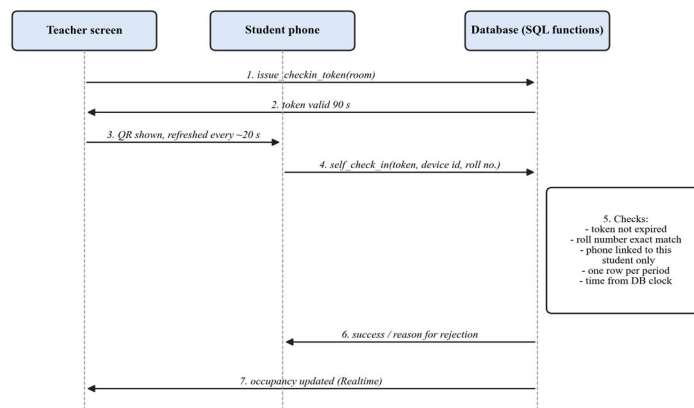


Fig. 2 Sequence of a self check-in.

A. Rotating Tokens

The teacher's screen requests a new random token from the database roughly every 20 seconds and encodes it in the QR code. Each token is valid for 90 seconds, which is long enough for a student who scans just before a refresh to type a roll number on first use, but short enough that a forwarded image quickly becomes useless. The classroom is looked up from the token on the server, so editing the link cannot redirect a check-in to another room.

B. Phone Binding

On first use the browser generates a random device identifier and keeps it in local storage. The first successful check-in binds that identifier to the student. From then on the phone may only check in that student, and that student cannot be claimed from another phone. A mentor can remove the binding when a phone is lost or replaced. This blocks the common case of one student marking several friends from a single handset.

C. Server-Side Checks

A single security-definer SQL function performs every check: the token exists and has not expired; the roll number matches exactly; the binding rules hold; the current timetable period is computed in Indian Standard Time; and at most one row per student per period is written, stamped with the database clock. An earlier version matched roll numbers with a pattern operator, so an input such as "NKS%" would have checked in whichever student matched first; replacing it with exact matching was one of several fixes found by deliberately attacking our own system. This function is the only operation open to anonymous users.

V. LIVE MAP AND SUPPORT BANDS

A. Campus Digital Shadow

The map uses MapLibre GL JS with a Deck.gl overlay in a tilted 2.5D view. Buildings are extruded to a fixed height, and each classroom appears as a bar whose height and colour (green to amber to red) both encode occupancy, so the view remains readable for colour-blind users. The base style is a plain background layer with no external tile server. Change feeds push each check-in to every open map within seconds. Data only flows from campus to screen, so following [9] we call this a digital shadow, not a twin.

B. Support Bands

For each student the dashboard computes attendance over a 14-day window as $pct = 100 \cdot P/C$, where P is the number of days with a check-in and C is the number of class days, defined as days on which any check-in occurred on campus. Dividing by 14 calendar days instead would score a perfect weekday attender at $10/14 \approx 71\%$ and wrongly flag them. Students with no records in the window are shown as "No data" rather than 0 %, so newly added students are not flagged on their first day. The band logic is given in Fig. 3. Labels are deliberately support-framed: Needs attention, Watching and On track. Mentors can filter by programme, year and section and export all students below 75 % as a CSV file.

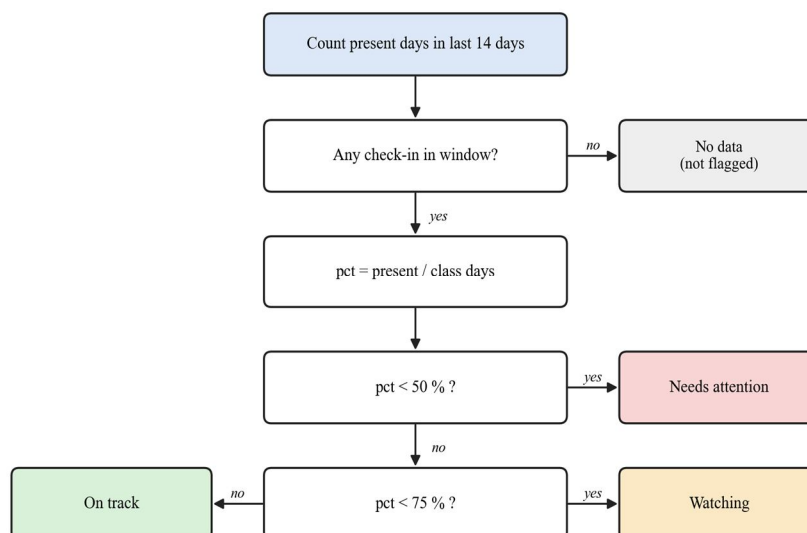


Fig. 3 Rule used to assign support bands.

VI. IMPLEMENTATION AND EVALUATION

A. Implementation

The front end uses React 19, Vite and TypeScript with Tailwind CSS, Recharts, qrcode.react and html5-qrcode, and is hosted on Vercel. The back end is Supabase on its free tier. The code base contains about 6,500 lines of TypeScript and 1,340 lines of SQL. Beyond check-in, the map and the mentor dashboard, the system provides building floor pages, a student page showing attendance, timetable and marks, a timetable editor, marks entry and a CSV roster import that matches column headings automatically and requires recorded consent for every student. Timetable slots and one set of unit test marks are generated as clearly labelled placeholders until real data is entered; all other simulated data was removed.

B. Functional and Adversarial Testing

We tested the deployed system from a smartphone and a laptop, with emphasis on attempts to defeat the check-in. Table I lists the adversarial cases and the observed behaviour.

TABLE I
ADVERSARIAL CHECK-IN TESTS

Attempt	Outcome
Reuse QR image older than 90 s	Rejected (expired)
Unknown roll number	Rejected
Wildcard input, e.g. NKS%	Rejected (no match)
Bound phone, other student's roll	Rejected
Second phone, already bound roll	Rejected
Repeat scan in same period	No duplicate row
Open dashboard without login	Redirected to login

C. Comparison

Table II compares NAKSHA with common approaches. Times are rough estimates for a class of about sixty students.

TABLE II
COMPARISON WITH EXISTING METHODS

Property	Roll call	Biometric	Fixed QR	NAKSHA
Hardware	None	Per room	None	None
Class time	5–10 min	Queue	1–2 min	1–2 min
Proxy risk	High	Low	High	Low
Live room view	No	Rare	Rare	Yes
Early warning	No	Reports	Reports	Yes
Biometric data	No	Yes	No	No

VII. PRIVACY AND ETHICS

- Row-level security on every table; personal data is readable only by accounts in a staff list, and public sign-up is disabled.
- The anonymous role can call only the check-in function and can read no table directly.
- The map shows room totals only, never individual locations.
- No protected attributes are collected, so none can become model features.
- Students are imported only with a stored consent time and source, as required by [11].
- A human mentor always decides what action, if any, follows a band.

VIII. LIMITATIONS AND FUTURE WORK

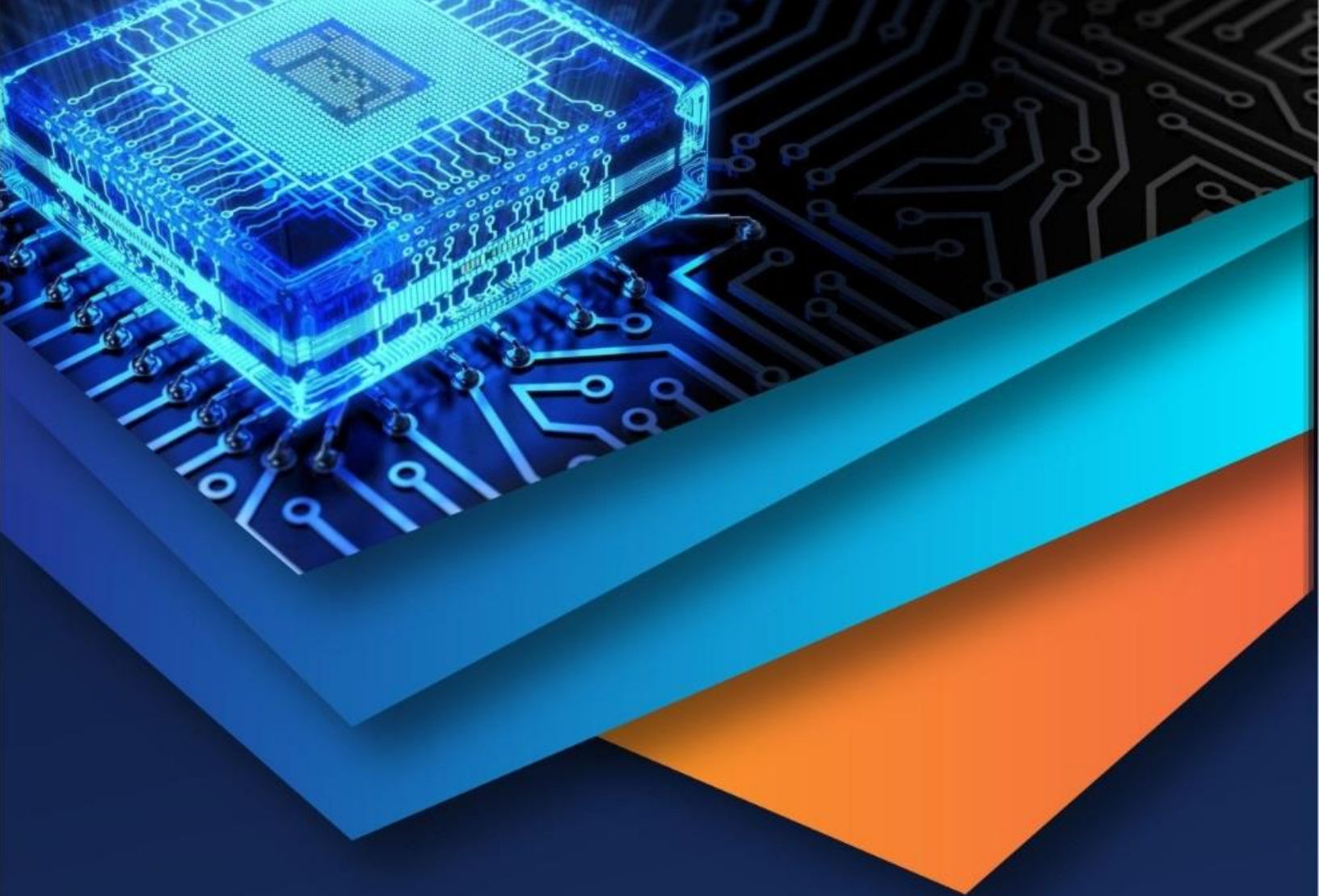
Phone binding stops one handset marking many students but cannot stop a student from lending their own phone, so a teacher's head count remains a useful final check. Clearing browser storage removes the device identifier and requires a mentor to reset the binding. Buildings are drawn as simple footprints, and free-tier limits would need review for a whole university. Most importantly, the current bands are a transparent rule rather than a learned model, and the system has so far been tried with a small group rather than a full term. The next stage is an XGBoost model trained first on the public dataset of [2], served by a FastAPI service with a scheduled batch endpoint that writes scores and top SHAP reasons for the dashboard, and an on-demand endpoint that explains a single student. A fairness audit comparing error rates across groups will precede any use. Further work includes mentor-to-class assignment, late-arrival flags, guardian consent records for minors and a semester-long multi-class trial.

IX. CONCLUSION

NAKSHA shows that a college can obtain fast, tamper-resistant and live attendance using only the phones students already own and the screens already in classrooms, at no recurring cost. The key design choice was to trust nothing that arrives from the phone and to let a single database record drive every view. On that foundation the system offers a campus digital shadow and a support-oriented early warning dashboard, and it is ready for an explainable predictive model.

REFERENCES

- [1] L. P. Macfadyen and S. Dawson, "Mining LMS data to develop an 'early warning system' for educators: A proof of concept," *Comput. Educ.*, vol. 54, no. 2, pp. 588–599, 2010.
- [2] P. Cortez and A. Silva, "Using data mining to predict secondary school student performance," in *Proc. FUBUTEC*, Porto, Portugal, 2008, pp. 5–12.
- [3] H. Yao, D. Lian, Y. Cao, Y. Wu, and T. Zhou, "Predicting academic performance for college students: A campus behavior perspective," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 3, 2019.
- [4] V. Das Swain et al., "Leveraging WiFi network logs to infer student collocation and its relationship with academic performance," *EPJ Data Sci.*, vol. 12, 2023.
- [5] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD*, 2016, pp. 785–794.
- [6] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. NeurIPS*, 2017, pp. 4765–4774.
- [7] J. Brazauskas, A. Danish, V. Safronov, R. Verma, R. Mortier, and A. Lewis, "CDBB West Cambridge digital twin: Lessons learned," *arXiv:2209.15290*, 2022.
- [8] C. Bäcklund, J. Lundqvist, and M. Molinari, "Showcasing a digital twin for higher educational buildings," *Front. Built Environ.*, vol. 10, 2024.
- [9] W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn, "Digital twin in manufacturing: A categorical literature review and classification," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1016–1022, 2018.
- [10] T. Feathers, "Major universities are using race as a 'high impact predictor' of student success," *The Markup*, Mar. 2021.
- [11] Government of India, *The Digital Personal Data Protection Act*, 2023, No. 22 of 2023.



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)