Edge-Based Indoor Bluetooth Low Energy Fingerprinting for Low-Cost, Offline Room-Level Localization in College Campus Buildings

[1]Om Hage, [2]Prabhat Madke, [3]Nishith Gajjalwar, [4]Om Ebitwar, [5]Om Kadu, [6]Sumit Fulzele
[1-6]Department of Artificial Intelligence
[1-6]G H Raisoni College of Engineering & Management Nagpur, India

***Abstract*—A common problem faced by first-year students, visitors and service staff is finding specific rooms in large campus buildings. Satellite navigation does not work indoors, and most commercially available indoor localization solutions require expensive infrastructure and a constant internet connection. This paper presents CampusNav-32, a low-cost, fully offline solution for room-level localization based on Bluetooth Low Energy (BLE) received signal strength indicator (RSSI) fingerprinting. Eight low-cost BLE beacons were placed along a representative corridor on campus, and a fingerprint database of 31 reference points was built over four survey sessions. A k-Nearest Neighbour (k-NN) classifier, benchmarked against a Random Forest, runs directly on a smartphone or an ESP32 microcontroller, so no server or internet connection is required. In on-site testing with 150 independent scans across five zones, the system identified the correct room 130 times, an overall accuracy of 86.7% (95% Wilson interval: 80.3–91.2%), ranging from 80% to 93% per zone. Accuracy was lowest in the open corridor and around the office, where the coverage of several beacons overlaps; a simple analysis of the fingerprint database shows that these are also the zones whose signal signatures lie closest together. The hardware for the whole block cost approximately ₹3,250. The system design, survey methodology, pre-processing, classification and confidence output are described, and failure modes, limitations and privacy considerations are analysed so that other institutions can replicate and extend the system.**

***Index Terms*—indoor positioning, Bluetooth Low Energy, RSSI fingerprinting, k-nearest neighbour, edge computing, ESP32, campus navigation, wayfinding**

I. Introduction

Large educational campuses can be difficult to navigate. Similar-looking corridors, inconsistent room numbers on notices and the lack of a mental map among visitors such as first-year students, parents, guest speakers, external examiners and delivery personnel regularly cause confusion. During project planning, students and staff shared many anecdotes about arriving late to seminars and interviews for this reason. Although no formal survey was conducted, these accounts were considered a realistic and tangible motivation for developing an indoor navigation system.

Although outdoor navigation is effectively solved by satellite systems, GPS signals are strongly attenuated by roofing materials such as reinforced concrete and steel, which makes positioning inaccurate indoors. Several alternatives have been considered, including Wi-Fi RSSI, Ultra-Wideband (UWB), visible-light positioning, inertial dead reckoning and Bluetooth Low Energy (BLE). BLE has become popular in resource-constrained applications because beacons are inexpensive, consume very little power (a coin-cell battery can last for more than a year) and are supported natively by almost all modern smartphones, so no additional user hardware is required.

A campus is a good use case: while many new people enter its buildings, the students themselves are a unique resource, as they can survey, deploy and maintain the beacons at no ongoing cost. However, the radio environment of Indian college buildings poses challenges. Corridors are narrow, and walls are

typically brick and reinforced concrete, which attenuate 2.4 GHz signals more than the plasterboard common in newer buildings. Rooms contain metal furniture and shutters, and personal devices cause considerable interference. RSSI values are therefore noisy, and naive trilateration based on path-loss models often gives unreliable room identification.

This paper presents CampusNav-32, an edge-based system for finding rooms within a building. Rather than converting RSSI into distance, the system records a fingerprint—the vector of RSSI values from all beacons
—at known locations during an offline survey. At runtime, a newly observed RSSI vector is classified against this database. The classifier is simple enough to run on the user's phone or on an ESP32 microcontroller, so the system can predict the user's room with no ongoing cloud cost and without location data ever leaving the device. The contributions of this work are as follows.

- **A complete low-cost design.** The beacons, their deployment, the survey method, pre-processing and the k-NN classifier used for deployment are described. The complete design costs approximately
₹3,250.
- **On-site evaluation.** A total of 150 scans were collected using a formal protocol, divided among five zones, recorded in randomised order and scored blind. The overall room-level accuracy is 86.7%, reported with confidence intervals.
- **Failure analysis.** The zones with the lowest accuracy were associated with overlapping beacon coverage and small distances between room fingerprints. A walkthrough demonstrates the system in use and highlights its known weakness.
- **Transparent limitations and ethics.** Device variation, battery drift, single-floor testing, sample size and privacy by design are discussed, together with an extension roadmap.

The remainder of this paper is organised as follows. Section 2 reviews related work. Section 3 describes the system design and test environment. Section 4 details the methodology and Section 5 the experimental protocol. Section 6 presents the results, Section 7 discusses the findings, and Section 8 covers limitations and threats to validity. Section 9 addresses privacy and ethics, and Section 10 concludes the paper.

II. RELATED WORK

A. Indoor Positioning Technologies

Several studies have categorised smartphone-based indoor positioning technologies by the type of signal and algorithm used. Proximity systems report the closest beacon but are very unspecific. Trilateration uses the log-distance path-loss model to convert RSSI values into distances and then computes the intersection of circles drawn around known anchors; however, the path-loss exponent must be calibrated for each environment, and the method is highly sensitive to multipath propagation and body blocking. Fingerprinting builds a database of real-world measurements and compares live measurements against it. This avoids radio-propagation modelling but requires a site survey. UWB can provide accuracy below one metre but requires specialised and expensive anchors and tags. Wi-Fi fingerprinting can exploit existing access points but is sensitive to changes in the access-point layout.

B. Bluetooth Low Energy Fingerprinting and Classifiers

BLE fingerprinting with smartphones has been shown to be feasible using consumer-grade devices [8]. Bai et al. demonstrated that a BLE-only system can be truly affordable and well suited to resource-limited student environments [7]. By considering neighbouring reference points, Li et al. reduced the average positioning error to approximately one metre using an eight-neighbourhood template-matching technique [1]. Zuo et al. showed that modelling the relationship between beacons and the target as a graph-based optimisation problem improves on simple fingerprinting when only a limited number of beacons is

available [3]. Safwat et al. reported a real-world deployment of a very affordable system that uses k-NN to position users reliably within a room using only a small number of beacons per zone [2].

Several classifiers have been investigated. Chatzimichail et al. compared k-NN, decision trees (DT), random forest (RF) and support vector machines (SVM) and found that the spatial separation of the rooms mattered more than the choice of classifier [11]. Convolutional neural networks (CNN) trained on RSSI radio maps have been shown to outperform classical k-NN, but they require a larger training dataset [5]. Wu et al. quantified the accuracy and complexity of deep-learning models relative to traditional RSS-based methods [6]. A recent systematic review of Wi-Fi and BLE systems shows that RSSI-based k-NN fingerprinting remains dominant because of its affordability, although deep learning is becoming more popular when higher accuracy is required [4]. Sadowski and Spachos compared trilateration, fingerprinting and proximity methods on Internet of Things (IoT)-class hardware of similar cost to the ESP32 used here [9]. Table 1 summarises these studies.

#	Reference	Approach	Main finding relevant to this work
1	Li et al. (2019) [1]	Eight-neighbourhood template matching	Average error of about 1 m; approx. 20% better than plain or weighted k-NN
2	Safwat et al. (2023) [2]	k-NN / weighted k-NN on a real BLE deployment	Room-level localisation with few beacons per zone
3	Zuo et al. (2018) [3]	Graph optimisation	Better than plain fingerprinting where beacons are sparse
4	Chatzimichail et al. (2020) [11]	Comparison of k-NN, DT, RF and SVM	Room separability matters more than classifier choice
5	Martín-Frechina et al. (2025) [4]	Systematic review	k-NN remains the dominant low-cost approach; deep learning is rising
6	Sun et al. (2020) [5]	CNN on RSSI radio maps	Higher accuracy than k-NN given more training data
7	Wu et al. (2024) [6]	Deep learning vs. classical RSS methods	Quantified accuracy/complexity trade-off
8	Bai et al. (2020) [7]	Low-cost BLE-only system	Suited to student-built, resource-constrained deployments
9	Zhuang et al. (2016) [8]	Smartphone + BLE fingerprinting	Practical on commodity phones
10	Sadowski & Spachos (2018) [9]	Trilateration vs. fingerprinting vs. proximity	Compared on IoT hardware comparable to the ESP32

Table 1. Summary of closely related studies.

C. Positioning of This Work

Three main themes emerge from the literature. (i) k-NN fingerprinting provides a strong, low-complexity baseline that is suitable for edge devices. (ii) System accuracy depends heavily on how well room signals are separated, so beacon deployment and the site survey must be planned with great care. (iii) Deep learning can increase accuracy, but its greater data and computational requirements make it unsuitable for an initial deployment. We found few open-source tools that combine an offline fingerprint database with a local classifier for Indian college buildings, which are characterised by narrow corridors, brick and concrete walls and high levels of interference from personal devices. CampusNav-32 aims to address this gap with a design that students can replicate.

III. SYSTEM DESIGN

A. Hardware and Software

The prototype uses four types of hardware, listed in Table 2. The BLE beacons belong to the Nordic nRF52 family and support the iBeacon and Eddystone protocols. They act as fixed anchor points, run on coin-cell batteries and transmit advertising packets at regular intervals, from which any receiving device can measure signal strength. The ESP32 development board serves as an optional standalone edge scanner and classifier; it is a dual-core microcontroller with Wi-Fi and BLE capability and 4 MB of flash memory. The main end-user device is an Android smartphone (version 8.0 or above) with BLE capability, used for both the site survey and live room lookup. A standard laptop is used for offline model training and database management.

Component	Specification	Purpose
BLE beacon	Nordic nRF52-based; iBeacon/Eddystone compatible; coin-cell powered	Fixed reference anchor transmitting an RSSI-detectable advertisement
ESP32 development board	Dual-core; integrated Wi-Fi and BLE; 4 MB flash	Optional standalone edge scanner/classifier for a kiosk
Smartphone	Android 8.0 or later; BLE support	End-user device for site survey and live room lookup
Laptop	Any machine able to run Python 3	Offline model training and database management

Table 2. Hardware components used in the prototype.

The software was implemented in Python 3 using scikit-learn, pandas and numpy to manage the database and to train and compare the k-NN and Random Forest models. RSSI values were collected using the Android BLE scanning API and the nRF Connect tool. The ESP32 firmware was written with the Arduino IDE or ESP-IDF. The fingerprint database was stored as a CSV file and migrated to SQLite as it grew.

B. Test Environment and Beacon Layout

The prototype was implemented in one representative block of the campus (Block A). The block comprises 11 rooms, 2 washrooms, 1 lift lobby, 1 stairwell and 1 corridor, giving a realistic mix of small classrooms, an open corridor and small utility rooms. Its layout is a central corridor with rooms on either side. The block is built from reinforced-concrete columns and brick partition walls, which cause much greater loss of 2.4 GHz BLE signals than plasterboard; closed wooden doors add a further 3–6 dB of loss.

Eight beacons (B1–B8) were installed at fixed points along the corridor walls at a height of about 2.5 m to limit obstruction by furniture and people. Spacing was kept to no more than about 12–15 m, based on the signal loss through the walls, so that usable signals from at least two anchors were available at every point. The identifier of each beacon (MAC address or iBeacon UUID/Major/Minor) was recorded in a beacon list according to its installed location (Fig. 1). The beacon locations and the rooms they primarily help to identify are listed in Table 3.

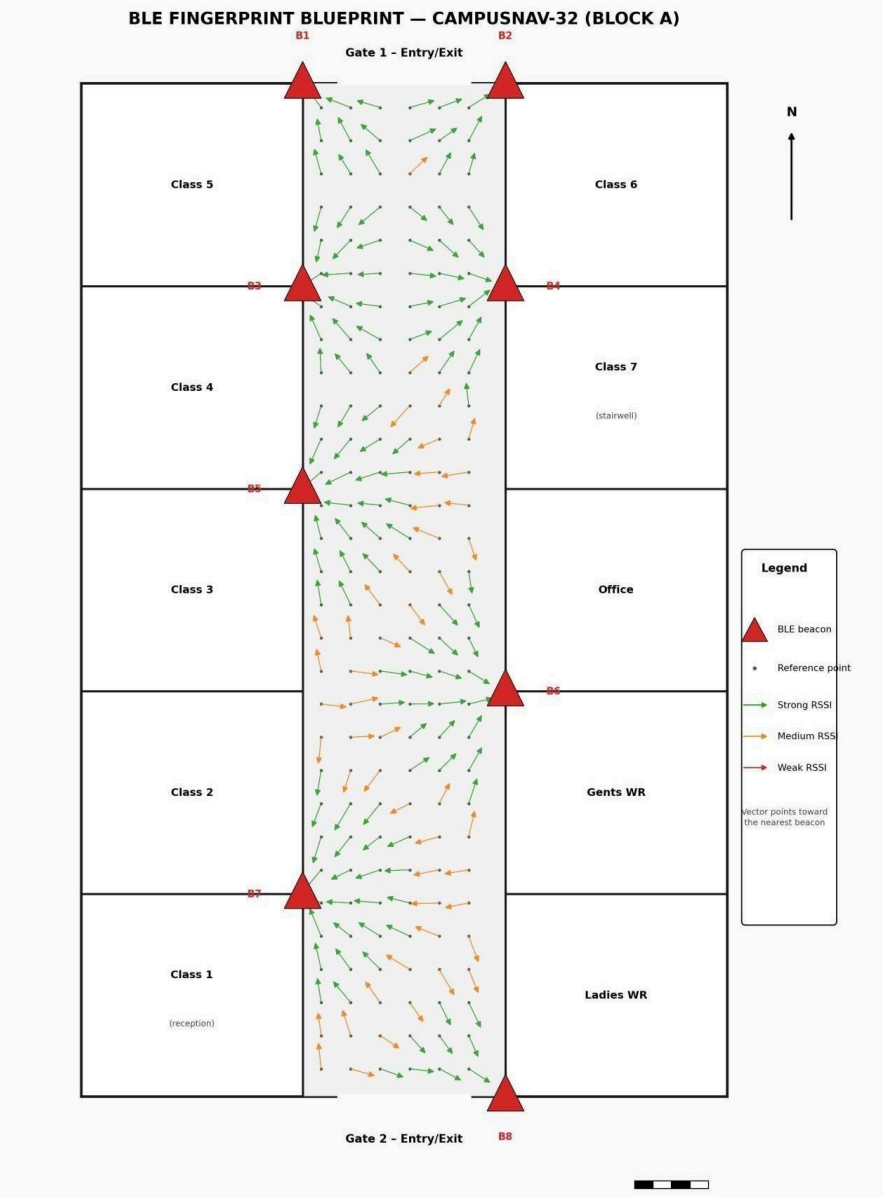

Fig. 1. Surveyed floor layout of Block A showing the central corridor, the eight beacons (B1–B8) and the reference-point grid. Arrows point toward the strongest beacon at each reference point (green: strong and reliable; orange: medium; red: weak, typically near doorways and boundaries).

Beacon	Mounting location	Primary rooms covered	Range
B1	Corridor wall, Gate 1 end (near Class 6/7)	Class 6, Class 7, Gate 1 approach	8–10 m
B2	Opposite corridor wall, Gate 1 end	Class 6, upper corridor	8–10 m
B3	Corridor wall, between Class 5/6 and Class 4/7	Class 4, Class 5, mid-upper corridor	8–10 m
B4	Opposite wall, near Class 7 / Office boundary	Class 7, Office, mid corridor	8–10 m
B5	Corridor wall, near Class 3 / Office	Class 3, Office, Gents WR, central corridor	8–10 m
B6	Opposite wall, near Gents WR	Gents WR, Class 4, lower-mid corridor	8–10 m
B7	Corridor wall, near Class 1/2	Class 1, Class 2, lower corridor	8–10 m

B8	Corridor wall, Gate 2 end (near Ladies WR)	Ladies WR, Class 1, Gate 2 approach	8–10 m

Table 3. Beacon placement and primary coverage.

C. System Architecture

Fig. 2 shows the flow of data from the beacon signal to the room label presented to the user. Both phases of the system use identical scanning and pre-processing code. In the offline phase, the surveyor records RSSI values at each reference point and adds the labelled average vector to the fingerprint database. In the online phase, the user's phone (or an ESP32 with identical scanning logic) records a new vector, which the trained classifier compares against the database to predict the user's room. The result is displayed together with a confidence level.



Fig. 2. System architecture and workflow of CampusNav-32: beacon deployment and RSSI site survey (offline phase), followed by fingerprint matching and room output (online phase).

Data flow. Raw RSSI scans from each beacon are converted into a feature vector and matched against the offline fingerprint database, and the closest room label is returned. *Edge-first design.* All matching and prediction take place locally on the phone or ESP32; no internet connection is required at runtime.

IV. METHODOLOGY

A. Offline Survey

Once the beacons were placed, the survey team walked around the corridor and rooms with a basic Android app that records the RSSI of each beacon. The team visited a grid of reference points spaced approximately 3 m apart. At each point, the app recorded a 30–45 second reading of the RSSI from every visible beacon, and outliers were removed using a z-score filter. The cleaned readings were averaged, and the resulting vector was saved in the fingerprint database together with the label of the room in which the reference point is located. Several readings were taken at each reference point and averaged, which reduces the effect of noise such as people walking past. Because human bodies interfere with 2.4 GHz signals, the survey was carried out at different times of the day so that the database represents a variety of occupancy conditions. In total, 31 reference points were used across the five zones.

B. Data Pre-processing

Raw RSSI data are noisy: even with a static phone, successive readings can vary by 5–10 dB. Three steps are applied during both the survey and runtime. First, the readings from each beacon are smoothed with a simple moving average, which suppresses drop-outs. Second, any missed beacon is filled with a fixed floor RSSI value so that every fingerprint has the same length. Third, no conversion to distance is performed: no log-distance path-loss model is applied, and the classifier is trained directly on raw RSSI values. The literature indicates that this gives results similar to distance-based approaches while avoiding

calibration of the path-loss exponent for each room.

C. Fingerprint Database

Table 4 shows an example of the data collected, with one averaged RSSI vector (in dBm) per room; values closer to zero indicate a stronger signal. Each room can be identified from its signature. Class 1 has strong signals from B7 (−62 dBm) and B8 (−70 dBm), and Class 6 from B2 (−63 dBm). Class 3 has a strong signal from B3 (−64 dBm) and a second strongest from B5 (−68 dBm). The Office has a strong signal from B5 (−66 dBm) and a second strongest from B4 (−70 dBm). Even where beacon coverage overlaps between rooms, the rooms can be distinguished by examining the full vector. Fig. 3 shows the same data as a heat map.

Room label	B1	B2	B3	B4	B5	B6	B7	B8
Class 1	−88	−91	−93	−95	−90	−92	−62	−70
Class 2	−85	−89	−90	−93	−88	−70	−65	−78
Class 3	−82	−86	−64	−79	−68	−84	−87	−90
Class 4	−79	−83	−68	−72	−85	−88	−89	−92
Class 5	−66	−74	−80	−84	−89	−91	−92	−94
Class 6	−68	−63	−81	−85	−90	−92	−93	−95
Office	−90	−92	−83	−70	−66	−78	−88	−90
Corridor (mid)	−84	−86	−75	−77	−71	−73	−86	−88

Table 4. Sample fingerprint database records (RSSI in dBm).

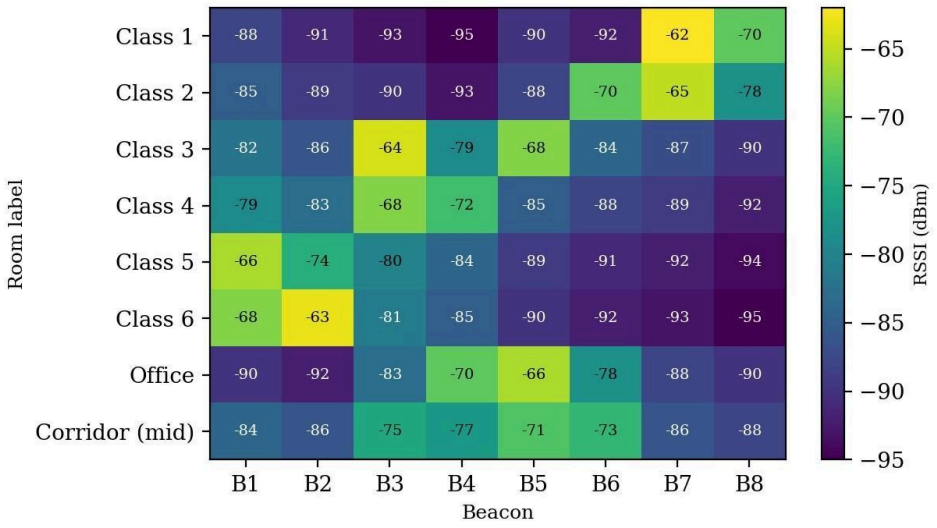

Fig. 3. Heat map of the sample fingerprints in Table 4. Each row is the RSSI signature of one room across the eight beacons.

To estimate how well the rooms can be separated, the Euclidean distance between each room's fingerprint and that of its nearest other room in signal space was computed (Table 5).

Room	Nearest other room	Distance (dB)
Class 1	Class 2	24.2
Class 2	Class 1	24.2
Class 3	Corridor (mid)	16.2

Class 4	Class 3	19.9
Class 5	Class 6	11.4
Class 6	Class 5	11.4
Office	Corridor (mid)	15.6
Corridor (mid)	Office	15.6

Table 5. Nearest other room in signal space (Euclidean distance in dB).

D. Classification

k- Nearest Neighbour computes the Euclidean distance between the current RSSI vector q and each stored fingerprint f_i:

$$d(q, f_i) = \sqrt{\sum_{j=1}^{m} (q_j - f_{i,j})^2}$$

The predicted room $\hat{y}$ is determined by a majority vote among the k nearest fingerprints:

$$\hat{y} = \arg\max_{l \in L} \sum_{i \in N_k(q)} \delta(y_i, l) \quad (2)$$

where $N\square(q)$ is the set of the k fingerprints closest to q, y_i is the room label of fingerprint f_i, L is the set of room labels, m is the number of beacons, and δ returns 1 if the neighbour's label equals l and 0 otherwise. We use k = 3 to 5, a value found empirically to balance response time against sensitivity to noise. A confidence score is computed from the fraction of neighbours that agree with the predicted room:

$$c(q) = \frac{1}{k} \sum_{i \in N_k(q)} \delta(y_i, \hat{y}) \quad (3)$$

This score is reported to the user as High, Medium or Low confidence. The complete procedure is given in Algorithm 1.

Algorithm 1. Runtime room classification

Input: RSSI scan from the visible beacons; fingerprint database $D = \{(f_i, y_i)\}$; number of neighbours k; floor RSSI value r_0.

Output: predicted room $\hat{y}$ and confidence level.

1. Smooth the RSSI of each beacon with a moving average.
2. Replace the RSSI of any missed beacon with r_0 to form the query vector q.
3. For every fingerprint f_i in D, compute $d(q, f_i)$ using Eq. (1).
4. Select the k fingerprints with the smallest distances to form $N\square(q)$.
5. Assign $\hat{y}$ by majority vote using Eq. (2).
6. Compute $c(q)$ using Eq. (3); report High if $c \geq \tau\square$, Medium if $\tau\square \leq c < \tau\square$, otherwise Low.
7. Return $\hat{y}$ and the confidence level.

During training, a Random Forest is also fitted as a secondary model, since it tends to be more robust to noise and missing beacons. It adds a small computational cost, which remains well within the capability of a modern smartphone. Table 6 compares the candidate classifiers against edge-deployment considerations. k-NN was chosen as the main algorithm for the prototype because it requires no training beyond storing the sampled fingerprints, scales well as references are added, and is simple to implement

directly in C++ on an ESP32. A small neural network could be deployed on the ESP32 using quantisation, but this is better left until more data are available.

Algorithm	Training time	Inference	Memory	Suitability for edge
k-NN	Negligible (lazy learner)	Fast for small datasets	Low: stores raw fingerprints	Excellent; simplest on ESP32 or phone
Random Forest	Seconds to minutes	Fast	Moderate: stores trees	Good; more setup but robust to noise
SVM	Minutes	Fast	Low to moderate	Fair; needs kernel/parameter tuning
Small NN / MLP	Minutes to hours	Fast once trained	Moderate to high	Possible with quantisation; later phase

Table 6. Comparison of candidate classifiers for edge deployment.

E. User Interface

The prototype interface is a lightweight mobile-phone page that can be opened in the browser of any BLE-capable smartphone. It consists of a large text box showing the predicted room name, a smaller box showing the confidence level (High, Medium or Low, based on the consensus among the nearest neighbours) and a debug panel listing all visible beacons with their raw RSSI values. On first launch only the basic interface is shown, so that users are not overwhelmed by technical information. The confidence panel was included as an element of honesty in the system: in zones where the neighbours disagree, the interface reports lower confidence in its prediction. Fig. 4 shows screenshots of the application.

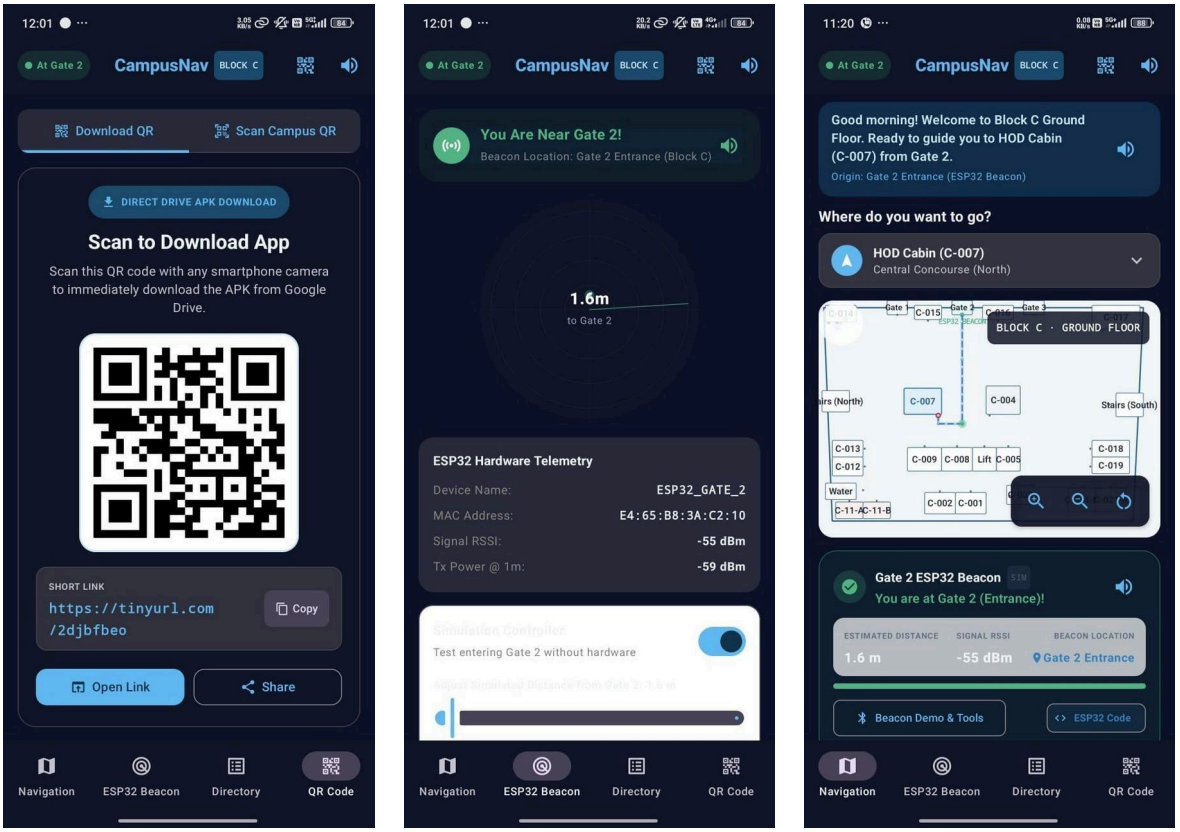

Fig. 4. Screenshots of the CampusNav mobile interface: (left) app download screen with QR code, (centre) ESP32 beacon proximity and telemetry view, and (right) navigation view with destination selection and floor map.

V. EXPERIMENTAL PROTOCOL

A. Data Collection

Survey data were collected over four sessions during the 2026–27 academic session (Table 7). The sessions covered periods of low traffic, moderate traffic and peak traffic between lectures, so that a variety of occupancy levels was captured in the database. The final session was a focused resurvey of the areas found to have low accuracy in the first three sessions.

Session	Date	Time of day	Rooms covered
Session 1	Week 2, June	Morning (low traffic)	Class 1–Class 5
Session 2	Week 3, June	Afternoon (moderate traffic)	Class 6–Class 11, Office
Session 3	Week 1, July	Between-lecture peak traffic	Full corridor, both washrooms
Session 4 (resurvey)	Week 3, August	Morning (low traffic)	Zones flagged for low accuracy in Sessions 1–3

Table 7. Data collection schedule.

B. Evaluation

Accuracy was assessed by returning to the surveyed reference points and collecting 30 independent scans for each zone at different times of day. In total, 150 scans were collected across five zones (Class 1, Class 2, Class 3, Office and the corridor section near B5/B6). A scan was scored as correct if the classifier's top prediction matched the room in which the user was actually standing. The phone was held at a natural waist-to-chest height and users remained static during scanning, which is representative of how someone would use the app while pausing in the building.

The order of testing was randomised across the four sessions to reduce bias, and users noted their actual location before seeing the classifier's prediction. Overall accuracy and per-zone accuracy are reported. A 95% Wilson score interval is included to indicate the uncertainty arising from the relatively small number of scans per zone.

VI. RESULTS

A. Room-Level Accuracy

As shown in Table 8 and Fig. 5, the system identified the correct room in 130 of 150 scans, an overall accuracy of 86.7% (95% Wilson confidence interval: 80.3–91.2%). Class 3 had the highest accuracy (93.3%), followed by Class 1 (90.0%) and Class 2 (86.7%). The lowest accuracies were obtained for the Office (83.3%) and the corridor near B5/B6 (80.0%). Because each zone has only 30 scans, the confidence intervals are wide—for example, 62.7–90.5% for the corridor—and they overlap across zones, so differences between adjacent zones should be regarded as indicative only.

Room / zone	Reference points	Correct / test scans	Accuracy (95% CI)
Class 1	6	27 / 30	90.0% (74.4–96.5)
Class 2	6	26 / 30	86.7% (70.3–94.7)
Class 3	5	28 / 30	93.3% (78.7–98.2)

Office	4	25 / 30	83.3% (66.4–92.7)
Corridor (near B5/B6)	10	24 / 30	80.0% (62.7–90.5)
Overall	31	130 / 150	86.7% (80.3–91.2)

Table 8. Room-level accuracy from on-site testing.

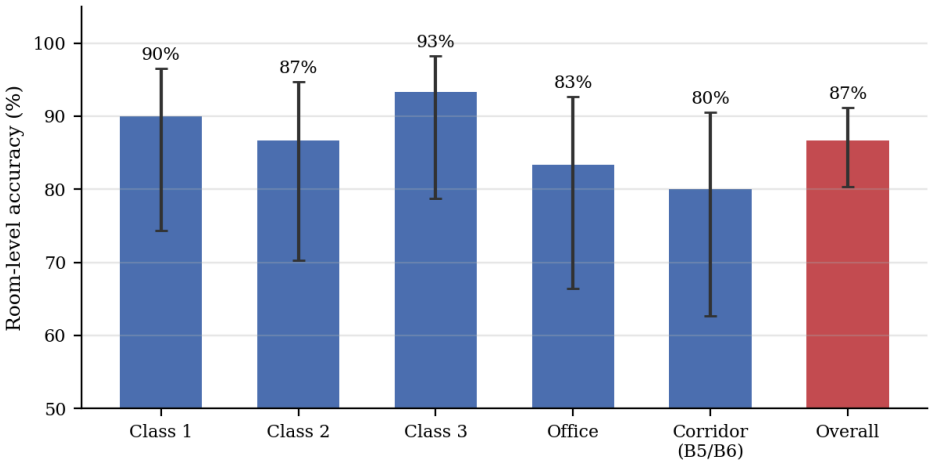

Fig. 5. Room-level accuracy by zone. Error bars show 95% Wilson confidence intervals.

B. Error Analysis

There were 20 errors in total, of which 11 (55%) occurred in two regions: the Office (5 errors) and the corridor around B5/B6 (6 errors). Both regions lie in areas where the coverage of two or more beacons overlaps, and B5 and B6 have similar medium signal strengths there. The nearest neighbours do not agree on whether such samples belong to the corridor or to the adjacent room. This is consistent with the signal-space analysis in Section 4-C (Table 5): the fingerprints of the Office and the middle of the corridor differ by only
15.6 dB. It also agrees with the finding of Chatzimichail et al. [11] that accuracy is determined by how well spaces can be separated in signal space rather than by the classifier used.

However, the relationship is not strictly linear. Class 3 is also close to the corridor fingerprint (16.2 dB) yet achieved the highest accuracy. This is explained by its distinctive signal from B3 (−64 dBm), which the corridor fingerprint does not have. Because each zone has multiple reference points, the signal-space distance can serve as a screening tool for identifying likely problem areas. It also highlights a risk: Class 5 and Class 6 have the smallest distance between fingerprints in the sampled database (11.4 dB), yet neither was among the five zones tested, so they should be prioritised in the next test cycle.

C. Continuous Walkthrough

Table 9 shows one continuous walkthrough from Class 1 to the Gents washroom, sampled approximately every 15 s. Seven of the eight samples matched (87.5%), consistent with the overall results, although the sample is very small. The single mismatch (at 75 s) occurred when the system briefly predicted ‘Office’ after the user had already entered Class 3, within the beacon-overlap area described above. The system corrected itself at the next sample as the user’s signal profile moved away from the boundary. Medium confidence was reported for the corridor and Office predictions and High confidence within rooms, so the interface signalled lower certainty in the uncertain area.

Time (s)	Predicted room	Actual room	Match	Confidence

0	Class 1	Class 1	Yes	High
15	Class 1	Class 1	Yes	High
30	Corridor	Corridor	Yes	Medium
45	Corridor	Corridor	Yes	Medium
60	Class 3	Class 3	Yes	High
75	Office	Class 3	No	Medium
90	Office	Office	Yes	Medium
105	Gents WR	Gents WR	Yes	High

Table 9. Sample walkthrough log (Class 1 to Gents washroom).

VII. DISCUSSION

A. Accuracy

The overall accuracy of 86.7% is in line with the low-cost k-NN systems described in the literature, such as the real-world deployment of Safwat et al. [2]. A direct comparison is not possible, however, because building structures, beacon densities, room definitions and test procedures differ. The result shows that a student-designed solution with a small number of beacons (eight) and simple classification can give useful accuracy in a challenging building. It also supports the design decision to prioritise good beacon placement and careful surveying: errors are concentrated in areas of overlapping beacon coverage, so targeted beacon densification in these areas is likely to yield greater improvement than a more complex classification algorithm.

B. Cost and Practicality

A main aim of the system was to be cheap enough for any department to replicate. The approximate costs of instrumenting the block are listed in Table 10; smartphone and laptop costs are omitted because these are assumed to be available. Overall, instrumenting the block with eight beacons, an edge scanner and spares costs approximately ₹3,250. Extending coverage to another block incurs only the cost of additional beacons; all other components are reused unchanged.

Item	Unit cost (approx.)	Quantity	Total (approx.)
BLE beacon	₹250	8	₹2,000
ESP32 development board (edge scanner)	₹450	1	₹450
Coin-cell batteries (spare)	₹50	10	₹500
Miscellaneous (mounting, cabling)	—	—	₹300
Total			₹3,250

Table 10. Approximate hardware cost for one block.

C. Comparison with Other Navigational Methods

Table 11 compares CampusNav-32 with alternative wayfinding methods that can be used on a campus. Static signs and QR tags are very cheap but give no information about the visitor's location. Wi-Fi fingerprinting can reach a similar room-level accuracy but needs a dense grid of access points and is susceptible to access-point changes. UWB offers much higher accuracy at significantly higher cost and with more specialised hardware. BLE fingerprinting is a practical compromise: it is low-cost to implement, works without internet connectivity and provides room-level accuracy, but it requires periodic re-surveying.

Approach	Upfront cost	Works offline	Room-level accuracy	Maintenance effort
Static paper maps / signage	Very low	Yes	None (manual reading)	Low, but quickly outdated
QR-code room labels	Low	Partial (needs app and scan)	Exact, but only at the label	Low
Wi-Fi RSSI fingerprinting	Medium–High (dense AP grid)	Yes	Room-level, similar to BLE	Medium; sensitive to AP changes
UWB anchors	High	Yes	Sub-metre, better than BLE	Medium–High; specialised hardware
CampusNav-32 (BLE fingerprinting)	Low	Yes	Room-level (~87% here)	Medium; periodic re-survey

Table 11. Comparison with existing wayfinding approaches.

VIII. LIMITATIONS AND THREATS TO VALIDITY

The following limitations should be considered when interpreting the results.

- **Single block, single floor.** The prototype was validated in one block and on one floor only. It cannot distinguish between floors, since no barometric pressure sensor or floor-specific beacons are used, and its performance cannot be assumed to carry over to buildings with different structures.
- **Sample size.** Each zone was tested with 30 scans, giving wide confidence intervals (Table 8). Only two of the five zones tested (the Office and the corridor) lie at the boundary where accuracy is lowest, and several multi-room zones (such as Class 4–6) were not tested for accuracy.
- **Drift over time.** The database represents the radio environment at the time of the survey. Because the beacons are battery-powered, their transmission power declines slightly over a period of weeks, and changes in furniture or Wi-Fi access points also alter the radio environment, so periodic re-surveying is required.
- **Different phones.** Different phone models report slightly different absolute RSSI values for the same beacon and location. This is not accounted for in the current design; adding a per-device offset is a planned improvement.
- **Crowding.** Testing took place under normal occupancy. The behaviour of the system in abnormally busy situations, such as admission days or the periods between lectures, has not been measured.
- **Ground-truth collection.** During ground-truth collection the testers remained still, so accuracy may be lower when walking quickly or when the phone is in a pocket or bag. Testers recorded their actual location before seeing the classifier's output, so they were blind to the prediction.

The risks identified during design and testing, together with their mitigations, are summarised in Table 12.

Risk	Likelihood	Impact	Mitigation
Beacon battery depletion lowers signal over time	Medium	Medium	Scheduled battery checks; log RSSI trend per beacon to flag drift early
Furniture or partition changes alter the RF environment	Medium	Medium	Re-survey affected zones after any layout change
Device heterogeneity gives inconsistent RSSI across	High	Low–Medium	Per-device calibration offset in a future release

phones			
High foot traffic degrades accuracy beyond tested conditions	Medium	Medium	Dedicated high-traffic test cycle before full rollout
Beacon theft or accidental removal	Low	Medium	Tamper-resistant enclosures mounted at height

Table 12. Risk assessment and mitigation.

IX. PRIVACY AND ETHICAL CONSIDERATIONS

The system was designed to protect user privacy in three ways. First, RSSI scanning and room prediction are performed entirely on the user's device; no location data are transmitted to, or stored on, any server controlled by the project team. Second, the fingerprint database contains only the signal characteristics of specific rooms with respect to the beacons; it is recorded only during intentional mapping sessions conducted by team members who have consented to take part. Third, any future analytics feature (for example, to determine which rooms are most difficult to locate) will present only aggregated data locally on the device. This approach follows the institute's code of ethical practice. In terms of inclusivity, the users who would benefit most from room prediction are those with visual or mobility impairments, for whom paper maps and static signage work poorly; an audio or haptic extension of the room prediction would therefore be valuable.

X. CONCLUSION AND FUTURE WORK

CampusNav-32 demonstrates that BLE RSSI fingerprinting combined with edge-based k-NN classification can provide affordable room-level indoor navigation in a college building without GPS, a constant internet connection or specialised location hardware. The system uses eight BLE beacons at a total hardware cost of approximately ₹3,250 and identified the correct room in 130 of 150 on-site scans (86.7%). Errors occurred mainly in the beacon-overlap zones around the Office and the corridor. A simple analysis of the signal space predicts where such errors are likely to occur, which provides a useful screening method for future deployments. The survey methodology, pre-processing and classification are documented so that other departments can replicate this work. Planned next steps are as follows.

- Expand the fingerprint database to all academic blocks and add floor-level detection using a barometric pressure sensor or floor-specific beacons.
- Increase the number of reference points in the overlapping areas, and evaluate a weighted k-NN that favours closer matches and more recent survey sessions to reduce sensitivity to beacon drift.
- Implement device-specific RSSI calibration, and run dedicated test sessions with more users, focusing on high-traffic areas and including more samples to compare Class 5 and Class 6.
- Once a larger dataset has been gathered over several seasons, evaluate a lightweight convolutional or feed-forward neural network to improve on the k-NN baseline [5], [6].
- Add text-to-speech and haptic feedback for users with visual impairments, and provide turn-by-turn navigation between rooms.
- Automate regular re-surveys using a small robot or scheduled volunteer walks, and create a toolkit that other departments and affiliated colleges can use to place beacons and survey their buildings.

XI. APPENDIX A. REPLICATION CHECKLIST

The following process can be used to replicate the system in another department.

1. **Survey the block.** Identify the rooms that need to be recognised and draw a simple corridor plan. Note wall materials and doors, as these affect beacon placement (Section 3-B).
2. **Place beacons.** Mount beacons at a height of about 2.5 m and no more than 12–15 m apart, so that

every point is covered by at least two anchors. Record the ID and location of each beacon.

3. **Create reference points.** Lay out a grid of reference points about 3 m apart across the corridor and rooms, and label each with the room it lies in.
4. **Gather fingerprints.** At each point, collect RSSI readings from the beacons in view for 30–45 s, remove outliers with a z-score filter, and save the averaged vector with its label. Repeat at different times of day (Section 4-A).
5. **Pre-process.** Ensure that the moving average and floor value used for testing are the same as those used during surveying (Section 4-B).
6. **Train.** Save the database as a CSV file and compare k-NN with k values between 3 and 5 against a Random Forest. Keep k-NN as the on-device classifier (Section 4-D).
7. **Test.** Return to the reference points and complete at least 30 real-time scans per zone in randomised order, scoring each against where you actually were (Section 5-B).
8. **Maintain.** Check beacon battery levels regularly, and re-survey a zone after any change to furniture or layout (Section 8).

XII. ACKNOWLEDGMENT

We would like to thank our project guide Dr. Tabassum Khan, the Project In-charge and Head of Department of Artificial Intelligence, and our co-guide Mr. Binaya Kumar Suna, as well as G H Raisoni College of Engineering & Management, Nagpur, for their guidance and for providing access to the campus block. This work was carried out under the Community Engagement Project (CEP) during the 2026–27 academic session.

References

[1] M. Li, L. Zhao, D. Tan and X. Tong, “BLE fingerprint indoor localization algorithm based on eight-neighborhood template matching,” *Sensors*, vol. 19, no. 22, art. 4859, 2019.

[2] R. Safwat, E. Shaaban, S. M. Al-Tabbakh and K. Emara, “Fingerprint-based indoor positioning system using BLE: real deployment study,” *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 1, pp. 240–249, 2023.

[3] Z. Zuo, L. Liu, L. Zhang and Y. Fang, “Indoor positioning based on Bluetooth low-energy beacons adopting graph optimization,” *Sensors*, vol. 18, no. 11, art. 3736, 2018.

[4] S. Martín-Frechina, E. Dura, I. Miralles and J. Torres-Sospedra, “From fingerprinting to advanced machine learning: a systematic review of Wi-Fi and BLE-based indoor positioning systems,” *Sensors*, vol. 25, no. 22, art. 6946, 2025.

[5] D. Sun, E. Wei, L. Yang and S. Xu, “Improving fingerprint indoor localization using convolutional neural networks,” *IEEE Access*, vol. 8, pp. 193396–193411, 2020.

[6] C. Wu, I.-C. Wong, Y. Wang, W. Ke and X. Yang, “Experimental study of Bluetooth indoor positioning using RSS and deep learning algorithms,” *Mathematics*, vol. 12, art. 1386, 2024.

[7] L. Bai, F. Ciravegna, R. Bond and M. Mulvenna, “A low cost indoor positioning system using Bluetooth low energy,” *IEEE Access*, vol. 8, pp. 136858–136871, 2020.

[8] Y. Zhuang, J. Yang, Y. Li, L. Qi and N. El-Sheimy, “Smartphone-based indoor localization with Bluetooth low energy beacons,” *Sensors*, vol. 16, no. 5, art. 596, 2016.

[9] S. Sadowski and P. Spachos, “RSSI-based indoor localization with the Internet of Things,” *IEEE Access*, vol. 6,
pp. 30149–30161, 2018.