IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511127

# Interactive Humanoid Robot for College Information, Task Reminder and Navigation Using QR Code and ESP32-CAM

**Asst. Prof. Abhijith N[1], Aishwarya M[2], Amulya Y T[3], Khushi B M[4], Kruthika S Hegde[5]**

Assistant Professor, Department of Electronics and Communication, JNN College of Engineering, Shivamogga, Karnataka, India[1]

B.E Student, Department of Electronics and Communication, JNN College of Engineering, Shivamogga, Karnataka, India[2-5]

**Abstract**: Locating departments, laboratories and offices in a large campus is difficult for new students and visitors, and GPS is unreliable indoors. This paper presents the design of an interactive humanoid robot that provides college information, task reminders and indoor navigation using QR code-based localization. QR codes carrying unique location identifiers are placed at key campus locations such as the entrance, library and laboratories. An ESP32-CAM module captures these markers and passes the decoded identifier to a laptop, which holds the campus map as a graph, identifies the robot's current location and computes the shortest route to the selected destination using Dijkstra's algorithm. The route is converted into forward, turn and stop commands and sent over serial communication to an Arduino UNO, which drives the DC motors through a motor driver. A web-based management application maintains student records, assigned tasks and college events, which supply the information and reminder content. The paper describes the system architecture, hardware and software design, graph-based path planning and the evaluation procedure. The web application and the basic robot movement (forward motion, turning and stopping along a predefined path) have been implemented and demonstrated, while full QR-based localization and path-planning integration are under development. The system offers a low-cost, modular and scalable platform for smart-campus assistance.

**Keywords:** Interactive Humanoid Robot, QR Code Localization, Indoor Navigation, ESP32-CAM, Arduino UNO, Path Planning, Task Reminder, Smart Campus

## I. INTRODUCTION

Smart campuses use automation and embedded intelligence to give students, staff and visitors easier access to services. In a large college, finding departments, laboratories, the library and administrative offices is a common difficulty, particularly for new students, parents and visitors. Printed maps, static signboards and information desks give only limited and non-interactive help, and human assistance is not always available during peak hours, holidays or after working hours [1].

Interactive service robots are already used in airports, hospitals and malls to guide people and answer questions. For a college, such a robot must know where it is. GPS works poorly inside buildings because of signal attenuation and multipath, so indoor robots rely on LiDAR, SLAM, Wi-Fi or vision-based landmarks. LiDAR and SLAM give high accuracy but need costly sensors and heavy computation, which puts them out of reach of low-cost educational projects [1], [7]. Artificial visual landmarks such as QR codes are a simple, inexpensive alternative: each code carries a location identifier or coordinates that a camera can read, giving the robot a reference point whenever a marker is seen [2], [3].

This paper presents an interactive humanoid robot for college information, task reminders and navigation using QR codes and the ESP32-CAM module. The main contributions are: (i) a modular three-unit architecture in which an ESP32CAM detects QR markers, a laptop performs decoding, campus mapping and path planning, and an Arduino UNO controls the motors; (ii) a graph-based campus model with Dijkstra shortest-path routing and a command generator for the motor controller; (iii) a web-based management application that supplies college events, student information and task reminders; and (iv) an evaluation methodology for localization, navigation and interaction, together with the results obtained so far from the working prototype.

 This work is licensed under a Creative Commons Attribution 4.0 International License 

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511127

The rest of the paper is organised as follows. Section II reviews related work. Section III presents the system model and problem formulation. Section IV describes the proposed framework. Section V explains the experimental setup, Section VI discusses results, and Sections VII and VIII give the conclusion and future work.

## II. RELATED WORK

Indoor localization for mobile robots has been studied with LiDAR, inertial sensing, visual SLAM and artificial markers. Yang et al. [1] combine LiDAR, an IMU and QR code landmarks, using an Extended Kalman Filter (EKF) and factor-graph optimization to reduce drift; the cost of LiDAR and the computation needed for fusion make it hard to use in low-cost educational robots. Bach et al. [2] place QR codes carrying global coordinates in the environment and fuse the camera-based position with wheel odometry through an EKF, which corrects accumulated dead-reckoning error at low cost but depends on marker visibility, lighting and camera quality.

Li and Xiong [3] estimate the pose of a camera from the four feature points of a QR code using the Perspective-4-Point (P4P) algorithm followed by iterative optimization. The pose is precise, but it needs careful camera calibration and additional computation. Alqobali et al. [4] survey geometric and semantic indoor navigation and highlight sensor limitations, localization error and computational cost, but propose no specific system. Kim et al. [5] use QR codes to reduce drift in SLAM-based localization, which again relies on correct marker placement. Sampathkrishna [6] uses ArUco markers and a graph of nodes and edges for low-cost localization and navigation in laboratories, although a dense marker network is hard to maintain in large areas. Liu [7] presents a visual SLAM system with dynamic feature tracking for moving obstacles, at the price of high computational power and sophisticated sensors. Table I summarises these works.

TABLE I SUMMARY OF RELATED WORK

| Ref. | Technique | Sensors / hardware | Main limitation |
|---|---|---|---|
| [1] | LiDAR-inertial SLAM with QR landmarks; EKF and factor graph | LiDAR, IMU, camera | High cost and computation |
| [2] | QR code coordinates fused with wheel odometry (EKF) | Camera, wheel encoders | Depends on marker visibility, lighting |
| [3] | P4P pose estimation from QR feature points with iterative optimization | Calibrated camera | Calibration and computation load |
| [4] | Survey of geometric and semantic indoor navigation | LiDAR, depth, vision | No implementation or results |
| [5] | QR codes to reduce drift in SLAM localization | Camera, SLAM sensors | Marker placement and upkeep |
| [6] | ArUco markers with graph-based mapping | Camera | Limited scalability in large areas |
| [7] | Visual SLAM with dynamic feature tracking | Camera, advanced sensors | Computational power and cost |

**Research gap.** Most of the reviewed work concentrates on localization accuracy and relies on costly sensors, complex fusion algorithms or heavy computation. QR code-based localization is simple and economical, but few low-cost designs combine it with campus path planning, college information, task reminders and human–robot interaction in a single platform. Motivated by this, the present work integrates an ESP32-CAM QR detector, laptop-based graph path planning, Arduino motor control and a web-based information and task management module in one modular, low-cost system.

## III. SYSTEM MODEL AND PROBLEM FORMULATION

### *A) System Model*

The proposed system has three processing units, shown in Fig. 1. The perception unit is an ESP32-CAM mounted on the robot, which captures images of the QR markers placed at the entrance, library, laboratories and other locations. The processing unit is a laptop running Python, which decodes the QR data, maintains the campus database and map, identifies the current location and calculates the route to the selected destination. The actuation unit consists of an Arduino UNO, a motor driver and DC motors, which execute the movement commands received from the laptop over serial communication. A touchscreen or voice interface lets the user choose a destination, ask for college information or receive reminders.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511127

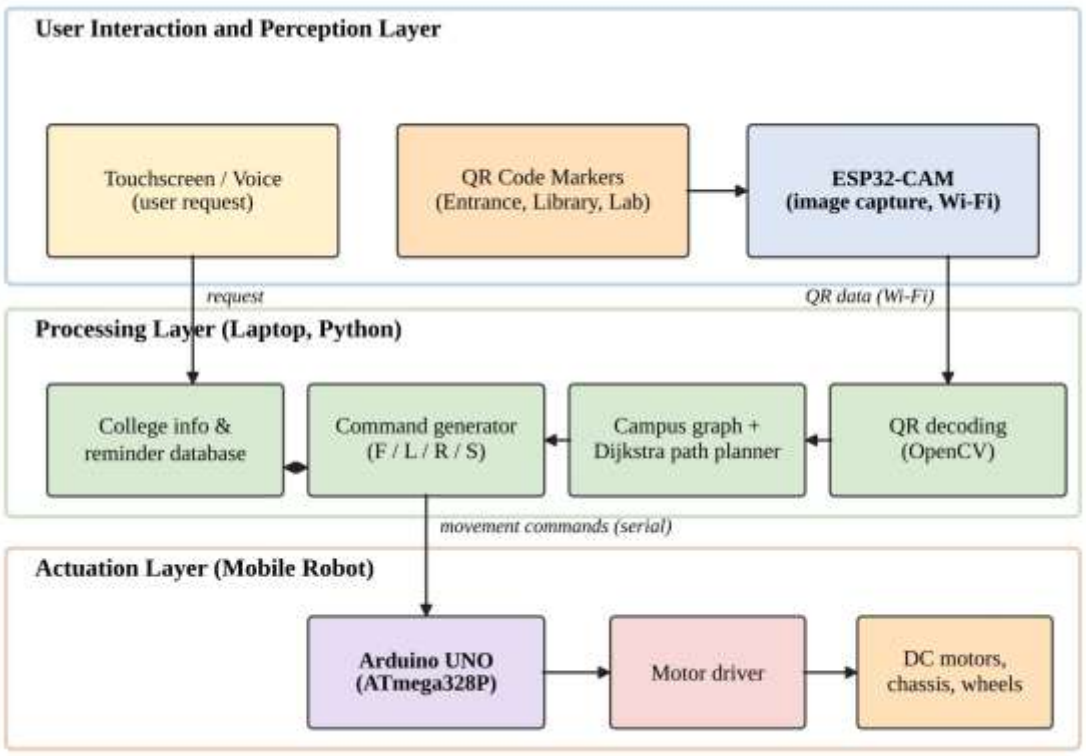



Fig. 1. Overall architecture of the proposed interactive humanoid robot.

***B) Problem Formulation***

The campus is modelled as a weighted graph in which each QR-marked location is a node and each walkable connection is an edge:

$$G = (V, E, w), \quad w: E \rightarrow R^{+} \tag{1}$$

where V is the set of QR-identified locations, E the set of connecting paths and w(u, v) the distance or movement cost of the edge between u and v. If s V is the location decoded from the latest QR code and d V is the destination selected∈ ∈ by the user, the navigation task is to find the minimum-cost path:

$$P^{*} = \arg\min \Sigma\, w(u, v), \quad (u, v)\ P, \quad P: s \rightarrow d\in \tag{2}$$

Dijkstra's algorithm solves this by repeatedly relaxing the tentative distance of every neighbour v of the node u with the smallest distance:

$$dist[v] = min(\, dist[v],\ dist[u] + w(u, v)\,) \tag{3}$$

Because the robot interacts with a person, the response from marker detection to the start of the motion command should also stay within an acceptable limit:

$$Tresp = Tcapture + Tdecode + Tplan + Tserial \leq Tmax \tag{4}$$

The design goal is therefore to identify the location reliably from QR markers, compute the minimum-cost route on a low-cost platform, and execute the movement commands so that the robot stops at the intended location while $T_{resp}$ stays within $T_{max}$.

 This work is licensed under a Creative Commons Attribution 4.0 International License 

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511127

## IV. PROPOSED FRAMEWORK

### *A) QR Code Placement and Image Acquisition*

Unique QR codes are printed and placed at predefined points such as the entrance, library, laboratories, classrooms and administrative office. Each code encodes a location identifier (and, where required, coordinates). When a marker enters the camera's view, the ESP32-CAM captures its image. The camera must have a suitable viewing angle and distance, and adequate lighting, so that the marker is captured clearly. The ESP32-CAM also provides Wi-Fi, which can be used to send images or decoded data to the laptop.

### *B) QR Decoding and Location Identification*

On the laptop, the captured image is processed with OpenCV and a QR decoding library to extract the location identifier. The identifier is matched with the campus database, which stores the location ID, location name, college-related information and task or reminder details. A successful match gives the robot's current location s; if no marker is detected or the identifier is invalid, the previous location is retained and the system does not update its position from invalid data.

### *C) Campus Mapping and Path Planning*

Classrooms, laboratories, the library and offices are stored as nodes of the graph G of Eq. (1), and the corridors between them as weighted edges. Fig. 2 shows an example campus graph in which Dijkstra's algorithm finds the route from the Entrance to Lab 1 through the Admin Office and the Corridor Junction (total cost 10). The database and graph can be extended by adding new location identifiers and connections, so the system can be adapted to other buildings or floors.

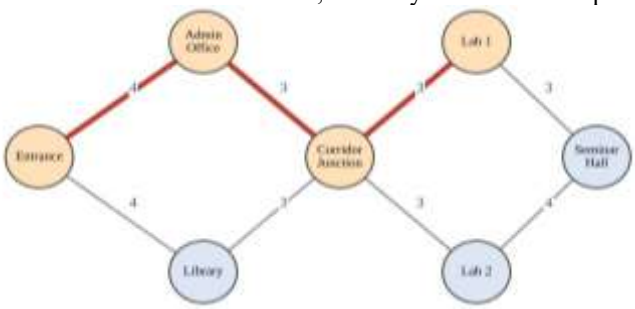



Fig. 2. Example campus graph with the shortest route (red) from Entrance to Lab 1.

### *D) Command Generation and Motor Control*

The computed route is converted into a sequence of simple movement commands: forward (F), left turn (L), right turn (R) and stop (S). The laptop sends these to the Arduino UNO over serial communication. The Arduino validates each command and generates digital and PWM signals for the motor driver, which supplies the current required by the DC motors. As the Arduino cannot drive the motors directly, the motor driver acts as the interface between the logic signals and the motor supply. The hardware connections are shown in Fig. 3.

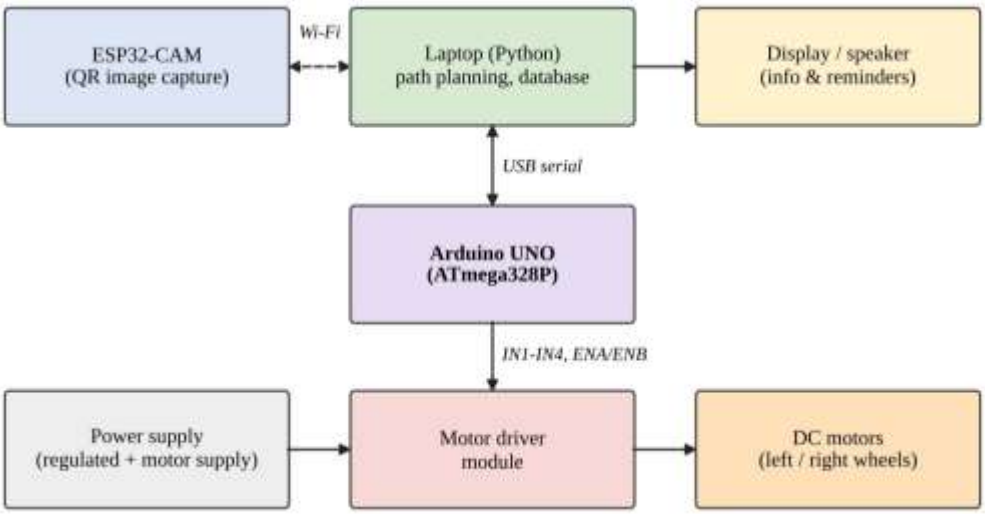



Fig. 3. Block diagram of the robot hardware and its interfaces.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511127

### *E) Information and Reminder Module*

Apart from navigation, the robot works as an interactive assistant. The laptop retrieves information about departments, classrooms, laboratories and other facilities from the database when the user asks for it. Stored tasks and events are used to generate reminders about academic schedules, meetings, laboratory sessions and other activities. The output is given through a display or voice prompts. These records are managed through the web-based application described in Section V.

### *F) Operating Procedure, Safety and Error Handling*

The complete operating sequence is shown in Fig. 4. The system is initialised, QR markers are scanned and decoded, the user selects a destination, the route is computed and sent as commands, and the robot stops when the destination marker is reached; otherwise scanning continues and the location is updated to verify progress along the route. For safe operation, a failed QR detection does not update the location, the robot enters a safe stop if serial communication between the laptop and Arduino is interrupted, the Arduino validates every command before driving the motors, and a suitable regulated supply is used for the controller, camera and motors.

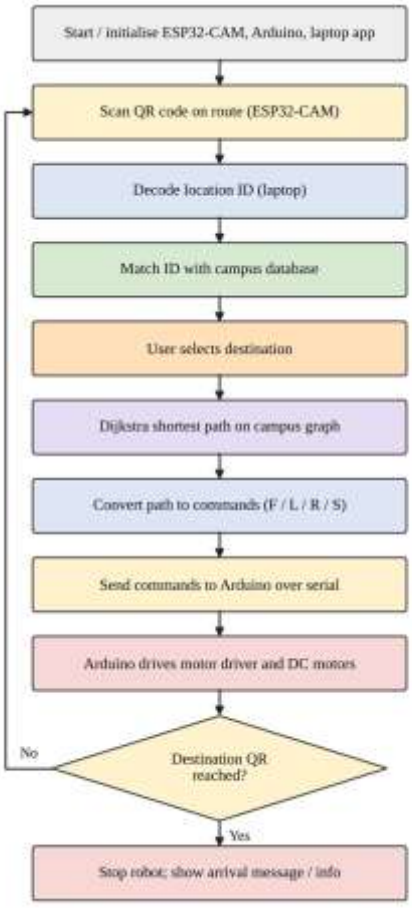



Fig. 4. Flowchart of QR-based navigation and movement control.

## V. EXPERIMENTAL SETUP

### *A) Hardware Environment*

The prototype uses the components listed in Table II. The ESP32-CAM is the visual input unit, the Arduino UNO (ATmega328P) is the motor control unit, and the laptop is the central processing unit. The robot moves along a predefined path marked on the floor, with a stopper mark as the reference point for stopping.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511127

TABLE II HARDWARE COMPONENTS OF THE PROTOTYPE

| Component | Function in the system | Interface |
|---|---|---|
| ESP32-CAM | Captures QR marker images; Wi-Fi connectivity | Wi-Fi / serial |
| Arduino UNO | Receives commands and controls the motor driver | Serial, digital / PWM pins |
| Motor driver module | Supplies motor current; controls direction and stop | Arduino digital pins |
| DC motors, wheels, chassis | Provide movement; mount the electronics | Motor driver |
| Laptop | QR decoding, graph path planning, database and interface | USB serial / Wi-Fi |
| QR code markers | Printed markers carrying unique location identifiers | Visual (camera) |
| Power supply | Regulated supply for controller and camera; motor supply | — |

### *B) Software Environment*

The Arduino IDE is used to program the Arduino UNO in embedded C/C++. Python is used on the laptop for location management, path calculation, database handling and serial communication with the Arduino. OpenCV and a QR decoding library handle image processing and decoding, and the campus database stores location identifiers, names, connections and information. The management application is a web-based Student Management System with a navigation sidebar and four modules: administrator dashboard, student registration, event management and task assignment. It provides the information, events and task reminders that the robot presents to users.

### *C) Test Protocol and Evaluation Parameters*

The web application is checked by verifying that the dashboard, student registration, event management and task assignment interfaces are available and operate correctly. The robot is tested for forward movement, turning, path following and stopping at the designated point along the marked path; it is not evaluated as a free-navigation system because its movement is restricted to the predefined route. For the complete system, the following parameters are defined: QR detection rate $R_{det} = N_{detected} / N_{scans}$; route correctness, i.e. whether the computed path matches the minimum-cost path; stopping accuracy at the destination; and the response time $T_{resp}$ of Eq. (4). Movement accuracy, movement stability and system integration are also observed qualitatively.

## VI. RESULTS AND DISCUSSION

### *A) Information Management Application*

Fig. 5 shows the outputs of the web application. The administrator dashboard summarises the number of registered and active students, upcoming events and the total, pending, in-progress and completed tasks, and gives quick actions for adding students, assigning tasks, adding events and viewing students. In the captured dashboard, one registered student, one active student and two pending tasks are shown. The registration form collects the name, email, phone number, date of birth, gender, enrollment number, course, department, year, password, address and photograph. The event page lists events with title, category, date, time and venue and supports filtering, editing, marking and deletion. The task form collects title, description, assigned student, priority and due date, and tasks are tracked as pending, in progress or completed.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511127

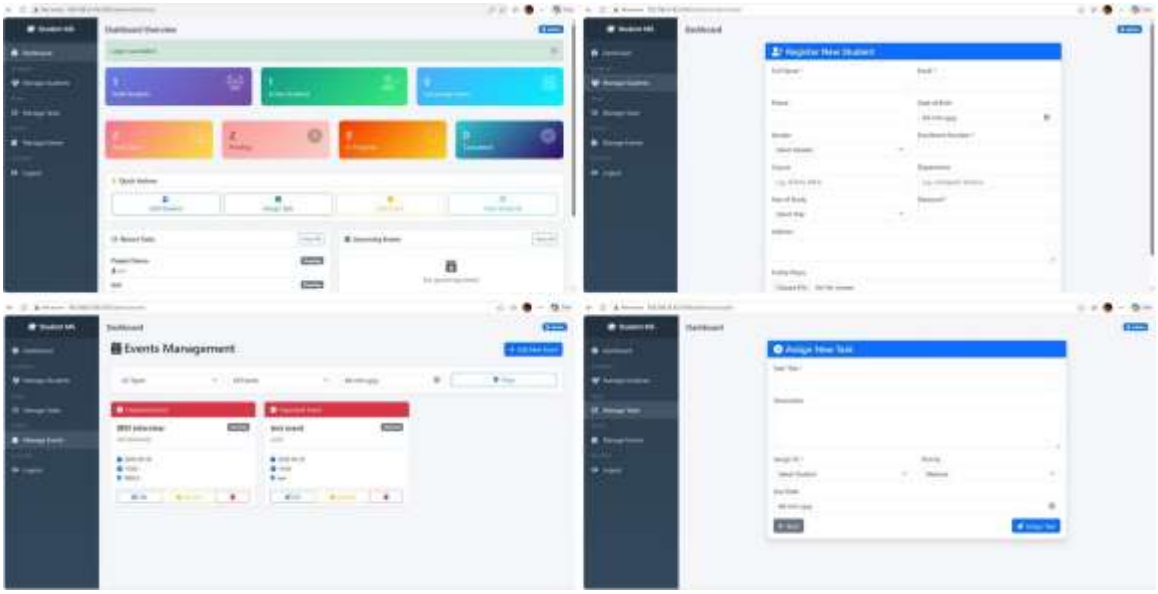



Fig. 5. Web application outputs: (a) administrator dashboard, (b) student registration, (c) event management, (d) task assignment.

***B) Robot Movement***

The robot successfully performs forward motion, turning and stopping by following predefined movement commands, which confirms the operation of the Arduino controller, motor driver and movement control logic. It follows the predefined path and stops when it reaches the designated stopper mark, showing that the control logic and hardware integration work as intended. The robot does not select routes independently or navigate through unmarked areas at this stage.

***C) Test Status***

Table III lists the current status of the functional tests. The web application functions have been observed in the application, while the numerical measurements for the robot are still to be recorded; the QR-based localization and graphbased routing are being integrated with the motion controller, so no detection-rate or accuracy figures are reported yet.

TABLE III FUNCTIONAL TEST STATUS

| No. | Test case | Expected operation | Status |
|---|---|---|---|
| 1 | Admin dashboard | Displays student, task and event summary | Observed |
| 2 | Student registration | Displays fields for student information | Observed |
| 3 | Event management | Displays event records and management options | Observed |
| 4 | Task assignment | Displays task, student, priority and due-date fields | Observed |
| 5 | Forward movement | Robot moves forward along the marked path | Demonstrated |
| 6 | Turning operation | Robot performs the programmed turn | Demonstrated |
| 7 | Path following | Robot stays on the predefined path | Demonstrated |
| 8 | Stopping operation | Robot stops at the designated stopping point | Demonstrated |
| 9 | QR detection, routing | Marker decoded; shortest route computed and executed | Under integration |

***D) Discussion***

The results show that the information-management side and the basic motion control of the robot are operational, which validates the hardware set-up and the motor control mechanism. The QR code-based approach is a cost-effective alternative to GPS-based indoor positioning and to LiDAR or SLAM, as it needs only a low-cost camera module, a microcontroller and printed markers. The distribution of work, with the ESP32-CAM sensing, the laptop planning and the Arduino actuating, keeps each module simple and allows it to be modified independently.



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511127

The main limitations are that the complete QR-based navigation is still under development, so localization accuracy, detection performance, path-planning efficiency and response time need further testing after integration. Detection depends on QR code visibility, lighting, image blur and marker placement; motion accuracy depends on motor calibration; and the laptop is needed for planning and database operations. Larger trials with repeated runs are required before general claims about navigation accuracy are made.

## VII. CONCLUSION

This paper presented an interactive humanoid robot for college information, task reminders and indoor navigation using QR codes and the ESP32-CAM. The ESP32-CAM identifies QR markers placed at campus locations, a laptop decodes them and computes the shortest route on a graph model of the campus using Dijkstra's algorithm, and an Arduino UNO executes the resulting movement commands through a motor driver. A web-based management application supplies student, event and task information for the assistance and reminder functions. The current implementation demonstrates the working web application and the basic robot movement of forward motion, turning, path following and stopping. The modular, low-cost design suits educational institutions and provides a foundation for the full campus assistant.

## VIII. FUTURE WORK

Future work will complete the integration of QR code-based localization with automatic path planning and measure detection rate, route correctness, stopping accuracy and response time. Further enhancements include obstacle detection and avoidance with ultrasonic or infrared sensors, voice-based interaction using speech recognition and text-to-speech, an AI-based conversational interface, face recognition with appropriate authorization and privacy protection, automated reminder scheduling, a mobile application, multi-floor campus navigation, cloud database integration and an improved chassis with a display and speaker.

## REFERENCES

[1] L. Yang, Y. Tao, M. Li, J. Zhou, K. Jiao and Z. Li, "Lidar-inertial SLAM method integrated with visual QR codes for indoor mobile robots," *Scientific Reports*, vol. 16, Art. no. 4020, 2026, doi: 10.1038/s41598-025-34165-2.
[2] S.-H. Bach, P.-B. Khoi and S.-Y. Yi, "Application of QR code for localization and navigation of indoor mobile robot," *IEEE Access*, vol. 11, pp. 28384–28390, 2023, doi: 10.1109/ACCESS.2023.3250253.
[3] H. Li and T. Xiong, "Cyber physical system information collection: Robot location and navigation method based on QR code," arXiv:2310.03470, 2023.
[4] R. Alqobali et al., "A survey on robot semantic navigation systems for indoor environments," *Applied Sciences*, vol. 14, no. 1, Art. no. 89, 2024.
[5] Y. Kim et al., "Application of QR codes for enhancing SLAM and localization in mobile robots," KoreaScience, 2025.
[6] A. Sampathkrishna, "ArUco marker based localization and graph mapping for indoor robots," arXiv:2208.09355, 2022.
[7] Y. Liu, "Localization and navigation system for indoor mobile robot with dynamic features tracking," arXiv:2212.06391, 2022.
[8] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
[9] G. Bradski, "The OpenCV library," *Dr. Dobb's Journal of Software Tools*, 2000.
[10] Espressif Systems, "ESP32 Series Datasheet," Shanghai, China. [Online]. Available: https://www.espressif.com
[11] Microchip Technology, "ATmega328P: 8-bit AVR microcontroller datasheet," Chandler, AZ, USA.

 This work is licensed under a Creative Commons Attribution 4.0 International License 