International Journal of Electrical, Electronics and Computer Systems
ISSN: 2347-2820
Volume 15 Issue 1s (2026) | Pages: 223-228

IoT-Enabled Smart Parking System with Slot Monitoring

Rushikesh Jadhav[1], Shivtej Karle[2], Paras Vishwakarma[3], Yash Marne[4], Mithali Thakur[5], Alka Kumbhar[6]

[1,2,3,4,5,6]Department of Computer Engineering, Genba Sopanrao Moze College of Engineering Balewadi, Pune, India

Peer Review Information

Type: *Article*
Received: *3 February 2026*
Revised: *4 March 2026*
Accepted: *5 April 2026*
Published: *22 May 2026*

Abstract

The rapid increase in vehicle population has created significant parking challenges, leading to congestion and inefficient space utilization. This paper presents the design and implementation of an IoT-based smart parking system for real-time slot monitoring in indoor and outdoor environments. The system uses ultrasonic sensors to detect vehicle presence, with an ESP32 microcontroller processing data and transmitting it via Wi-Fi to a cloud-based backend. The backend is developed using Node.js and Express, with MongoDB Atlas for data storage. A web dashboard built using React and Vite provides real- time visualization of parking availability. The system demonstrates reliable detection, real-time updates, and improved parking efficiency, making it a scalable and cost-effective solution for smart parking applications.

Keywords: Smart Parking; Internet of Things; ESP32; Ultrasonic Sensor; Real-Time Monitoring; Web Dashboard; Cloud Computing.

How to Cite This Article

Jadhav, R., Karle, S., Vishwakarma, P., Marne, Y., Thakur, M., & Kumbhar, A. (2026). IoT-Enabled Smart Parking System with Slot Monitoring. *International Journal of Electrical, Electronics and Computer Systems*, 15(1s), 223-228.

2347-2820/ © 2026 The Authors. This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0).

Introduction

The rapid growth in vehicle ownership has led to significant challenges in parking management, particularly in urban and semi-urban areas. Drivers often spend considerable time searching for available parking spaces, which contributes to traffic congestion, increased fuel consumption, and environmental pollution. Traditional parking systems rely heavily on manual monitoring and lack real-time information, resulting in inefficient utilization of available parking spaces. With the advancement of the Internet of Things (IoT), smart parking systems have emerged as an effective solution to address these challenges. IoT-based systems enable real-time monitoring of parking spaces using sensors, microcontrollers, and cloud-based technologies. These systems provide users with up-to- date information about parking availability, thereby reducing search time and improving overall efficiency.

In this paper, an IoT-enabled smart parking system is designed and implemented for both indoor and outdoor organized parking environments. The proposed system uses ultrasonic sensors installed at individual parking slots to detect vehicle presence. An ESP32 microcontroller collects sensor data and transmits it via Wi-Fi to a cloud-based backend system. The backend is developed using Node.js and Express, with MongoDB Atlas used for data storage and management. A web-based dashboard built using React provides a real-time grid-based visualization of parking slot availability. The main objective of this work is to develop a cost- effective, scalable, and reliable parking solution that can be deployed in environments such as shopping malls, educational campuses, hospitals, and public parking facilities. The implementation focuses on accurate slot detection, real-time data updates, and user-friendly visualization to enhance parking efficiency.

Literature Review

Several approaches have been proposed for smart parking systems, including sensor-based, camera- based, RFID-based, and reservation-based methods. Key findings from important survey and research works are summarized below. Sensor-Based IoT Parking Systems Many systems use IR or ultrasonic sensors at each slot. Sensors detect vehicle presence and send slot status to a controller. These systems are low-cost, easy to install, and suitable for small to medium parking areas. However, sensor maintenance and environmental conditions (dust, rain) can affect accuracy.

Camera-Based (Computer Vision) Parking Systems Vision-based systems use cameras and image processing to identify vacant slots. These systems provide high accuracy and cover a large area with fewer devices, but require high installation cost, good lighting, and higher computation. RFID-Based Parking Systems RFID-based parking provides improved gate access control and security. It supports quick entry/exit operations but requires RFID tags and infrastructure, which increases setup cost.

Reservation and Recommendation-Based Systems Advanced systems allow users to reserve a slot via mobile app and sometimes recommend a nearby vacant slot based on traffic and historical occupancy. These systems improve user experience but depend strongly on internet availability and cloud performance.

Methodology

The proposed smart parking system is designed using an Internet of Things (IoT)-based approach that integrates sensing, processing, communication, and visualization to enable real-time parking slot monitoring. The methodology is structured into multiple stages, ensuring efficient data acquisition, processing, and user interaction. Sensor Deployment and Data Acquisition Ultrasonic sensors are deployed at each parking slot to detect the presence of vehicles. These sensors operate by emitting ultrasonic waves and measuring the time taken for the reflected signal to return. Based on this time-of-flight measurement, the distance between the sensor and the object is calculated. A significant reduction in distance indicates the presence of a vehicle, while a larger distance represents an empty slot.

The sensors continuously monitor the parking space and send distance measurements to the microcontroller for further processing.

Data Processing using ESP32

An ESP32 microcontroller is used to process the sensor data. The ESP32 reads the distance values from the ultrasonic sensors and applies a threshold-based algorithm to determine the occupancy status of each parking slot.

If the measured distance is less than a predefined threshold value, the slot is marked as occupied; otherwise, it is considered vacant. This processing is performed in real time to ensure accurate and timely updates.

Data Transmission via Wi-Fi

After determining the slot status, the ESP32 transmits the processed data to the backend server using Wi-Fi communication. The built-in Wi-Fi capability of the ESP32 enables seamless integration with the network, eliminating the need for additional communication modules.

Data transmission is performed using HTTP requests, ensuring reliable and efficient communication between the hardware layer and the server.

Backend Processing and Data Storage

The backend of the system is developed using Node.js and Express, which handles incoming data from the ESP32. The server processes the received data and updates the parking slot status in a cloud-based database.

MongoDB Atlas is used as the database for storing real-time and historical parking data. The cloud-based nature of the database ensures scalability, accessibility, and efficient data management.

Data Visualization

A web-based dashboard is developed using React and Vite to provide users with real-time information about parking availability. The frontend application retrieves data from the backend server through APIs and displays it in a grid-based layout.

Each parking slot is visually represented, indicating whether it is occupied or available by use of colour coding. This enables users to quickly identify free parking spaces.

System Workflow

The overall workflow of the system can be summarized as follows:

- Ultrasonic sensors detect vehicle presence.
- ESP32 processes sensor data and determines slot status.
- Data is transmitted via Wi-Fi to the backend server.
- Backend updates the MongoDB Atlas database.
- Web dashboard displays real-time parking availability.

System architecture

The proposed smart parking system is designed using a layered IoT architecture that enables real-time monitoring of parking slot occupancy. The system integrates sensing, communication, cloud processing, and user interface components to provide an efficient and scalable parking solution for both indoor and outdoor environments. The overall system architecture consists of four main layers: (i) sensing layer, (ii) processing layer, (iii) communication layer, and (iv) application layer.

Sensing Layer

The sensing layer comprises ultrasonic sensors installed at individual parking slots. Each ultrasonic sensor measures the distance between the sensor and the object above it. When a vehicle is present in the parking slot, the measured distance decreases below a predefined threshold, indicating that the slot is occupied. In the absence of a vehicle, the measured distance remains higher, indicating that the slot is vacant. This approach enables accurate and real-time detection of parking occupancy.

Processing Layer

The processing layer is implemented using the ESP32 microcontroller. In this system Single sensor is connected to single ESP32 microcontroller for each slot. The microcontroller reads sensor data, processes it based on threshold logic, and determines the occupancy status of each parking slot. The ESP32 also prepares the data in a structured format for transmission

Communication Layer

The communication layer is responsible for transmitting processed data from the ESP32 to the cloud server. Wi-Fi connectivity is used for data transmission due to its high data rate, ease of integration, and suitability for organized parking environments. Each ESP32 connects to a nearby Wi-Fi access point and sends real-time updates to the backend server using HTTP protocols.

Application Layer

The application layer consists of the cloud backend and the user interface. The backend is developed using Node.js, Express and is responsible for handling incoming data, processing requests, and updating the database. MongoDB Atlas is used as a cloud database to store parking slot information and maintain real- time status.

Data Flow

The data flow in the system begins with the ultrasonic sensors detecting slot occupancy. The sensor data is read and processed by the ESP32, which determines the slot status. The processed data is then transmitted via Wi-Fi to the cloud server, where it is stored in the database. Finally, the web dashboard fetches updated data from the server and displays the current parking status to the user.

The proposed architecture is scalable and the modular design ensures ease of maintenance, cost- effectiveness, and reliable real-time performance.

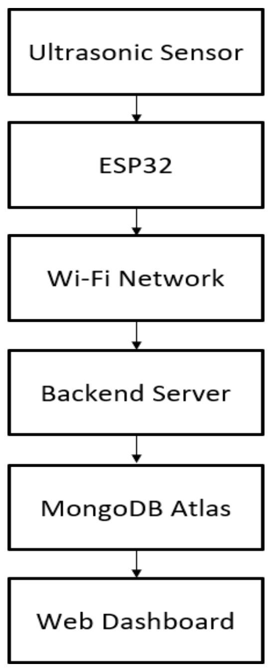

Fig. 1. *System Architecture of Proposed Smart Parking System*

Ultrasonic Sensor

Ultrasonic sensors are used for detecting the presence of vehicles in individual parking slots. In this system, the HC-SR04 ultrasonic sensor is employed due to its low cost, ease of integration, and reliable distance measurement capabilities. Each sensor emits ultrasonic waves and measures the time taken for the echo to return after reflecting from an object.The sensor is installed at ground level within each parking slot and measures the distance between the sensor and the object above it.

***Fig. 2**. Ultrasonic Sensor*

When a vehicle is present, the measured distance decreases significantly due to the proximity of the vehicle's underside. In the absence of a vehicle, the sensor detects a larger distance. A predefined threshold value is used to classify the slot status as either occupied or vacant.

ESP32 Microcontroller

The ESP32 microcontroller serves as the main processing unit in the system. It is responsible for reading sensor data, processing occupancy status, and transmitting data to the cloud server. The ESP32 is selected due to its integrated Wi-Fi capability, low power consumption, and sufficient processing power for real-time applications.

In this system Single sensor is connected to single ESP32 microcontroller for each slot. The microcontroller reads distance values from the sensors, applies threshold logic to determine occupancy, and formats the data for transmission.

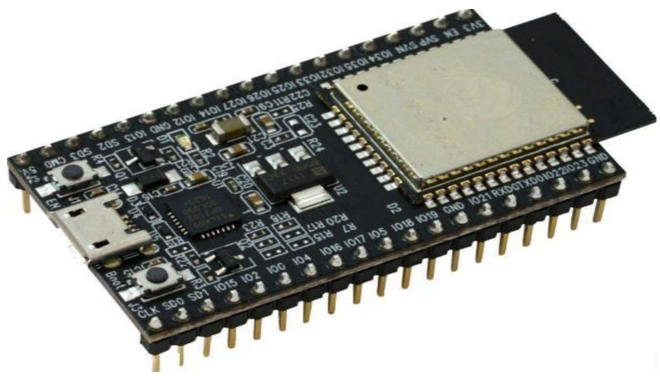

***Fig. 3**. ESP32 Microcontroller*

Interfacing and Connections

The ultrasonic sensor is interfaced with the ESP32 using GPIO pins. The sensor's TRIG and ECHO pins are connected to digital input/output pins of the ESP32. The ESP32 generates trigger pulses and measures the echo response time to calculate distance.

Jumper wires are used for connections during prototyping, and all components are mounted within a protective enclosure to safeguard against environmental factors such as dust and moisture, especially for outdoor deployment.

Table 1. Components

Component	**Specification**	**Purpose**
ESP32	Wi-fi enabled MCU	Data Processing & communication
HC-SR04 SENSOR	2-400 cm range	Vehicle Detection
3.7V Li-ion Battery	18650 Type	Power supply
Jumper Wires	-	Connections
Enclosure	Plastic casing	Protection

Results and Discussion

The proposed IoT-based smart parking system was implemented and tested in a prototype environment consisting of multiple parking slots equipped with ultrasonic sensors and ESP32 microcontrollers. The system was evaluated based on parameters such as detection accuracy, response time, and system reliability for indoor and outdoor conditions.

Experimental Setup

The prototype system was deployed with ultrasonic sensors installed at individual parking slots. Each slot has one ESP32 microcontroller connected to one sensor. The ESP32 transmitted slot status data via Wi- Fi to a Node.js backend server, which stored the data in MongoDB Atlas. A React-based web dashboard was used to visualize real-time parking availability.

Detection Accuracy

The ultrasonic sensor-based detection mechanism was tested under different conditions. The system achieved higher detection accuracy under controlled indoor environments. For outdoor conditions, slight variations were observed due to environmental factors such as uneven surfaces and external disturbances, resulting in lower accuracy compared to controlled environment. The results indicate that ultrasonic sensors provide reliable performance for parking slot detection, especially in structured parking environments.

Response Time Analysis

The response time of the system was measured from the moment a vehicle occupies or vacates a slot to the update displayed on the web dashboard.

1.Sensor detection and processing delay: ~1–2 seconds

2.Data transmission via Wi-Fi: ~1–3 seconds

3.Backend processing and database update: ~1 second

4.Frontend update delay: ~1–2 seconds

The total system response time was observed to be approximately 3–6 seconds, which is suitable for real- time parking applications.

System Reliability and Scalability

The system demonstrated stable performance during continuous operation. The cloud-based MongoDB Atlas database ensured reliable data storage and accessibility.

Conclusion

This paper presented the design and implementation of an IoT-based smart parking system for indoor and outdoor environments. The system uses ultrasonic sensors for slot detection, ESP32 for processing, and Wi-Fi for real-time data transmission, with a Node.js and Express backend and MongoDB Atlas for cloud storage. A React-based dashboard provides real-time visualization of parking availability. The results show that the system achieves reliable detection and acceptable response time, making it suitable for real- time applications. The solution is scalable and cost- effective for deployment in various parking environments. Future improvements can focus on enhancing sensor reliability and communication robustness. Overall, the system improves parking efficiency by reducing search time and optimizing space utilization.

References

1. Muhamed Alarbi, Abdelikareem Jaradat, Hanan Lutfiyya, Anwar Haque., “SCOPE: Smart Cooperative Parking Environment” IEEE Access, 2023.
2. H. Tanti et al., “Smart Parking System based on IoT,” 2020.
3. I. B. Asianuba and A. Aliyu, “An Improved IoT Smart Parking System,” 2023.
4. M. Alarbi et al., “SCOPE: Smart Cooperative Parking Environment,” 2023.
5. Anika et al., “Smart Parking for Urban Cities Using IoT and Edge AI,” 2023.
6. T. Hovorushchenko et al., “Method of Increasing the Security of Smart Parking System,” 2023.
7. S. Yan et al., “U-Park: A User-Centric Smart Parking Recommendation System for Electric Shared Micromobility Services,” 2024.
8. A. Javaheri et al., “Evaluating the Impact of Smart Parking Systems on Parking Violations,” 2024.
9. M. Turki et al., “Puf Park Chain: Secure and Smart Parking Based on PUF Authentication and Lightweight Blockchain,” 2024.
10. Love Kumar, Manoj Varshney “Smart Parking: A New IoT-Based Ideas for Vehicle Parking”, 2024.
11. P. Rajesh et al., “Intelligent Parking Guidance System Using Computer Vision, IoT, and Edge Computing,” 2025.
12. S. S. Channamallu et al., “Smart Parking Systems: A Comprehensive Review of Digitalization of Parking Services,” 2025.