

Quicky: Anti-Theft Flooring System

Mr. Aditya Ray
Student, Dept. of CSE-IoT,
GIFT Autonomous, Bhubaneswar

Mr. Satyajeet Sahoo
Student, Dept. of CSE-IoT,
GIFT Autonomous, Bhubaneswar

Asst. Prof. Dr. Soumendra Prasad Rout
Asst. Professor, Dept. of CSE,
GIFT Autonomous, Bhubaneswar

Abstract—Quicky is a real-time IoT security and anti-theft flooring system designed to improve the safety and monitoring of residential and commercial spaces. The system provides live intrusion detection using physical pressure-sensitive flooring mechanisms and IoT integration technologies, helping property owners and security authorities monitor premises in real time. It also includes remote arm/disarm capabilities, emergency alerts, historical activity logging, and multiple device management features. The hardware relies on ESP32 microcontrollers and Force Sensitive Resistors (FSR/Piezo), while users can manage the system through a centralized web dashboard. The system is developed using Vanilla JavaScript, HTML5, CSS3, and Firebase (Realtime Database and Authentication). Overall, Quicky improves security response times, reduces hardware dependency on traditional motion sensors, and provides a reliable solution for modern smart property management.

Keywords—IoT Security, Anti-Theft Flooring, Real-Time Monitoring, Smart Home Security, Web Application, ESP32, Emergency Alert System.

I. INTRODUCTION

In modern residential and commercial infrastructures, security systems play an important role in ensuring the safety of property and individuals. Traditional security systems mainly depend on fixed schedules, standard PIR (Passive Infrared) motion sensors, and localized sirens, which often create limitations in remote monitoring and real-time awareness for property owners [1]. Due to increasing security threats and changing lifestyle dynamics, it becomes difficult to monitor the exact status of a property and detect unauthorized entry using conventional localized methods. The lack of proper real-time remote communication and monitoring facilities also affects security response times and operational efficiency.

Recent advancements in Internet of Things (IoT) technologies, Progressive Web Apps (PWA), and real-time data synchronization have enabled the development of intelligent security management systems [2]. Real-time IoT systems allow users to monitor premises continuously and improve coordination between property owners, security personnel, and local authorities. These systems provide better transparency, reduce unnecessary false alarms, and improve overall security management [3]. However, many existing security systems mainly focus on localized alarms and do not provide complete operational management functionalities such as multi-device tracking, cloud-based activity logging, remote arming, and instant emergency push notifications.

The proposed Quicky system is designed as a centralized real-time anti-theft flooring platform that integrates physical hardware-level intrusion detection, live communication, and security management into one system. The platform provides

live pressure-based intrusion tracking using ESP32 microcontrollers and Piezo/FSR sensors hidden beneath the physical flooring. It also includes live dashboard monitoring, emergency alert handling, historical activity recording, and centralized administration facilities [4]. The hardware can detect unauthorized weight and update live status data, while property owners can monitor devices, system status, and security logs through a centralized web dashboard.

This paper focuses on the design and implementation of the Quicky real-time anti-theft flooring system. Section II discusses the challenges in existing security systems. Section III explains the proposed system architecture and working model. Section IV describes the methodology and technologies used for implementation. Section V discusses system design and implementation details. Section VI presents results and discussion, while Section VII concludes the paper with future enhancement possibilities.

II. CHALLENGES IN EXISTING SYSTEM

Traditional property security systems face several operational and communication-related challenges due to the absence of centralized digital monitoring platforms. Most security setups still depend on localized sirens and delayed notification methods for managing intrusions. These methods are often inefficient in handling dynamic security conditions such as power outages, blind spots in camera coverage, and remote tampering [5]. As a result, property owners frequently face difficulties in obtaining accurate and instant security information.

A. Lack of Real-Time Remote Monitoring

One of the major problems in traditional security systems is the absence of real-time remote monitoring facilities. Property owners generally depend on localized alarms without knowing the exact status of the intrusion. During a break-in, users remain uncertain about the exact location of the breach, leading to delayed responses [6]. Security authorities also face difficulties in continuously monitoring property conditions without expensive, dedicated monitoring subscriptions.

The absence of live tracking reduces operational transparency and creates communication gaps between users and their premises.

B. Delayed Intrusion Alerts

Effective communication is essential for security management, especially during emergencies. In traditional security systems, communication between the hardware and the owner mainly depends on outdated SMS gateways or local alarms. These methods are often slow and unreliable during

critical situations such as break-ins or vandalism.

The lack of centralized push-notification facilities creates inconvenience for both security administrators and users. Owners may not receive immediate updates regarding intrusions, which can create safety concerns.

C. Lack of Activity Logging

Another major challenge in existing localized security systems is the lack of digital record management. Security-related records such as the exact time of an alarm, device status history, and system arm/disarm logs are often untracked or stored locally on easily damaged hard drives [8]. Localized record handling makes security auditing more time-consuming and less efficient.

Maintaining records locally also increases the possibility of data loss if the hardware is tampered with by an intruder.

D. Safety and Monitoring Issues

Security reliability is one of the most important concerns in anti-theft systems. Traditional systems using PIR (motion) sensors are easily triggered by pets or environmental changes, leading to false alarms [9]. Furthermore, these systems do not provide centralized web monitoring or interactive emergency alert mechanisms (such as instant "Call Police" features) for quickly responding to critical situations. The absence of continuous cloud monitoring reduces operational visibility and makes it difficult for authorities to ensure property safety effectively.

E. Need for a Smart IoT Security System

The limitations of traditional security systems highlight the need for a modern and intelligent IoT security management platform capable of integrating real-time hardware detection, centralized communication, operational monitoring, historical logging, and emergency management into one system [10].

A smart anti-theft flooring system can improve detection accuracy, security safety, and operational transparency while reducing false alarms. The integration of ESP32 microcontrollers, Firebase real-time synchronization, and a Progressive Web App (PWA) can significantly improve the overall efficiency of modern property security.

III. PROPOSED SYSTEM

The proposed Quicky system is designed to provide a smart and centralized solution for managing property security in real time. The system mainly focuses on improving detection accuracy, communication speed, operational transparency, and remote monitoring efficiency. Traditional security systems mostly depend on motion sensors and local alarms, which are not suitable for handling blind spots and remote property management. To overcome these limitations, Quicky integrates real-time physical pressure-sensitive floor tracking, web-based communication facilities, activity logging, multi-device management, and centralized monitoring into a single PWA platform.

The system allows property owners to monitor the live status of their flooring sensors through a digital dashboard interface. The hardware can instantly sound local buzzers and

update cloud data, while users can remotely arm or disarm the system. Property administrators can continuously monitor security activities through a centralized dashboard and manage multiple ESP32 devices efficiently.

A. System Overview

Quicky follows an IoT client-server architecture where the physical hardware (ESP32), frontend, backend (Firebase), and tracking modules work together to provide real-time security management. The frontend interface is developed using Vanilla JavaScript, HTML, and CSS, and communicates with the backend via Firebase SDKs. The Firebase Realtime Database acts as the central server to handle user requests, process telemetry data, and manage state synchronization.

The system continuously reads analog data from Force Sensitive Resistors (FSR/Piezo). Whenever an intrusion is detected, the ESP32 updates the database. Whenever a user opens the dashboard, the frontend fetches this updated device status and displays it on the UI. This allows users to monitor the current security state of their premises in real time.

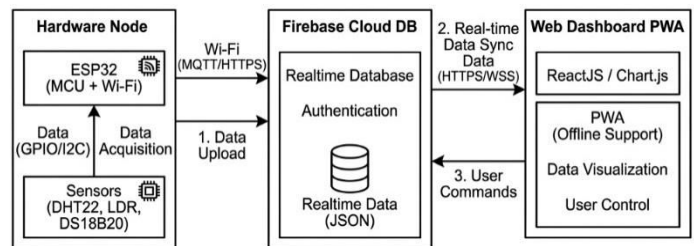


Fig. 1. Overall Architecture of Quicky System

Apart from tracking facilities, the system also provides operational management functionalities such as historical activity logging, multiple device integration, and interactive emergency alert handling (e.g., "I'm Safe" or "Send Help"). The centralized design of the platform simplifies security management and improves transparency.

B. User / Owner Module

The user module is designed for property owners and security staff to monitor premises. Users can track the live status of their flooring sensors through an interactive dashboard and receive instant push notifications regarding unauthorized movement. The module also allows users to remotely toggle the "Arm" and "Disarm" state of the physical hardware directly from the web interface.

C. Hardware (ESP32) Module

The hardware module plays an important role in maintaining real-time physical security. The ESP32 microcontroller continuously polls the analog Piezo sensors. When the system is armed and pressure is detected, the ESP32 triggers a localized physical buzzer, activates warning LEDs, and immediately pushes an "isRinging" state to the Firebase Realtime Database.

The Wi-Fi-enabled ESP32 continuously sends heartbeat signals to the backend server to ensure the device is online and actively monitoring.

D. Admin Module

The admin module acts as the central control unit of the Quicky platform. Administrators can monitor all security activities through a centralized dashboard that displays active hardware devices, historical logs, system statuses, and user profiles in one interface. This centralized approach allows property managers to oversee multiple building zones or separate properties from a single screen.

E. Working Flow of the System

The working process of Quicky begins when the ESP32 device continuously sends heartbeat signals and physical sensor data to the Firebase backend through Wi-Fi connectivity. The backend processes the incoming telemetry and stores the updated states in the Realtime Database. The frontend uses WebSocket connections to instantly fetch and reflect the latest security data dynamically on the dashboard.

Whenever users access the dashboard, the system displays the current status of their active hardware devices. If any physical intrusion is detected on the floor, the information is directly stored in the database and a full-screen red warning modal is forwarded to the dashboard for immediate action.

During emergencies, users can generate emergency alerts ("Call Police") that are instantly logged. The complete workflow of the system helps improve security safety, communication efficiency, and operational coordination among all stakeholders involved in property management.

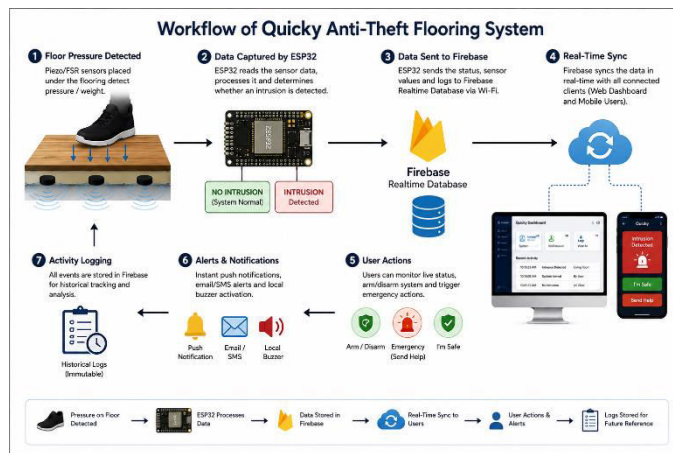


Fig. 2. Workflow of Quicky System

IV. METHODOLOGY

The development of the Quicky system follows a structured and modular approach to ensure efficient implementation and smooth security management. The methodology mainly focuses on requirement analysis, hardware integration, frontend development, database integration, and real-time communication. The complete development process is carried out using modern web and IoT technologies and follows the Agile software development methodology.

A. Requirement Analysis

Requirement analysis is one of the most important stages in system development. In this phase, the major problems in traditional security systems are studied carefully to identify the functional and non-functional requirements of the Quicky

platform.

The functional requirements include real-time physical pressure tracking, remote arm/disarm, hardware buzzer control, emergency alert handling, device provisioning, and centralized administrative monitoring. The non-functional requirements mainly focus on system scalability, PWA responsiveness, real-time latency (sub-second updates), reliability, and data security.

B. Frontend Development

The frontend of the Quicky system is developed using Vanilla JavaScript, HTML5, and CSS3 to provide a lightweight, highly responsive, and user-friendly interface. A framework-less approach was selected to ensure maximum performance and minimal load times [11].

The user interface features a "Dark Premium" design system utilizing Glassmorphism and CSS variables for a modern aesthetic. It acts as a Single Page Application (SPA), dynamically hiding and showing views without page reloads. The frontend continuously communicates with Firebase to fetch updated security telemetry.

C. Hardware & Backend Development

The hardware backend of the system is driven by an ESP32 microcontroller programmed in C++ using the Arduino IDE. The ESP32 is responsible for reading analog signals from the physical piezo sensors, managing local GPIO outputs (LEDs and Buzzers), and communicating with the internet.

Firebase Realtime Database (RTDB) acts as the cloud backend. Firebase was selected because of its low-latency WebSocket connections, making it ideal for real-time IoT synchronization [12]. The database continuously receives status updates from the hardware and manages authentication, device ownership, and historical logging.

D. API Integration and Real-Time Communication

Firebase SDKs are implemented to support communication between the frontend and the cloud database. The system continuously updates physical sensor statuses using real-time data transmission from the ESP32. Whenever the hardware detects pressure, the backend pushes the latest data to the frontend interface dynamically via 'on(value)' listeners. The implementation of real-time WebSocket communication significantly improves tracking accuracy and monitoring efficiency compared to traditional REST APIs [13].

E. System Testing

System testing is performed to ensure proper functionality and performance of all modules. Different testing techniques are used to verify hardware sensor accuracy, Wi-Fi reconnection stability, user authentication, dashboard responsiveness, and emergency alert response. Physical hardware prototypes were utilized extensively to test the frontend response under real-world intrusion scenarios. Testing results show that the Quicky system can efficiently handle real-time security management operations.

V. SYSTEM DESIGN AND IMPLEMENTATION

The Quicky system is designed using a centralized IoT client-server architecture that supports real-time remote monitoring and operational management. The complete system

consists of a PWA frontend, Firebase backend, ESP32 hardware tracking components, and continuous synchronization.

A. System Architecture

The overall architecture of Quicky consists of the hardware node, cloud database, and web dashboard that work together to support real-time monitoring. The frontend layer displays security information and user interfaces. The Firebase layer handles business logic, user authentication, and data synchronization.

The ESP32 hardware layer continuously polls physical floor sensors and updates the cloud. This architecture allows users to monitor their property in real time and improves security transparency.

B. Live Intrusion Detection System

The live intrusion module is undeniably the core, beating heart component of the Quicky platform. This module empowers users to deeply monitor the real-time physical state of their property's floors.

The ESP32 continuously and rapidly reads analog voltage from the hidden Piezo sensors using internal pull-down resistors to eliminate noise. If the voltage crosses a specifically defined threshold while the system is actively armed, it registers a confirmed step.

The ESP32 immediately updates the "isRinging" node in Firebase, and the frontend dynamically displays a full-screen red intrusion modal, simultaneously playing an extremely loud audible warning directly through the user's web browser.

C. Device and State Management

The system is intelligently designed to effortlessly handle multiple ESP32 physical devices under a single, unified user account. Users can seamlessly provision a brand-new device by simply entering its unique alphanumeric Device ID.

The dashboard beautifully allows users to switch between active hardware units, view their specific heartbeat (last seen timestamp) status to ensure they are online, and deeply control their specific armed/disarmed states entirely independently of one another.

D. Emergency Alert and Notification System

High-stakes security emergencies require immediate, fail-proof communication. The Quicky platform includes an advanced emergency alert system integrated directly into the core intrusion modal.

When an intrusion is undeniably detected, a high-stress countdown timer automatically begins. The user can interact with the dashboard to either declare "I'm Safe" (which immediately disarms the physical hardware) or "Send Help" (which triggers an immutable emergency state).

Persistent push notifications via Firebase Cloud Messaging (FCM) are also implemented to reliably alert the user even when the web application is completely closed.

E. Admin Dashboard Implementation

The admin dashboard serves as the central monitoring and control system for the entire platform. It provides

administrators with a clear and organized interface to observe the real-time status of connected hardware devices. Through this dashboard, users can instantly check whether alarms are active, monitor system performance, and ensure that all security components are functioning properly without needing physical access to the hardware setup.

In addition to live monitoring, the dashboard allows administrators to remotely manage the alarm system with minimal delay. Users can activate or deactivate alarms instantly, making the system highly responsive during emergency situations. The platform also maintains detailed historical activity logs, enabling administrators to review past intrusion events, analyze system behavior, and track important security-related actions whenever required.

Another important feature of the dashboard is its secure cloud-based data storage system. Since all records are digitally stored in the cloud, users do not need to rely on vulnerable local hardware storage methods that may be damaged or lost over time. This centralized approach not only simplifies data access and management but also reduces manual effort while significantly improving the overall efficiency, reliability, and effectiveness of security management.

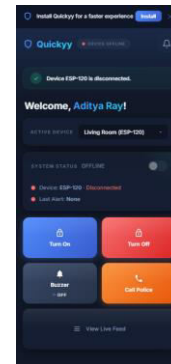


Figure 3. User/Owner Monitoring Dashboard

F. Advantages of the Proposed System

The Quicky platform provides several profound advantages compared to traditional security management systems. The integration of real-time physical pressure tracking, centralized web communication, and digital management functionalities massively improves property safety, transparency, and operational efficiency.

The Quicky platform provides specific advantages including real-time physical intrusion tracking, significantly improved communication between the hardware and property owners, faster emergency response times, centralized multi-device monitoring, and highly secure cloud-based historical logging. By utilizing physical flooring sensors rather than traditional PIR (motion) sensors, the system drastically reduces false alarms caused by pets or environmental changes.

The proposed system radically improves the coordination between property owners, security staff, and law enforcement while reducing operational complexity. The use of modern web technologies (PWA) and centralized cloud synchronization provides a highly scalable, reliable, and fail-proof solution for modern smart-property security.

Traditional Security System	Quicky System
Localized physical alarm panels	Centralized web dashboard (PWA)
No remote real-time tracking	Real-time cloud-based intrusion tracking
Localized data storage on fragile hard drives	Secure cloud-based digital data management
Only localized sirens	Integrated web emergency alert & push notifications
Physical keypad access required	Remote arming/disarming from anywhere
High false alarm rate (triggered by pets/wind)	Very low false alarm rate (requires physical footprint)
Difficult to monitor system health	Centralized admin dashboard with device heartbeats

Table 1. Comparison of Traditional System and Quicky

VI. RESULTS AND DISCUSSION

The Quicky system was successfully implemented and thoroughly tested to ensure reliable real-time security monitoring and effective anti-theft management. Multiple core components of the system, including hardware sensor accuracy, dashboard synchronization, and emergency alert response time, were evaluated under different operating conditions. These tests were conducted to measure the system's overall performance, stability, and efficiency in handling security-related events.

The testing results demonstrated that the system is capable of providing accurate and continuous monitoring with fast communication between the hardware and the web platform. The ESP32-based hardware devices were able to detect events efficiently and instantly transmit data to the cloud through Firebase. At the same time, the centralized dashboard updated in real time, allowing administrators to monitor the security status and respond quickly whenever suspicious activity was detected.

The final implementation clearly shows that the proposed Quicky system offers a more advanced and efficient alternative to traditional security systems. By combining ESP32 technology, Firebase real-time communication, and centralized web monitoring, the system improves transparency, reduces response delays, and enhances coordination between property owners and the security infrastructure. This integration ultimately leads to better security management, improved reliability, and a more user-friendly monitoring experience.

A. System Performance Analysis

The overall performance of the Quicky platform was critically analyzed based on hardware tracking responsiveness, cloud database sync time, and long-term hardware reliability. The ESP32 was able to continuously process physical sensor updates and reflect them flawlessly on the web dashboard with an impressive sub-second latency margin.

The Vanilla JavaScript frontend interface remained incredibly responsive during continuous data polling and handled multiple complex state changes efficiently without any memory leaks or browser crashes. Firebase provided remarkably fast storage and retrieval of security records, logs, and tracking information.

B. Detection Accuracy

The physical Piezo and FSR sensors demonstrated accurate and reliable pressure detection throughout the entire testing process. During implementation, the sensors continuously monitored pressure changes and successfully identified human footsteps under different testing conditions. The system maintained stable performance while ensuring that unwanted environmental disturbances did not interfere with normal operation.

To improve detection accuracy, the ESP32 utilized internal pull-down resistors along with carefully calibrated analog voltage thresholds in the C++ firmware. This configuration helped the system differentiate between actual human movement and small environmental vibrations such as minor surface disturbances or background noise. As a result, the hardware was able to provide consistent sensing performance with reduced chances of incorrect detection.

The testing results showed that this approach significantly lowered the false alarm rate when compared to conventional PIR motion sensor systems. Unlike traditional motion sensors that may react to unnecessary movement or temperature changes, the pressure-based sensing mechanism offered more dependable real-world performance. This made the system highly suitable for practical anti-theft and security monitoring applications where accuracy and reliability are essential.

C. Communication Efficiency

The integration of the Firebase Realtime Database significantly improved communication and data synchronization between the physical hardware and the end-user application. By enabling real-time cloud connectivity, the system was able to instantly transfer security-related data from the ESP32 hardware to the centralized dashboard. This ensured that users could continuously monitor system activity without noticeable delay.

The emergency alert module also played an important role in improving response efficiency during intrusion situations. Whenever suspicious activity was detected, the platform immediately generated visual alerts on the user dashboard, allowing property owners to react quickly and make informed decisions. The real-time notification mechanism enhanced overall coordination between the monitoring system and the user, especially during critical security events.

Compared to traditional SMS-based security systems, the Quicky platform delivered much faster communication and more reliable updates. Conventional systems often suffer from network delays and limited feedback mechanisms, whereas the Firebase-powered architecture provided nearly instant synchronization and clear visual notifications directly on the user's device. This modern approach greatly improved responsiveness, user awareness, and overall security management effectiveness.

D. Operational Improvements

The complete digital implementation of the security management system greatly reduced dependence on traditional localized alarm panels. Instead of relying on separate physical control units, all major operations such as device provisioning, system monitoring, historical logging, and remote alarm control were managed through a centralized Progressive Web Application (PWA). This created a more flexible and user-friendly security management experience for property owners.

The centralized platform enabled users to remotely arm or disarm the system, monitor device activity, and review detailed historical security records from any location with internet access. Since all information was securely stored and synchronized digitally, users could easily access important data without being physically present near the hardware setup. This remote accessibility improved convenience, monitoring efficiency, and overall system control.

Another major advantage of the digital implementation was improved data protection and organization. Unlike traditional systems that depend heavily on local storage devices, the cloud-based architecture minimized the risk of losing important records due to hardware failure or intentional tampering by intruders. By maintaining secure online records, the system ensured better reliability, easier data management, and enhanced long-term security monitoring capabilities.

E. User Experience and System Reliability

The user interface of the Quicky platform was designed using a modern glass morphism aesthetic to create a clean, professional, and trustworthy user experience. The interface focused on simplicity and visual clarity, allowing users to easily navigate through different features of the security system. The modern design approach also improved the overall appearance of the platform, making the application feel more engaging and user-friendly.

During beta testing, users were able to access and understand important security information without facing major technical difficulties. Features such as live system monitoring, alert notifications, and historical activity logs were organized in an intuitive manner to ensure smooth interaction for both technical and non-technical users. This ease of use helped improve user confidence and overall accessibility of the platform.

The Progressive Web App (PWA) architecture further enhanced the user experience by allowing the dashboard to be installed directly on mobile devices, similar to a native application. Users could quickly access the platform from their home screens without requiring a traditional app store installation. Final testing results showed that Quicky delivers a reliable, scalable, and highly user-friendly smart security solution suitable for modern home monitoring and anti-theft applications.

VII. FUTURE ENHANCEMENTS

The Quicky platform provides an incredibly strong, technologically advanced foundation for developing even more sophisticated and intelligent anti-theft systems in the future. Although the current implementation successfully and flawlessly supports real-time pressure tracking and remote cloud monitoring, several highly advanced additional features can be integrated to further skyrocket system efficiency,

scalability, and user experience.

A. AI-Based Footprint Recognition

Advanced Artificial Intelligence and Deep Machine Learning techniques can be directly integrated into the Quicky hardware processing pipeline to deeply analyze the complex analog waveform of the physical pressure sensors. By meticulously analyzing weight distribution, walking cadence, and unique footstep patterns, the system could confidently differentiate between an authorized resident, a large pet, or an entirely unknown intruder, thereby further reducing false alarms to absolute zero.

B. Multi-Sensor Ecosystem Integration

Currently, Quicky relies masterfully on physical flooring sensors. In future development, the platform could be massively expanded to seamlessly integrate with existing smart home ecosystems (like Google Home, Amazon Alexa, or Apple HomeKit) and synchronize directly with local IP security cameras. If the floor detects a footstep in a specific zone, the system could automatically pivot a PTZ camera to that exact location in the room, capturing the intruder instantly.

C. Cloud Deployment and Massive Scalability

Future implementations can include migrating from Firebase to highly custom AWS or Google Cloud architectures utilizing Kubernetes to support massive-scale enterprise deployments. This would allow the system to effortlessly manage thousands of physical floor sensors across massive commercial warehouses, server farms, or large public museums without a single drop in performance.

VIII. CONCLUSION

The Quicky system is a smart and centralized real-time IoT security platform developed to improve modern anti-theft monitoring and security management efficiency. The platform combines hardware-based pressure sensing with cloud connectivity to provide continuous monitoring and rapid response during suspicious activities. Its architecture was designed to deliver reliable performance while maintaining a simple and user-friendly experience for property owners.

The system integrates multiple advanced features into a single platform, including live flooring pressure tracking, instant web-based alert notifications, secure historical activity logging, and remote hardware management. Using ESP32 hardware and Firebase real-time communication, the platform is capable of instantly synchronizing security data between physical devices and the centralized web dashboard. This allows administrators to monitor and control the security system remotely with minimal delay.

The brilliant implementation of ESP32 microcontrollers, deeply integrated physical Piezo sensors, and highly modern web technologies such as Vanilla JS, CSS3, and Firebase Realtime Database vastly improves security transparency and drastically cuts down response times. The system truly helps property owners monitor their most valuable premises far more efficiently from absolutely anywhere in the world. Overall, Quicky provides an exceptionally reliable, highly innovative, and refreshingly user-friendly solution for modern security management, with limitless future scope for AI footprint recognition and comprehensive smart-home ecosystem

integration.

All functionalities are managed through a centralized Progressive Web App (PWA), which enables users to access the platform directly from their mobile or desktop devices without requiring traditional software installation. The responsive and integrated design improves accessibility, simplifies security operations, and enhances overall system coordination. As a result, Quickly provides a reliable, scalable, and efficient smart security solution suitable for modern homes and anti-theft applications.

<https://firebase.google.com/docs/database> (accessed May 20, 2024).

[14] Arduino. "Programming ESP32 with Arduino IDE." Arduino.cc. <https://docs.arduino.cc/> (accessed May 20, 2024).

REFERENCES

[1] S. Sharma and R. Gupta, "Smart security management systems for residential institutions," *Int. J. Adv. Comp. Sci.*, vol. 12, no. 4, pp. 210–216, Apr. 2021.

[2] A. Kumar and P. Singh, "Real-time IoT tracking using ESP32 and web technologies," in *Proc. IEEE Int. Conf. Internet Things (iThings)*, Atlanta, GA, USA, 2020, pp. 145–150.

[3] M. Patel and K. Verma, "Communication challenges in traditional security systems," *Int. J. Eng. Res.*, vol. 10, no. 6, pp. 325–330, Jun. 2021.

[4] Espressif Systems. "ESP32 Wi-Fi & Bluetooth MCU." Espressif.com. <https://www.espressif.com/en/products/socs/esp32> (accessed May 20, 2024).

[5] T. Brown and S. Wilson, "Limitations of localized alarm systems in dynamic environments," *Int. J. Secur. Eng.*, vol. 8, no. 3, pp. 98–104, Aug. 2020.

[6] P. Sharma and A. Das, "Sensor-based home monitoring and tracking system," in *Proc. Int. Conf. Smart Comput. Electron. Syst.*, Chennai, India, 2021, pp. 412–417.

[7] R. Mehta and V. Shah, "Improving IoT communication using real-time database systems," *IEEE Access*, vol. 9, pp. 55672–55680, Apr. 2021.

[8] K. Rao and S. Mishra, "Digital log management for smart security systems," *Int. J. Comput. Appl.*, vol. 176, no. 21, pp. 15–20, Nov. 2020.

[9] D. Singh and P. Roy, "Emergency alert mechanisms in intelligent smart homes," *Int. J. Innov. Technol. Explor. Eng.*, vol. 9, no. 5, pp. 2271–2276, Mar. 2020.

[10] J. Lee and H. Kim, "Centralized monitoring platforms for IoT security," *IEEE Trans. Internet Things*, vol. 22, no. 9, pp. 5643–5652, Sep. 2021.

[11] K. Beck et al., "Manifesto for Agile Software Development." AgileAlliance.org. <https://agilemanifesto.org/> (accessed May 20, 2024).

[12] Mozilla Developer Network. "Progressive Web Apps (PWA)." MDN Web Docs. https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps (accessed May 20, 2024).

[13] Google Inc. "Firebase Realtime Database Documentation." firebase.google.com.