



**GOVERNMENT OF TAMILNADU**  
**DIRECTORATE OF TECHNICAL EDUCATION, CHENNAI**

**NAAN MUDHALVAN SCHEME (TNSDC) SPONSORED**  
**STUDENTS DEVELOPMENT PROGRAMME**

**ON**

**IoT AND ITS APPLICATIONS**

**HOST INSTITUTION**

**XXXX**

**COIMBATORE – 04**

**TRAINING PARTNER**

**ENTHU TECHNOLOGY SOLUTIONS INDIA PVT LTD**

**DATE:**

<b>NAME</b>	<b>ROLL NO</b>
Name 1	Roll no 1
Name 2	Roll no 2
Name 3	Roll no 3
Name 4	Roll no 4
Name 5	Roll no 5

## TABLE OF CONTENTS

S.NO	TITLE	PAGE NO
1	ABSTRACT	1
2	INTRODUCTION	2
3	HARDWARE AND SOFTWARE REQUIREMENTS	3
4	BLOCK DIAGRAM	10
5	CODE	11
6	OUTPUT RESULTS	16
7	CLOUD OUTPUT	17
8	CONCLUSION	18

## **ABSTRACT**

The increasing demand for enhanced security in both residential and commercial environments has highlighted the limitations of traditional mechanical locks, which are often vulnerable to tampering and unauthorized access. To overcome these challenges, the RFID (Radio Frequency Identification) door lock system offers a modern, digital alternative that enhances security and convenience. Unlike conventional locks that require physical keys, the RFID system utilizes RFID technology to grant access through an RFID tag recognized by a reader connected to a microcontroller, such as the ESP32. When a valid RFID tag is detected, the system activates a servo motor to unlock the door, with a visual indicator confirming the action. This keyless entry system not only improves security by reducing the risk of unauthorized access but also integrates seamlessly with other digital technologies, making it an ideal solution for modern access control needs.

## INTRODUCTION

In today's world, where security is a paramount concern, traditional mechanical locks are increasingly viewed as inadequate due to their vulnerability to tampering and unauthorized access. To address these shortcomings, this project introduces an RFID (Radio Frequency Identification) door lock system, a modern solution designed to enhance security while offering the convenience of keyless entry. Leveraging the capabilities of RFID technology, this system replaces physical keys with RFID tags that are recognized by a microcontroller, such as the ESP32, to control access to secured areas. Upon presenting a valid RFID tag, the system unlocks the door via a servo motor, providing a secure and user-friendly alternative to conventional locks. This project aims to demonstrate how digital technology can be effectively applied to create a robust, reliable, and easily manageable security system suitable for both residential and commercial use.

# **HARDWARE AND SOFTWARE REQUIREMENTS**

## **HARDWARE REQUIREMENT**

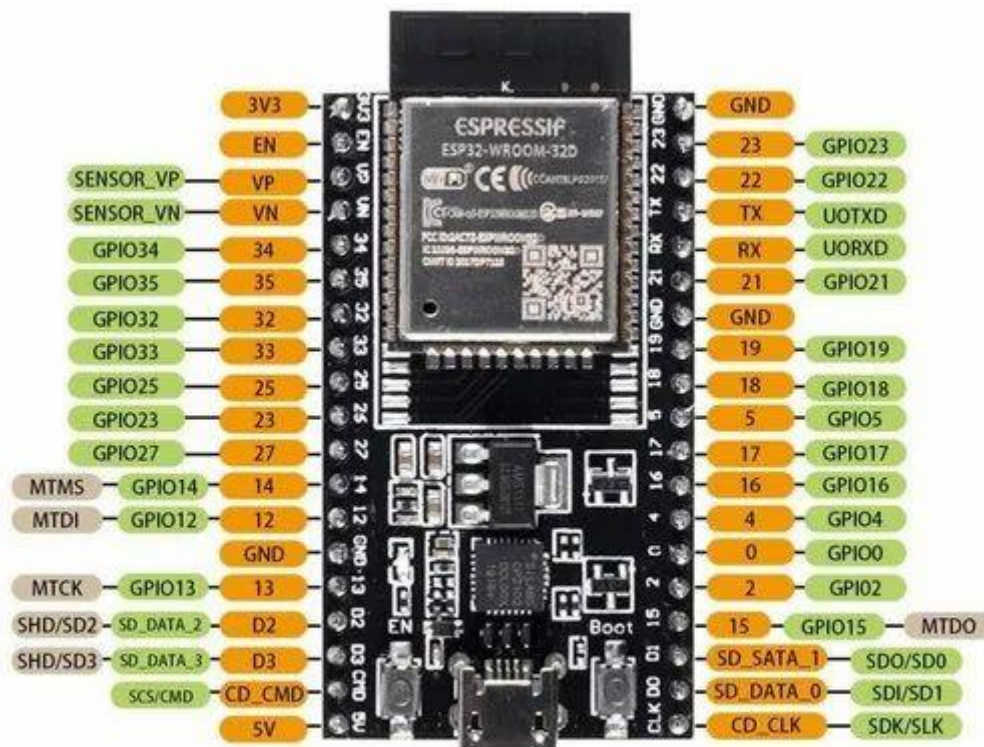
- ESP32 Microcontroller
- RFID MFRC522
- Servo Motor
- LED
- BUZZER
- Jumper Wires
- USB Cable

## **SOFTWARE REQUIREMENT**

- Arduino IDE
- ThingzMate Cloud

## ESP32 Microcontroller

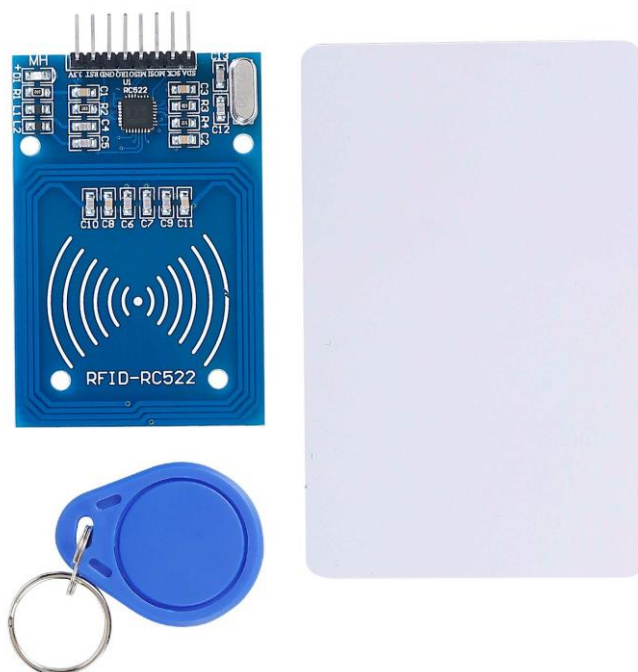
The ESP32 is a highly versatile microcontroller developed by Espressif Systems, specifically designed to cater to a wide range of applications, particularly within the Internet of Things (IoT) domain. It is celebrated for its blend of high performance, integrated wireless connectivity, and a comprehensive feature set—all offered at an affordable price point. The ESP32 is widely utilized in projects requiring both Wi-Fi and Bluetooth functionalities, making it an ideal choice for smart home devices, sensor networks, and wearable technology. Its adaptability and powerful capabilities position it as a go-to solution for developers looking to create innovative, connected devices.



**ESP32-WROOM-32D**

## RFID MFRC522

The RFID MFRC522 is a widely used RFID reader/writer module designed for short-range communication with RFID tags. It operates at a frequency of 13.56 MHz and is based on the NXP MFRC522 integrated circuit. This module is commonly used in various applications, including access control systems, identification, and asset tracking. The MFRC522 module can read and write data to compatible RFID tags, such as MIFARE cards, and is known for its ease of use and integration with microcontrollers like the Arduino and ESP32. It communicates with the microcontroller using standard interfaces like SPI (Serial Peripheral Interface), which simplifies the connection and data exchange processes.



## LED

An LED (Light Emitting Diode) is a semiconductor light source that emits light when an electric current flows through it. Known for its energy efficiency and longevity, the LED is one of the most advanced lighting technologies available today. Due to its small size, low power consumption, and durability, LEDs are extensively used in a wide range of applications, from simple status indicators to complex display systems.

In the context of an anti-theft security system, the LED functions as a visual alert, signaling the detection of motion. When the PIR sensor detects movement, the ESP32 microcontroller sends a signal to the LED, causing it to illuminate. This visual cue serves as an immediate indication of potential unauthorized access or an intruder in the monitored area. The LED's rapid response and high visibility make it an effective component in security systems, providing a clear and instant indication of a security breach, which can be crucial for prompt action.

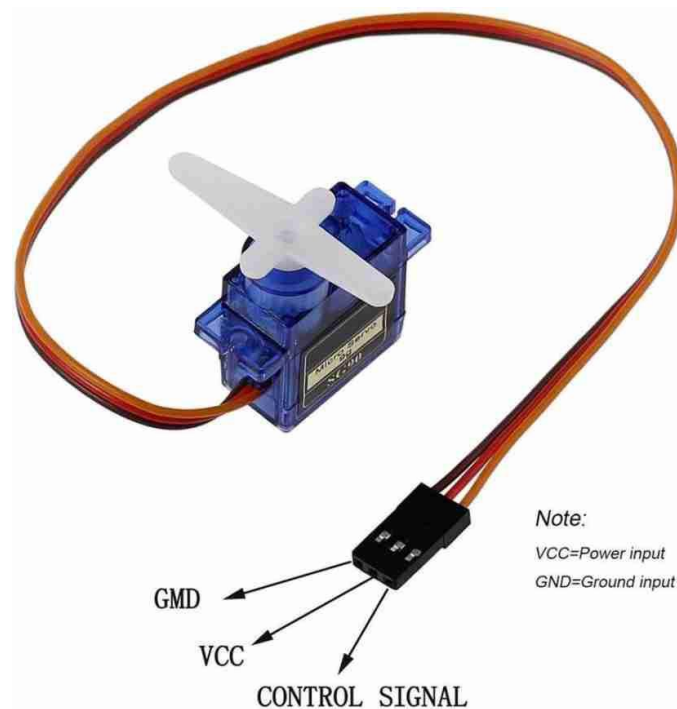




## Servo Motor

A servo motor is a type of rotary actuator that provides precise control of angular position, velocity, and acceleration. It consists of a motor coupled with a feedback device, such as a potentiometer or an encoder, which allows for accurate control of its position. Servo motors are widely used in various applications requiring high precision and reliability, including robotics, automation systems, and control mechanisms.

In a servo motor, the control signal, often a PWM (Pulse Width Modulation) signal, dictates the motor's position. The motor then adjusts its position based on this signal until it aligns with the desired position as indicated by the feedback device. This feedback loop ensures precise and stable control, making servo motors ideal for applications where accurate positioning is critical.



## **BUZZER**

A buzzer is an electromechanical or electronic device that produces an audible sound, often used as a signaling or alerting mechanism. Buzzers work by converting electrical energy into sound through the rapid movement of a diaphragm or a piezoelectric element. They are commonly employed in a wide range of applications, including alarms, timers, and electronic devices, where they serve as a means of providing notifications or warnings.

In addition to these uses, buzzers are also found in interactive systems, such as game shows, where they signal a participant's response or indicate a turn. Their ability to deliver a clear, attention-grabbing sound makes them an essential component in systems that require immediate auditory feedback.



## JUMPER WIRE

Jumper wires are essential in this project, used to connect the ESP32 microcontroller to the components on the breadboard. These wires provide a flexible and reliable way to link the microcontroller's GPIO pins to the LEDs, resistors, and other circuit elements, enabling proper signal and power flow. Their ease of use allows for quick modifications and testing during the prototyping stage.



## USB Cable

The USB cable is a critical tool in this project, used to connect the ESP32 microcontroller to a computer for power supply, programming, and debugging. It enables the transfer of code and data between the development environment and the microcontroller, facilitating the upload of firmware and real-time communication during the development process. The USB connection also allows for serial monitoring, providing valuable insights into the system's performance and behavior.



## **Wokwi Simulator**

To simulate this anti-theft security system using ESP32, PIR sensor, LED, and buzzer on Wokwi, you can create a virtual environment to test the circuit and code. Wokwi allows you to connect these components digitally, providing a platform to visualize the system's operation in response to simulated motion detection. This helps in verifying the functionality before implementing it in a real-world scenario.

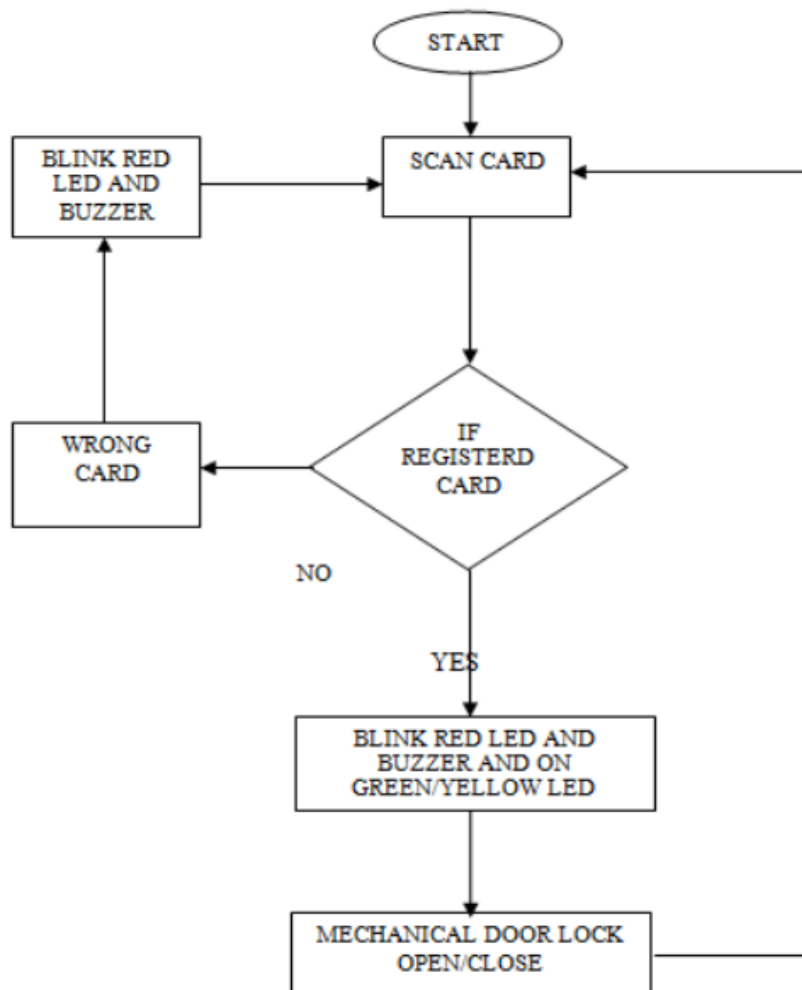
## **Arduino IDE**

To develop this anti-theft security system, you can use the Arduino IDE to write and upload code to the ESP32. The IDE provides a user-friendly environment for coding, debugging, and interfacing with the ESP32, making it easier to control the PIR sensor, LED, and buzzer. With Arduino IDE, you can quickly implement and test the security system's functionality.

## **ThingzMate Cloud**

Integrating the anti-theft security system with Thingzmate Cloud allows you to monitor and control the ESP32 remotely. By connecting the ESP32 to Thingzmate, you can receive real-time alerts and view the status of the PIR sensor, LED, and buzzer from anywhere. This cloud integration enhances the system by providing remote access and data logging capabilities.

## BLOCK DIAGRAM



## CODE

```
#include <SPI.h>

#include <MFRC522.h>

#include <WiFi.h>

#include <HTTPClient.h>

#include <Adafruit_Sensor.h>

#include <DHT.h>


// Pin definitions

#define SS_PIN 5 // ESP32 pin GPIO5 for RFID reader

#define RST_PIN 22 // ESP32 pin GPIO22 for RFID reader

#define LED_PIN 2 // ESP32 pin GPIO2 for LED

#define DHT_PIN 4 // ESP32 pin GPIO4 for DHT11 sensor

#define MQ9_PIN 34 // ESP32 pin GPIO34 for MQ9 sensor (Analog pin)


MFRC522 rfid(SS_PIN, RST_PIN); // Create MFRC522 instance

DHT dht(DHT_PIN, DHT11); // Create DHT instance


// WiFi credentials

const char* ssid = "iphone13"; // Replace with your WiFi SSID

const char* password = "password123"; // Replace with your WiFi password


// ThingzMate settings
```

```
const String serverName = " https://console.thingzmate.com/api/v1/device-  
types/esp321/devices/field1/uplink"; // Replace with your ThingzMate endpoint URL  
  
const String AuthorizationToken = " Bearer 0613a1c65d95057d975a85f8faaa529b"; //  
Replace with your ThingzMate Authorization Token
```

```
// Define UIDs for access control
```

```
byte uidOn[] = {0x93, 0xEC, 0xD9, 0x13}; // UID for turning LED on
```

```
byte uidOff[] = {0x73, 0x18, 0x20, 0x2A}; // UID for turning LED off
```

```
void setup() {
```

```
    Serial.begin(115200);    // Initialize Serial Monitor
```

```
    SPI.begin();            // Initialize SPI bus
```

```
    rfid.PCD_Init();        // Initialize MFRC522 RFID reader
```

```
    pinMode(LED_PIN, OUTPUT); // Set LED_PIN as an output
```

```
    digitalWrite(LED_PIN, LOW); // Ensure the LED is off initially
```

```
    dht.begin();            // Initialize DHT11 sensor
```

```
// Connect to WiFi
```

```
WiFi.begin(ssid, password);
```

```
Serial.println();
```

```
Serial.print("Connecting to ");
```

```
Serial.println(ssid);
```

```
// Wait until connected to WiFi

while (WiFi.status() != WL_CONNECTED) {

    delay(500);

    Serial.print(".");

}


Serial.println("Connected to WiFi");

Serial.print("IP Address: ");

Serial.println(WiFi.localIP());

}


void loop() {

    // Monitor temperature, humidity, and gas levels

    float temperature = dht.readTemperature();

    float humidity = dht.readHumidity();

    int gasLevel = analogRead(MQ9_PIN);


    // Check if readings are valid

    if (isnan(temperature) || isnan(humidity)) {

        Serial.println("Failed to read from DHT sensor!");

    } else {

        Serial.print("Temperature: ");
```



```
Serial.print(temperature);

Serial.println(" °C");


Serial.print("Humidity: ");

Serial.print(humidity);

Serial.println(" %");

}


Serial.print("MQ9 Gas Level: ");

Serial.println(gasLevel);


if (rfid.PICC_IsNewCardPresent()) { // Check if a new card is present

    if (rfid.PICC_ReadCardSerial()) { // Read the card's UID

        boolean accessGranted = false;

        String statusMessage;


        // Check UID and perform corresponding action

        if (compareUID(rfid.uid.uidByte, uidOn)) {

            accessGranted = true;

            Serial.println("Light on");

            digitalWrite(LED_PIN, HIGH); // Turn on the LED

            statusMessage = "Light on";

        }

    }

}
```

```
else if (compareUID(rfid.uid.uidByte, uidOff)) {  
    accessGranted = true;  
  
    Serial.println("Light off");  
  
    digitalWrite(LED_PIN, LOW); // Turn off the LED  
  
    statusMessage = "Light off";  
  
}  
  
else {  
  
    Serial.print("Unknown UID: ");  
  
    for (int i = 0; i < rfid.uid.size; i++) {  
  
        Serial.print(rfid.uid.uidByte[i] < 0x10 ? "0" : "");  
  
        Serial.print(rfid.uid.uidByte[i], HEX);  
  
    }  
  
    Serial.println();  
  
    statusMessage = "Unknown UID";  
  
}  
  
  
// Send data to ThingzMate  
  
sendDataToThingzMate(statusMessage);  
  
  
rfid.PICC_HaltA(); // Halt PICC  
  
rfid.PCD_StopCrypto1(); // Stop encryption on PCD  
  
}  
  
}
```

```
    delay(2000); // Wait for 2 seconds before the next loop iteration
}

void sendDataToThingzMate(String message) {
    if (WiFi.status() == WL_CONNECTED) {

        HTTPClient http;

        http.begin(serverName); // Specify the URL

        http.addHeader("Content-Type", "application/json"); // Add content-type header
        http.addHeader("Authorization", AuthorizationToken); // Add authorization header

        // Create JSON payload

        String payload = "{\"message\":\"" + message + "\"}";

        int httpResponseCode = http.POST(payload); // Send POST request

        if (httpResponseCode > 0) {

            String responsePayload = http.getString();

            Serial.println(httpResponseCode);

            Serial.println(responsePayload);

        } else {

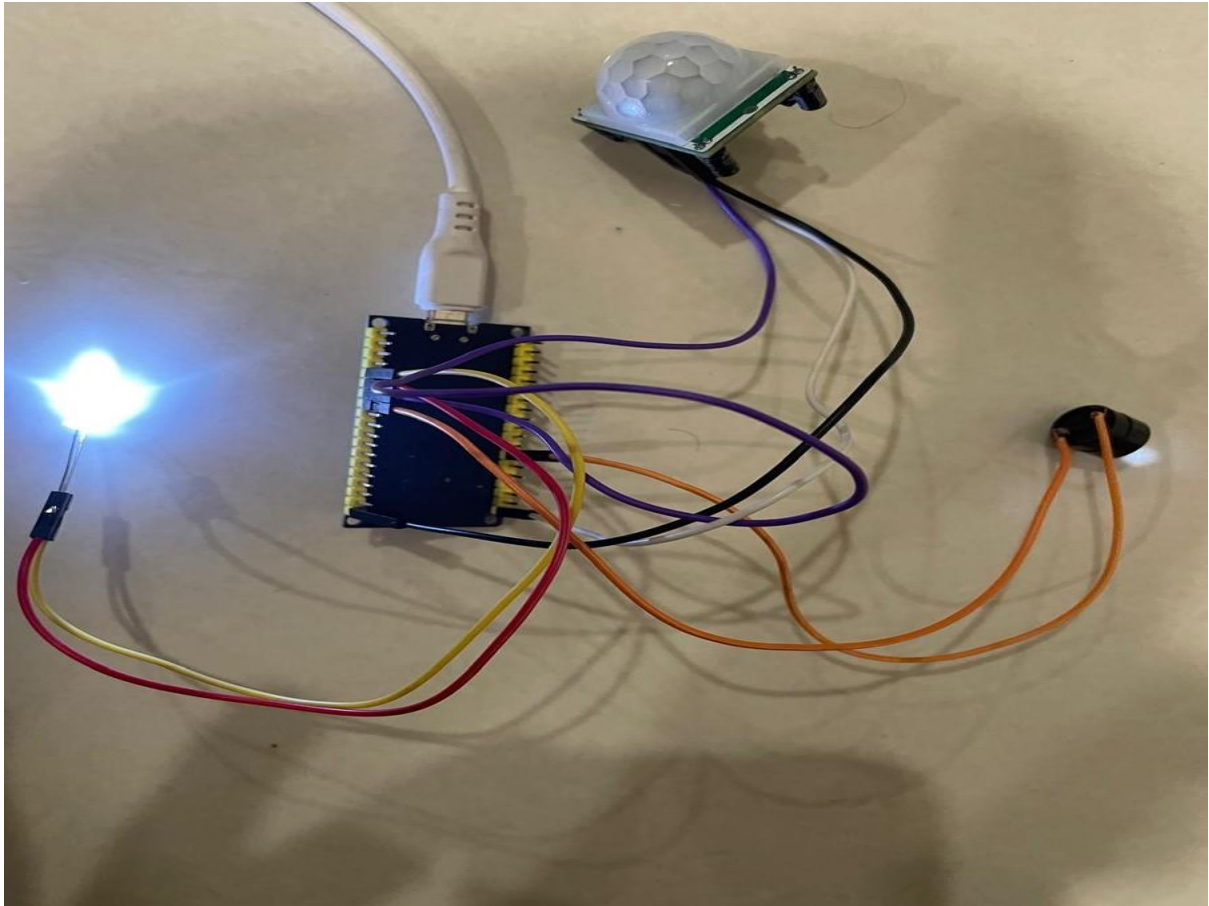
            Serial.print("Error code: ");
```

```
        Serial.println(httpResponseCode);
    }

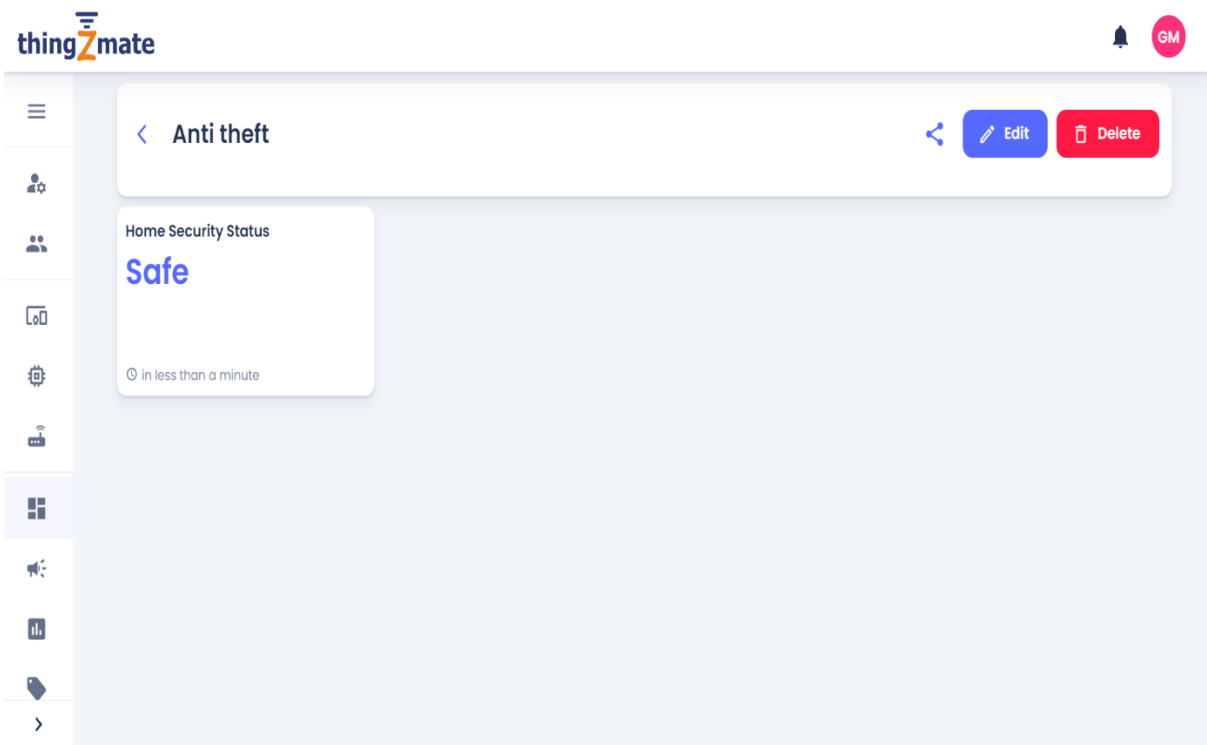
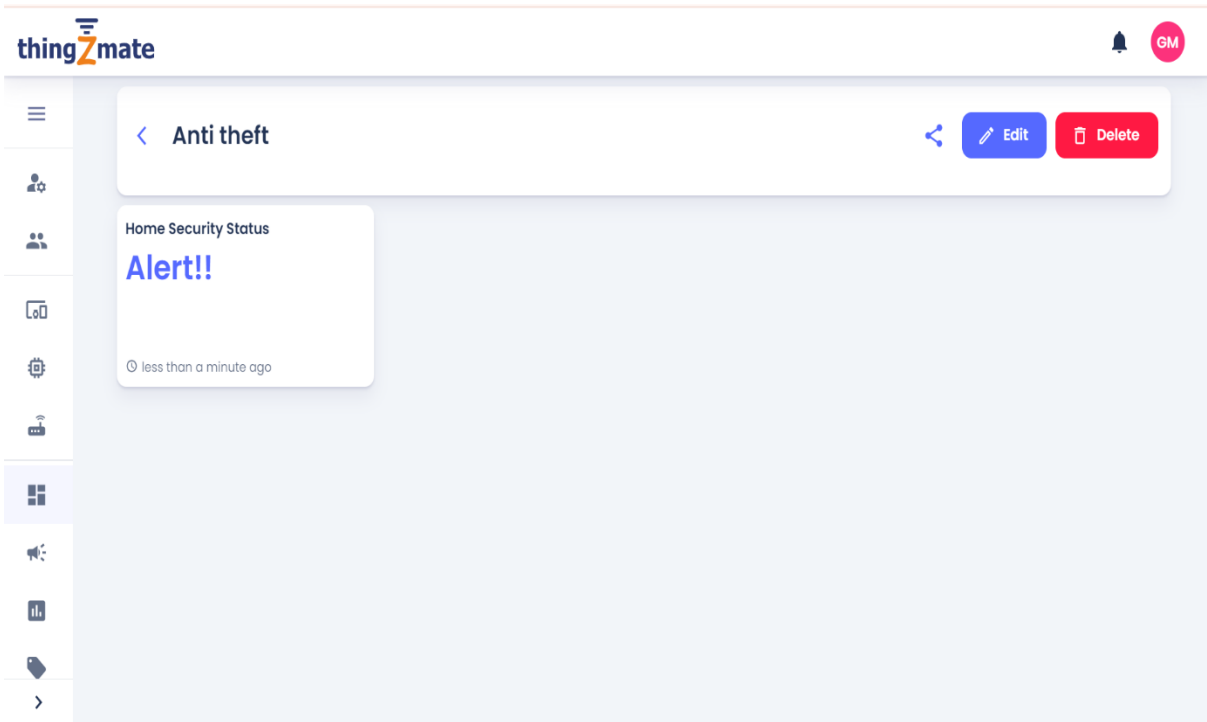
    http.end(); // Free resources
} else {
    Serial.println("Error: Not connected to WiFi");
}
}

// Function to compare the UID read from the RFID reader with a valid UID
boolean compareUID(byte *uid1, byte *uid2) {
    for (int i = 0; i < 4; i++) {
        if (uid1[i] != uid2[i]) {
            return false;
        }
    }
    return true;
}
```

## OUTPUT RESULTS



# CLOUD OUTPUT



## **CONCLUSION**

The development of an anti-theft security system using an ESP32, PIR sensor, LED, and buzzer offers a simple yet effective solution for enhancing security in various environments. This project demonstrates how readily available electronic components can be integrated to create a responsive system that detects motion and immediately triggers visual and audible alarms. The ESP32's versatility, combined with the reliability of the PIR sensor, ensures that any unauthorized movement is quickly detected and reported. This system provides a cost-effective, scalable, and efficient means of securing property, making it a valuable addition to homes, offices, and small businesses.