**Mini Project**

**Tittle: "IoT-Based Smart Parking Slot Monitoring System"**

**Abstract**

This project implements an IoT-based parking slot monitoring system using an ultrasonic sensor integrated with an ESP32 microcontroller. It measures the distance to detect the availability of parking slots. The system transmits real-time slot status (vacant or occupied) and distance measurements to a cloud server (ThinkzMate). This cloud-based monitoring makes parking management more efficient and accessible.

**Software Requirements**

1. Arduino IDE (for coding and uploading firmware).
2. ThinkzMate Cloud Platform (for data visualization and storage).
3. Serial Monitor (for debugging).
4. Libraries:
   * WiFi.h (for connecting to WiFi).
   * HTTPClient.h (for HTTP POST requests).

**Hardware Requirements**

1. ESP32 Microcontroller.
2. Ultrasonic Sensor (HC-SR04).
3. Jumper Wires.
4. Breadboard.
5. 5V Power Supply or USB cable for ESP32.

**Connection Details**

| **Component** | **ESP32 Pin** | **Description** |
| --- | --- | --- |
| Ultrasonic Sensor - Trig Pin | GPIO 5 | Trigger for ultrasonic waves. |
| Ultrasonic Sensor - Echo Pin | GPIO 18 | Receives reflected waves. |
| VCC (Ultrasonic Sensor) | 3.3V or 5V | Power supply to the sensor. |
| GND | GND | Common ground. |

**Program**

The complete program is provided below:

#include <WiFi.h>

#include <HTTPClient.h>

#define WIFI\_SSID "Bhavankumar"

#define WIFI\_PASSWORD "12345689"

int trigPin = 5;

int echoPin = 18;

//define sound speed in cm/uS

#define SOUND\_SPEED 0.034

#define CM\_TO\_INCH 0.393701

long duration;

float distanceCm;

float distanceInch;

bool slotstatus;

const char \*serverUrl = "https://console.thingzmate.com/api/v1/device-types/bhavan-iot/devices/bhavan-iot/uplink"; // Replace with your server endpoint

String AuthorizationToken = "Bearer 12cdc3fbc1f0aa5ab911b836496071ea";

void setup() {

  Serial.begin(115200);

  delay(4000);

   // Starts the serial communication

  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output

  pinMode(echoPin, INPUT); // Sets the echoPin as an Input

  pinMode(22,OUTPUT);

   WiFi.begin(WIFI\_SSID, WIFI\_PASSWORD);

  Serial.print("Connecting to WiFi");

  while (WiFi.status() != WL\_CONNECTED) {

    delay(1000);

    Serial.print(".");

  }

  Serial.println("Connected to WiFi");

}

void loop() {

   digitalWrite(trigPin, LOW);

  delayMicroseconds(2);

  // Sets the trigPin on HIGH state for 10 micro seconds

  digitalWrite(trigPin, HIGH);

  delayMicroseconds(10);

  digitalWrite(trigPin, LOW);

  // Reads the echoPin, returns the sound wave travel time in microseconds

  duration = pulseIn(echoPin, HIGH);

  // Calculate the distance

  distanceCm = duration \* SOUND\_SPEED/2;

  // Convert to inches

  distanceInch = distanceCm \* CM\_TO\_INCH;

  // Prints the distance in the Serial Monitor

  Serial.print("Distance (cm): ");

  Serial.println(distanceCm);

 if(distanceCm<30)

 {

  slotstatus=0;

  Serial.println("VACANT");

   digitalWrite(22,LOW);

 }

 else

 {

  slotstatus=1;

  Serial.println("OCCUPIED");

  digitalWrite(22,HIGH);

 }

  HTTPClient http;

  http.begin(serverUrl);

  http.addHeader("Content-Type", "application/json");

http.addHeader("Authorization", AuthorizationToken); //Authorization token

  // Create JSON payload

  String payload = "{\"DistanceCm\":" + String(distanceCm) +",\"SLOTSTATUS\":" + String(slotstatus) + "}";

  // Send POST request

  int httpResponseCode = http.POST(payload);

  if (httpResponseCode > 0) {

    String response = http.getString();

    Serial.println("HTTP Response code: " + String(httpResponseCode));

    Serial.println(response);

  } else {

    Serial.print("Error code: ");

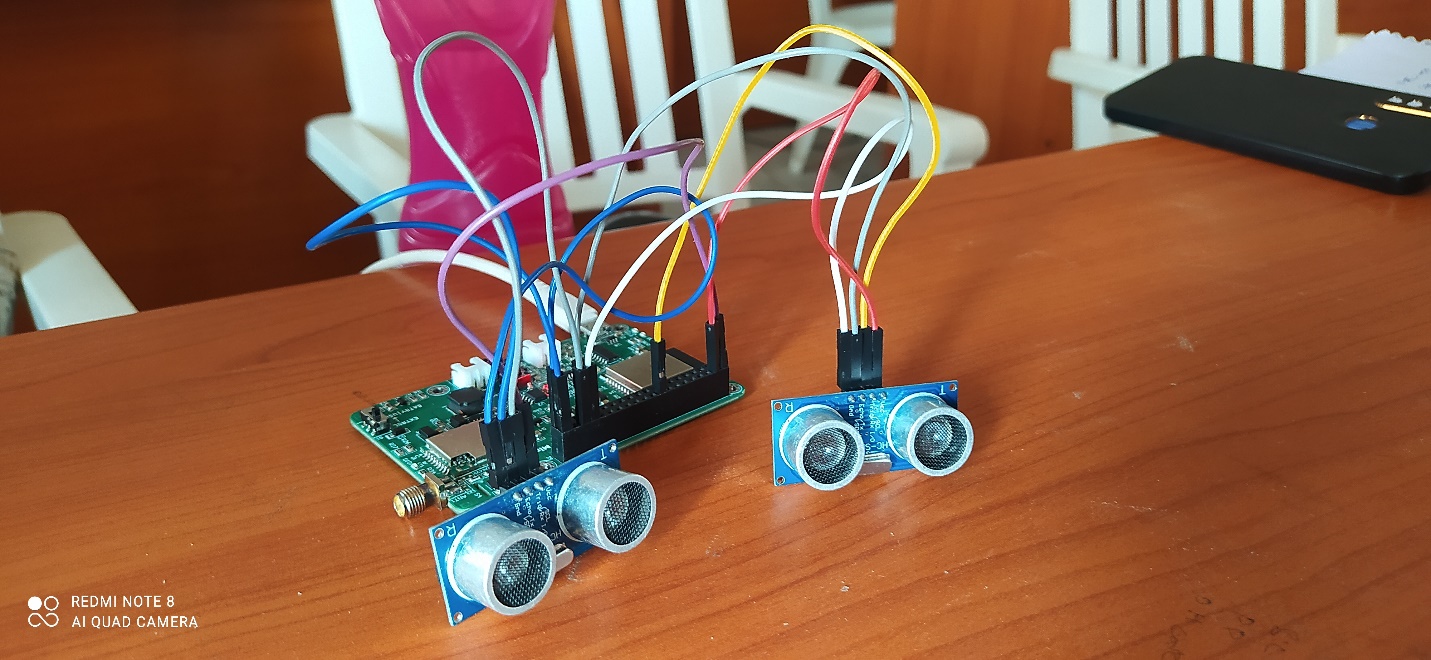
    Serial.println(httpResponseCode);

  }

  http.end(); // Free resources

  delay(1000); // Wait for 1 minute before sending next request

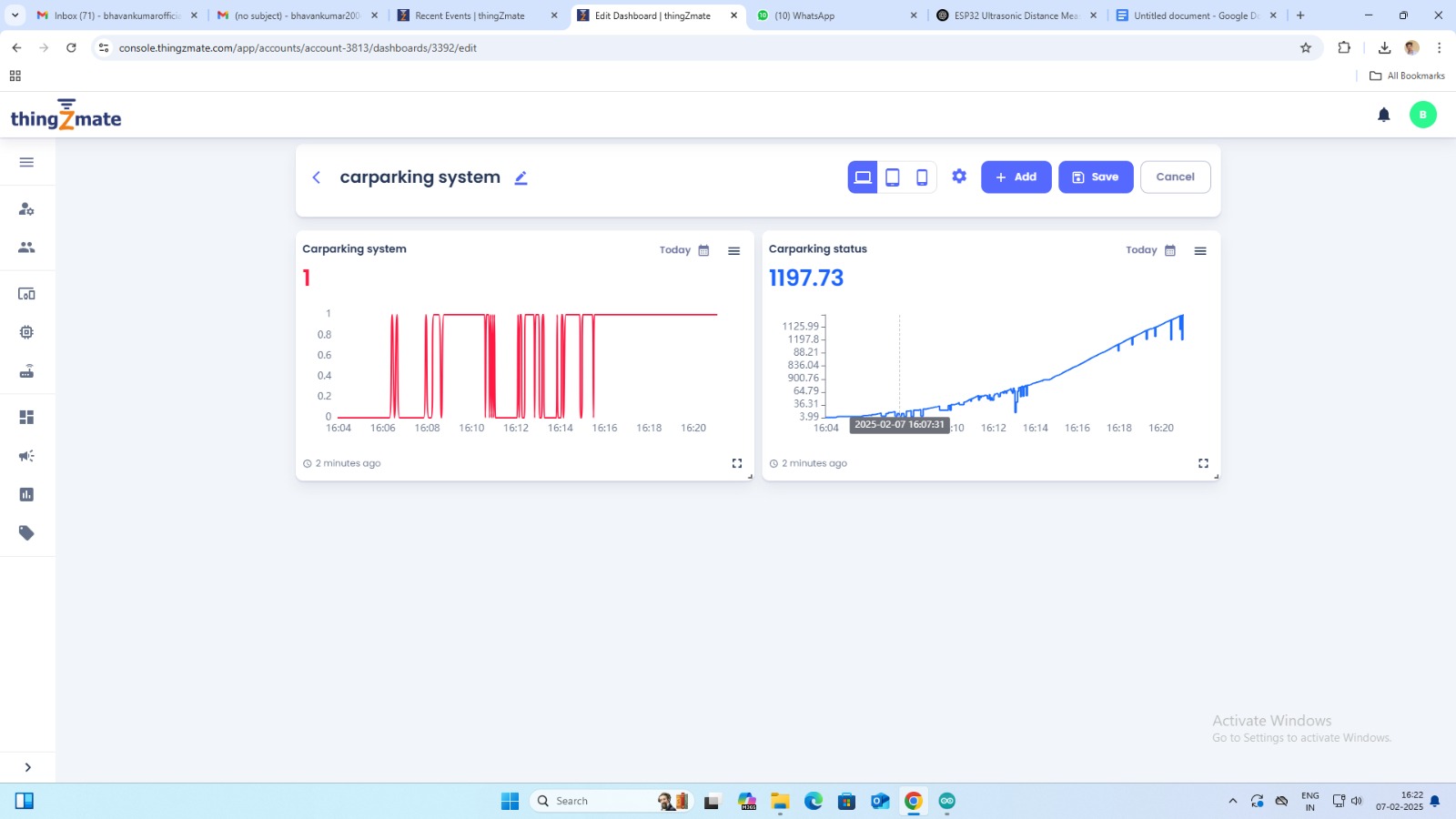
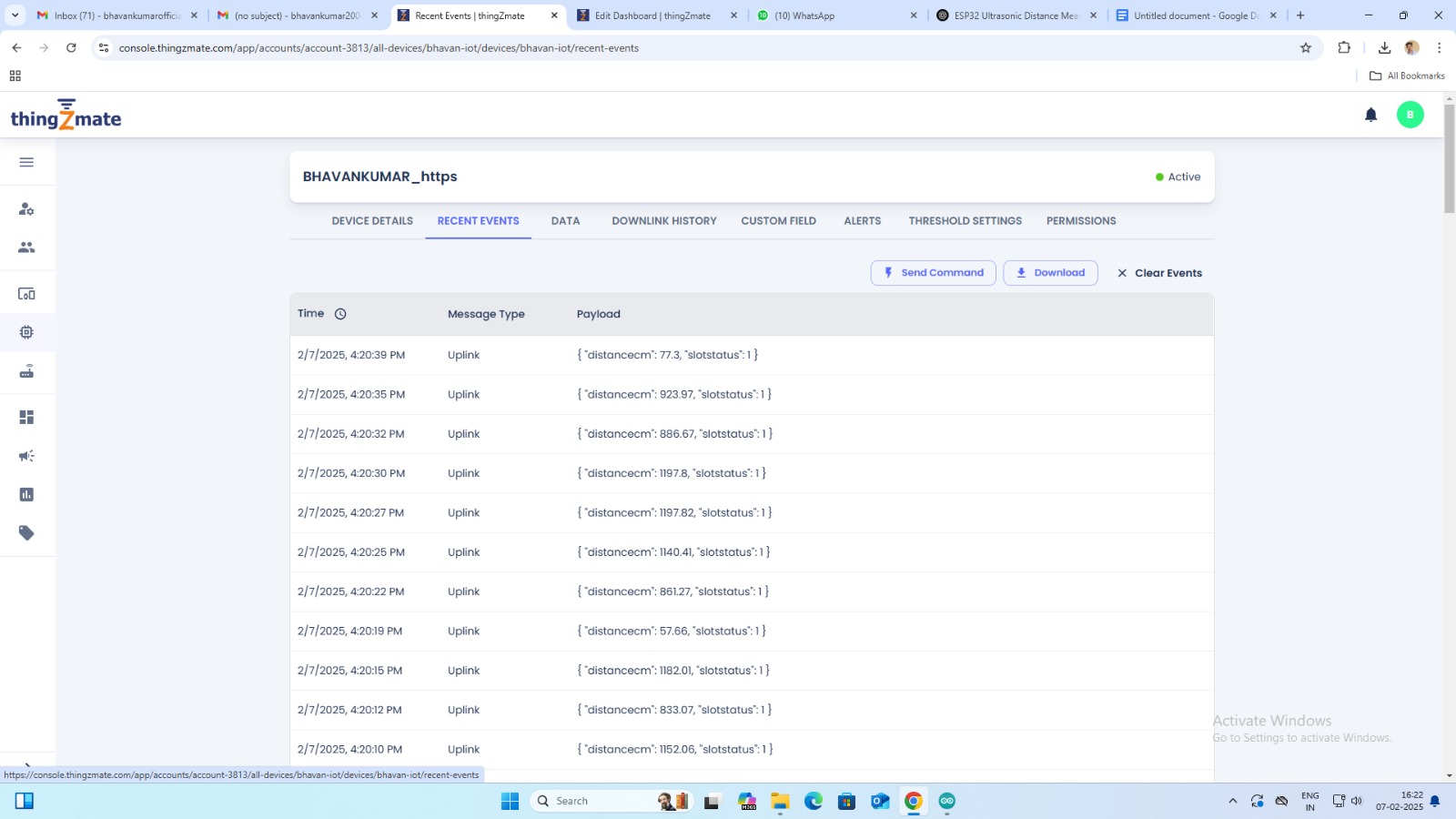
}



**Hardware Connection Output**

1. **Slot VACANT**:
   * Ultrasonic sensor detects a distance less than 30 cm.
   * Serial Monitor logs "VACANT."
2. **Slot OCCUPIED**:
   * Ultrasonic sensor detects a distance greater than or equal to 30 cm.
   * Serial Monitor logs "OCCUPIED."

**Cloud Output**



**Conclusion**

This project successfully demonstrates an IoT-based system for monitoring parking slot availability. By using an ultrasonic sensor and ESP32 microcontroller, the system efficiently transmits real-time data to a cloud server. The absence of LEDs simplifies the design and reduces power consumption, while cloud integration enables remote access to parking slot status.

**TEAM MEMBERS**

S. BHAVAN KUMAR, IIIrd year IT, AIT.

S. DEEPIKA, IIIrd year IT, AIT.