UUM

Universiti Utara Malaysia

STTHK3113 SENSOR BASED SYSTEM

FINAL PROJECT ASSIGNMENT

TITLE: ROOM ENTRY LOGGER WITH ENVIRONMENT MONITORING

SYAKEER AHMAD BIN NAFEES AHMAD	**289918**

Project Title

ROOM ENTRY LOGGER WITH ENVIRONMENT MONITORING

Project Description

1. **Problem statement**

In educational institutions, laboratories, and shared working spaces, it is often challenging to track the number of people entering a room and monitor the environmental conditions simultaneously. Manual entry logs are prone to human error and inefficiency, while existing systems may lack real-time alerts and display features.

This system addresses the need for an automated, real-time room entry logging system integrated with environmental monitoring using smart sensors and IoT-based technology.

2. **Objective**

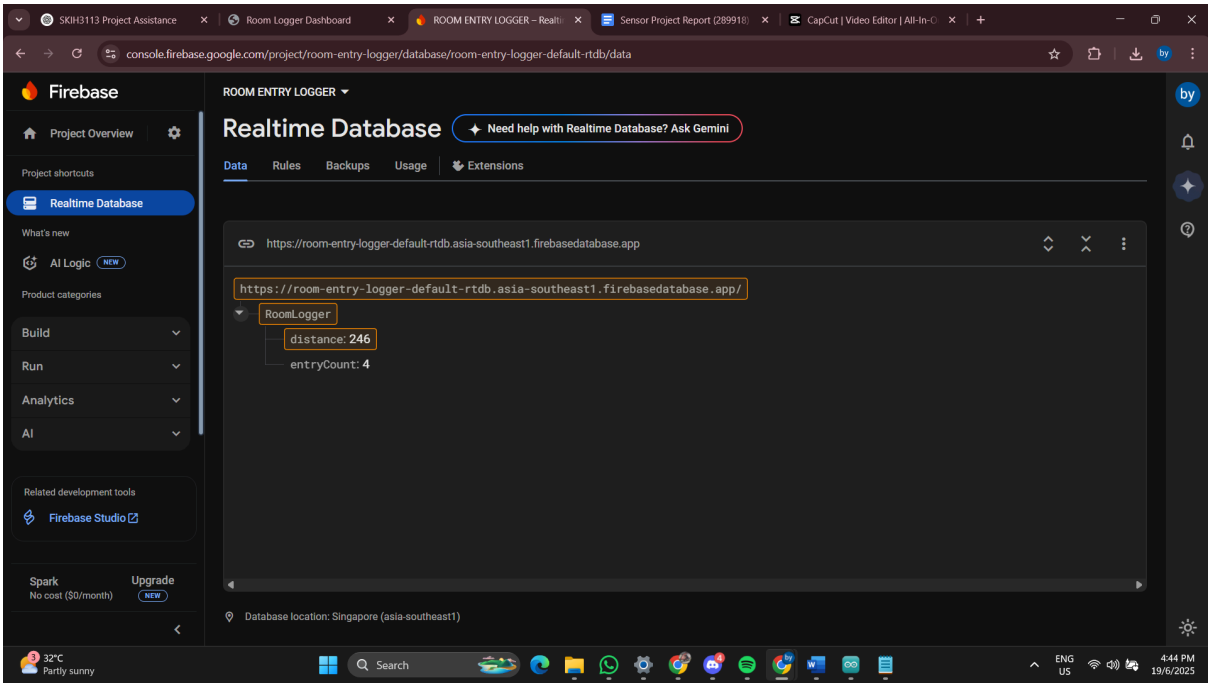
To develop a system that automatically logs room entries using a PIR motion sensor and measures the proximity of objects using an Ultrasonic sensor. The system will display real-time data on an OLED display and a web interface accessible through Wi-Fi. Additionally, a Relay Module will be triggered when proximity conditions are met, and entry count data will be monitored live.

3. **How the selected sensors solve the problem**

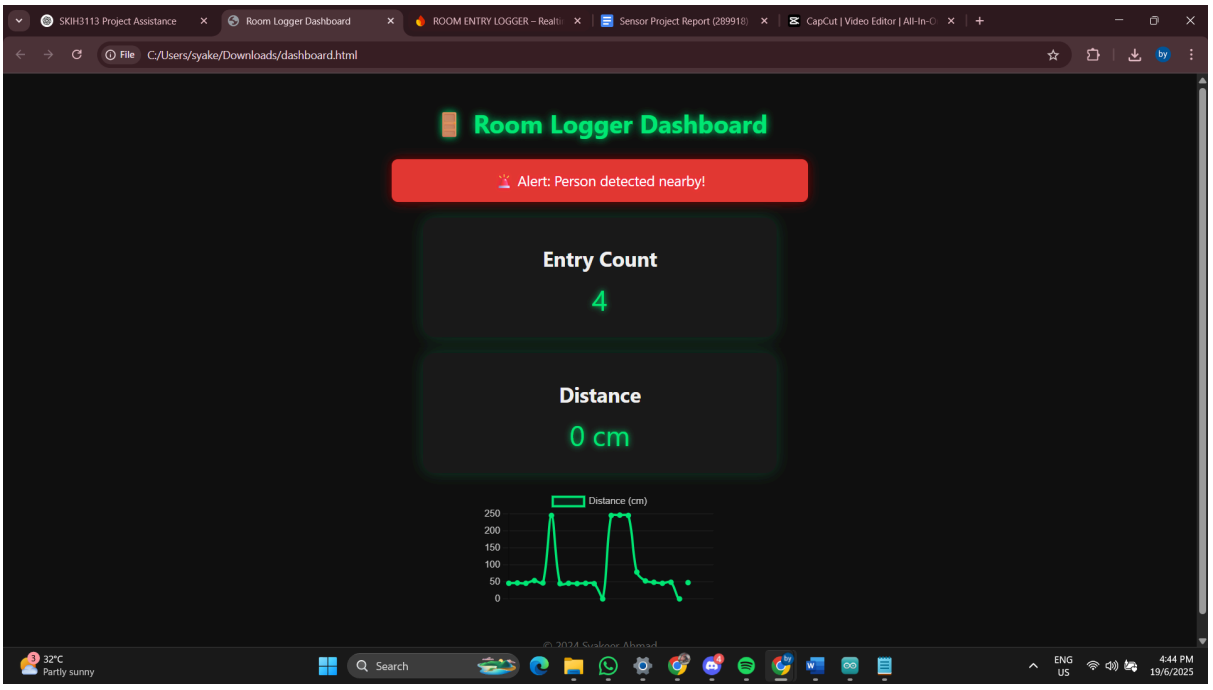
Sensor type	**Functionality**
PIR Sensor	Detects motion when a person enters the room, incrementing the entry count
Ultrasonic Sensor	Measures the distance of objects in front of it. If someone is detected too close, it triggers the relay for alerts.

These sensors enable the system to effectively track room entries and monitor object proximity without manual intervention.

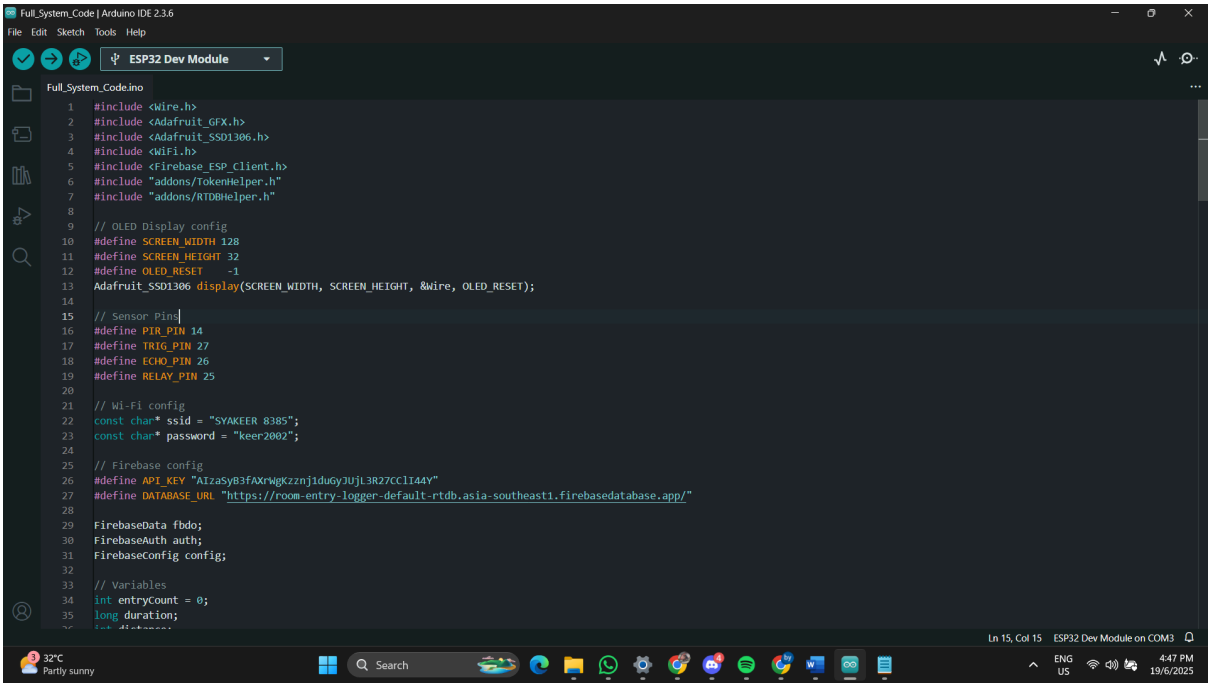
Database



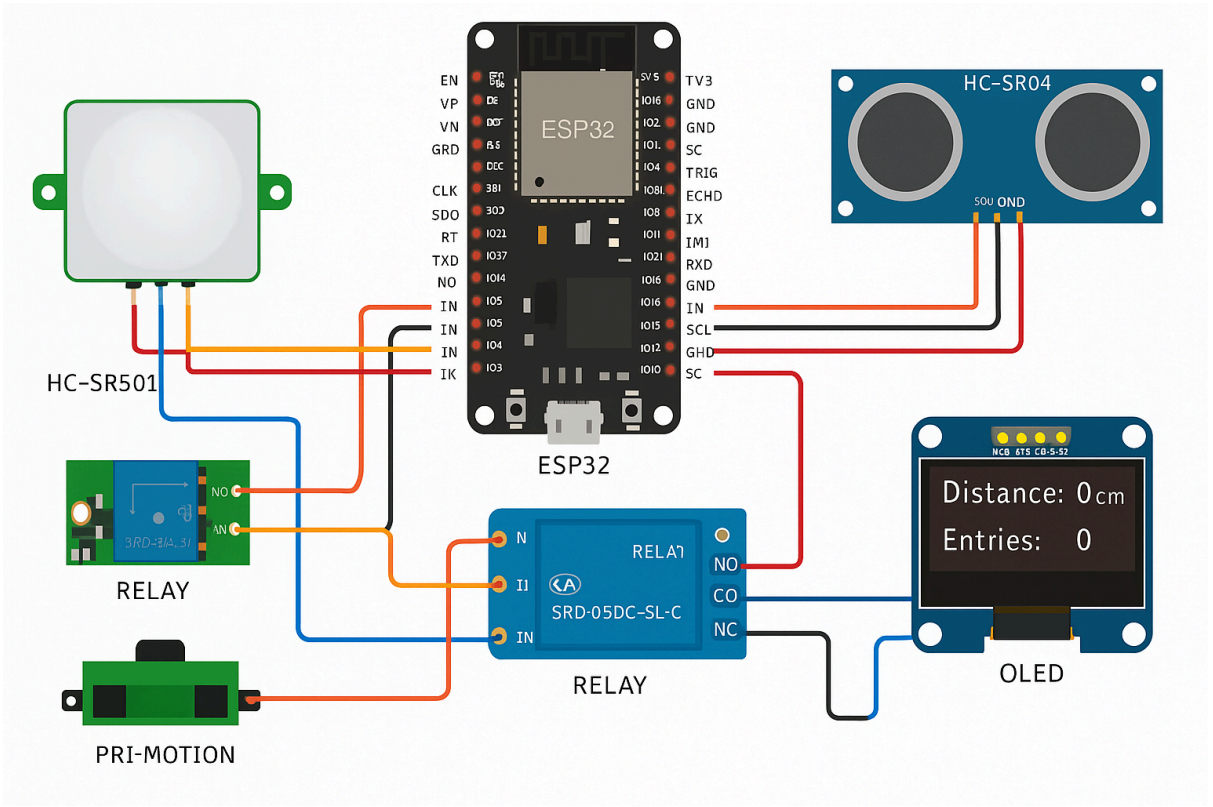
Web Interface



Arduino



Schematic



Arduino Code

```
#include <Wire.h>

#include <Adafruit_GFX.h>
```

```
#include <Adafruit_SSD1306.h>

#include <WiFi.h>

#include <Firebase_ESP_Client.h>

#include "addons/TokenHelper.h"

#include "addons/RTDBHelper.h"



// OLED Display config

#define SCREEN_WIDTH 128

#define SCREEN_HEIGHT 32

#define OLED_RESET    -1

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET);



// Sensor Pins

#define PIR_PIN 14

#define TRIG_PIN 27

#define ECHO_PIN 26

#define RELAY_PIN 25



// Wi-Fi config

const char* ssid = "SYAKEER 8385";

const char* password = "keer2002";



// Firebase config

#define API_KEY "AIzaSyB3fAXrWgKzznj1duGyJUjL3R27CClI44Y"
```

```
#define DATABASE_URL
"https://room-entry-logger-default-rtdb.asia-southeast1.firebasedatabas
e.app/"



FirebaseData fbdo;

FirebaseAuth auth;

FirebaseConfig config;



// Variables

int entryCount = 0;

long duration;

int distance;

int pirState = LOW;

int lastPirState = LOW;



// Server setup

WiFiServer server(80);



void displayMessage(String message, int count); // prototype



void setup() {

  Serial.begin(115200);

  pinMode(PIR_PIN, INPUT);

  pinMode(TRIG_PIN, OUTPUT);

  pinMode(ECHO_PIN, INPUT);

  pinMode(RELAY_PIN, OUTPUT);
```

```
digitalWrite(RELAY_PIN, LOW);


if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
  Serial.println(F("OLED allocation failed"));
  for (;;);
}
display.clearDisplay();
display.display();


WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("WiFi connected");
Serial.println(WiFi.localIP());


server.begin();


config.api_key = API_KEY;
config.database_url = DATABASE_URL;


auth.user.email = "";
auth.user.password = "";
```

```
  config.signer.tokens.legacy_token =
"0ddZaO4nF5EDwScvS7SEhbADsqwUwjdghpNbRqeD";



  Firebase.begin(&config, &auth);

  Firebase.reconnectWiFi(true);



  if (Firebase.ready()) {

    Serial.println("Firebase is ready.");

  } else {

    Serial.println("Firebase not ready!");

  }

}



void loop() {

  // PIR Sensor with state-change detection

  pirState = digitalRead(PIR_PIN);

  if (pirState == HIGH && lastPirState == LOW) {

    entryCount++;

    displayMessage("Motion Detected!", entryCount);

    Serial.println("Entry detected!");

    delay(500);

  }

  lastPirState = pirState;



  // Ultrasonic Sensor: Check distance
```

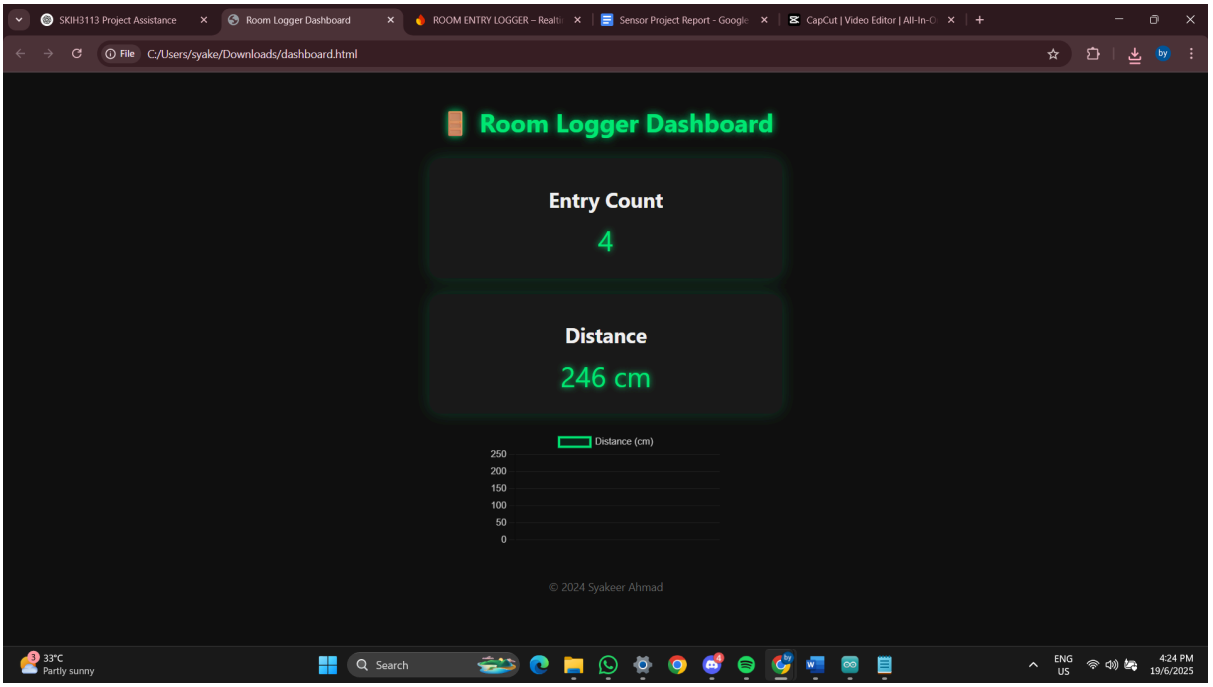
```
digitalWrite(TRIG_PIN, LOW);

delayMicroseconds(2);

digitalWrite(TRIG_PIN, HIGH);

delayMicroseconds(10);

digitalWrite(TRIG_PIN, LOW);



duration = pulseIn(ECHO_PIN, HIGH, 30000);

if (duration == 0) {

  distance = 0;

} else {

  distance = duration * 0.034 / 2;

}



Serial.print("Distance: ");

Serial.print(distance);

Serial.println(" cm");



display.clearDisplay();

display.setTextSize(1);

display.setTextColor(SSD1306_WHITE);

display.setCursor(0, 0);

display.print("Distance: ");

display.print(distance);

display.println(" cm");
```

```
  display.setCursor(0, 12);
  display.print("Entries: ");
  display.println(entryCount);
  display.display();


  if (distance > 0 && distance < 20) {
    digitalWrite(RELAY_PIN, HIGH);
  } else {
    digitalWrite(RELAY_PIN, LOW);
  }


  // Send to Firebase with result check
  if (Firebase.RTDB.setInt(&fbdo, "/RoomLogger/entryCount",
entryCount)) {
    Serial.println("Entry Count sent to Firebase.");
  } else {
    Serial.print("Failed to send Entry Count: ");
    Serial.println(fbdo.errorReason());
  }


  if (Firebase.RTDB.setInt(&fbdo, "/RoomLogger/distance", distance)) {
    Serial.println("Distance sent to Firebase.");
  } else {
    Serial.print("Failed to send Distance: ");
    Serial.println(fbdo.errorReason());
```

```
  }


  // Web server handler
  WiFiClient client = server.available();
  if (client) {
    String request = client.readStringUntil('\r');
    client.flush();


    if (request.indexOf("GET / ") >= 0) {
      String response = "HTTP/1.1 200 OK\r\nContent-Type:
text/html\r\n\r\n";
      response += "<!DOCTYPE html><html><head><title>Room
Logger</title></head><body
style='font-family:Arial;background:#121212;color:#f1f1f1;text-align:ce
nter;'>";
      response += "<h2>Room Logger System</h2>";
      response += "<p><strong>Entry Count:</strong> " +
String(entryCount) + "</p>";
      response += "<p><strong>Distance:</strong> " + String(distance) +
" cm</p>";
      response += "</body></html>";
      client.print(response);
    }


    client.stop();
  }
```

```
  delay(500);
}


// OLED motion message display
void displayMessage(String message, int count) {
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 0);
  display.println(message);
  display.setCursor(0, 20);
  display.print("Entries: ");
  display.println(count);
  display.display();
}
```

Web or Mobile App Interface



Links:

Youtube Link: https://youtube.com/shorts/OxDSwzscNTo?feature=share
Github Link: https://github.com/Syakeermad/Final-Project-Assignment-Smart-Sensor-Based-System-