// Libraries for ESP32-CAM and sensors

#include <WiFi.h>

#include <WebServer.h>

#include <WiFiClient.h>

#include <ESP32Camera.h>

#include <EEPROM.h>

#include <Wire.h>

// Constants for WiFi and Blynk

const char\* ssid = "your\_wifi\_ssid";

const char\* password = "your\_wifi\_password";

const char\* blynk\_auth\_token = "your\_blynk\_auth\_token";

// Constants for ultrasonic sensor

const int trigPin = 13;

const int echoPin = 12;

long duration;

int distance;

// Constants for PIR sensor and buzzer

const int pirPin = 15;

const int buzzerPin = 14;

// Function to send notification to Blynk app

void sendNotification() {

WiFiClient client;

if (!client.connect("blynk-cloud.com", 80)) {

return;

}

String url = "/"+ blynk\_auth\_token + "/notify";

String postData = "Intruder detected!";

client.println("POST " + url + " HTTP/1.1");

client.println("Host: blynk-cloud.com");

client.println("Content-Type: application/x-www-form-urlencoded");

client.print("Content-Length: ");

client.println(postData.length());

client.println();

client.println(postData);

}

void setup() {

// Initialize serial communication for debugging

Serial.begin(115200);

// Connect to WiFi network

WiFi.begin(ssid, password);

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

delay(1000);

Serial.println("Connecting to WiFi...");

}

Serial.println("Connected to WiFi");

// Initialize ultrasonic sensor

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

// Initialize PIR sensor and buzzer

pinMode(pirPin, INPUT);

pinMode(buzzerPin, OUTPUT);

}

void loop() {

// Read distance from ultrasonic sensor

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance = duration \* 0.034 / 2;

// If intruder is detected outside, send notification to Blynk app and trigger ESP-2 Cam

if (distance < 50) {

sendNotification();

// Code to trigger ESP-2 Cam to start live streaming video

}

// If intruder is detected inside, sound the buzzer

if (digitalRead(pirPin) == HIGH) {

digitalWrite(buzzerPin, HIGH);

delay(1000);

digitalWrite(buzzerPin, LOW);

}

// Wait for 1 second before checking sensors again

delay(1000);

}