// code

#include <ESP8266WiFi.h>

#include <ESP8266HTTPClient.h>

#include <SPI.h>

#include <MFRC522.h>

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#define TOTAL\_SLOTS 50

const int TRIG\_PINS[TOTAL\_SLOTS] = {5, 6, 7, 8, 9};

const int ECHO\_PINS[TOTAL\_SLOTS] = {4, 3, 2, 1, 0};

const int LED\_RED\_PINS[TOTAL\_SLOTS] = {15, 14, 13, 12, 11};

const int LED\_GREEN\_PINS[TOTAL\_SLOTS] = {10, 9, 8, 7, 6};

#define SS\_PIN 15

#define RST\_PIN 16

MFRC522 rfid(SS\_PIN, RST\_PIN);

const char\* ssid = "Your\_WiFi\_SSID";

const char\* password = "Your\_WiFi\_Password";

const char\* serverURL = "http://your-cloud-server.com/upload";

WiFiClient client;

LiquidCrystal\_I2C lcd(0x27, 16, 2); // LCD at I2C address 0x27, 16x2 display

void setup() {

    Serial.begin(115200);

    SPI.begin();

    rfid.PCD\_Init();

    lcd.begin();

    lcd.backlight();

    lcd.setCursor(0, 0);

    lcd.print("Smart Parking");

    delay(2000);

    lcd.clear();

    for (int i = 0; i < TOTAL\_SLOTS; i++) {

        pinMode(TRIG\_PINS[i], OUTPUT);

        pinMode(ECHO\_PINS[i], INPUT);

        pinMode(LED\_RED\_PINS[i], OUTPUT);

        pinMode(LED\_GREEN\_PINS[i], OUTPUT);

    }

    WiFi.begin(ssid, password);

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

        delay(1000);

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

lcd.setCursor(0, 0);

        lcd.print("Connecting WiFi...");

    }

    Serial.println("Connected to WiFi!");

    lcd.clear();

    lcd.setCursor(0, 0);

    lcd.print("WiFi Connected!");

    delay(1000);

}

void loop() {

    bool slotOccupied[TOTAL\_SLOTS] = {false};

    int availableSlot = -1;

    for (int i = 0; i < TOTAL\_SLOTS; i++) {

        digitalWrite(TRIG\_PINS[i], LOW);

        delayMicroseconds(2);

        digitalWrite(TRIG\_PINS[i], HIGH);

        delayMicroseconds(10);

        digitalWrite(TRIG\_PINS[i], LOW);

        long duration = pulseIn(ECHO\_PINS[i], HIGH);

        int distance = duration \* 0.034 / 2;

        slotOccupied[i] = (distance < 40);

        if (slotOccupied[i]) {

            digitalWrite(LED\_RED\_PINS[i], HIGH);

            digitalWrite(LED\_GREEN\_PINS[i], LOW);

            Serial.print("Slot ");

            Serial.print(i + 1);

            Serial.println(" is Occupied");

        } else {

            digitalWrite(LED\_RED\_PINS[i], LOW);

            digitalWrite(LED\_GREEN\_PINS[i], HIGH);

            Serial.print("Slot ");

            Serial.print(i + 1);

            Serial.println(" is Available");

            if (availableSlot == -1) {

                availableSlot = i + 1;

            }

        }

    }

    // Display the first available slot on LCD

    lcd.clear();

    lcd.setCursor(0, 0);

    lcd.print("Parking Status:");

if (availableSlot != -1) {

        lcd.setCursor(0, 1);

        lcd.print("Slot Free: ");

        lcd.print(availableSlot);

    } else {

        lcd.setCursor(0, 1);

        lcd.print("No Slots Free!");

    }

    // RFID Detection

    String vehicleID = "";

    if (rfid.PICC\_IsNewCardPresent() && rfid.PICC\_ReadCardSerial()) {

        Serial.print("Vehicle ID: ");

        for (byte i = 0; i < rfid.uid.size; i++) {

            Serial.print(rfid.uid.uidByte[i], HEX);

            vehicleID += String(rfid.uid.uidByte[i], HEX);

        }

        Serial.println();

        rfid.PICC\_HaltA();

    }

    // Send data to cloud server

    if (WiFi.status() == WL\_CONNECTED) {

        HTTPClient http;

        http.begin(client, serverURL);

        http.addHeader("Content-Type", "application/x-www-form-urlencoded");

        String postData = "vehicleID=" + vehicleID + "&availableSlot=" + String(availableSlot);

        int httpResponseCode = http.POST(postData);

        Serial.println("Data Sent! Response: " + String(httpResponseCode));

        http.end();

    }

    delay(2000);

}