**SMART PARKING SYSTEM**

#include <Servo.h>

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#include <ESP8266WiFi.h>

#include <ESP8266HTTPClient.h>

Servo gateServo1;

Servo gateServo2;

LiquidCrystal\_I2C lcd(0x27, 16, 2); // Change the address if your LCD is different

const int irSensor1 = D2; // IR sensor pins

const int irSensor2 = D3;

const int irSensor3 = D4;

const int irSensor4 = D5;

const int gatePin1 = D6; // Servo motor control pins

const int gatePin2 = D7;

bool isOccupied1 = false;

bool isOccupied2 = false;

const char\* ssid = "YourSSID";

const char\* password = "YourPassword";

const char\* serverURL = "http://yourserver.com"; // Change to your server

void setup() {

pinMode(irSensor1, INPUT);

pinMode(irSensor2, INPUT);

pinMode(irSensor3, INPUT);

pinMode(irSensor4, INPUT);

gateServo1.attach(gatePin1);

gateServo2.attach(gatePin2);

lcd.init();

lcd.backlight();

Serial.begin(115200);

// Connect to Wi-Fi

WiFi.begin(ssid, password);

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

delay(1000);

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

}

}

void loop() {

int sensor1Value = digitalRead(irSensor1);

int sensor2Value = digitalRead(irSensor2);

int sensor3Value = digitalRead(irSensor3);

int sensor4Value = digitalRead(irSensor4);

if (sensor1Value == LOW || sensor2Value == LOW) {

openGate(1);

isOccupied1 = true;

} else {

closeGate(1);

isOccupied1 = false;

}

if (sensor3Value == LOW || sensor4Value == LOW) {

openGate(2);

isOccupied2 = true;

} else {

closeGate(2);

isOccupied2 = false;

}

updateLCD(isOccupied1, isOccupied2);

sendOccupancyStatus(isOccupied1, isOccupied2);

}

void openGate(int gate) {

if (gate == 1) {

gateServo1.write(90); // Open the first gate (adjust as needed)

} else if (gate == 2) {

gateServo2.write(90); // Open the second gate (adjust as needed)

}

delay(2000); // Open for 2 seconds (adjust as needed)

closeGate(gate);

}

void closeGate(int gate) {

if (gate == 1) {

gateServo1.write(0); // Close the first gate

} else if (gate == 2) {

gateServo2.write(0); // Close the second gate

}

}

void updateLCD(bool occupied1, bool occupied2) {

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Spot 1: ");

lcd.print(occupied1 ? "Occupied" : "Vacant");

lcd.setCursor(0, 1);

lcd.print("Spot 2: ");

lcd.print(occupied2 ? "Occupied" : "Vacant");

}

void sendOccupancyStatus(bool occupied1, bool occupied2) {

HTTPClient http;

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

String url = serverURL + "/update?spot1=" + (occupied1 ? "occupied" : "vacant") + "&spot2=" + (occupied2 ? "occupied" : "vacant");

http.begin(url);

int httpCode = http.GET();

if (httpCode == 200) {

Serial.println("Status updated successfully");

} else {

Serial.println("Failed to update status");

}

http.end();

}

}