PARKBAI: PARKING SENSOR SOURCE CODE

```
#include <ESP8266WiFi.h>
#include "FirebaseESP8266.h"
#include <NewPing.h>

#define FIREBASE_HOST "your firebase Realtime database link here"
#define FIREBASE_AUTH " your firebase Realtime database key here "

// Replace with your network credentials
const char* ssid = "your ssid name here";
const char* password = " your ssid password here ";
#define TRIGGER_PIN D5
#define ECHO_PIN D6
#define MAX_DISTANCE 396.0
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);  //Max distance 20 cm
//const int ledPin = D7;

FirebaseConfig config;
FirebaseAuth auth;
FirebaseData firebaseData;
FirebaseJson json;

String ownerUID = "bFnYTNgnXMYLrwwjqDoxzreu7Rn2";

void setup() {
  Serial.begin(9600);
  //pinMode(ledPin, OUTPUT);

  // Connect to WiFi network
```

```
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println(".");
  }
  Serial.println("Successfully connected.");
  Serial.print("ESP Board MAC Address:");
  Serial.println(WiFi.macAddress());
  Serial.print("http://");
  Serial.print(WiFi.localIP());
  config.database_url = FIREBASE_HOST;
  config.signer.tokens.legacy_token = FIREBASE_AUTH;
  Firebase.reconnectWiFi(true);
  Firebase.begin(&config, &auth);
  //Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}

void loop() {
  delay(100);
  unsigned int distance_cm = sonar.ping_cm();
  unsigned int distance_inches = distance_cm / 2.54; // CENTIMETER TO INCHES
  unsigned int detectOccupied =  3.0;
  unsigned int detectImproperPark1 =  4.0;
  unsigned int detectImproperPark2 =  7.0;
  unsigned int detectVacant =  8.0;

  Serial.print("Distance: ");
  Serial.print(distance_inches);
  Serial.println(" inches");
```

```
 json.set("parking_number", 1);
 json.set("parking_section", "A");


 if (distance_inches <= detectOccupied) {

  //(ledPin, HIGH);
  Serial.println("OCCUPIED");
  json.set("parking_space", "OCCUPIED");
  Firebase.updateNode(firebaseData, "PARK_OWNER/" + ownerUID + "/PARKING_AREA/A01", json);

 } else if (distance_inches >= detectImproperPark1 && distance_inches < detectImproperPark2) {

  //digitalWrite(ledPin, HIGH);
  Serial.println("IMPROPER PARK");
  json.set("parking_space", "IMPROPER PARK");
  Firebase.updateNode(firebaseData, "PARK_OWNER/" + ownerUID + "/PARKING_AREA/A01", json);


 }

 else if (distance_inches >= detectVacant){
  //digitalWrite(ledPin, LOW);
  Serial.println("VACANT");
  json.set("parking_space", "VACANT");
  Firebase.updateNode(firebaseData, "PARK_OWNER/" + ownerUID + "/PARKING_AREA/A01", json);
 }
}
```