PARKBAI: RFID SCANNER SOURCE CODE (RFID CARD REGISTRATION)

```
#include <ESP8266WiFi.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <SPI.h>
#include <RFID.h>
#include "FirebaseESP8266.h" // Install Firebase  ESP8266 library

#define FIREBASE_HOST "your firebase Realtime database link here"  //Without http:// or https:// schemes
#define FIREBASE_AUTH " your firebase Realtime database key here"
RFID rfid(D4, D3); //D4:pin of tag reader SDA. D3:pin of tag reader RST
unsigned char str[MAX_LEN]; //MAX_LEN is 16: size of the array
LiquidCrystal_I2C lcd(0x27,16,2); // set the LCD address to 0x27 for a 16 chars and 2 line display

const char ssid[] = "your ssid name here";
const char pass[] = " your ssid password here";
// String message;
FirebaseConfig config;
FirebaseAuth auth;
FirebaseData firebaseData; //Define FirebaseESP8266 data object

void connect() {

  Serial.print("CONNECTING TO WIFI PLEASE WAIT.");

  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(1000);
  }

  Serial.println("\nYOUR RFID SCANNER CONNECTED SUCCESSFULLY");
  Serial.print("ESP Board MAC Address:");
  Serial.println(WiFi.macAddress());
  Serial.print("IP Address: ");
  Serial.print("http://");
  Serial.print(WiFi.localIP());
  Serial.println("/");

}

void setup() {

  Serial.begin(9600);
  WiFi.begin(ssid, pass);

  lcd.init();  // initialize the lcd
  lcd.clear();
  lcd.backlight();

  SPI.begin();
  rfid.init();

  connect();
  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 pushUser(String rfidnumber)  //Function to check if an identified tag is registered to allow access
{

  lcd.setCursor(3,0);
  lcd.print("RFID CARD");

  lcd.setCursor(3,1);
  lcd.print("REGISTERED!");

  delay(1000);
  lcd.clear();

  Serial.println("RFID CARD SUCCESSFULLY REGISTERED: " + rfidnumber);

  String rfidvalue = "not_owned";

  Firebase.setString(firebaseData, "/ADMIN/RFID_CARDS/" + rfidnumber, rfidvalue);
}
void loop() {
  if (rfid.findCard(PICC_REQIDL, str) == MI_OK)  //Wait for a tag to be placed near the reader
  {
    Serial.println("RFID CARD DETECTED!");
    String rfidcard = "";             //Temporary variable to store the read RFID number
    if (rfid.anticoll(str) == MI_OK)  //Anti-collision detection, read tag serial number
    {
      Serial.print("The RFID card number is : ");
      for (int i = 0; i < 4; i++)  //Record and display the tag serial number
      {
        rfidcard = rfidcard + (0x0F & (str[i] >> 4));
        rfidcard = rfidcard + (0x0F & str[i]);
      }
      Serial.println(rfidcard);
      pushUser(rfidcard);  //Check if the identified tag is an allowed to open tag
    }
    rfid.selectTag(str);  //Lock card to prevent a redundant read, removing the line will make the sketch read cards continually
  }
  rfid.halt();


  lcd.setCursor(1,0);
  lcd.print("TAP NEW CARD");


  lcd.setCursor(2,1);
  lcd.print("TO REGISTER");
  delay(300);
  lcd.clear();
}
```