CODE :

```
#include "HX711.h"
#include <ESP8266WiFi.h>
#include <Firebase_ESP_Client.h>
#include "addons/TokenHelper.h"
#include "addons/RTDBHelper.h"

// HX711 circuit wiring
const int LOADCELL_DOUT_PIN = D1;
const int LOADCELL_SCK_PIN = D2;
const int BUZZER_PIN = D3; // Digital pin for BUZZER module
const int LED_PIN = D4;   // Digital pin for LED

HX711 scale;

#define WIFI_SSID "12345678"
#define WIFI_PASSWORD "12345678"
#define API_KEY "AIzaSyCfagYS4c8SvrSwuGT6XSWQP1ABSAPfSRM"
#define DATABASE_URL "https://nodemcu-abee1-default-rtdb.firebaseio.com/"

FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;
unsigned long sendDataPrevMillis = 0;
bool signupOK = false;

void setup() {
  Serial.begin(115200);
  Serial.println();
  Serial.println("HX711 Demo");
Serial.println("Initializing the scale");

scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);

Serial.println("Before setting up the scale:");
Serial.print("read: \t\t");
Serial.println(scale.read());

Serial.print("get value: \t\t");
Serial.println(scale.get_value(5));

Serial.print("get units: \t\t");
Serial.println(scale.get_units(5), 1);

scale.set_scale(-478.507);
scale.tare();
pinMode(BUZZER_PIN, OUTPUT);
digitalWrite(BUZZER_PIN, LOW);

pinMode(LED_PIN, OUTPUT);
digitalWrite(LED_PIN, LOW);

Serial.println("After setting up the scale:");

Serial.print("read: \t\t");
Serial.println(scale.read());

Serial.print("get value: \t\t");
Serial.println(scale.get_value(5));
Serial.print("get units: \t\t");
Serial.println(scale.get_units(5), 1);

Serial.println("Readings:");

 WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
Serial.print("Connecting to Wi-Fi");
while (WiFi.status() != WL_CONNECTED)
{
  Serial.print(".");
  delay(300);
}
Serial.println();
Serial.print("Connected with IP: ");
Serial.println(WiFi.localIP());
Serial.println();
config.api_key = API_KEY;
config.database_url = DATABASE_URL;
if (Firebase.signUp(&config, &auth, "", ""))
{
  Serial.println("ok");
  signupOK = true;
}
else
{
  Serial.printf("%s\n", config.signer.signupError.message.c_str());
}
config.token_status_callback = tokenStatusCallback; //see addons/TokenHelper.h
Firebase.begin(&config, &auth);
Firebase.reconnectWiFi(true);
```

```
}

void loop() {
  // Read one unit from the scale
  float currentLoad = scale.get_units(1);

  Serial.print("Current load:\t");
  Serial.println(currentLoad, 1);

  if (currentLoad >= 10.0) {
    // Turn off BUZZER
    digitalWrite(BUZZER_PIN, LOW);
    Serial.println("BUZZER OFF. Load is 10 units or more.");
  } else {
    // Turn on BUZZER
    digitalWrite(BUZZER_PIN, HIGH);
    Serial.println("BUZZER ON. Load is less than 10 units.");
  }

  // Check if the load is greater than or equal to 5 units
  if (currentLoad >= 5.0) {
    // Turn on LED
    digitalWrite(LED_PIN, HIGH);
    Serial.println("LED ON. Load is 5 units or more.");
  } else {
    // Turn off LED
    digitalWrite(LED_PIN, LOW);
    Serial.println("LED OFF. Load is less than 5 units.");
  }
  scale.power_down(); // put the ADC in sleep mode
  delay(5000);

  scale.power_up();

  delay(1000); // Delay before next reading

  if (Firebase.ready() && signupOK && (millis() - sendDataPrevMillis > 1000 || sendDataPrevMillis == 0))
  {
    sendDataPrevMillis = millis();

    if (Firebase.RTDB.setInt(&fbdo, "main/currentLoad", currentLoad))
    {
      Serial.println("PATH: " + fbdo.dataPath());
      Serial.println("TYPE: " + fbdo.dataType());
    }
    else
    {
      Serial.println("Failed REASON: " + fbdo.errorReason());
    }
}
}
```