**CONTROLLING LED (BULB)USING MOBILE APP**

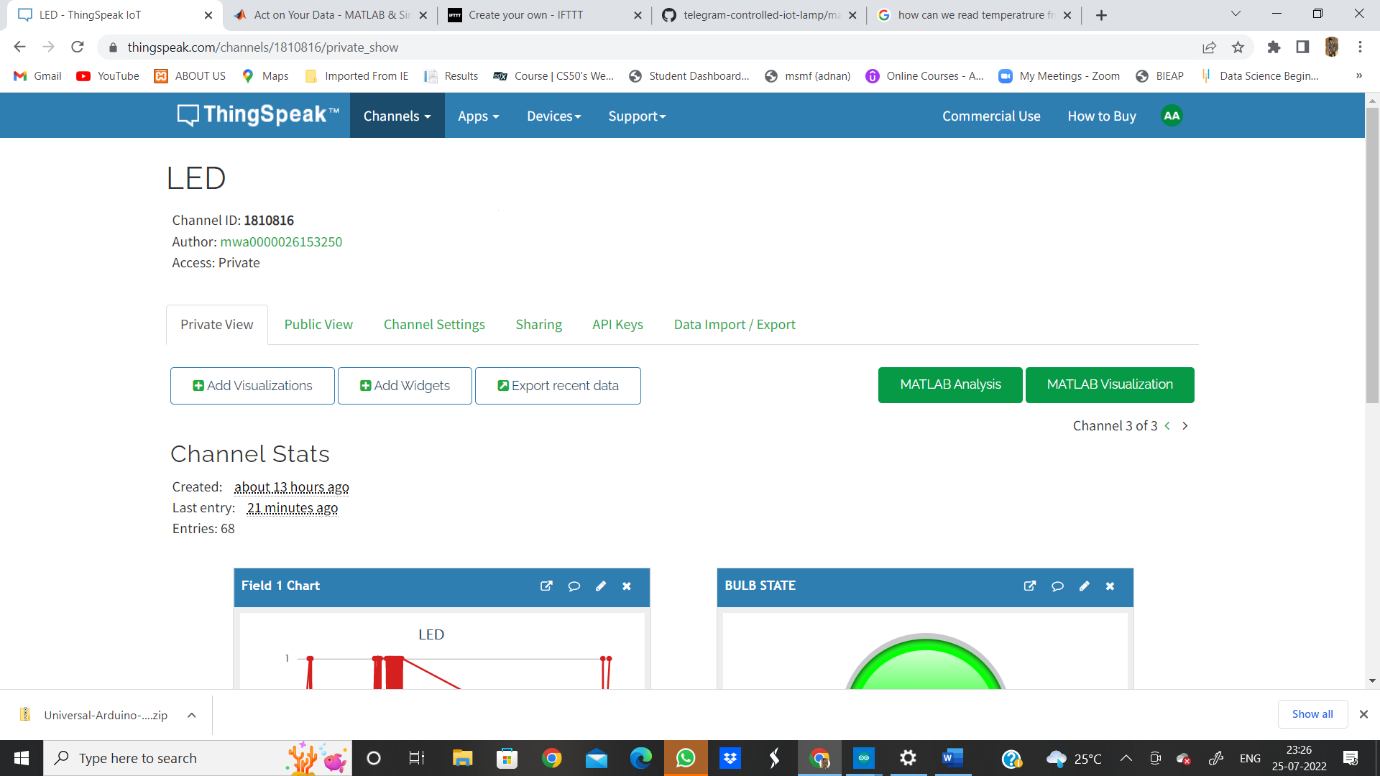
HERE WE WILL BE USING PLATFORMS LIKE “HTTP SHORTCUTS” AND “THINGSPEAK” AS SERVER

**COMPONENTS REQUIRED**

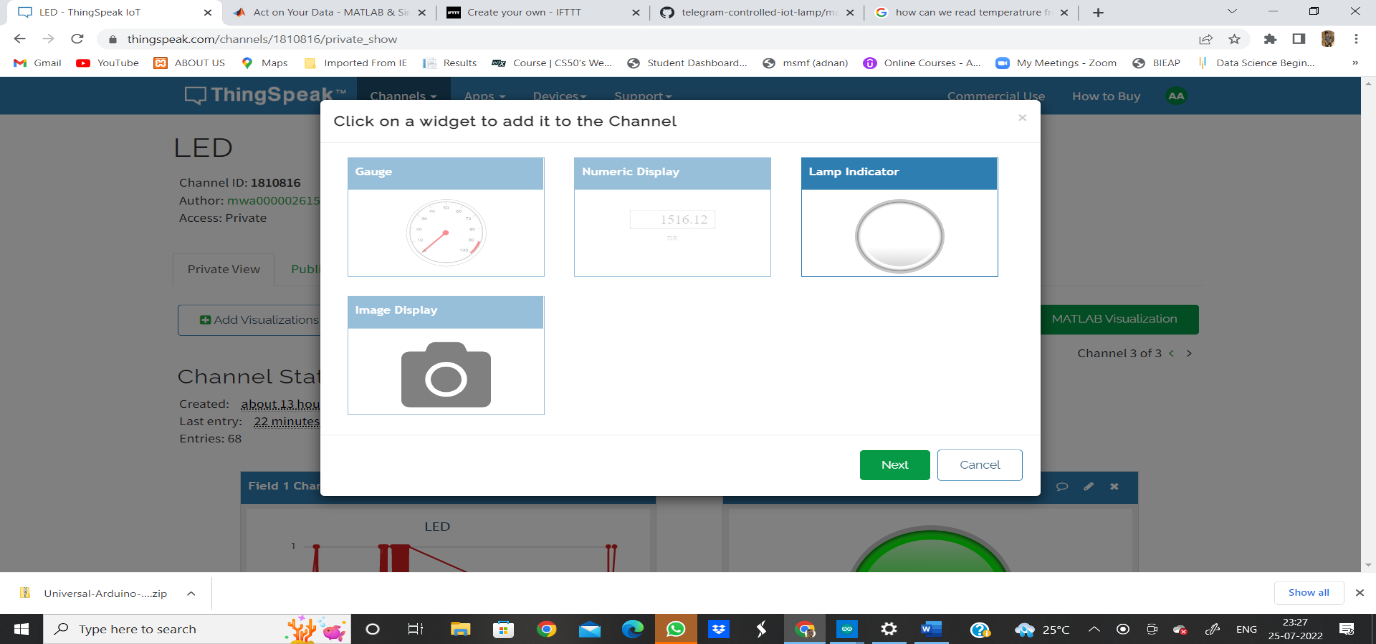
* NodeMCU ESP32
* 1 LED
* CONNECTING WIRES.

**STEP 1:**

CREATE A CHANNEL named ‘’LED” as shown AND ADD FIELD 1 AS “BULB”

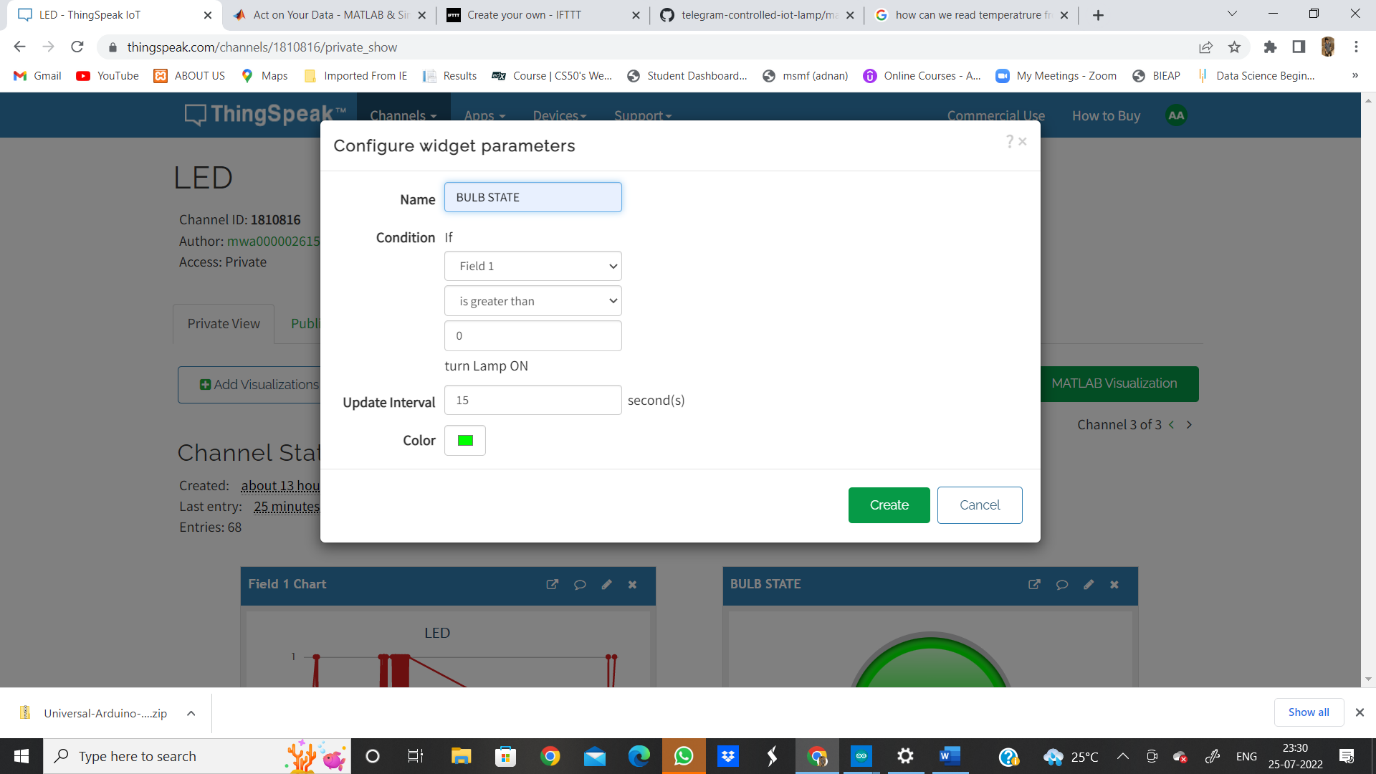
  
**STEP 2 :**

GO TO ADD WIDGETS AND CREATE A WIDGET AND NAME IT AS “BULB STATE”:



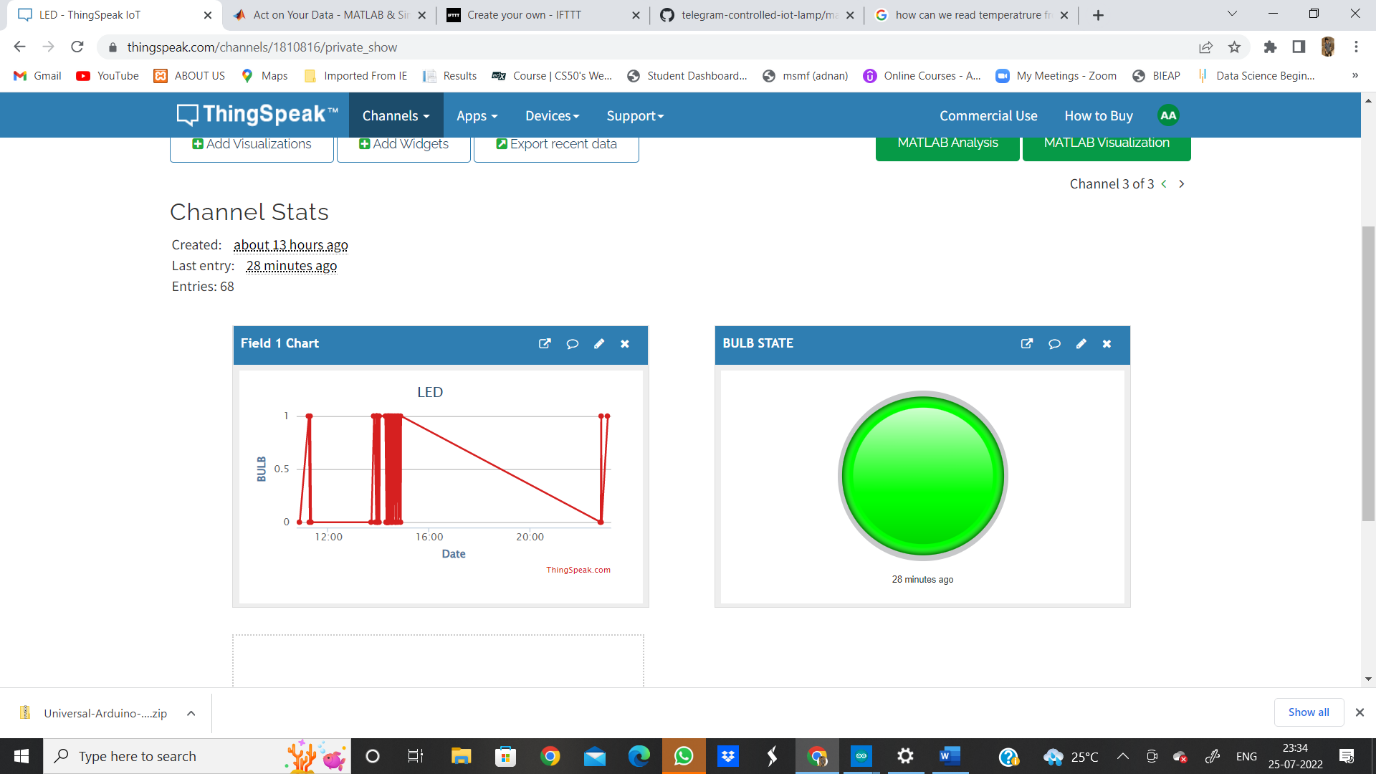
**STEP 3:**

AFTER ADDING WIDGET FILL IN THE FOLLOWING DETAILS : AND PUT THE CONDITION FIELD 1 “ON” WHEN VALUE IS GREATER THAN 0 .



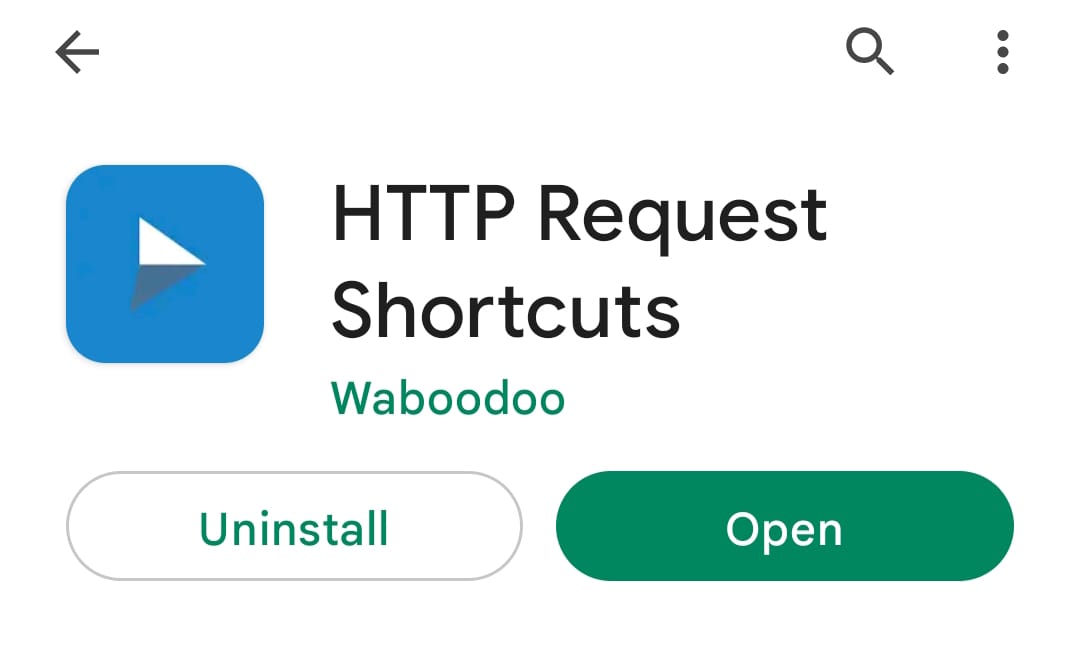
**STEP 4:**

NOW THE DASH BOARD LOOKS SOME THING LIKE THIS.

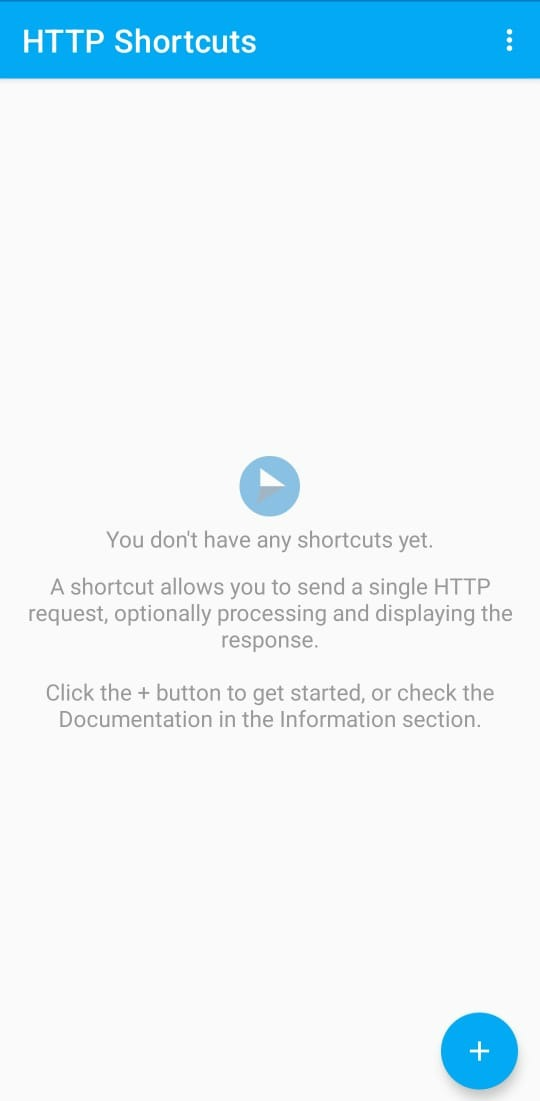


STEP 5 :

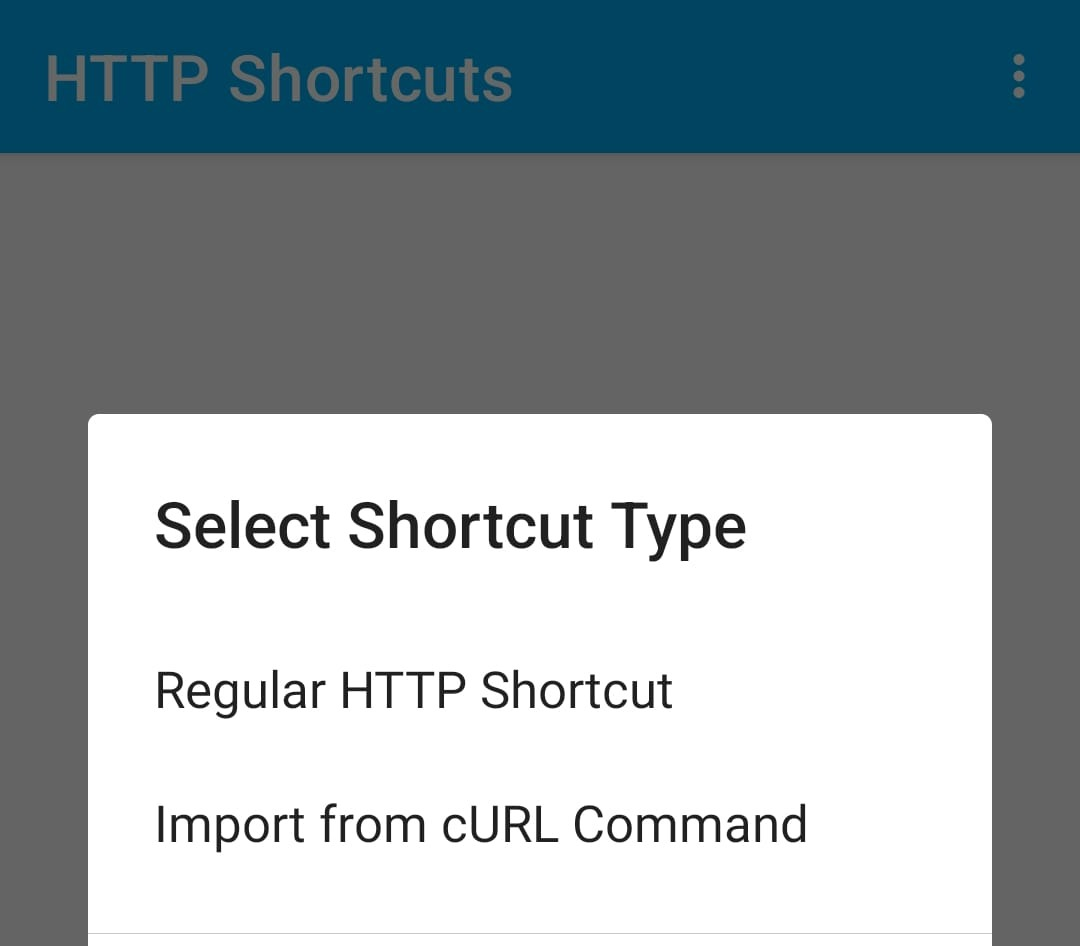
NOW DOWNLOAD “HTTP SHORTCUT APP” FROM GOOGLE PLAYSTORE.



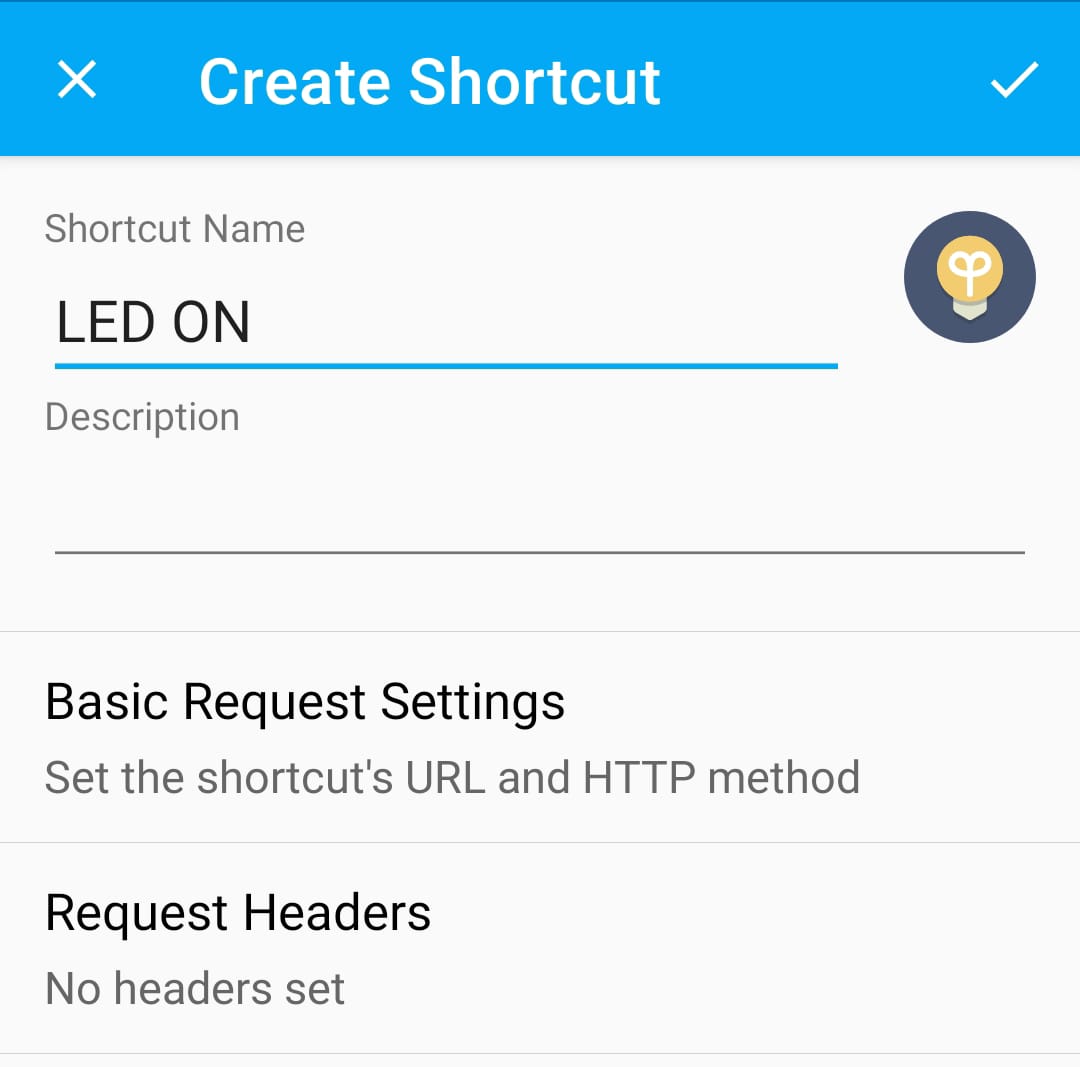
ONCE YOU DOWNLOAD THIS APP OPEN IT AND U WILL FIND AN INTERFACE SOMETHING LIKE THIS .. CLICK ON “+” WHICH IS PRESENT ON THE LEFT BOTTOM.



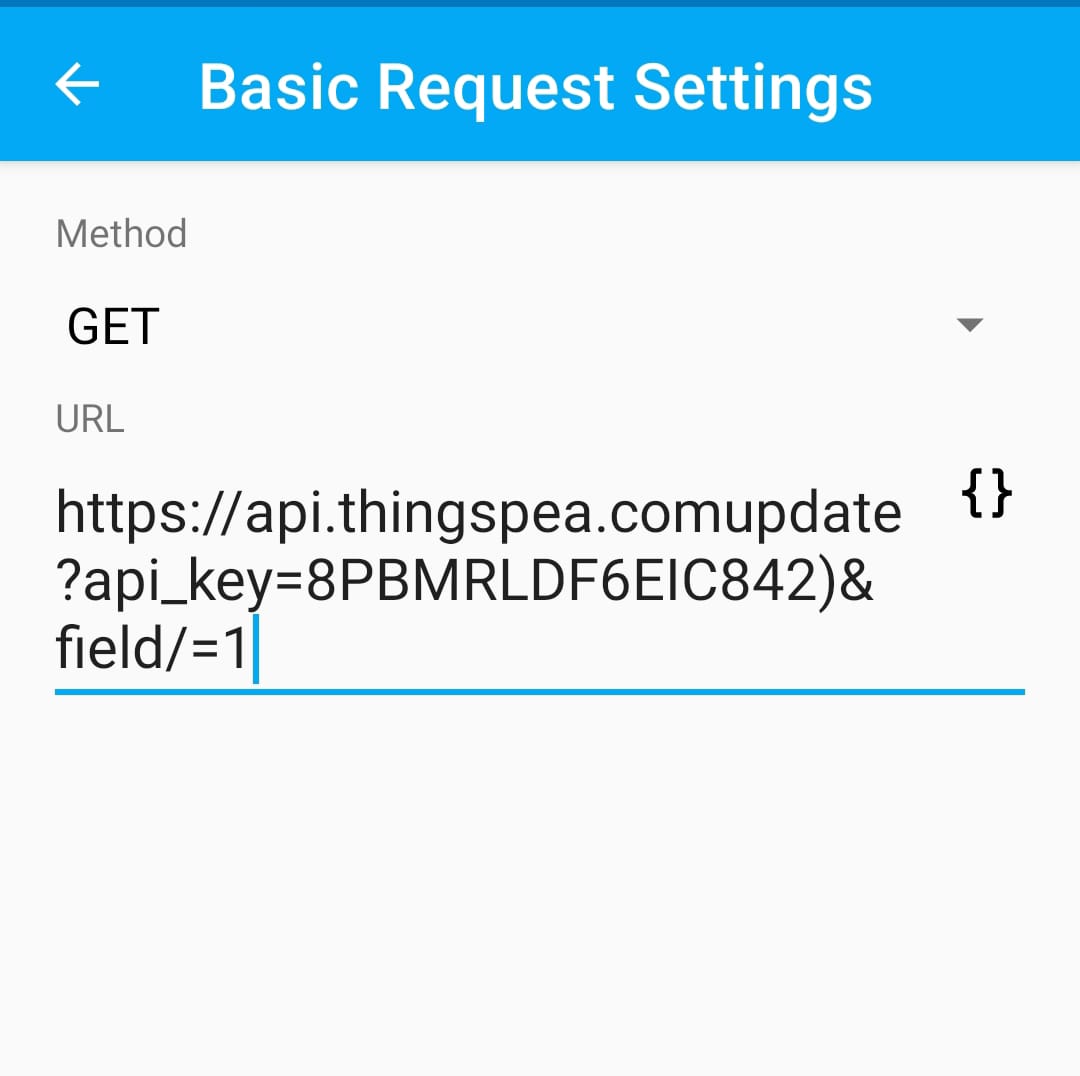
NOW AFTER CLICK IN ON “+” 🡪 CLICK ON “REGULAR HTTP SHORTCUT”.



NEXT FILL IN ALL THE DETAILS AS FOLLOWS.

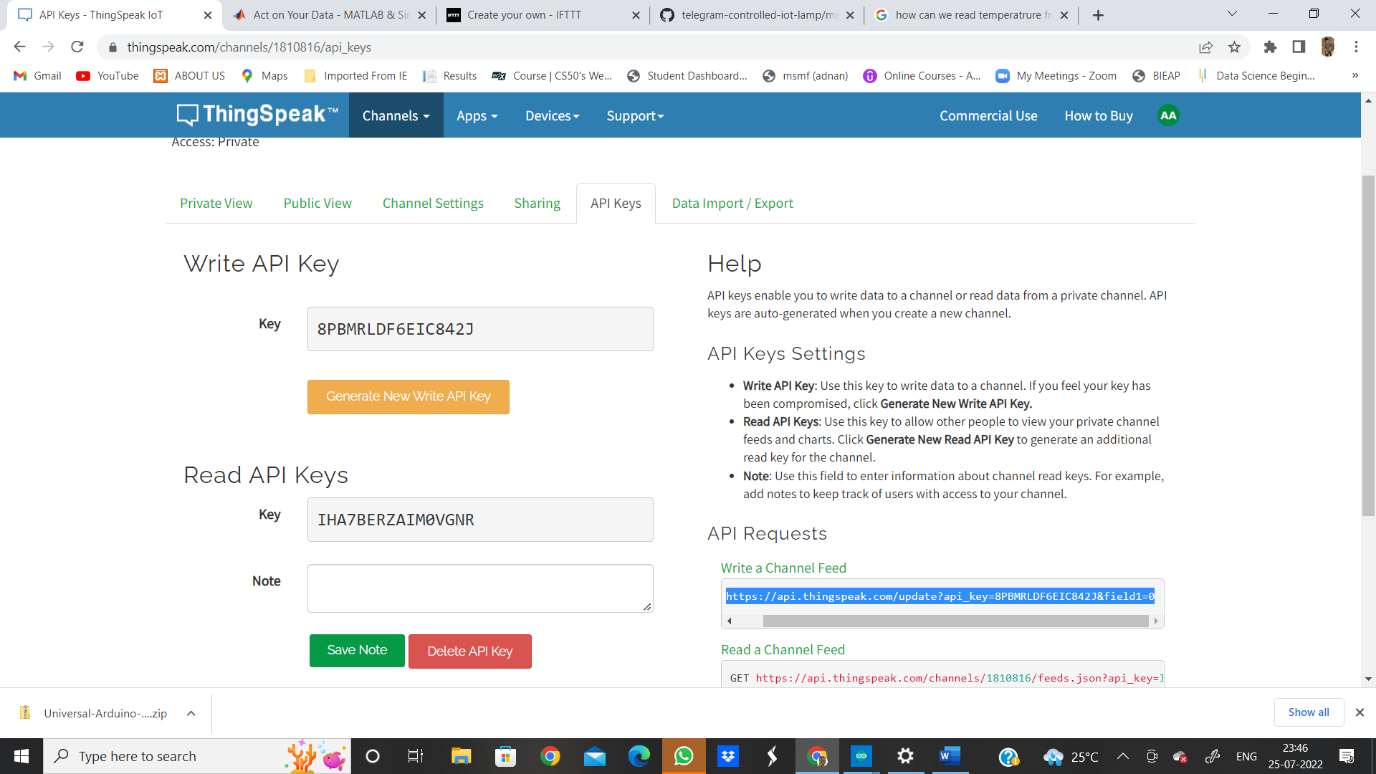


AFTER WRITING THE SHORTCUT NAME 🡪 CLICK ON “BASIC REQUEST SETTINGS”

  
THIS IS FOR TURN – ON.

THE LINK PASTED HERE IS BOUHT FROM THINGSPEAK .. TO RETREIVE THIS LINK FOLLOW THESE STEPS.

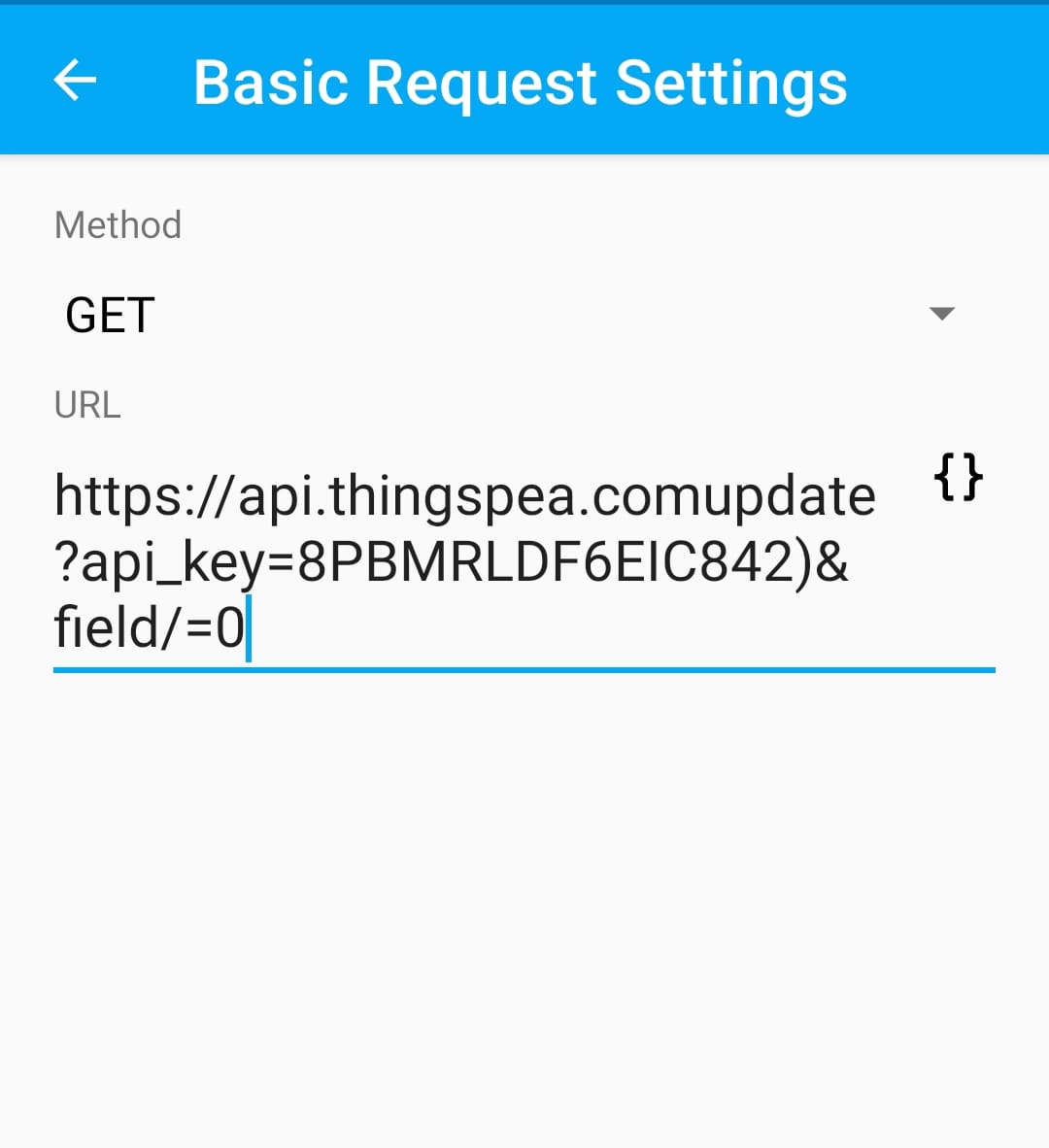
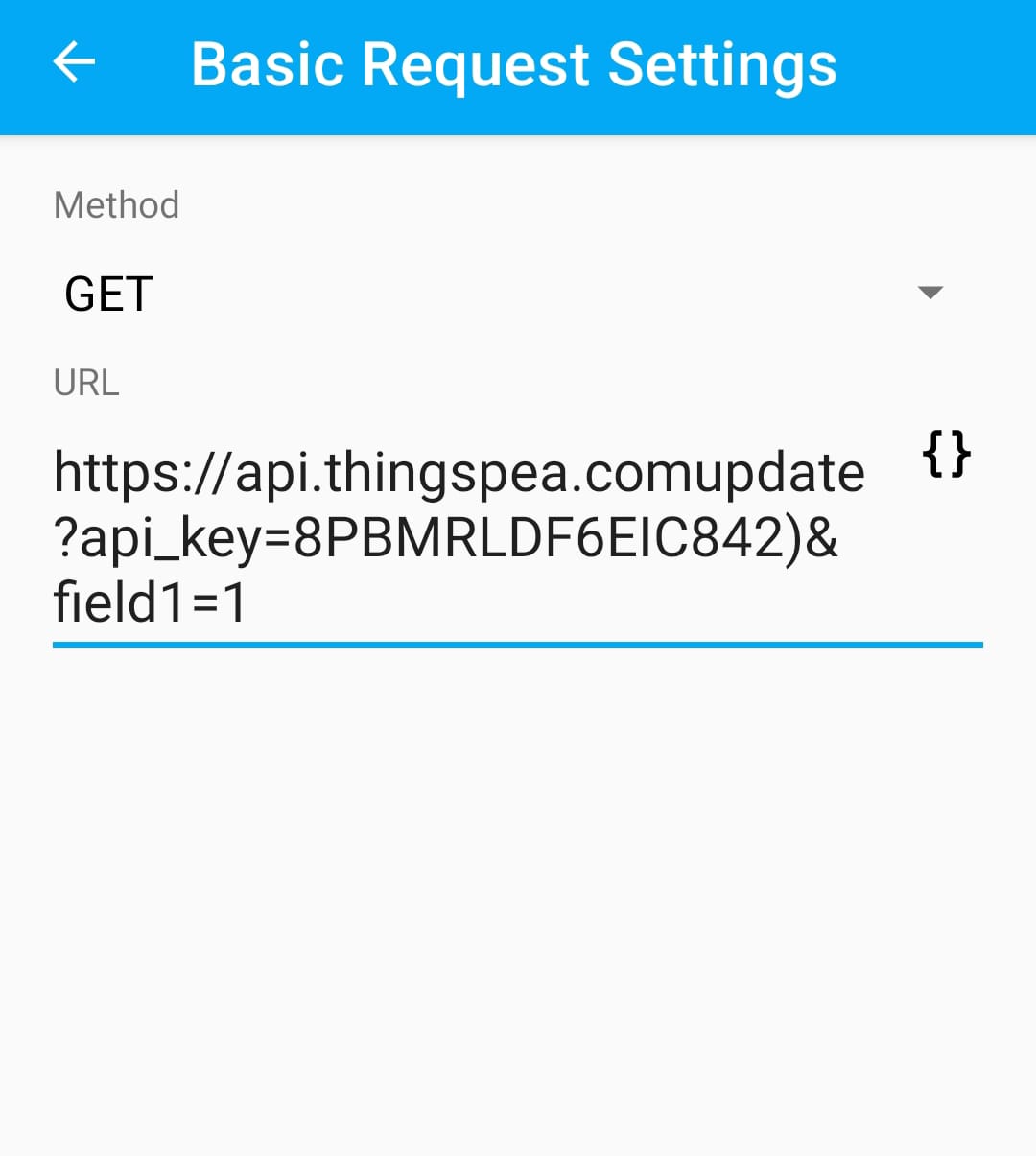
IN THINGSPEAK > GO TO CHANNELS > LED > API KEYS > COPY WRITE CHANNEL API KEY.



PASTE THE COPIED URL HERE:

TURN ON-(CHANGE FILED 1 =1 ) IN THE END OF URL.

SIMILARLY TO TURN OFF – MAKE NO CHANGES.

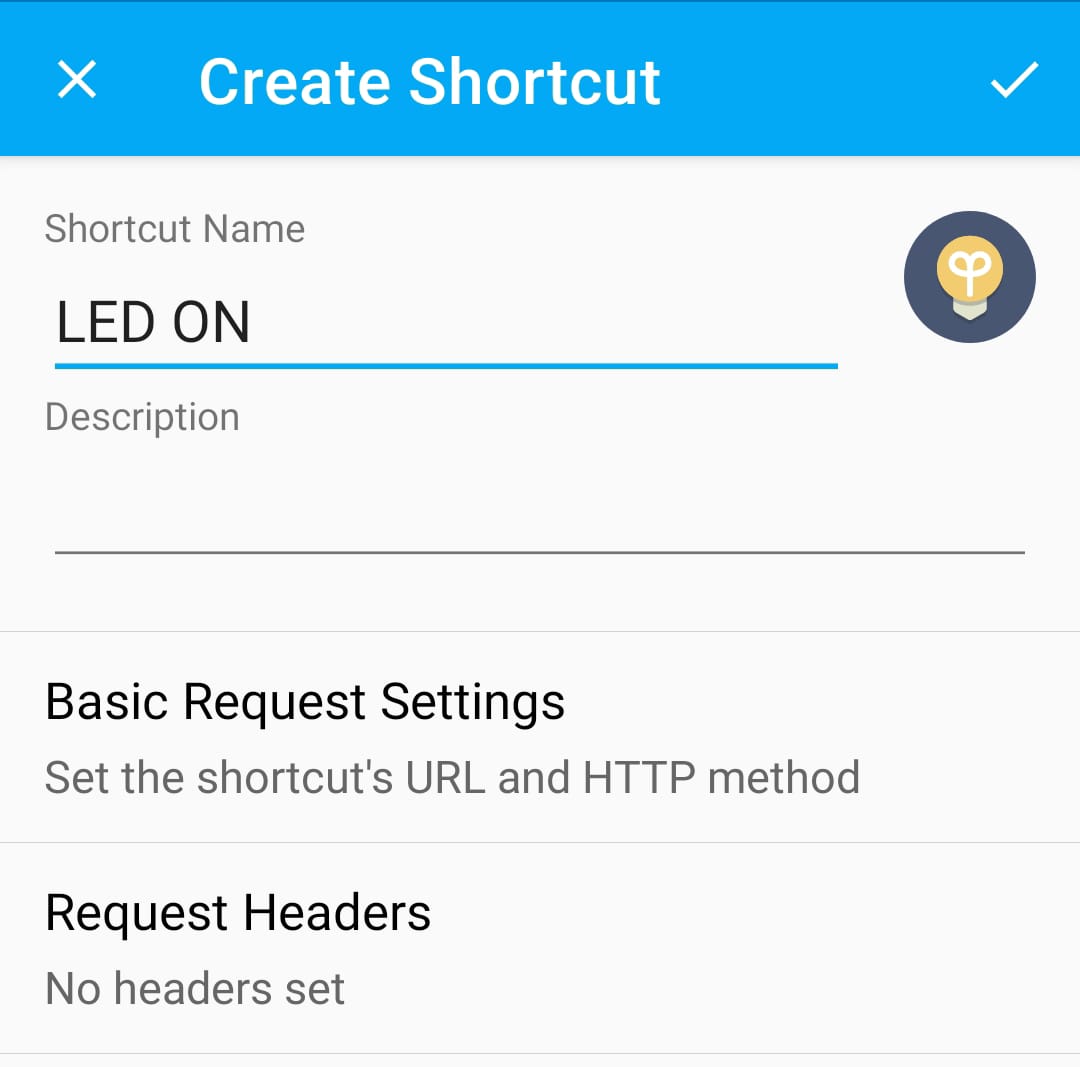


IF FIELD1=0 🡪 OFF.

IF FIELD1=1 🡪 ON.

AND THEN SAVE BOTH THE SHORTCUTS.

THEN OUR HTTP SHORTCUTS APP LOOKS SOMETHING LIKE THIS.



**NOW COMING TO THE HARDWARE PART AND CONNECTIONS:**

HERE WILL USE SIMPLE NODEMCU ESP32 AND LED

CONNECTIONS

* NEGATIVE END (LED) - GND (ESP32)
* POSITIVE END (LED) - D15 (ESP32)

**PROGRAM:**

#include <WiFi.h>

#include <ThingSpeak.h>

int led=15;

char ssid[] = "LG"; // your network SSID (name)

char pass[] = "getjar123";

WiFiClient client;

unsigned long myChannelNumber = 1810816;

const char \* myReadAPIKey = "IHA7BERZAIM0VGNR";

void setup() {

// put your setup code here, to run once:

WiFi.mode(WIFI\_STA);

Serial.begin(115200);

pinMode(led,OUTPUT);

ThingSpeak.begin(client);

}

void loop() {

// put your main code here, to run repeatedly:

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

Serial.print("Attempting to connect to SSID: ");

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

WiFi.begin(ssid, pass); // Connect to WPA/WPA2 network. Change this line if using open or WEP network

Serial.print(".");

delay(5000);

}

Serial.println("\nConnected.");

}

int a = ThingSpeak.readIntField(myChannelNumber, 1, myReadAPIKey);

Serial.println("LED STATUS");

Serial.print(a);

delay(1000);

if (a == 1){

digitalWrite(led,HIGH);

}

else{

digitalWrite(led,LOW);

}

}

SAVE AND RUN THE PROGRAM AND NOW WE CAN SEE THE OUTPUT.

**OR**

HERE WILL USE SIMPLE NODEMCU ESP8266 AND LED

CONNECTIONS

* NEGATIVE END (LED) - GND (ESP8266)
* POSITIVE END (LED) - D4 (ESP8266)

**PROGRAM:**

#include <ESP8266WiFi.h>

#include <ThingSpeak.h>

int bulb1=4;            //(any pin to esp8266)

char ssid[] = "xxxxx";   // your network SSID (wifi hotspot name)

char pass[] = "xxxxx";   //wifi hotspot password

WiFiClient client;

unsigned long myChannelNumber = xxxxx; //channel id/number

const char \* myWriteAPIKey = "xxxxx";     //my channal write key from API KEY

const char \* myReadAPIKey = "xxxxx";     //my channal read key from API KEY

void setup() {

  // put your setup code here, to run once:

   pinMode(bulb1,OUTPUT);

   WiFi.mode(WIFI\_STA);

   ThingSpeak.begin(client);

   Serial.begin(9600);

}

void loop() {

  // put your main code here, to run repeatedly:

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

    Serial.print("Attempting to connect to SSID: ");

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

      WiFi.begin(ssid, pass);  // Connect to WPA/WPA2 network. Change this line if using open or WEP network

      Serial.print(".");

      delay(5000);

    }

    Serial.println("\nConnected.");

  }

  ThingSpeak.setField(1, bulb1);         //(field,bulb)

  int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);

  if(x == 200){

    Serial.println("Channel update successful.");

  }

  else{

    Serial.println("Problem updating channel. HTTP error code " + String(x));

  }

  delay(20000);

  float a = ThingSpeak.readFloatField(myChannelNumber, 1, myReadAPIKey);       //(myChannelNumber, field number, myReadAPIKey)

  Serial.println("ReadValue from ThingSpeak of the mobile app on/off : ");

  Serial.println(a);

  if (a > 0){

  digitalWrite(bulb1,HIGH);

 Serial.println("bulb1 IS ON : ");

  }

  else {

    digitalWrite(bulb1,LOW);

 Serial.println("bulb1 IS OFF: ");

  }

   }