**CONTROL BULB USING GOOGLE ASSISTANT, IFTTT AND THINGSPEAK FROM ANYWAHERE**

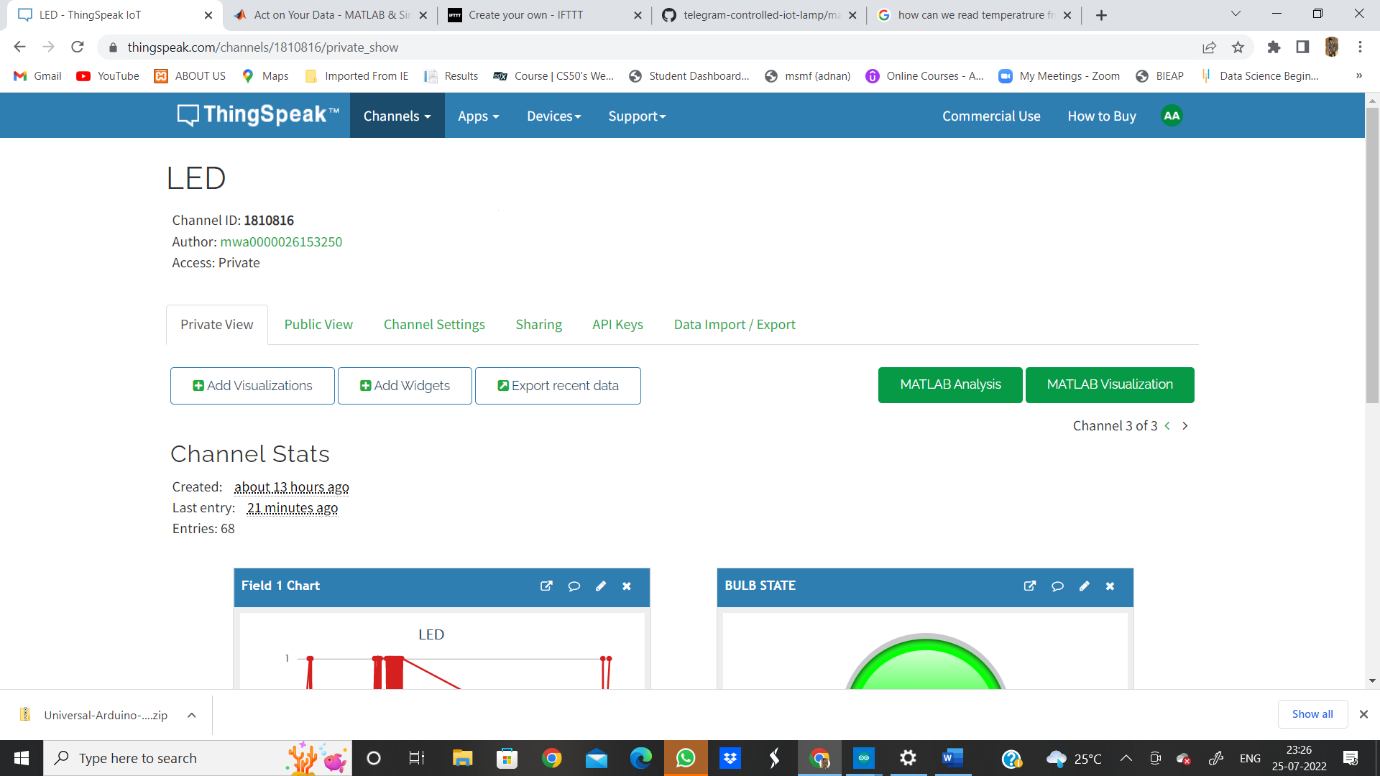
HERE WE WILL BE USING PLATFORMS LIKE “IFTTT” AND “THINGSPEAK” AS SERVER

COMPONENTS REQUIRED

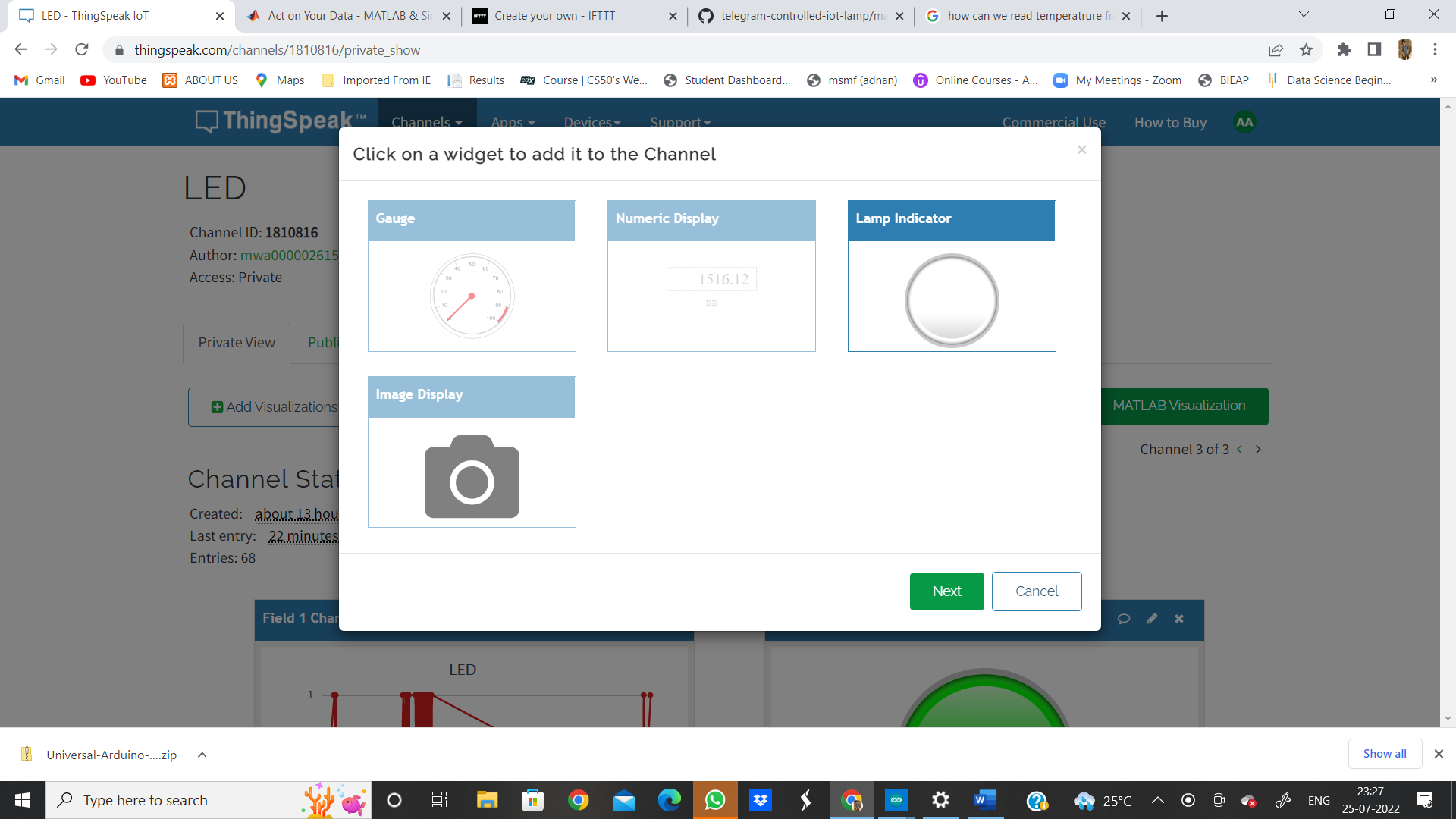
* NodeMCU ESP32
* 1 LED
* CONNECTING WIRES.

**STEP 1:**

CREATE A CHANNEL” LED” AND ADD FIELD 1 AS “BULB”

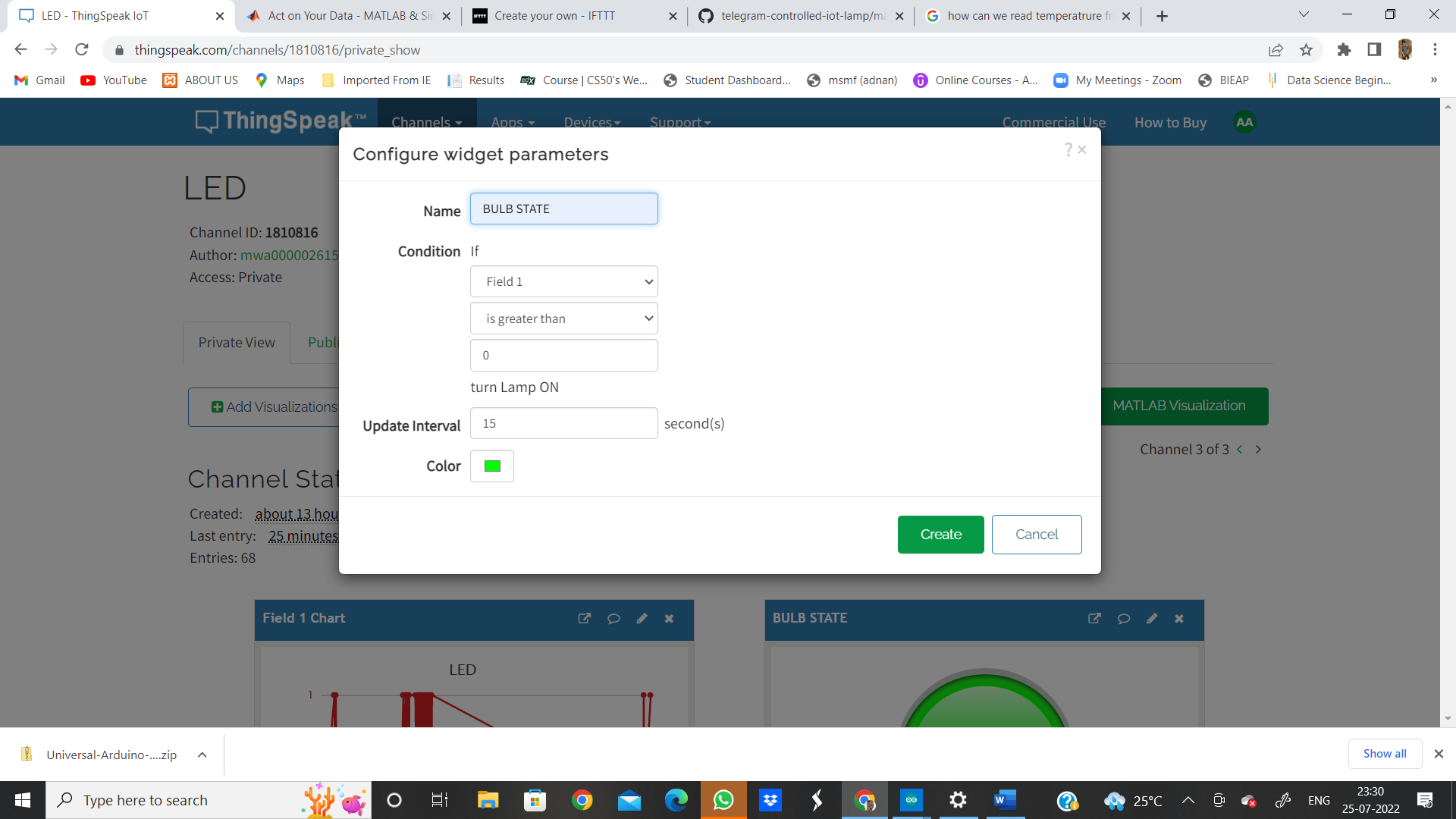
  
**STEP 2 :**

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



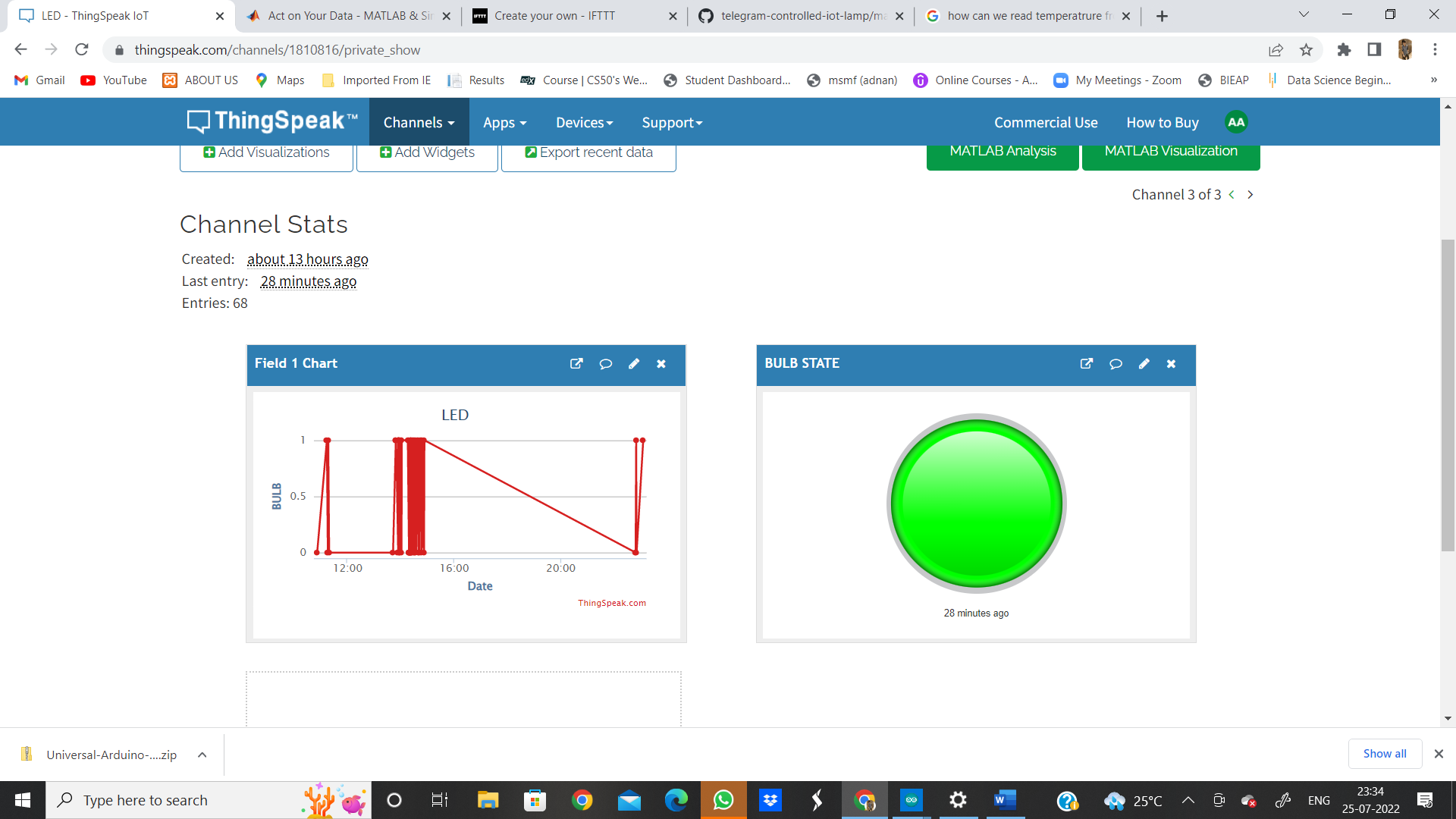
**STEP 3:**

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



**STEP 4:**

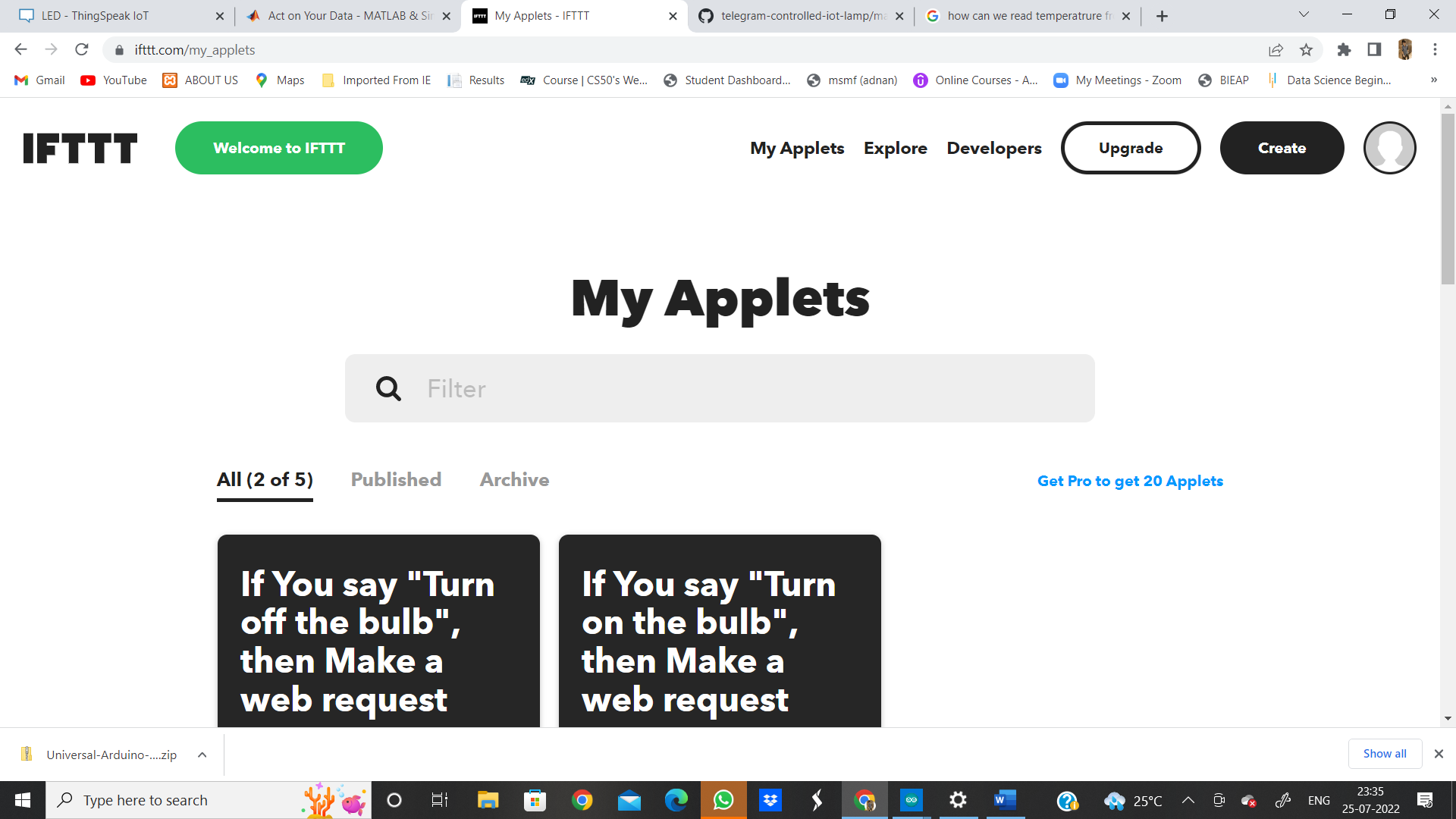
NOW THE DASH BOARD LOOKS SOME THING LIKE THIS.



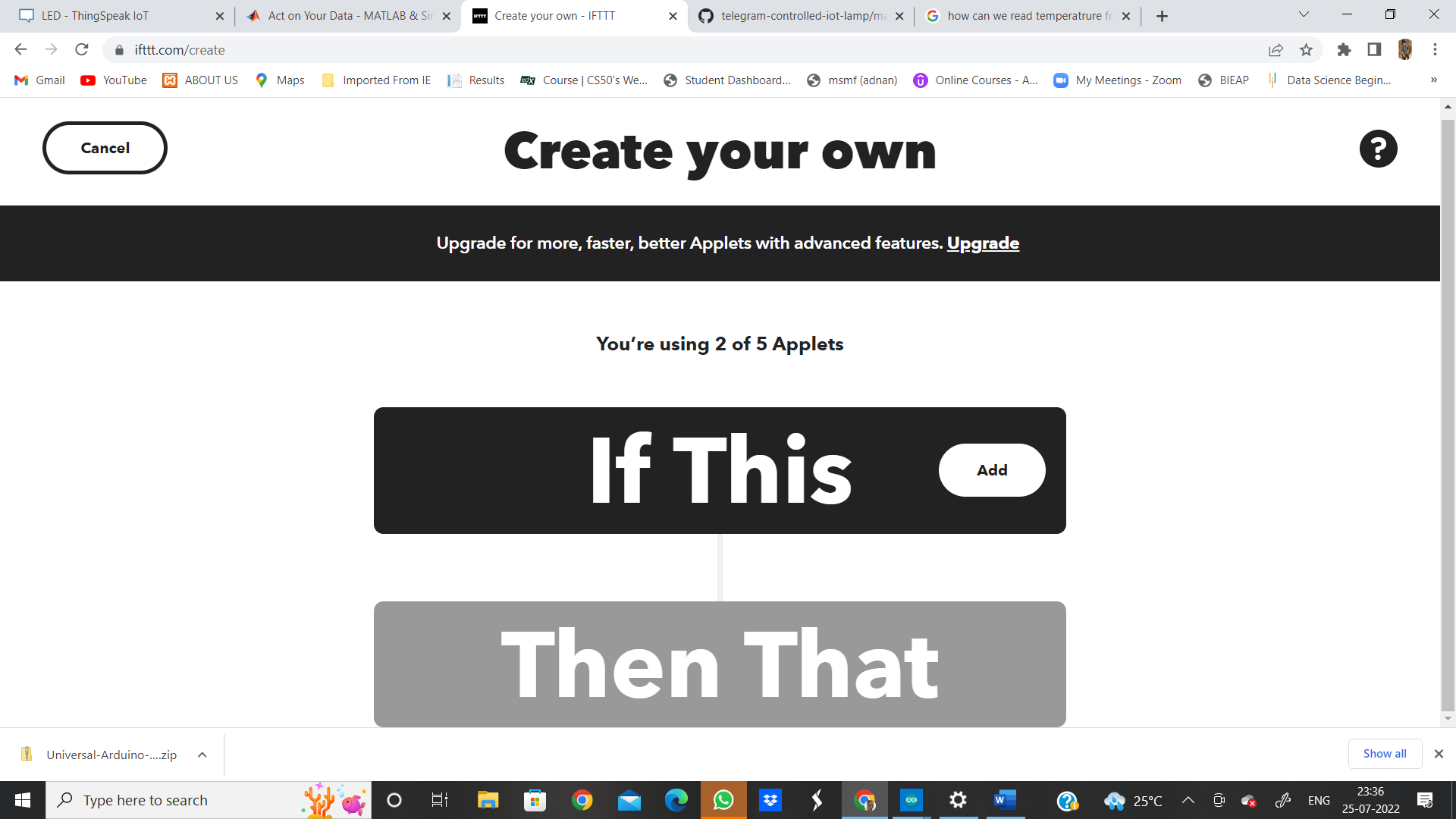
**STEP 5:**

NOW OPEN “IFTTT – IF THIS THEN THAT ” AND SIGNUP WITH GOOGLE MAKE SURE U HAVE SAME ID ON MOBILE PHONE.

CLICK ON CREATE .

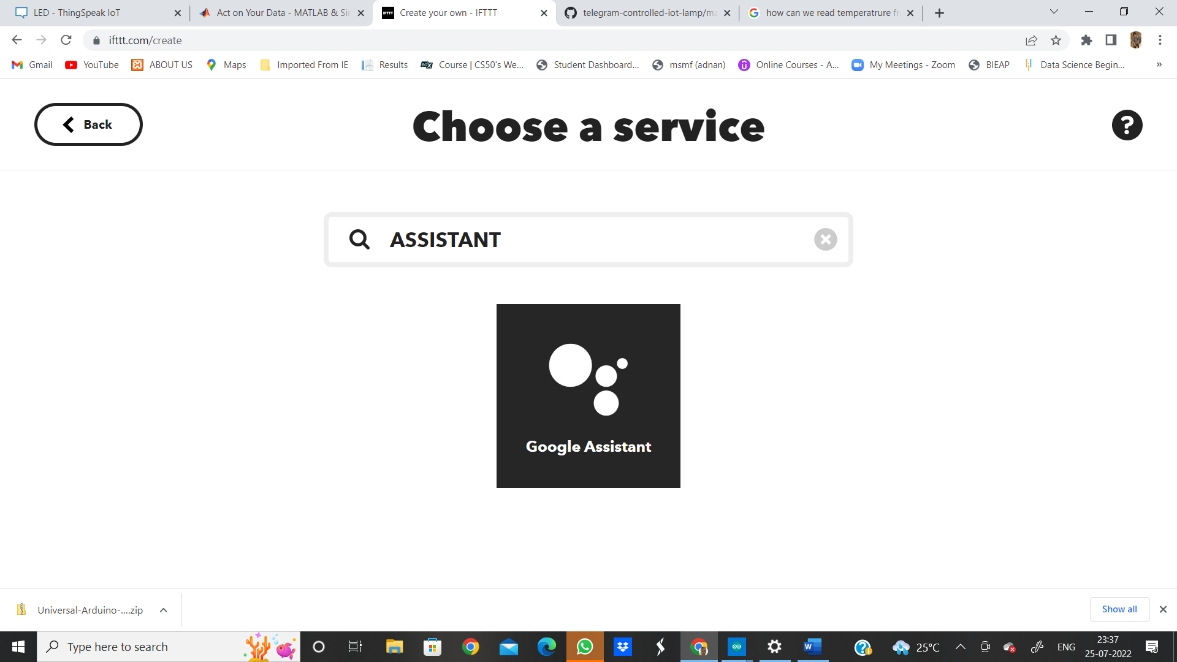


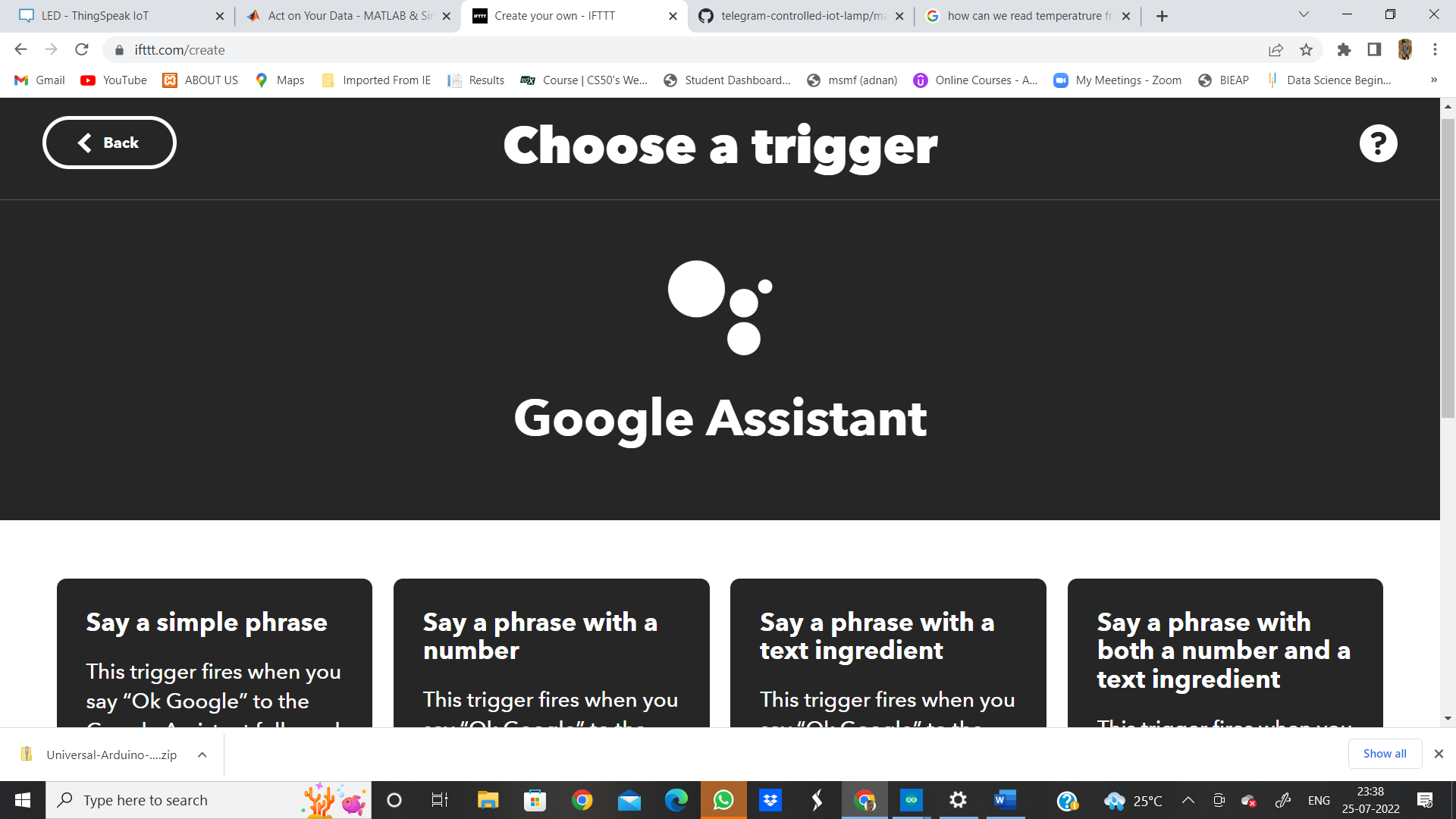
**STEP 6:**



**STEP 7:**

CLICK ON “ADD” WHICH IS PRESENT BESIDES “IF THIS ” AND SEARCH GOOGLE ASSISTANT.





**STEP 8:**

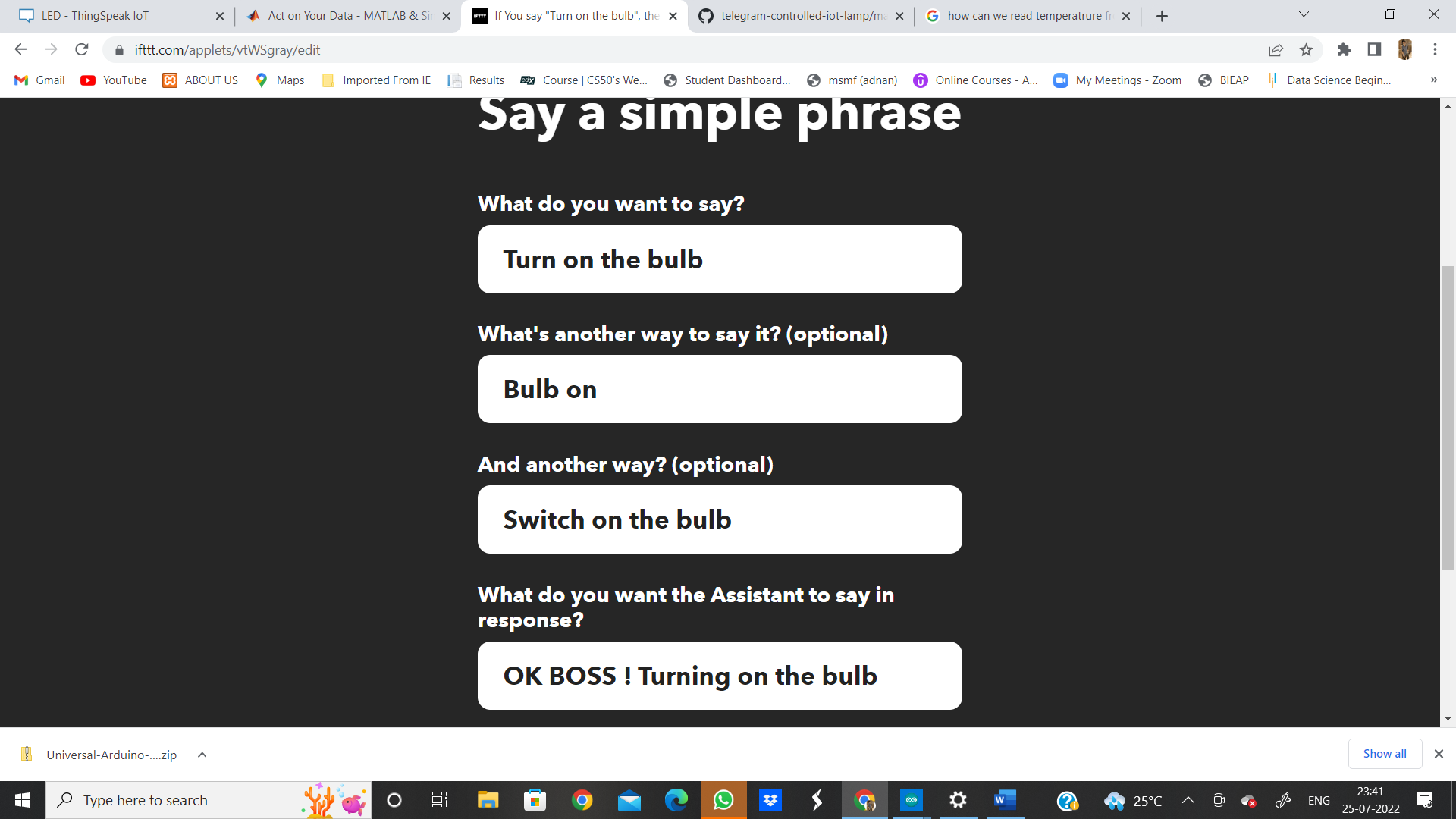
SELECT 1 OPTION “SAY A SIMPLE PHASE”:

ENTER THE PHASE WHICH IS TO BE INPUT

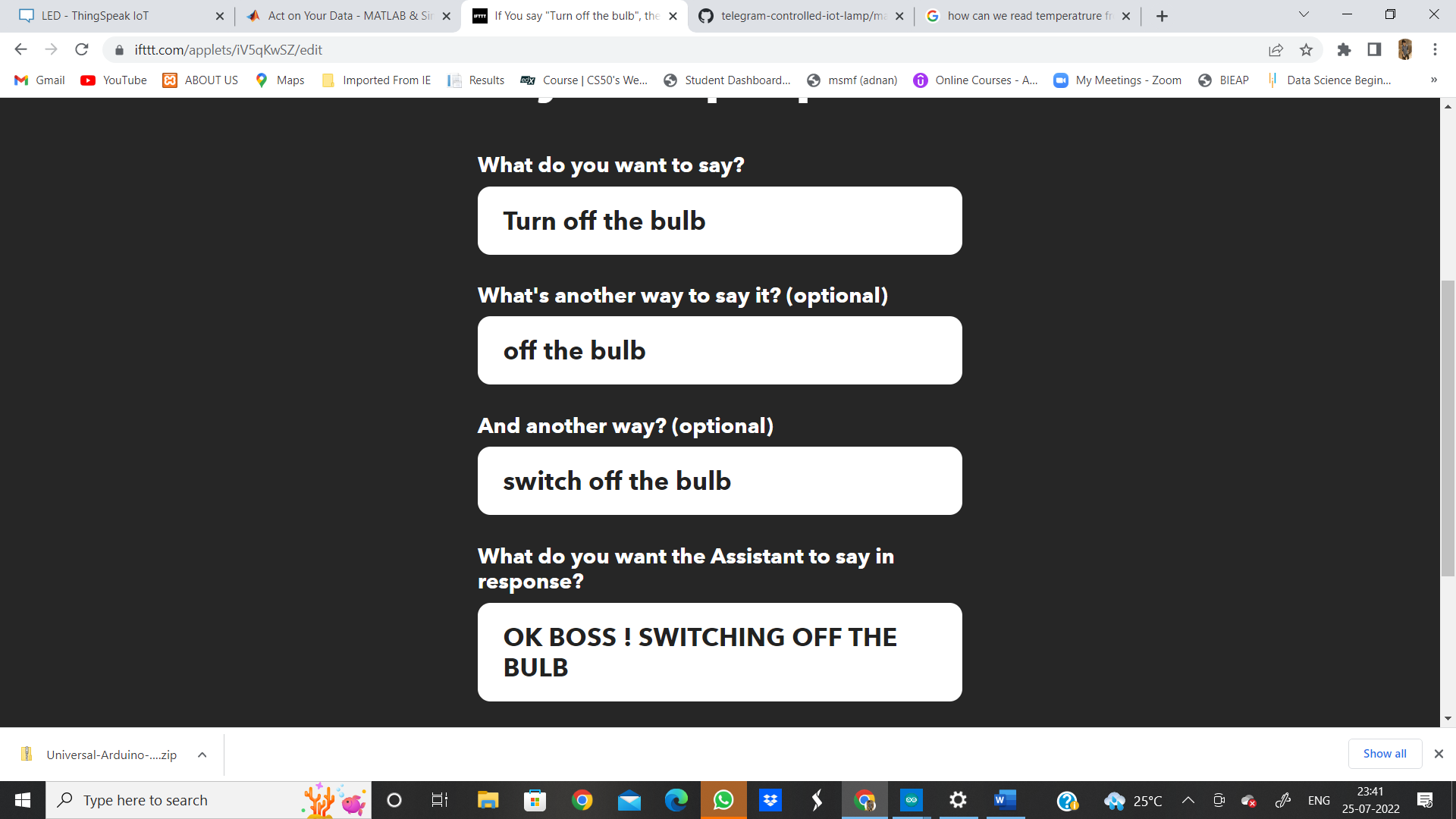
AND 2 OPTION PHASES WHICH ACT AS INPUT TO GOOGLE ASSITANT

AND 1 PHASE WHICH ASSITANT WILL READ OUT LOUD AS AN OUPUT (ACKNOLEDGMENT).

SAVE ALL THE DETAILS AS FOLLOWS.



SIMILARLY FOR TURN OFF THE BULB:



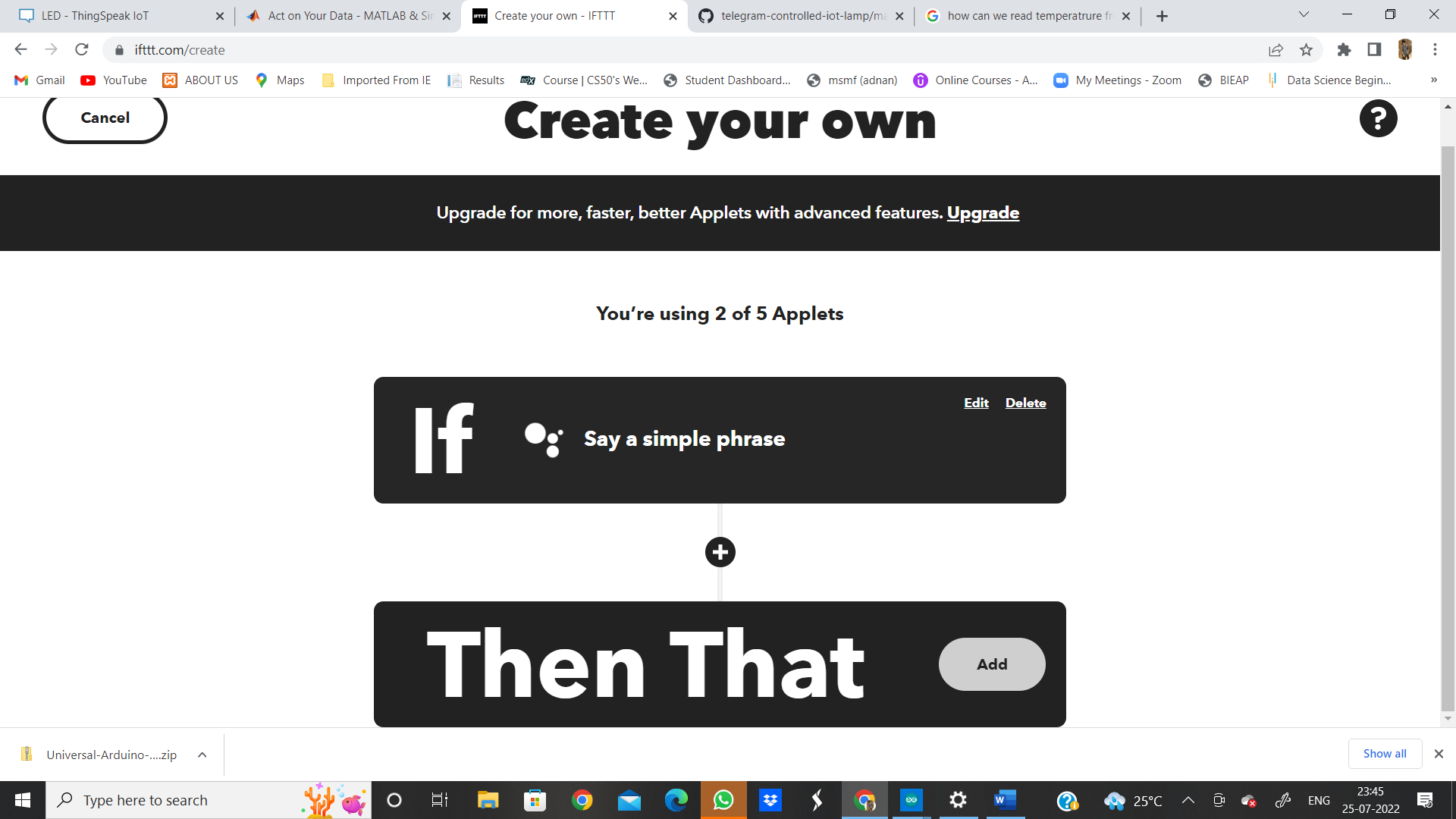
HENCE HERE OUT “IF THIS” CONDITON GETS COMPLETED .. NOW SAVE THIS AND COME TO “THEN THAT” CONDITION.

**STEP 9 :**

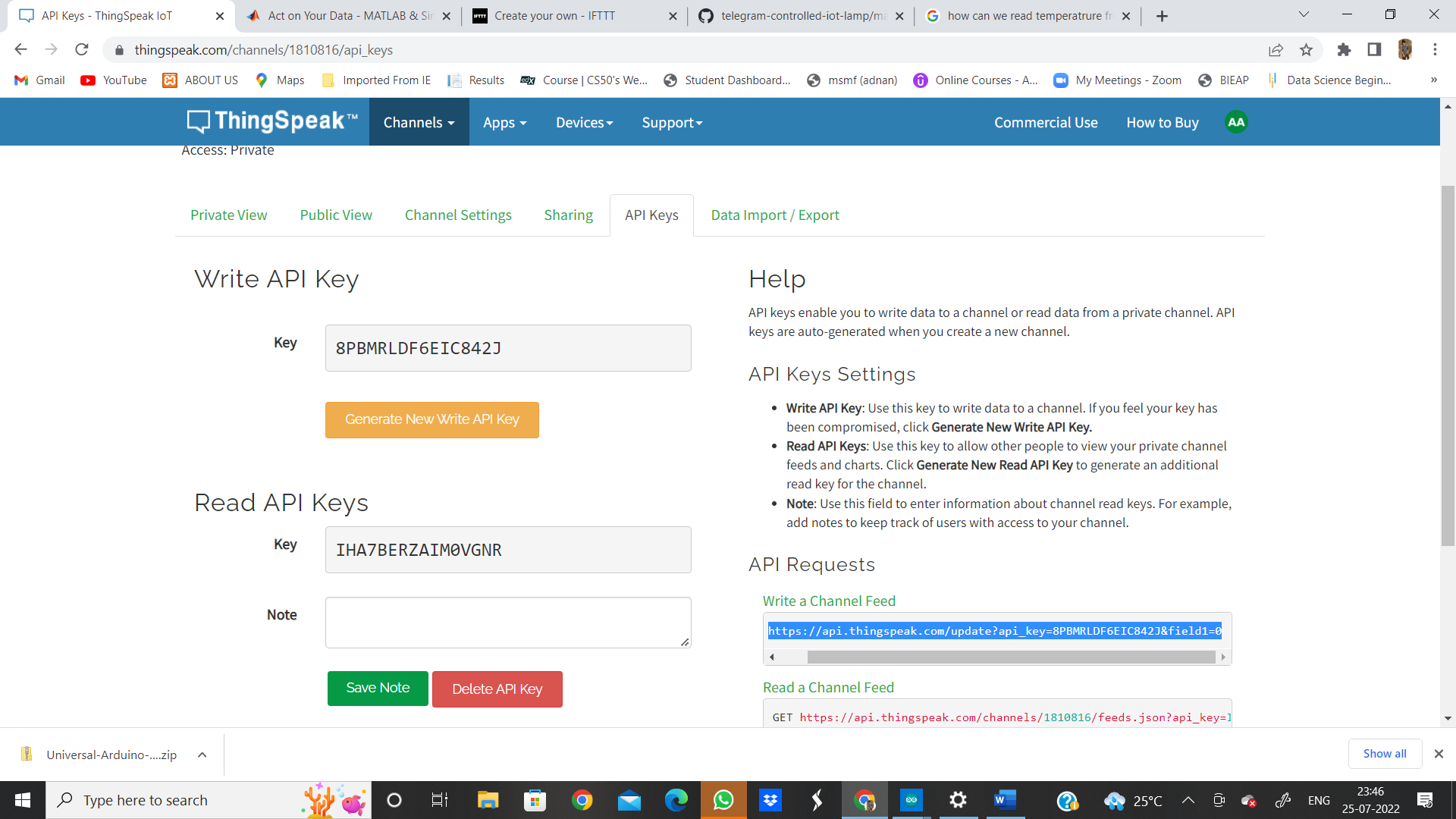
WHAT WE ARE GOING TO DO IS .. OPEN OUR CHANNEL IN THINGSPEAK AGAIN AND FOLLOW THESE STEPS.

THINGSPEAK > MY CHANNELS > LED CHANNEL > API KEYS > COPY THE WRITE API KEY.

LETS SEE HOW ITS DONE.



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



**STEP 10:**

NOW OPEN “IFTTT” AND IN “THEN THAT”

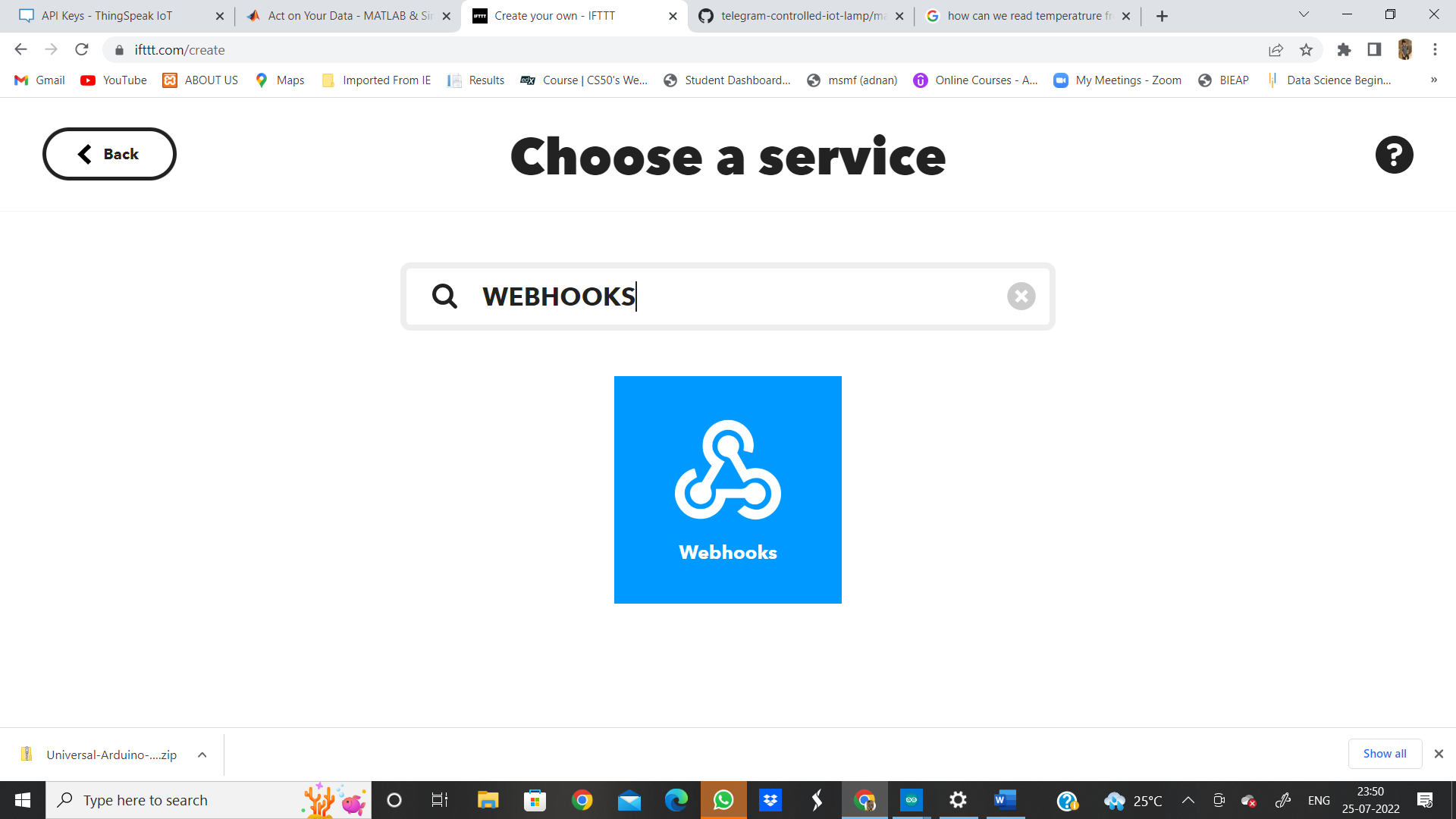
SEARCH “WEB-HOOKS”

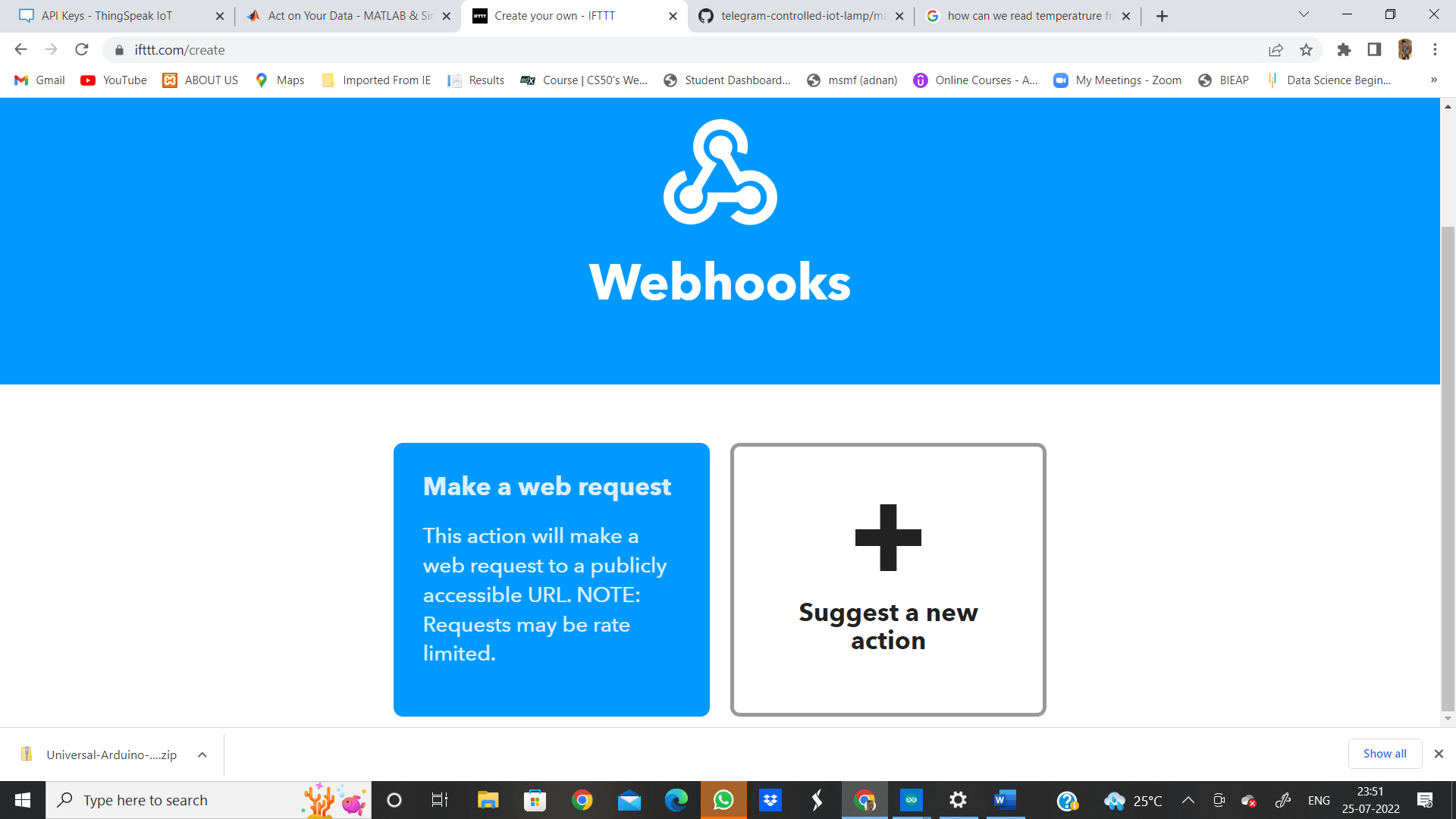
PASTE THIS URL COPIED FROM THINGSPEAK.

AS THE CONDITION WE KEPT FOR “TURN ON” LED IS “FIELD 1 > 0”—THEREFORE IN THE END OF URL.. CHANGE THE (FILED1=0 🡪 FIELD1=1) \*\*\* IMPORTANT STEP.

AND FOR “TURN OFF” NO NEED TO MAKE ANY CHANGES.

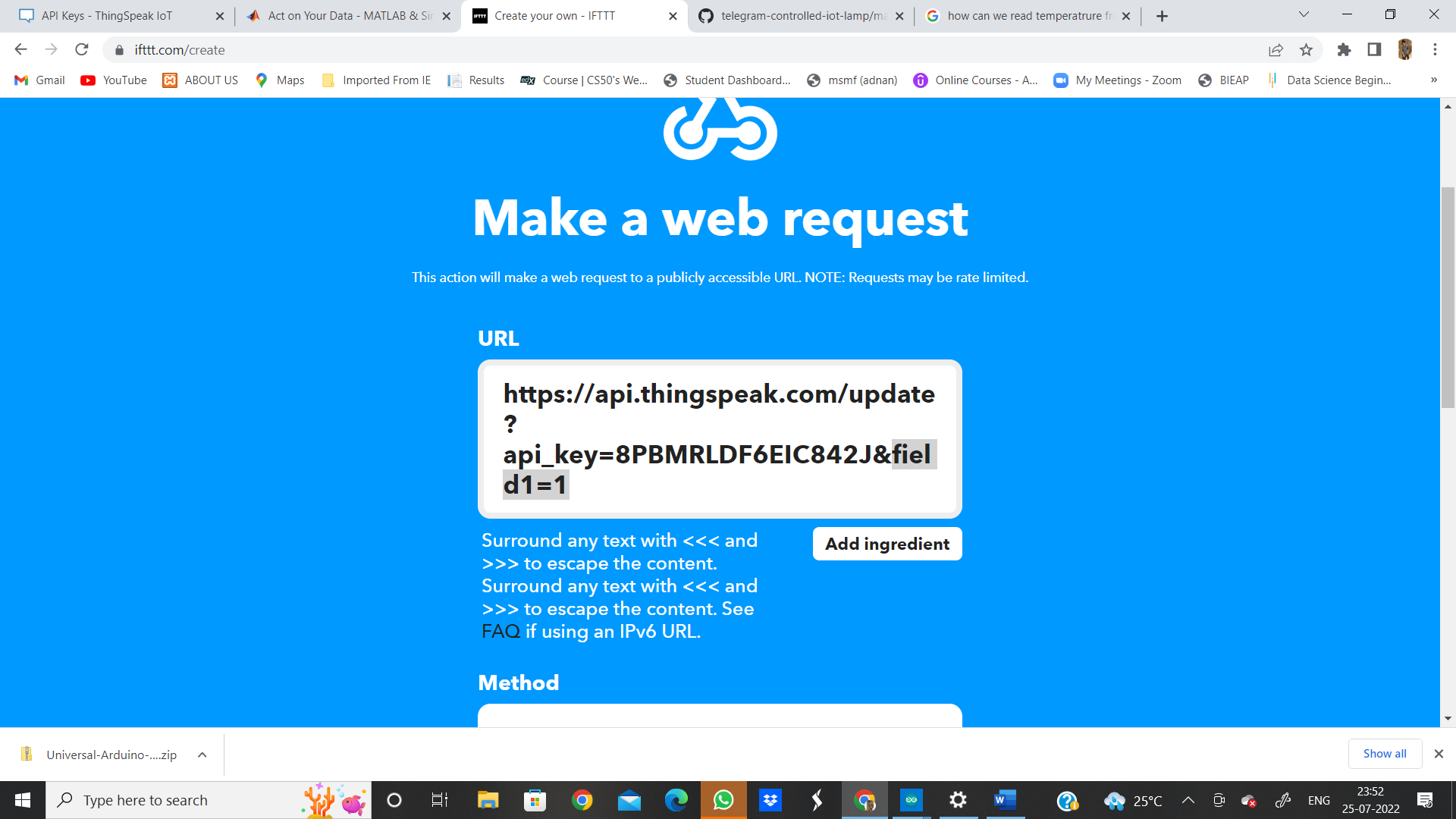
AND THEN SAVE BOTH THE APPLETS . AND THEN WE WILL APPROARCH TO HARDWARE PART.



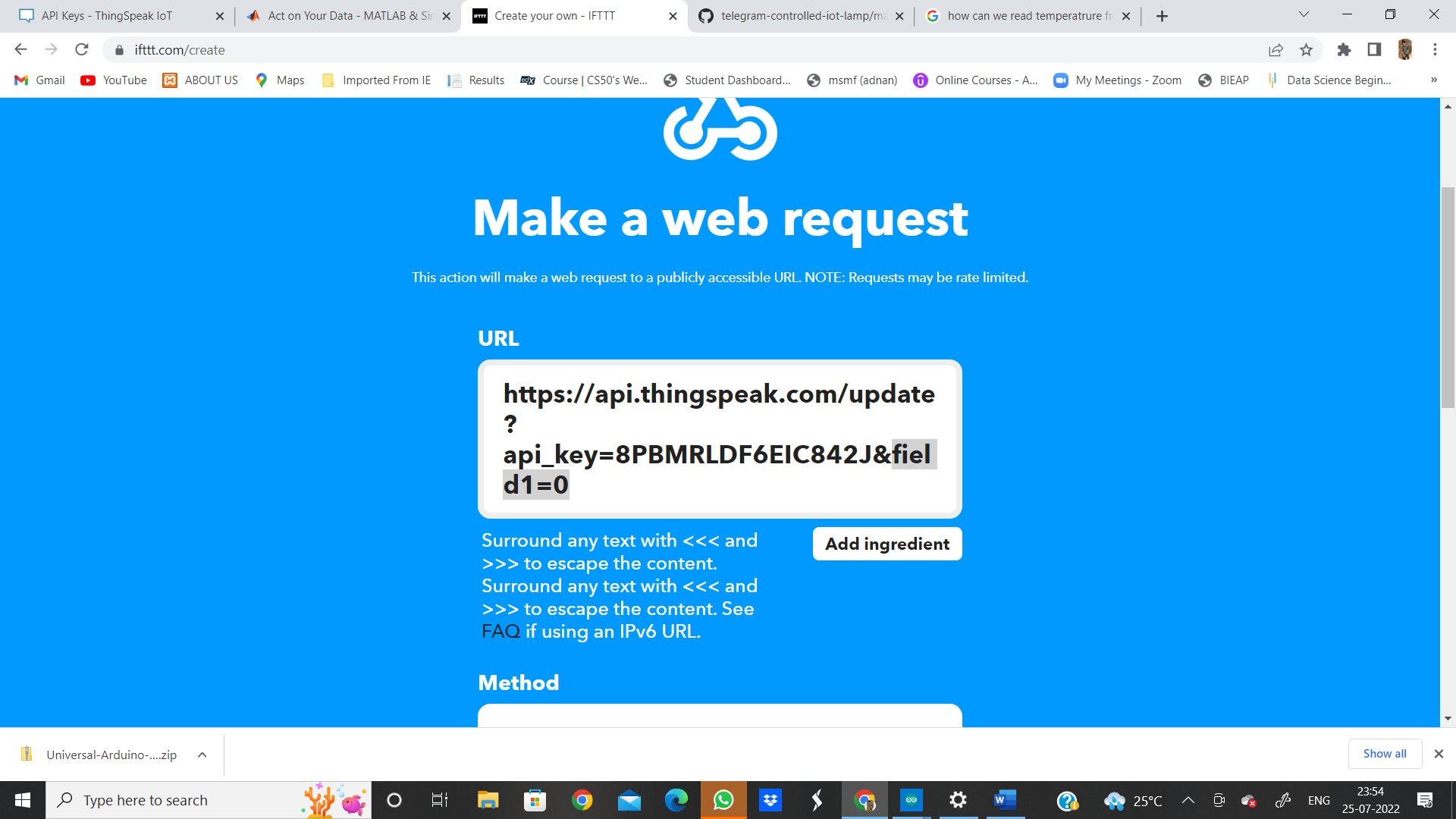


PASTE THE COPIED URL HERE:

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

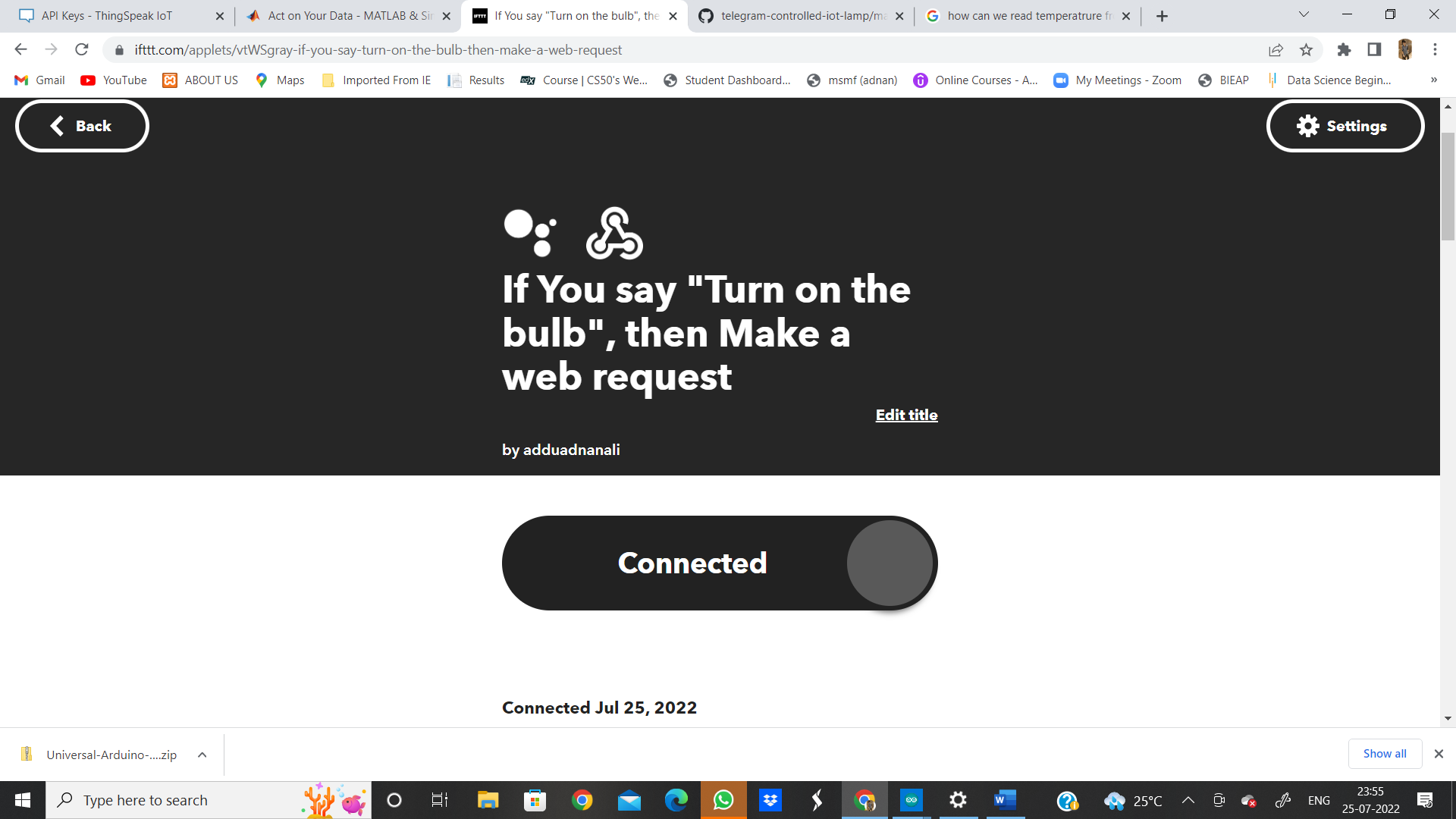


SIMILARLY TO TURN OFF – MAKE NO CHANGES.



AND MAKE NO OTHER CHANGES AND SAVE

CLICK ON UPDATE AND FINISH AND OUR “APPLET” WILL BE CONNECTED.



NOW COME TO HARDWARE PART—

HERE WILL USE SIMPLE NODEMCU ESP32 AND LED

CONNECTIONS

GND -GND (LED) AND POSITIVE END (LED) TO “D1”.

**STEP 11:**

**OPEN ARDUINO IDE AND TYPE THIS PROGRAM.**

**Using ESP32**

#include <WiFi.h>

#include <ThingSpeak.h>

int led=15;

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

char pass[] = "\*\*\*\*\*\*\*\*\*";

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);

}

}

AND THEN SAVE THE PROGRAM AND UPLOAD TO ESP32 MICROCONTROLLER.

AND THEN WE CAN OBSERVE THE OUTPUT.

OR

**OPEN ARDUINO IDE AND TYPE THIS PROGRAM.**

**Using ESP8266**

**#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: ");**

**}**

**}**

AND THEN SAVE THE PROGRAM AND UPLOAD TO ESP32 MICROCONTROLLER.

AND THEN WE CAN OBSERVE THE OUTPUT.