**PROJECT REPORT**

**on**

**IOT based Home automation with hindi voice commands using Google assistant and ESP 32**

**2022-2023**

****

**Submitted by:**

Mr. Shivam Goyal

Roll No.: 1018686

CSE-C-VIII-Sem

Session: 2022-2023

**DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY**

**GRAPHIC ERA HILL UNVERSITY, DEHRADUN**

**CERTIFICATE**

Certified that **Mr. Shivam Goyal** (**Roll No.- 1018686**) has developed mini project on **“ IOT based Home automation with hindi voice commands using Google assistant and ESP 32”** for the CS V Semester Mini Project Lab (PCS-504) in **Graphic Era Hill University**, **Dehradun**. The project carried out by Student is their own work as best of my knowledge.

**Date: 12 December 2020**

**(Mr. Samir Rana)**

**Project Co-ordinator**

**CC-CSE-C-V-Sem**

(CSE Department)

GEHU Dehradun

**Mr. Umang Garg**

**Project Guide**

(CSE Department)

GEHU Dehradun

**ACKNOWLEDGMENT**

I wish to thank my parents for their continuing support and encouragement. I also wish to thank them for providing us with the opportunity to reach this far in our studies.

I would like to thank particularly our project Co-ordinator **Mr. Samir Rana** and our Project Guide **Mr. Umang Garg** for his patience, support and encouragement throughout the completion of this project and having faith in us.

At last but not the least I greatly indebted to all other persons who directly or indirectly helped us during this work.

**Mr. Shivam Goyal**

**Roll No.- 1018686**

**CSE-C-V-Sem**

**Session: 2020-2021**

**GEHU, Dehradun**

**TABLE OF CONTENTS**

**CHAPTER NO. TITLE PAGE NO.**

1. LIST OF FIGURES 5
2. INTRODUCTION 6
3. PROJECT 7-21

3.1 Hardware Used

3.2 To build a home automation application, I used three different platforms

3.3 To use the above services we need to configure them.

3.4 Now, create dashboard at Adafruit. This dashboard is a user interface to control things remotely.

3.5 After following the above steps, provide a name to the dashboard and save it. We can see our dashboard as follows,

3.6 Now, create a feed (user interface) to control light On-Off. To create it, just click on the ‘+’ symbol and select the toggle feed shown below,

3.7 After selecting the toggle feed, a pop-up window appears as shown below

3.8 In the next step configure the feed which is shown below,

3.9 IFTTT (If This Then That)

3.10 Configure IFTTT

3.11 After account creation, click on My Applets and then select New Applet shown below,

3.12 After selecting a new applet, we get a new page in which we should click on to This as shown in the below image.

3.13 Then search for Google Assistant and select it.

3.14 Then search for Adafruit and select it.

3.15 After selecting Adafruit, choose action as shown below,

3.16 Click on Create Action.

3.17 Interfacing Diagram

3.18 Control Home’s Light using Google Assistant and NodeMCU

3.19 Library

3.20 Program

**LIST OF FIGURES**

**FIGURE Page No.**

Figure 3.3 Adafruit home page 7

Figure 3.4 Adafruit new dash board 8

Figure 3.5 Adafruit home dash board 8

Figure 3.6 Create new block 9

Figure 3.7 Choose feed 9

Figure 3.8.1 Configuration of feed 10

Figure 3.8.2 All feed Display 11

Figure 3.10 ifttt login 12

Figure 3.11 My applets 12

Figure 3.12 if this then that 13

Figure 3.13 Google assistant 13

Figure 3.14 Search Adafruit 16

Figure 3.15 Send data to adafruit 16

Figure 3.16 Create action 17

Figure 3.17 Interfacing Datagram 17

**2.INTRODUCTION**

Google assistant is AI (Artificial Intelligence) based voice command service. Using voice, we can interact with the google assistant and it can search on the internet, schedule events, set alarms, control appliances, etc.

This service is available on smartphones and Google Home devices.

We can control smart home devices including lights, switches, fans, and thermostats using our Google Assistant.

We will build an application that can control home appliances. Here, we will control a small fan module using Google Assistant service.

This application includes Google assistant along with Adafruit server and IFTTT service.

**3.PROJECT**

# **3.1 Hardware Used**

* NodeMCU – 32-bit ESP8266 development board with Wi-Fi SoC.
* Relay module
* One fan module
* Jump wires
* 9v battery
* USB cable

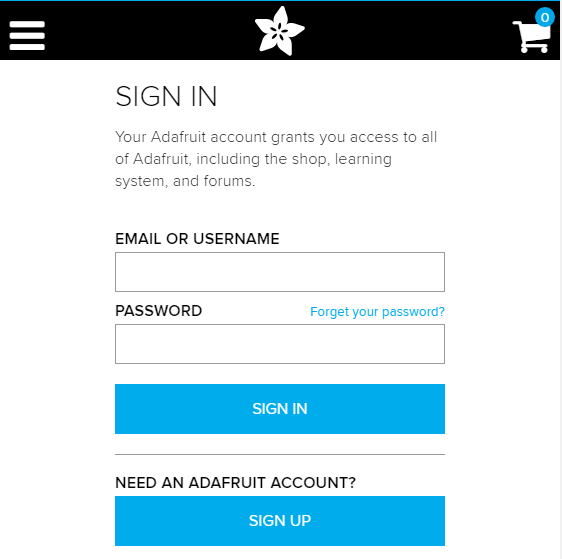
3.2 To build a home automation application, I used three different platforms

* Google Assistant
* Adafruit
* IFTTT

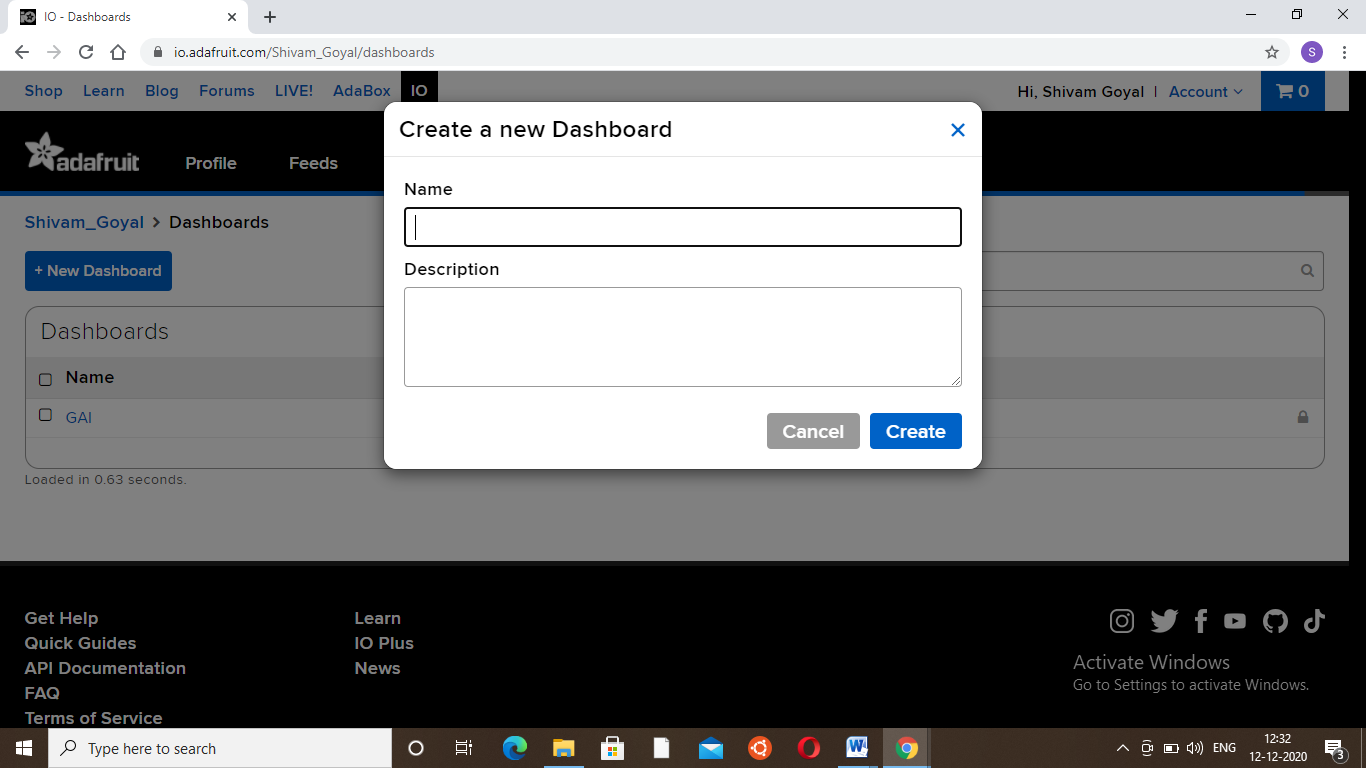
3.3 To use the above services we need to configure them.

# **Adafruit**

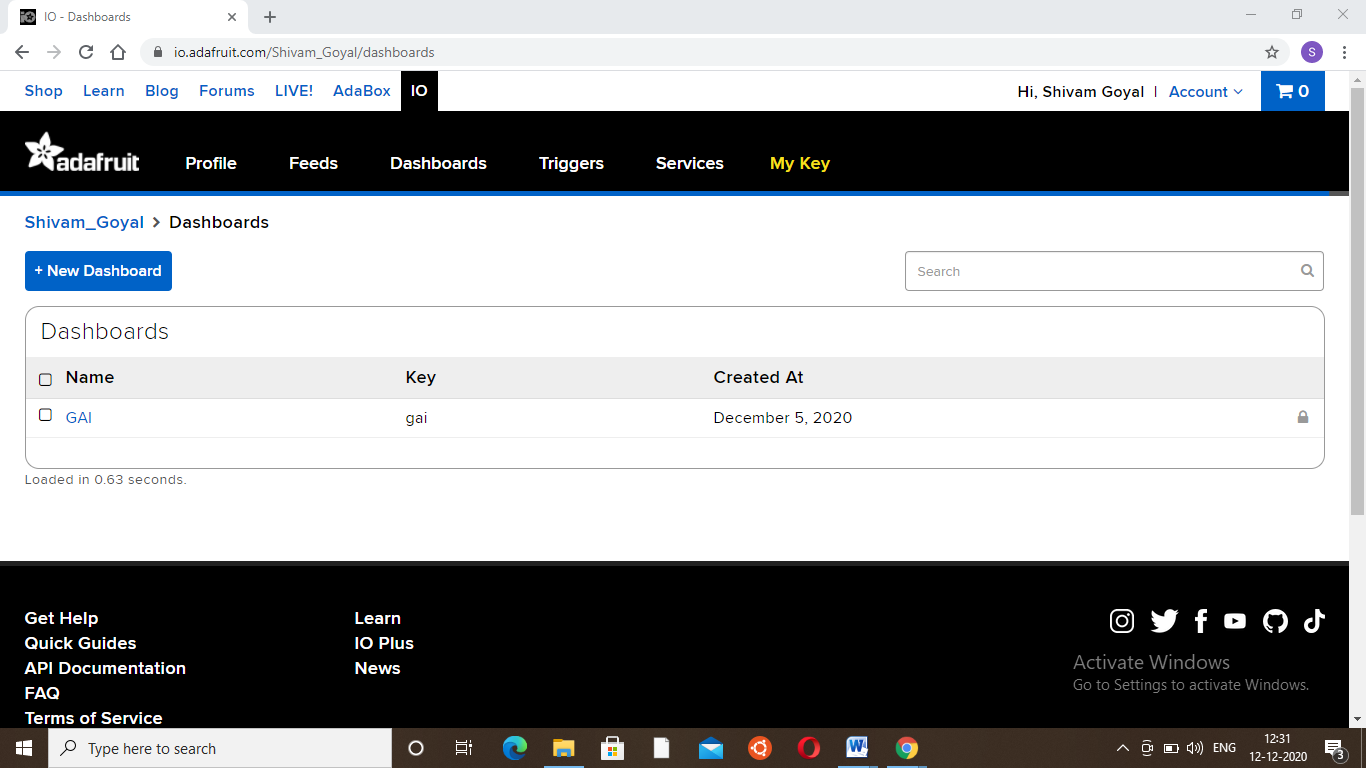
First, created an account at [www.Adafruit.io](http://www.adafruit.io/)



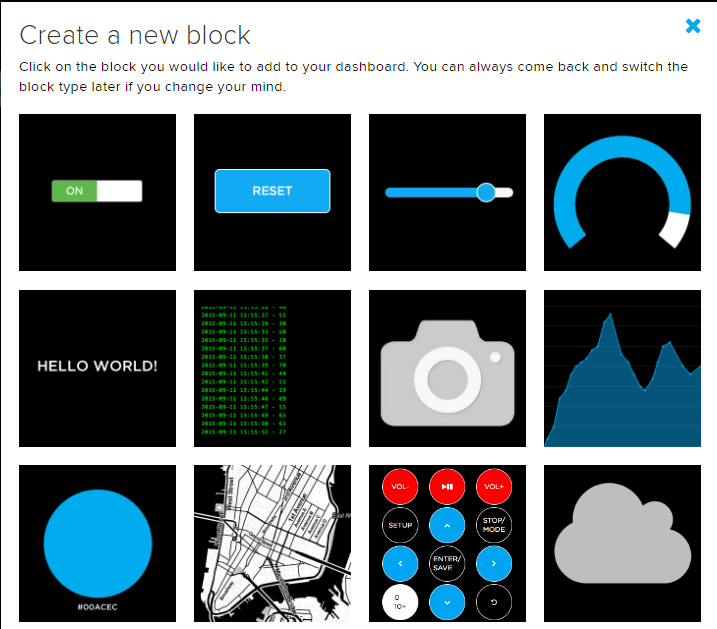
3.4 Now, create dashboard at Adafruit. This dashboard is a user interface to control things remotely.



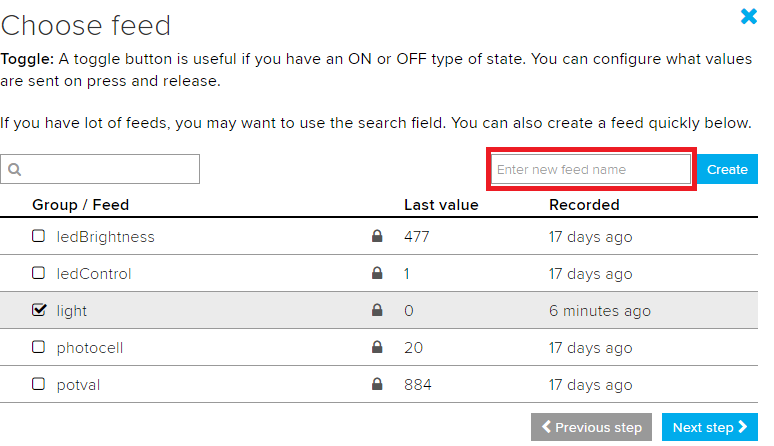
3.5 After following the above steps, provide a name to the dashboard and save it. We can see our dashboard as follows,



3.6 Now, create a feed (user interface) to control light On-Off. To create it, just click on the **‘+’**symbol and select the toggle feed shown below,

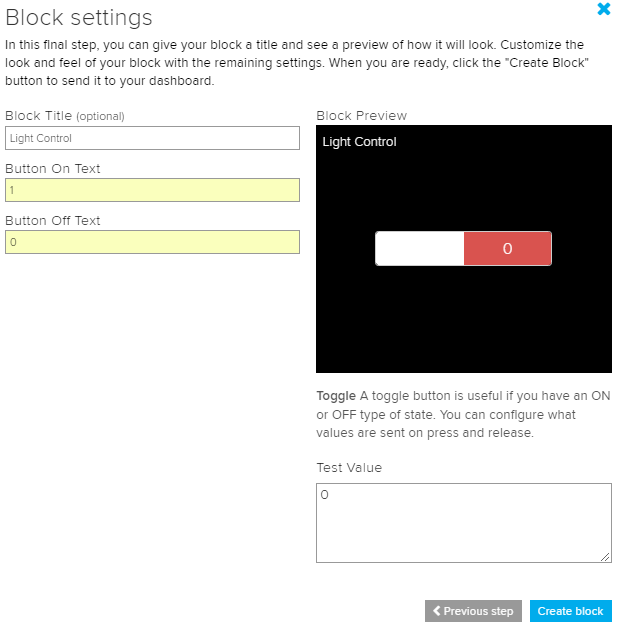


3.7 After selecting the toggle feed, a pop-up window appears as shown below.

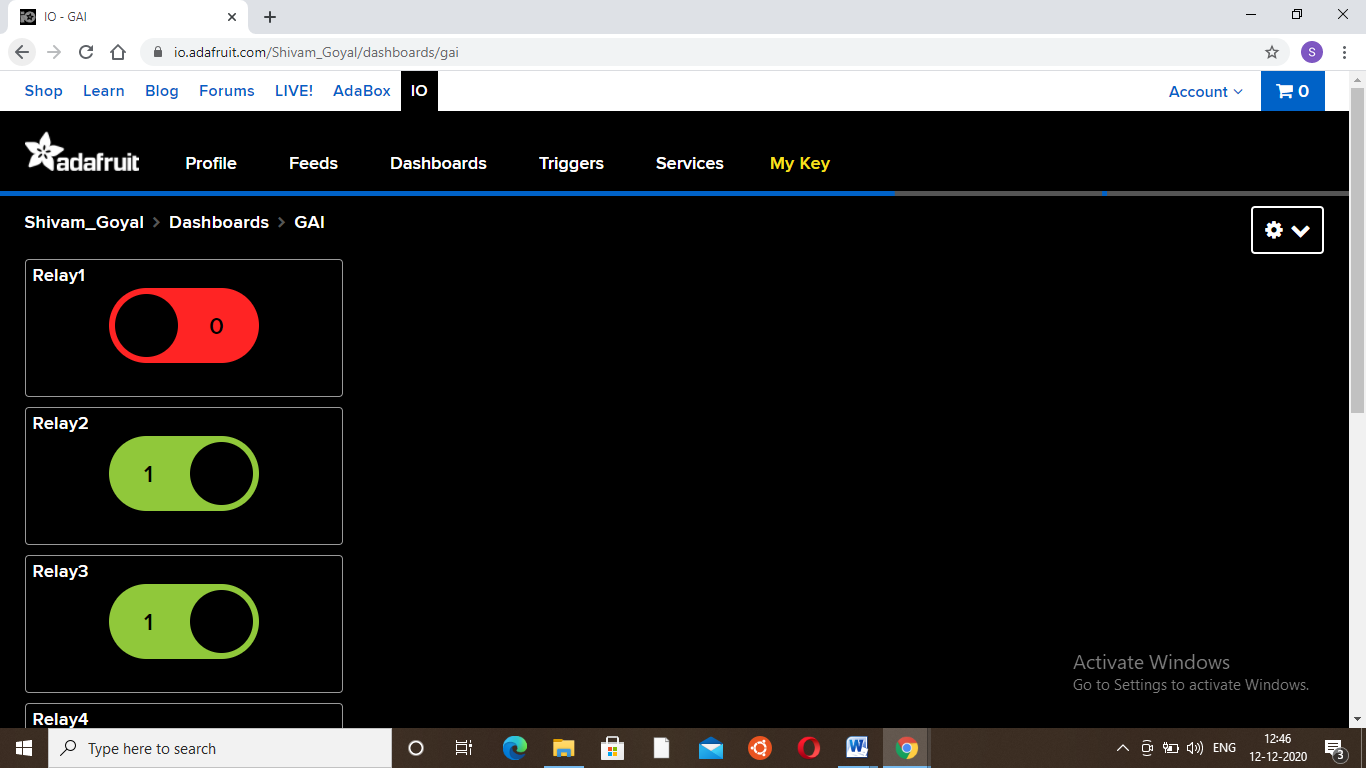


Enter the name of our feed (shown in the red box) and create it. After creation, select the created feed (here mine is **light**) and then click on the **Next step.**

3.8 In the next step configure the feed which is shown below,



Here, I used **0**(OFF) and **1**(ON) text for button and then click on create. This will create a toggle button on your dashboard which can be used to control things remotely.



 Note : I have changes my feed name to relay 1 and so on as they are easy to tackle.

Now, my dashboard is ready for IoT applications like home automation.

**3.9 IFTTT**(If This Then That)

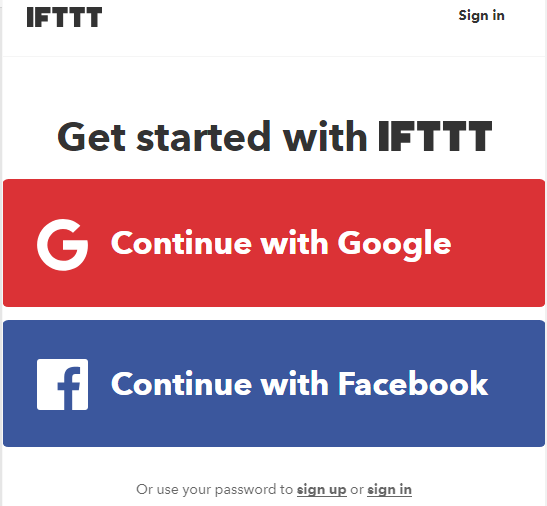
If This Then That, also known as IFTTT is a free web-based service to create chains of simple conditional statements, called applets. An applet is triggered by changes that occur within other web services such as Gmail, Facebook, Telegram, Instagram, or Pinterest.

For example, an applet may send an e-mail message if the user tweets using a hashtag or copy a photo on Facebook to a user's archive if someone tags a user in a photo.

Here, I used IFTTT to use google assistant service and Adafruit service in the chain. So, when I use google assistant to control the light of my home by saying Ok Google, turn the light ON or OFF. Then IFTTT interprets the message and can send it to Adafruit’s dashboard as an understandable command to the created feed.

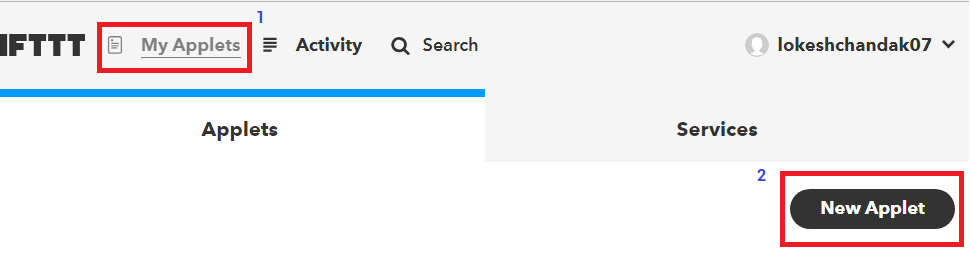
\**3.10 Configure IFTTT**

The first step is creating an account on IFTTT.​​​​​

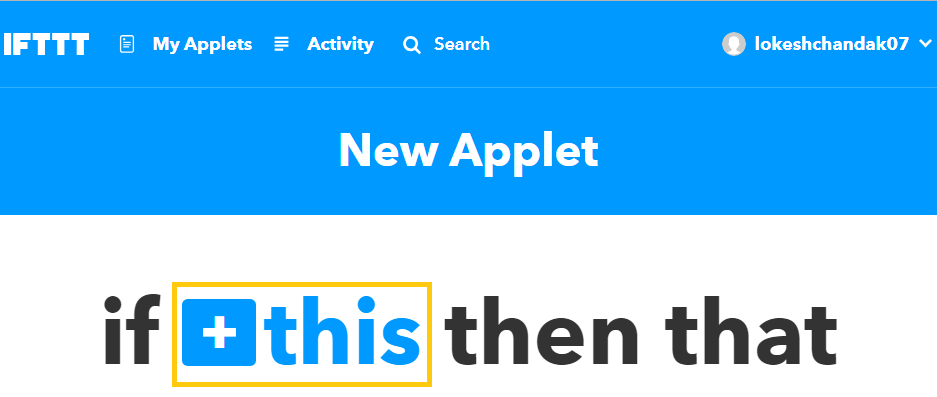


**Note:**Create an account on IFTTT byusing the same e-mail id which you have used for Adafruit.

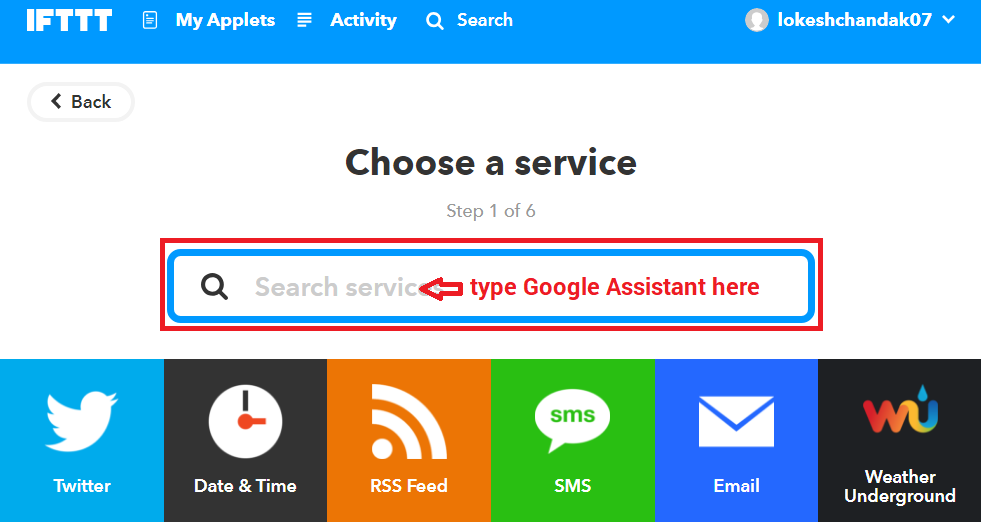
3.11 After account creation, click on **My Applets** and then select **New Applet** shown below,



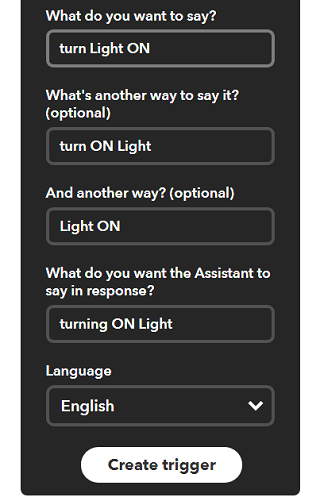
3.12 After selecting a new applet, we get a new page in which we should click on to **This**as shown in the below image.



3.13 Then search for **Google Assistant**and select it.

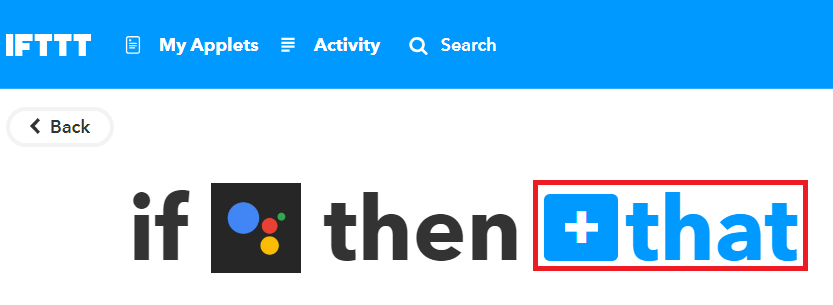


​​​​​​​Now, enter voice phrases which we will use as a command for google assistant.​​​​​​​​​​​​

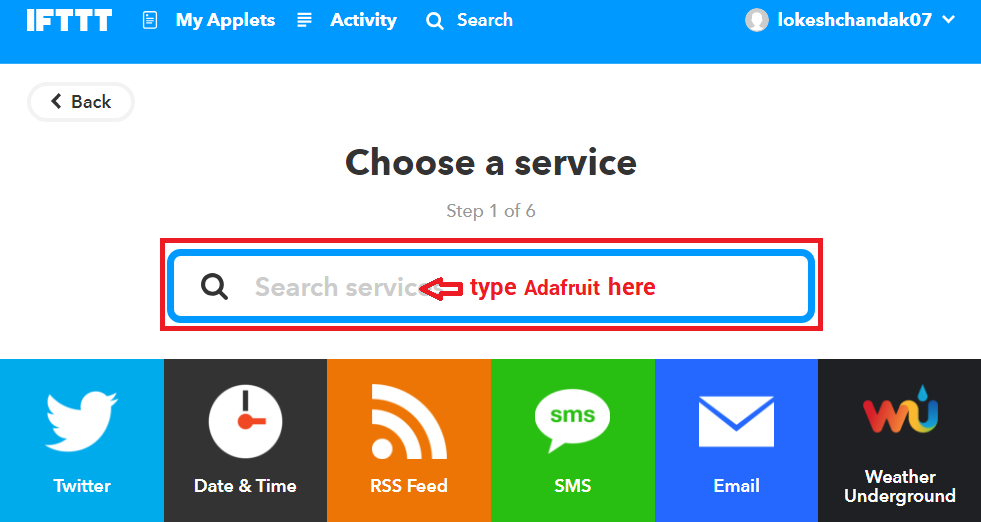


We can enter any phrase as per our application. As you can see, the phrases entered in the above fields is for making **Light ON.**For making **Light OFF**, we have to create another applet with different phrases.

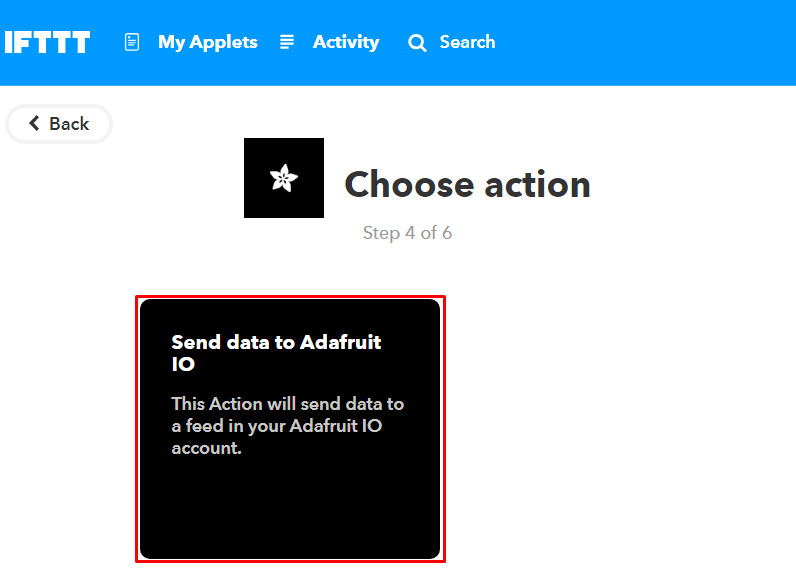
Now, we get another page on which we have to click on **that**option which is used to connect Google Assistant with Adafruit.​​​​​​​​​​​​​​



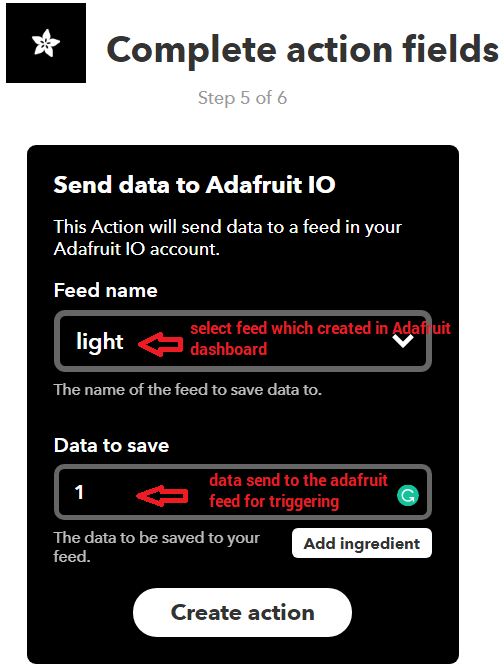
3.14 Then search for **Adafruit**and select it.



3.15 After selecting Adafruit, choose action as shown below,

​​​​​​​

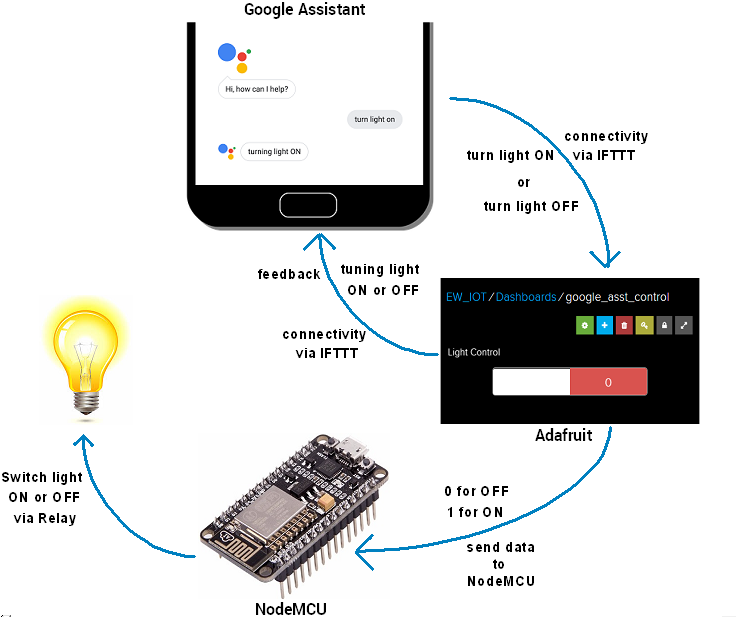
Now enter what data we need to send to which feed of Adafruit dashboard.​​​​​​​​​​​​​​



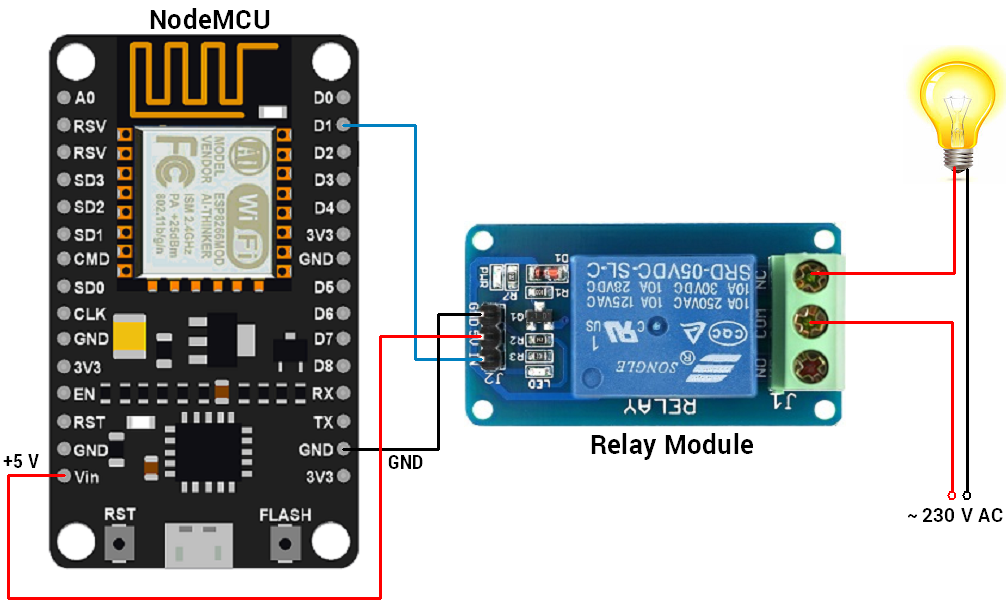
3.16 Click on **Create Action.**

So, when I use Google Assistant on my mobile and give a voice command as “Ok Google, Turn LED ON”, the applet created in IFTTT receives this command and will send data ‘1’ to the Adafruit feed. This will trigger the event on the Adafruit dashboard which is continuously monitored by the microcontroller (here NodeMCU). This microcontroller will take action as per the data change on the Adafruit dashboard.

​​​​​​​​​​​​



# **Interfacing Diagram**



**3.18 Library**

Here, I used the Adafruit MQTT library for receiving data from the Adafruit server. To install this library, select option **Sketch -> Include Library -> Manage Libraries.**

In that library, search for Adafruit MQTT and installed it.

**3.19 Control Home’s Light using Google Assistant and NodeMCU**

I build an IoT based home automation application in which I control the 100 W bulb at remotely using AI-based Google Assistant.

Here, I used NodeMCU to read data from the Adafruit server and act accordingly. 100 W bulb connected to NodeMCU via relay for controlling its voice command using google assistant.

# **3.20 Program**

#include <ESP8266WiFi.h>

#include "Adafruit\_MQTT.h"

#include "Adafruit\_MQTT\_Client.h"

#define Relay1 D1

#define WLAN\_SSID "Enter SSID NAME" // Your SSID

#define WLAN\_PASS "SSID Password" // Your password

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Adafruit.io Setup \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#define AIO\_SERVER "io.adafruit.com"

#define AIO\_SERVERPORT 1883 // use 8883 for SSL

#define AIO\_USERNAME "Enter your UserName" // Replace it with your username

#define AIO\_KEY "Enter you AIO\_KEY" // Replace with your Project Auth Key

/\*\*\*\*\*\*\*\*\*\*\*\* Global State (you don't need to change this!) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Create an ESP8266 WiFiClient class to connect to the MQTT server.

WiFiClient client;

// or... use WiFiFlientSecure for SSL

//WiFiClientSecure client;

// Setup the MQTT client class by passing in the WiFi client and MQTT server and login details.

Adafruit\_MQTT\_Client mqtt(&client, AIO\_SERVER, AIO\_SERVERPORT, AIO\_USERNAME, AIO\_KEY);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Feeds \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Setup a feed called 'onoff' for subscribing to changes.

Adafruit\_MQTT\_Subscribe Light1 = Adafruit\_MQTT\_Subscribe(&mqtt, AIO\_USERNAME"/feeds/Enter your Feed Name"); // FeedName

void MQTT\_connect();

void setup() {

Serial.begin(115200);

pinMode(Relay1, OUTPUT);

// Connect to WiFi access point.

Serial.println(); Serial.println();

Serial.print("Connecting to ");

Serial.println(WLAN\_SSID);

WiFi.begin(WLAN\_SSID, WLAN\_PASS);

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

delay(500);

Serial.print(".");

}

Serial.println();

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

// Setup MQTT subscription for onoff feed.

mqtt.subscribe(&Light1);

}

void loop() {

MQTT\_connect();

Adafruit\_MQTT\_Subscribe \*subscription;

while ((subscription = mqtt.readSubscription(5000))) {

if (subscription == &Light1) {

Serial.print(F("Got: "));

Serial.println((char \*)Light1.lastread);

int Light1\_State = atoi((char \*)Light1.lastread);

digitalWrite(Relay1, !(Light1\_State));

}

}

}

void MQTT\_connect() {

int8\_t ret;

// Stop if already connected.

if (mqtt.connected()) {

return;

}

Serial.print("Connecting to MQTT... ");

uint8\_t retries = 3;

while ((ret = mqtt.connect()) != 0) { // connect will return 0 for connected

Serial.println(mqtt.connectErrorString(ret));

Serial.println("Retrying MQTT connection in 5 seconds...");

mqtt.disconnect();

delay(5000); // wait 5 seconds

retries--;

if (retries == 0) {

// basically die and wait for WDT to reset me

while (1);

}

}

Serial.println("MQTT Connected!");

}