

## Sprint- 1

<b>Team ID</b>	<b>PNT2022TMID11481</b>
<b>Project Title</b>	<b>Smart Farmer -IoT Enabled Smart Farming Application</b>
<b>Date</b>	<b>12.11.2022</b>

### 1. Introduction

The main aim of this project is to help farmers automate their farms by providing them with a Web App through which they can monitor the parameters of the field like Temperature, soil moisture, humidity and etc and control the equipment like water motor and other devices remotely via internet without their actual presence in the field

### 2. Problem Statement

Farmers are to be present at the farm for its maintenance irrespective of the weather conditions. They have to ensure that the crops are well watered and the farm status is monitored by them physically. Farmers have to stay in the field most of the time in order to get a good yield. In difficult times like in the presence of a pandemic also they have to work hard in their fields risking their lives to provide food for the country.

### 3. Proposed Solution

In order to improve the farmer's working conditions and make them easier, we introduce IoT services to him in which we use cloud services and the internet to enable farmers to continue his work remotely via the internet. He can monitor the field parameters and control the devices in farm

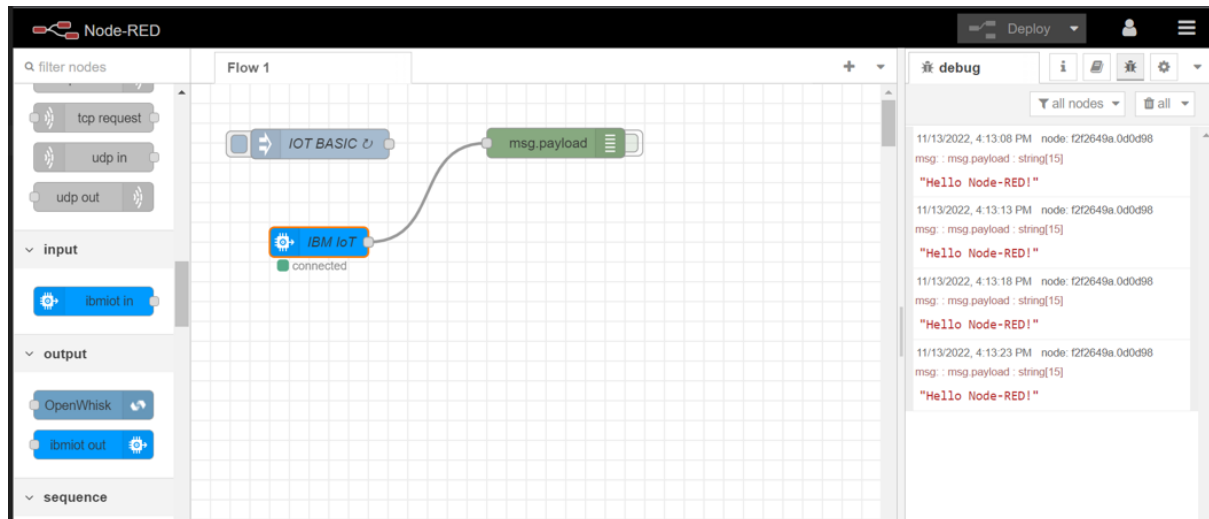
### Theoretical Analysis : Block Diagram

In order to implement the solution , the following approach as shown in the block diagram is used:

### Required Software Installation :

#### Node-Red

It is a flow-based development tool for visual programming developed originally by IBM for wiring together hardware devices, APIs and online services as part of the Internet of Things. Node-RED provides a web browser-based flow editor, which can be used to create JavaScript functions.



### Installation:

- First install npm/node.js
- Open cmd prompt
- Type => npm install node-red

### To run the application :

- Open cmd prompt
- Type=>node-red
- Then open <http://localhost:1880/> in browser

### Installation of IBM IoT and Dashboard nodes for Node-Red

In order to connect to IBM Watson IoT platform and create the Web App UI these nodes are required

1. IBM IoT node
2. Dashboard node
2. Dashboard node

### IBM Watson IoT Platform

A fully managed, cloud-hosted service with capabilities for device registration, connectivity, control, rapid visualisation and data storage. IBM Watson IoT Platform Is a managed, cloud hosted service designed to make it simple to derive value from your IoT devices.

### Steps to configure:

- Create an account in IBM cloud using your email ID
- Create IBM Watson Platform in services in your IBM cloud account
- Launch the IBM Watson IoT Platform
- Create a new device
- Give credentials like device type, device ID, Auth. Token
- Create API key and store API key and token elsewhere.

IBM Cloud

Search resources and products...

Catalog

Manage

Madhava narayana K's ...

Resource list /

Internet of Things Platform-1h

Active

Add tags

Details

Actions...

Manage

Plan

Connections

Let's get started with IBM Watson IoT Platform

Securely connect, control, and manage devices. Quickly build IoT applications that analyze data from the physical world.

Launch

Docs

Ready for the next level?

IBM Watson IoT Platform Journey

✓

Lite

The Lite service plan provides a lightweight development environment to get you started

○

Non-Production

The Non-Production service plan is a full-featured, fully-integrated offering that enables

○

Production

The Production service is a fully managed SaaS offering that enables you to manage and analyze

IBM Watson IoT Platform

910619106003@smartinternz.com

ID: 471218

Browse

Action

Device Types

Interfaces

Add Device +

Browse Devices

All Devices

Diagnose

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Search by Device ID

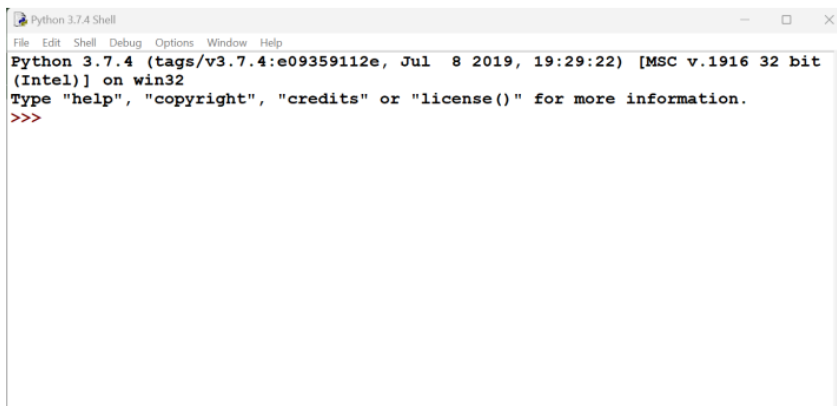
Device Simulator ☒

101

	Device ID	Status	Device Type	Class ID	Date Added
>	2005	Disconnected	akk	Device	1 Nov 2022 10:11

## Python IDE: 3.7.4

Version	Operating System	Description	MD5 Sum	File Size	GPG
<a href="#">Gzipped source tarball</a>	Source release		68111671e5b2db4aef7b9ab01bf0f9be	23017663	<a href="#">SIG</a>
<a href="#">XZ compressed source tarball</a>	Source release		d33e4aae66097051c2eca45ee3604803	17131432	<a href="#">SIG</a>
<a href="#">macOS 64-bit/32-bit installer</a>	macOS	for Mac OS X 10.6 and later	6428b4fa7583daff1a442cba8cee08e6	34898416	<a href="#">SIG</a>
<a href="#">macOS 64-bit installer</a>	macOS	for OS X 10.9 and later	5dd605c38217a45773bf5e4a936b241f	28082845	<a href="#">SIG</a>
<a href="#">Windows help file</a>	Windows		d63999573a2c06b2ac56cade6b47cd2	8131761	<a href="#">SIG</a>
<a href="#">Windows x86-64 embeddable zip file</a>	Windows	for AMD64/EM64T/x64	9b00c8cf6d9ec0b9abe83184a40729a2	7504391	<a href="#">SIG</a>
<a href="#">Windows x86-64 executable installer</a>	Windows	for AMD64/EM64T/x64	a702b4b0ad76debd3043a583e563400	26680368	<a href="#">SIG</a>
<a href="#">Windows x86-64 web-based installer</a>	Windows	for AMD64/EM64T/x64	28cb1c608bbd73ae8e53a3bd351b4bd2	1362904	<a href="#">SIG</a>
<a href="#">Windows x86 embeddable zip file</a>	Windows		9fab3b81f8841879fda94133574139d8	6741626	<a href="#">SIG</a>
<a href="#">Windows x86 executable installer</a>	Windows		33cc602942a54446a3d6451476394789	25663848	<a href="#">SIG</a>
<a href="#">Windows x86 web-based installer</a>	Windows		1b670cfa5d317df82c30983ea371d87c	1324608	<a href="#">SIG</a>



```
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 19:29:22) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

## Python Code:

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

#Provide your IBM Watson Device Credentials
organization = "47I2i8"
deviceType = "akk"
deviceId = "2005"
authMethod = "token"
authToken = "confidential"

# Initialize GPIO
```

```

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="switchon":
        print ("Switch is on")
    else :
        print ("Switch is off")

    #print(cmd)
try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,
"auth-method": authMethod, "auth-token": authToken}
    deviceCli = ibmiotf.device.Client(deviceOptions)
    #.....

except Exception as e:
    print("Caught exception connecting device: %s" % str(e))
    sys.exit()

# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type
"greeting" 10 times
deviceCli.connect()

while True:
    #Get Sensor Data from DHT11

    temp=random.randint(0,100)
    Humid=random.randint(0,100)
    SoilMoisture=random.randint(0,100)

    data = { 'temp' : temp, 'Humid': Humid, "SoilMoisture": SoilMoisture}

    #print data
    def myOnPublishCallback():
        print ("Published Temperature = %s C" % temp, "Humidity = %s %" % Humid,
"soilMoisture = %s %" % SoilMoisture, "to IBM Watson")

    success = deviceCli.publishEvent("IoT Sensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
    if not success:
        print("Not connected to IoT")
        time.sleep(1)
    deviceCli.commandCallback = myCommandCallback
# Disconnect the device and application from the cloud
deviceCli.disconnect()

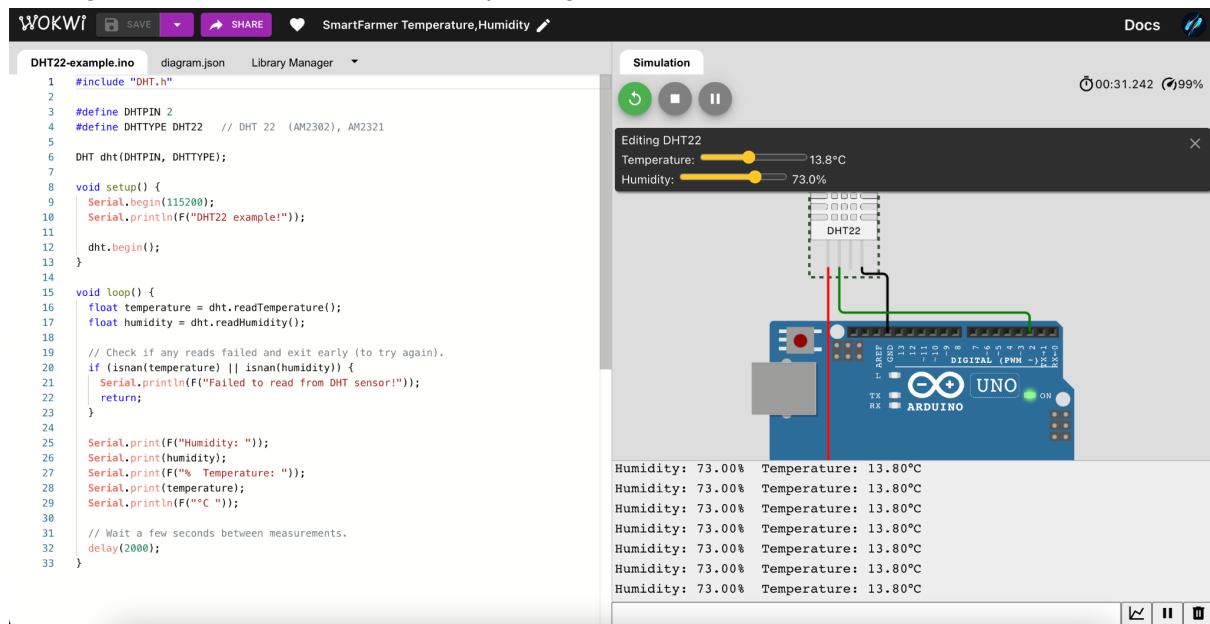
```

## To get the Temperature/Humidity & Soil Moisture (2 ways/method )

- 1) Sensor/Simulation ( Wokwi, TinkerCad)
- 2) OpenWeatherApi (Live data based on locational latitude and longitude)

### Sensor/Simulation Method:

#### Getting the Temperature and Humidity using the DHT22 in WOKWI



#### Code:

```
#include "DHT.h"
```

```
#define DHTPIN 2
```

```
#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321
```

```
DHT dht(DHTPIN, DHTTYPE);
```

```
void setup() {
```

```
  Serial.begin(115200);
```

```
  Serial.println(F("DHT22 example!"));
```

```
  dht.begin();
```

```
}
```

```

void loop() {
  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();

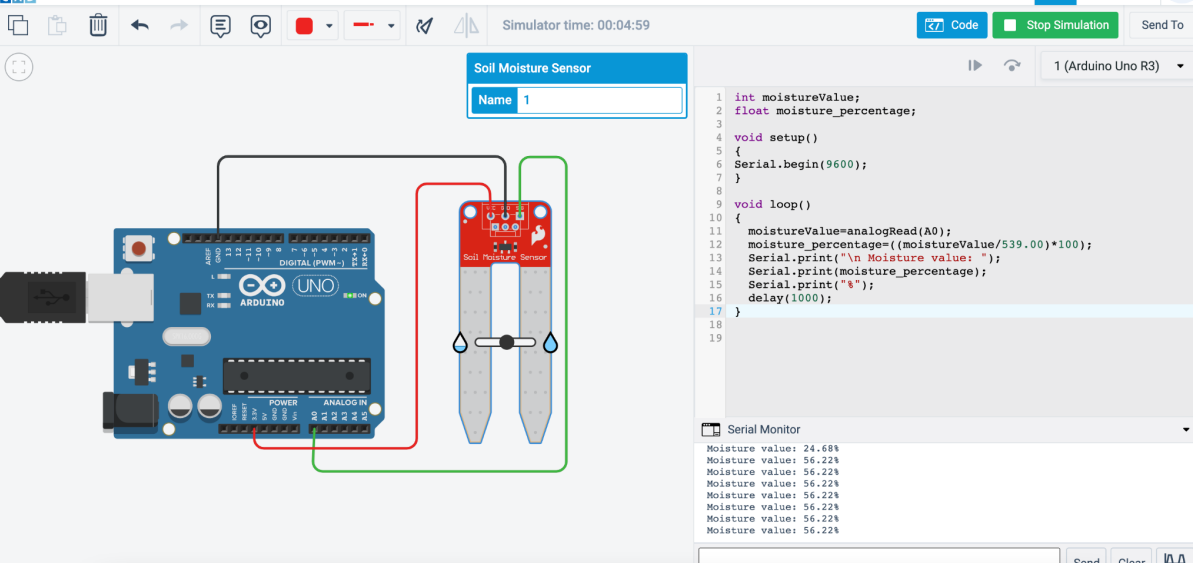
  // Check if any reads failed and exit early (to try again).
  if (isnan(temperature) || isnan(humidity)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }

  Serial.print(F("Humidity: "));
  Serial.print(humidity);
  Serial.print(F("% Temperature: "));
  Serial.print(temperature);
  Serial.println(F("°C "));

  // Wait a few seconds between measurements.
  delay(2000);
}

```

## Getting the Soil Moisture using Soil Moisture sensor in TINKERCAD



The screenshot shows the Tinkercad web interface with an Arduino Uno R3 connected to a Soil Moisture Sensor. The sensor is connected to the Arduino's A0 pin. The code in the Serial Monitor shows the sensor reading a moisture value of 24.68% and then 56.22%.

```

1 int moistureValue;
2 float moisture_percentage;
3
4 void setup()
5 {
6   Serial.begin(9600);
7 }
8
9 void loop()
10 {
11   moistureValue=analogRead(A0);
12   moisture_percentage=((moistureValue/539.00)*100);
13   Serial.print("\n Moisture value: ");
14   Serial.print(moisture_percentage);
15   Serial.print("%");
16   delay(1000);
17 }
18
19

```

Serial Monitor

```

Moisture value: 24.68%
Moisture value: 56.22%
Moisture value: 56.22%
Moisture value: 56.22%
Moisture value: 56.22%
Moisture value: 56.22%
Moisture value: 56.22%
Moisture value: 56.22%

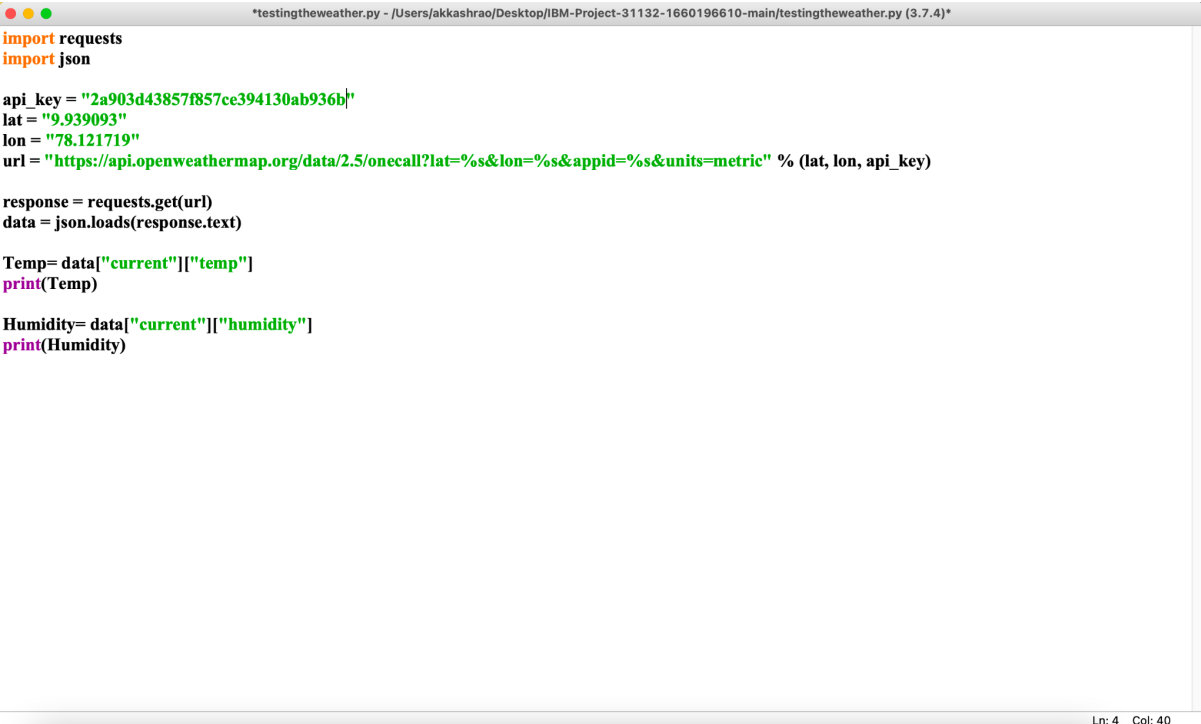
```

**Code:**

```
int moistureValue;
float moisture_percentage;

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  moistureValue=analogRead(A0);
  moisture_percentage=((moistureValue/539.00)*100);
  Serial.print("\n Moisture value: ");
  Serial.print(moisture_percentage);
  Serial.print("%");
  delay(1000);
}
```

**OpenWeatherApi method (Live data based on locational latitude and longitude)**

```
*testingtheweather.py - /Users/akkaashrao/Desktop/IBM-Project-31132-1660196610-main/testingtheweather.py (3.7.4)*

import requests
import json

api_key = "2a903d43857f857ce394130ab936b"
lat = "9.939093"
lon = "78.121719"
url = "https://api.openweathermap.org/data/2.5/onecall?lat=%s&lon=%s&appid=%s&units=metric" % (lat, lon, api_key)

response = requests.get(url)
data = json.loads(response.text)

Temp= data["current"]["temp"]
print(Temp)

Humidity= data["current"]["humidity"]
print(Humidity)
```

Ln: 4 Col: 40

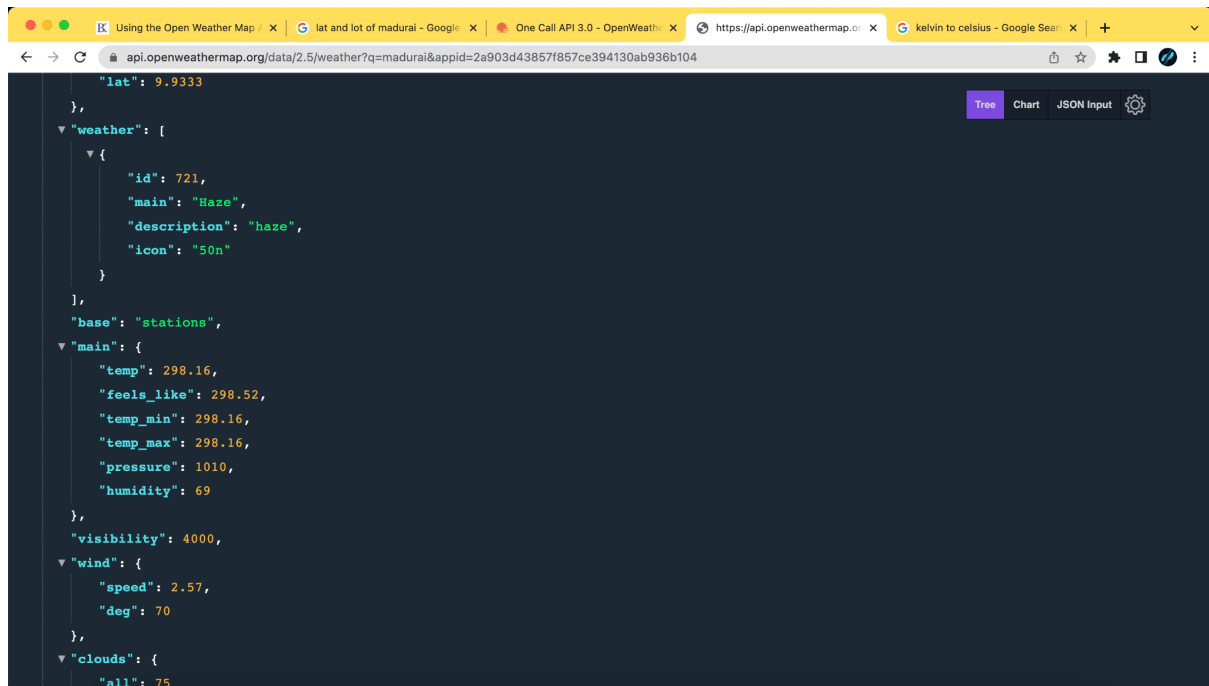


```
Python 3.7.4 Shell

Python 3.7.4 (v3.7.4:e09359112e, Jul 8 2019, 14:54:52)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: /Users/akkaashrao/Desktop/IBM-Project-31132-1660196610-main/testingtheweat
her.py
25.01
69
>>> |
```

Ln: 8 Col: 4

## Testing the Value:



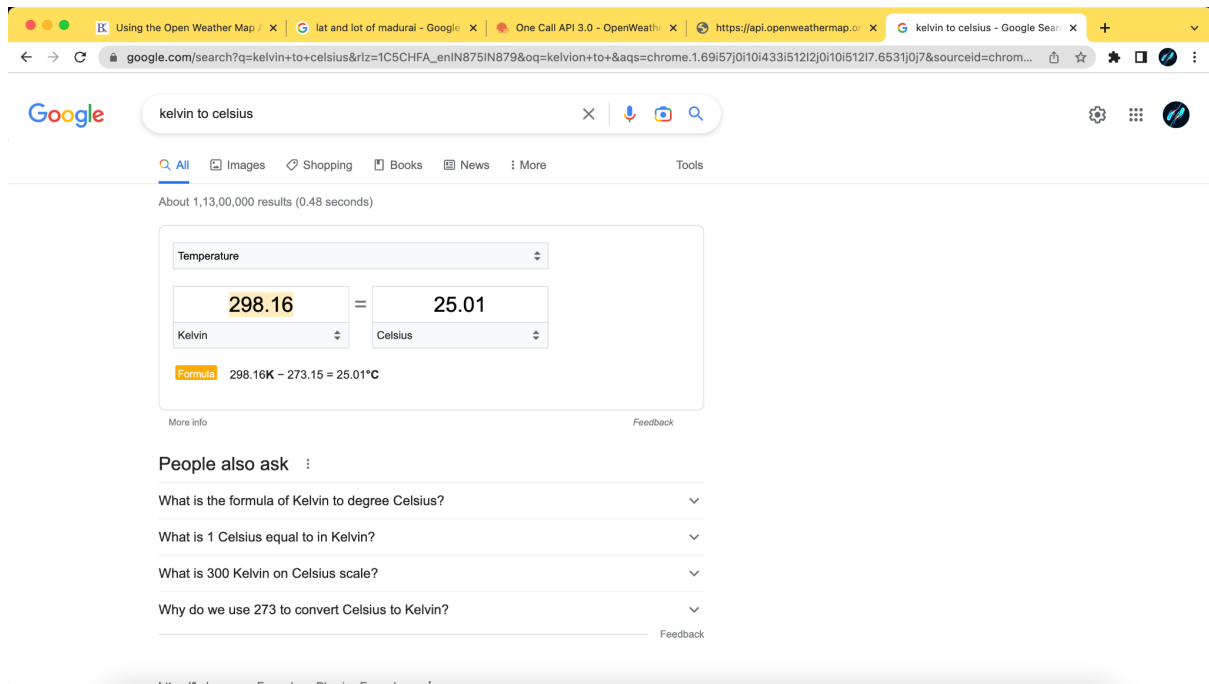
The screenshot shows a web browser displaying the JSON response from the OpenWeatherMap API. The URL in the address bar is `https://api.openweathermap.org/data/2.5/weather?q=madurai&appid=2a903d43857f857ce394130ab936b104`. The JSON data is as follows:

```
{
  "lat": 9.9333,
  "lon": 78.1233,
  "weather": [
    {
      "id": 721,
      "main": "Haze",
      "description": "haze",
      "icon": "50n"
    }
  ],
  "base": "stations",
  "main": {
    "temp": 298.16,
    "feels_like": 298.52,
    "temp_min": 298.16,
    "temp_max": 298.16,
    "pressure": 1010,
    "humidity": 69
  },
  "visibility": 4000,
  "wind": {
    "speed": 2.57,
    "deg": 70
  },
  "clouds": {
    "all": 75
  }
}
```

Humidity : 69

Temperature : 298.16 (in Kelvin)

25.01 (in Celsius)



The screenshot shows a Google search result for the query 'kelvin to celsius'. The search bar contains the text 'kelvin to celsius'. Below the search bar, the results show a conversion tool. The tool displays the input value '298.16' in Kelvin, which is converted to '25.01' in Celsius. The formula used for the conversion is  $298.16K - 273.15 = 25.01^{\circ}C$ . Below the conversion tool, there are several 'People also ask' questions related to the conversion of Kelvin to Celsius.

Temperature

298.16 = 25.01

Kelvin Celsius

Formula  $298.16K - 273.15 = 25.01^{\circ}C$

More info Feedback

People also ask :

- What is the formula of Kelvin to degree Celsius?
- What is 1 Celsius equal to in Kelvin?
- What is 300 Kelvin on Celsius scale?
- Why do we use 273 to convert Celsius to Kelvin?

Feedback

<https://bvius.com/Formulas/Physics/Formulas/>