SMART AGICULTURAL SYSTEM BASED ON IOT

PROJECT REPORT:

Contents:

\-

\-

\-

INTRODUCTION:

1.1 Overview

The objectives of this report is to proposed IoT based Smart Agriculture System which will

enable farmers to have live data of soil moisture environment temperature at very low cost

so that live monitoring can be done.

- Smart Agriculture System based on IoT can monitor soil moisture and climatic conditions to grow and yield a good crop.
- The farmer can also get the realtime weather forecasting data by using external platforms like Open Weather API.
- Farmer is provided a mobile app using which he can monitor the temperature,humidity and soil moisture parameters along with weather forecasting data.

- Based on all the parameters he can water his crop by controlling the motors using the mobile application.

- Even if the farmer is not present near his crop he can water his crop by controlling the motors using the mobile application

- Here we are using the Online IoT simulator for getting the Temperature,Humidity and Soil Moisture values.

- The farmer can get the weather forecasting data by external platforms like open weather API.By using IBM cloud platform we will perform all the required activities.This platform will provide us simulators that enable the users to represent the test phenomena likely to occur in actual performance.

- From the IBM cloud platform we will connect the we will connect the IOT Simulator to the IBM watson IOT platform.This platform enables us to create a new device and we use Node-Red programming tool for wiring together hardware devices,API's and online services.

- We need to build a web application by using IBM cloud services in such a way that the farmer can access his motor through his mobile and also he gets the information about the weather conditions in his field.

1.2 Purpose

- With the rapid growth in world population, food consumption worldwide also grows rapidly. A rapid escalation in food production to cater to the growing demand is not an easy task. Agriculture being the oldest industry has evolved so far to the age of what can now be termed as The Third Green Revolution. Modern Information and Communication Technologies are implemented into agriculture, in order to deliver a sustainable agricultural production.

- By using this smart agriculture farmers can be benefited a lot as there is no use of going to farm everytime.With the help of mobile application farmer can operate his motor from his home comfortably.We are using IOT simulator to get the temperature,humidity and soil moisture.

LITERATURE SURVEY:

2.1 Existing problem

- Farmers are facing problem to yield the crops,as they have to visit farms frequently to check the crops and for watering them,even in hot summers they have to visit farms for taking care of their fields.

- So we need to design Smart Agriculture System based on IoT that can monitor soil moisture and climatic conditions to grow and yield a good crop.

- The farmer can also get the realtime weather forecasting data by using external platforms like Open Weather API.

- Based on all the parameters he can water his crop by controlling the motors using the mobile application.
- Even if the farmer is not present near his crop he can water his crop by controlling the motors using the mobile application from anywhere.
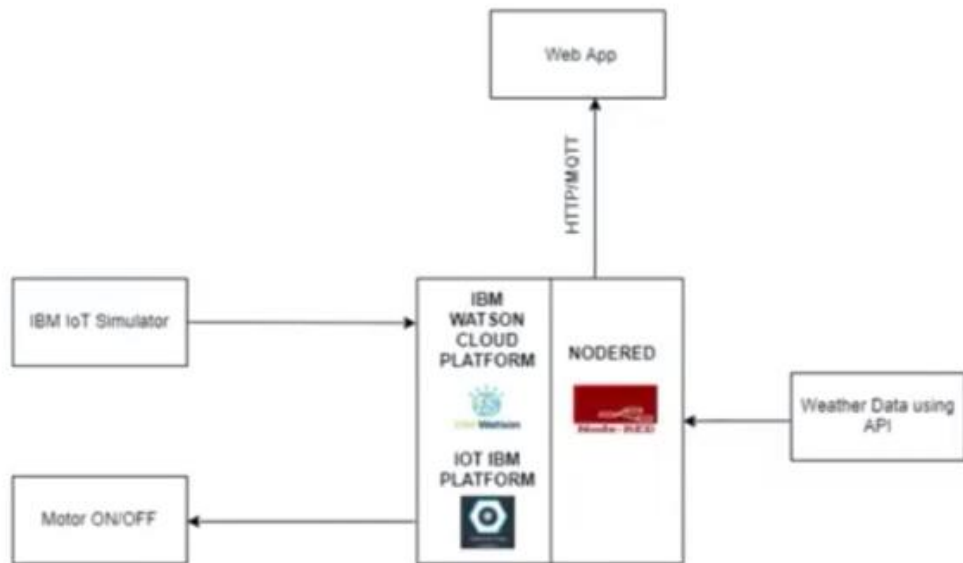
2.2 Proposed Solution

- We need to create a device using IBM Watson IOT platform and IOT sensor is also available in IOT platform to sense the weather conditions.

- In Node-Red we have to create required nodes and develop a web UI which is easily accessible to farmers.

- Next we need to write a python code and configure our device to receive the data from the webapplication and control our motors and then we are done with our project.

-

-

THEORETICAL ANALYSIS

3.1 Block Diagram

```
          Web App

            ↑
         HTTP/MQTT

IBM IoT Simulator  →  IBM          NODERED      ←   Weather Data using
                      WATSON                          API
                      CLOUD
                      PLATFORM

                      IBM Watson

Motor ON/OFF     ←    IOT IBM
                      PLATFORM
```

Block Diagram

Mobile Phone

Watson IOT sensor simulator

Open Weather API

-

-

-

3.2 Required Software Installation

3.2 (A)Node Red

Node-Red is a flow based development tool for visual programming developed originally by IBM wiring together hardware devices,API's and online services as part of Internet of Things.

Node-Red provides a web browser based flow editor,which can be used to create javascript functions.

-

-

-

-

-
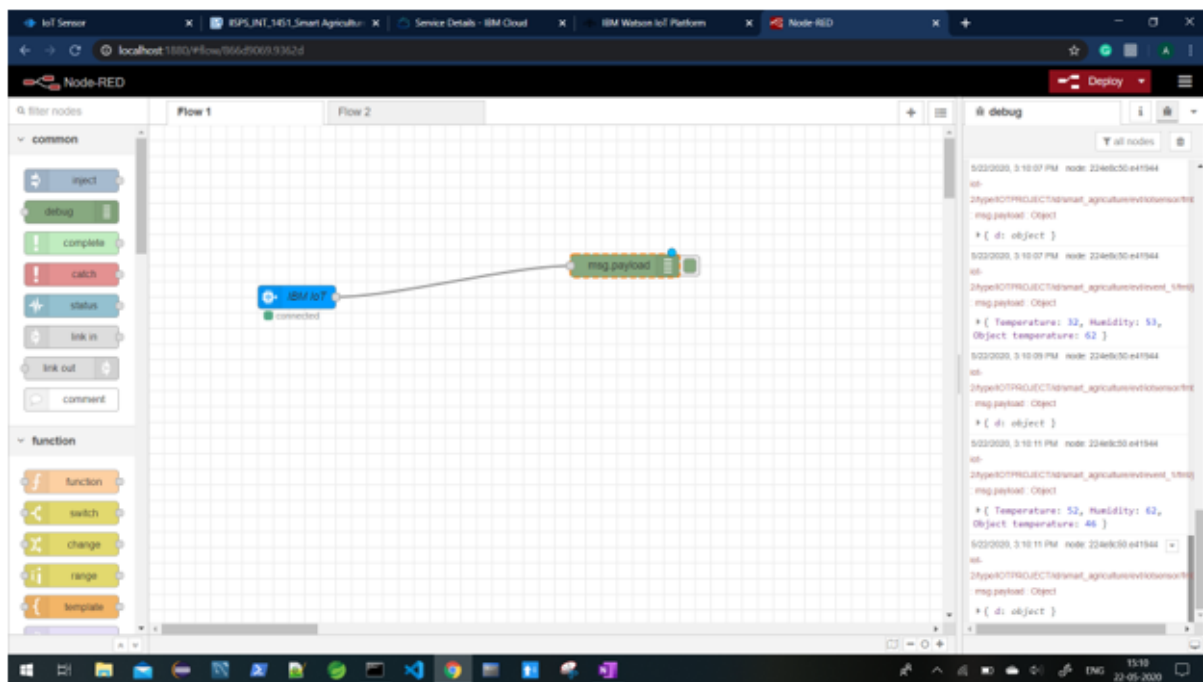
Installation:

- First install the the Node js.
- open command prompt
- Type ->npm install node-red

To Run the application:

- open command prompt
- And then type "node-red"
- Now open <u>http://localhost:1880/ in browser</u>

Installation of IBM IOT nodes and Dashboard nodes for Node-Red

- In order to connect to IBM Watson IOT platform and create the web UI these nodes are required

1.IBM IOT Node

2.Dash Board Node

3.2(B)IBM Watson IOT Platform

- A fully managed cloud hosted service with capabilities for device registration,connectivity ,control,rapid visualization and datastorage,IBM Watson IOT is a managed,cloud-hosted service designed to make it simple to derive value from your IOT devices.
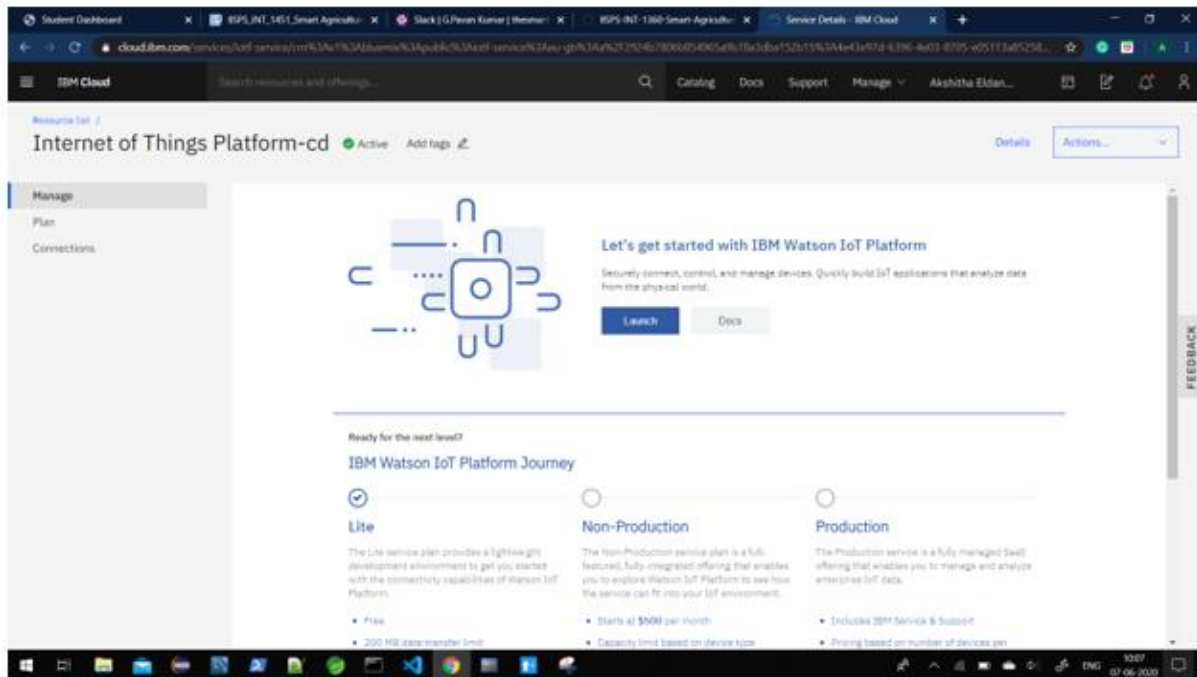
- ## Steps To Configure:
- Create an account in IBM cloud using your email ID
- Create IBM Watson Platform in services in your IBM cloud account
- Launch the IBM Watson Iot Platform
- Create a new device
- Give credentials like device type, device ID, Auth. Token
- Create API key and store API key and token elsewhere
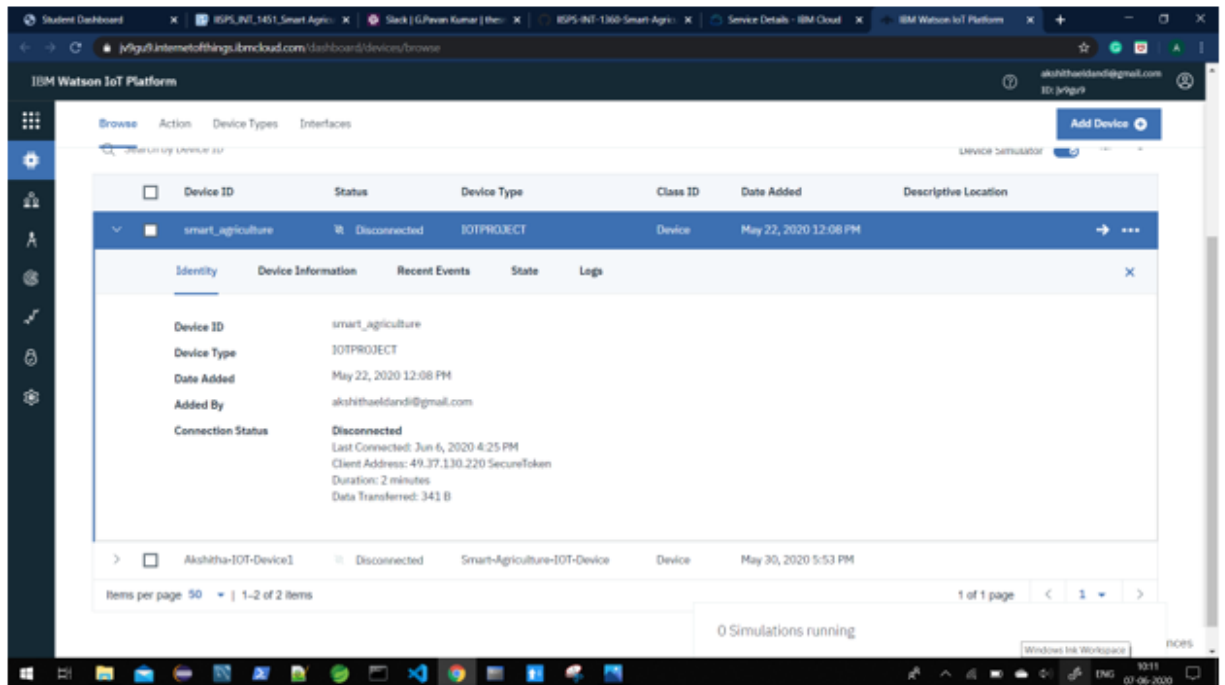
3.2 (C) Python IDE

- Install python 3 Compiler
- I Installed PyCharm Community Edition 2020
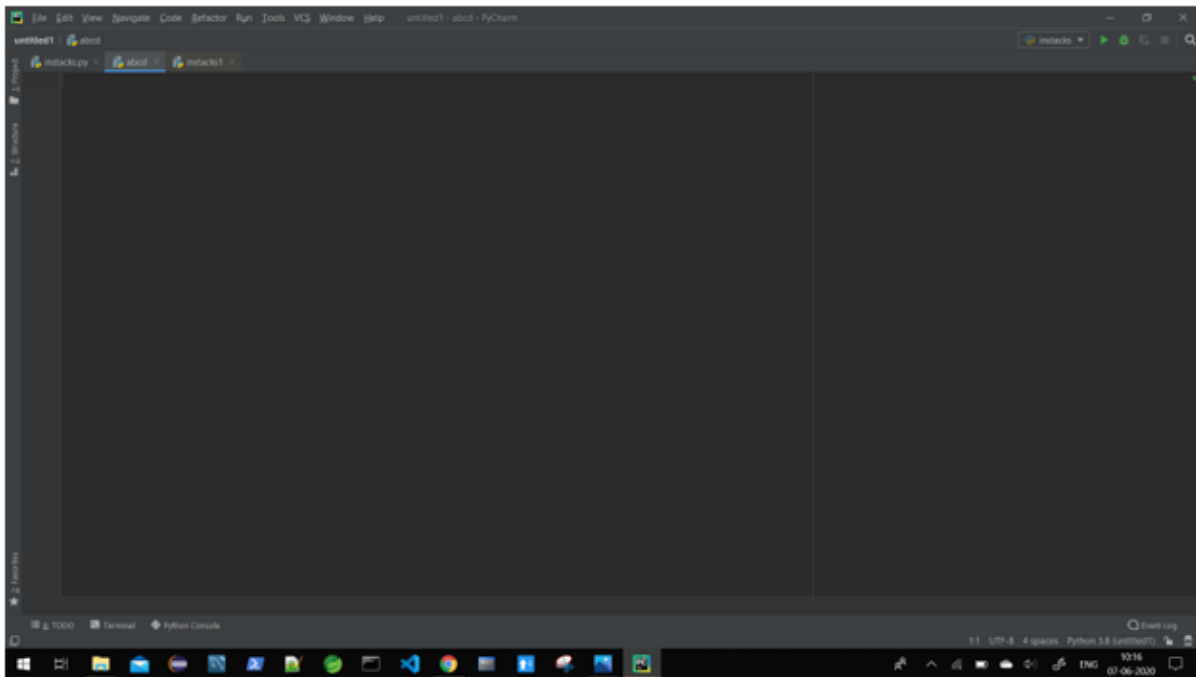
3.3 IOT Simulator

- In our project in the place of sensors we are going to use loT sensor simulator which give random readings to the connected cloud.
- The link to simulator: https://watson-iot-sensor-simulator.mybluemix.net/
- We need to give the credentials of the created device in IBM Watson lot Platform to connect cloud to simulator.

-

-

-

-

3.4 OpenWeather API

- OpenWeatherMap is an online service that provides weather data. It provides current weather data,forecasts and historical data to more than 2 million customer.
- Website link: https://openweathermap.org/guideSteps to configure:
- Create account in OpenWeather
- Find the name of your city by searching
- Create API key to your account
- Replace "city name" and "your api key" with your city and API key in below text api.openweathermap.org/data/2.5/weather?q={city name}&appid={your api key}
- Link I used in my project:
- http://api.openweathermap.org/data/2.5/weather?q=Hyderabad,IN&appid=89d8b67c1cab164eedf32aa4d8da723f
- The below link is for getting weather Forecasting data:

- http://api.openweathermap.org/data/2.5/forecast?q=Hyderabad,IN&appid=89d8b67c1cab164eedf32aa4d8da723f

-

4.Building Project

4.1 Connecting IOT Simulator to IBM Watson IOT Platform

- Open link Provided in section 3.4
- Give the credentials of your device in IBM Watson IOT Platform.
- Click on Connect
- My credentials given to simulator are:
- Organization ID:ka1gns
- Device Type:nodemcu
- Device ID:1234five6789
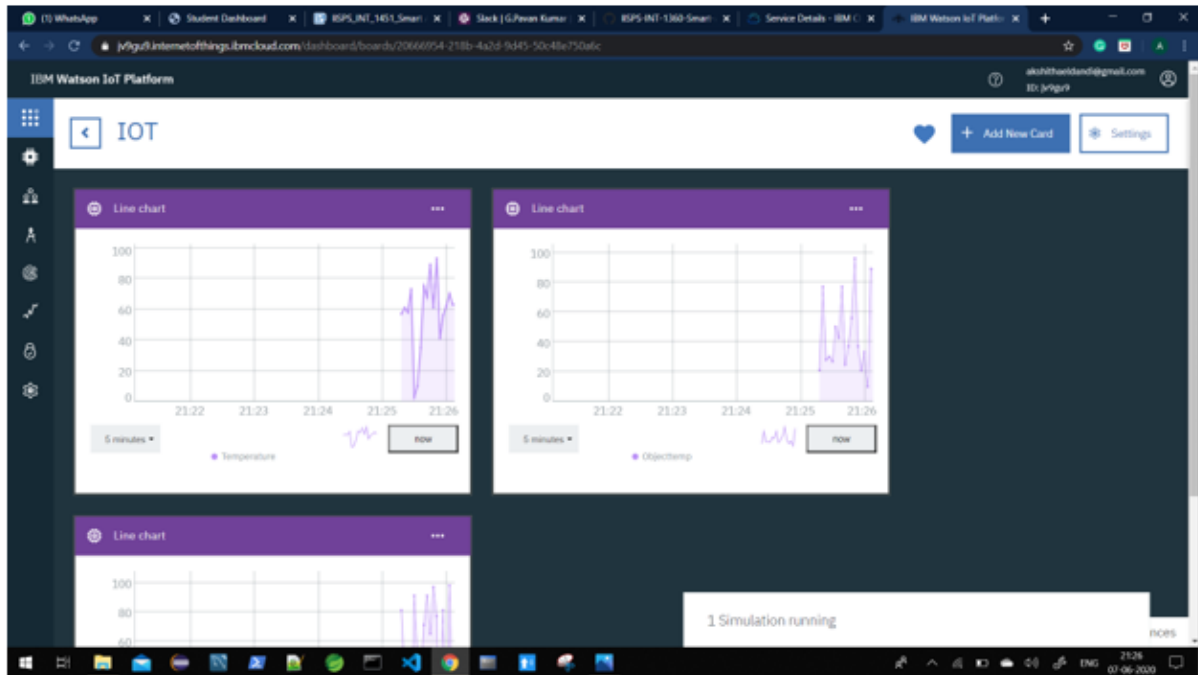- Authentication Method:use-token-auth
- Authentication Token:*********
-
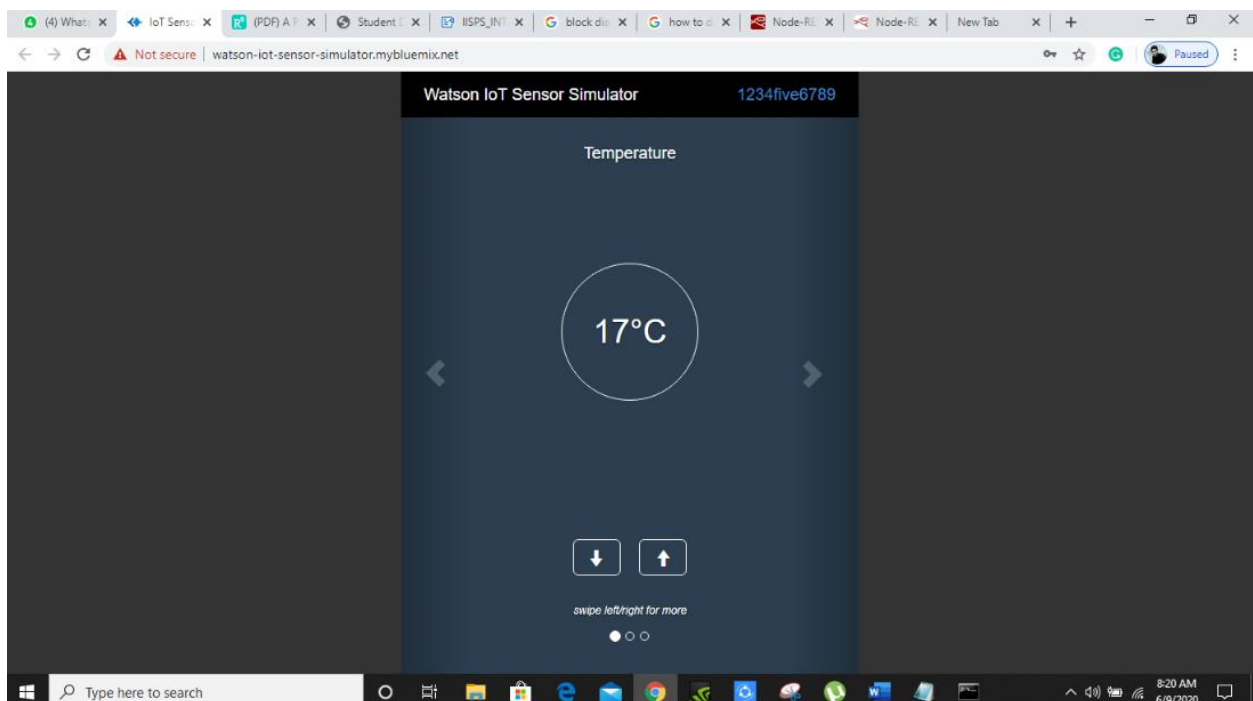
- You Will recieve the simulator data in cloud
- You can see the received data in Recent events
- Data is received in this format (json)

- You can see the received data in cards by creating cards in Boards Tab

4.2 Configuration of Node-Red to collect IBM Cloud Data

- The Node-Red IBM IOT App is added in Node-Red Work flow.The appropriate device credentials obtained earlier are entered into node to connect and fetch device to Node-Red
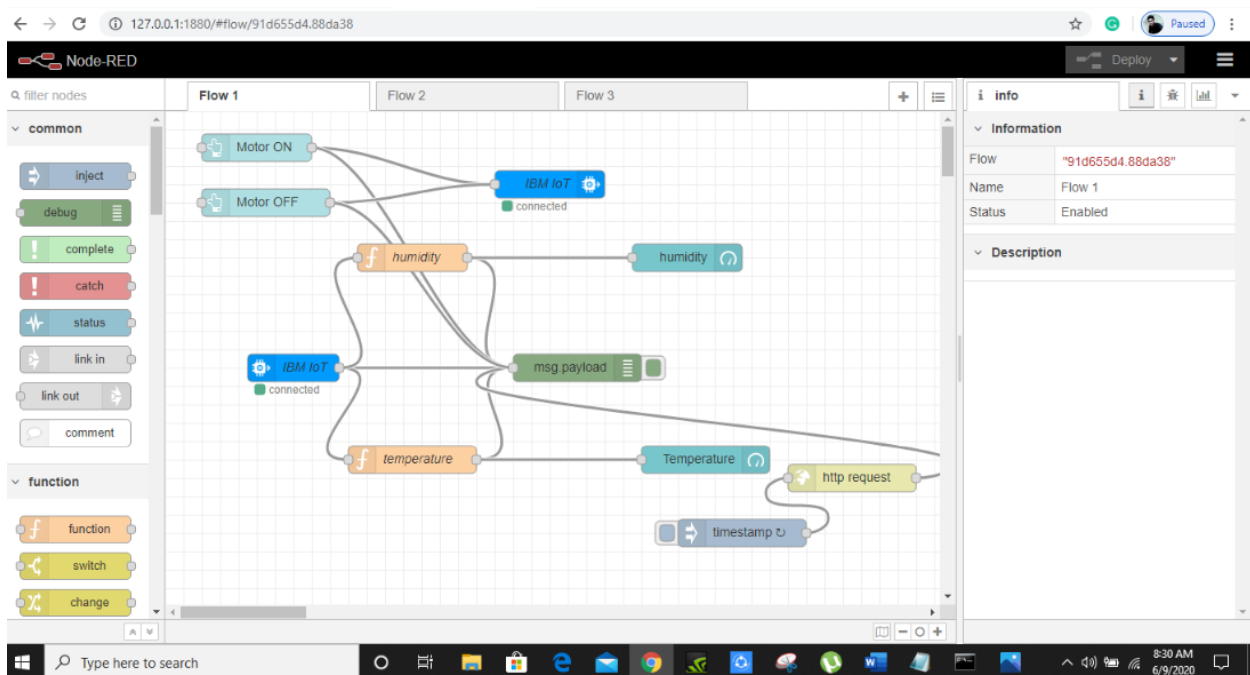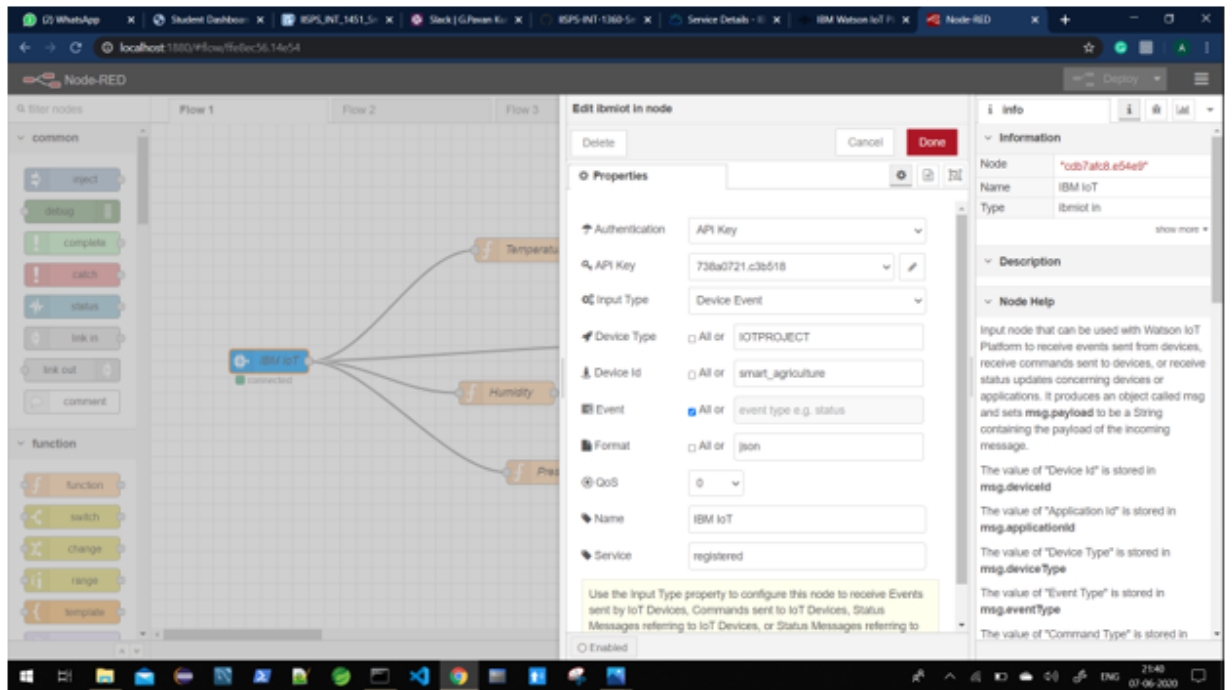
- Once it is connected Node-Red receives data from the device

- Display the data using debug node for verification.
- These are my flows
- FLOW 1
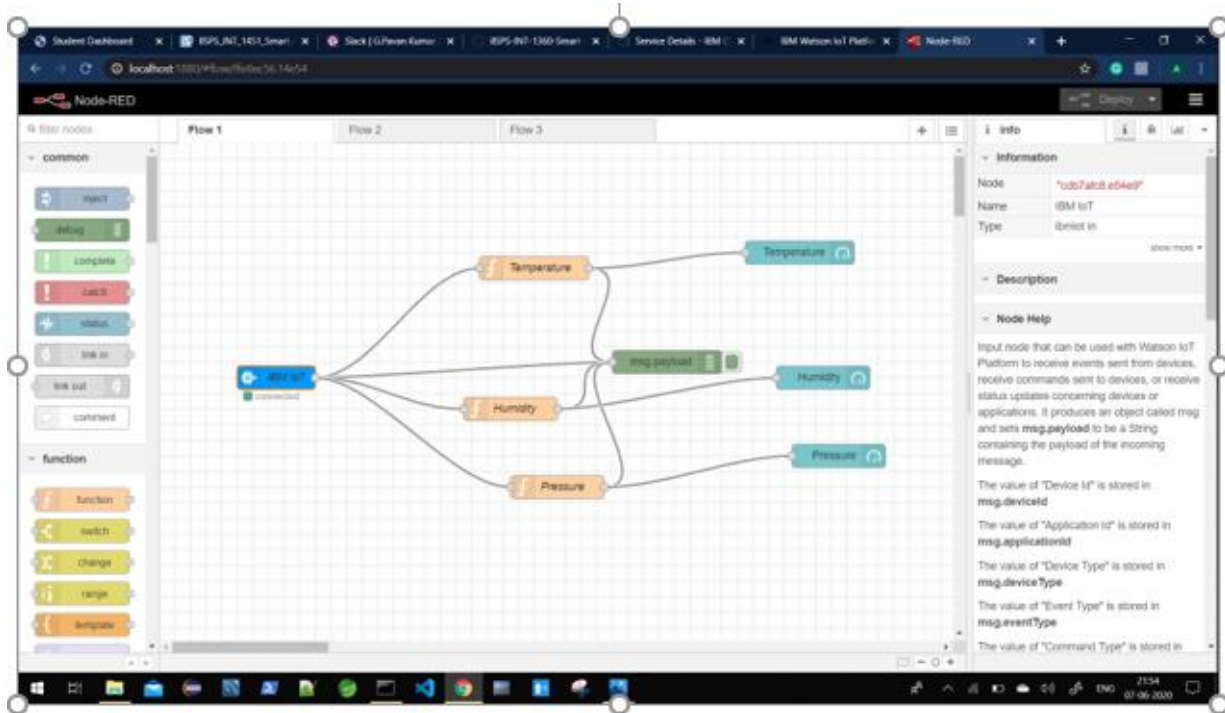
- FLOW 2
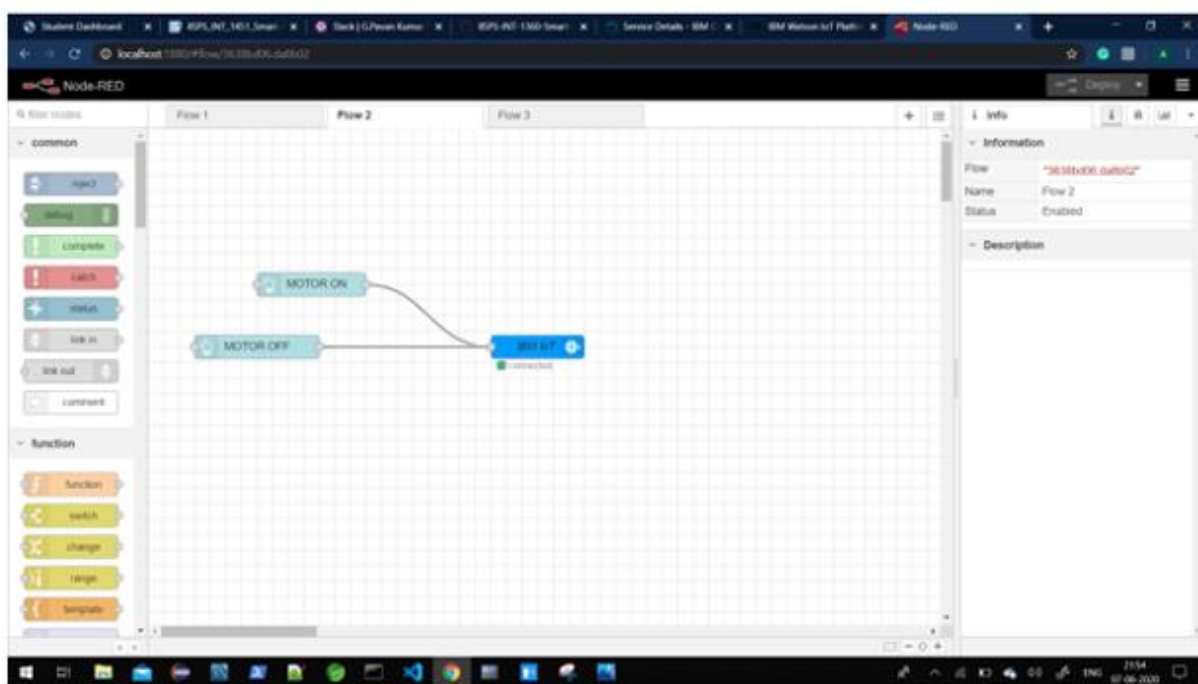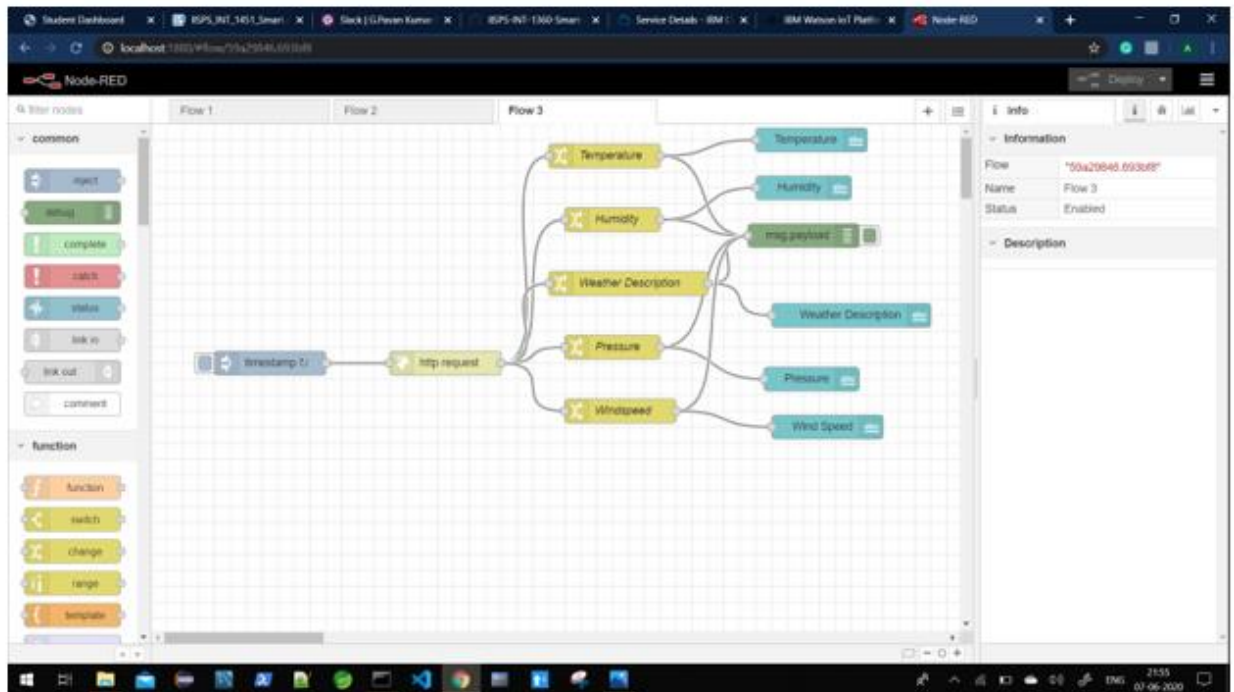
4.3 Configuration of Node-Red to collect data from Open weather API

- The Node-Red also receive data from the OpenWeather API by HTTP GET request. An inject trigger is added to perform HTTP request for every certain interval.
- The data we receive from OpenWeather after request is in JSON format

4.4 Configuration of Node-Red to send commands to IBM Cloud

- By using IBM Iot out Node I used to send data from Node-Red to IBM Watson device.So,after adding it to flow we need to configure it with credentials of our Watson device.

4.5 Adjusting UserInterface

- By connecting all the flows shown above
- We can display our UI by clicking on the dashboard tab in Node red platform.

- In the above page we can display the sensor data and motor commands.

- I
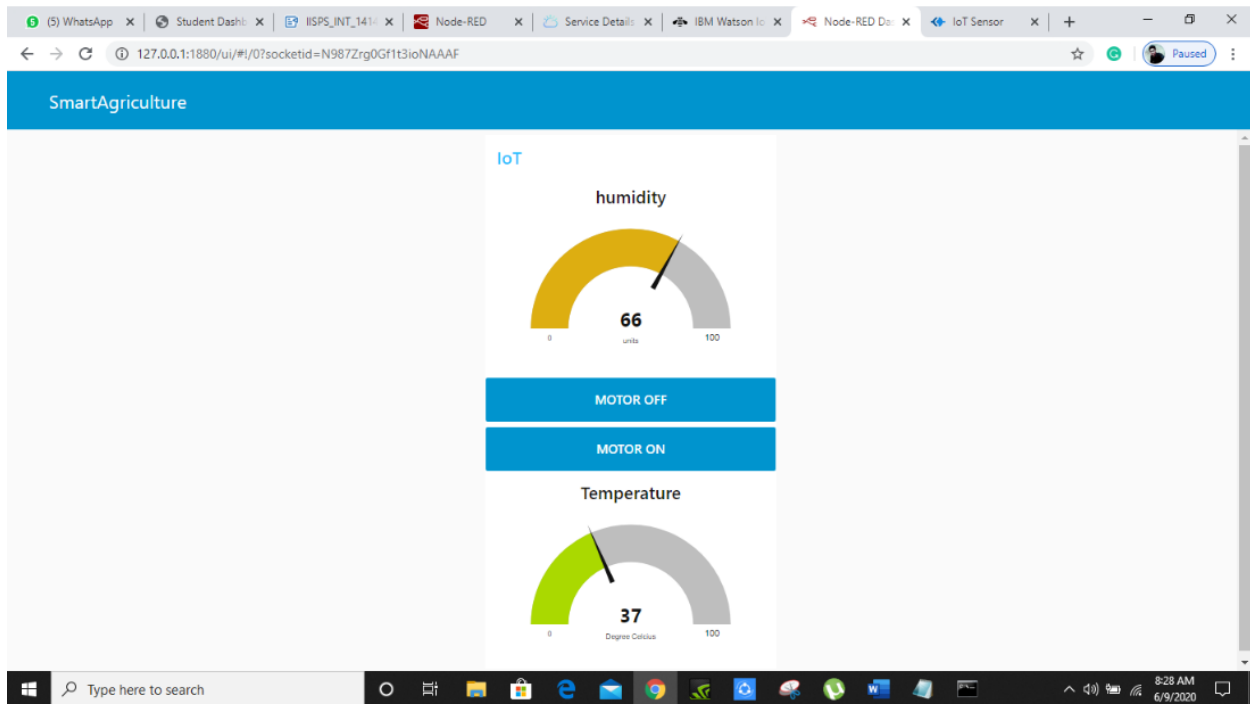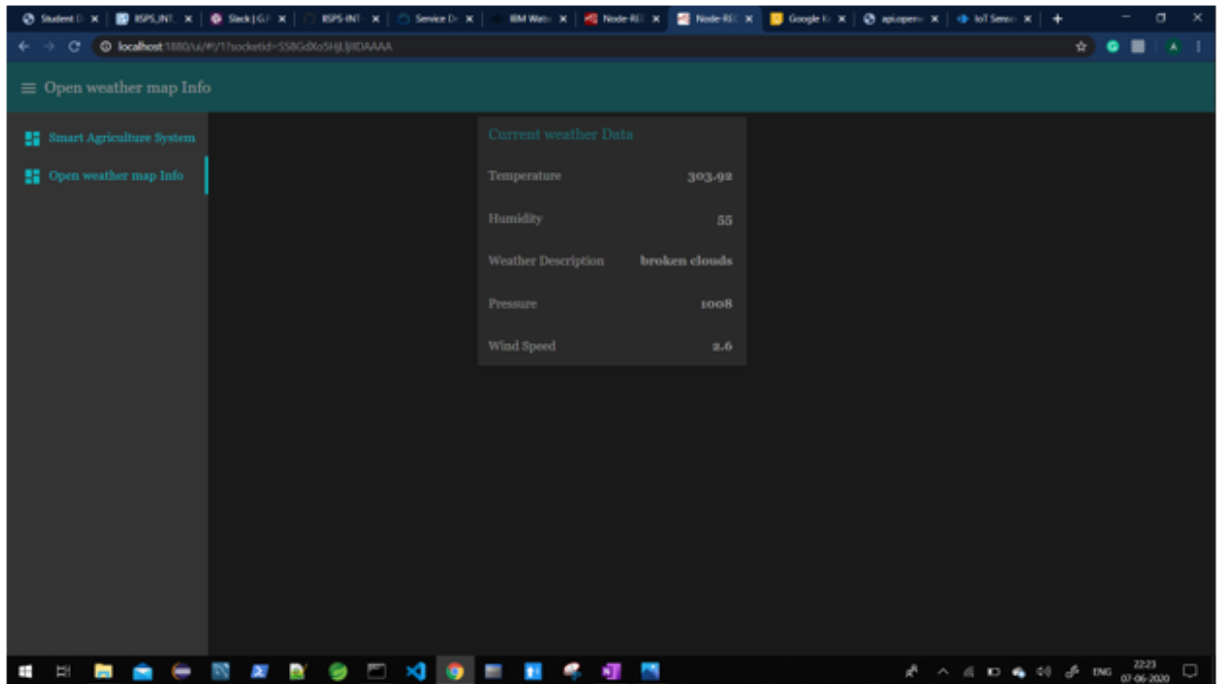
- In this page we open weather API data is displayed

-

4.6 Receiving Commands from IBM Cloud using python program

- This is my python code

```python
import time
import sys
import ibmiotf.application # to install pip install ibmiotf
import ibmiotf.device

#Provide your IBM Watson Device Credentials
organization = "ka1gns" #replace the ORG ID
deviceType = "nodemcu"#replace the Device type wi
deviceId = "*********"#replace Device ID
authMethod = "token"
authToken = "**********" #Replace the authtoken
```

```python
def myCommandCallback(cmd): # function for Callback
    print("Command received: %s" % cmd.data)
    if cmd.data['command']=='motoron':
        print("MOTOR ON IS RECEIVED")
        #time.sleep(1)
        #print("Motor Started")
    elif cmd.data['command']=='motoroff':
        print("MOTOR OFF IS RECEIVED")
        #time.sleep(1)
        #print("Motor Stopped")
    #elif cmd.data['command'] == 'runfor30minutes':
        #print("Motor Runs For 30 Minutes")
        #print("Motor Started")
        #for i in range(1,31):
        #print("%d minutes to stop "%(30-i))
        #print("Motor Stopped")

    if cmd.command == "setInterval":

        if 'interval' not in cmd.data:
            print("Error - command is missing required information: 'interval'")
        else:
            interval = cmd.data['interval']
    elif cmd.command == "print":
        if 'message' not in cmd.data:
            print("Error - command is missing required information: 'message'")
        else:
            output=cmd.data['message']
            print(output)

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:

    deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud
deviceCli.disconnect()
```
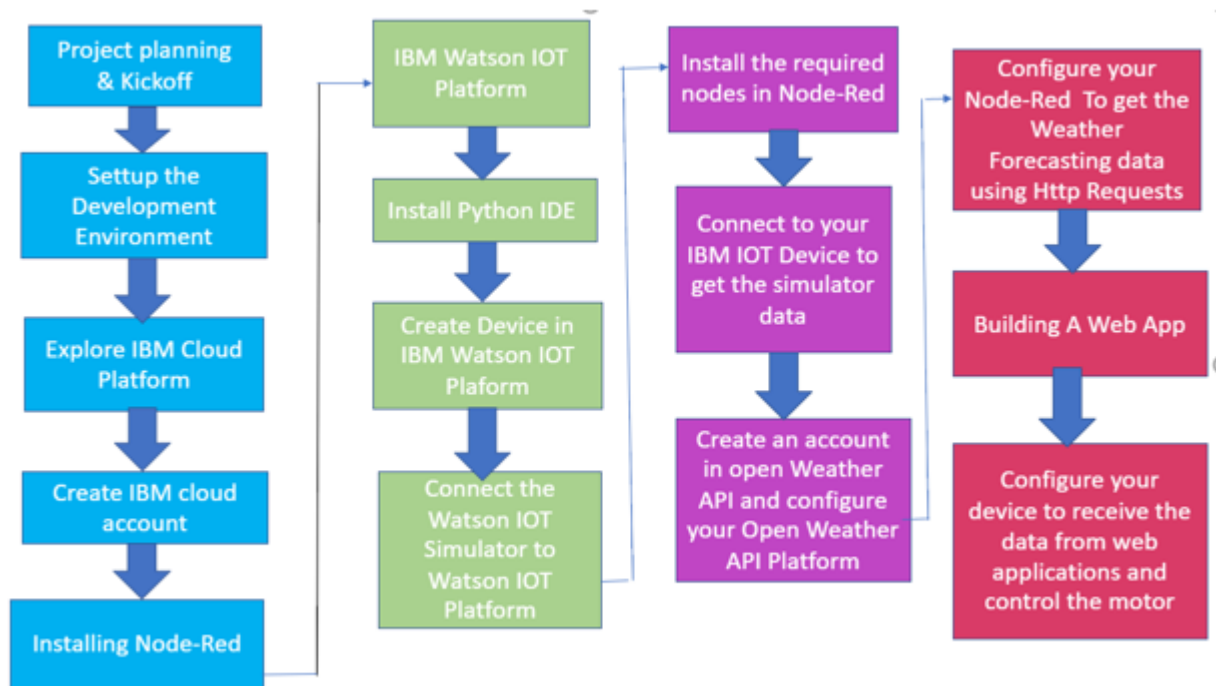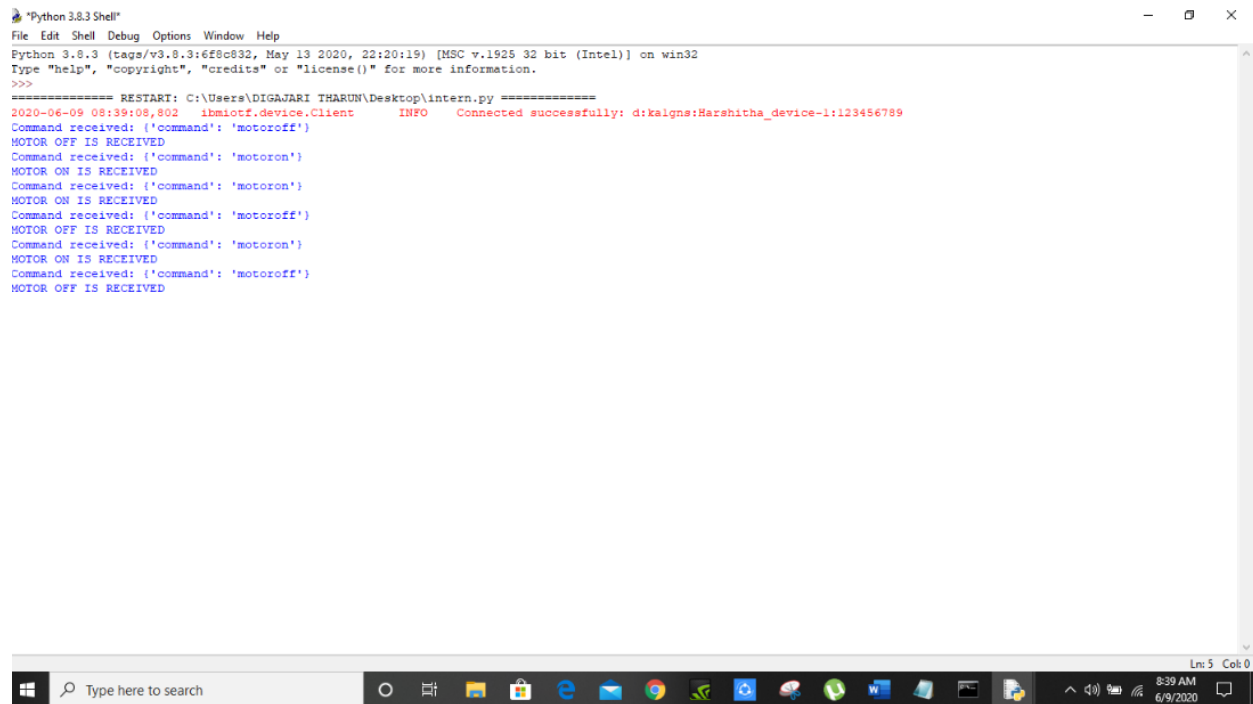
5.FLOW CHART

```
┌─────────────────┐      ┌─────────────────┐      ┌─────────────────┐      ┌─────────────────┐
│ Project planning│      │ IBM Watson IOT  │      │ Install the     │      │ Configure your  │
│ & Kickoff       │      │ Platform        │      │ required        │      │ Node-Red To get │
│                 │      │                 │      │ nodes in        │      │ the Weather     │
│                 │      │                 │      │ Node-Red        │      │ Forecasting data│
│                 │      │                 │      │                 │      │ using Http      │
│                 │      │                 │      │                 │      │ Requests        │
└────────┬────────┘      └────────┬────────┘      └────────┬────────┘      └────────┬────────┘
         │                        │                        │                        │
┌────────▼────────┐      ┌────────▼────────┐      ┌────────▼────────┐      ┌────────▼────────┐
│ Setup the       │      │ Install Python  │      │ Connect to your │      │ Building A      │
│ Development     │      │ IDE             │      │ IBM IOT Device  │      │ Web App         │
│ Environment     │      │                 │      │ to get the      │      │                 │
│                 │      │                 │      │ simulator data  │      │                 │
└────────┬────────┘      └────────┬────────┘      └────────┬────────┘      └────────┬────────┘
         │                        │                        │                        │
┌────────▼────────┐      ┌────────▼────────┐      ┌────────▼────────┐      ┌────────▼────────┐
│ Explore IBM     │      │ Create Device in│      │ Create an       │      │ Configure your  │
│ Cloud Platform  │      │ IBM Watson IOT  │      │ account in open │      │ device to       │
│                 │      │ Plaform         │      │ Weather API and │      │ receive the     │
│                 │      │                 │      │ configure your  │      │ data from web   │
│                 │      │                 │      │ Open Weather    │      │ applications and│
│                 │      │                 │      │ API Platform    │      │ control the     │
└────────┬────────┘      └────────┬────────┘      └─────────────────┘      │ motor           │
         │                        │                                        └─────────────────┘
┌────────▼────────┐      ┌────────▼────────┐
│ Create IBM cloud│      │ Connect the     │
│ account         │      │ Watson IOT      │
│                 │      │ Simulator to    │
│                 │      │ Watson IOT      │
│                 │      │ Platform        │
└────────┬────────┘      └─────────────────┘
         │
┌────────▼────────┐
│ Installing      │
│ Node-Red        │
└─────────────────┘
```

-

-

-

6.OBSERVATION & RESULTS

- When I click Motor ON and Motor OFF on my UI I will be receving commands like below.

-

-

-

-

-

7.ADVANTAGES AND DISADVANTAGES

Advantages

- Farms can be monitored and controlled remotely.
- Increase in convenience to farmers.
- Less labour cost.
- Better standards of living.

Disadvantages:

- Lack of internet/connectivity issues.
- Added cost of internet and internet gateway infrastructure.

- Farmers wanted to adapt the use of WebApp.

8.APPLICATIONS

The most common IoT applications in smart agriculture are:

1. Sensor-based systems for monitoring crops, soil, fields, livestock, storage facilities, or basically any important factor that influences the production.

2. Smart agriculture vehicles, drones, autonomous robots and actuators.

3. Connected agriculture spaces such as smart greenhouses or hydroponics.

4. Data analytics, visualization and management systems.

9. CONCLUSION

- Thus the objective of the project to implement an IoT system in order to help farmers to control and monitor their farms has been implemented successfully.

10.FUTURE SCOPE:

- Smart farming based on IoT technologies enables growers and farmers to reduce waste and enhance productivity ranging from the quantity of fertilizer utilized to the number of journeys the farm vehicles have made, and enabling efficient utilization of resources such as water, electricity, etc

11. BIBILOGRAPHY

- IBM cloud reference: https://cloud.ibm.com/
- Python code reference: https://github.com/rachuriharish23/ibmsubscribe
- IoT simulator : https://watson-iot-sensor-simulator.mybluemix.net/
- Open Weather : https://openweathermap.org/