PROJECT REPORT

PROJECT NAME:Smart Agriculture system based on IOT

INTERNSHIP AT SMARTINTERNZ

Internship Title : Smart Agriculture system based on IoT - SB45738

Application ID : SPS_APL_20200002696

Project ID : SPS_PRO_101

Project Team : G POOJITHA

INTRODUCTION:

The Agriculture is one of the important business the normally affected the people life. From the ancient to the agricultural gyration in India, farming is the way that human used to harvest plants and consumed them in their daily life. Farming has been improved by many applied sciences

supporting cropping system and high yield harvesting technology. In addition to the agricultural revolution era, there have been many technologies that have impacts on agriculture.IoT encompasses many new thinking concepts for using in the near future such as smart home, smart city, smart transportation, and smart farming. To increase the crop yield, the smart farming technology in use to help.

OVERVIEW:

Smart farming System based on IoT can monitor soil moisture and climatic conditions to grow and yield a good crop.The farmer can also get the real time weather forecasting data.The farmer uses mobile app and he monitors the temperature,humidity and soil moisture. Along with these, weather forecasting details are also predicted. Even if the farmer is not present near his crop, he can water his crop by controlling the motors using the mobile app from anywhere.

PURPOSE:

In terms of any environmental issues,**IOT-based smart farming** provides great benefits like more efficient water usage, improvement in quality of crop.It also shows way towards the **Remote monitoring**,where a local or a commercial farmer can monitor multiple crops present at multiple locations just through an internet connection.

SCOPE OF WORK:

1. Create an IBM account.
2. Create a device in IBM Cloud Account.
3. Install Node-RED and download all the required nodes and configure the nodes.
4. Create the open weather map account and get the api key and the weather conditions using api key in the Node-RED.
5. Create a web application for user interaction for observation and control actions.

LITERATURE SURVEY:

EXISTING PROBLEM:

In long time past days agriculturists used to figure the ripeness of soil and influenced presumptions to develop which to kind of product. They did not think about the dampness, level of water and especially climate condition which unacceptable an agriculturist more. They utilise pesticides in view of a few suspicions which made lead a genuine impact to the yield if the supposition isn't right .The profitability relies upon the last phase of the harvest on which agriculturist depends. Due to ignorance of farmers,they do not supply water to crop in adequate quantities.This leads to prolonged dry spells of crop and in turn reduces crop growth and yield.Sometimes,they do not even check few parameters like temperature,humidity and weather conditions for their crop's irrigation.This also decreases the crop yield.

PROPOSED SOLUTION:

1. To develop a Smart Agricultural System based on IOT which can give real time data and can help farmers in a very efficient manner.
2. Soil Moisture can be checked by using the sensors that can sense the soil condition and send the data (moisture content in the soil) over the cloud services to the web application.
3. The supply of water can be controlled from anywhere by controlling the motor state (ON/OFF), using web application.
4. Surrounding temperature can also be sensed by the sensors and displayed on the application.
5. Real time weather conditions can also be known by using different weather API's from different websites and displayed on our

Application.

THEORITICAL ANAIYSIS:

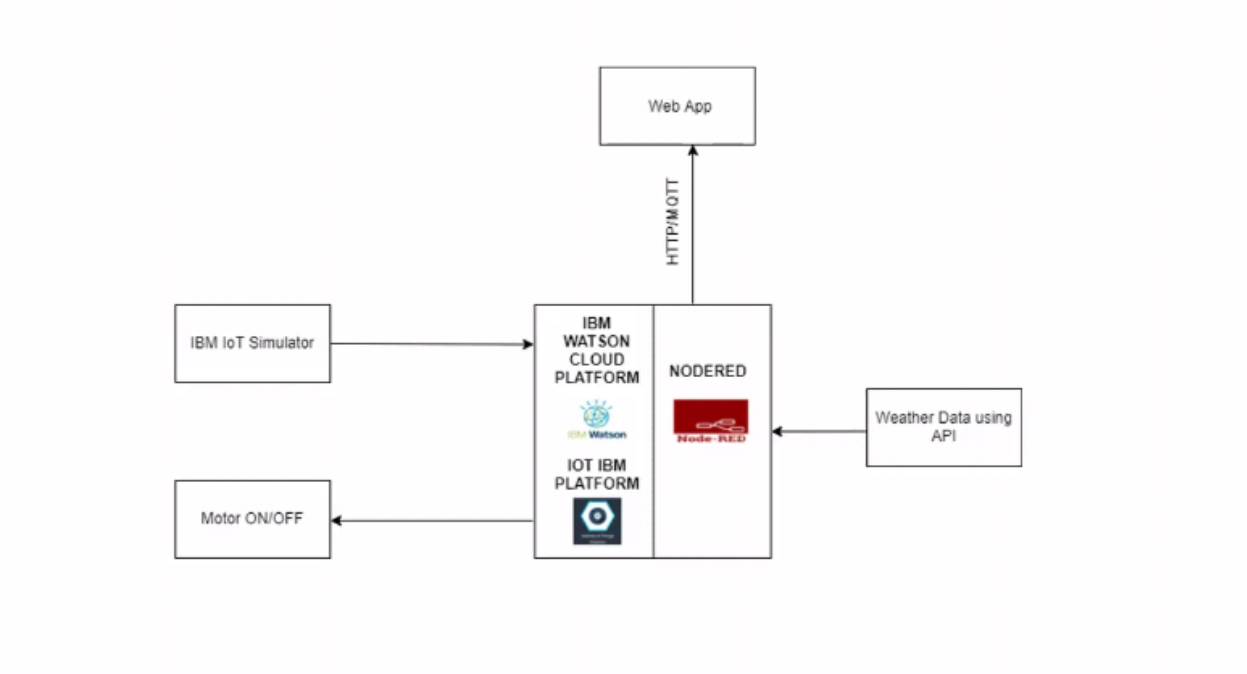
PROJECT ESSENTIALS:

- Online IOT simulator
- Open weather APIs
- IBM Watson code platform
- NODERED
- Web app

SOFTWARE REQUIREMENTS:

- Python 3 IDLE software
- HTTP/MQTT web language,
- Software of NODERED

BLOCK DIAGRAM:



EXPERIMENTAL INVESTIGATION:

- Primarily, we need to create IBM Watson cloud account and create the device credentials like Device type,Device ID etc

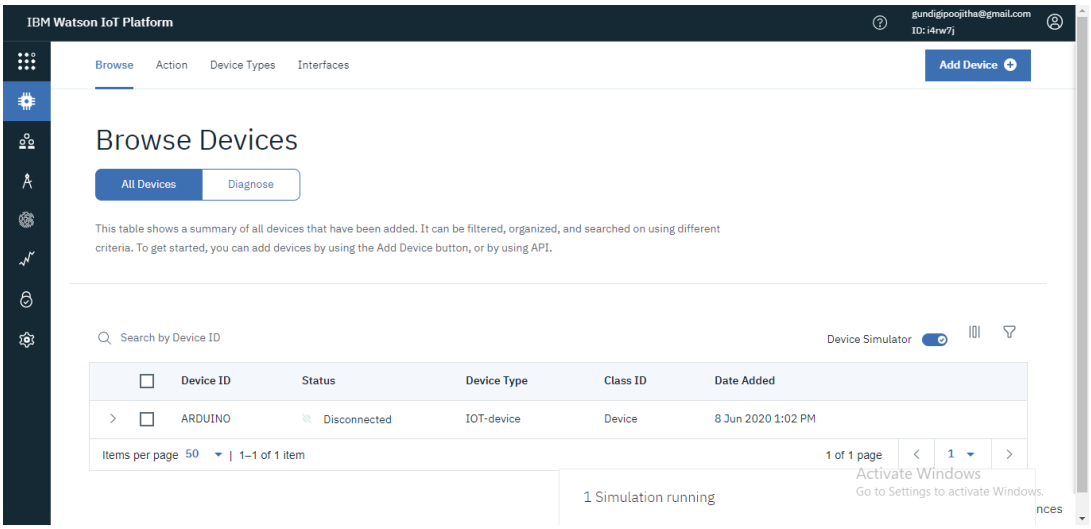
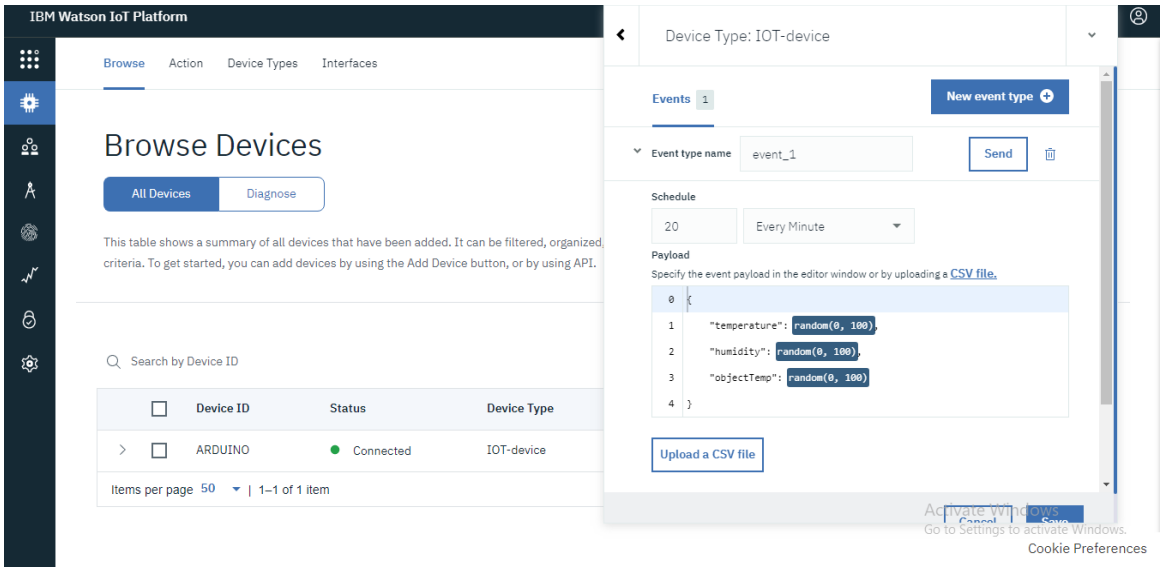
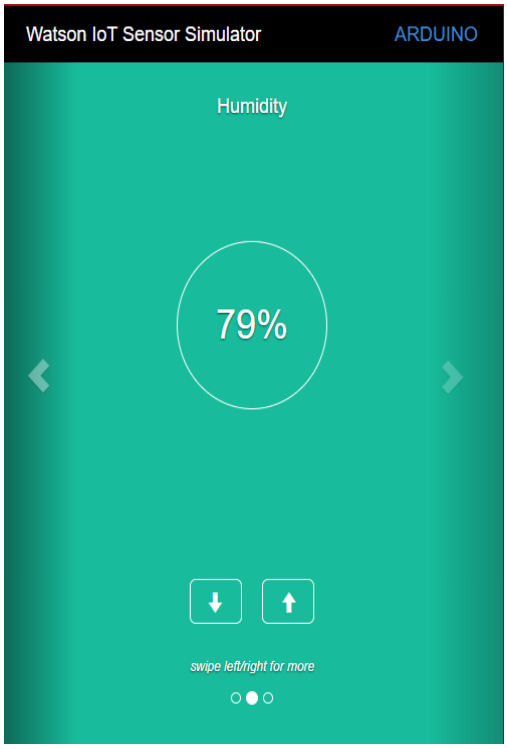


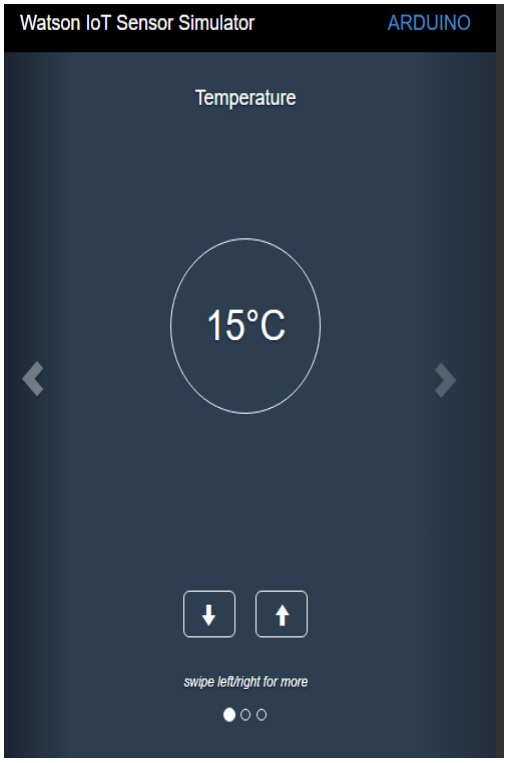
Fig.1:Creation of device type,device ID

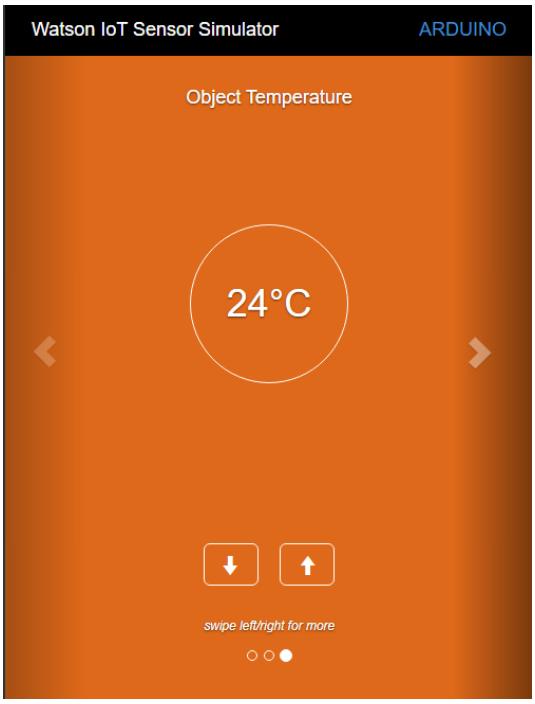
- Simulate the event with required parameters with the fig below:



- Generate the data like temperature,humidity,Object temperature from IBM IOT simulator and send them to IBM Watson IOT platform(refer the figures given below).


Watson IoT Sensor Simulator
ARDUINO
Humidity
79%
swipe left/right for more


Watson IoT Sensor Simulator
ARDUINO
Temperature
15°C
swipe left/right for more


Watson IoT Sensor Simulator
ARDUINO
Object Temperature
24°C
swipe left/right for more

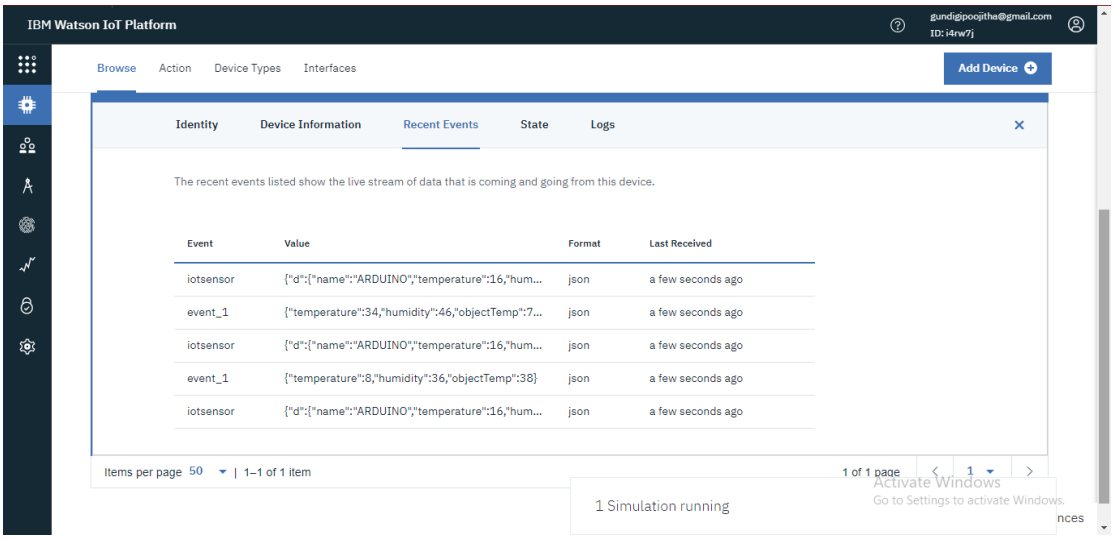
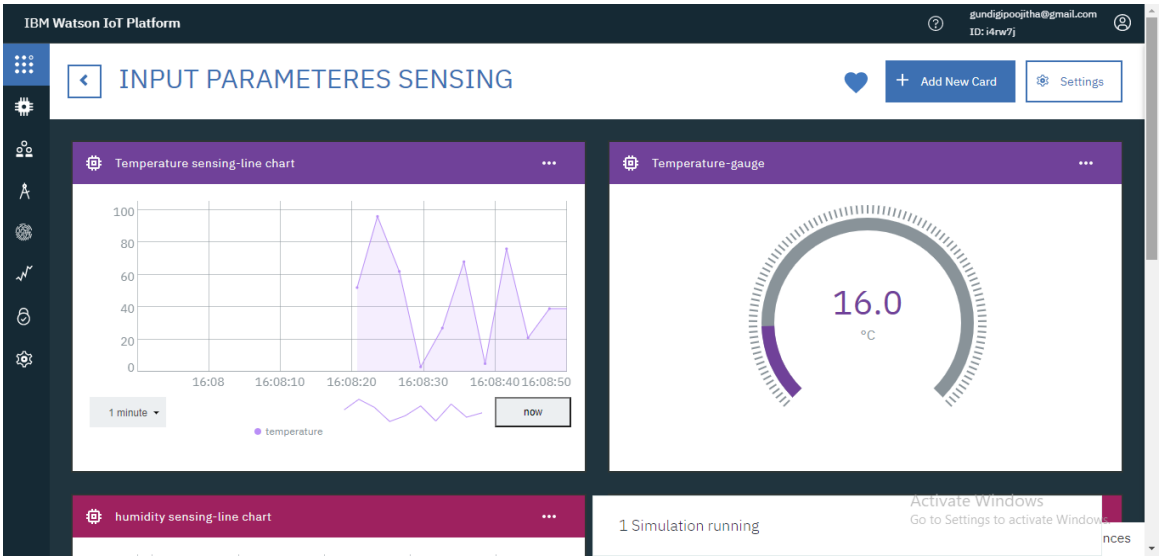


Fig 3: Receiving data from IOT sensor Simulator by the IBM Watson IOT platform through IBM Cloud.

- Represent the data received in any format like bar graphs, line charts ,gauges etc.



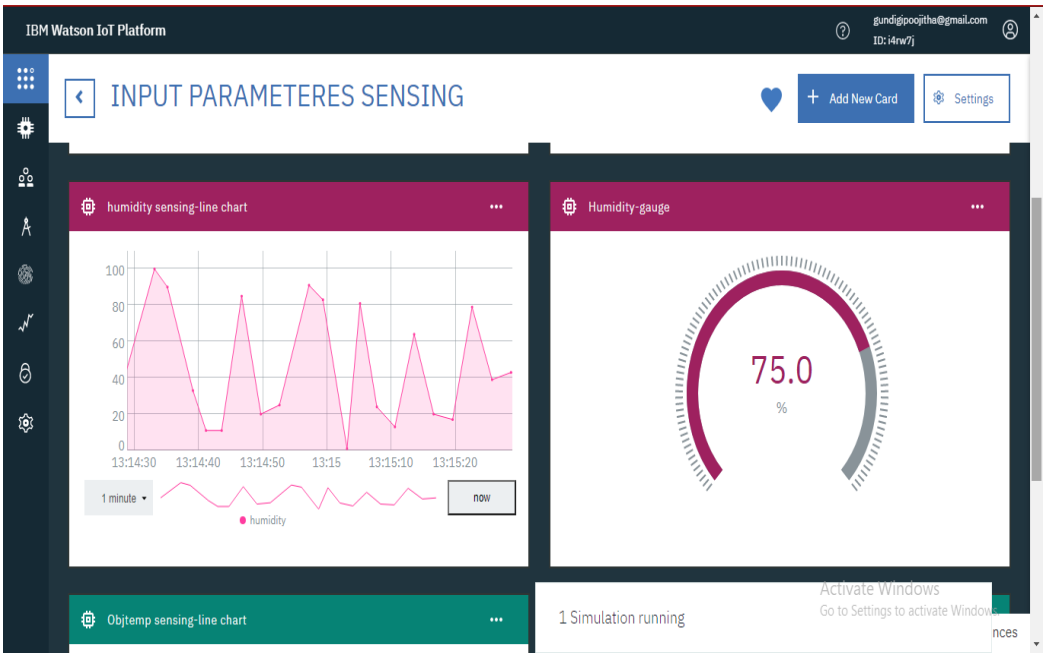
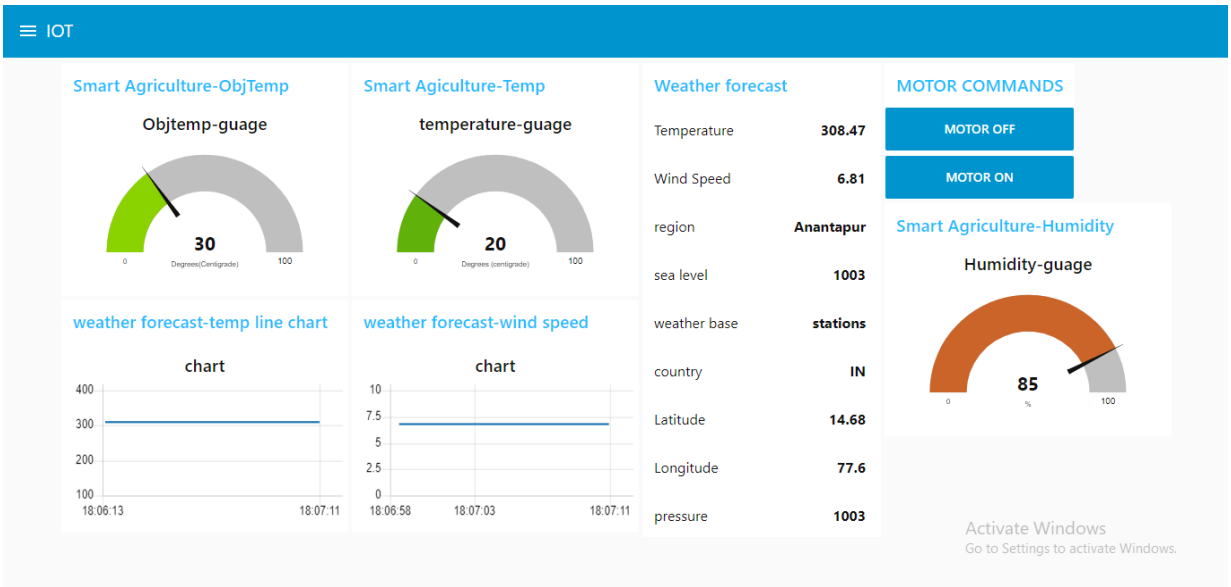


Fig 4: Representation of data that is received in line chart and gauge.

- Now, these parameters need to be sent to web application.This can be done by NODERED.Parameters that are sent for the web application are temperature,humidity,Object temperature and also weather information generated by Open weather API.This weather info can be processed by using HTTP request and response.
- The parameters are analysed can be represented in dashboard.With these, extra parameters can also be visualised as given below:



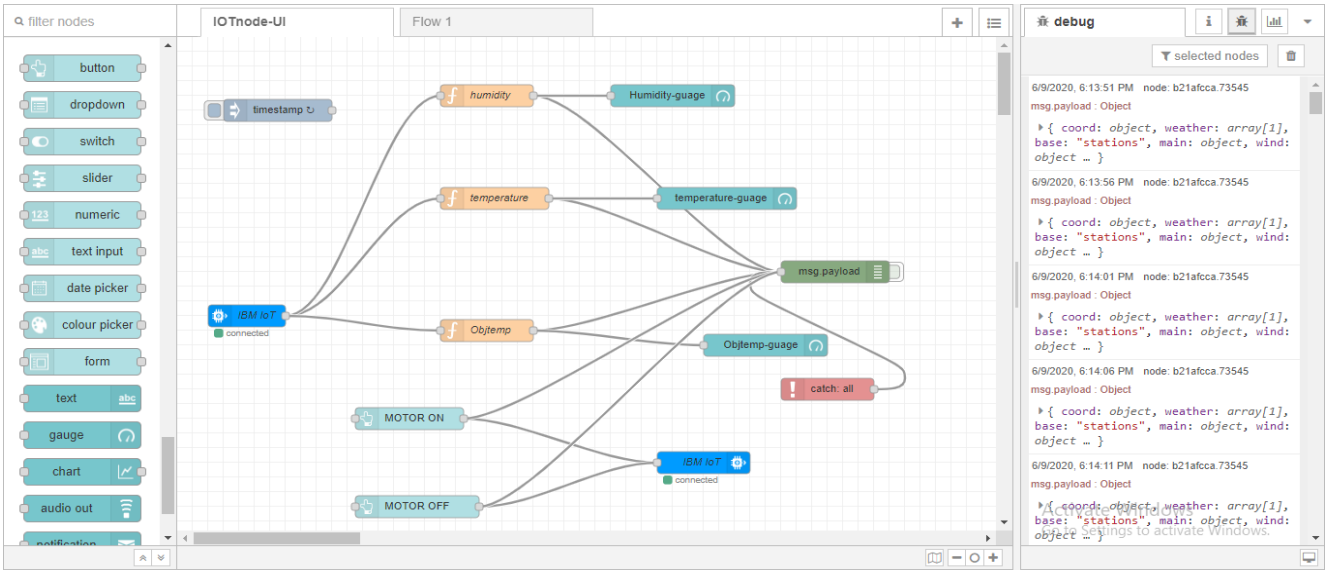
- By executing the python code, along with motor commands,the web app can be worked.

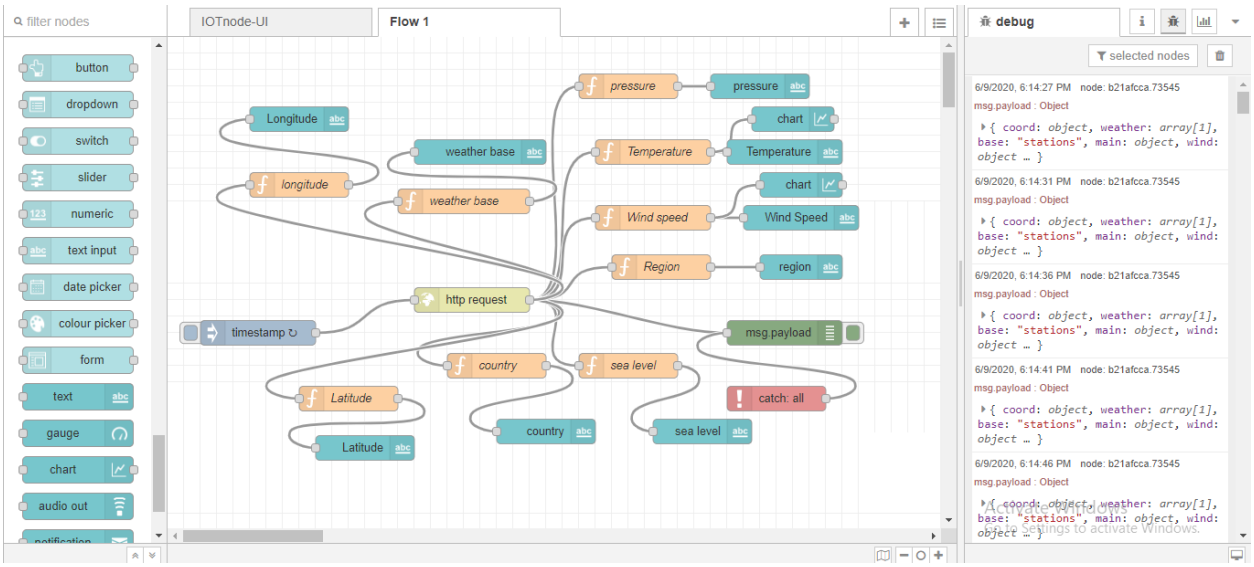
```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\admin\AppData\Local\Programs\Python\Python38\pythoncode2.0.py
2020-06-09 18:13:13,365   ibmiotf.device.Client      INFO    Connected successfully: d:i4rw7j:IOT-device:ARDUINO
Command received: {'command': 'motoron'}
Motor On IS RECEIVED
Command received: {'command': 'motoron'}
Motor On IS RECEIVED
Command received: {'command': 'motoron'}
Motor On IS RECEIVED
Command received: {'command': 'motoron'}
Motor On IS RECEIVED
Command received: {'command': 'motoron'}
Motor On IS RECEIVED
Command received: {'command': 'motoron'}
Motor On IS RECEIVED
Command received: {'command': 'motoron'}
Motor On IS RECEIVED
Command received: {'command': 'motoron'}
Motor On IS RECEIVED
Command received: {'command': 'motoron'}
Motor On IS RECEIVED
Command received: {'command': 'motoron'}
Motor On IS RECEIVED
Command received: {'command': 'motoron'}
Motor On IS RECEIVED
Command received: {'command': 'motoron'}
Motor On IS RECEIVED
Command received: {'command': 'motoron'}
Motor On IS RECEIVED
```

Fig 6: Device control actions

FLOW CHART:

web app can be generated by the flowcharts which are drawn in NODERED.





Following are the nodes used in the project in the Web Application

1. IBM IoT : IN and OUT Nodes.
2. function Nodes.
3. Gauge Nodes.
4. Chart Nodes
5. Debug Nodes
6. Button Nodes.

Following are the nodes used for the weather condition from open weather map:

1. Timestamp Node.
2. http request Node
3. Function Nodes.
4. Text Nodes.
5. Debug Nodes.

ADVANTAGES:

- Increases crop production and it is quality.
- Leads to conservation of natural resources.
- Farming process become burden free .
- Remote monitoring is possible
- Beneficial in water limited geographical isolated areas.

- Due to low cost, it is available to all farmers.
- Reduced Environment footprint.
- Increased efficiency via automation.
- Increased business efficiency through automation process.

LIMITATIONS:

- It requires continuous internet connection which cannot be fulfilled by farmers, who were mostly in rural areas.
- Fault sensors or data processing engines can cause faulty decisions which may lead to over usage of water,fertilisers etc.
- Smart farming based app requires farmer to understand and learn the use of technology.This is the major challenge in adopting smart agriculture at large scale across the country.

APPLICATIONS:

- **Precision Agriculture/Precision Farming** is one of the most famous applications of IoT in Agriculture. It makes the farming practice more precise and controlled by realising smart farming applications such as livestock monitoring, vehicle tracking, field observation, and inventory monitoring. The goal of precision farming is to analyse the data, generated via sensors, to react accordingly. Precision Farming helps farmers to generate data with the help of sensors and analyse that information to take intelligent and quick decisions.
- Just like weather stations, **crop management devices** should be placed in the field to collect data specific to crop farming; from temperature and precipitation to leaf water potential and overall crop health, these can all be used to collect data and information for improved farming practices readily.

- **Efficiency in livestock farming:**

The health of farm animals such as cattle or chicken can be monitored to detect potential signs of disease. This can be linked to a central system which can trigger relevant advice to be sent to farmers, and contribute towards analytics that can be used to identify any outbreaks or trends.

- **Cattle monitoring and management:**

Just like crop monitoring, there are IoT agriculture sensors that can be attached to the animals on a farm to monitor their health and log performance.

CONCLUSION:

This smart farming will revolutionise the world of farming and it will increase the productivity and improve the quality and can save the lives of farmer.There is an urgent need for a system that makes agriculture process easier and burden free from the farmer's side.With the advancement of technology, it has become necessary to increase the annual crop production of our country India, an entirely agro centric economy. The ability to conserve the natural resources ,as well as living a splendid boost to the production, is one of the main aims of incorporating such technology into agricultural domain of the country.To save farmer's effort, time and water has been the most important consideration.

FUTURE SCOPE:

This project has enormous potential and may be used in various other ways, due to its cheap and cost effective design.

- Remotely perform jobs
- Due to extensible feature of sensors, we can add as per our crop specific need.
- It will help the farmers to do work in any seasonal conditions.
- It will reduce danger for the farmers from different breathing and physical problems.
- Moreover, IOT is expected to have dramatic impact in our lives in future.This project can further be implemented by integrating WSNs.
- One of the limitations of this system is that continuous internet connectivity is required at user end which might prove to be costly for farmer. This can be overcome by extending the system to send suggestion via SMS to the farmer directly on his mobile using GSM module instead of mobile app.

BIBLIOGRAPHY:

- https://cloud.ibm.com/docs/overview?topic=overview-whatis

platform

- Watson IoT :https://www.iotone.com/software/ibm-watson-iot-platform IBM Cloud :/s62
- Open weather map:https://openweathermap.org/
- GitHUB:

  https://github.com/SmartPracticeschool/llSPS-INT-1382-Smart-Agriculture-system-based-on-IoT.git
- Node-RED:

  https://www.youtube.com/watch?v=cicTw4SEdxk

  https://nodered.org/docs/getting-started/windows#3-run node-red

APPENDIX:

SOURCE CODE:

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


#Provide your IBM Watson Device Credentials
organization = "i4rw7j" #replace the ORG ID
deviceType = "IOT-device"#replace the Device type
deviceId = "ARDUINO"#replace Device ID
authMethod = "token"
authToken = "123456789" #Replace the authtoken


def myCommandCallback(cmd): # function for Callback
```

```
        print("Command received: %s" % cmd.data)
        if cmd.data['command']=='motoron':
            print("Motor On IS RECEIVED")

        elif cmd.data['command']=='motoroff':
            print("Motor Off IS RECEIVED")

        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()
```

****

THANK YOU