

# **SMART AGRICULTURE SYSTEM BASED ON IoT**

## **1.INTRODUCTION**

### **OVERVIEW:**

In olden Days Farmers used to figure the ripeness of soil and influenced suspicions to develop which to kind of yield. They didn't think about the humidity, level of water and especially climate condition which terrible a farmer increasingly. The Internet of things (IoT) is remodeling the agribusiness empowering the agriculturists through the extensive range of strategies. This project involves building a smart Internet of Things based agriculture system to monitor the weather conditions and soil conditions and help the farmer to gain better yield. This will be accomplished by using the IBM Watson IoT platform and Openweather API. We use Python language to interact with the system. The eye-catching features of this project include smart irrigation with smart control based on real time field data. Secondly temperature maintenance, humidity maintenance and other environmental parameters. And finally the recommendation to farmer for smart agriculture.

### **PURPOSE:**

The aim of the project is to provide the farmers with the data regarding the weather and soil conditions through a web app. This makes farming profitable and prevents the damage of the crop in a feasible manner.

## **2.LITERATURE SURVEY**

### **EXISTING PROBLEM:**

Indian agriculture is plagued by several problems; some of them are natural and some others are manmade. Of those, being unable to predict the climate and plant

requirements is the major problem. Also some of the other problems include soil erosion, lack of mechanisation, irrigation. Soil quality testing is not done effectively in India. Moreover, not being able to adapt to technological trends has become fatal to agriculture.

### ***PROPOSED SOLUTION:***

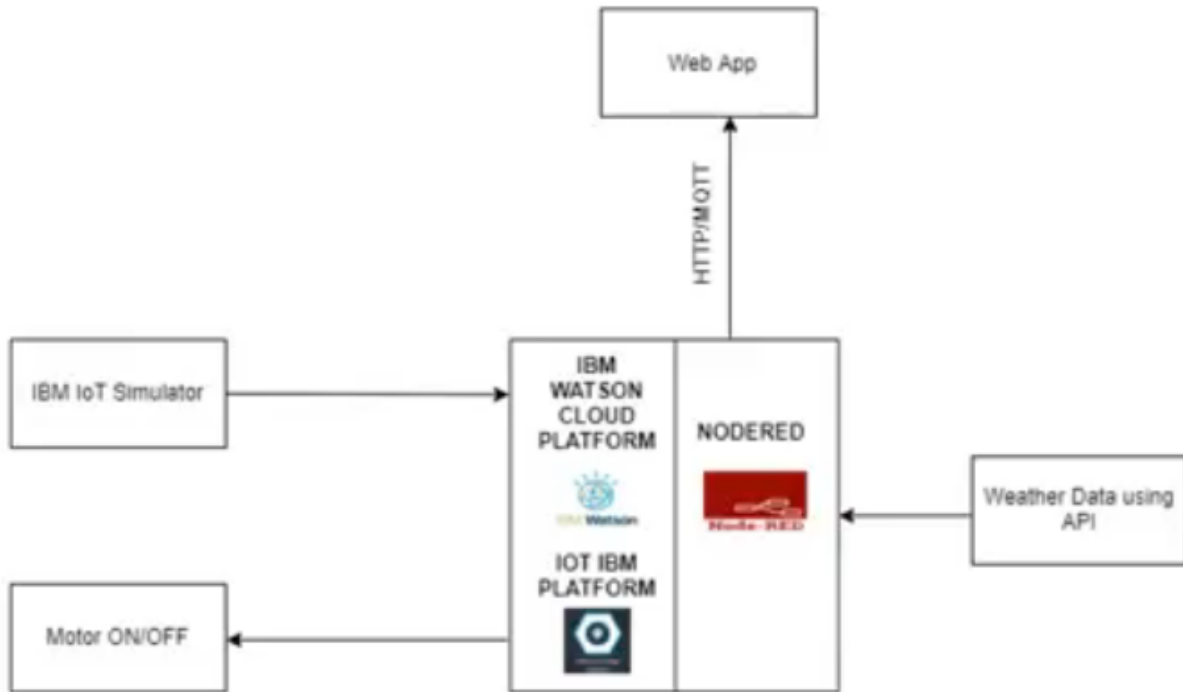
There are many new farming techniques like precision farming, GSM based farming and smart irrigation control. Here, we propose a solution using cloud and IoT to monitor the soil and weather conditions. Temperature, humidity and soil moisture sensors are used to obtain the necessary information and push them to the cloud platform. Further we create a web page which is accessed by the farmers to monitor their crop.

### ***3.THEORETICAL ANALYSIS***

#### ***PROJECT SCOPE:***

We create a device in the IBM Watson IoT platform and enable simulation. The simulation is done in the watson IOT sensor simulator. The sensors take reading every minute and upload to the cloud. Node-red is used to wire together the hardware,online services and APIs. To simulate weather information , we create an account in Openweather.org and provide through the sensors. Later, these are used through a web interface to control the motor.

#### ***BLOCK DIAGRAM:***



## ***HARDWARE/SOFTWARE DESIGN:***

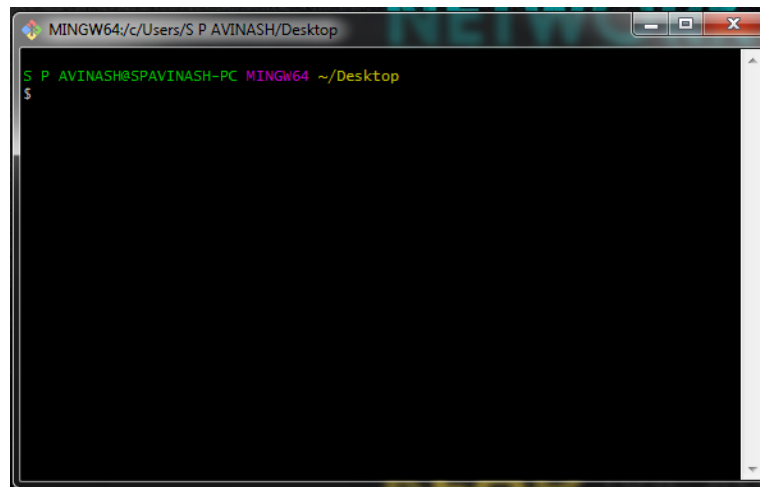
1. Install the required tools and create the required accoounts.

### **SETTING UP IDE:**

We use Python IDE for the project.

A screenshot of a 'Python 3.7.0 Shell' window. The window has a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main text area displays the following information: 'Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32', followed by 'Type "copyright", "credits" or "license()" for more information.', and a prompt '>>>' with a cursor.

## INSTALL GIT:



## 2.CREATE A DEVICE IN THE IBM WATSON IOT PLATFORM:

We need to create two devices in the platform . One acts as a processor for the sensor information and the other is an instance of a motor.

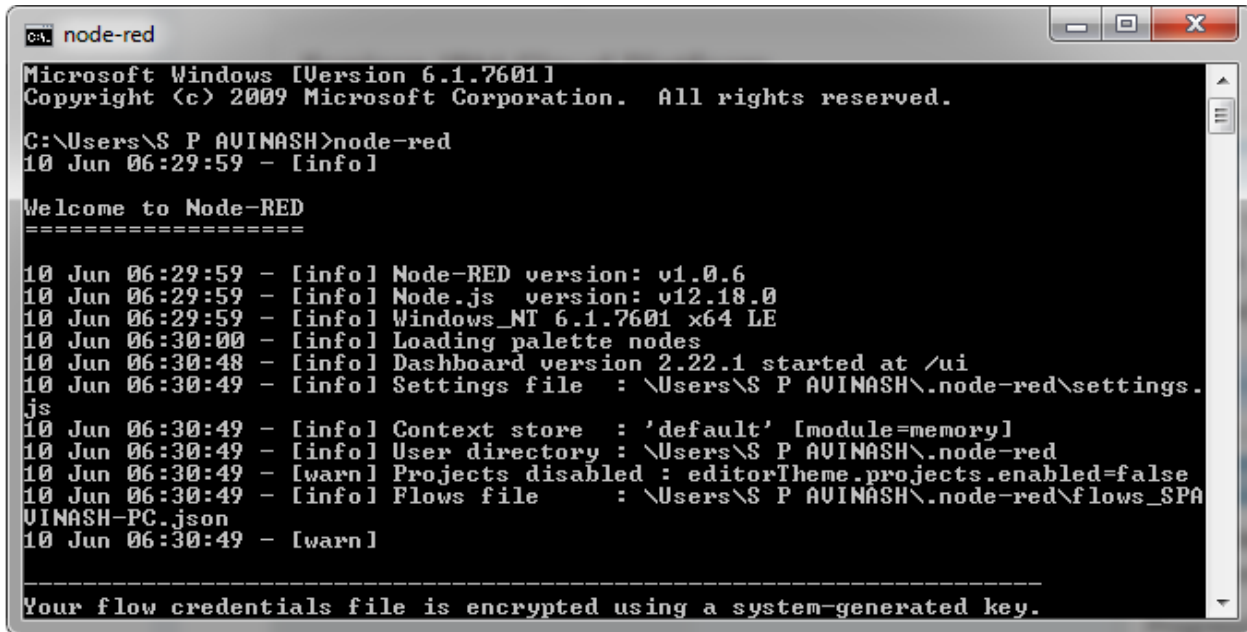
Also create an API key-token pair and save in a safe place.

A screenshot of the IBM Watson IoT Platform web interface. The page title is "Browse Devices". It features a sidebar with navigation icons and a top navigation bar with "Browse", "Action", "Device Types", and "Interfaces". A blue "Add Device" button is in the top right. Below the title, there are "All Devices" and "Diagnose" tabs. A descriptive paragraph explains that the table shows a summary of added devices. Below this is a search bar and a "Device Simulator" toggle. The main content is a table with columns: Device ID, Status, Device Type, Class ID, and Date Added. It lists two devices: an "Arduino" (Disconnected) and a "Motor" (Connected). At the bottom, there is a pagination bar showing "1 of 1 page" and "1" items per page.

	Device ID	Status	Device Type	Class ID	Date Added
>	Arduino	Disconnected	IOTDevice	Device	May 25, 2020 8:26 PM
>	Motor	Connected	IOTDevice	Device	Jun 10, 2020 12:04 AM

### 3.INSTALL NODE-RED LOCALLY:

Node-red is a Node.js based implementation and can be installed using chocolatey or yarn.



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\S P AVINASH>node-red
10 Jun 06:29:59 - [info]

Welcome to Node-RED
=====

10 Jun 06:29:59 - [info] Node-RED version: v1.0.6
10 Jun 06:29:59 - [info] Node.js version: v12.18.0
10 Jun 06:29:59 - [info] Windows_NT 6.1.7601 x64 LE
10 Jun 06:30:00 - [info] Loading palette nodes
10 Jun 06:30:48 - [info] Dashboard version 2.22.1 started at /ui
10 Jun 06:30:49 - [info] Settings file : \Users\S P AVINASH\.node-red\settings.
js
10 Jun 06:30:49 - [info] Context store : 'default' [module=memory]
10 Jun 06:30:49 - [info] User directory : \Users\S P AVINASH\.node-red
10 Jun 06:30:49 - [warn] Projects disabled : editorTheme.projects.enabled=false
10 Jun 06:30:49 - [info] Flows file : \Users\S P AVINASH\.node-red\flows_SPA
VINASH-PC.json
10 Jun 06:30:49 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.
```

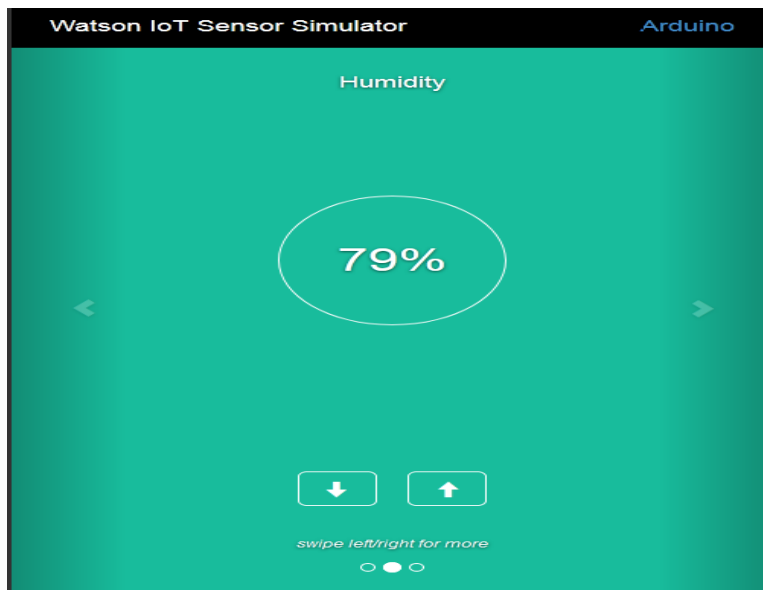
### 4.CONNECT THE IOT DEVICE TO A SIMULATOR

The iot sensor simulator requires the following information.

- 1.Device organization
- 2.Device Type
- 3.Device Id
- 4.Device Token

Once given all the required instructions, the sensor is connected to the device we created.

This can be verified by the device name in blue colour on top right.



## 5.VISUALISING THE DATA:

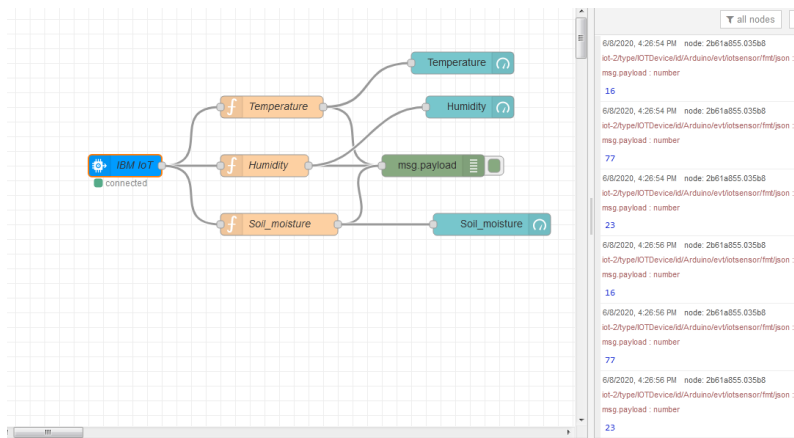
1. Once the device is connected, we can create boards in the IBM IoT platform.  
Boards-->Create new board--> Add Cards
- Different types of visualizations like line plot,gauge can be done in the IBM platform.

## 6.CONNECTING TO OPENWEATHER API:

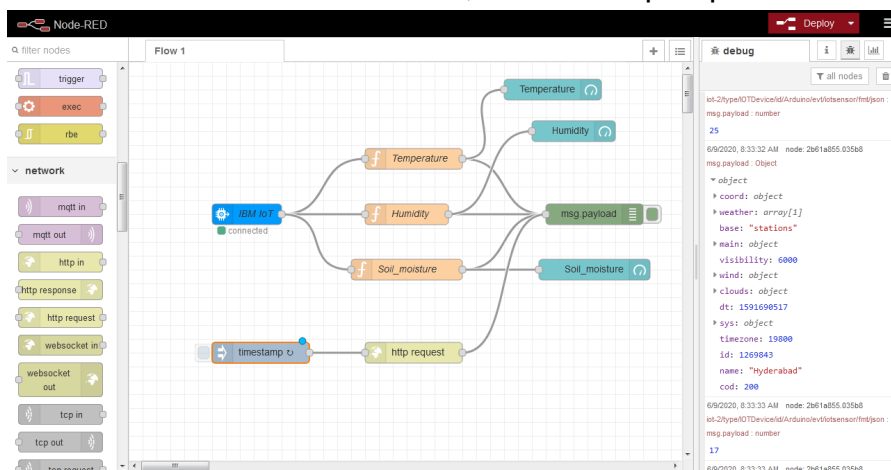
1. Create an account in openweather.org
2. Go to API marketplace and register a API key
3. Select "By city Name' in API options
4. Copy and save the API call to be used later

## 7.CREATING THE UI USING NODE-RED:

1. We need to create flows for our purposes.
2. We need to install the ibm iot node package and ibm watson package. To install a new node-set,  
**manage palette-->install-->search for required nodes**
3. Firstly, to take the input data from the sensor we make the following flow .
4. This also creates a UI for the user to interact with the data and the devices.



1. To view the data from API,we use a http request node and inject node.

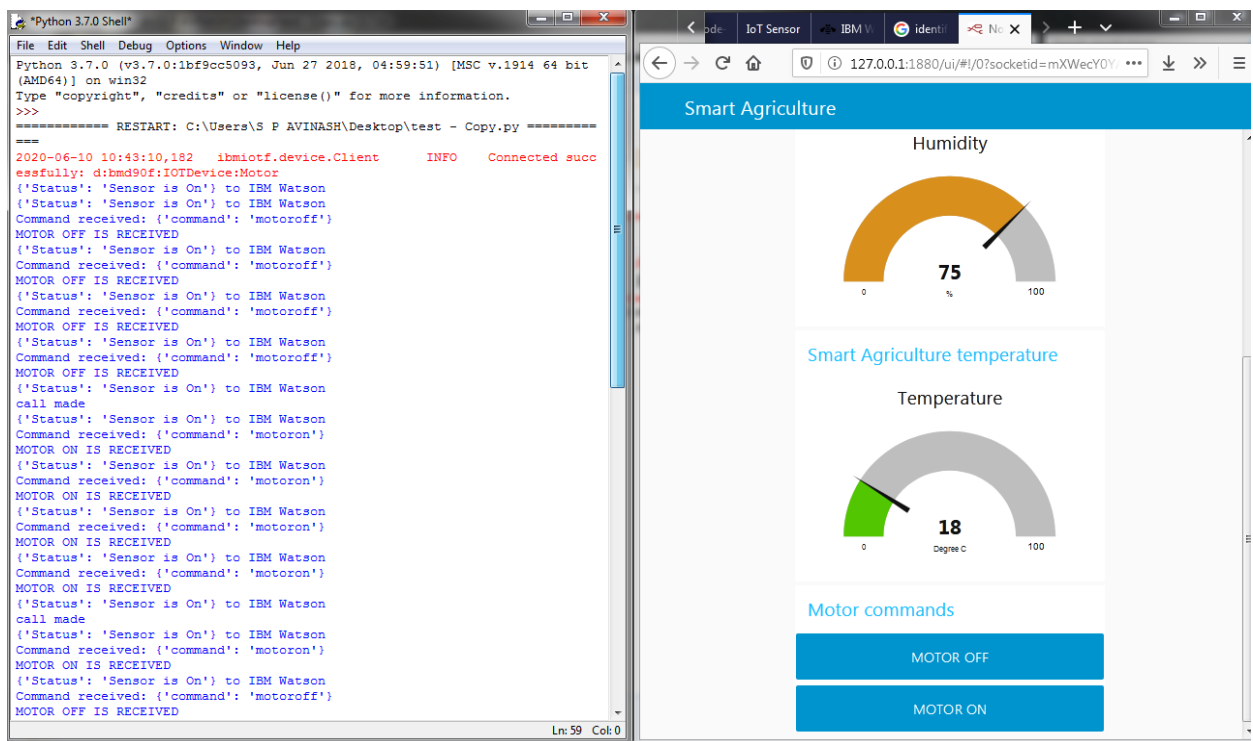
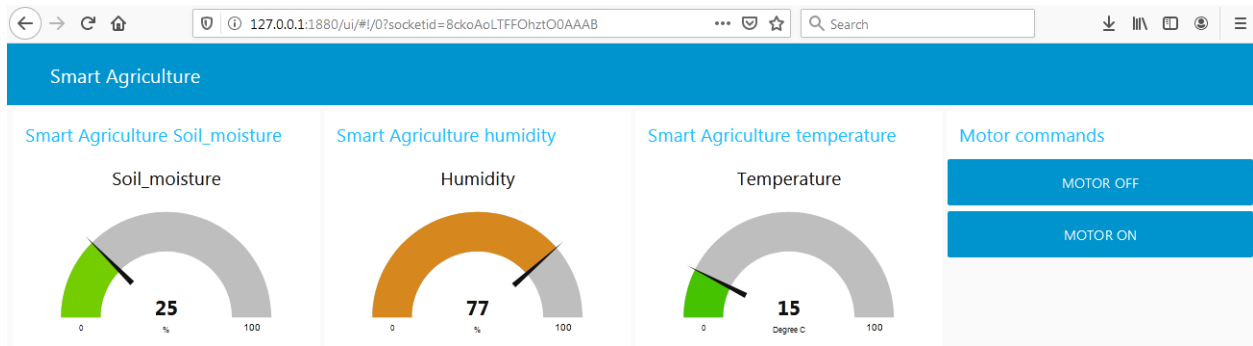


1. To give the instructions to the device,in our case motor we create another flow using the ibm out node and Motor ON and OFF buttons.

## 8.RUNNING THE PYTHON CODE AND TURNING MOTOR ON OR OFF:

We use the python programming language to interact with the devices and the cloud. On clicking the button in the UI,the motor can be made ON or OFF.

## RESULTS:





**FUTURE SCOPE:**

The project can be further extended to enabling the usage of AI in the agriculture ecosystem. We can also integrate the system using solar pannels which replace the conventional electricity methods. We can suggest crops based on the climatic conditions of the data. Based on the water level, we can alert the farmer or automatically turn the motor off.