

PROJECT SCOPE

Project Name	
Smart Irrigation System Using Weather Forecasting Api	
Project Manager	VIKAS KUMAR NIGAM
Project Sponser	SmartBridge
Kickoff Date	1/08/2021

Project Summary

Farming is a major input sector for economic development of any country. Livelihood of majority of population of the country like India depends on agriculture. In this project, it is proposed to develop a Smart Farming System that uses advantages of cutting edge technologies such as IoT, Wireless Sensor Network and Cloud computing to help farmers enhance the way farming is done.

Using sensors like temperature, humidity, soil moisture etc. are used to get information about the field and help farmers to take precise decisions on insights such as turning ON/OFF the pumping Motor and recommendations based on the collected data.

Need of Project

Measuring soil moisture is important in agriculture to help farmers manage their irrigation systems more efficiently. Not only are farmers able to generally use less water to grow a crop, they are able to increase yields and the quality of the crop by better management of soil moisture during critical plant growth stages. Embedded system for automatic irrigation of an agriculture field offers a potential solution to support site-specific irrigation management that allows producers to maximize their productivity while saving the water.

Project Requirements

Github
Slack Channel
Zoho Writer
IBM Cloud
IBM Watson IoT Simulator
Open source whether API platform
MIT App Inventor
Python IDE

Technical Requirements

Node-Red App
IBM Watson IoT
Cloud Service : Cloudant
Slack Channel for communication with mentor
MIT App Inventor

Software Requirements

Chrome Web Browser
Slack Bot
Python 3 IDE

Project Deliverables

The scope of the project is to build an Smart Agriculture system based on IoT and to advance the farmers to control thier irrigation work remotely through mobile.

Project Team

Individual Project

Project Schedule

Project Planning & Kickoff	1 Day
Setup of Development Environment	1 Day
Exploring IBM Cloud platform	3 Days
Exploring IBM Watson Service	2 Day
Setup of Node-red, Watson IoT with IoT simulator & Open whether API	2 Days
Building a Web App	3 Days
Building Mobile App through MIT	2 Days
Developing Python Code	3 Days

Project Duration 15 Days

From 18/07/2021 to 3/08/2021

TASKS COMPLETED

Project Scope Document :

Project planning is completed by developing the schedule, requirements needful and Deliverables of the project.

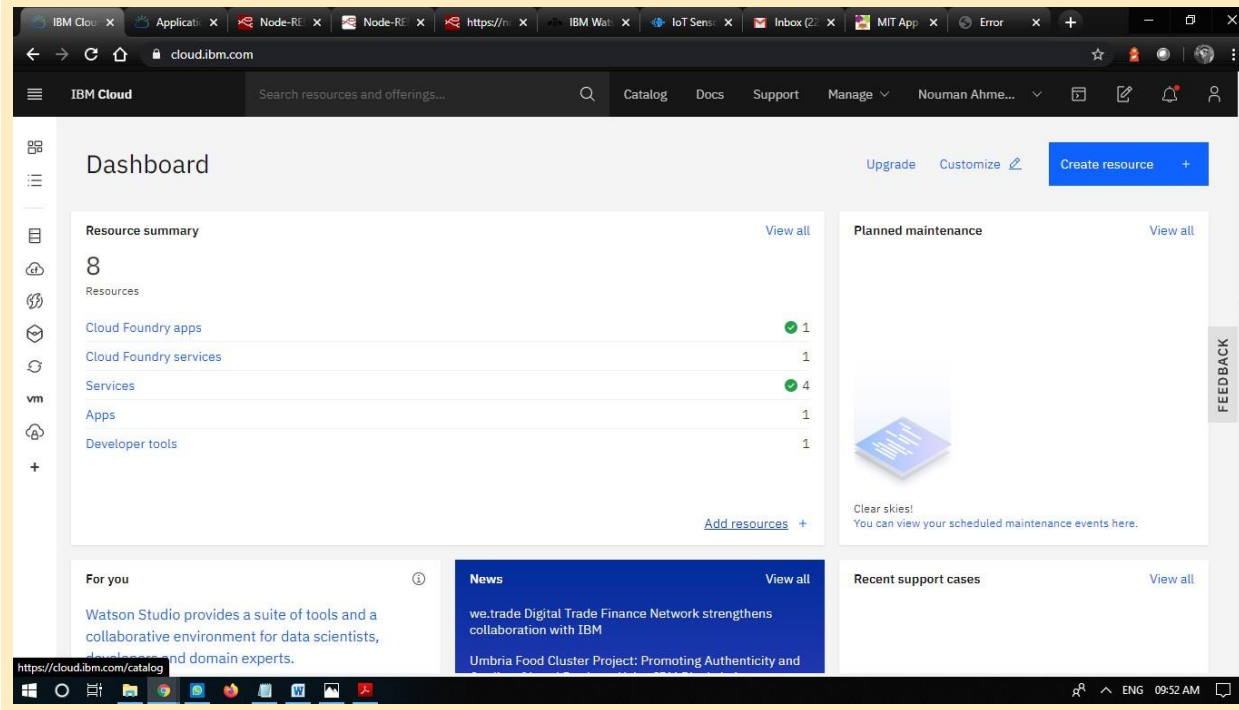
Setup of Development Environment :

- ✓ Git Hub account is created and the smartschool practice git repository is linked with the account.
 - ✓ Slack account is created and joined the slack channel of our IoT6 internship group.
 - ✓ Got familiar working with zoho writer.
-



Explore IBM Cloud platform:

- ✓ IBM Cloud academic initiative account is created with SmartInternz Email ID.
- ✓ Node red is Installed succesfully.
- ✓ IBM Watson IoT is created and all the credentials are connected with the IOT simulator.



The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A search bar labeled 'Search by Device ID' is present. The main content area shows a table of devices with columns: Device ID, Status, Device Type, Class ID, and Date Added. One device is listed with ID '555777', status 'Connected', and type 'Nodemcu'. Below the table, the 'Recent Events' tab is selected, showing a list of events with columns: Event, Value, Format, and Last Received. The events are from 'iotsensor' and contain JSON data. A 'Device Simulator' toggle is visible in the top right of the device details section.

Device ID	Status	Device Type	Class ID	Date Added
555777	Connected	Nodemcu	Device	6 Jun 2020 14:41

Event	Value	Format	Last Received
iotsensor	{"d":{"name":"555777","temperature":26,"humi..."	json	a few seconds ago
iotsensor	{"d":{"name":"555777","temperature":26,"humi..."	json	a few seconds ago
iotsensor	{"d":{"name":"555777","temperature":26,"humi..."	json	a few seconds ago
iotsensor	{"d":{"name":"555777","temperature":26,"humi..."		
iotsensor	{"d":{"name":"555777","temperature":26,"humi..."		

Configuration of Node-red with Open Weather API:

- ✓ An account in Open weather website is created and API is set up.
- ✓ Weather URL : `api.openweathermap.org/data/2.5/weather?q=Vellore, IN&units=metric&appid=a0bfd7e12cd43d39a44c301354137ea5`
- ✓ With the help of http request node, Live Forecasting Weather data is obtained.

Smart Irrigation System Using Weather Forecasting Api

8

The image shows two screenshots of a web browser. The top screenshot is the OpenWeather API keys management page. The bottom screenshot shows the JSON response of a weather API call.

OpenWeather API Keys Page:

URL: home.openweathermap.org/api_keys

Navigation: New Products, Services, **API keys**, Billing plans, Payments, Block logs, My orders, My profile

Message: You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them.

Key	Name	Create key
a0bfd7e12cd43d39a44c301354137ea5	Default	<input type="text" value="API key name"/> <button>Generate</button>

Product Collections: Current and Forecast APIs, Historical Weather Data, Weather Maps, Widgets

Subscription: How to start, Pricing, Subscribe for free

About us: OpenWeather Ltd is a British-based tech company that provides weather and satellite data worldwide. OpenWeather collects and processes raw data from a variety of sources, and gives its customers access to the data archive.

API Call Response:

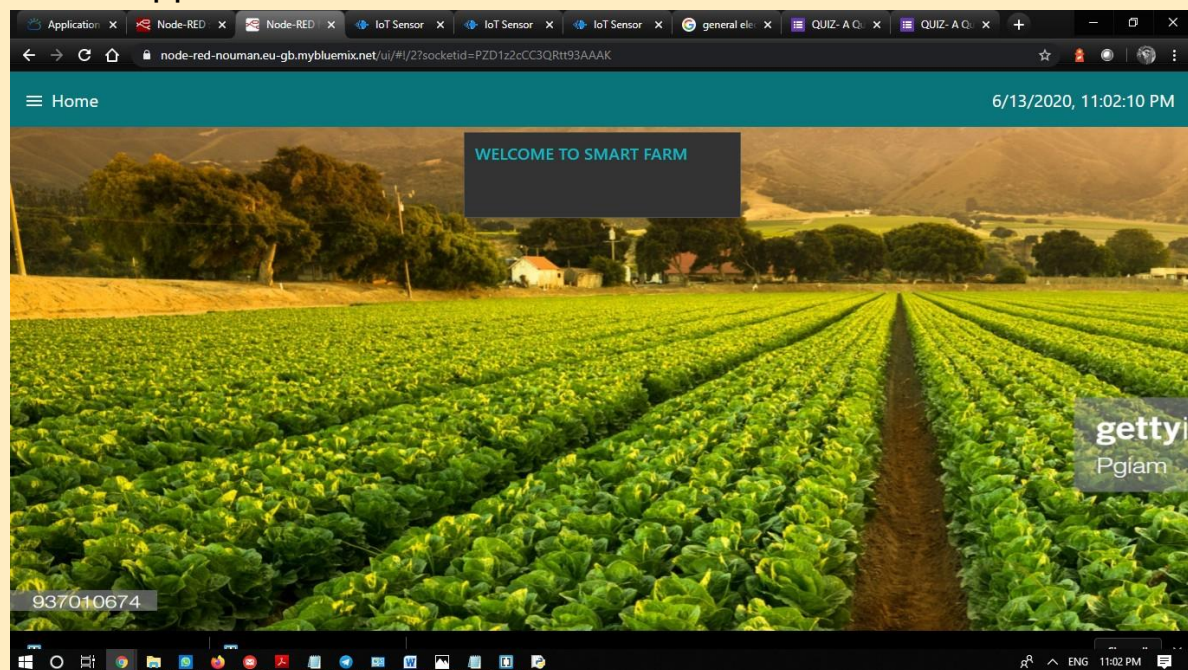
URL: api.openweathermap.org/data/2.5/weather?q=Mysore.%20%20IN&units=metric&appid=a0bfd7e12cd43d39a44c301354137ea5

```
{
  "coord": {
    "lon": 76.65,
    "lat": 12.31
  },
  "weather": [
    {
      "id": 802,
      "main": "Clouds",
      "description": "scattered clouds",
      "icon": "03n"
    }
  ],
  "base": "stations",
  "main": {
    "temp": 28,
    "feels_like": 28.09,
    "temp_min": 28,
    "temp_max": 28,
    "pressure": 1010,
    "humidity": 65,
    "visibility": 6000,
    "wind": {
      "speed": 5.7,
      "deg": 270
    },
    "clouds": {
      "all": 40
    },
    "dt": 1591622467,
    "sys": {
      "type": 1,
      "id": 9212,
      "country": "IN",
      "sunrise": 1591576067,
      "sunset": 1591622246,
      "timezone": 19800,
      "id": 1262321,
      "name": "Mysore",
      "cod": 200
    }
  }
}
```

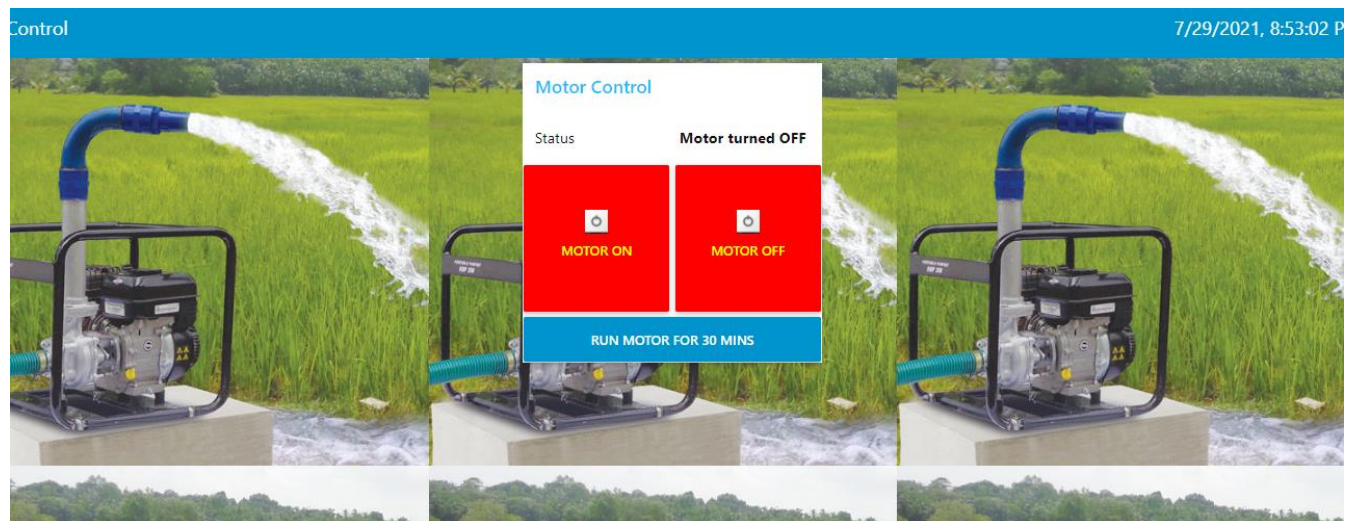

Building Web Application :

- ✓ A User Interface(UI) is developed using Node red dashboard nodes.
- ✓ Buttons are configured for sending commands to IoT platform.

UI WebApplication:







```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\HP\OneDrive\Desktop\ibm-project\project.py =====
2021-07-29 20:29:16,837 ibmiotf.device.Client INFO Connected successfully: d:7jh6o2:VITDevice:12345
Command received: {'command': 'Motor turned ON'}
Command received: {'command': 'Motor turned OFF'}
Command received: {'command': 'Running Motor for 30 minutes'}
Command received: {'command': 'Motor turned ON'}
Command received: {'command': 'Running Motor for 30 minutes'}
Command received: {'command': 'Motor turned OFF'}
Command received: {'command': 'Motor turned OFF'}
Command received: {'command': 'Running Motor for 30 minutes'}
Command received: {'command': 'Running Motor for 30 minutes'}
Command received: {'command': 'Motor turned OFF'}
```