

PROJECT SCOPE

Project Name	Smart Agriculture System Based on IoT
Project Manager	Nouman Ahmed
Project Sponser	SmartBridge
Kickoff Date	22/05/2020

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

28 Days

From

22/05/2020 to 18/06/2020

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 spository 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.
-

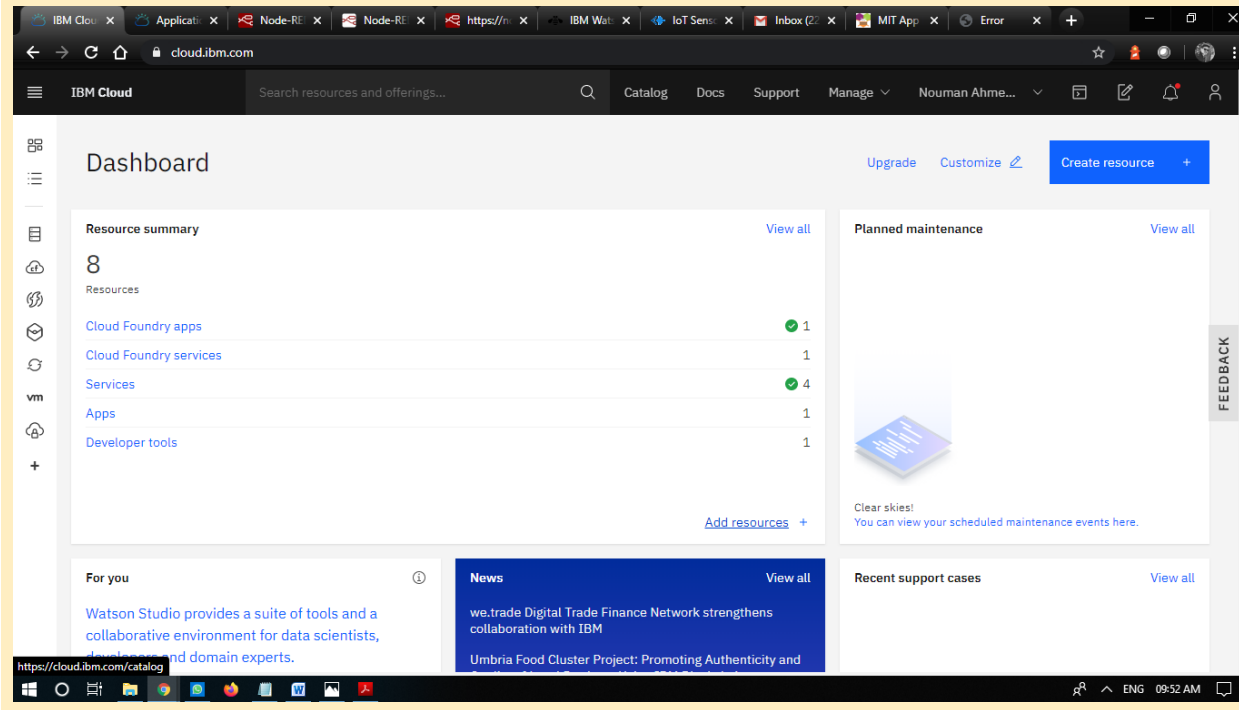
IISPS_INT_2308_Smart Agriculture System Based on IoT

5

The image displays two screenshots from a computer screen. The top screenshot shows a web browser at the GitHub repository page for 'SmartPracticeschool / IISPS-INT-2308-Smart-Agriculture-system-based-on-IoT'. The page features a large blue banner with the text 'Learn Git and GitHub without any code!' and a green 'Read the guide' button. Below the banner, the repository name is shown with 1 watch, 0 stars, and 0 forks. The 'Code' tab is selected, showing a 'Quick setup' section with links for 'Set up in Desktop', 'HTTPS', and 'SSH'. The bottom screenshot shows a Slack chat window for the '#internship' channel. The chat history shows a message from Hanisha Dhananjaya at 12:06 AM saying 'but still there is no output' with an attached image of a flowchart. Below it, a message from Snehasish Pradhan at 2:03 AM says '@Durgaprasad Sir, I have been shifting modules to review stage as soon as I am completing each...but they are not being shifted to complete stage ...Please guide'. The Slack interface includes a sidebar with channel and direct message lists, and a bottom status bar.

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 is present with the text 'Search by Device ID'. The main content area shows a table of devices with columns: Device ID, Status, Device Type, Class ID, and Date Added. The device '555777' is highlighted with a status of 'Connected' and a device type of '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 message at the bottom indicates '0 Simulations running'.

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=Mysore,IN&units=metric&appid=a0bfd7e12cd43d39a44c301354137ea5`
- ✓ With the help of http request node, Live Forecasting Weather data is obtained.

The screenshot shows the OpenWeather API keys management interface. The top navigation bar includes links for Get Started, API, Pricing, Maps, Partners, Blog, Marketplace, and a user profile. The main content area has a tab for API keys, showing a message about generating keys and a table with one key named 'Default'. A 'Create key' form is also visible. Below the table, there are sections for Product Collections, Subscription, and About us.

API Keys Table:

Key	Name
a0bfd7e12cd43d39a44c301354137ea5	Default

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.

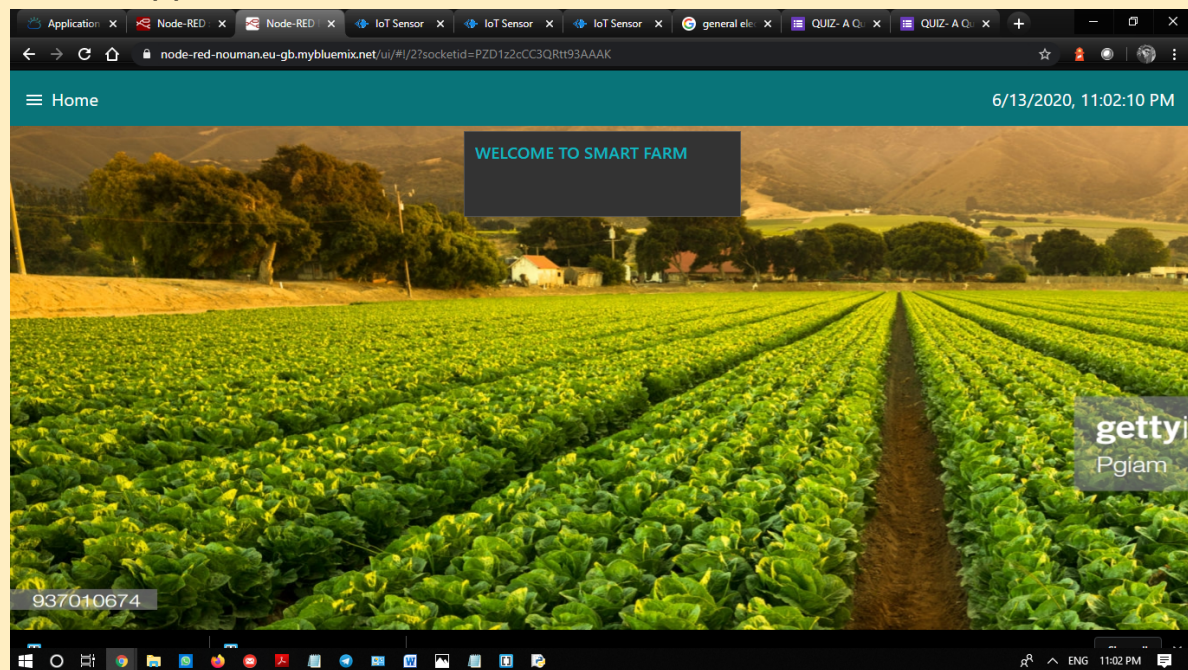
The bottom part of the screenshot shows a JSON response from the API endpoint `api.openweathermap.org/data/2.5/weather?q=Mysore%20%20IN&units=meter&appid=a0bfd7e12cd43d39a44c301354137ea5`.

```
{
  "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": 16000,
    "wind": {
      "speed": 5.7,
      "deg": 127
    },
    "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 :

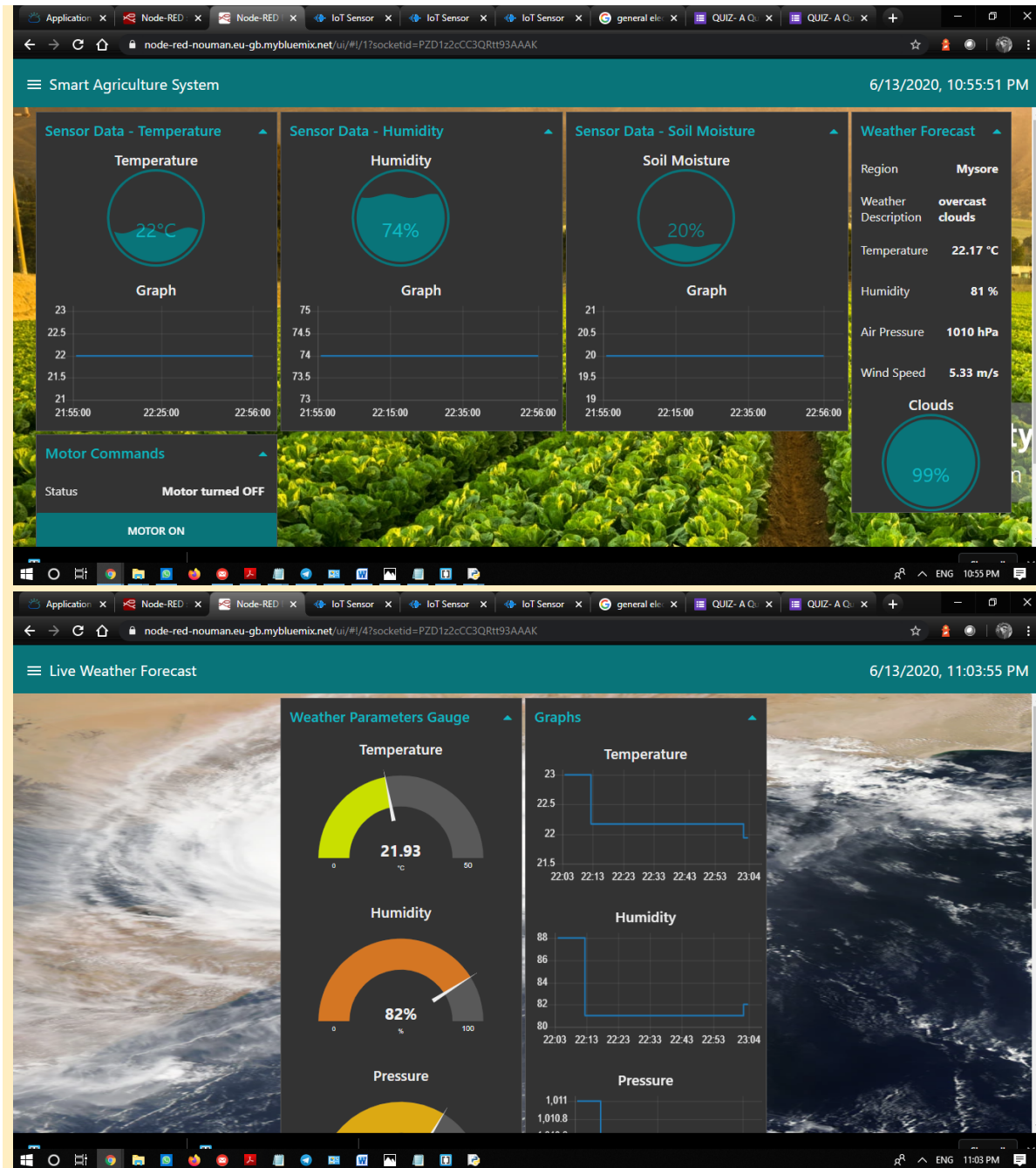
- ✓ A User Interface(UI) is developed using Node red dashboard nodes.
- ✓ Buttons are configured for sending commands to IoT platform.

UI WebApplication:



IISPS_INT_2308_Smart Agriculture System Based on IoT

10



Motor Commands Recieved Locally:

IISPS_INT_2308_Smart Agriculture System Based on IoT

1

```
Python 3.8.3 Shell
File Edit Shell Debug Options Window Help
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/ZeeShanAhmed/AppData/Local/Programs/Python/Python38/Scripts/Smart Agriculture System Motor Commands.py
2020-06-19 22:29:25,297 ibmiotf.device.Client INFO Connected successfully: d:mrcy9a:MOTOR:77777
Command received: {'command': 'Motor turned ON'}
MOTOR ON IS RECEIVED
Command received: {'command': 'Motor turned OFF'}
MOTOR OFF IS RECEIVED
Command received: {'command': 'Running Motor for 30 minutes'}
MOTOR ON FOR 30 MINUTES IS RECEIVED
Command received: {'command': 'Motor turned OFF'}
MOTOR OFF IS RECEIVED
Command received: {'command': 'Running Motor for 30 minutes'}
MOTOR ON FOR 30 MINUTES IS RECEIVED
Command received: {'command': 'Running Motor for 30 minutes'}
MOTOR ON FOR 30 MINUTES IS RECEIVED
Command received: {'command': 'Running Motor for 30 minutes'}
MOTOR ON FOR 30 MINUTES IS RECEIVED
Command received: {'command': 'Running Motor for 30 minutes'}
MOTOR ON FOR 30 MINUTES IS RECEIVED
Command received: {'command': 'Motor turned ON'}
MOTOR ON IS RECEIVED
Command received: {'command': 'Motor turned ON'}
MOTOR ON IS RECEIVED
Command received: {'command': 'Motor turned OFF'}
MOTOR OFF IS RECEIVED
Command received: {'command': 'Motor turned OFF'}
MOTOR OFF IS RECEIVED
```

Ln 1 Col 79