Project

Topic: Smart Agriculture System based on IoT

Submitted By:

INDIREDDY SRINIVASULA REDDY

INTRODUCTION:

1.1 OVERVIEW:

The objectives of this report are to proposed IoT based Smart Farming 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. The structure of the report is as follows: It will cover over of overview of IoT Technology and agriculture-concepts and definition, IOT enabling technologies, IOT application in agriculture, benefits of IOT in agriculture and IOT and agriculture current scenario and future forecasts, it will cover definition of IOT based smart farming system , the components and modules used in it and working principal of it, it also covers algorithm and flowchart of the overall process carried out in the system and its final graphical output .At last it consists of conclusion, future scope and reference and source code.

1.2 PURPOSE:

It is completely depending upon the monitoring the agriculture fields by the farmers when they are far from their fields.

- Need for technology to monitor the important parameters like soil moisture, temperature, humidity etc. to improve the cultivation process.
- The weather monitor is fixed for the particular location.
- Due to the IoT based technology it makes work and monitor faster by the farmers.

LITERATURE SURVEY:

2.1 EXISTING PROBLEM:

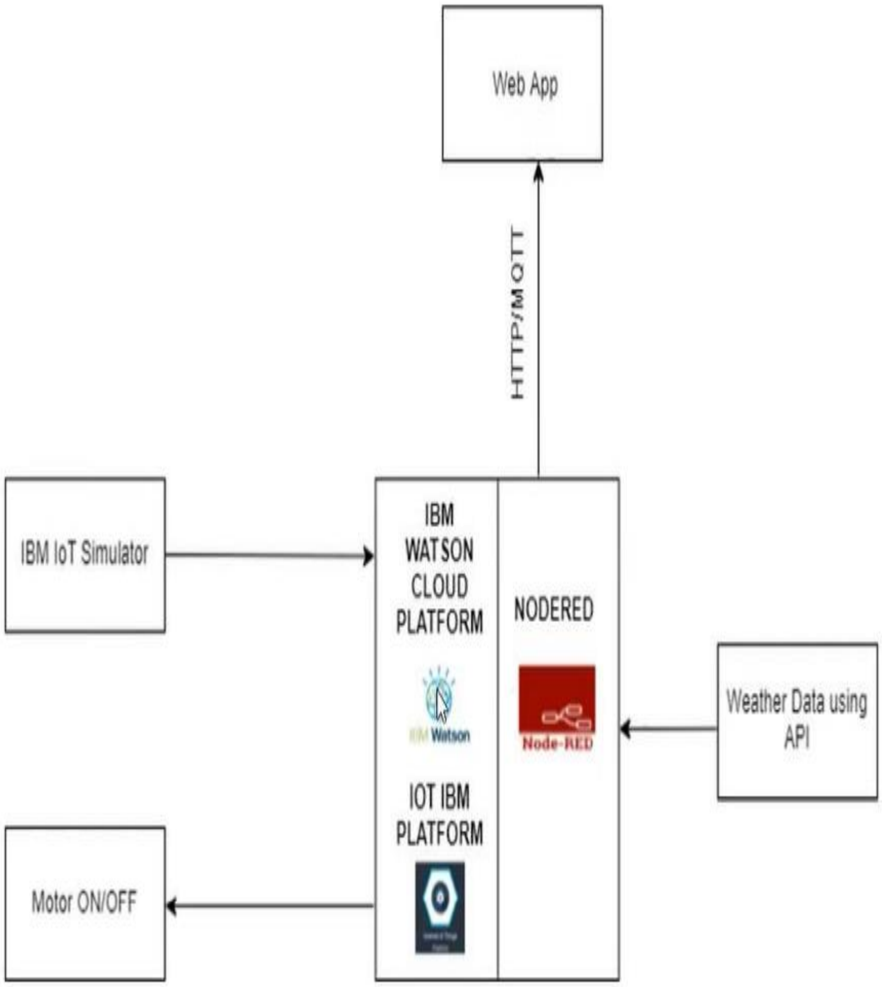
The global population is predicted to touch 9.6 billion by 2050 – this poses a big **problem** for the **agriculture** industry. Despite combating challenges like extreme weather conditions, rising climate change, and farming's environmental impact, the demand for more food has to be met. To meet these increasing needs, **agriculture** has to turn to new technology.
New **smart farming** applications based on **IOT** technologies will enable the **agriculture** industry to reduce waste and enhance productivity. It is the application of modern ICT (Information and Communication Technologies) into agriculture. In IOT-based smart farming, a system is built for monitoring the crop field with the help of sensors (light, humidity, temperature, soil moisture, etc.). The farmers can monitor the field conditions from anywhere

2.2 PROPOSED SOLUTION:

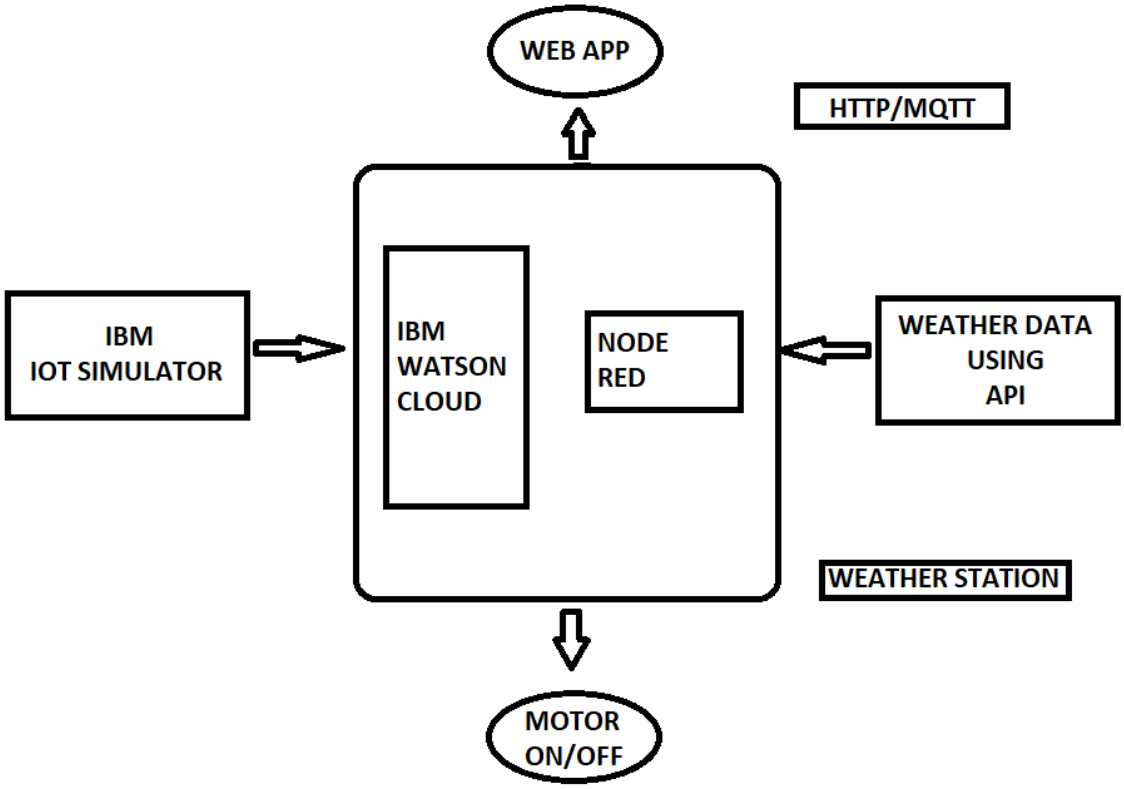
Propose a **Smart Farming IOT** based **Agriculture System** assisting farmers in getting Live Data (Temperature, Humidity, Soil Moisture, etc.) for efficient environment monitoring which will enable them to do **smart farming** and improve their overall yield and quality of products.
The **agriculture** system **proposed** in this paper is integrated with Arduino Technology with various sensors and live information feed can be obtained online from IBM IoT Simulator and by the help of node red. The product being proposed is tested on Live Agriculture Fields giving high accuracy in data feeds.

THEORITICAL ANALYSIS:

3.1 BLOCK DIAGRAM:



3.2 SOFTWARE DESIGNING:



EXPERIMENTAL INVESTIGATION:

4.1 SETTING THE DEVICE IN IoT PLATFORM IN IBM CLOUD:

Step1: After logging into the system a dashboard will appear and in search panel type IBM IoT Platform.

Step2: Select the London option from drop down list and click on create.

Step3: Click on the Launch Button.

Step4: Click on the Device type.

Step5: Click on the add device button.

Step6: Enter the Device name and Description and click Next.

Step7: The field of the device information can be skip and click on Finish.

Step8: Click on the Register Device.

Step9: Enter the Device Name and Click Next Button.

Step10: Skip the field data and Click Next Button.

Step11: Filling the Authentication token and Click Next Button.

Step12: Finally Summary Tab will show the device type and Device Name Information.

4.2 GENERATING THE DEVICE API:

Step1: Click on the app icon and click on Generate API Key Button

Step2: Click on the standard application from the drop-down list and Click Generate Key.

Step3: Note the API Key and Authentication token for future reference.

4.3 SETTING THE IOT SENSOR SIMULATOR:

Step1: Enter the details like Organisation ID, Device Type, Device ID and Device Token.

4.4 SETTING UP THE UI USING NODE-RED:

Step1: Select the IBM IoT in node red from the pallet.

Step2: Double click the IBM IoT, Select the API option from the Drop Down and click the Device Event.

Step3: Click the pencil key icon in the API Key.

Step4: Enter the API Key, API token and click Update Button.

Step5: Click on the Done Button and Click the Deploy Button.

Step6: After Deploying Connection Indication will be highlighted in the IBM IoT node.

Step7: Place the Debug node in the flow Editor and Click on Deploy to see the Temperature and Humidity Value in the Debug Tab.

Step8: Drag and Place the function node in the flow editor to separate the Temperature and Humidity Value.

Step9: Type **msg.payload=msg.payload.d.temperature** in the one function and Type **msg.payload=msg.payload.d.humidity** in another function to separate the Temperature and Humidity values form payload and Click deploy

Step10: Add the Gauge node from the Dashboard.

Step11: Double click the Gauge and Select the Group Name, Type the label and unit of the Gauge.

Step12: After editing the node, Click deploy button.

Step13: Output of the UI will be obtained by typing the localhost address/ui.

4.5 SETTING UP THE IBM IoT OUT SIMULATOR:

Step1: Drag and Place the IBM IoT Output Node in the Flow Editor.

Step2: Give the Device credentials and API key in the IBM IoT Output Node and Deploy it so that the Status of the IBM IoT Output Node will be in connected status.

1.Select the API Key in the Authentication.

2.Select the Option in the Output Type as Device Command and fill the Device Credentials.

Step3: Click the Pencil Icon in the Dialog Box.

Step4: Enter the API Key and API Token in the Dialog box and Click Update Button.

Step5: Click on the Deploy Button.

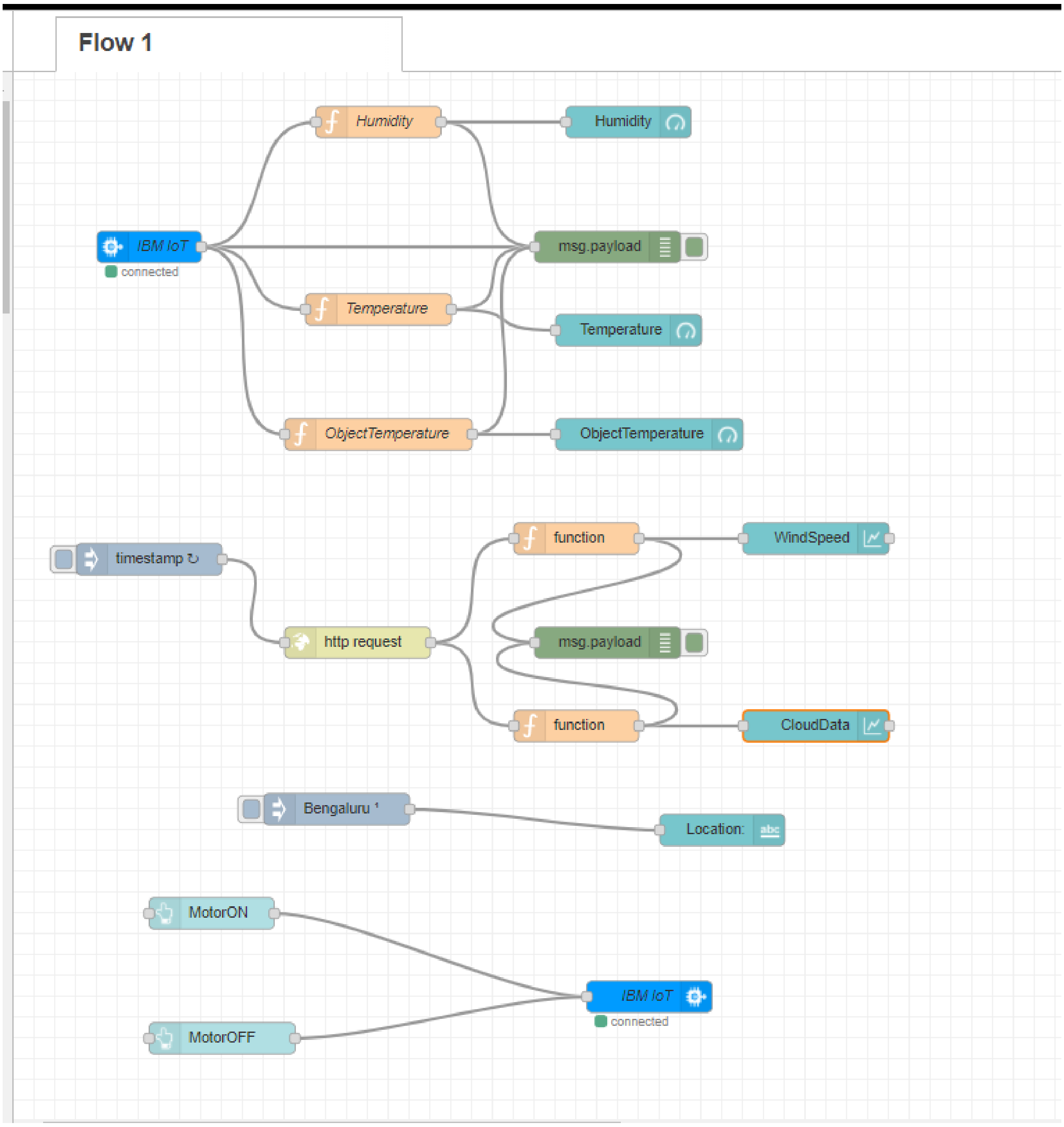
Step6: IBM IoT Output node will be shown in a connected status.

Step7: Add the Button from the dashboard and connect the button to the IBM IoT node.

Step8: Enter the Tab name, Group name for the button and Click Done Button.

Step9: Output of the UI will be obtained by typig the localhost address/ui.

Step10: The Template had been added to the node to give to give the background to the web page.



4.6 SETTING THE API FROM OPEN WEATHER:

Step1: Timestamp node will trigger the API for certain interval of time. Set the required timing for the triggering.

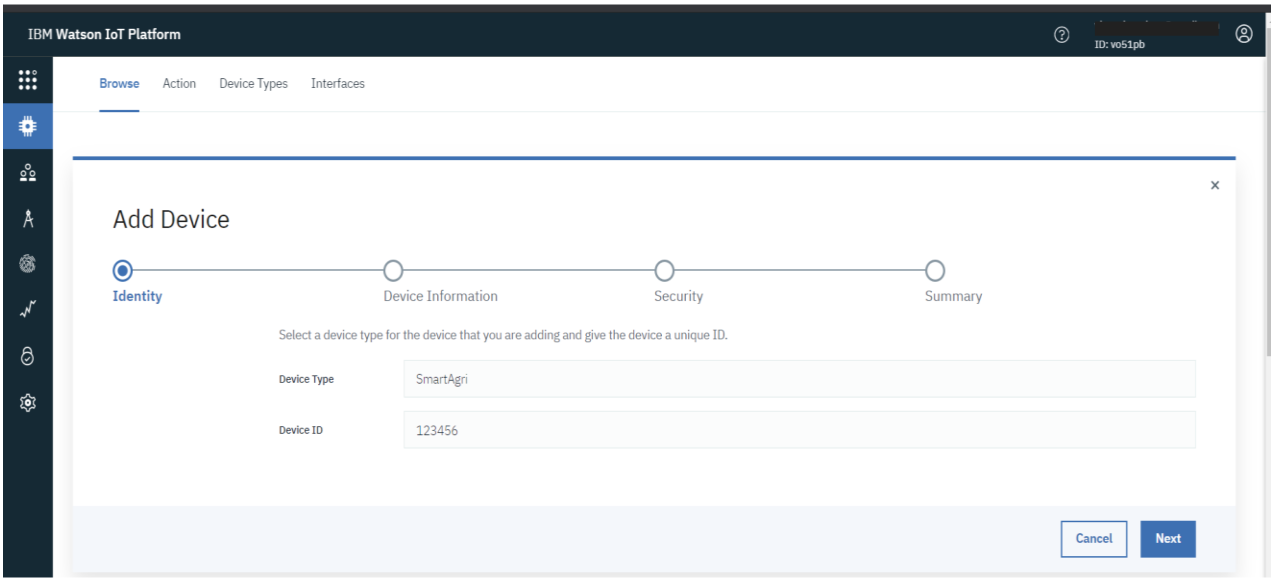
Step2: In the API node, give the API link and Return type as Jason Parson to Separate the API Data.

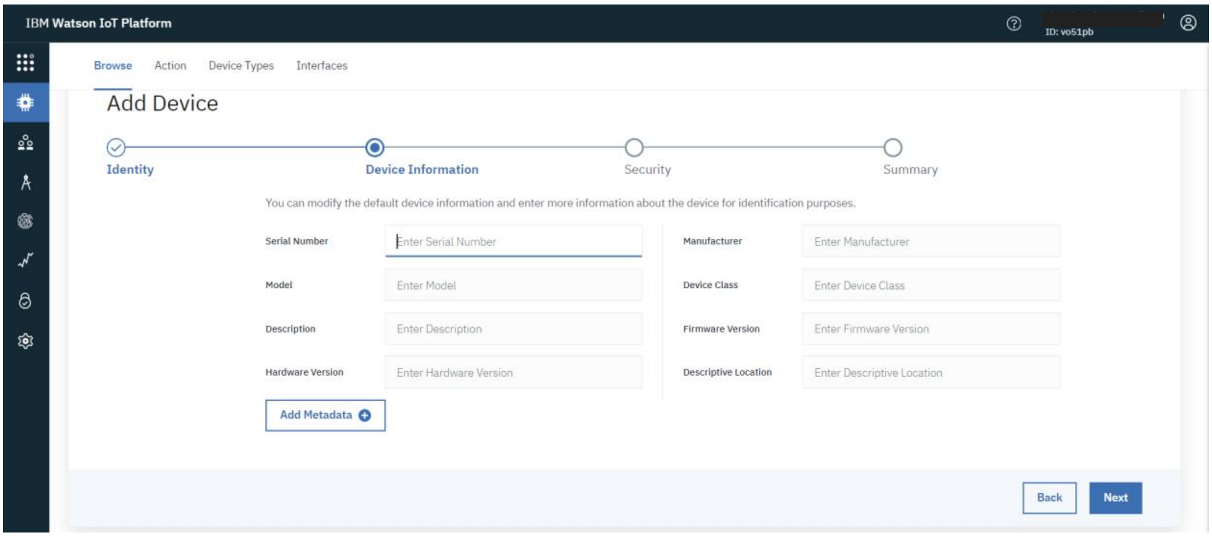
Step3: In the change node separate Temperature value from the payload by following code given in the image.

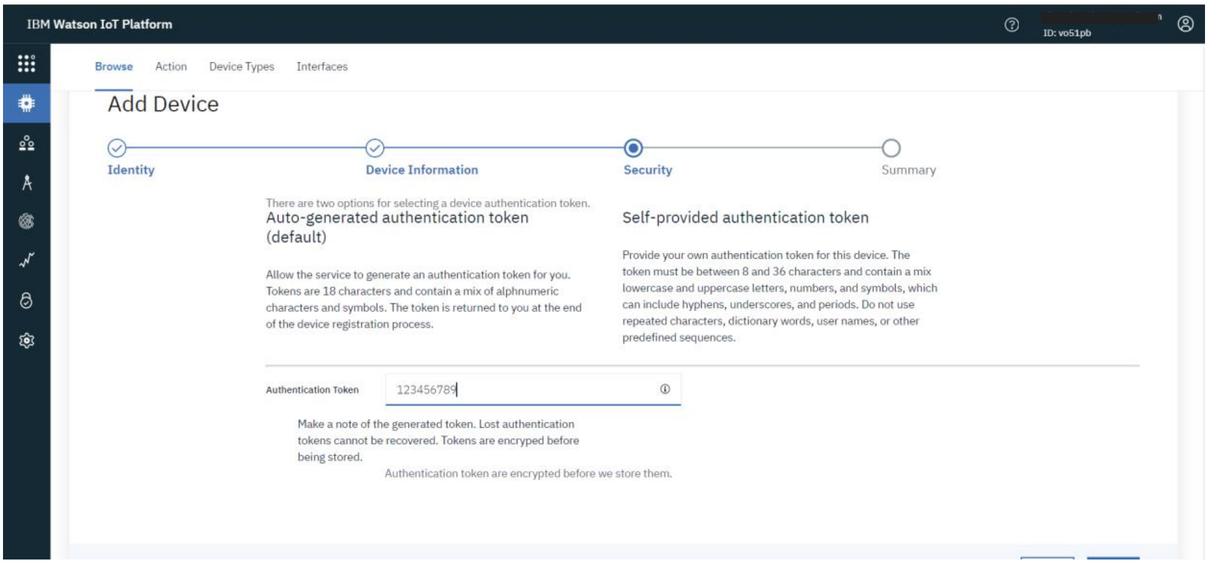
Repeat the same procedure for Windspeed and Humidity.

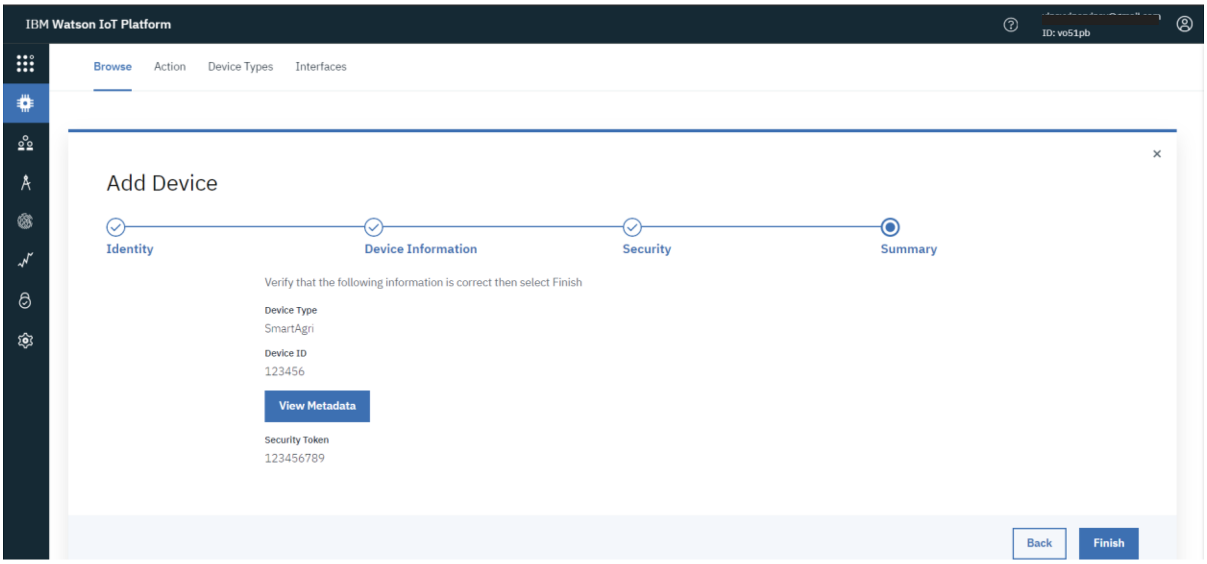
IMAGES OF THE CREATING THE ABOVE INSTRUCTIONS:

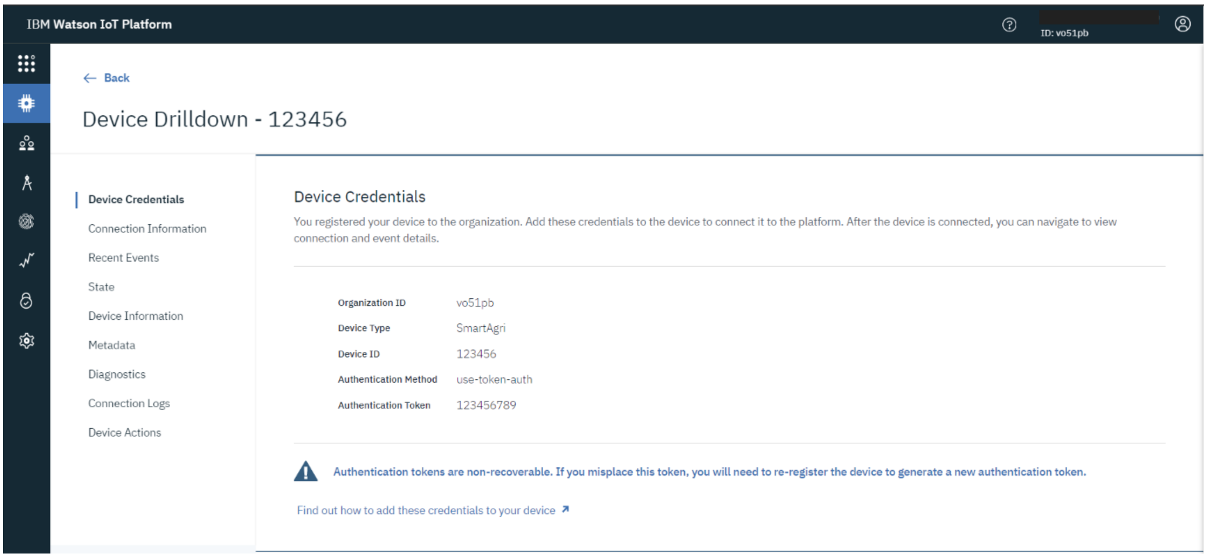
IBM IoT WATSON PLATFORM:

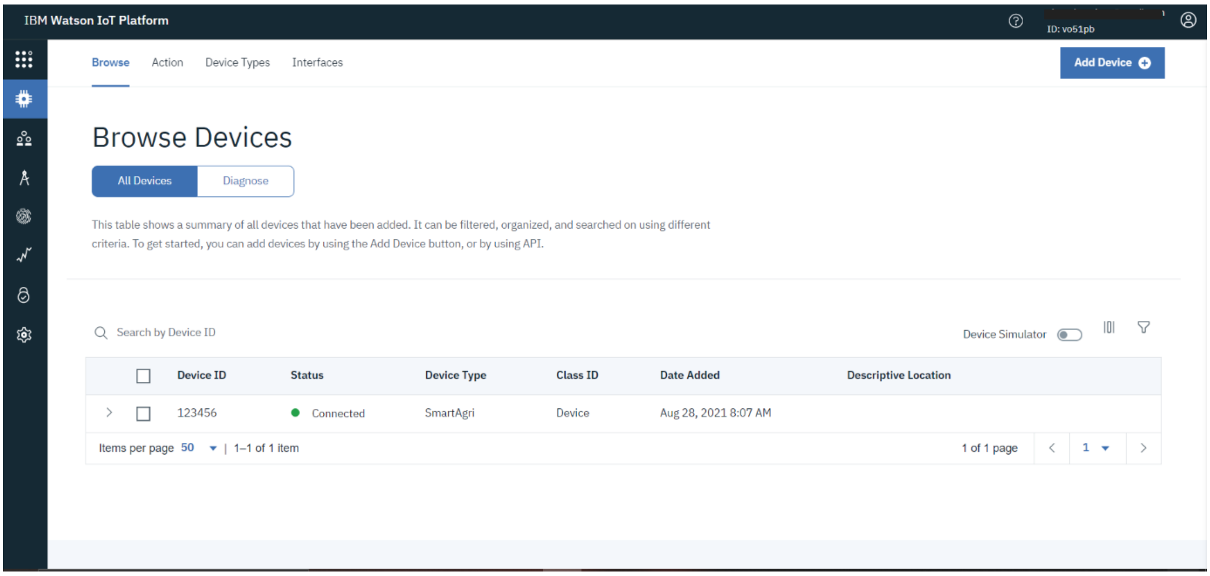


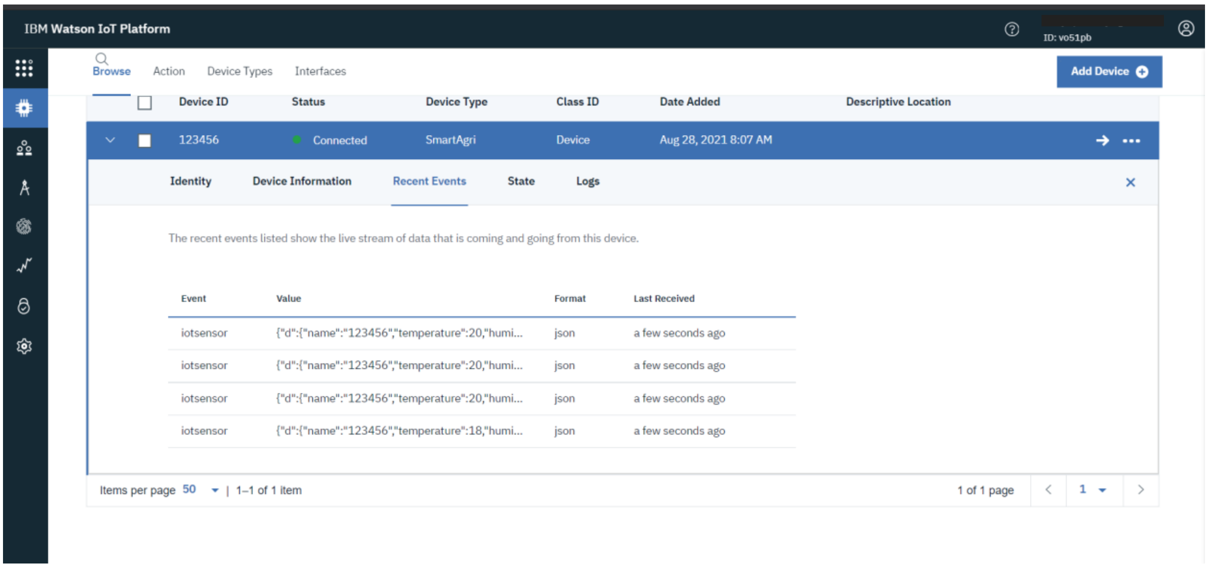

IBM Watson IoT Platform
[redacted]
ID: vo51pb
Browse
Action
Device Types
Interfaces
Add Device
Identity
Device Information
Security
Summary
You can modify the default device information and enter more information about the device for identification purposes.
Serial Number
Enter Serial Number
Manufacturer
Enter Manufacturer
Model
Enter Model
Device Class
Enter Device Class
Description
Enter Description
Firmware Version
Enter Firmware Version
Hardware Version
Enter Hardware Version
Descriptive Location
Enter Descriptive Location
Add Metadata
Back
Next


IBM Watson IoT Platform
[redacted]
ID: vo51pb
Browse
Action
Device Types
Interfaces
Add Device
Identity
Device Information
Security
Summary
There are two options for selecting a device authentication token.
Auto-generated authentication token (default)
Allow the service to generate an authentication token for you. Tokens are 18 characters and contain a mix of alphnumeric characters and symbols. The token is returned to you at the end of the device registration process.
Self-provided authentication token
Provide your own authentication token for this device. The token must be between 8 and 36 characters and contain a mix lowercase and uppercase letters, numbers, and symbols, which can include hyphens, underscores, and periods. Do not use repeated characters, dictionary words, user names, or other predefined sequences.
Authentication Token
123456789
Make a note of the generated token. Lost authentication tokens cannot be recovered. Tokens are encryped before being stored.
Authentication token are encrypted before we store them.

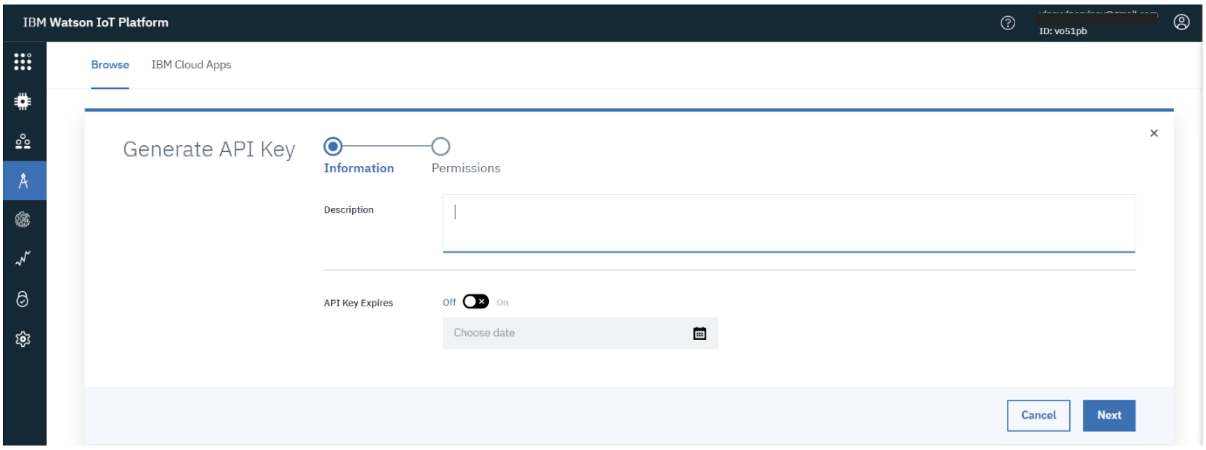

IBM Watson IoT Platform
[redacted]
ID: vo51pb
Browse
Action
Device Types
Interfaces
Add Device
Identity
Device Information
Security
Summary
Verify that the following information is correct then select Finish
Device Type
SmartAgri
Device ID
123456
View Metadata
Security Token
123456789
Back
Finish

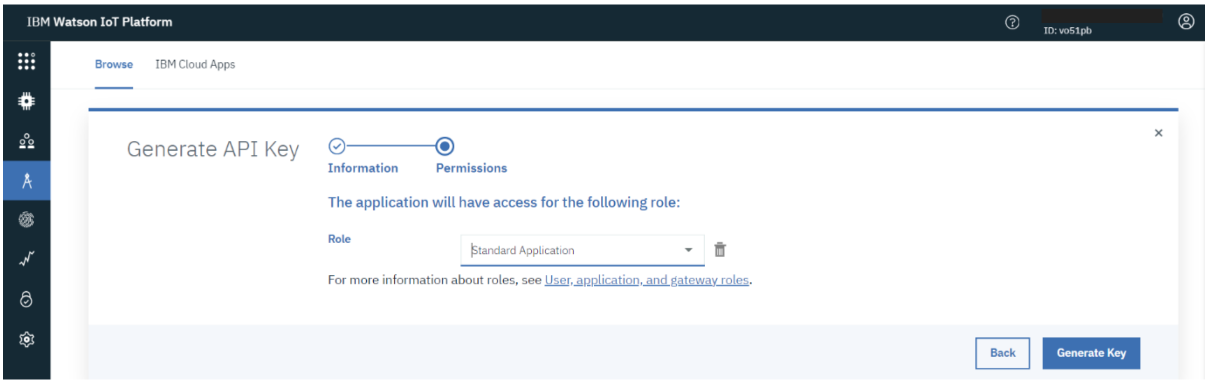

IBM Watson IoT Platform
[redacted]
ID: vo51pb
← Back
Device Drilldown - 123456
Device Credentials
Connection Information
Recent Events
State
Device Information
Metadata
Diagnostics
Connection Logs
Device Actions
Device Credentials
You registered your device to the organization. Add these credentials to the device to connect it to the platform. After the device is connected, you can navigate to view connection and event details.
Organization ID vo51pb
Device Type SmartAgri
Device ID 123456
Authentication Method use-token-auth
Authentication Token 123456789
Authentication tokens are non-recoverable. If you misplace this token, you will need to re-register the device to generate a new authentication token.
Find out how to add these credentials to your device

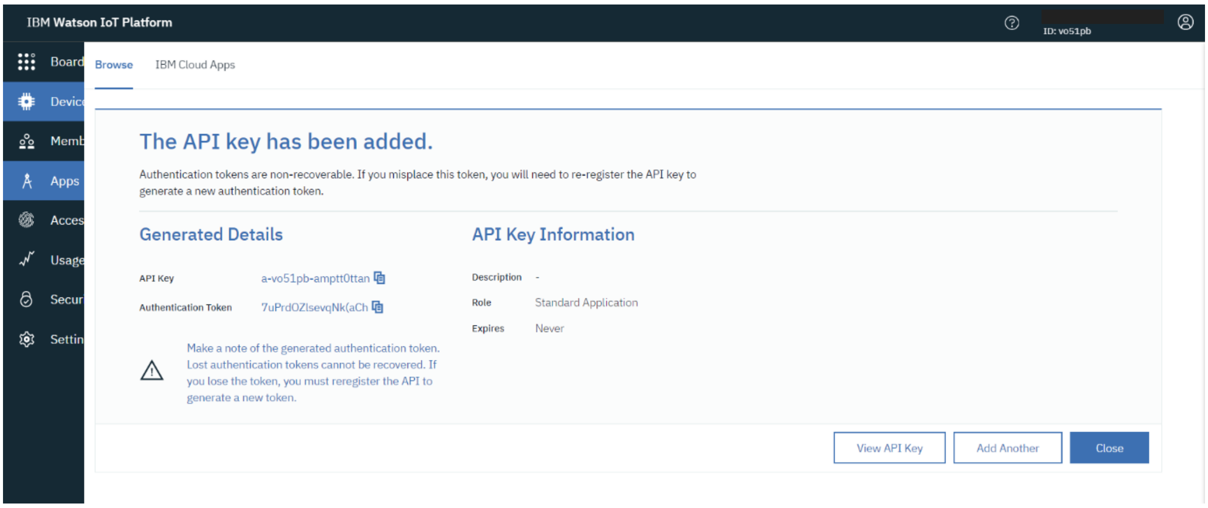

IBM Watson IoT Platform
[redacted]
ID: vo51pb
Browse
Action
Device Types
Interfaces
Add Device
Browse Devices
All Devices
Diagnose
This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.
Search by Device ID
Device Simulator
Device ID
Status
Device Type
Class ID
Date Added
Descriptive Location
123456
Connected
SmartAgri
Device
Aug 28, 2021 8:07 AM
Items per page 50 | 1–1 of 1 item
1 of 1 page
1


IBM Watson IoT Platform
[redacted]
ID: vo51pb
Browse
Action
Device Types
Interfaces
Add Device
Device ID
Status
Device Type
Class ID
Date Added
Descriptive Location
123456
Connected
SmartAgri
Device
Aug 28, 2021 8:07 AM
Identity
Device Information
Recent Events
State
Logs
The recent events listed show the live stream of data that is coming and going from this device.
Event
Value
Format
Last Received
iotsensor {"d":{"name":"123456","temperature":20,"humi... json a few seconds ago
iotsensor {"d":{"name":"123456","temperature":20,"humi... json a few seconds ago
iotsensor {"d":{"name":"123456","temperature":20,"humi... json a few seconds ago
iotsensor {"d":{"name":"123456","temperature":18,"humi... json a few seconds ago
Items per page 50 | 1–1 of 1 item
1 of 1 page
1

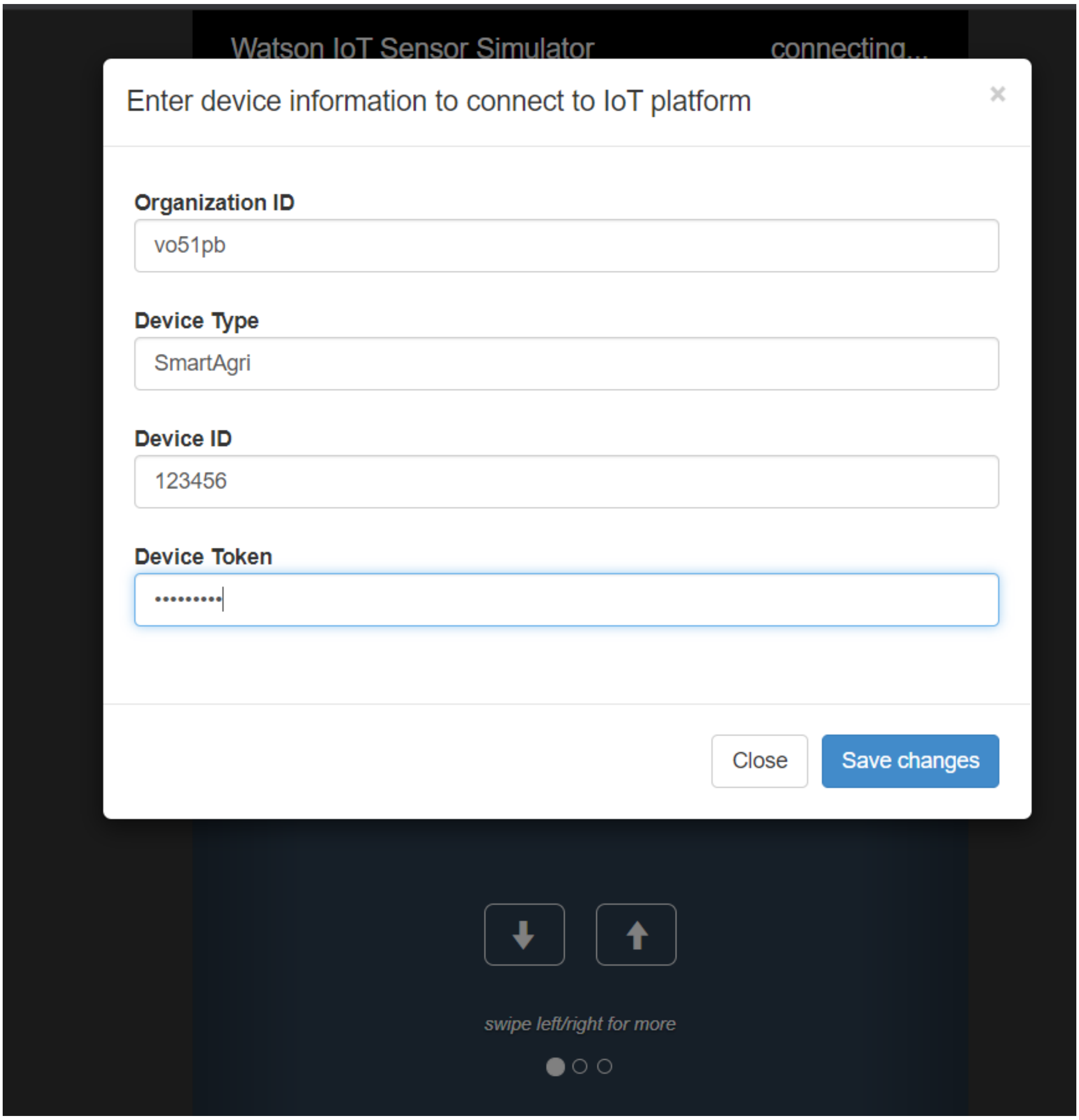
API KEY:


IBM Watson IoT Platform
[redacted]
ID: vo51pb
Browse
IBM Cloud Apps
Generate API Key
Information
Permissions
Description
API Key Expires
Off
On
Choose date
Cancel
Next

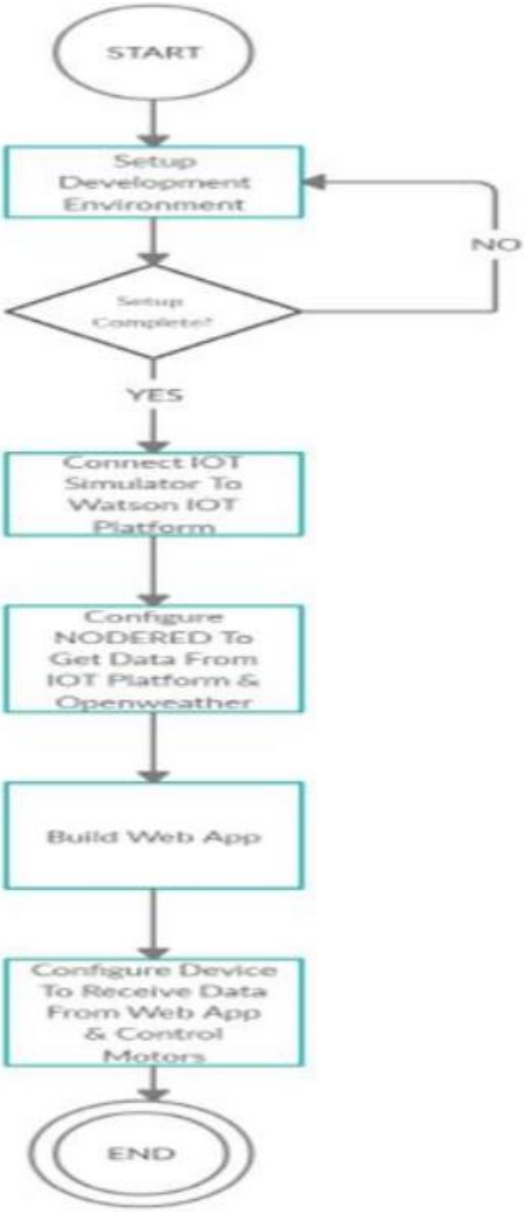

IBM Watson IoT Platform
[redacted]
ID: vo51pb
Browse
IBM Cloud Apps
Generate API Key
Information
Permissions
The application will have access for the following role:
Role
Standard Application
For more information about roles, see User, application, and gateway roles.
Back
Generate Key


IBM Watson IoT Platform
[redacted]
ID: vo51pb
Board
Devic
Memb
Apps
Acces
Usage
Secur
Settin
Browse
IBM Cloud Apps
The API key has been added.
Authentication tokens are non-recoverable. If you misplace this token, you will need to re-register the API key to generate a new authentication token.
Generated Details
API Key
a-vo51pb-ampttOttan
Authentication Token
7uPrdOZIsevqNk(aCh
Make a note of the generated authentication token. Lost authentication tokens cannot be recovered. If you lose the token, you must reregister the API to generate a new token.
API Key Information
Description
-
Role
Standard Application
Expires
Never
View API Key
Add Another
Close

SENSOR SIMULATOR:

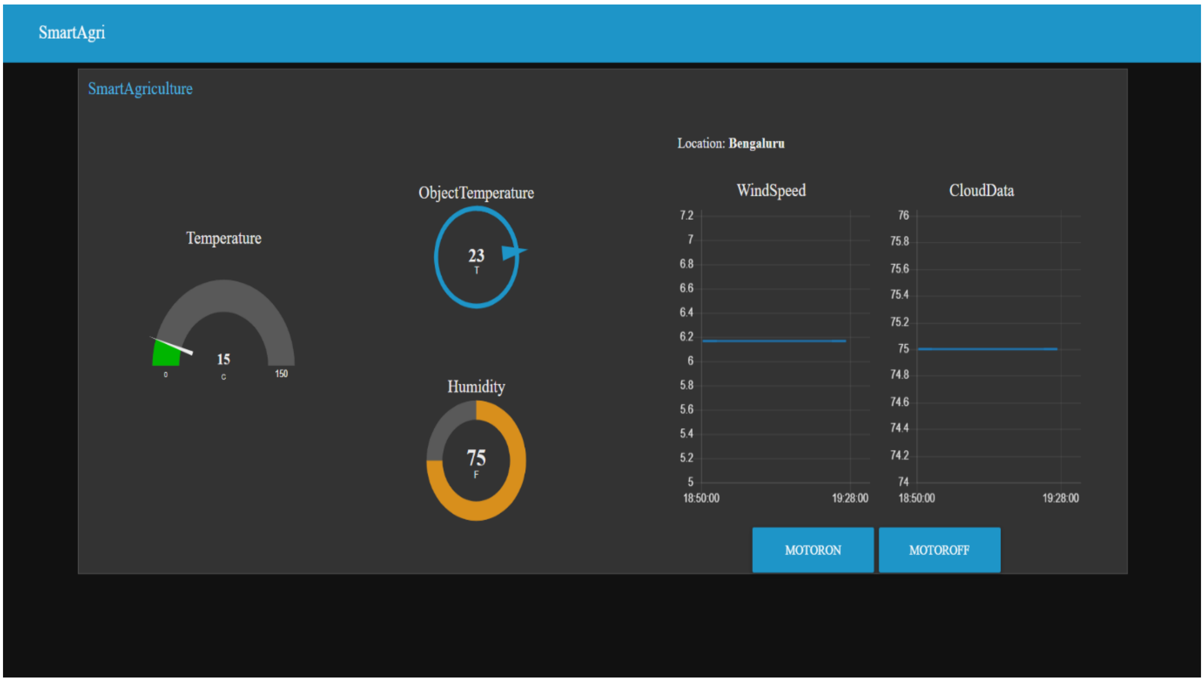


FLOWCHART:


START
Setup Development Environment
Setup Complete?
NO
YES
Connect IOT Simulator To Watson IOT Platform
Configure NODERED To Get Data From IOT Platform & Openweather
Build Web App
Configure Device To Receive Data From Web App & Control Motors
END

RESULT:

DASHBOARD: [Sensor Simulator and Motor Control]



ADVANTAGES & DISADVANTAGES:

7.1 ADVANTAGES:

- A remote-control system can help in working irrigation system valves dependent on schedule. Irrigating remote farm properties can be exceptionally troublesome and labour-intensive. It gets hard to comprehend when the valves were started and whether the ideal measure of water was distributed
- Various solutions are available to monitor engine statistics and starting or stopping the engine. When the client chooses to begin or stop the motor, the program transmits a sign to the unit within seconds by means of a mobile phone system

7.2 DISADVANTAGES:

- The Data observed by the client requires Internet continuous.
- Cost is higher for some components.

APPLICATIONS:

1. Crop Water Management
2. Precision Agriculture
3. Integraratted Pest Management or Control (IPM/C)
4. Food Productivity and Safety

CONCLUSION:

8.1 CONCLUSION:

IoT based SMART FARMING SYSTEM for Live Monitoring of Temperature and Soil Moisture has been proposed using Arduino and Cloud Computing. The System has high efficiency and accuracy in fetching the live data of temperature and soil moisture. The IoT based smart farming System being proposed via this report will assist farmers in increasing the agriculture yield and take efficient care of food production as the System will always provide helping hand to farmers for getting accurate live feed of environmental temperature and soil moisture with more than 99% accurate results.

8.2 FUTURE SCOPE:

Future work would be focused more on increasing sensors on this system to fetch more data especially with regard to Pest Control and by also integrating GPS module in this system to enhance this Agriculture IoT Technology to full-fledged Agriculture Precision ready product.

REFERENCE:

9.1 REFERENCE:

http://www.researchgate.net/

http://www.wikipedia.org/

http://www.youtube.com/

9.2 SOURCE CODE [PYTHON]:

```python
import time
import sys
import ibmiotf.application
import ibmiotf.device

# Provide your IBM Watson Device Credentials
organization = "vo51pb"  # repalce it with organization ID
deviceType = "SmartAgri"  # replace it with device type
deviceId = "123456"  # repalce with device id
authMethod = "token"
authToken = "123456789"  # repalce with token


def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data)
    if cmd.data['command'] == 'motoron':
        print("MOTOR ON")
    elif cmd.data['command'] == 'motoroff':
        print("MOTOR OFF")


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()

deviceCli.connect()

while True:
    T = 50;
    H = 32;
    # Send Temperature & Humidity to IBM Watson
    data = {'Temperature': T, 'Humidity': H}


    # print data
    def myOnPublishCallback():
        print("Published Temperature = %s C" % T, "Humidity = %s %%" % H,
"to IBM Watson")
```

```
    success = deviceCli.publishEvent("event", "json", data, qos=0,
on_publish=myOnPublishCallback)
    if not success:
        print("Not connected to IoTF")
    time.sleep(1)

    deviceCli.commandCallback = myCommandCallback

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