SMART AGRICULTURE SYSTEM BASED ON IOT

PROJECT REPORT

INTRODUCTION:

Smart Agriculture is a way through which farmer can have effortless farming techniques. In this project, the farmer can operate the motor without visiting the farm-land. This project is built using IBM cloud platform, Node-red and Watson IOT sensor. Using this project farmers can ON/OFF the motor without visiting the farm-land based on the temperature and moisture content.

PROJECT REQUIREMENTS:

* IBM cloud Platform
* Python IDLE
* Watson IOT Simulator
* Node-Red service

PROCEDURE:

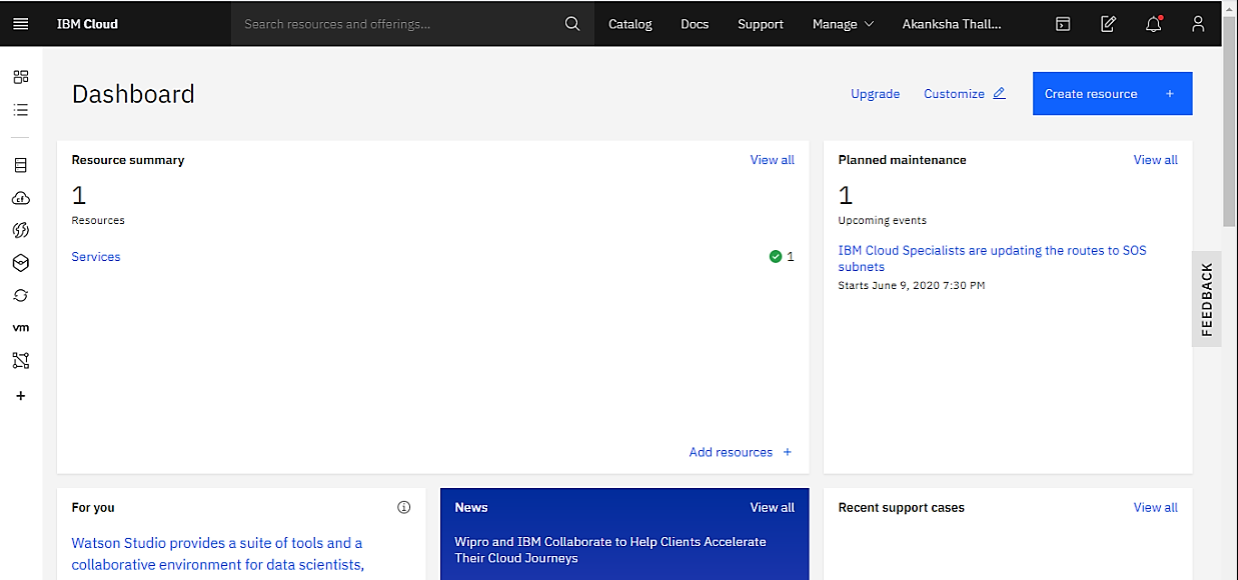
Create an account in IBM cloud platform. Search for IOT platform (Internet Of Things platform) in catalog and create a resource in it then, a platform is created.

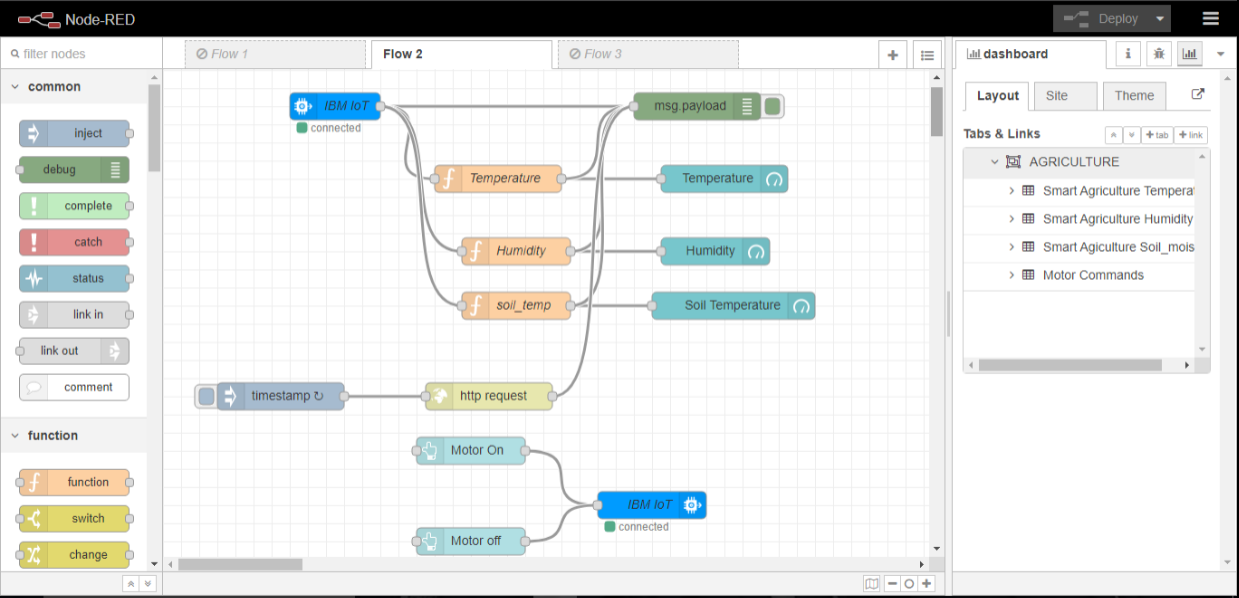
Create device and note all the device credentials. Now open the Watson IOT platform (IOT sensor) and apply all the device credentials in it. The device will be connected with all the parameters like temperature, object temperature and humidity. The device and the sensor platform are connected.

Install the Node-Red platform locally in your system. Run the command node-red and open the link in new tab, it will be redirected to the node-red flow page. Create the flow with the device created on IBM IOT platform and all the required options like motor ON and motor OFF. Weather API is open source platform where the weather details of a particular region are noted. This Weather API is integrated in the node-red platform. A web application is created with the interface of IBM IOT platform and Node-red. The web application is created with all the data parameters and motor buttons.

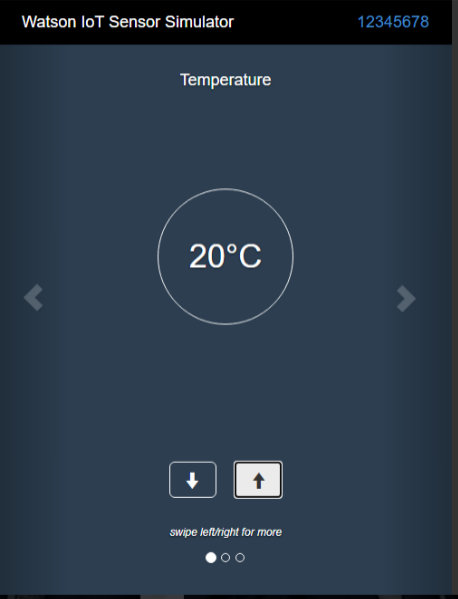
Install the python IDLE in your system and write the code with the noted device credentials, motor commands and run the code. The commands given in the web application are viewed in python idle platform (MOTOR ON/MOTOR OFF).

IBM CLOUD ACCOUNT:

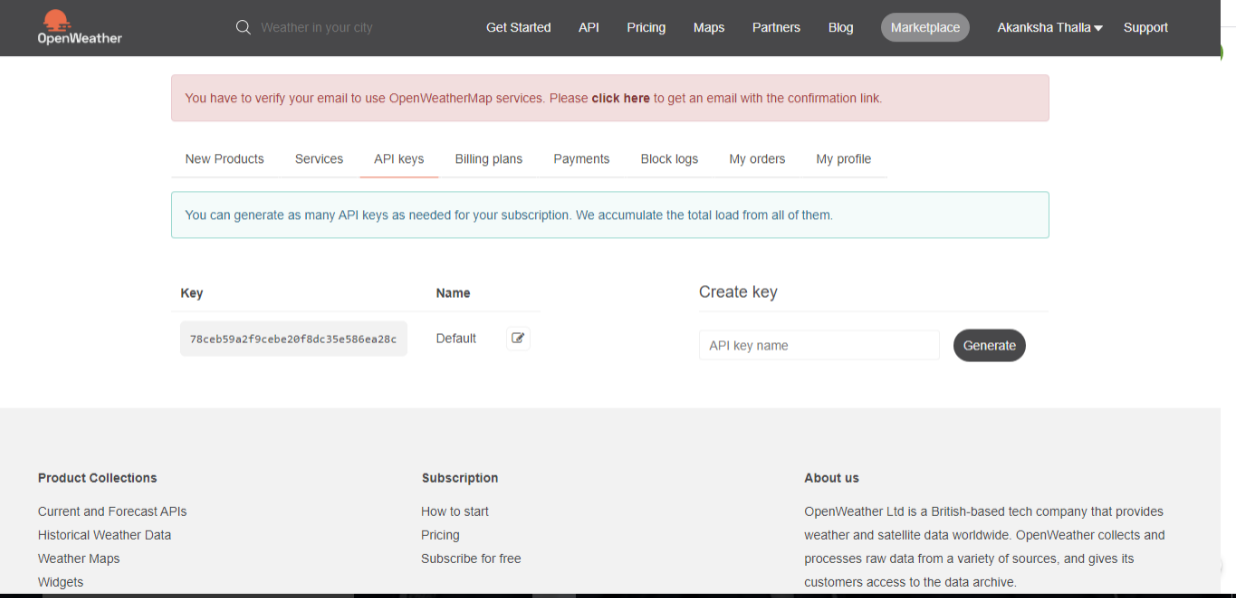


NODE-RED FLOW: 

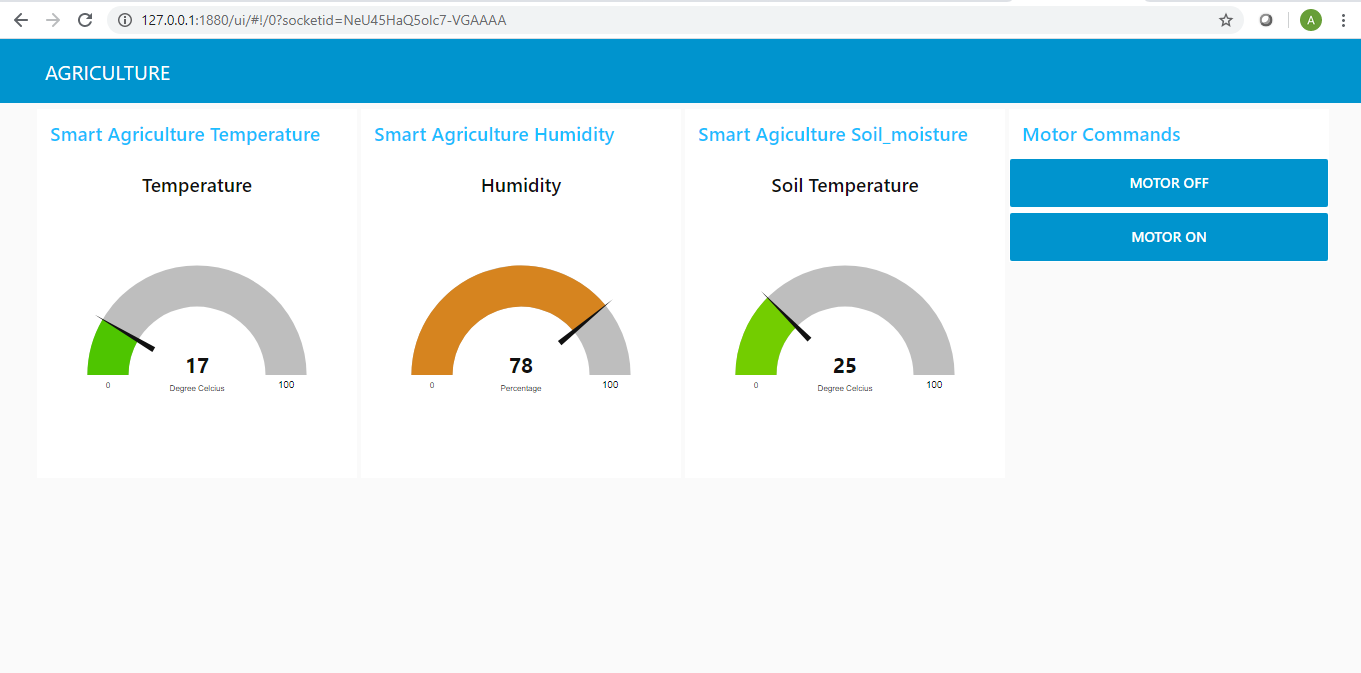
IBM IOT SIMULATOR:



WEATHER API:



USER INTERFACE:



PYTHON CODE:

import time

import sys

import ibmiotf.application # to install pip install ibmiotf

import ibmiotf.device

#Provide your IBM Watson Device Credentials

organization = "nt897s" #replace the ORG ID

deviceType = "nodemcu1"#replace the Device type wi

deviceId = "123456789"#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()

RESULT:

The Smart agriculture system helps the farmers to reduce their efforts and time of visiting the farm frequently. The motor at the farm can be operated without visiting the farm using this project.

RESULT:

