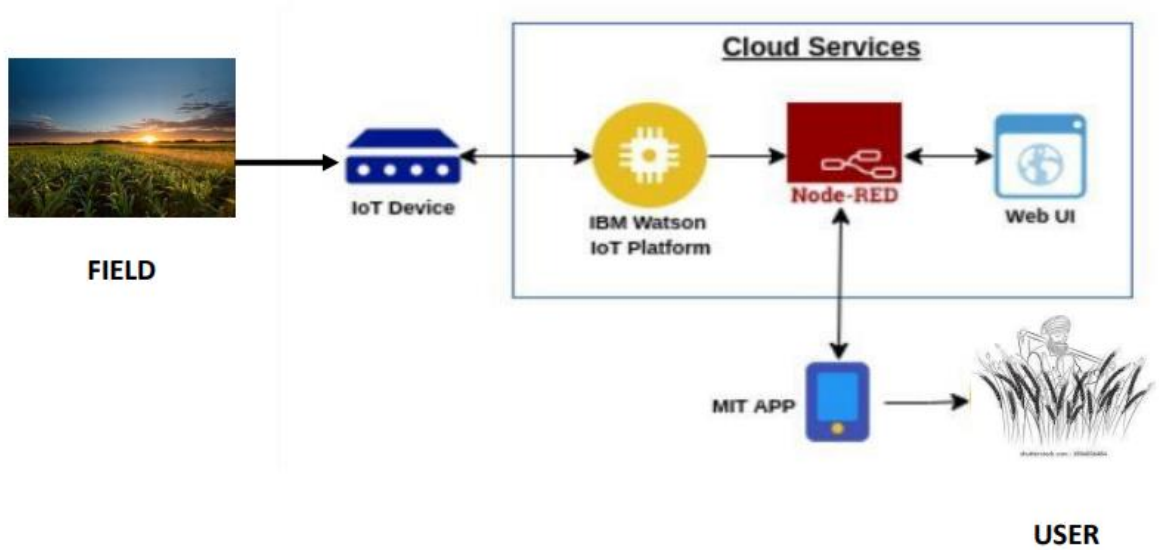
DELIVERY OF SPRINT PHASE 4

Date	18 November 2022
Team ID	PNT2022TMID11119
Project Name	Smart Farmer - IoT Enabled Smart Farming Application

Block Diagram:



Program:

```
#IBM Watson IOT Platform #pip install wiotp-sdk import wiotp.sdk.device

import time

import random

 import requests, json

ms=0

api_key = "a0db30a689a774b93ffcb58ef2eddfda"



base_url = "http://api.openweathermap.org/data/2.5/weather?"

city_name = 'Chennai, IN'

complete_url = base_url + "appid=" + api_key + "&q=" + city_name
```

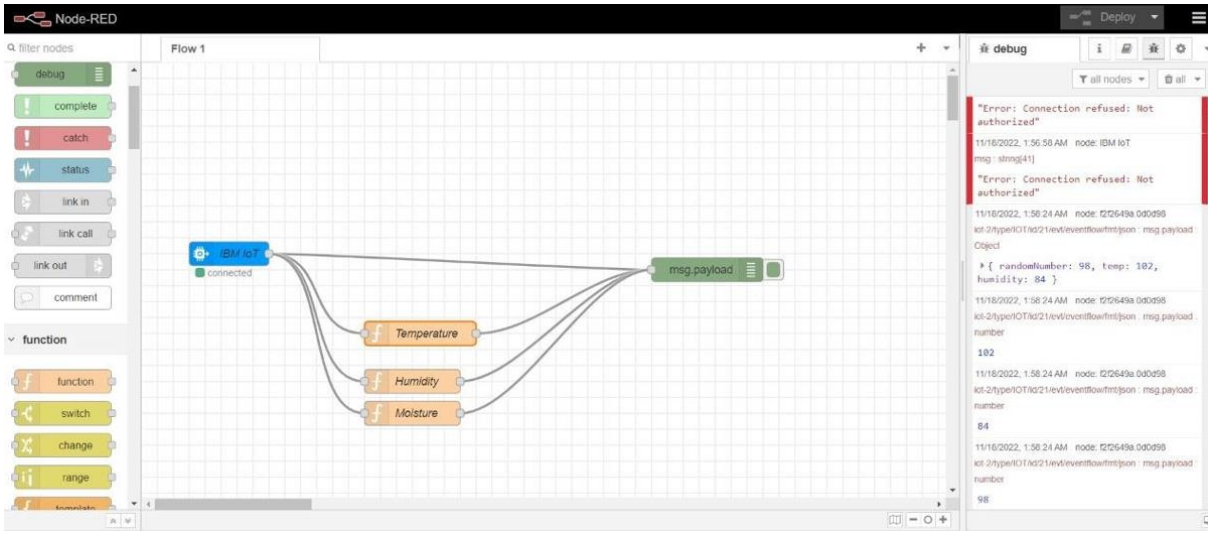
```
status='motor off'
myConfig = {
    "identity": {
       "orgId": "17lsro",
"typeId": "MyDeviceType",
"deviceId":"12345"
    },
    "auth": {
       "token": "GkatKdiUS?UVHKvnAD"
    }
 }
 def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" %
 cmd.data['command'])
    m=cmd.data['command']
    if(m=="MOTOR ON"):
    print("MOTOR IS ON") global
    status status='motor on'
    myData={'temperature':temp,
 'humidity':hum,'soilmoisture':sm_percentage,'status':status,'api_temperature'
:
 api_temperature,'api_pressure':api_pressure,'api_humidity':api_humidity,'api
 _weather_description':api_weather_description}
       client.publishEvent(eventId="status", msgFormat="json", data=myData,
 qos=0, onPublish=None) print("Published data
       Successfully: %s", myData)
       time.sleep(2)
```

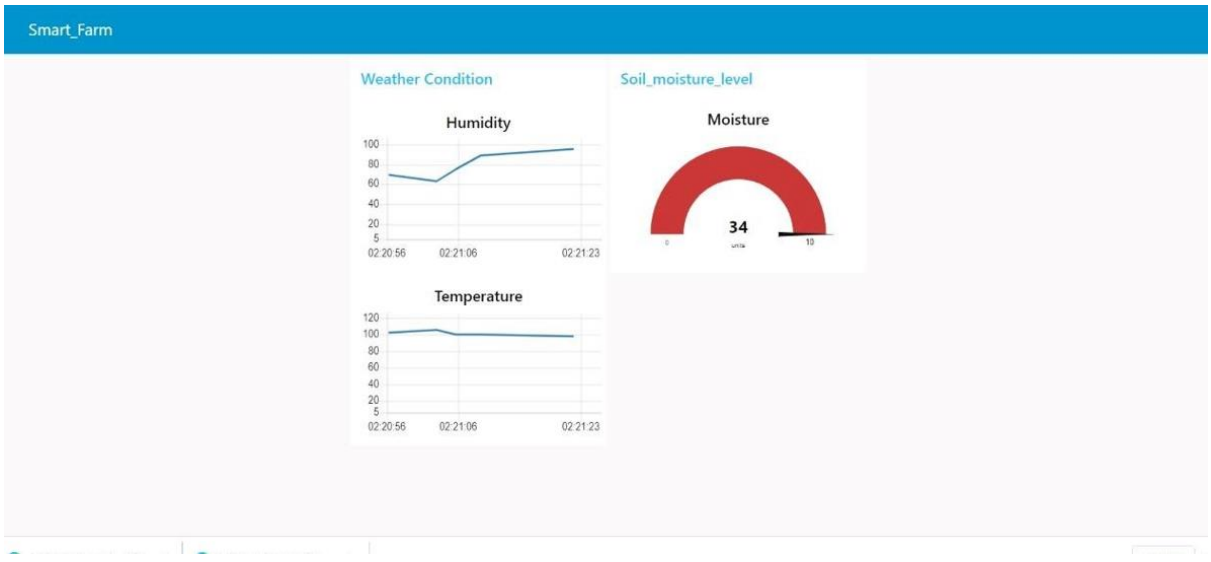
```
    elif(m=="MOTOR OFF"): print("MOTOR
        IS OFF") status='motor off'
        myData={'temperature':temp,
'humidity':hum,'soilmoisture':sm_percentage,'status':status,'api_temperature'
:
api_temperature,'api_pressure':api_pressure,'api_humidity':api_humidity,'api
_weather_description':api_weather_description}
        client.publishEvent(eventId="status", msgFormat="json", data=myData,
qos=0, onPublish=None) print("Published data
        Successfully: %s", myData)



        time.sleep(2)
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()
while True:
    response =
    requests.get(complete_url) x =
    response.json() if x["cod"] != "404":
        y = x["main"]
        api_temperature = y["temp"]
        api_pressure = y["pressure"]
        api_humidity = y["humidity"]
        z = x["weather"]
        api_weather_description = z[0]["description"]
    temp=random.randint(-20,125)
    hum=random.randint(0,100)
```

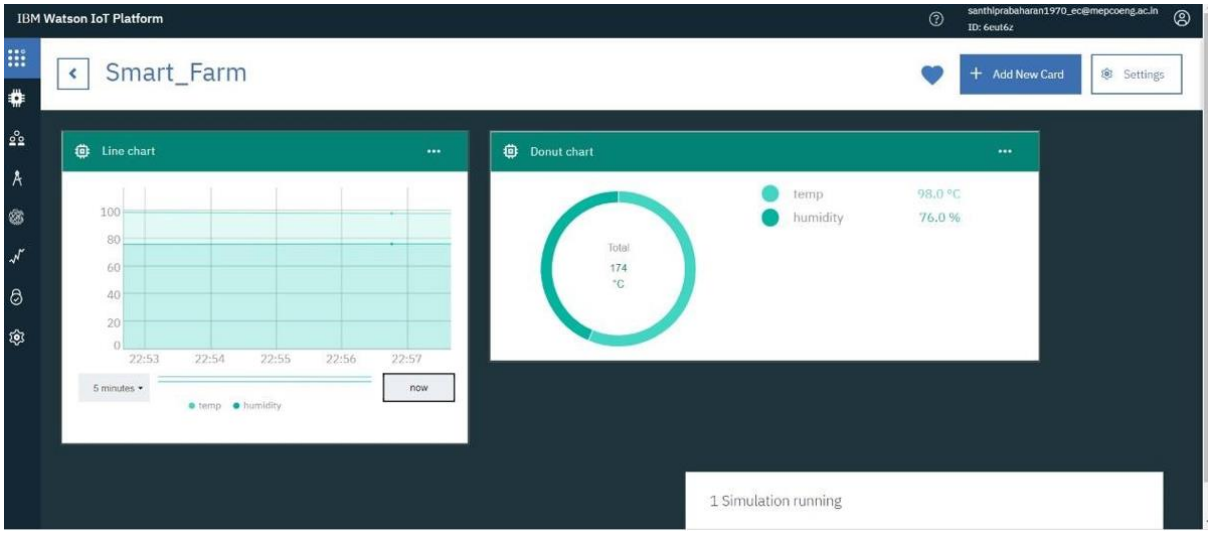
```
    soilmoisture=random.randint(0,1023)#analog sensor
    sm_percentage=(soilmoisture/1023)*100
    sm_percentage=int(sm_percentage) myData={'temperature':temp,
'humidity':hum,'soilmoisture':sm_percentage,'status':status,'api_temperature'
:
api_temperature,'api_pressure':api_pressure,'api_humidity':api_humidity,'api
_weather_description':api_weather_description}
    client.publishEvent(eventId="status", msgFormat="json", data=myData,
qos=0, onPublish=None)
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
time.sleep(2) client.disconnect()
```

OUTPUT:

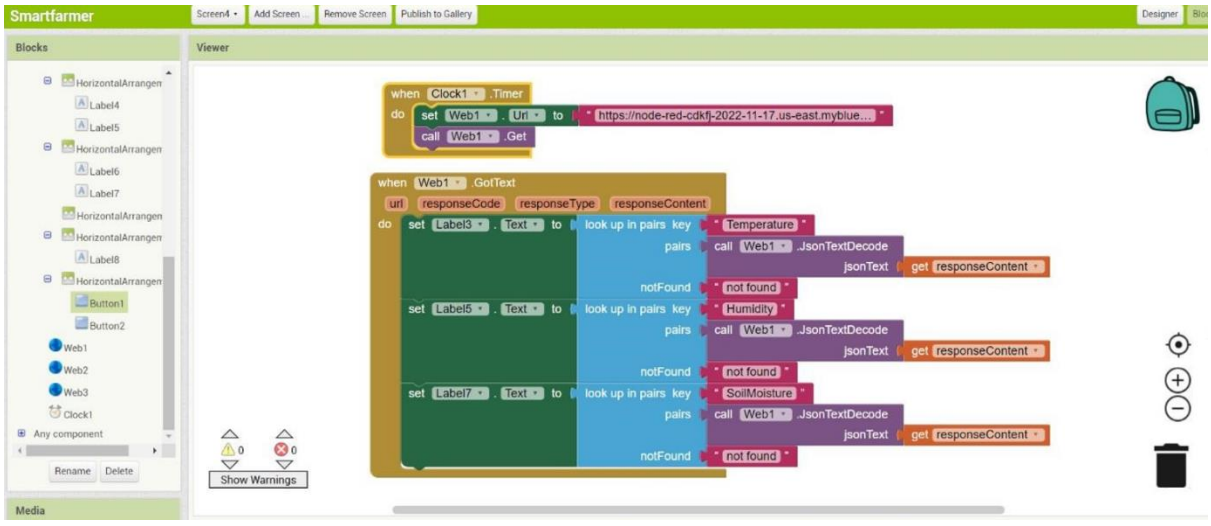
Node-RED Flow Connection:

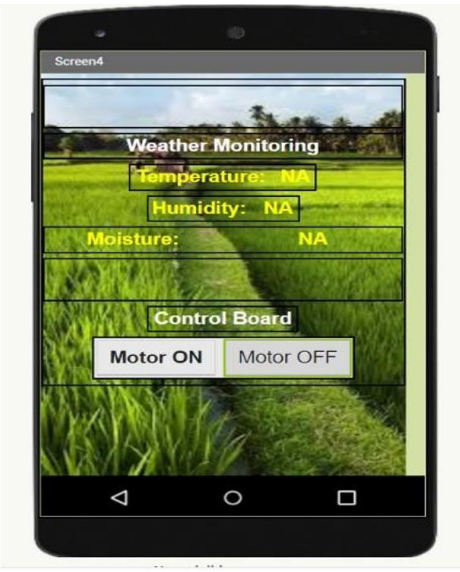






MIT App Inventor:




Screen4
Weather Monitoring
Temperature: NA
Humidity: NA
Moisture: NA
Control Board
Motor ON
Motor OFF