Project Development Phase

Delivery of Sprint -1

Team ID	PNT2022TMID48383
Project Name	Smart Farmer-IOT Enabled Smart Farming Application

In Sprint-1 we are going to develop the python code .

1. Introduction

The main aim of this project is to help farmers automate their farms by providing them with a Web App through which they can monitor the parameters of the field like Temperature, soil moisture, humidity and etc .And control the equipment like water motor and other devices remotely via internet without their actual presence in the field.

2. Problem Statement

Farmers are to be present at farm for its maintenance irrespective of the weather conditions. They have to ensure that the crops are well watered and the farm status is monitored by them physically. Farmer have to stay most of the time in field in order to get a good yield. In difficult times like in the presence of pandemic also they have to work hard in their fields risking their lives to provide food for the country.

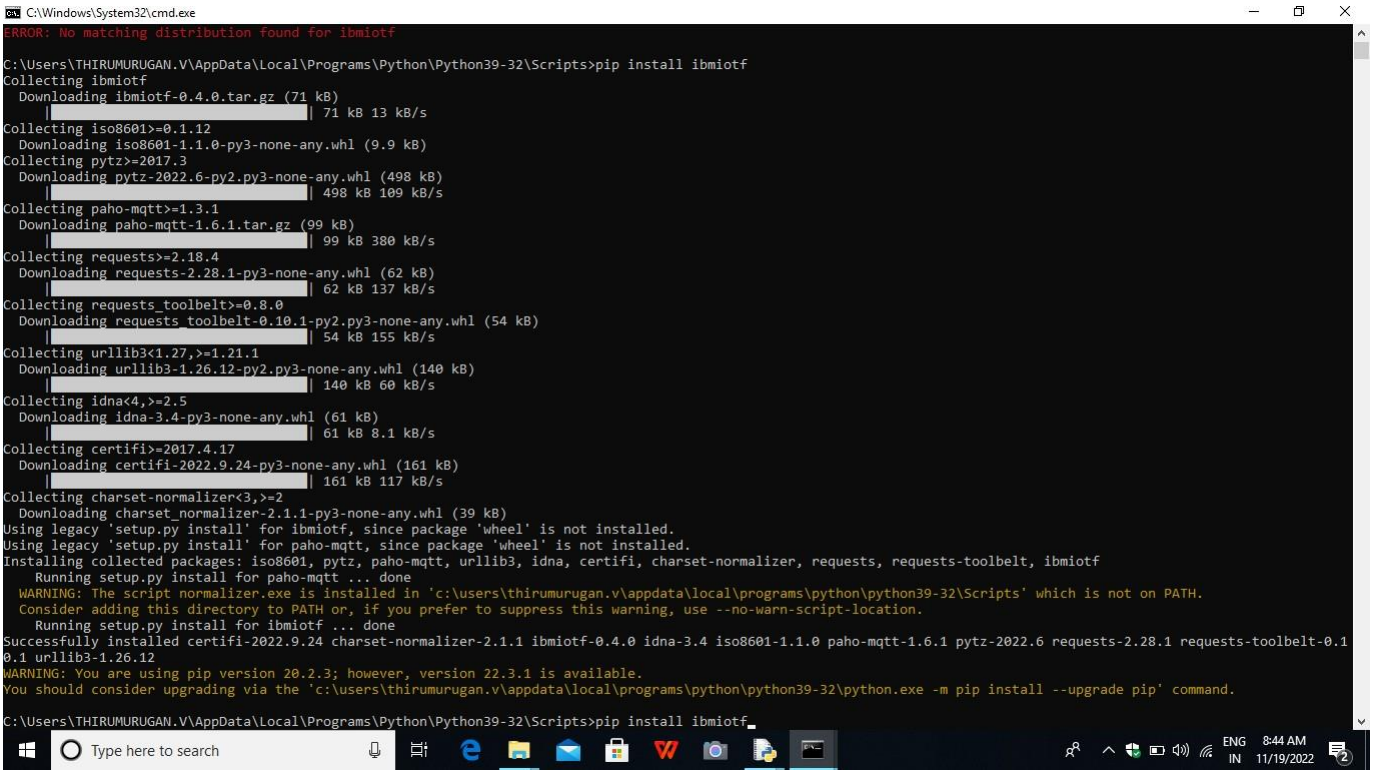
3. Proposed Solution

In order to improve the farmer's working conditions and make them easier, we introduce IoT services to him in which we use cloud services and internet to enable farmer to continue his work remotely via internet. He can monitor the field parameters and control the devices in farm.

4 . Software Requirements

1. Python IDLE 3.7.0 (64-Bit)

Installing pip ibmiotf packages in python idle.After that we have to write the python code.



python code..py - C:\Users\THIRUMURUGAN.V\Desktop\python code..py (3.9.1)

File Edit Format Run Options Window Help

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


#Provide your IBM Watson Device Credentials
organization = "49x4b9"
deviceType = "weather_monitoring"
deviceId = "weather_today"
authMethod = "token"
authToken = "Qp4oHg?bZHhaQeigMA"


def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    print(cmd)


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:
        temperature=random.randint(0,100)
        humidity=random.randint(0,100)
        soil= random.randint(0,100)

        data = {'temperature' : temperature, 'humidity': humidity ,'soil':soil}
        #print data
        def myOnPublishCallback():
            print ("Published Temperature = %s C" % temperature, "Humidity = %s %%" % humidity, "soil Moisture = %s %%"% soil,"to IBM Watson")
```

Ln: 1 Col: 0

Type here to search

ENG IN 8:37 AM 11/19/2022

python code..py - C:\Users\THIRUMURUGAN.V\Desktop\python code..py (3.9.1)

File Edit Format Run Options Window Help

```
authMethod = "token"
authToken = "Qp4oHg?bZHhaQeigMA"


def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    print(cmd)


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:
        temperature=random.randint(0,100)
        humidity=random.randint(0,100)
        soil= random.randint(0,100)

        data = {'temperature' : temperature, 'humidity': humidity ,'soil':soil}
        #print data
        def myOnPublishCallback():
            print ("Published Temperature = %s C" % temperature, "Humidity = %s %%" % humidity, "soil Moisture = %s %%"% soil,"to IBM Watson")

        success = deviceCli.publishEvent("IoTSensor", "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()
```

Ln: 1 Col: 0

Type here to search

ENG IN 8:37 AM 11/19/2022

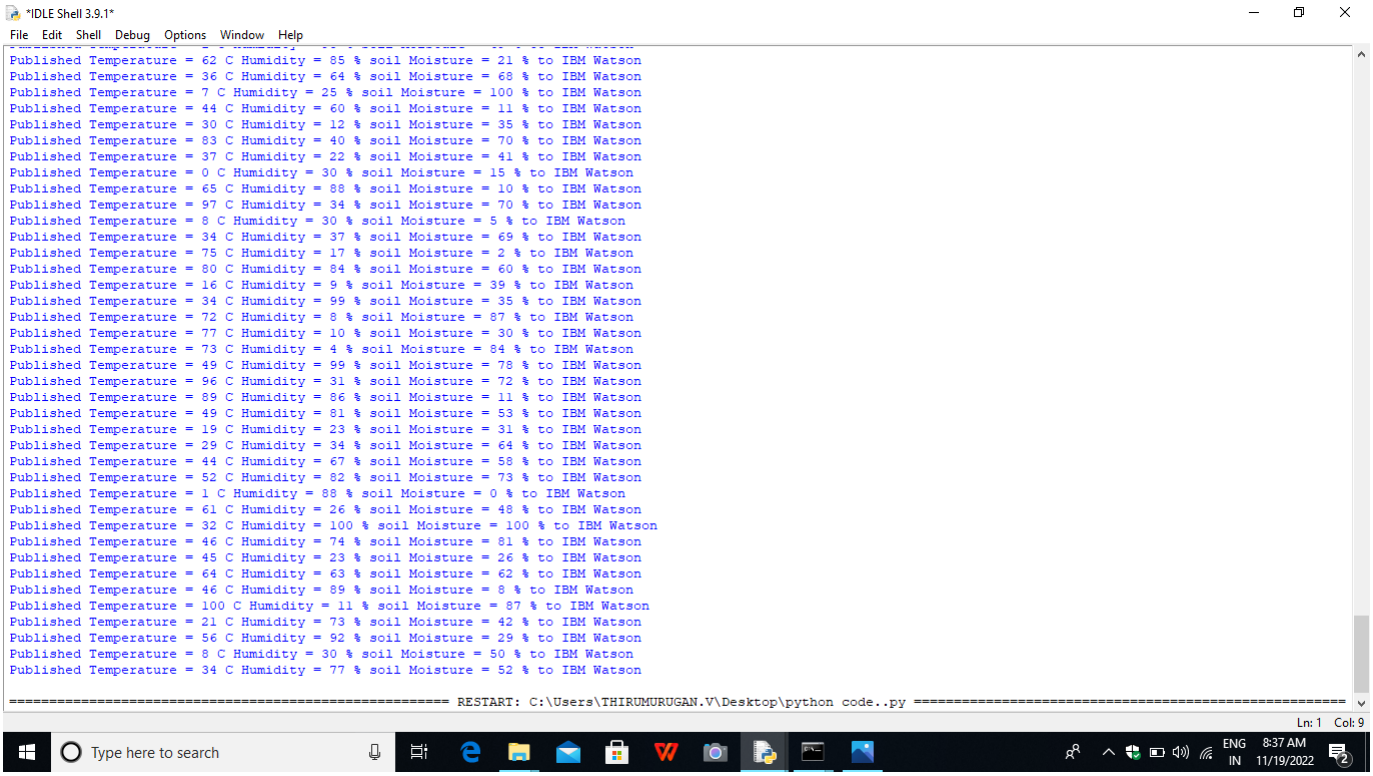
CODE:

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


#Provide your IBM Watson Device Credentials
organization = "49x4b9"
deviceType = "weather_monitoring"
deviceId = "weather_today"
authMethod = "token"
authToken = "Qp4oHg?bZHhaQeigMA"
```

```
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    print(cmd)



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:

        temperature=random.randint(0,100)
        humidity=random.randint(0,100)
        soil= random.randint(0,100)

        data = {'temperature' : temperature, 'humidity':
humidity ,'soil':soil}
        #print data
        def myOnPublishCallback():
            print ("Published Temperature = %s C" %
temperature, "Humidity = %s %%" % humidity, "soil Moisture =
%s %%"% soil,"to IBM Watson")

        success = deviceCli.publishEvent("IoTSensor",
"json", data, qos=0, on_publish=myOnPublishCallback)
```

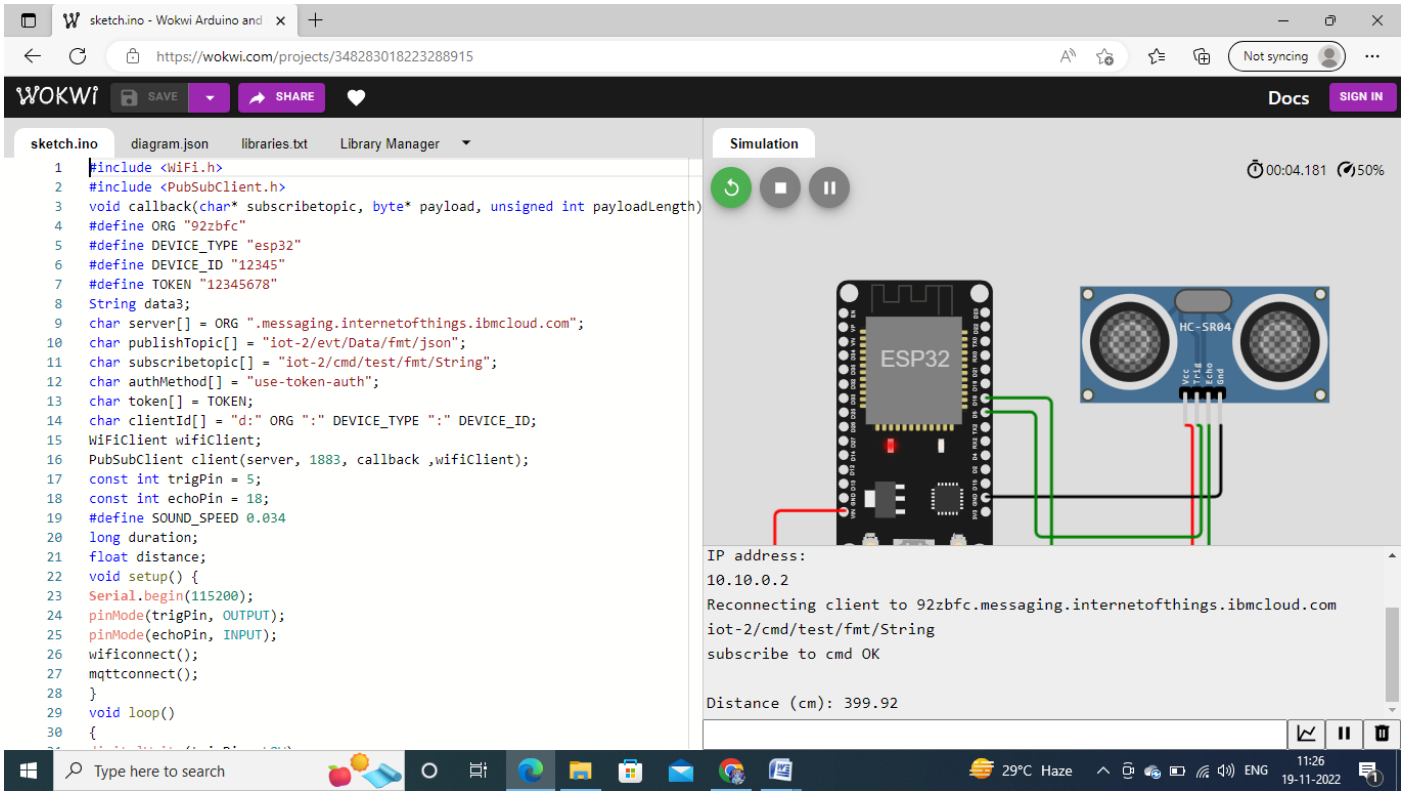
```
        if not success:
            print("Not connected to IoTF")
        time.sleep(1)

        deviceCli.commandCallback = myCommandCallback

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

Simulation output in the python idle:



Wokwi Link: https://wokwi.com/projects/348283018223288915

OUTPUT & SIMULATION:



Alert to IBM cloud: