Smart Farmer-IOT Enabled Smart Farming Application

DEVELOP A PYTHON CODE

Team ID	**PNT2022TMID11108**
Project Name	**Smart Farmer - IoT Enabled Smart FarmingApplication**
Maximum Marks	**4 Marks**

NAME	REGISTER NUMBER
SWEATHA.B	811519106162
SASI.K	811519106129
SUBHAA SHAKTHI.R	811519106150
SUSHMITHA.R.K	811519106158

The code of publishing messages

```
# python 3.6
import random


import time


from paho.mqtt import client as mqtt_client
 broker =


 'broker.emqx.io' port


 =1883 topic =


 "python/mqtt"
```

```
# generate client ID with pub prefix randomly

client_id = f'python-mqtt-{random.randint(0,

1000)}'

# username = 'emqx'
#password = 'public'
def connect_mqtt():

    def on_connect(client, userdata, flags, rc):
if rc == 0:

print("Connected to MQTT Broker!")

else:

print("Failed to connect, return code %d\n",rc)
```

```
client=mqtt_client.Client(client_id)

client.username_pw_set(username, password)

client.on_connect = on_connect


lient.connect(broker, port)

return client

def publish(client):

msg_count= 0
```

```
    whileTrue:

    time.sleep(1)

     msg =f"messages:

{msg_count}"

        result = client.publish(topic, msg)

        # result: [0, 1]

        status = result[0]
```

```
if status == 0:
            print(f"Send `{msg}` to topic `{topic}`")
        else:
            print(f"Failed to send message to topic
 {topic}")
 msg_count += 1
def run():
client = connect_mqtt()
client.loop_start()
publish(client)
if___name___== '__main__':
run()
```

```
def run():

    client = connect_mqtt()

    client.loop_start()

    publish(client)

if___name___== '__main__':

    run()
```

The code of subscribing

```
# python3.6

import random
from paho.mqtt import client as mqtt_client

broker = 'broker.emqx.io'
port = 1883 topic =
"python/mqtt"

# generate client ID with pub prefix randomly

client_id = f'python-mqtt-{random.randint(0, 100)}'

 # username = 'emqx'


 # password = 'public'

def connect_mqtt() -> mqtt_client:


def on_connect(client,userdata,flags,rc):


 if rc == 0:

print("Connected to MQTT Broker!")
```

```
else:
   print("Failed to connect, return code %d\n",rc)
 client = mqtt_client.Client(client_id)
 client.username_pw_set(username, password)
client.on_connect = on_connect
client.connect(broker, port)
 return client
 def subscribe(client: mqtt_client):
 def on_message(client, userdata, msg):
```

```
print(f"Received `{msg.payload.decode()}`from
`{msg.topic}` topic")

client.subscribe(topic)


client.on_message = on_message


def run():

client = connect_mqtt()

subscribe(client)

if__name___== '__main__':

run()
```