**Smart Farmer-IOT Enabled Smart Farming Application**

DEVELOP A PYTHON CODE

|  |  |
| --- | --- |
|  |  |
|  |  |
| **NAME** | **ROLL NO** |
| S.SANTHOSH KUMAR | 19BEC041 |
| R.SETHURAM | 19BEC081 |
| S.NESAMANI | 20BEC309 |
| R.HARI KRISHNA | 20BEC323 |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Team ID** | **PNT2022TMID08745** |
| **Project Name** | **Smart Farmer - IoT Enabled Smart Farming Application** |
| **Maximum Marks** | **4 Marks** |

**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

while True:

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()