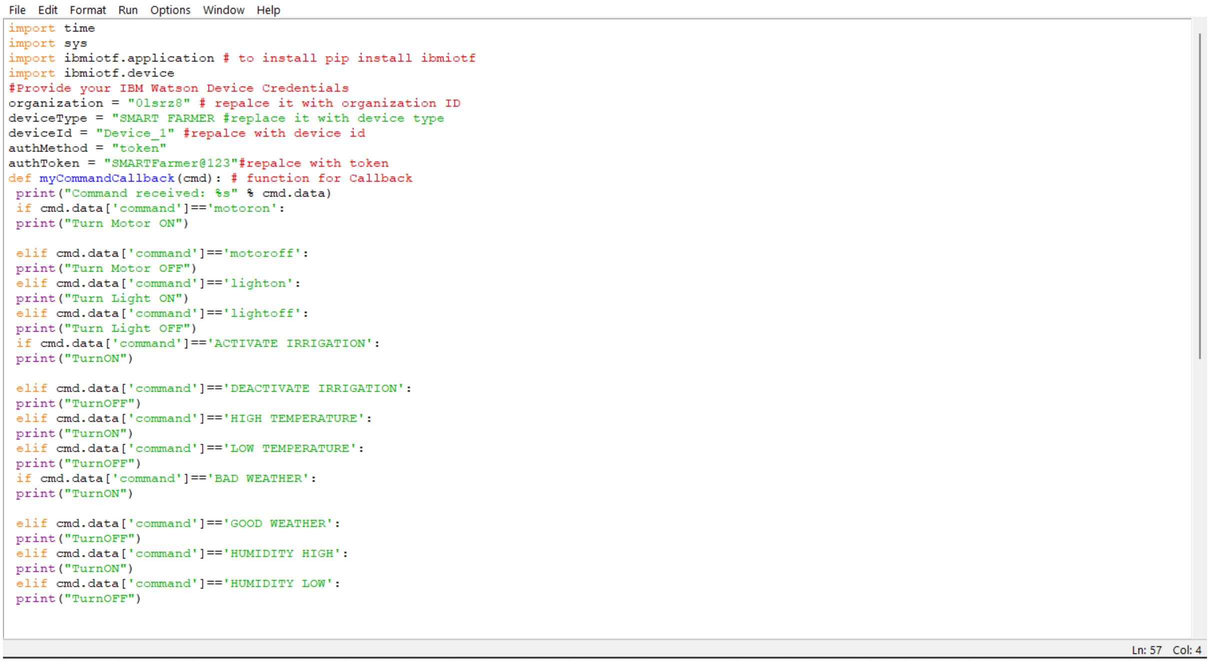
TEAM ID	PNT2022TMID12914
Project Name	Smart Farmer - IOT Enabled Smart Farming Application

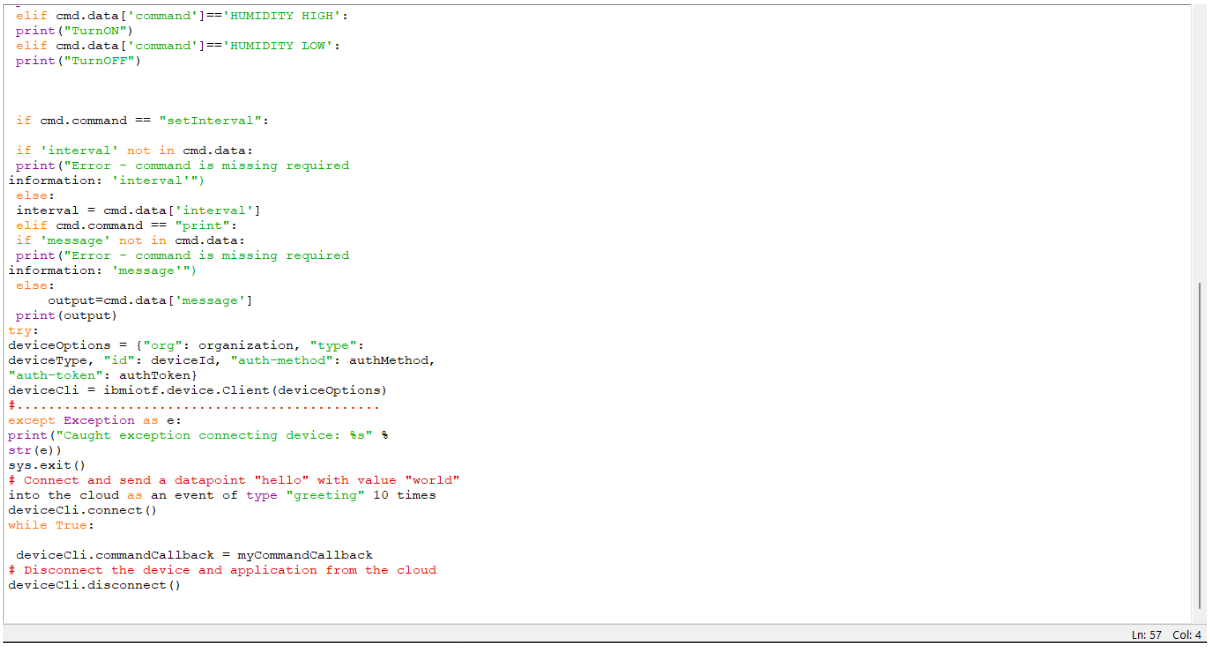
PYTHON CODE :

```
import time
import sys
import ibmiotf.application # to install pip install ibmiotf
import ibmiotf.device
#Provide your IBM Watson Device Credentials
organization = “jztdcw”
deviceType = "NodeMCU"
deviceId = " node-mcu-4321 "
authMethod = "use-token-auth"
authToken = "987654321"
 def myCommandCallback(cmd): # function for Callback
 print("Command received: %s" % cmd.data)
 if cmd.data['command']=='motoron':
 print("Turn Motor ON")

 elif cmd.data['command']=='motoroff':
 print("Turn Motor OFF")
 elif cmd.data['command']=='lighton':
 print("Turn Light ON")
 elif cmd.data['command']=='lightoff':
 print("Turn Light OFF")
 if cmd.data['command']=='ACTIVATE IRRIGATION':
 print("TurnON")

 elif cmd.data['command']=='DEACTIVATE IRRIGATION':
 print("TurnOFF")
 elif cmd.data['command']=='HIGH TEMPERATURE':
 print("TurnON")
 elif cmd.data['command']=='LOW TEMPERATURE':
 print("TurnOFF")
 if cmd.data['command']=='BAD WEATHER':
 print("TurnON")

 elif cmd.data['command']=='GOOD WEATHER':
 print("TurnOFF")
 elif cmd.data['command']=='HUMIDITY HIGH':
 print("TurnON")
 elif cmd.data['command']=='HUMIDITY LOW':
 print("TurnOFF")



 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()
```



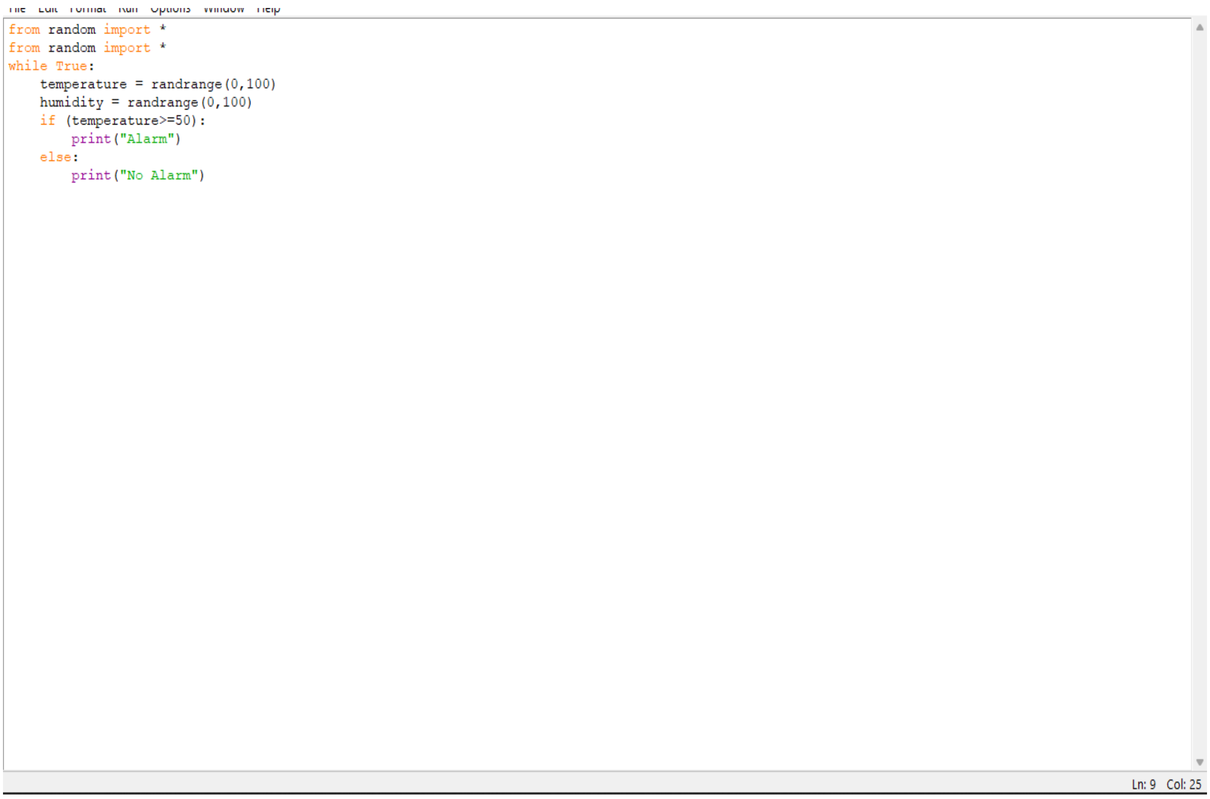
```
import time
import sys
import ibmiotf.application # to install pip install ibmiotf
import ibmiotf.device
#Provide your IBM Watson Device Credentials
organization = "01srz8" # repalce it with organization ID
deviceType = "SMART FARMER #replace it with device type
deviceId = "Device_1" #repalce with device id
authMethod = "token"
authToken = "SMARTFarmer@123"#repalce with token
def myCommandCallback(cmd): # function for Callback
 print("Command received: %s" % cmd.data)
 if cmd.data['command']=='motoron':
 print("Turn Motor ON")

 elif cmd.data['command']=='motoroff':
 print("Turn Motor OFF")
 elif cmd.data['command']=='lighton':
 print("Turn Light ON")
 elif cmd.data['command']=='lightoff':
 print("Turn Light OFF")
 if cmd.data['command']=='ACTIVATE IRRIGATION':
 print("TurnON")

 elif cmd.data['command']=='DEACTIVATE IRRIGATION':
 print("TurnOFF")
 elif cmd.data['command']=='HIGH TEMPERATURE':
 print("TurnON")
 elif cmd.data['command']=='LOW TEMPERATURE':
 print("TurnOFF")
 if cmd.data['command']=='BAD WEATHER':
 print("TurnON")

 elif cmd.data['command']=='GOOD WEATHER':
 print("TurnOFF")
 elif cmd.data['command']=='HUMIDITY HIGH':
 print("TurnON")
 elif cmd.data['command']=='HUMIDITY LOW':
 print("TurnOFF")
```



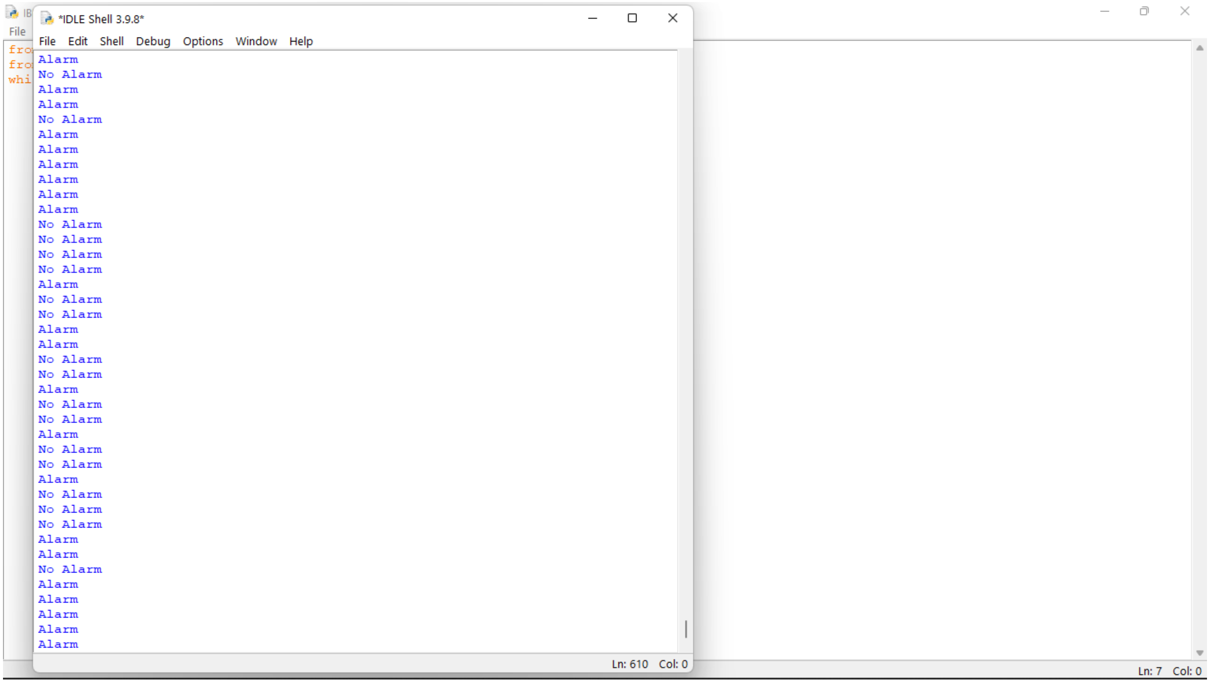
```
 elif cmd.data['command']=='HUMIDITY HIGH':
 print("TurnON")
 elif cmd.data['command']=='HUMIDITY LOW':
 print("TurnOFF")



 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()
```

PYTHON CODE FOR TEMPERATURE:

```
from random import *

from random import *

while True:

   temperature = randrange(0,100)

   humidity = randrange(0,100)

   if (temperature>=50):

      print("Alarm")

   else:

      print("No Alarm")
```



```
from random import *
from random import *
while True:
    temperature = randrange(0,100)
    humidity = randrange(0,100)
    if (temperature>=50):
        print("Alarm")
    else:
        print("No Alarm")
```



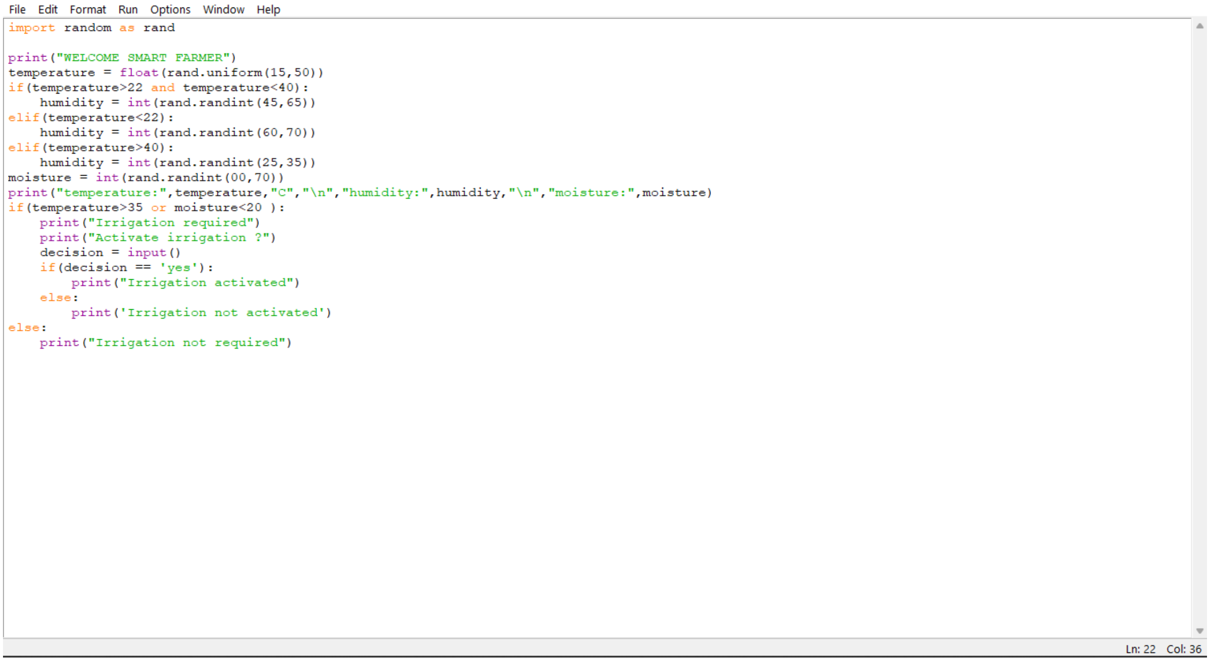
```
Alarm
No Alarm
Alarm
Alarm
No Alarm
Alarm
Alarm
Alarm
Alarm
Alarm
Alarm
No Alarm
No Alarm
No Alarm
No Alarm
Alarm
No Alarm
No Alarm
Alarm
Alarm
No Alarm
No Alarm
Alarm
No Alarm
No Alarm
Alarm
No Alarm
No Alarm
Alarm
No Alarm
No Alarm
No Alarm
Alarm
Alarm
No Alarm
Alarm
Alarm
Alarm
Alarm
Alarm
```

PYTHON CODE :

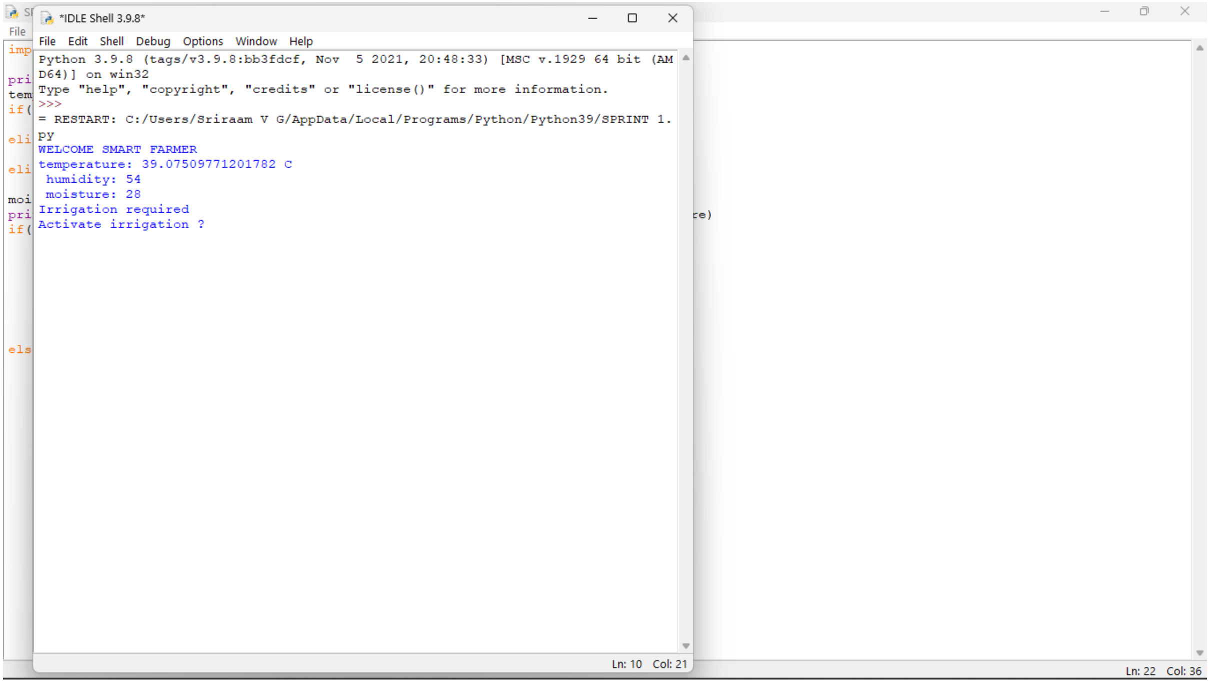
```
import random as rand
```

```
print("WELCOME SMART FARMER")
temperature = float(rand.uniform(15,50))
if(temperature>22 and temperature<40):
    humidity = int(rand.randint(45,65))
elif(temperature<22):
    humidity = int(rand.randint(60,70))
elif(temperature>40):
    humidity = int(rand.randint(25,35))
moisture = int(rand.randint(00,70))
print("temperature:",temperature,"C","\n","humidity:",humidity,"\n","moisture:",moisture)
if(temperature>35 or moisture<20 ):
    print("Irrigation required")
    print("Activate irrigation ?")
    decision = input()
    if(decision == 'yes'):
        print("Irrigation activated")
    else:
        print('Irrigation not activated')
else:
    print("Irrigation not required")
```

OUTPUT:



```
import random as rand

print("WELCOME SMART FARMER")
temperature = float(rand.uniform(15,50))
if(temperature>22 and temperature<40):
    humidity = int(rand.randint(45,65))
elif(temperature<22):
    humidity = int(rand.randint(60,70))
elif(temperature>40):
    humidity = int(rand.randint(25,35))
moisture = int(rand.randint(00,70))
print("temperature:",temperature,"C","\n","humidity:",humidity,"\n","moisture:",moisture)
if(temperature>35 or moisture<20 ):
    print("Irrigation required")
    print("Activate irrigation ?")
    decision = input()
    if(decision == 'yes'):
        print("Irrigation activated")
    else:
        print('Irrigation not activated')
else:
    print("Irrigation not required")
```



```
Python 3.9.8 (tags/v3.9.8:bb3fdcf, Nov  5 2021, 20:48:33) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
= RESTART: C:/Users/Sriraam V G/AppData/Local/Programs/Python/Python39/SPRINT 1.
py
WELCOME SMART FARMER
temperature: 39.07509771201782 C
 humidity: 54
 moisture: 28
Irrigation required
Activate irrigation ?
```