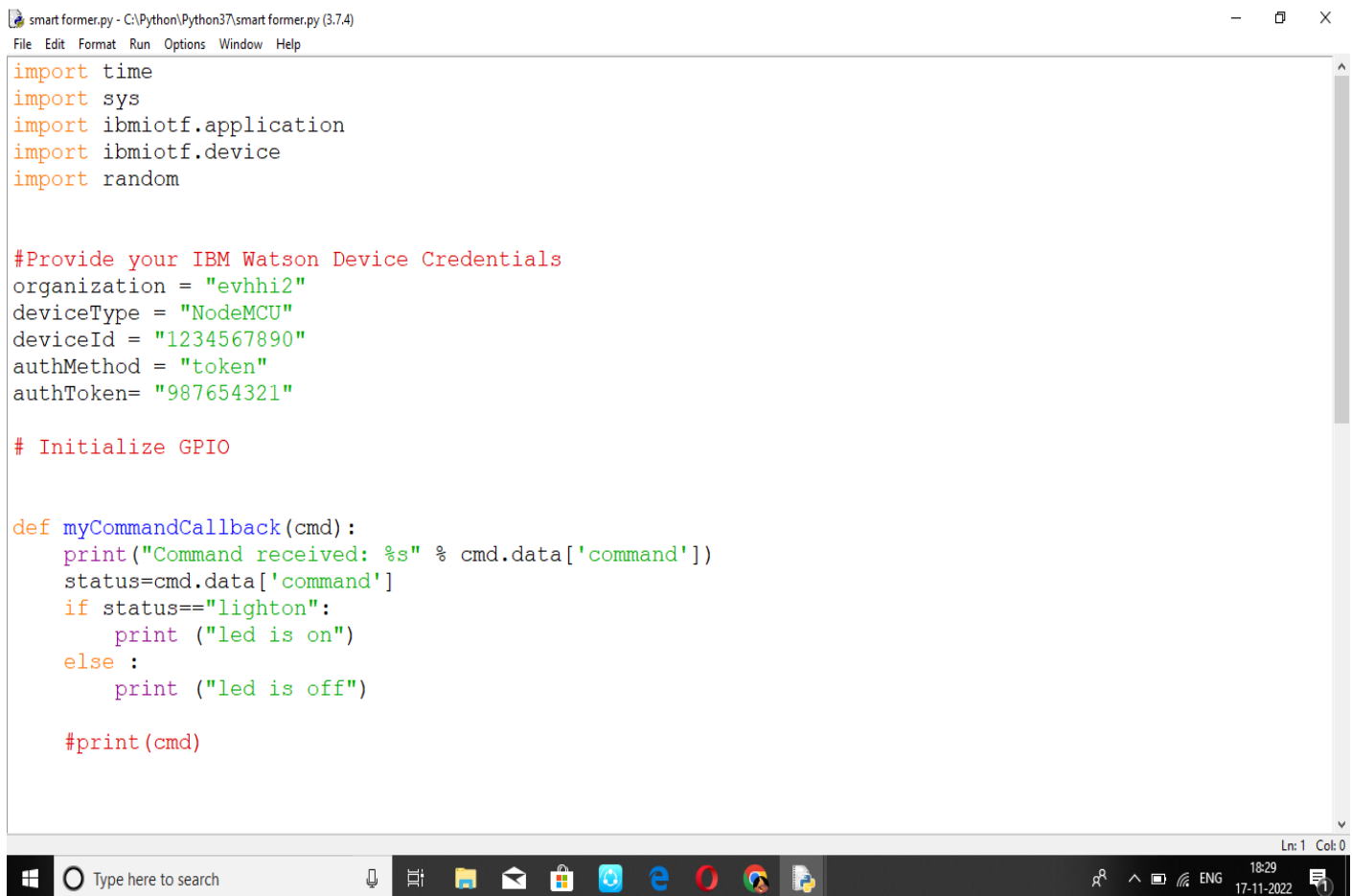


SPRINT-3

DATE	12 NOVEMBER 2022
TEAM ID	PNT2022TMID46164
PROJECT NAME	SMART FARMER-IOT ENABLED SMART FARMING APPLICATIONS

DEVELOP A PYTHON SCRIPT TO PUBLISH AND SUBSCRIBE TO IBM IOT PLATFORM

- ★ Develop the python code to publish and subscribe to the commands from the IBM cloud.



```
smart farmer.py - C:\Python\Python37\smart farmer.py (3.7.4)
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 = "evhhi2"
deviceType = "NodeMCU"
deviceId = "1234567890"
authMethod = "token"
authToken= "987654321"

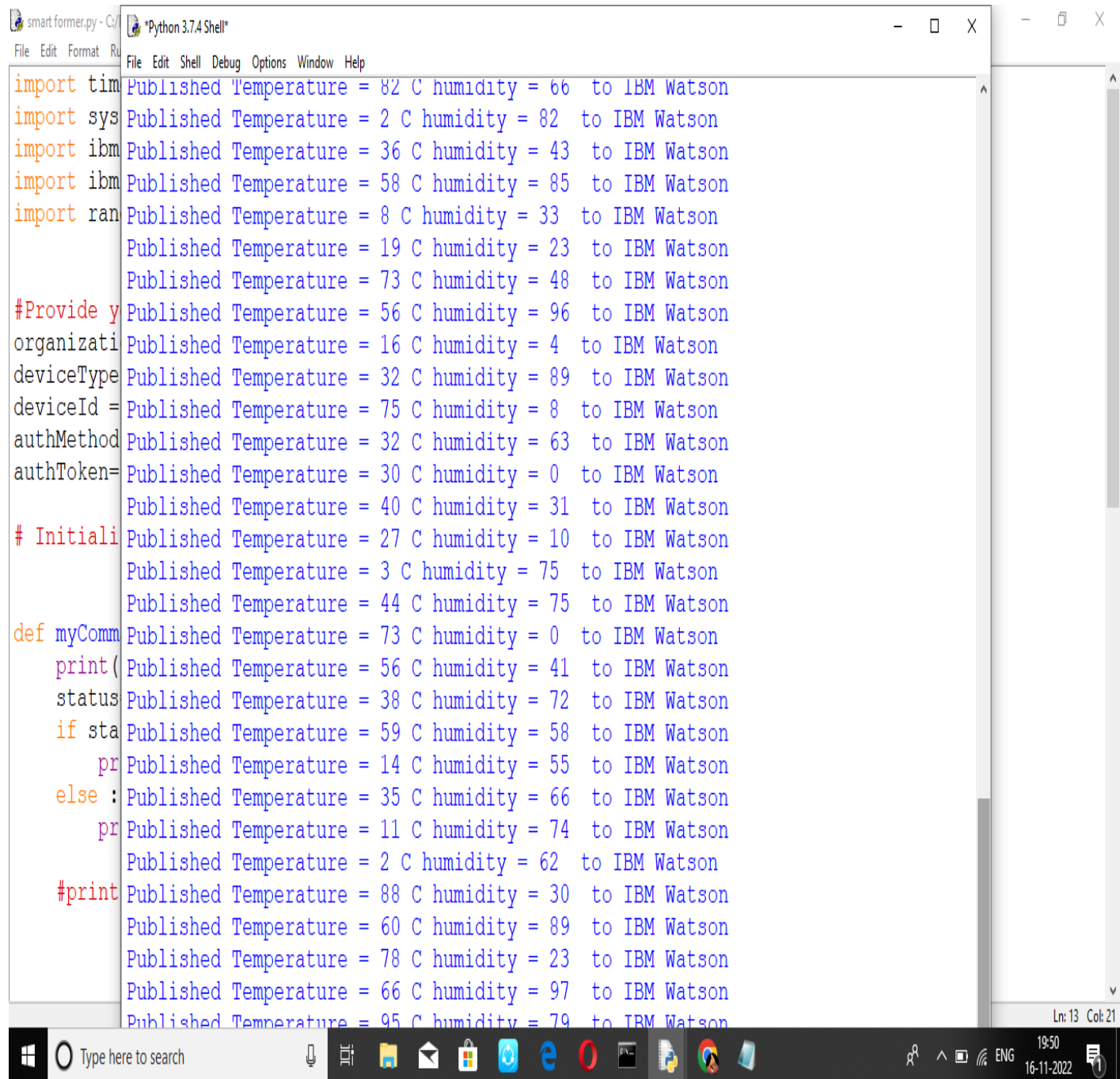
# Initialize GPIO

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("led is on")
    else :
        print ("led is off")

    #print(cmd)
```

The screenshot shows a code editor window with the title 'smart farmer.py - C:\Python\Python37\smart farmer.py (3.7.4)'. The menu bar includes 'File', 'Edit', 'Format', 'Run', 'Options', 'Window', and 'Help'. The code is a Python script for interacting with the IBM IoT platform. It imports 'time', 'sys', 'ibmiotf.application', 'ibmiotf.device', and 'random'. It defines IBM Watson Device Credentials: organization ('evhhi2'), deviceType ('NodeMCU'), deviceId ('1234567890'), authMethod ('token'), and authToken ('987654321'). It includes a comment '# Initialize GPIO'. A function 'myCommandCallback(cmd)' is defined, which prints the received command and its status, and handles 'lighton' and 'off' commands by printing 'led is on' or 'led is off'. The function also has a commented-out line '#print(cmd)'. The Windows taskbar at the bottom shows the Start button, a search bar, and several application icons. The system tray on the right shows the date and time as '17-11-2022 18:29'.

PYTHON CODING (OUTPUT):



The screenshot shows a Python 3.7.4 Shell window with a menu bar (File, Edit, Shell, Debug, Options, Window, Help) and a toolbar. The code in the editor is as follows:

```
import time
import sys
import ibmcloud
import ibmcloud__core
import random

#Provide your IBM Watson IoT Platform credentials
organizationId = 'your-organization-id'
deviceId = 'your-device-id'
authMethod = 'apikey'
authToken = 'your-api-token'

# Initialize the IBM Watson IoT Platform client
client = ibmcloud.IBMCloudIoTPlatformClient(organizationId, deviceId, authMethod, authToken)

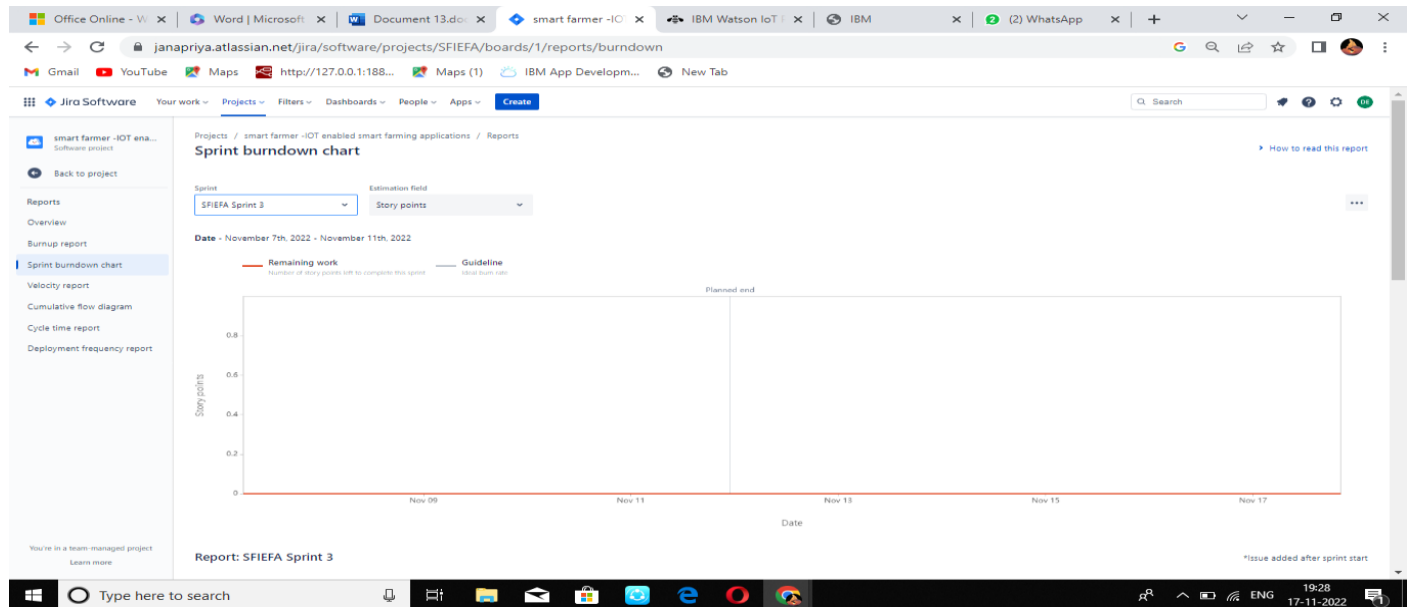
def myCommand():
    print('Temperature = 82 C humidity = 66 to IBM Watson')
    status = client.sendCommand(deviceId, 'setTemperature', 82)
    if status == 'success':
        print('Temperature = 2 C humidity = 82 to IBM Watson')
    else:
        print('Temperature = 36 C humidity = 43 to IBM Watson')
    print('Temperature = 58 C humidity = 85 to IBM Watson')
    print('Temperature = 8 C humidity = 33 to IBM Watson')
    print('Temperature = 19 C humidity = 23 to IBM Watson')
    print('Temperature = 73 C humidity = 48 to IBM Watson')
    print('Temperature = 56 C humidity = 96 to IBM Watson')
    print('Temperature = 16 C humidity = 4 to IBM Watson')
    print('Temperature = 32 C humidity = 89 to IBM Watson')
    print('Temperature = 75 C humidity = 8 to IBM Watson')
    print('Temperature = 32 C humidity = 63 to IBM Watson')
    print('Temperature = 30 C humidity = 0 to IBM Watson')
    print('Temperature = 40 C humidity = 31 to IBM Watson')
    print('Temperature = 27 C humidity = 10 to IBM Watson')
    print('Temperature = 3 C humidity = 75 to IBM Watson')
    print('Temperature = 44 C humidity = 75 to IBM Watson')
    print('Temperature = 73 C humidity = 0 to IBM Watson')
    print('Temperature = 56 C humidity = 41 to IBM Watson')
    print('Temperature = 38 C humidity = 72 to IBM Watson')
    print('Temperature = 59 C humidity = 58 to IBM Watson')
    print('Temperature = 14 C humidity = 55 to IBM Watson')
    print('Temperature = 35 C humidity = 66 to IBM Watson')
    print('Temperature = 11 C humidity = 74 to IBM Watson')
    print('Temperature = 2 C humidity = 62 to IBM Watson')
    print('Temperature = 88 C humidity = 30 to IBM Watson')
    print('Temperature = 60 C humidity = 89 to IBM Watson')
    print('Temperature = 78 C humidity = 23 to IBM Watson')
    print('Temperature = 66 C humidity = 97 to IBM Watson')
    print('Temperature = 95 C humidity = 79 to IBM Watson')
```

The output of the code is displayed in the shell window, showing the temperature and humidity values being sent to IBM Watson IoT Platform. The output is as follows:

```
Published Temperature = 82 C humidity = 66 to IBM Watson
Published Temperature = 2 C humidity = 82 to IBM Watson
Published Temperature = 36 C humidity = 43 to IBM Watson
Published Temperature = 58 C humidity = 85 to IBM Watson
Published Temperature = 8 C humidity = 33 to IBM Watson
Published Temperature = 19 C humidity = 23 to IBM Watson
Published Temperature = 73 C humidity = 48 to IBM Watson
Published Temperature = 56 C humidity = 96 to IBM Watson
Published Temperature = 16 C humidity = 4 to IBM Watson
Published Temperature = 32 C humidity = 89 to IBM Watson
Published Temperature = 75 C humidity = 8 to IBM Watson
Published Temperature = 32 C humidity = 63 to IBM Watson
Published Temperature = 30 C humidity = 0 to IBM Watson
Published Temperature = 40 C humidity = 31 to IBM Watson
Published Temperature = 27 C humidity = 10 to IBM Watson
Published Temperature = 3 C humidity = 75 to IBM Watson
Published Temperature = 44 C humidity = 75 to IBM Watson
Published Temperature = 73 C humidity = 0 to IBM Watson
Published Temperature = 56 C humidity = 41 to IBM Watson
Published Temperature = 38 C humidity = 72 to IBM Watson
Published Temperature = 59 C humidity = 58 to IBM Watson
Published Temperature = 14 C humidity = 55 to IBM Watson
Published Temperature = 35 C humidity = 66 to IBM Watson
Published Temperature = 11 C humidity = 74 to IBM Watson
Published Temperature = 2 C humidity = 62 to IBM Watson
Published Temperature = 88 C humidity = 30 to IBM Watson
Published Temperature = 60 C humidity = 89 to IBM Watson
Published Temperature = 78 C humidity = 23 to IBM Watson
Published Temperature = 66 C humidity = 97 to IBM Watson
Published Temperature = 95 C humidity = 79 to IBM Watson
```

The status bar at the bottom of the window shows the current line and column: Ln: 13 Col: 21. The Windows taskbar is visible at the bottom of the screen, showing the search bar, taskbar icons, and system tray.

BURNDOWN CHART:



ROADMAP:

