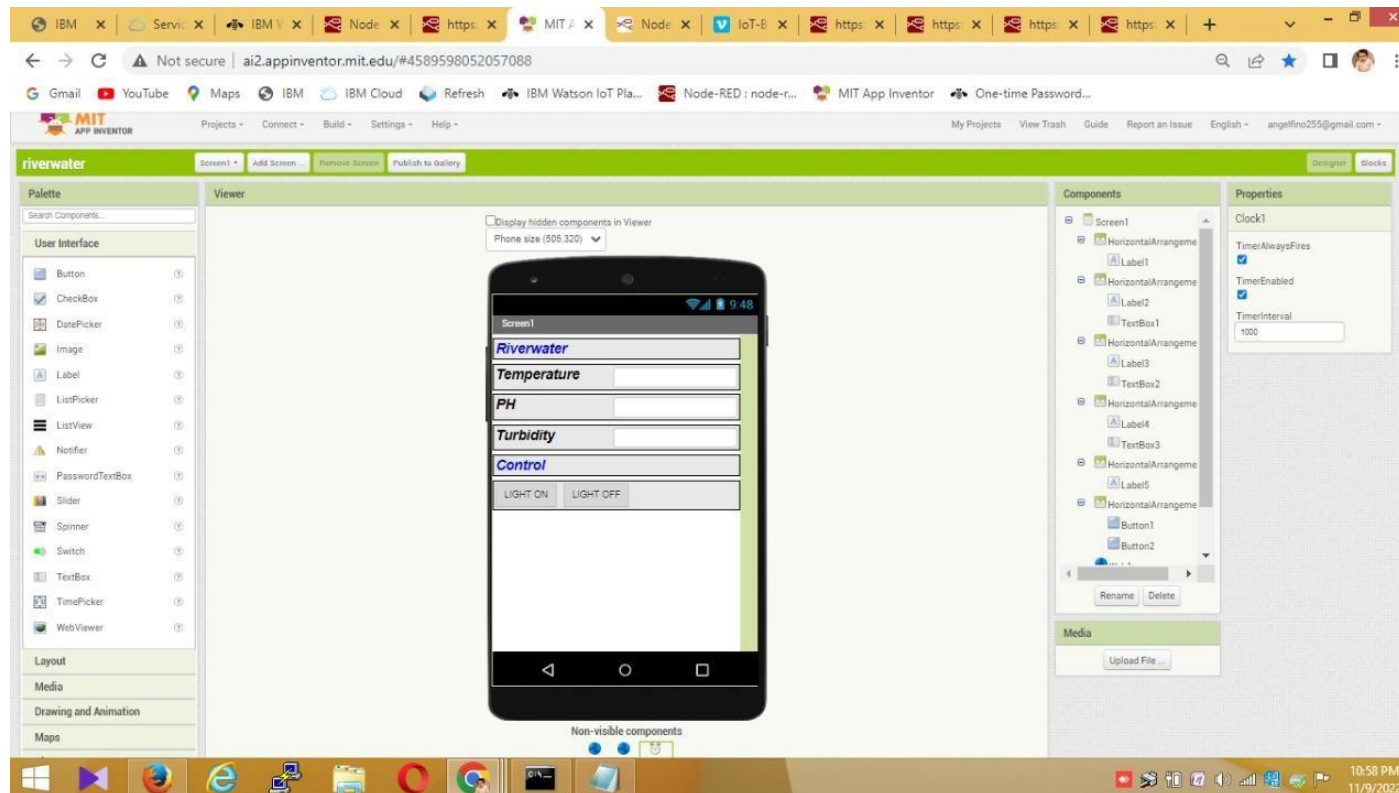


Project Development Phase Sprint-3

Date	12-NOVEMBER 2022
Team ID	PNT2022TMID35688
Project name	Project – IoT Based Real-time River water quality monitoring and control system

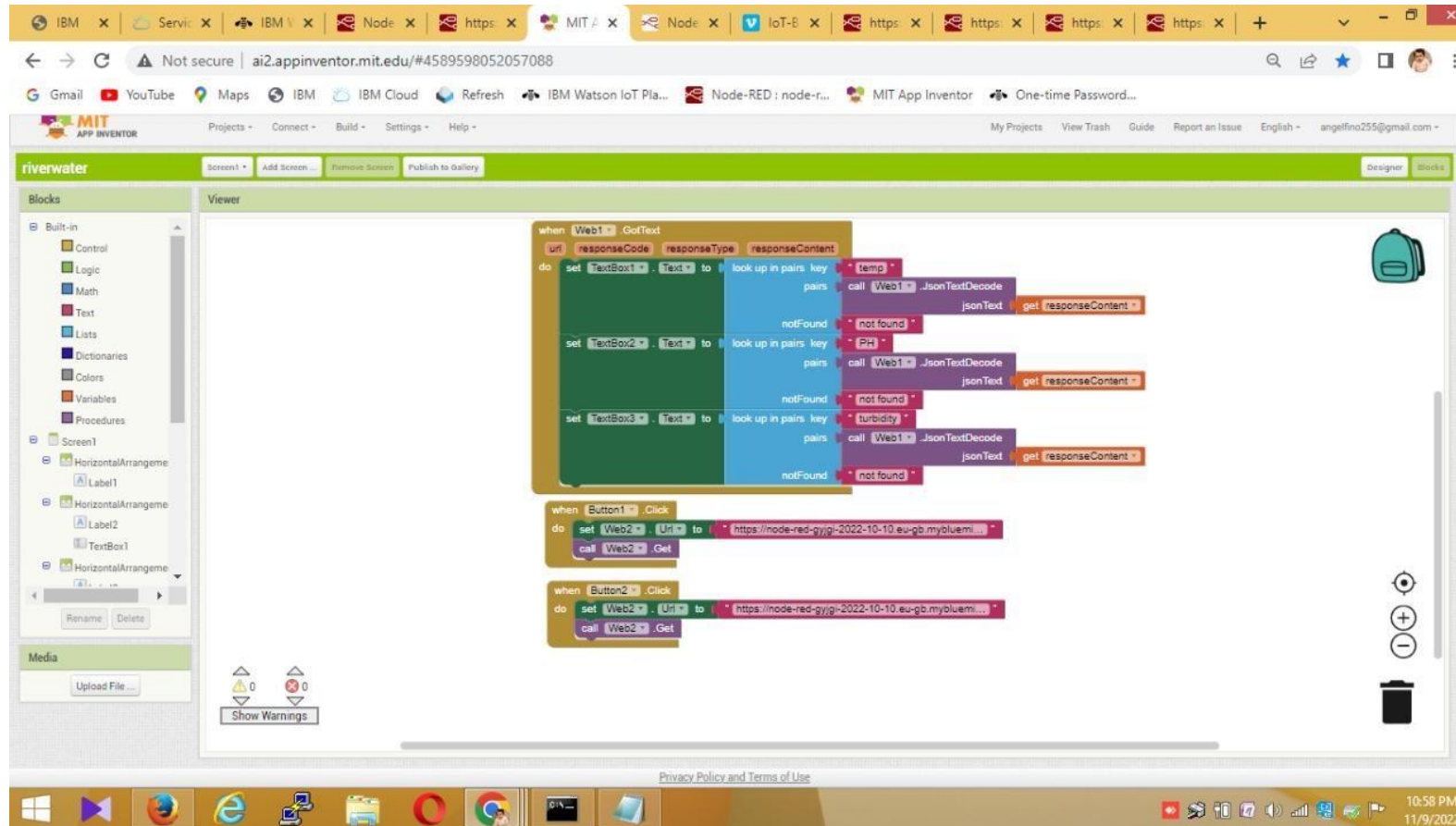
USN-11

As a user, I can design the front end in MIT app inventor.



USN-12

As a user, I can design the back end(blocks) in MIT app inventor



USN-13

As a user, I can develop the python script.

```
iet.py - C:\Users\chandra.mohan\Documents\IBM ASSIGNMENT\iet.py (3.7.0)
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 = "rv07c6"
deviceType = "Riverwaterquality-22_23"
deviceId = "123456"
authMethod = "token"
authToken = "wQ_43L5c0@ku8)sgd"

# 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)
```

```
iet.py - C:\Users\chandra.mohan\Documents\IBM ASSIGNMENT\iet.py (3.7.0)
File Edit Format Run Options Window Help

try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": 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"
deviceCli.connect()

while True:
    #Get Sensor Data from DHT11

    temp=random.randint(0,100)
    ph=random.randint(0,14)
    turb=random.randint(0,100)

    data = { 'temperature' : temp, 'ph': ph, 'turbidity' : turb }
    #print data
    def myOnPublishCallback():
        print ("Published Temperature = %s C" % temp, "ph = %s" % ph, "turbidity = %s NTU" % turb)
    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallba
    if not success:
        print("Not connected to IoT")
    time.sleep(1)
```

PROGRAM:

```
import ibmiotf.device
import random
```

```
#Provide your IBM Watson Device Credentials
organization = "rv07c6"
deviceType = "riverwaterquality-22_23"
deviceId = "123456"
authMethod = "token"
authToken = "wQ_)43L5c0@ku8)sgd"
```

```
# 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)
```

```
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:
    #Get Sensor Data from DHT11

    temp=random.randint(0,100)
    ph=random.randint(0,14)
    turb=random.randint(0,100)

    data = { 'temperature' : temp, 'ph': ph,'turbidity' :turb }
    #print data
    def myOnPublishCallback():
        print ("Published Temperature = %s C" % temp, "ph = %s %" % ph,"turbidity = %s NTU " % turb ,"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)

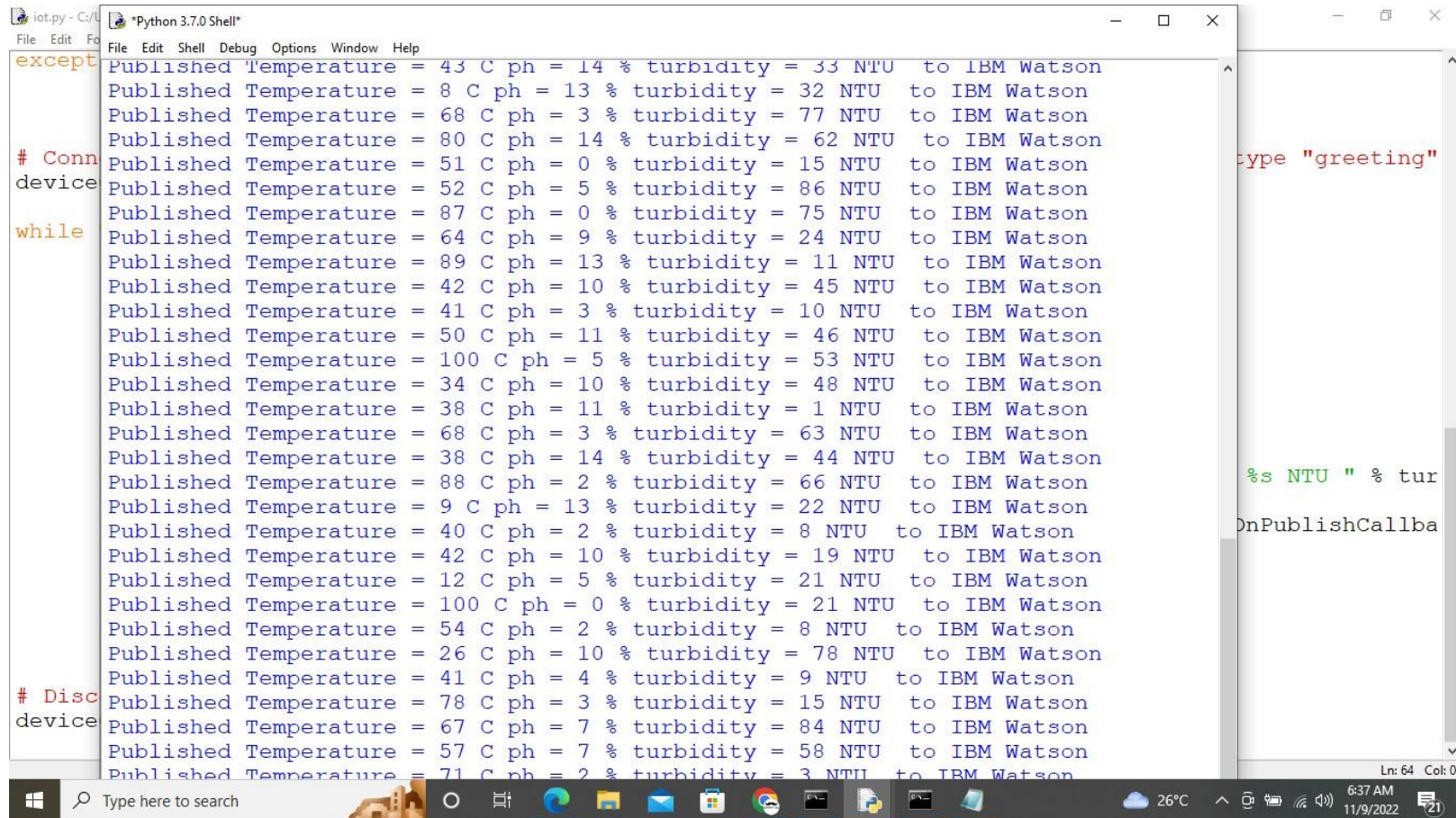
    deviceCli.commandCallback = myCommandCallback

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

```

USN-14

As a user,I can get the output of the program with the parameters



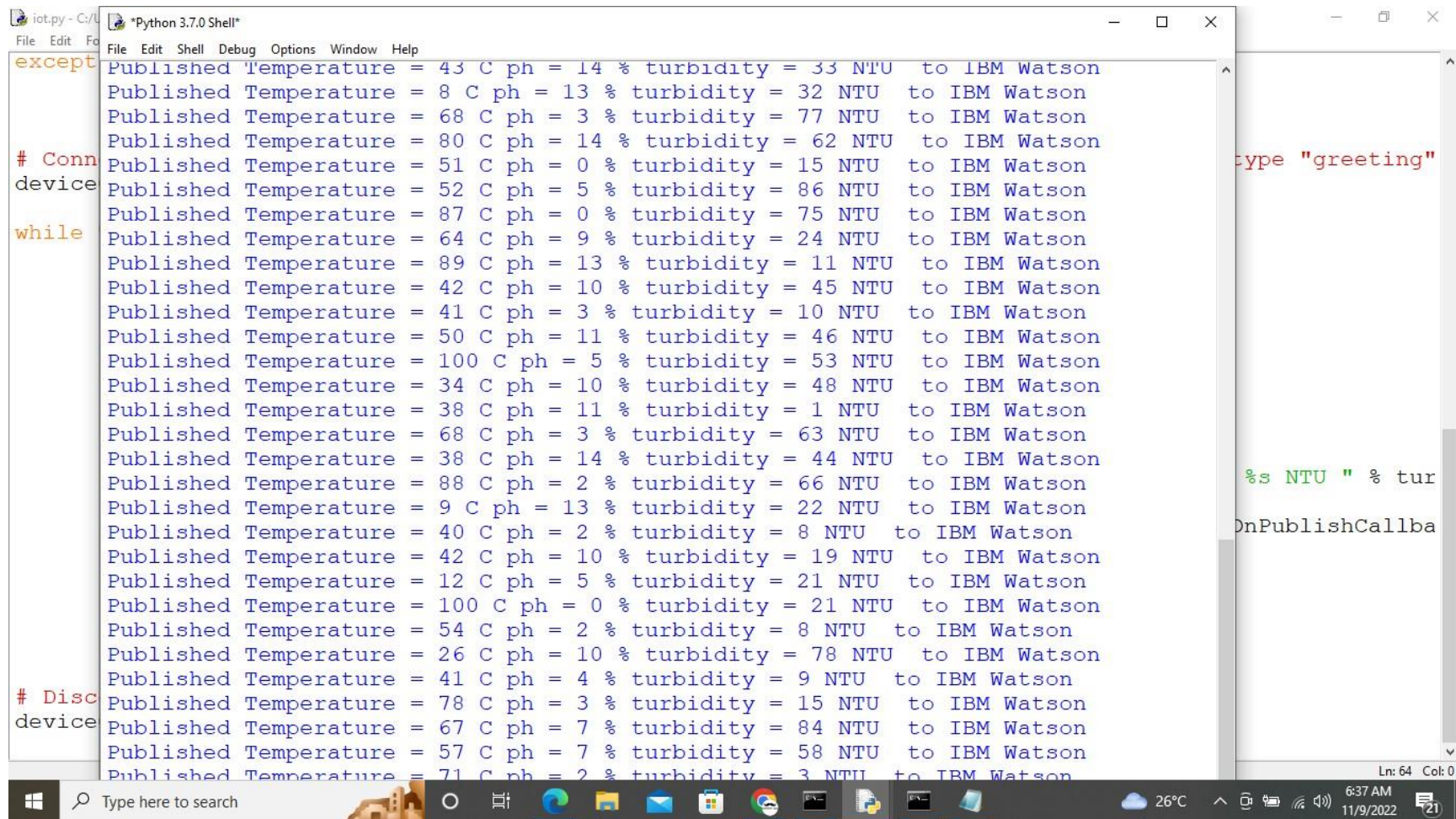
The screenshot shows a Windows desktop with a Python 3.7.0 Shell window in the foreground and a code editor in the background. The shell window displays the output of a program, which consists of 30 lines of data. Each line starts with 'Published Temperature' followed by a value, then 'C ph =', another value, '% turbidity =', a third value, and finally 'to IBM Watson'. The values for temperature, pH, and turbidity vary across the lines. The code editor in the background shows a Python script with a while loop and a print statement. The print statement uses string formatting to output the temperature, pH, and turbidity values, followed by a newline character. The status bar at the bottom of the shell window shows 'Ln: 64 Col: 0'.

```
Published Temperature = 43 C ph = 14 % turbidity = 33 NTU to IBM Watson
Published Temperature = 8 C ph = 13 % turbidity = 32 NTU to IBM Watson
Published Temperature = 68 C ph = 3 % turbidity = 77 NTU to IBM Watson
Published Temperature = 80 C ph = 14 % turbidity = 62 NTU to IBM Watson
Published Temperature = 51 C ph = 0 % turbidity = 15 NTU to IBM Watson
Published Temperature = 52 C ph = 5 % turbidity = 86 NTU to IBM Watson
Published Temperature = 87 C ph = 0 % turbidity = 75 NTU to IBM Watson
Published Temperature = 64 C ph = 9 % turbidity = 24 NTU to IBM Watson
Published Temperature = 89 C ph = 13 % turbidity = 11 NTU to IBM Watson
Published Temperature = 42 C ph = 10 % turbidity = 45 NTU to IBM Watson
Published Temperature = 41 C ph = 3 % turbidity = 10 NTU to IBM Watson
Published Temperature = 50 C ph = 11 % turbidity = 46 NTU to IBM Watson
Published Temperature = 100 C ph = 5 % turbidity = 53 NTU to IBM Watson
Published Temperature = 34 C ph = 10 % turbidity = 48 NTU to IBM Watson
Published Temperature = 38 C ph = 11 % turbidity = 1 NTU to IBM Watson
Published Temperature = 68 C ph = 3 % turbidity = 63 NTU to IBM Watson
Published Temperature = 38 C ph = 14 % turbidity = 44 NTU to IBM Watson
Published Temperature = 88 C ph = 2 % turbidity = 66 NTU to IBM Watson
Published Temperature = 9 C ph = 13 % turbidity = 22 NTU to IBM Watson
Published Temperature = 40 C ph = 2 % turbidity = 8 NTU to IBM Watson
Published Temperature = 42 C ph = 10 % turbidity = 19 NTU to IBM Watson
Published Temperature = 12 C ph = 5 % turbidity = 21 NTU to IBM Watson
Published Temperature = 100 C ph = 0 % turbidity = 21 NTU to IBM Watson
Published Temperature = 54 C ph = 2 % turbidity = 8 NTU to IBM Watson
Published Temperature = 26 C ph = 10 % turbidity = 78 NTU to IBM Watson
Published Temperature = 41 C ph = 4 % turbidity = 9 NTU to IBM Watson
Published Temperature = 78 C ph = 3 % turbidity = 15 NTU to IBM Watson
Published Temperature = 67 C ph = 7 % turbidity = 84 NTU to IBM Watson
Published Temperature = 57 C ph = 7 % turbidity = 58 NTU to IBM Watson
Published Temperature = 71 C ph = 2 % turbidity = 3 NTU to IBM Watson
```

```
except
# Conn
device
while
type "greeting"
%s NTU " % tur
OnPublishCallba
```


USN-15

As a user , I can get the commands in the output when the buttons are pressed.



```
File Edit Shell Debug Options Window Help
except Published Temperature = 43 C ph = 14 % turbidity = 33 NTU to IBM Watson
Published Temperature = 8 C ph = 13 % turbidity = 32 NTU to IBM Watson
Published Temperature = 68 C ph = 3 % turbidity = 77 NTU to IBM Watson
Published Temperature = 80 C ph = 14 % turbidity = 62 NTU to IBM Watson
# Conn Published Temperature = 51 C ph = 0 % turbidity = 15 NTU to IBM Watson
device Published Temperature = 52 C ph = 5 % turbidity = 86 NTU to IBM Watson
while Published Temperature = 87 C ph = 0 % turbidity = 75 NTU to IBM Watson
Published Temperature = 64 C ph = 9 % turbidity = 24 NTU to IBM Watson
Published Temperature = 89 C ph = 13 % turbidity = 11 NTU to IBM Watson
Published Temperature = 42 C ph = 10 % turbidity = 45 NTU to IBM Watson
Published Temperature = 41 C ph = 3 % turbidity = 10 NTU to IBM Watson
Published Temperature = 50 C ph = 11 % turbidity = 46 NTU to IBM Watson
Published Temperature = 100 C ph = 5 % turbidity = 53 NTU to IBM Watson
Published Temperature = 34 C ph = 10 % turbidity = 48 NTU to IBM Watson
Published Temperature = 38 C ph = 11 % turbidity = 1 NTU to IBM Watson
Published Temperature = 68 C ph = 3 % turbidity = 63 NTU to IBM Watson
Published Temperature = 38 C ph = 14 % turbidity = 44 NTU to IBM Watson
Published Temperature = 88 C ph = 2 % turbidity = 66 NTU to IBM Watson
Published Temperature = 9 C ph = 13 % turbidity = 22 NTU to IBM Watson
Published Temperature = 40 C ph = 2 % turbidity = 8 NTU to IBM Watson
Published Temperature = 42 C ph = 10 % turbidity = 19 NTU to IBM Watson
Published Temperature = 12 C ph = 5 % turbidity = 21 NTU to IBM Watson
Published Temperature = 100 C ph = 0 % turbidity = 21 NTU to IBM Watson
Published Temperature = 54 C ph = 2 % turbidity = 8 NTU to IBM Watson
Published Temperature = 26 C ph = 10 % turbidity = 78 NTU to IBM Watson
Published Temperature = 41 C ph = 4 % turbidity = 9 NTU to IBM Watson
# Disc Published Temperature = 78 C ph = 3 % turbidity = 15 NTU to IBM Watson
device Published Temperature = 67 C ph = 7 % turbidity = 84 NTU to IBM Watson
Published Temperature = 57 C ph = 7 % turbidity = 58 NTU to IBM Watson
Published Temperature = 71 C ph = 2 % turbidity = 3 NTU to IBM Watson

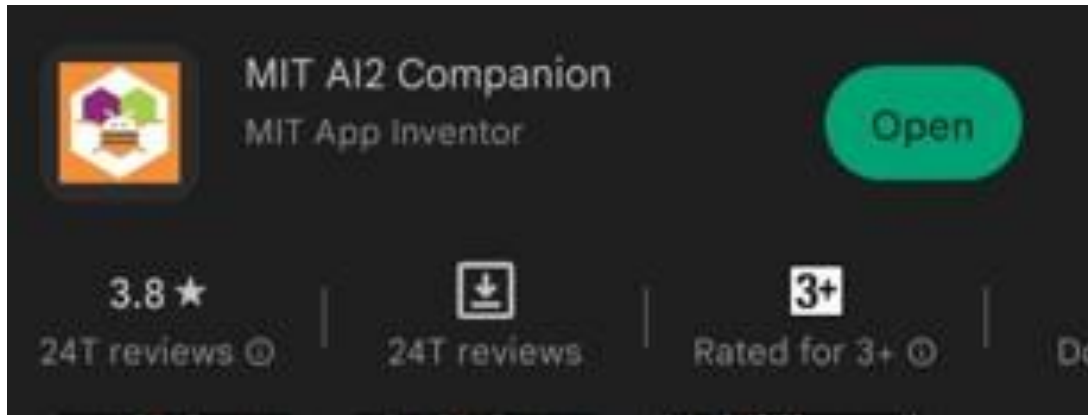
type "greeting"

%s NTU " % tur
OnPublishCallba

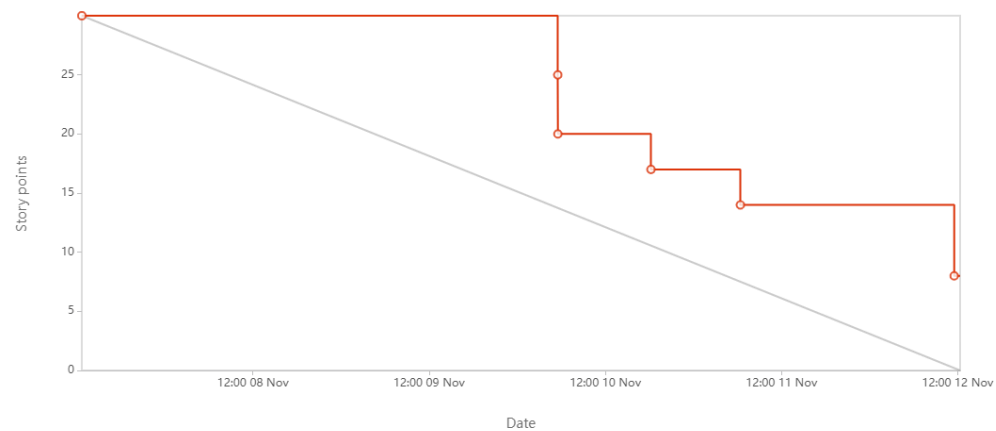
Ln: 64 Col: 0
```

USN-16

As a user, I can download MIT AI2 companion app in my mobile



SPRINT BURNDOWN CHART:



VELOCITY CHART:

