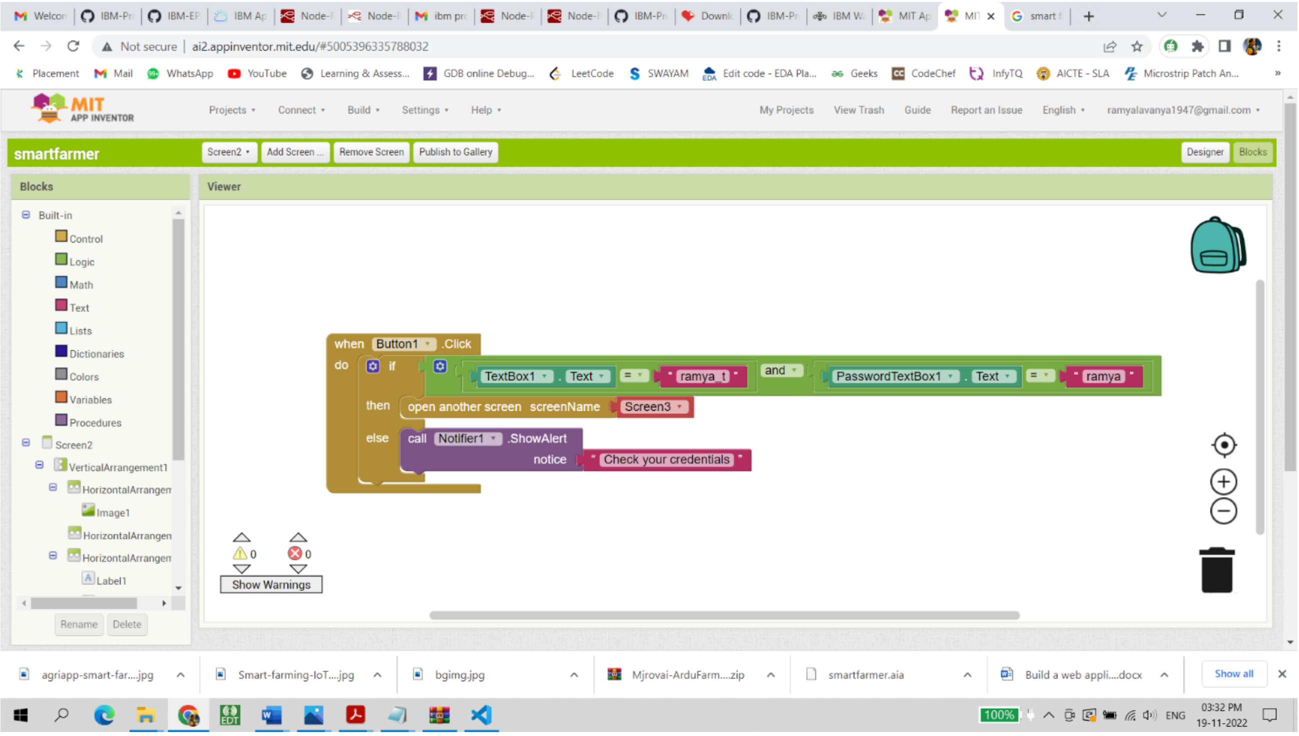
Project Delivery Sprint - 1

Date	20 Oct 2022
Team ID	PNT2022TMID04704
Project Name	Smart Farmer-IOT Enabled Smart Farming Application

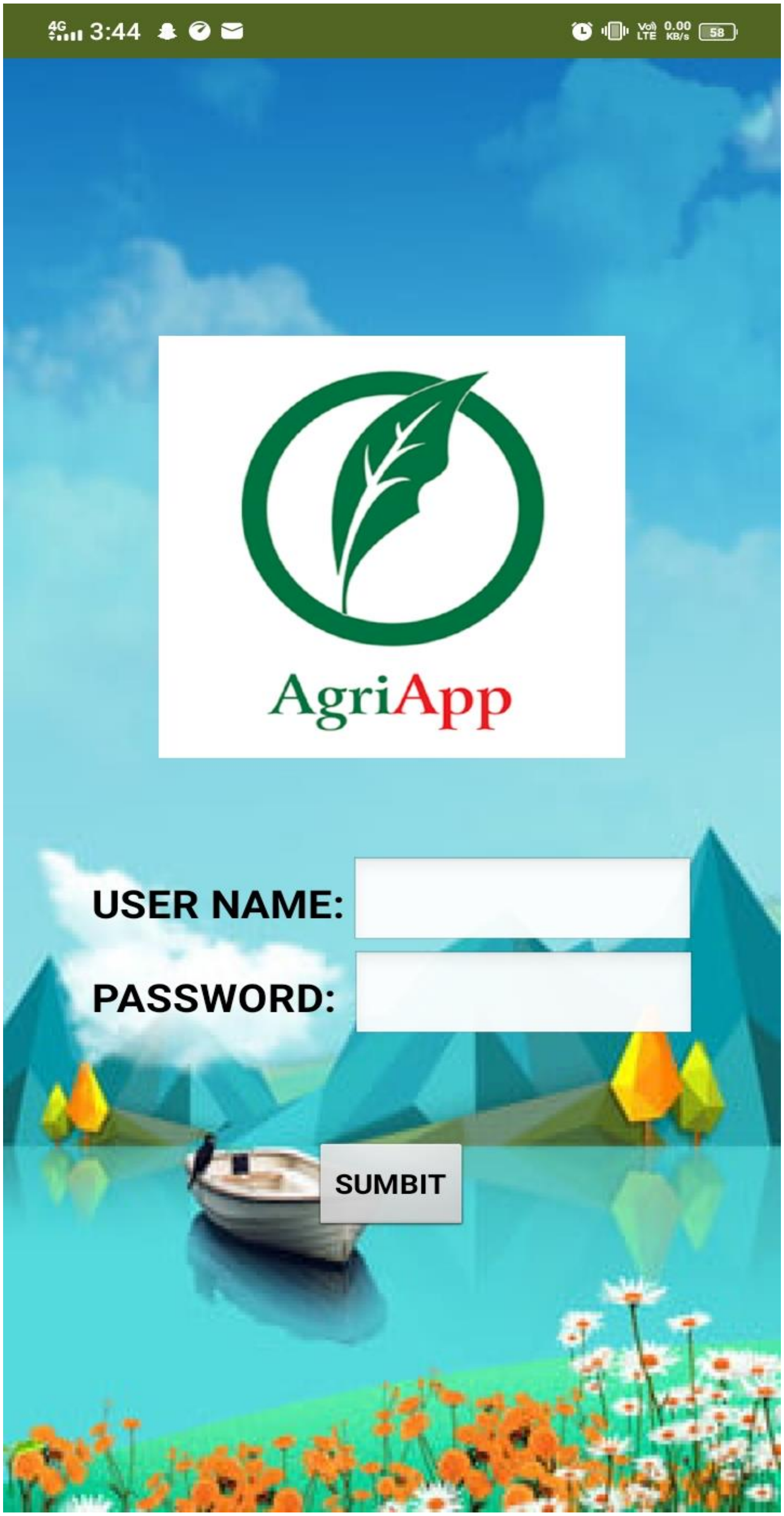
Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story /Task**
Sprint-1	Registration (Farmer)	USN-1	As a user, I can registerfor the application by entering my username, password.

Block diagram → Registration (Farmer)



```
when Button1 .Click
do  if  TextBox1 . Text = " ramya_t " and PasswordTextBox1 . Text = " ramya "
    then  open another screen  screenName  Screen3
    else  call Notifier1 .ShowAlert
                         notice  " Check your credentials "
```

Mobile App page

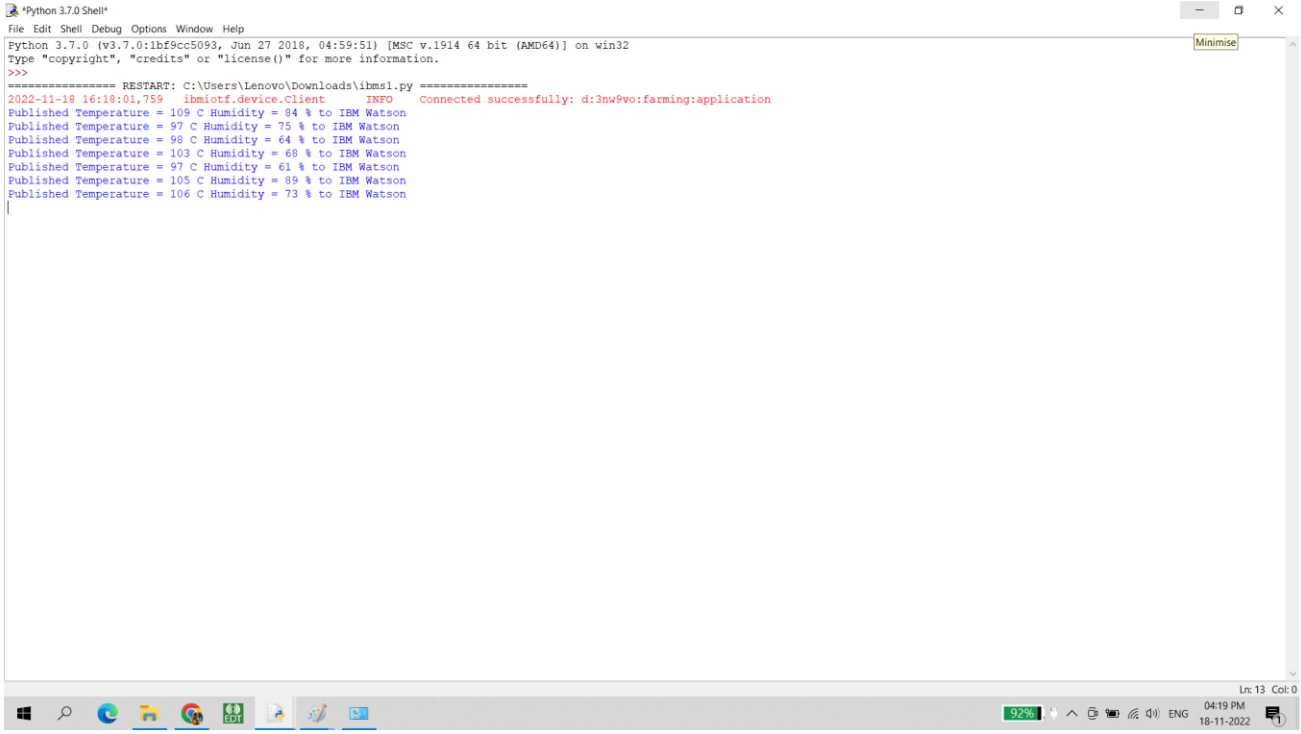

3:44
AgriApp
USER NAME:
PASSWORD:
SUMBIT

Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story /Task**
Sprint-1	IBM IoT cloud Service	USN-2	Publish and subscribe to IBM IoT cloud

Python code Connect With IBM IoT Cloud Service

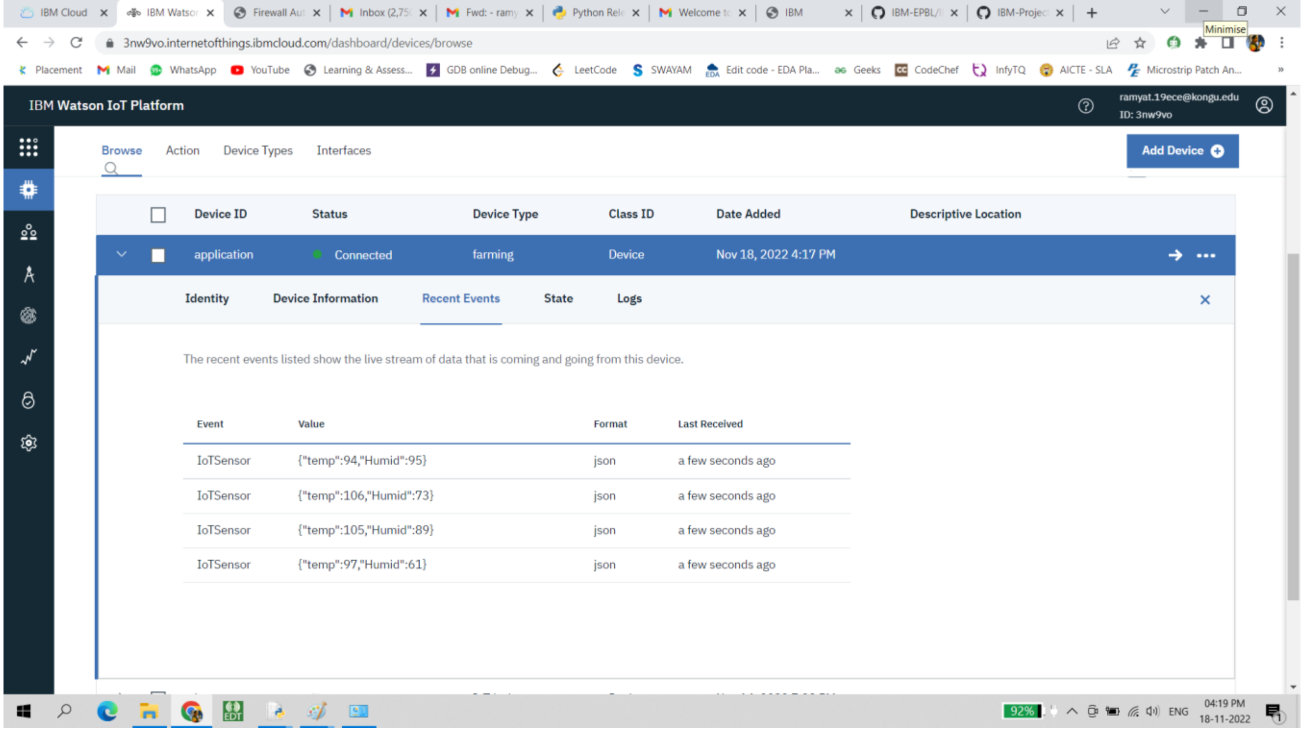
```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

#Provide your IBM Watson Device Credentials
organization = "3nw9vo"
deviceType = "farming"
deviceId = "application"
authMethod = "token"
authToken = "87654321"

# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="motoron":
        print ("Motor is on")
    elif status == "motoroff":
        print ("Motor is off")
    else :
        print ("please send proper command")

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(90,110)
        Humid=random.randint(60,100)

        data = { 'temp' : temp, 'Humid': Humid }
        #print data
        def myOnPublishCallback():
          print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid, "to IBM
Watson")

        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
        if not success:
          print("Not connected to IoTF")
        time.sleep(10)

        deviceCli.commandCallback = myCommandCallback

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

OUTPUT:



```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
================ RESTART: C:\Users\Lenovo\Downloads\ibms1.py ================
2022-11-18 16:18:01,759   ibmiotf.device.Client      INFO    Connected successfully: d:3nw9vo:farming:application
Published Temperature = 109 C Humidity = 84 % to IBM Watson
Published Temperature = 97 C Humidity = 75 % to IBM Watson
Published Temperature = 98 C Humidity = 64 % to IBM Watson
Published Temperature = 103 C Humidity = 68 % to IBM Watson
Published Temperature = 97 C Humidity = 61 % to IBM Watson
Published Temperature = 105 C Humidity = 89 % to IBM Watson
Published Temperature = 106 C Humidity = 73 % to IBM Watson
```

Data received



□	Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
□	application	● Connected	farming	Device	Nov 18, 2022 4:17 PM	

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
IoTSensor	{"temp":94,"Humid":95}	json	a few seconds ago
IoTSensor	{"temp":106,"Humid":73}	json	a few seconds ago
IoTSensor	{"temp":105,"Humid":89}	json	a few seconds ago
IoTSensor	{"temp":97,"Humid":61}	json	a few seconds ago