

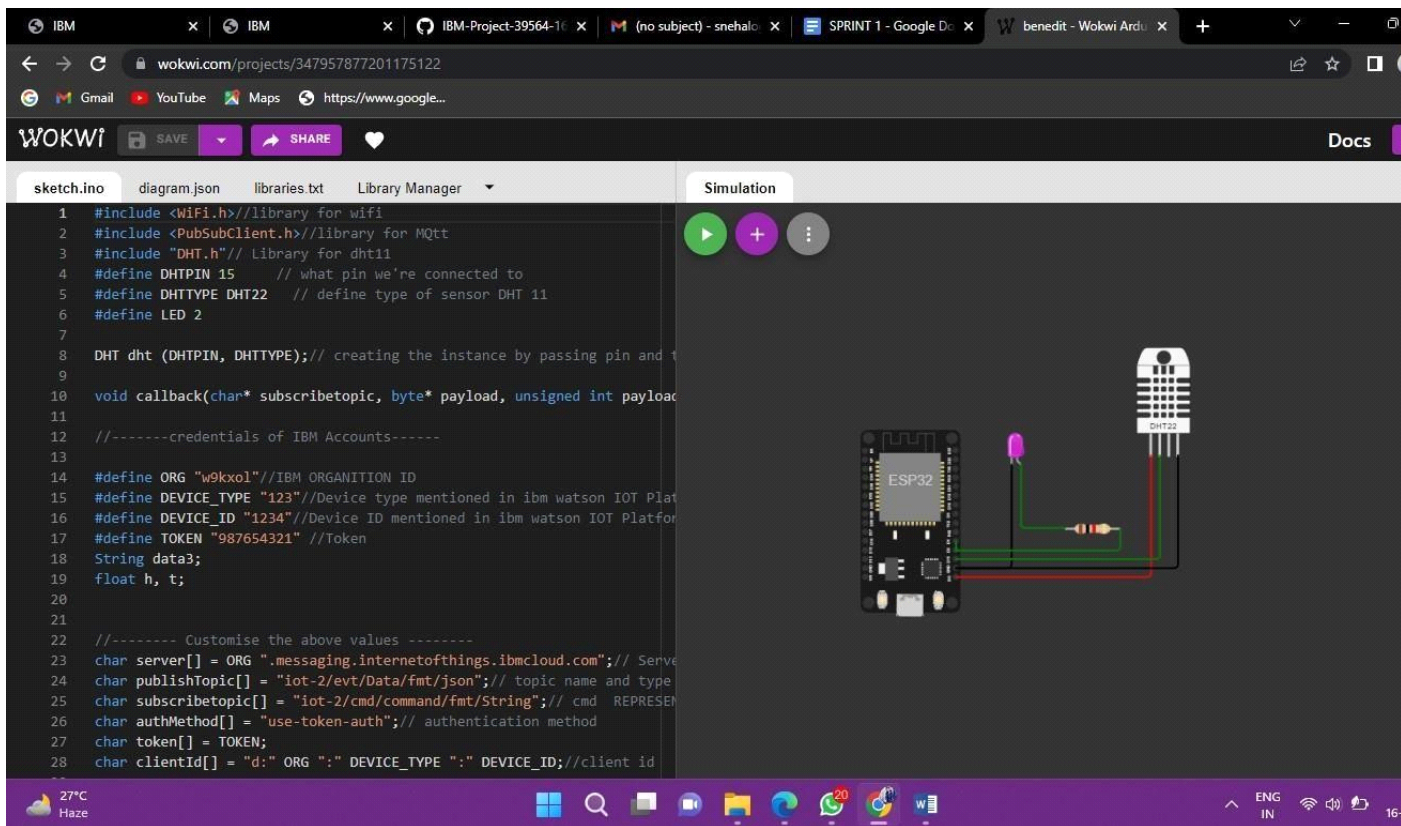
**SMART FARMER – IOT ENABLEDD SMART  
FARMIN  
GAPPLICATION  
PROJECT DEVELOPMENT – DELIVERY OF  
SPRINT - 1**

<b>DATE</b>	17 NOVEMBER 2022
<b>TITLE</b>	SMART FARMER – IOT ENABLED SMART FARMING APPLICATION
<b>TEAM ID</b>	PNT2022TMID11130
<b>TEAM LEADER NAME</b>	Yogehswaran A
<b>TEAM MEMBER NAME</b>	Vijay G Ajeeth A Vasanthakumar V

**Connect Sensor in ESP8266**

**CIRCUIT**

**DIAGRAM:**



## Develop a Python Code:

**Code:** import time  
import sys  
import ibmiotf.application  
import ibmiotf.device  
import random

#Provide your IBM Watson Device  
Credentials  
organization = "w9kxol"  
deviceType = "123" deviceId = "1234"  
authMethod = "token" authToken =  
"987654321"

# 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, "authmethod":
authMethod, "auth-token": authToken}
deviceCli
=
        ibmiotf.device.Client(deviceOptions
        ) #
        ..... exce
        pt

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)

    moist=random.randint(50,120)    data = { 'temp'
:temp, 'Humid': Humid , 'moist':moist}
    #print data def myOnPublishCallback():
        pri
nt("Published Temperature = %s C" % temp,
"Humidity = %s %% "
% Humid, "soilmoisture=%s %% " % moist, "to IBM
Watson")

```

```
    success =  
    deviceCli.publishEvent("IoTSensor","json",  
    data,  
    qos=0, on_publish=myOnPublishCallback  
    )  
    if not success: print("Not  
connectedto IoTf")  
    time.sleep(10)  
  
    deviceCli.commandCallback = myCommandCallback  
  
# Disconnect the device and application from the  
clouddeviceCli.disconnect()
```

**OUTPUT:**

```
File Edit Format Run Options Windows Help
import time
import sys
import ibmiotf.application
import ibmiotf.device

#Replace your IBM Watson Device Credentials
organization = "wskcc1"
deviceType = "123"
deviceId = "1234"
authMethod = "token"
authToken = "997454321"

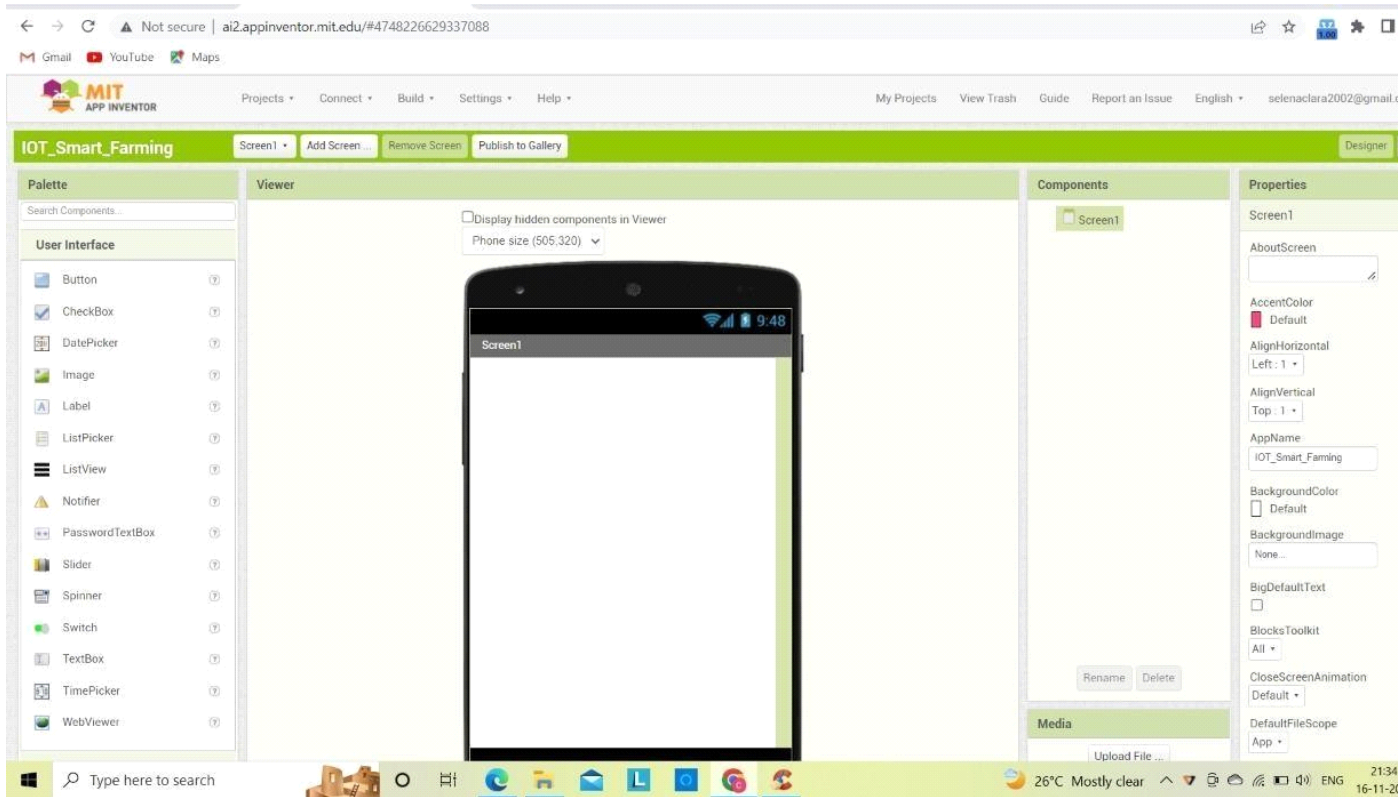
# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="motoron":
        print ("motor is on")
        #if status == "motoroff":
            print ("motor is off")
        else:
            print ("please send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

try:
    deviceOptions = {"org": organization, "type":
deviceCli = ibmiotf.device.Client(deviceOptions)
#.....

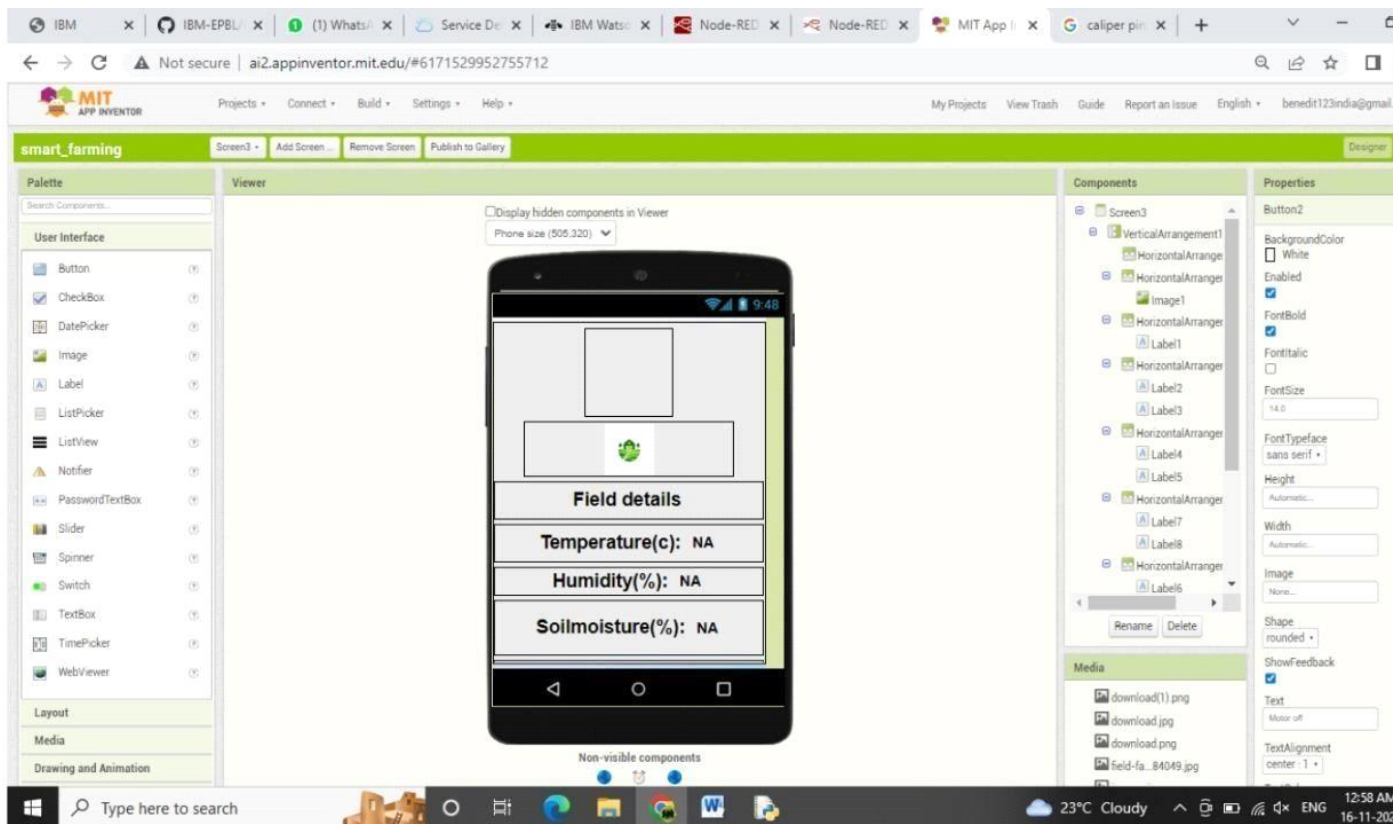
except Exception as e:
    print("Caught exception connecting device: %s" %
sys.exit(1))
```

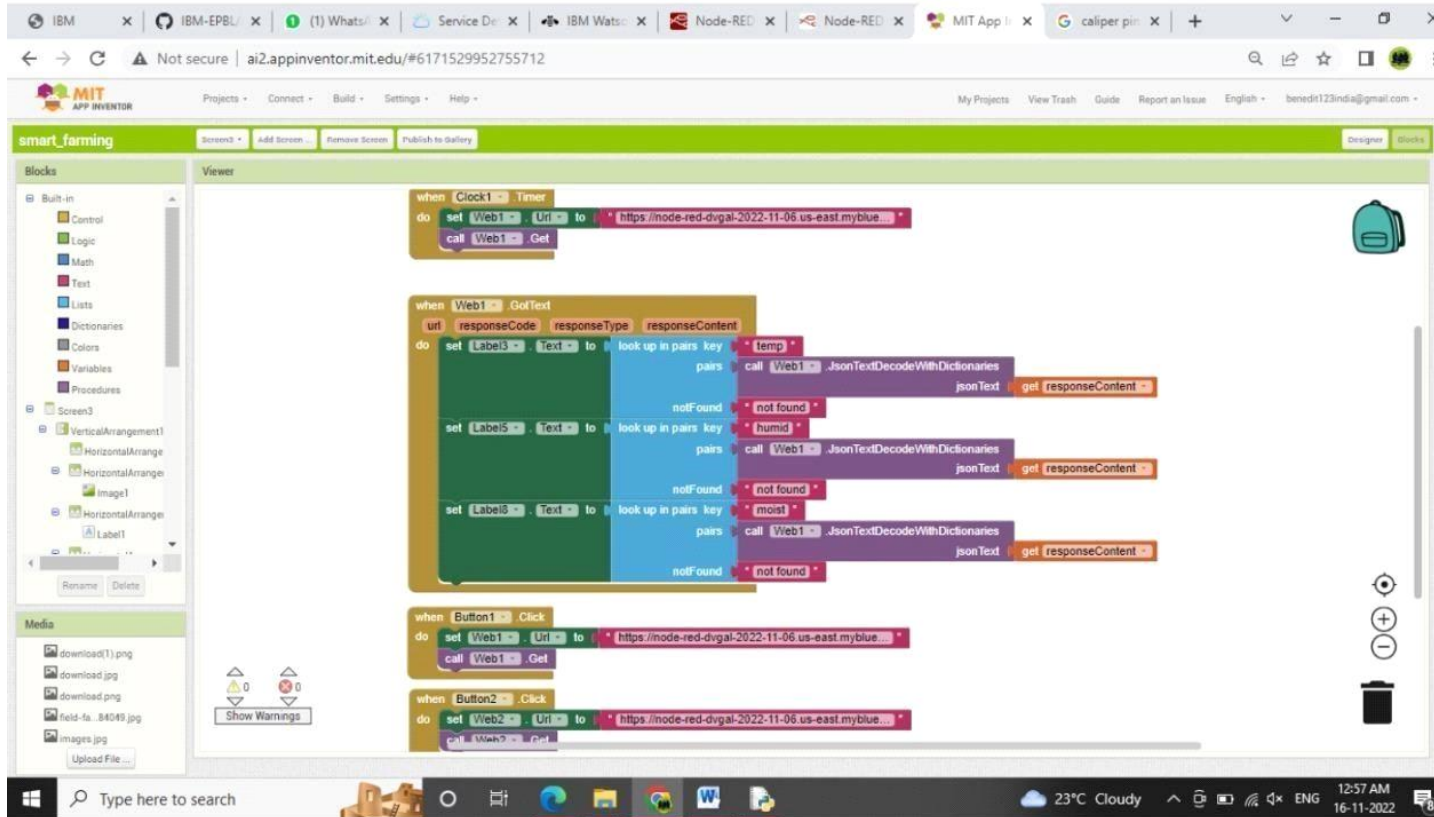
```
File Edit Shell Debug Console Window Help
Published Temperature = 90 C Humidity = 80 % soilmoisture=76 % to IBM Watson
Published Temperature = 100 C Humidity = 63 % soilmoisture=16 % to IBM Watson
Published Temperature = 110 C Humidity = 61 % soilmoisture=86 % to IBM Watson
Published Temperature = 100 C Humidity = 71 % soilmoisture=83 % to IBM Watson
Command received: motoron
motor is on
Published Temperature = 104 C Humidity = 81 % soilmoisture=101 % to IBM Watson
Published Temperature = 100 C Humidity = 74 % soilmoisture=78 % to IBM Watson
Published Temperature = 90 C Humidity = 89 % soilmoisture=53 % to IBM Watson
Published Temperature = 105 C Humidity = 68 % soilmoisture=115 % to IBM Watson
Published Temperature = 94 C Humidity = 85 % soilmoisture=81 % to IBM Watson
Published Temperature = 100 C Humidity = 82 % soilmoisture=74 % to IBM Watson
Published Temperature = 97 C Humidity = 65 % soilmoisture=61 % to IBM Watson
Published Temperature = 104 C Humidity = 93 % soilmoisture=85 % to IBM Watson
Published Temperature = 104 C Humidity = 88 % soilmoisture=81 % to IBM Watson
Published Temperature = 101 C Humidity = 97 % soilmoisture=100 % to IBM Watson
Published Temperature = 101 C Humidity = 82 % soilmoisture=78 % to IBM Watson
Published Temperature = 100 C Humidity = 76 % soilmoisture=88 % to IBM Watson
Published Temperature = 103 C Humidity = 66 % soilmoisture=96 % to IBM Watson
Published Temperature = 98 C Humidity = 97 % soilmoisture=71 % to IBM Watson
Published Temperature = 110 C Humidity = 94 % soilmoisture=81 % to IBM Watson
Published Temperature = 104 C Humidity = 68 % soilmoisture=119 % to IBM Watson
Published Temperature = 97 C Humidity = 88 % soilmoisture=79 % to IBM Watson
Published Temperature = 104 C Humidity = 71 % soilmoisture=116 % to IBM Watson
Published Temperature = 98 C Humidity = 84 % soilmoisture=111 % to IBM Watson
Published Temperature = 89 C Humidity = 88 % soilmoisture=75 % to IBM Watson
Published Temperature = 104 C Humidity = 87 % soilmoisture=87 % to IBM Watson
Published Temperature = 96 C Humidity = 92 % soilmoisture=91 % to IBM Watson
Published Temperature = 92 C Humidity = 70 % soilmoisture=79 % to IBM Watson
Published Temperature = 104 C Humidity = 79 % soilmoisture=88 % to IBM Watson
Published Temperature = 96 C Humidity = 87 % soilmoisture=104 % to IBM Watson
Published Temperature = 103 C Humidity = 74 % soilmoisture=88 % to IBM Watson
Published Temperature = 88 C Humidity = 71 % soilmoisture=102 % to IBM Watson
```

**Develop an application with MIT APP  
inventor:Mobile App opening page:**



## Mobile App Log in Page:





## JIRA Software Sprint Planning:



 Smart Farmer - IoT (IoT) Software project

1000

 Springer Springer

500 *Reviews*

Page 10 of 10

**►** **Small business**—A business that is independently owned and operated, is not dominant in its market, and is not part of a chain or franchise.

 Project partners

Results are a robust, meaningful project  
 (continued)

Projector V (Smart Camera) : self-identified, Subject Geometry Applications

## All apartments

  **Completion required**

Downloaded from ascelibrary.org by University of California, San Diego on 06/01/15. Copyright ASCE, for all rights reserved.

© 2006 Blackwell Publishing Ltd *Journal of Internal Medicine* 260: 391–400

© 2004 Blackwell Publishing Ltd  
Journal of Internal Medicine 255: 103–110

[illegible]

**Figure 1** | **Flowchart of the study design.**

Converting applications with  
Visual-Red and further  
application development

**00000000**

 Springer

- Testing developed application
- and working model of hardware

DOI: 10.1002/for

Connect the hardware with RPi  
Cloud and API integration:

100% CLOUTIER, 2007

**A participant's classification and the project**

[illegible]

Copyright © 2004 John Wiley & Sons, Ltd.

Template: MIPA Sprint 1 to 10 X

**Sprint program** 

From	In progress	Not started
100%	0%	0%

### Sprint breakdown

This weight helps you compare planned versus actual completed work. As you can track scope and cost as needed, **scope cost**

Epic products 

These authors do not distinguish between the two types of