

SMART FARMER – IOT ENABLEDD SMART FARMING

APPLICATION

PROJECT DEVELOPMENT – DELIVERY OF

SPRINT - 1

TITLE: SMART FARMER – IOT ENABLEDD SMARTFARMING APPLICATION

TEAM: PNT2022TMID47823

Create 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, "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)
    moist=random.randint(50,120)
    data = { 'temp' : temp, 'Humid': Humid , 'moist':moist}
    #print data
    def myOnPublishCallback():
        print ("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 connected to IoTTF")
        time.sleep(10)

    deviceCli.commandCallback = myCommandCallback

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

```

OUTPUT:



```
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"

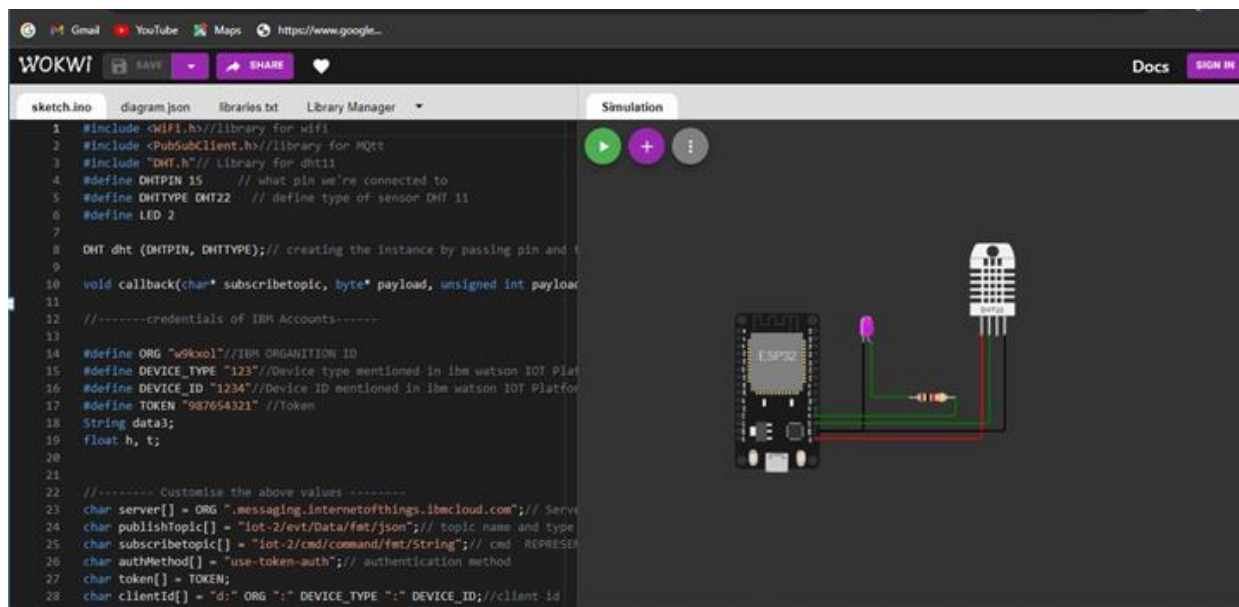
# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['status']
    if status=="Motor On":
        print("Motor is on")
    elif status=="Motor Off":
        print("Motor is off")
    else:
        print("Please send proper command")

try:
    deviceOptions = {"org": organization, "type":
deviceType, "id": deviceId, "auth": {"method": authMethod, "token": authToken}}
    device = ibmiotf.device.Device(deviceOptions)
    app = ibmiotf.application.Application(device)
    app.start(myCommandCallback)
except Exception as e:
    print("Caught exception connecting device: %s" % e)
    sys.exit()
```

Published Temperature = 90 C Humidity = 83 % soilmoisture=76 % to IBM Watson
Published Temperature = 108 C Humidity = 63 % soilmoisture=16 % to IBM Watson
Published Temperature = 110 C Humidity = 61 % soilmoisture=86 % to IBM Watson
Published Temperature = 103 C Humidity = 71 % soilmoisture=83 % to IBM Watson
Command received: Motor On
Motor is on
Published Temperature = 104 C Humidity = 81 % soilmoisture=101 % to IBM Watson
Published Temperature = 109 C Humidity = 74 % soilmoisture=18 % to IBM Watson
Published Temperature = 90 C Humidity = 89 % soilmoisture=53 % to IBM Watson
Published Temperature = 109 C Humidity = 64 % 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=41 % to IBM Watson
Published Temperature = 104 C Humidity = 90 % soilmoisture=85 % to IBM Watson
Published Temperature = 104 C Humidity = 84 % soilmoisture=91 % 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 = 74 % soilmoisture=88 % to IBM Watson
Published Temperature = 103 C Humidity = 64 % soilmoisture=96 % to IBM Watson
Published Temperature = 98 C Humidity = 87 % soilmoisture=73 % 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=78 % to IBM Watson
Published Temperature = 104 C Humidity = 71 % soilmoisture=114 % to IBM Watson
Published Temperature = 98 C Humidity = 84 % soilmoisture=111 % to IBM Watson
Published Temperature = 99 C Humidity = 84 % soilmoisture=71 % to IBM Watson
Published Temperature = 104 C Humidity = 67 % soilmoisture=97 % to IBM Watson
Published Temperature = 96 C Humidity = 90 % soilmoisture=97 % to IBM Watson
Published Temperature = 92 C Humidity = 70 % soilmoisture=76 % to IBM Watson
Published Temperature = 108 C Humidity = 79 % soilmoisture=88 % to IBM Watson
Published Temperature = 96 C Humidity = 87 % soilmoisture=114 % to IBM Watson
Published Temperature = 105 C Humidity = 74 % soilmoisture=88 % to IBM Watson
Published Temperature = 98 C Humidity = 71 % soilmoisture=192 % to IBM Watson

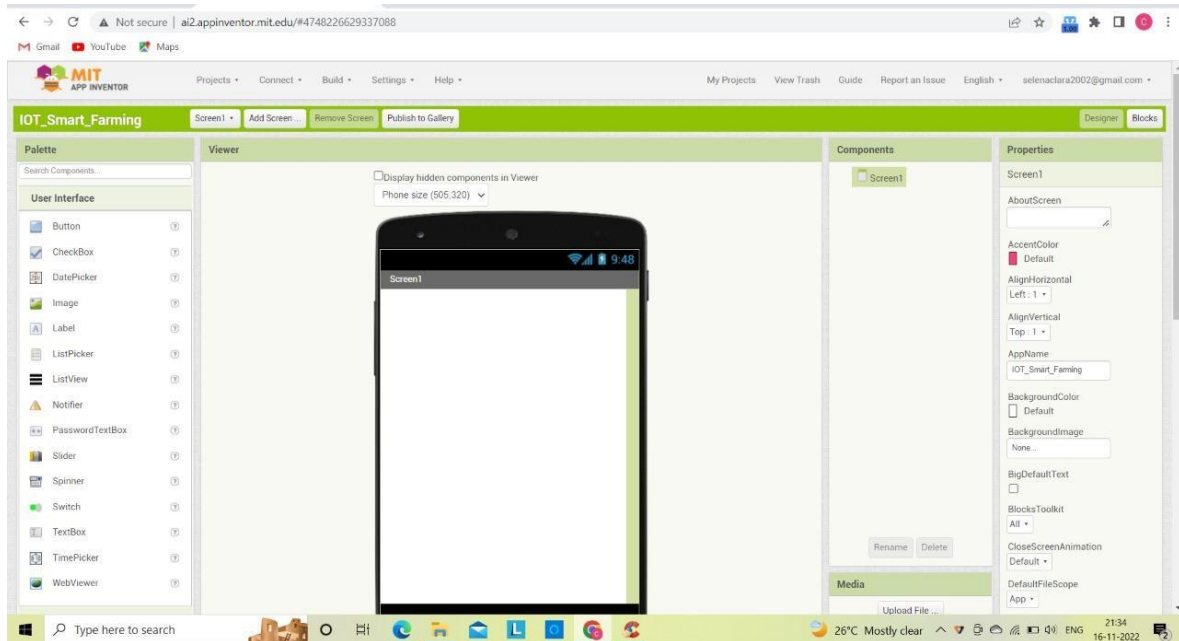
Connect Sensor in ESP8266

CIRCUIT DIAGRAM:

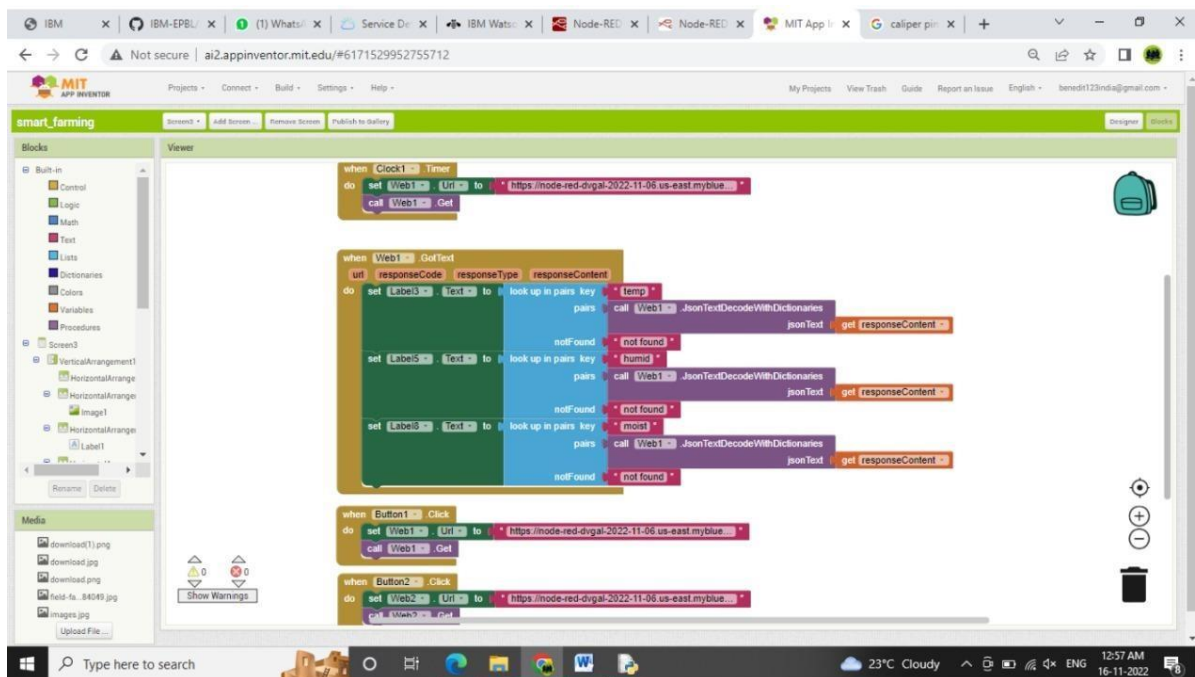
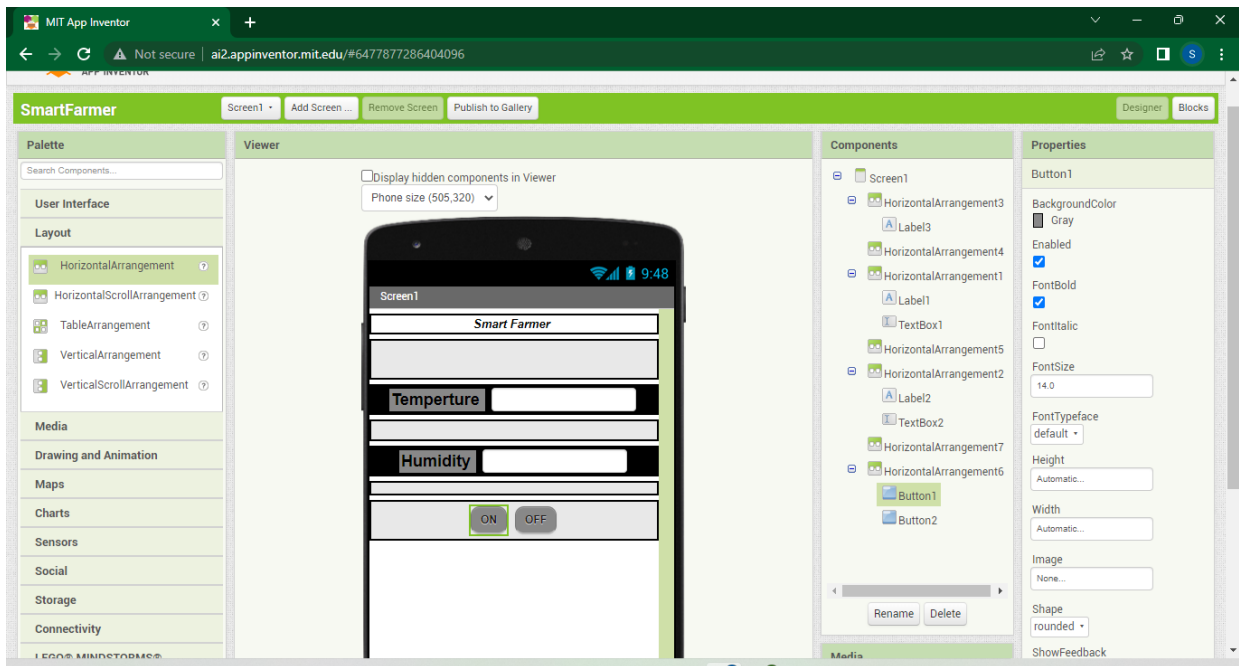


Develop an application with MIT APP inventor:

Mobile App opening page:



Mobile App Log in Page:



JIRA Software Sprint Planning:

