Python code:

```python
import ibmiotf.application
import ibmiotf.device
import random
import json
import time

#Provide your IBM Watson Device Credentials
organization = "2c6fii"
deviceType = "iotdevice"
deviceId = "100"
authMethod = "token"
authToken = "9666870383"

# Initialize the device client.
T=0
H=0

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
```

```python
        if cmd.data['command']=='lighton':
            print("LIGHT ON IS RECEIVED")


        elif cmd.data['command']=='lightoff':
            print("LIGHT OFF IS RECEIVED")

        if cmd.command == "setInterval":
            if 'interval' not in cmd.data:
                print("Error - command is missing
required information: 'interval'")
            else:
                interval = cmd.data['interval']
        elif cmd.command == "print":
            if 'message' not in cmd.data:
                print("Error - command is missing
required information: 'message'")
            else:
                print(cmd.data['message'])

    try:
```

```python
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:
        T=23
        H=45
        #Send Temperature & Humidity to IBM Watson
        data = {"d":{ 'temperature' : T, 'humidity': H }}
        print (data)
```

```python
    def myOnPublishCallback():
        print ("Published Temperature = %s C" % T,
"Humidity = %s %%" % H, "to IBM Watson")

    success = deviceCli.publishEvent("Data", "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()
```

Python code output:

Published Temperature = 23 C Humidity = 45 % to IBM Watson

{'d': {'temperature': 23, 'humidity': 45}}

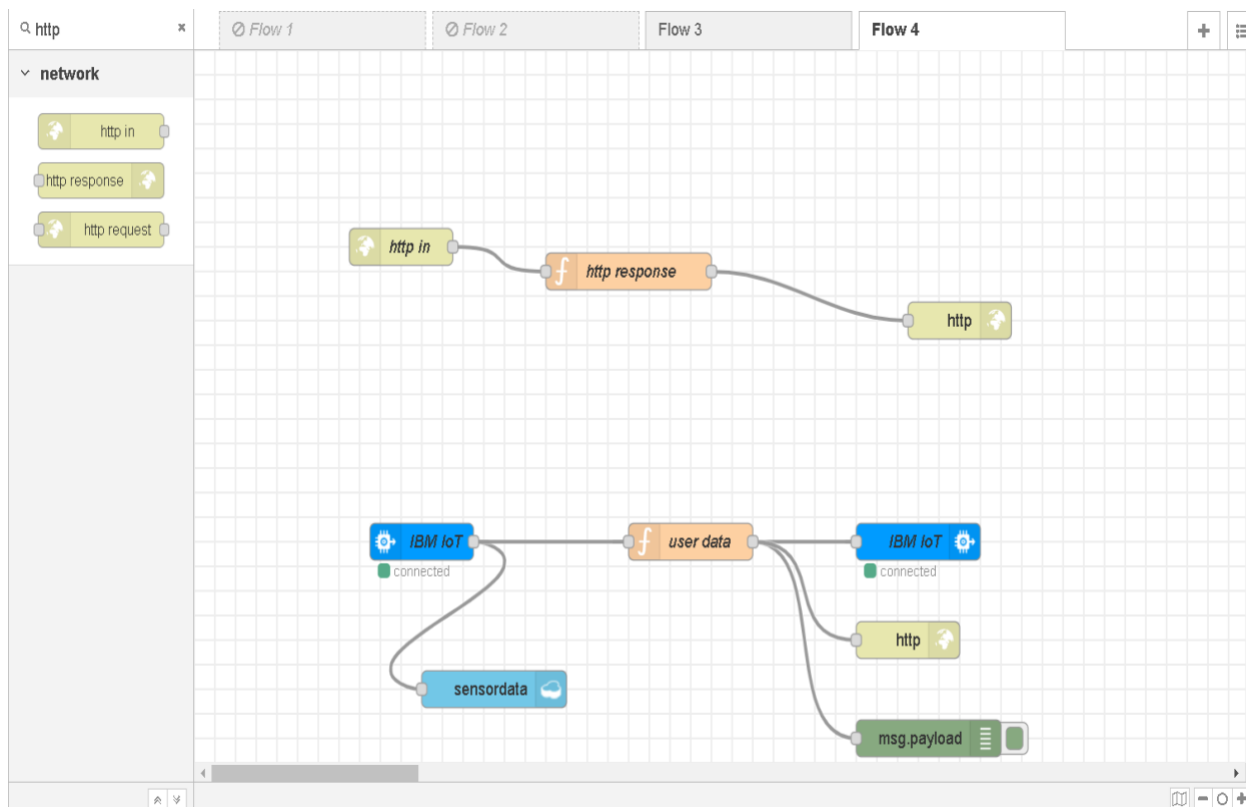Published Temperature = 23 C Humidity = 45 % to IBM Watson

{'d': {'temperature': 23, 'humidity': 45}}

Published Temperature = 23 C Humidity = 45 % to IBM Watson

{'d': {'temperature': 23, 'humidity': 45}}

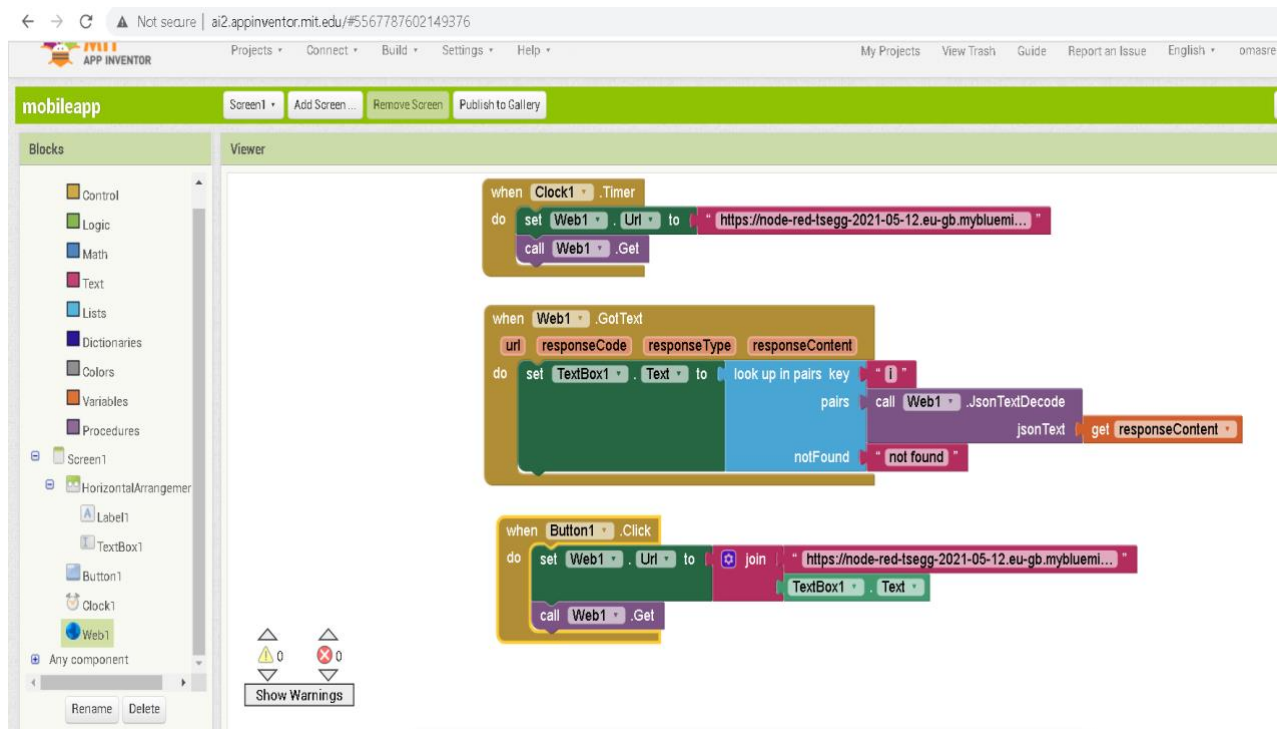Published Temperature = 23 C Humidity = 45 % to IBM Watson

Node red connections:

Mit app inventor:

Mobile app: