

## Assignment-4

Develop a mobile application that takes the user input and sends it to IoT device (python code). print the received data in python shell.

Keep a text box to accept the user input.integrate a submit button. whenever user enters the text input in text box and clicks the button the data should be sent to IBM cloud using URL(HTTP API).

### Code:

```
import ibmiotf.application
```

```
import ibmiotf.device
```

```
import random
```

```
import json
```

```
import time
```

```
#Provide your IBM Watson Device Credentials
```

```
organization = "frtx4v"
```

```
deviceType = "iotdevice"
```

```
deviceId = "1001"
```

```
authMethod = "token"
```

```
authToken = "1234567890"
```

```
# Initialize the device client.
```

T=0

H=0

```
def myCommandCallback(cmd):
```

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

```
    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:
```

```
    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)
```

```
    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 IoT")
```

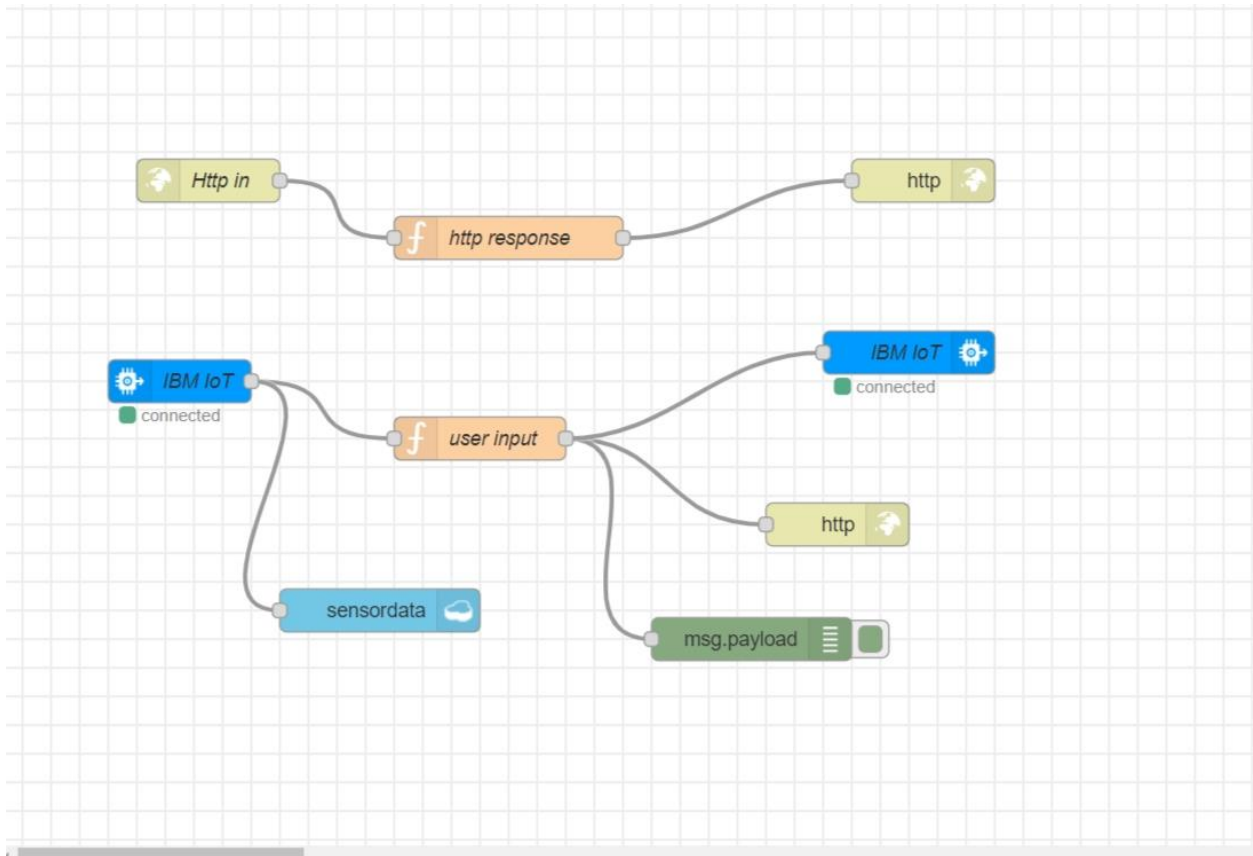
```
    time.sleep(1)
```

```
    deviceCli.commandCallback = myCommandCallback
```

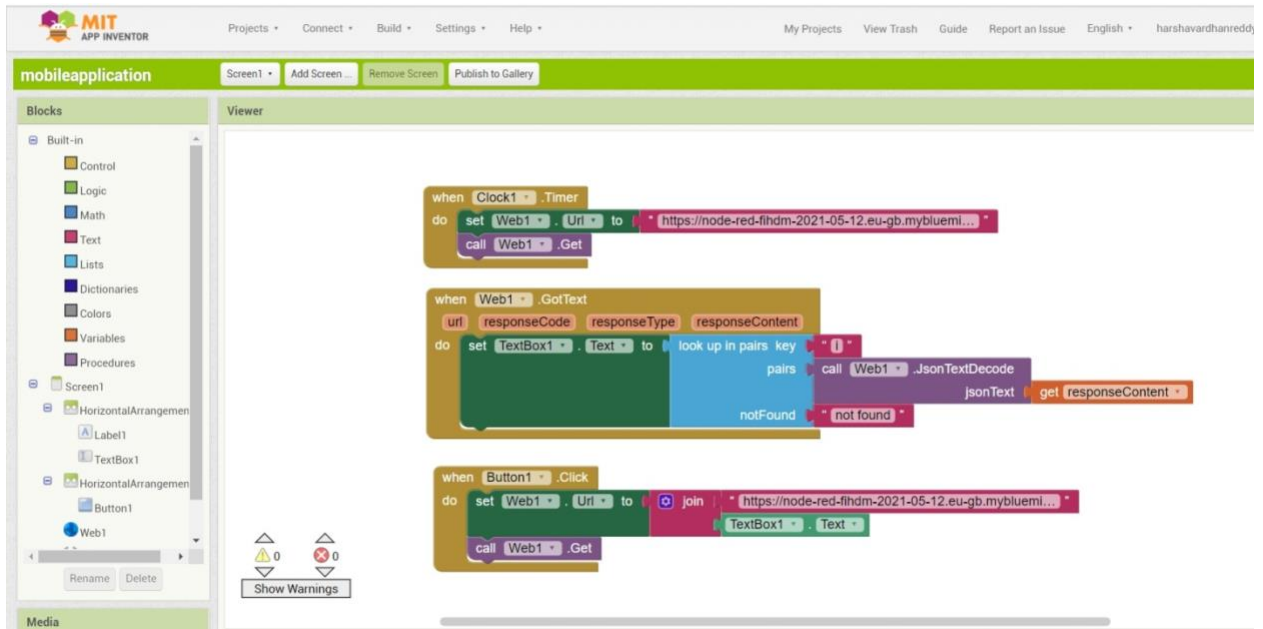
```
# Disconnect the device and application from the cloud
```

```
deviceCli.disconnect()
```

## **Node Red:**



## MIT APP INVENTOR:



# mobile application

Input hihi

Submit

## Python Output:

```
*IDLE Shell 3.9.2*
File Edit Shell Debug Options Window Help
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/admin/AppData/Local/Programs/Python/Python39/mobile app.py =
2021-05-25 17:11:25,368 ibmiotf.device.Client INFO Connected successfully: d:frtx4v:iotdevice:1001
Published Temperature = 23 C Humidity = 45 % to IBM Watson
Published Temperature = 23 C Humidity = 45 % to IBM Watson
Published Temperature = 23 C Humidity = 45 % to IBM Watson
Published Temperature = 23 C Humidity = 45 % to IBM Watson
Published Temperature = 23 C Humidity = 45 % to IBM Watson
Command received: hihi
Published Temperature = 23 C Humidity = 45 % to IBM Watson
```