Assignment – 4

Name – AYUSH KUMAR SINGH

Reg. No. – 19BEI0134

Email – ayushkumar.singh2019@vitstudent.ac.in

Application Id – SPS_APL_20210012580

Aim - 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).
.

Requirement – 1. IBM cloud service
2. MIT app developer
3. Python
4. MIT AI companion app

Working - Here we use the IOT device(python code) to send the home status like room temperature, humidity , fan status , light status to the mobile app.
Based on the data we use 4 buttons to send the data back to the IOT device to change the status of light and fan. Which will also be reflected on the app.
We also use a text box to take some random message input from the user and send it to the IOT device when a button is pressed which is printed on the python shell .

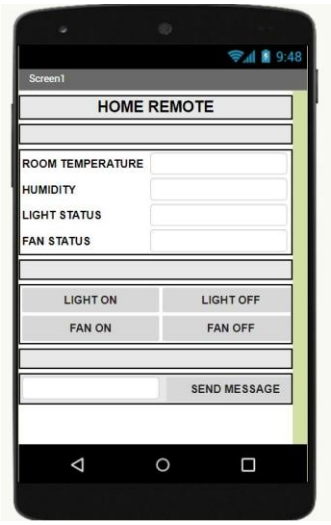
Python Code –

```
import wiotp.sdk.device
import time
import random

myConfig = {
    "identity": {
        "orgId": "d7luey",
        "typeId": "HomeStatus",
        "deviceId":"5683"
    },
    "auth": {
        "token": "12345678"
    }
}

lightsts = "OFF"
fansts = "OFF"
msgs = "HY"

def myCommandCallback(cmd):
    global lightsts
    global fansts
    global msgs
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']
    if(m=="lighton"):
        lightsts = "ON"
    elif(m=="lightoff"):
        lightsts = "OFF"
    elif(m=="fanon"):
        fansts = "ON"
    elif(m=="fanoff"):
        fansts = "OFF"
    else:
        msgs = cmd.data['command']



client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()
```

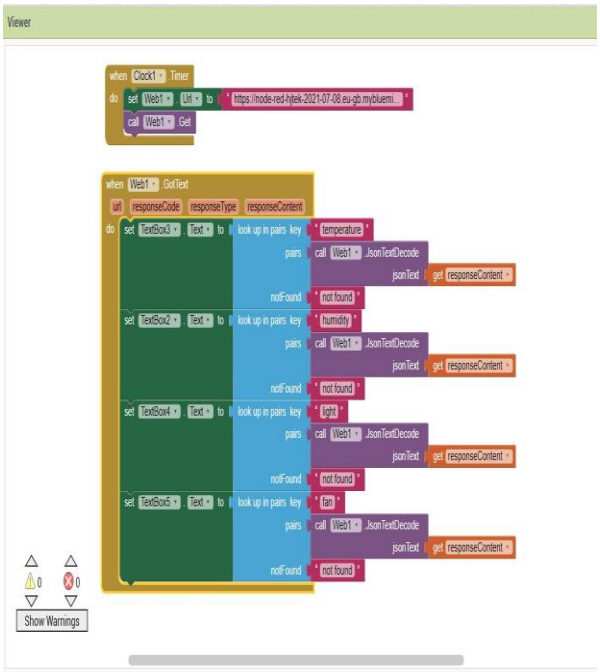
```
while True:
    roomTemp=random.randint(-10,60)
    humidity=random.randint(0,100)
    myData={'temperature':roomTemp, 'humidity':humidity,'light':lightsts,'fan':fansts,'mess':msgs}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)

client.disconnect()
```

Results –

1. Front end

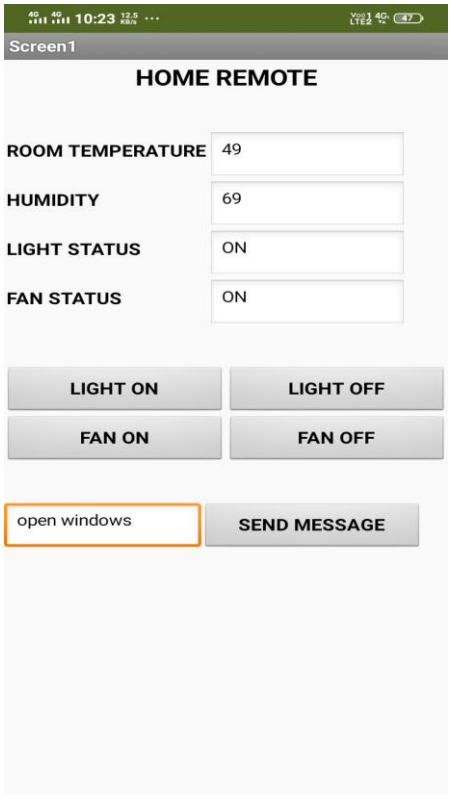


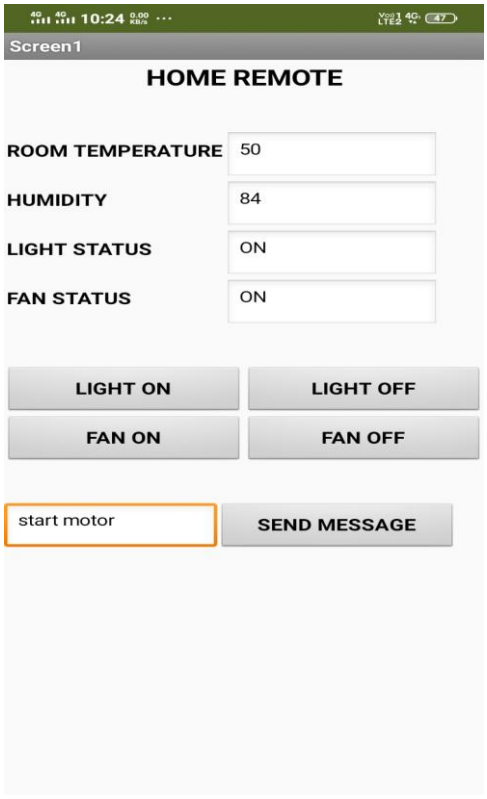
2. Back end



-

3. Mobile outputs





4. Python shell Output