**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).***

***Platform Required:- Node Red Services for connection between IBM cloud and Mobile Application.***

***Python IDE (Latest Version) for execution of code.***

***MIT APP INVENTOR for building of mobile application.***

***Procedure:***

***We use node red for connection between IOT device and IBM Watson cloud. The nodes are connected as shown in node red connections***

***We should run the below python code to get the output in python shell and in node red services.***

***We need to built a mobile app to submit user data using MIT INVENTOR App. Collect all the components required for applications***

***We need to connect all the blocks according to the given block diagram which is shown after the MIT design.***

***At last we need to build the app by scanning QR code using mobile phone.***

***We install the mobile app in our phone and we can verify that the user data is received by tapping submit button.***

***The output is displayed in Python Shell.***

**Python Code:**

***import time***

***import sys***

***import ibmiotf.application***

***import ibmiotf.device***

***import random***

***import json***

***#Provide your IBM Watson Device Credentials***

***organization ="kbfeya"***

***deviceType = "IOTDEVICE"***

***deviceId = "1010"***

***authMethod = "token"***

***authToken = "07\_13\*11&83"***

***# Initialize the device client.***

***T=0***

***H=0***

***O=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=26.85***

***H=78.64***

***O=37.58***

***#Send Temperature,Humidity & Object Temperature to IBM Watson***

***data =jsondata={"d":{ 'temperature': T, 'humidity': H, 'objectTemp': O}}***

***print (data)***

***def myOnPublishCallback():***

***print ("Published Temperature = %s C" % T, "Humidity = %s %%" % H, "Object\_Temperature = %s C" % O, "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***

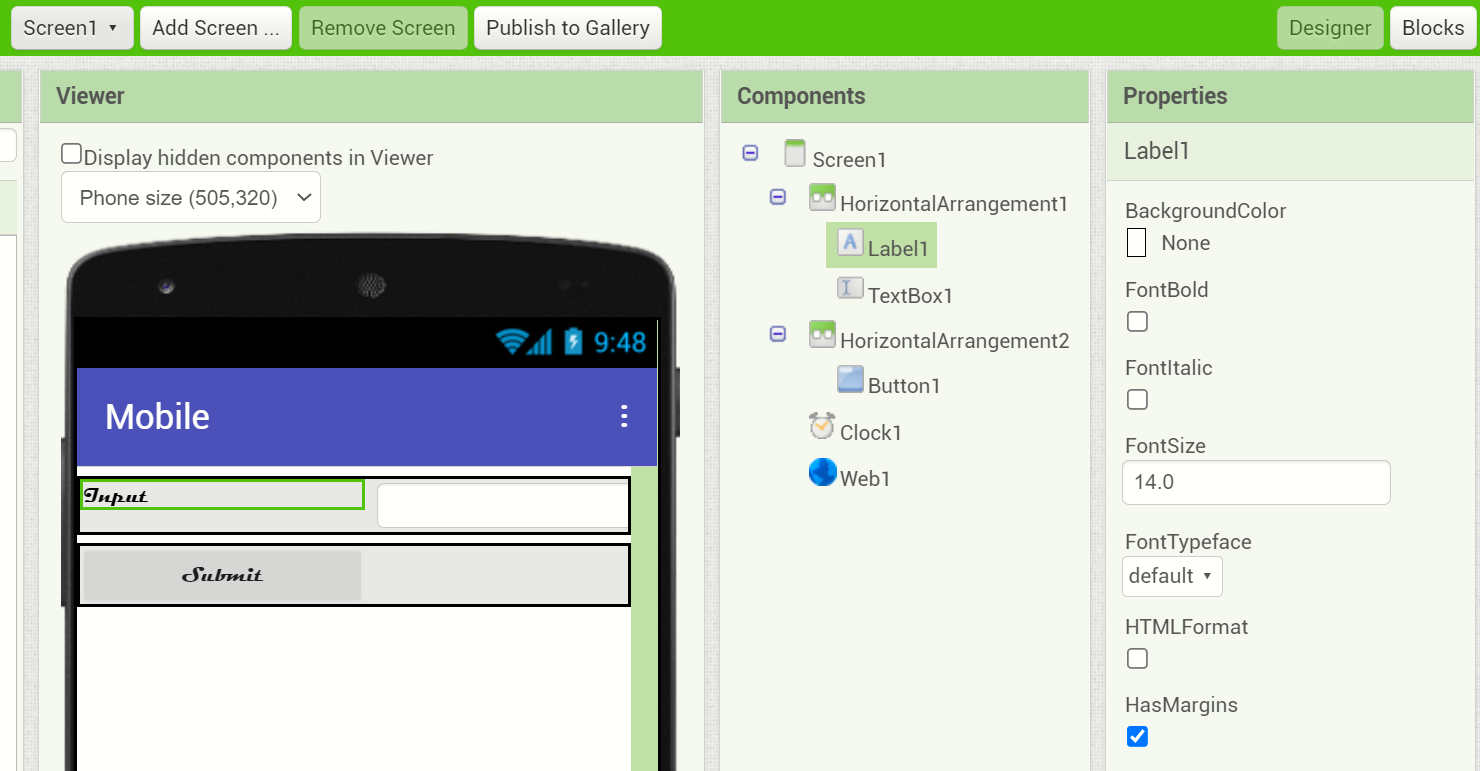
***deviceCli.disconnect()***

**Node Red Connections:**



**MIT APP INVENTOR:**

**Design:**



**Blocks Formation:**

****

**Mobile Application Input**

****

**Python Program Output:**

**'d': {'temperature': 26.85, 'humidity': 78.64, 'objectTemp': 37.58}}**

**{'d': {'temperature': 26.85, 'humidity': 78.64, 'objectTemp': 37.58}}**

**{'d': {'temperature': 26.85, 'humidity': 78.64, 'objectTemp': 37.58}}**

**Command received: hii**

**{'d': {'temperature': 26.85, 'humidity': 78.64, 'objectTemp': 37.58}}**

**Command received: hii**

**Command received: hii**

**{'d': {'temperature': 26.85, 'humidity': 78.64, 'objectTemp': 37.58}}**

**Command received: hii**

**Command received: hii**