**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).

**PYTHON CODE::**

import time

import sys

import ibmiotf.application

import ibmiotf.device

import random

import json

#Provide your IBM Watson Device Credentials

organization = "7icr8c"

deviceType = "iotdevice"

deviceId = "1001"

authMethod = "token"

authToken = "1234567890"

# Initialize the device client.

U=0

def myCommandCallback(cmd):

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

if cmd.data['command']=='Submit':

print("SUBMIT 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:

U=23

#Send Submit to IBM Watson

data = {"d":{ 'userinput' : U,}}

print (data)

def myOnPublishCallback():

print ("Published Userinput = %s C" % U, "to IBM Watson")

success = deviceCli.publishEvent("Data", "json", data, qos=0, on\_publish=myOnPublishCallback)

if not success:

print("Not connected to IoTF")

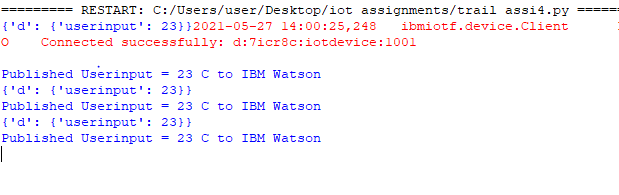
time.sleep(1)

deviceCli.commandCallback = myCommandCallback

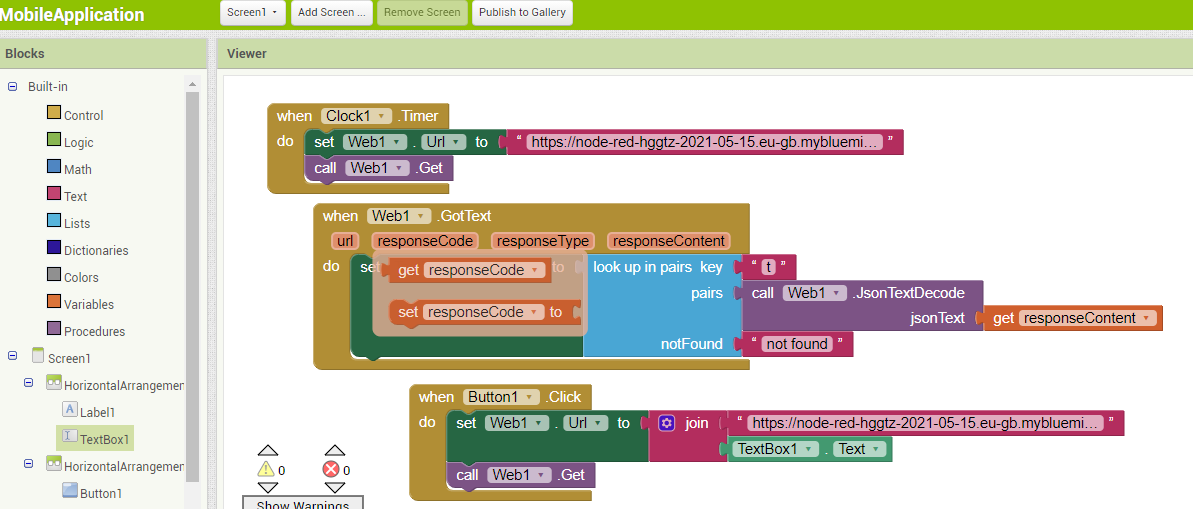
# Disconnect the device and application from the cloud

deviceCli.disconnect()

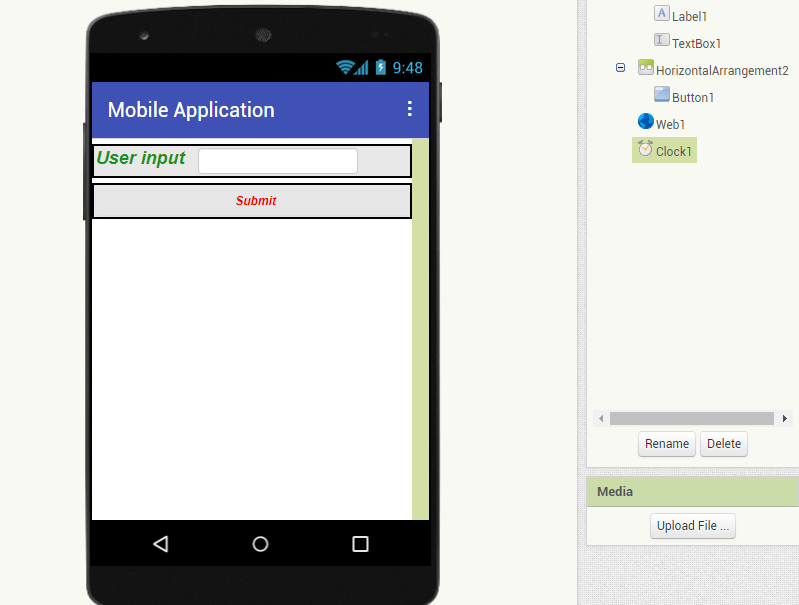
**Python Output:**

****

**MIT APP INVENTOR**::



**MIT UI::**

****

**NODE RED::**

