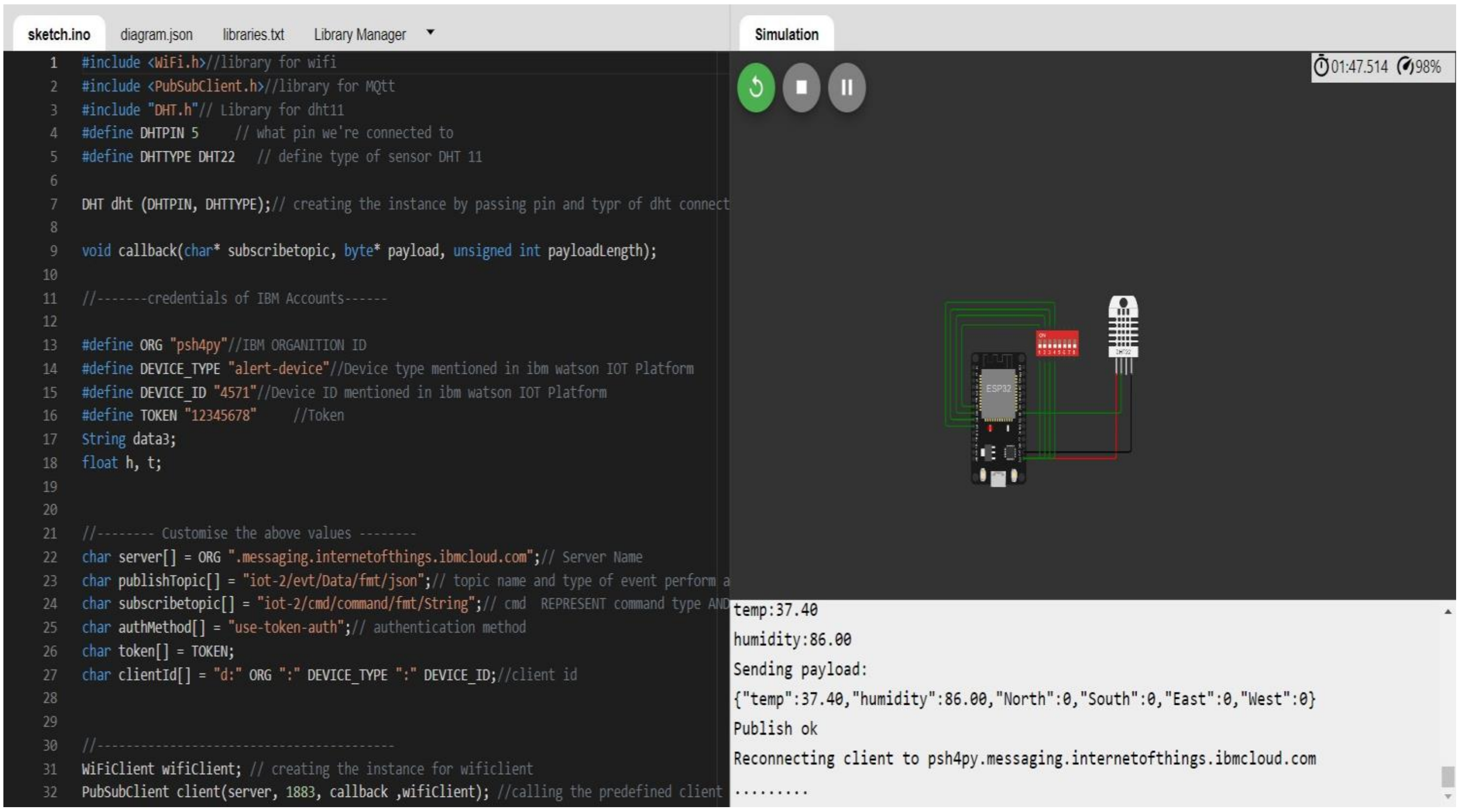
Project Development Phase Sprint-3

Date	17th November 2022
Team ID	PNT2022TMID27330
Project Name	Signs with Smart Connectivity for Better Road Safety.
Marks	20 Marks

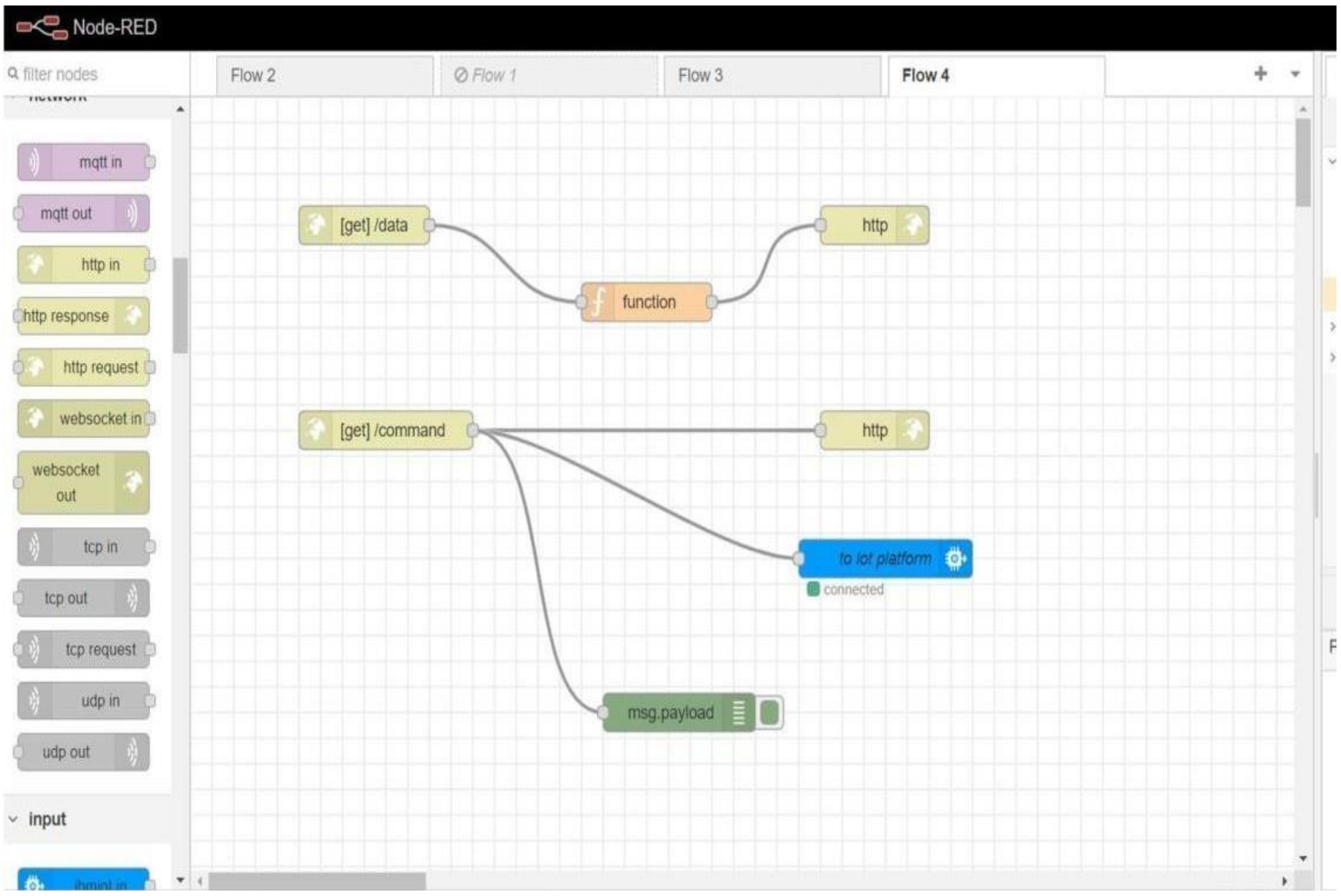
Sprint	Functional Requirement	User Story Number	User Story/Task	Story Points	Priority	Team Members
Sprint-3		US-1	Develop a python script to publish random sensor data such as temperature, humidity, visibility to the IBM IoT platform.	7	Medium	Anupama PH, Naveen Kumar Sai T, Ragini Kumari, Praveen Sharma
Sprint-3		US-2	After developing python code, commands are received print the statements which represent the control of the devices.	5	Low	Anupama PH, Naveen Kumar Sai T, Ragini Kumari, Praveen Sharma
Sprint-3		US-3	Publish Data to the IBM Cloud.	8	High	Anupama PH, Naveen Kumar Sai T, Ragini Kumari, Praveen Sharma



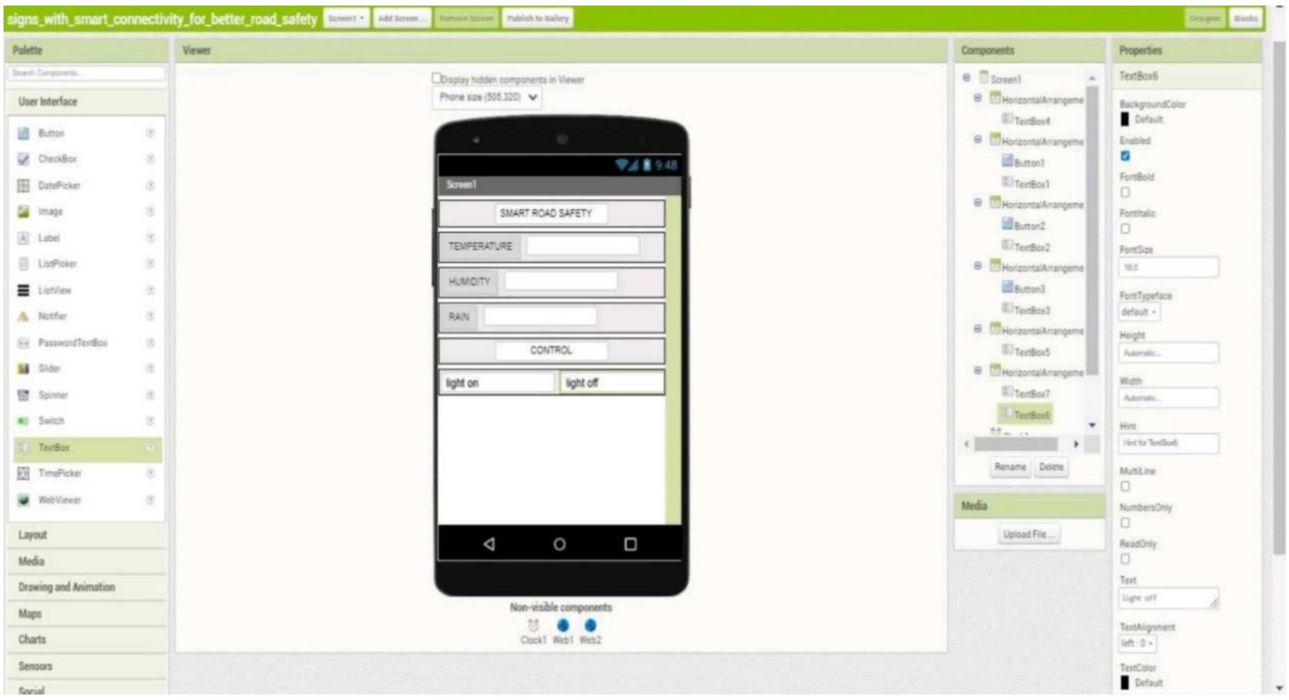
```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include "DHT.h"// Library for dht11
#define DHTPIN 5     // what pin we're connected to
#define DHTTYPE DHT22   // define type of sensor DHT 11

DHT dht (DHTPIN, DHTTYPE);// creating the instance by passing pin and typr of dht connect

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);

//-------credentials of IBM Accounts------

#define ORG "psh4py"//IBM ORGANITION ID
#define DEVICE_TYPE "alert-device"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "4571"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678"     //Token
String data3;
float h, t;


//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform a
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd  REPRESENT command type AND
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id


//-----------------------------------------
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient); //calling the predefined client
```

```
temp:37.40
humidity:86.00
Sending payload:
{"temp":37.40,"humidity":86.00,"North":0,"South":0,"East":0,"West":0}
Publish ok
Reconnecting client to psh4py.messaging.internetofthings.ibmcloud.com
.........
```

Node Red – Connect with MIT app inventor



MIT App Inventor UI Design



US-1 Develop a python script to publish random sensor data such as temperature, humidity and visibility to the IBM IoT Platform

```
import time import
sys
import ibmiotf.application
import ibmiotf.device import
random

#Provide your IBM Watson Device
Credentials organization = "33lnun" deviceType
= "PNT2022TMID47485" deviceId =
"PNT2022TMID47485" authMethod = "token"
authToken = "BGM(9-Tgfy&lrHmglp"


#Intialize GPIO

def myCommandCallback(cmd):
    print("Command received: %s % cmd.data['command']")
status=cmd.data['command']
    if status=="lighton":
print ("led is on")     else :
        print("led is off")
```

```
   #print(cmd)
    try:
   deviceOptions = {"org": organization,"type":
deviceType,"id":deviceId,"authmethod":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:
   #Get Sensor Data from DHT11

   temp=random.randint(0,100)
humid=random.randint(0,100)     visi=random.randint(0,100)

   data = {'temperature'=temp, 'humidity'=humid,'visibility'=visi}
   #print data
   def myOnPublishCallback():
      print("Published temperature=%s C" %temp,"humidity =%s %%"
%humid,"visibility =%s %%" %visi,"to IBM Watson")

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