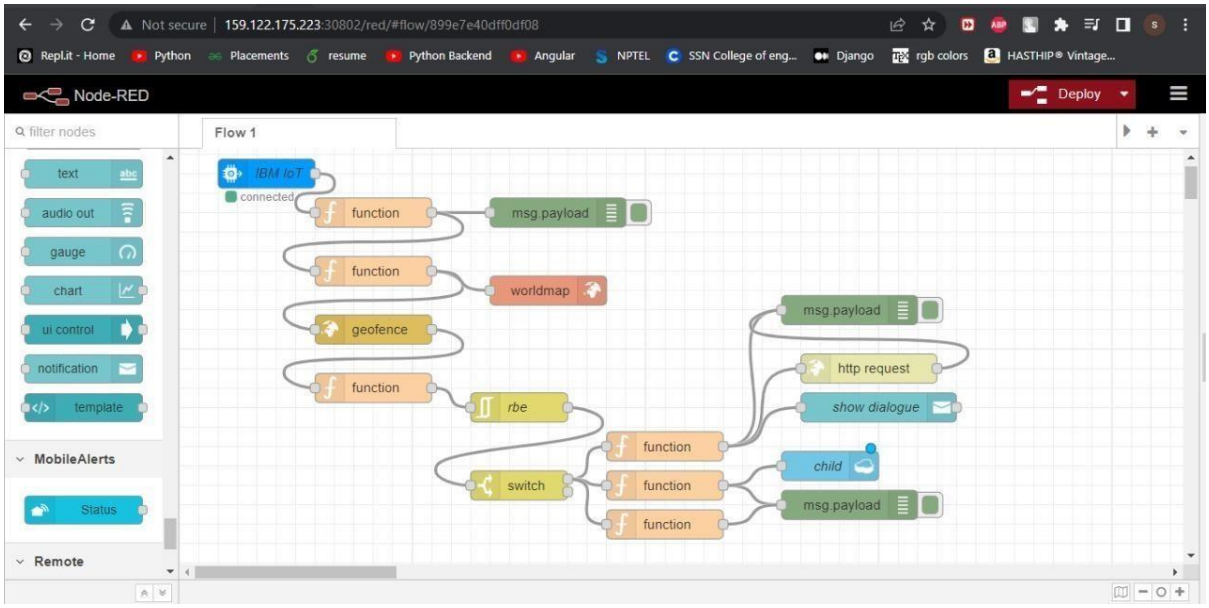
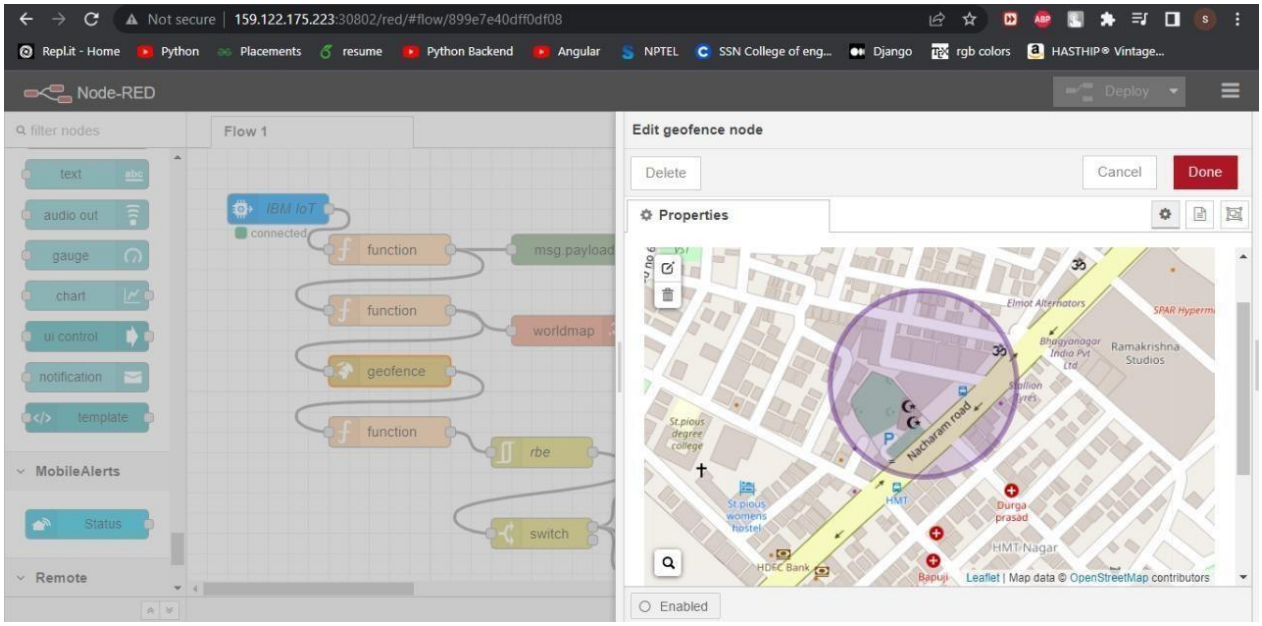
Project Development Phase
Sprint - 2

Date	17 November 2022
Team ID	PNT2022TMID29237
Project Name	IoT Based Safety Gadget for Child Safety Monitoring & Notification

SPRINT 2:

NODERED CONNECTION



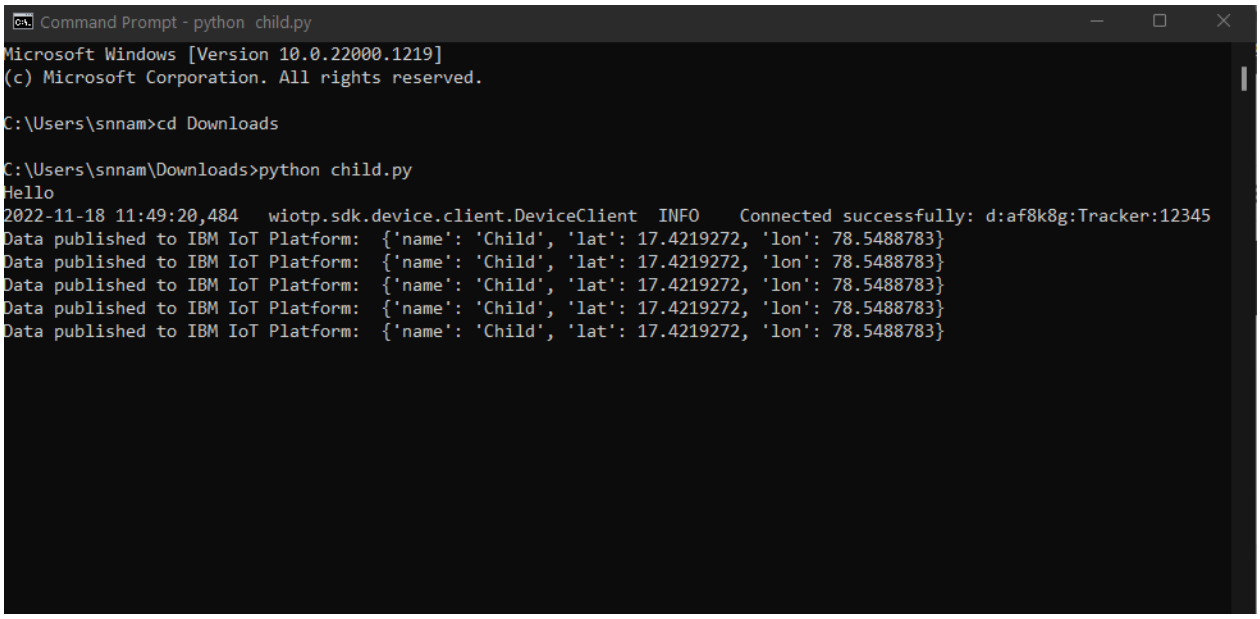


PYTHON FILE FOR CONNECTION TO IBM WATSON IOT PLATFORM:

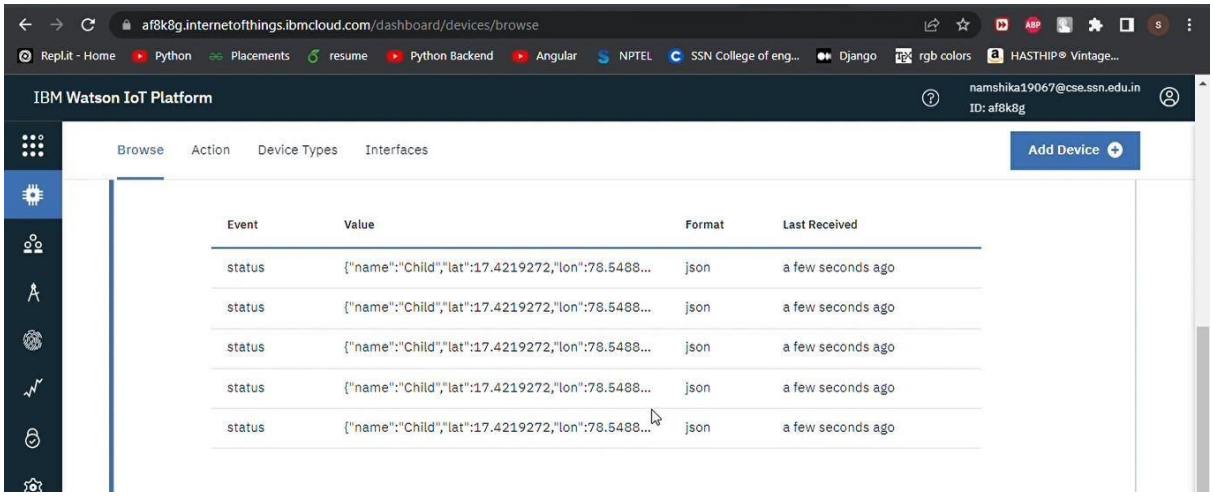
```
import time
import wiotp.sdk.application
print("Hello")
myConfig={
        "identity":{
                "orgId":"af8k8g",
                "typeId":"Tracker",
                "deviceId":"12345",
        },
        "auth":{
                "token":"12345678"
        }
}

client=wiotp.sdk.device.DeviceClient(config=myConfig,logHandlers=None)
client.connect()
while True:
        name="Child"
        latitude=17.4219272
        longitude=78.5488783
        myData={'name':name,'lat':latitude,'lon':longitude}
        client.publishEvent(eventId="status",msgFormat="json",data=myData,qos=0,onPublish=None)
        print("Data published to IBM IoT Platform: ",myData)
        time.sleep(5)
client.disconnect()
```

CONNECTING THE IOT DEVICE:



IBM CLOUD DATABASE:



NOTIFICATION ALERT CODE:

NOTIFICATION and STORE DATA

```
#include<WiFi.h>//library for wifi
#include<PubSubClient.h>//library for
MQTT
void callback(char* subscribe topic, byte* payload, unsigned int payloadlength);
//---------------credentials of IBM Account------- #define ORG "frpi8s"// IBM ORGANIZATION ID
#define DEVICE_TYPE "NodeMCU"//DEVICE TYPE MENTIONEDINIOTWATSON
PLATFORM #define DEVICE_ID "12345"//DEVICE ID MENTIONEDINIOTWATSONPLATEFORM
#define TOKEN "12345678"//Token String data3;float
dist;
//------------customize the above value -------char server []
=ORG".messaging.internetofthings.ibmcloud.com";//servername
char publish topic[]="ultrasonic/evt/Data/fmt/json";/*topic nameandtype
of event perform and format
in which data to be send*/
charsubscribetopic[]="ultrasonic/cmd/test/fmt/String";/*cmd
REPRESENT
Command tupe and
COMMAND IS TEST OF FORMAT STRING*/
char authMethod[]="use-token-auth";//authentication method char
token[]=TOKEN;
char clientid[]="d:" ORG ":" DEVICE_TYPE":" DEVICE_ID;//CLIENT ID//
WiFiClient wifiClient;// creating an instance for wificlient
PubSubClient client(server, 1883 , callback , wifiClient);/*calling thepredefined
client id by passing parameter like server id,portand wificredential*/
int LED =4;
int trig =5; int echo=18; void setup(){
Serial.begin(115200); pinMode(trig,OUTPUT); pinMode(echo,INPUT);
pinMode(LED,OUTPUT); delay(10); Serial.println(); wificonnect();
mqttconnect();
void loop() { digitalWrite(trig,LOW); digitalWrite(trig,HIGH);
delayMicroseconds(10); digitalWrite(trig,LOW);
float dur=pulseIn(echo,HIGH); float dist=(dur * 0.0343)/2;
Serial.print("distance in
cm"); Serial.println(dist); PublishData(dist);
delay(1000);
if (!client.loop()){ mqttconnect();
}
}
```

```
/* ..... retriving to
cloud ..... */
void PublishData(float dist){ mqttconnect();//function call for connectingto ibm
/*creating the string in form of JSON to update the data to ibmcloud*/String
object;if(dist<100)
{
digitalWrite(LED,HIGH); Serial.println("no object is near");
object="Near";
}
else
{
digitalWrite(LED,LOW); Serial.println("no object found"); object="No";
}
String payload="{\"distance\":"; payload +=dist;
payload +="," "\"object\":\""; payload += object;
payload += "\"}";
Serial.print("Sending payload: ");
Serial.println(payload);
if(client.publish(publishtopic, (char*) payload.c_str())){
Serial.println("Publish ok");/* if its sucessfully upload data on the
cloud then it will print publish ok in serial monitor or else it will print
publish
failed*/
} else{
Serial.println("Publish failed");
}
}
void mqttconnect(){ if(!client.connected()){
Serial.print("Reconnecting client to "); Serial.println(server);
while(!!!client.connect(clientid,authMethod, token)){ Serial.print("."); delay(500);
}
initManagedDevice();
Serial.println();
}
}
void wificonnect()//function defenition for wificonnect
{
Serial.println(); Serial.print("Connecting to ");WiFi.begin("vivo 1816", "taetae95",6);//PASSING THE WIFI
CREDIDENTIALS TO
ESTABLISH CONNECTION
while (WiFi.status() !=WL_CONNECTED){ delay(500);
Serial.print(".");
```

```
}
Serial.println(""); Serial.println("WiFi connected"); Serial.println("IPaddress");
Serial.println(WiFi.localIP());
}
void initManagedDevice(){ if(client.subscribe(subscribetopic)){
Serial.println((subscribetopic)); Serial.println("subscribe to cmd OK");
}else{
Serial.println("subscribe to cmd failed");
}
}
void callback(char* subscribetopic,byte*payload,unsigned int
payloadLength)
{
Serial.print("callback invoked for topic: ");
Serial.println(subscribetopic); for(int i=0; i< payloadLength; i++){
//Serial.print((char)payload[i]); data3 +=(char)payload[i];
}
//Serial.println("dta: "+ data3);
//if(data3=="Near")
//{
//Serial.println(data3);
//digitalWrite(LED,HIGH);
//}
//else //{
//Serial.println(data3);
//digitalWrite(LED,LOW);//} data3="";
}
```