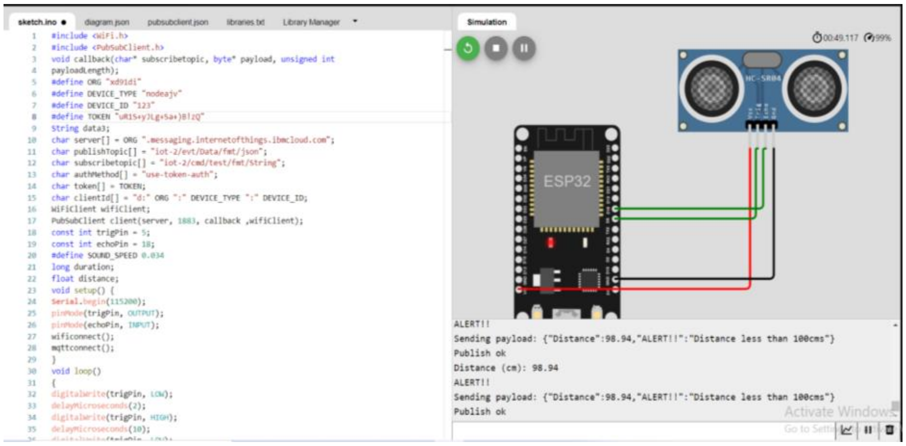
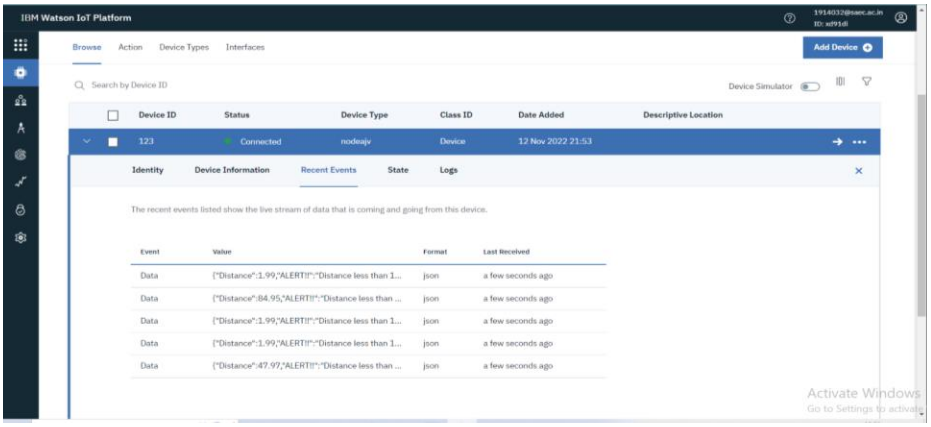
SPRINT -2



```
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);
#define ORG "xd91di"
#define DEVICE_TYPE "nodeajv"
#define DEVICE_ID "123"
#define TOKEN "uR15+yJLg+5a+)B!zQ"
String data3;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char subscribetopic[] = "iot-2/cmd/test/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
WiFiClient wifiClient;
PubSubClient client(server, 1883, callback ,wifiClient);
const int trigPin = 5;
const int echoPin = 18;
#define SOUND_SPEED 0.034
long duration;
float distance;
void setup() {
Serial.begin(115200);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
wificonnect();
mqttconnect();
}
void loop()
{
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
```

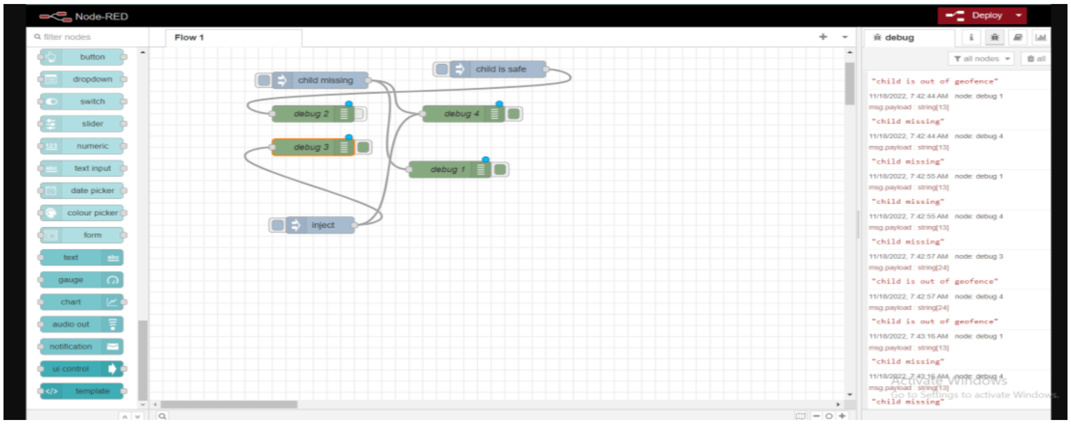
```
ALERT!!
Sending payload: {"Distance":98.94,"ALERT!!":"Distance less than 100cms"}
Publish ok
Distance (cm): 98.94
ALERT!!
Sending payload: {"Distance":98.94,"ALERT!!":"Distance less than 100cms"}
Publish ok
```



	Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
□	123	● Connected	nodeajv	Device	12 Nov 2022 21:53	

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Data	{"Distance":1.99,"ALERT!!":"Distance less than 1...	json	a few seconds ago
Data	{"Distance":84.95,"ALERT!!":"Distance less than ...	json	a few seconds ago
Data	{"Distance":1.99,"ALERT!!":"Distance less than 1...	json	a few seconds ago
Data	{"Distance":1.99,"ALERT!!":"Distance less than 1...	json	a few seconds ago
Data	{"Distance":47.97,"ALERT!!":"Distance less than ...	json	a few seconds ago



```
"child is out of geofence"
11/18/2022, 7:42:44 AM   node: debug 1
msg.payload : string[13]
"child missing"
11/18/2022, 7:42:44 AM   node: debug 4
msg.payload : string[13]
"child missing"
11/18/2022, 7:42:55 AM   node: debug 1
msg.payload : string[13]
"child missing"
11/18/2022, 7:42:55 AM   node: debug 4
msg.payload : string[13]
"child missing"
11/18/2022, 7:42:57 AM   node: debug 3
msg.payload : string[24]
"child is out of geofence"
11/18/2022, 7:42:57 AM   node: debug 4
msg.payload : string[24]
"child is out of geofence"
11/18/2022, 7:43:16 AM   node: debug 1
msg.payload : string[13]
"child missing"
11/18/2022, 7:43:16 AM   node: debug 4
msg.payload : string[13]
"child missing"
```

MAIN LOGIN CODE

```
<!DOCTYPE html>

<html> <head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<title> Login Page </title>

<style>

Body {

font-family: Calibri, Helvetica, sans-serif;

background-color: #9FE2BF;

}

button {

background-color: #9FE2BF;
```

Team ID PNT2022TMID27063

Date 5 November 2022

Project Title IoT Based Safety

Gadget for Child

Safety Monitoring

and Notification

```
width: 100%;
color: black;
padding: 15px;
margin: 10px 0px;
border: none;
cursor: pointer;
}
form {
border: 3px solid #f1f1f1;
}
input[type=text], input[type=password] {
width: 100%;
margin: 8px 0;
padding: 12px 20px;
display: inline-block;
border: 2px white;
box-sizing: border-box;
}
button:hover {
opacity: 0.7;
}
.cancelbtn {
width: auto;
```

```
padding: 10px 18px;
margin: 10px 5px;
}
.container {
padding: 25px;
background-color: #CCCCFF;
}
</style> </head>
<body>
<center> <h1> Login Form </h1> </center>
<form>
<div class="container">
<label>Device ID/Number: </label>
<input type="password" placeholder="Enter Password" name="password" required>
<label>E-Mail : </label>
<input type="text" placeholder="Enter Username" name="username" required>
<label>Password : </label>
<input type="password" placeholder="Enter Password" name="password" required>
<button type="submit">Login</button>
<button class="loginBtn loginBtn--facebook">Login with Facebook.</button>
<button class="loginBtn loginBtn--google">Login with Google.</button>
<input type="checkbox" checked="checked"> Remember me
<button type="button" class="cancelbtn"> Cancel</button>
Forgot <a href="#"> password? </a>
</div>
```

```
</form>

</body>

</html>
```

NOTIFICATION CODE

```
#include <WiFi.h>

#include <PubSubClient.h>

void callback(char* subscribetopic, byte* payload, unsigned int

payloadLength);

#define ORG "xd91di"

#define DEVICE_TYPE "nodeajv"

#define DEVICE_ID "123"

#define TOKEN "uR1S+yJLg+Sa+)B!zQ"

String data3;

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";

char publishTopic[] = "iot-2/evt/Data/fmt/json";

char subscribetopic[] = "iot-2/cmd/test/fmt/String";

char authMethod[] = "use-token-auth";
char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;

WiFiClient wifiClient;

PubSubClient client(server, 1883, callback ,wifiClient);

const int trigPin = 5;

const int echoPin = 18;

#define SOUND_SPEED 0.034

long duration;

float distance;

void setup() {

Serial.begin(115200);

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

wificonnect();

mqttconnect();

}

void loop()

{

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance = duration * SOUND_SPEED/2;
```

```
Serial.print("Distance (cm): ");

Serial.println(distance);

if(distance<100)

{

Serial.println("ALERT!!");

delay(1000);

PublishData(distance);

delay(1000);

if (!client.loop()) {

mqttconnect();

}

}

delay(1000);

}

void PublishData(float dist) {

mqttconnect();

String payload = "{\"Distance\":";

payload += dist;

payload += ",\"ALERT!!\":""\"Distance less than 100cms\"";

payload += "}";

Serial.print("Sending payload: ");

Serial.println(payload);


if (client.publish(publishTopic, (char*) payload.c_str())) {

Serial.println("Publish ok");
} 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()

{

Serial.println();

Serial.print("Connecting to ");

WiFi.begin("Wokwi-GUEST", "", 6);

while (WiFi.status() != WL_CONNECTED) {

delay(500);

Serial.print(".");

}
```

```
Serial.println("");

Serial.println("WiFi connected");

Serial.println("IP address: ");

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++) {

data3 += (char)payload[i];

}

Serial.println("data: "+ data3);

data3="";

}
```