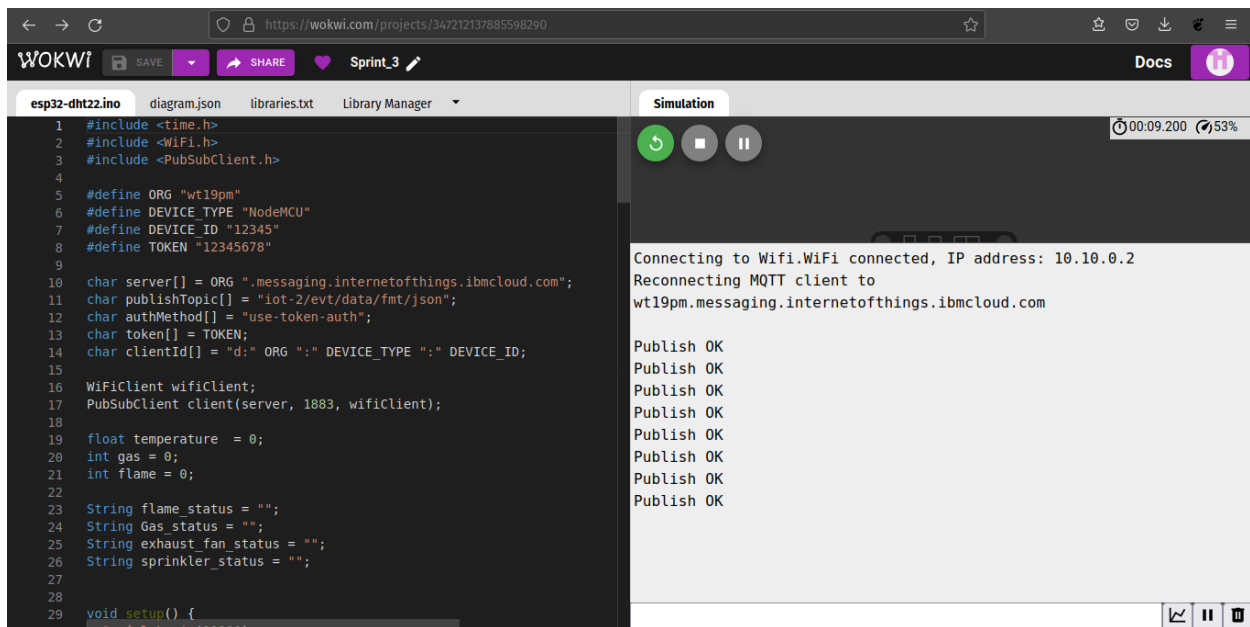


Sprint 3

WOKWI SIMULATOR LINK : <https://wokwi.com/projects/347212137885598290>

WEB PAGE LINK : <https://node-red-dashboard059.eu-gb.mybluemix.net/fire>

Results:



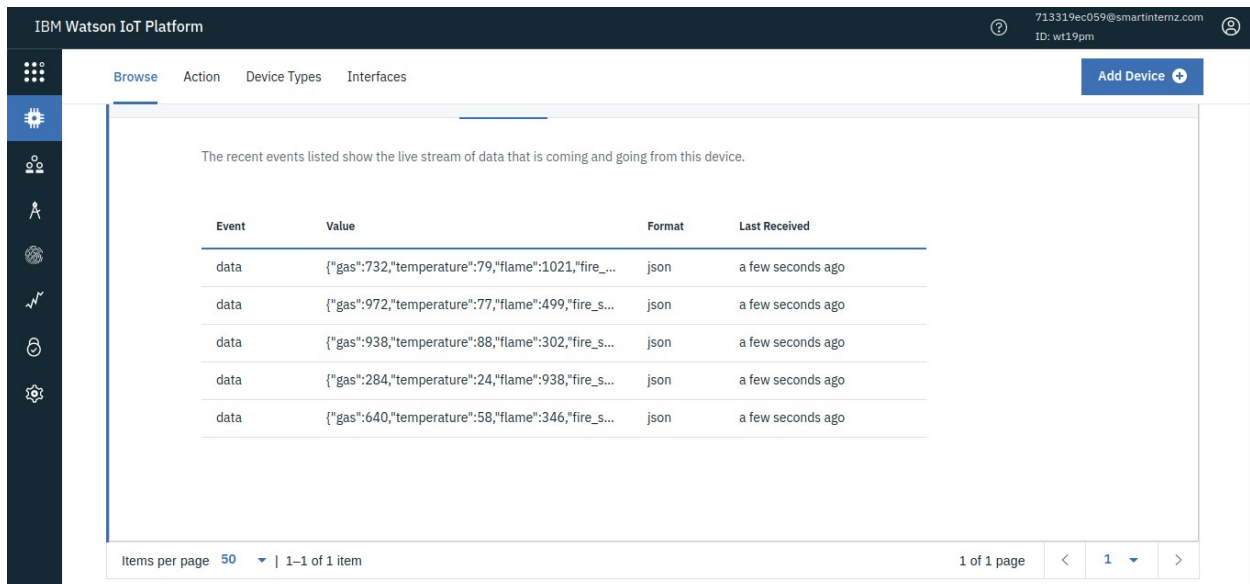
The screenshot shows the Wokwi simulator interface. On the left, a code editor displays a C++ program for an ESP32. The code includes headers for time, WiFi, and PubSubClient, defines constants for the organization, device type, ID, and token, and sets up an MQTT client. The main loop publishes data to an MQTT topic. On the right, the simulation console shows the output of the program, including connection status and successful publish messages.

```
1 #include <time.h>
2 #include <WiFi.h>
3 #include <PubSubClient.h>
4
5 #define ORG "wt19pm"
6 #define DEVICE_TYPE "NodeMCU"
7 #define DEVICE_ID "12345"
8 #define TOKEN "12345678"
9
10 char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
11 char publishTopic[] = "iot-2/evt/data/fmt/json";
12 char authMethod[] = "use-token-auth";
13 char token[] = TOKEN;
14 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
15
16 WiFiClient wifiClient;
17 PubSubClient client(server, 1883, wifiClient);
18
19 float temperature = 0;
20 int gas = 0;
21 int flame = 0;
22
23 String flame_status = "";
24 String Gas_status = "";
25 String exhaust_fan_status = "";
26 String sprinkler_status = "";
27
28
29 void setup() {
30   Serial.begin(115200);
31 }
```

Simulation console output:

```
Connecting to Wifi.WiFi connected, IP address: 10.10.0.2
Reconnecting MQTT client to
wt19pm.messaging.internetofthings.ibmcloud.com

Publish OK
Publish OK
Publish OK
Publish OK
Publish OK
Publish OK
Publish OK
Publish OK
```

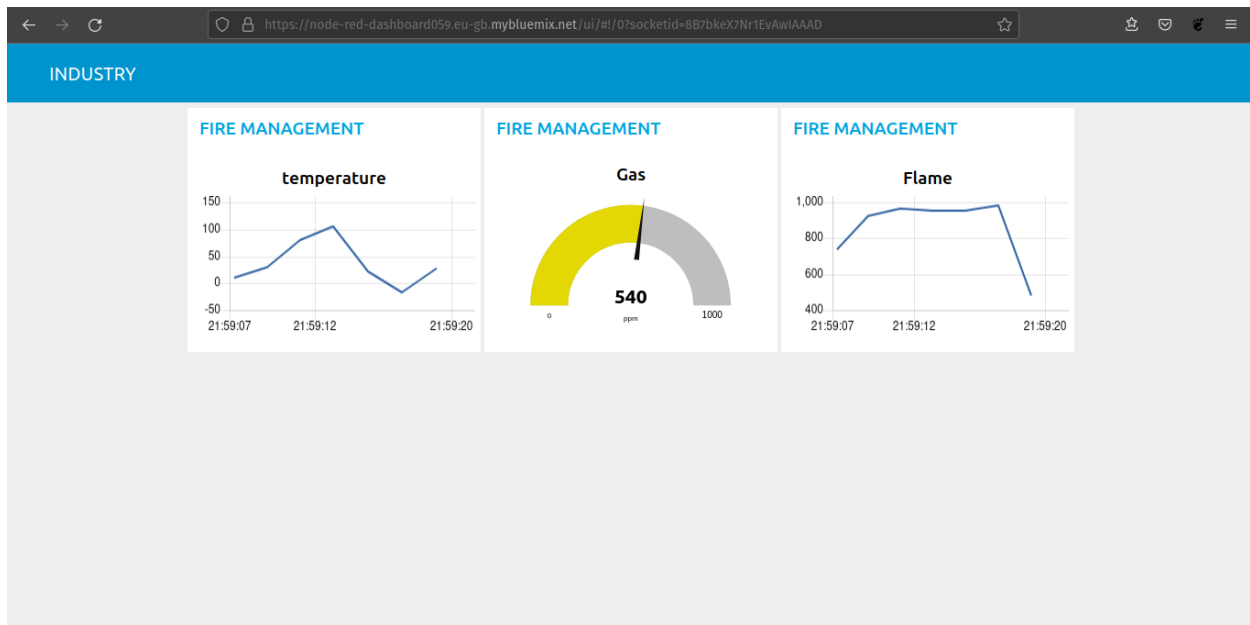
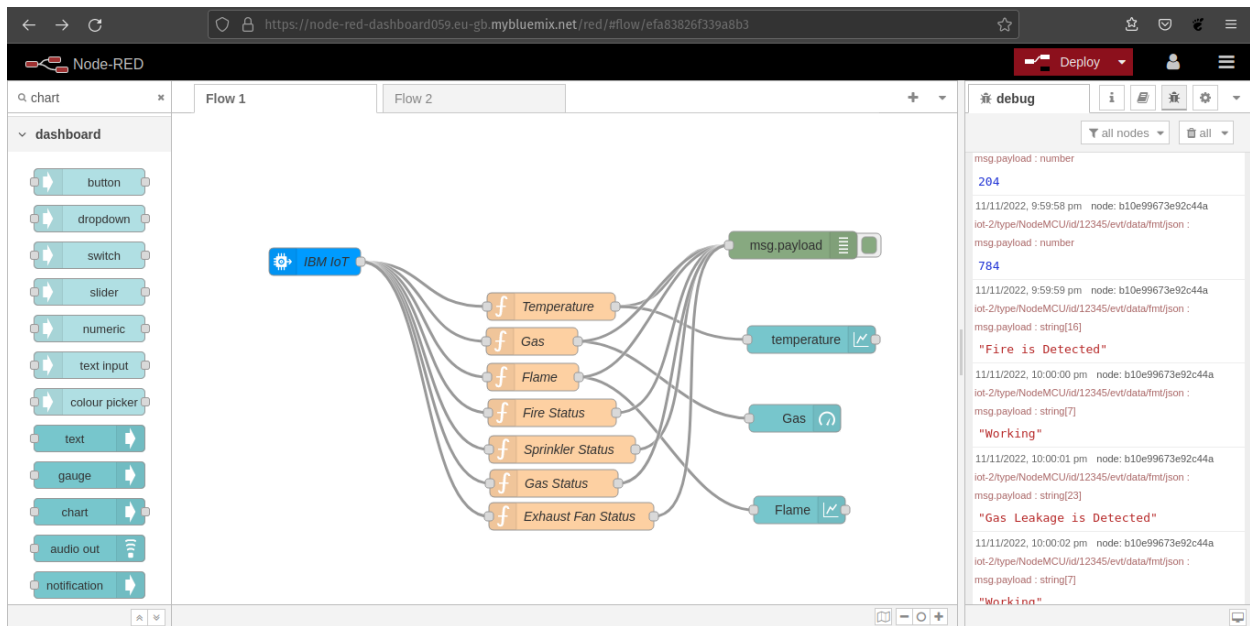


The screenshot shows the IBM Watson IoT Platform dashboard. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains icons for various IoT functions. The main content area displays a table of recent events, which are live stream data from a device. The table has four columns: Event, Value, Format, and Last Received. The events are listed in descending order of time, with the most recent at the top. The table is currently showing 5 items per page, and there is 1 item in total.

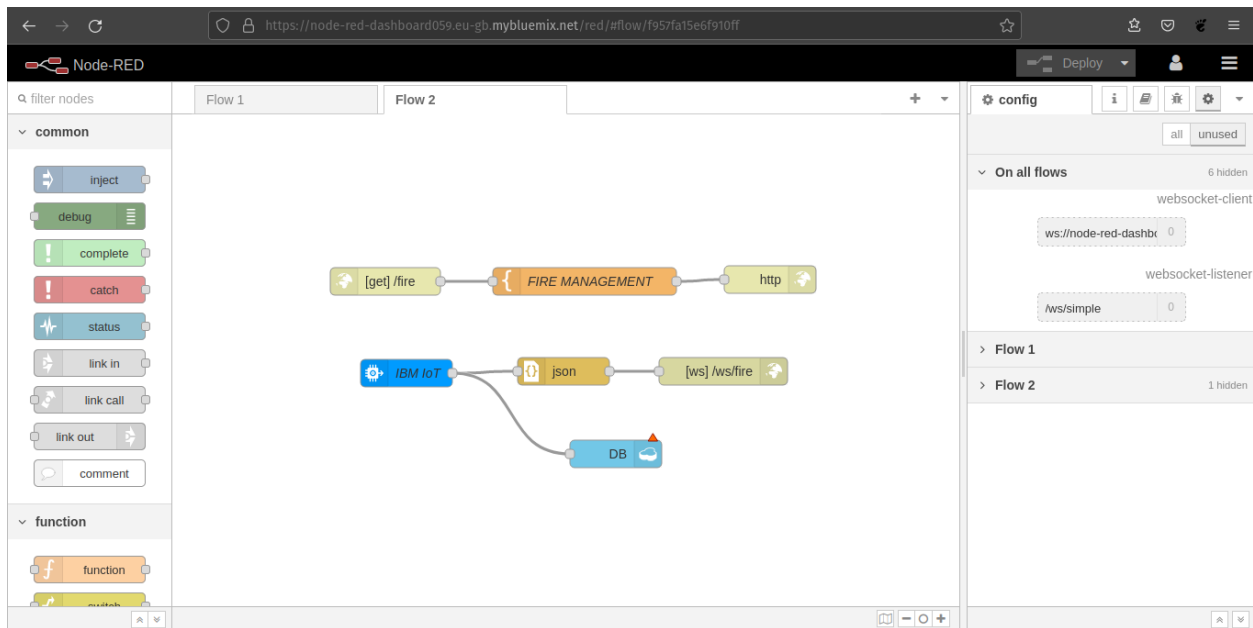
Event	Value	Format	Last Received
data	{"gas":732,"temperature":79,"flame":1021,"fire_...	json	a few seconds ago
data	{"gas":972,"temperature":77,"flame":499,"fire_s...	json	a few seconds ago
data	{"gas":938,"temperature":88,"flame":302,"fire_s...	json	a few seconds ago
data	{"gas":284,"temperature":24,"flame":938,"fire_s...	json	a few seconds ago
data	{"gas":640,"temperature":58,"flame":346,"fire_s...	json	a few seconds ago

Items per page 50 | 1-1 of 1 item

1 of 1 page

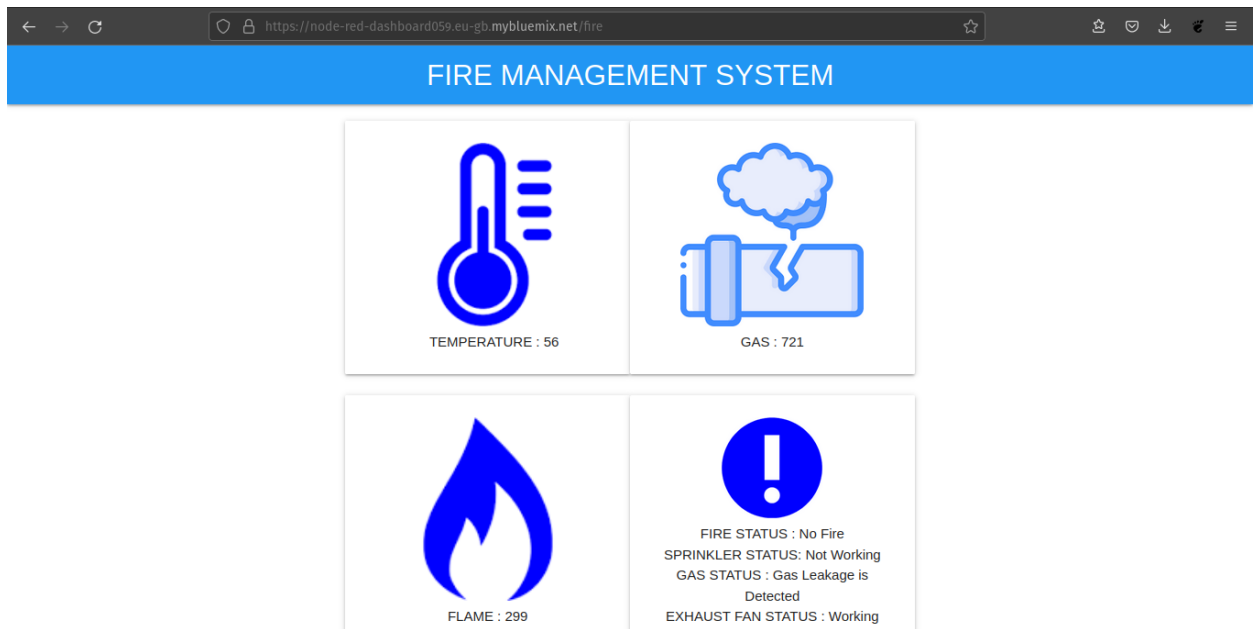


TRANSFERRING DATA FROM NODE-RED INTO WEB PAGE USING WEB SOCKET:



WEB UI:

DESKTOP VIEW



MOBILE VIEW:

8:30 PM

VoLTE 4G 44



.eu-gb.mybluemix.net



29



FIRE MANAGEMENT SYSTEM



TEMPERATURE : 121



GAS : 3



FLAME : 650



FIRE STATUS : Fire is Detected
SPRINKLER STATUS: Working
GAS STATUS : No Gas Leakage is Detected
EXHAUST FAN STATUS : Not Working

CLOUDANT:

The screenshot shows the Cloudant dashboard interface. The left sidebar contains navigation icons for database management, query, permissions, changes, and design documents. The main area displays a list of documents in a table format. The table has columns for 'id', 'key', and 'value'. The 'id' column shows document IDs, the 'key' column shows keys, and the 'value' column shows JSON objects. The table is paginated, showing documents 1 to 20. The 'value' column shows JSON objects with keys like 'rev', 'gas', 'temperature', 'flame', 'fire_status', 'sprinkler_status', 'gas_status', and 'exhaust_fan_status'.

id	key	value
657846f21e0cb8ead462fd89321d28...	657846f21e0cb8ead462fd89321d28...	{ "rev": "1-1c9683229f242d4133b7f...
657846f21e0cb8ead462fd89321dd3...	657846f21e0cb8ead462fd89321dd3...	{ "rev": "1-8aeee9d453a632f539ee9c...
657846f21e0cb8ead462fd8932201e...	657846f21e0cb8ead462fd8932201e...	{ "rev": "1-7b6df30912cf9fde43ca8b...
657846f21e0cb8ead462fd8932203d...	657846f21e0cb8ead462fd8932203d...	{ "rev": "1-a9bec25d7f94ccc71ce692...
70ea2e4bb2a9c635be3ce2603a25a...	70ea2e4bb2a9c635be3ce2603a25a...	{ "rev": "1-b567b4cce122c31e1666fc...
70ea2e4bb2a9c635be3ce2603a268...	70ea2e4bb2a9c635be3ce2603a268...	{ "rev": "1-217497b95c16c3d228800...
70ea2e4bb2a9c635be3ce2603a272...	70ea2e4bb2a9c635be3ce2603a272...	{ "rev": "1-a01738b27517a2bb4b93b...
70ea2e4bb2a9c635be3ce2603a273...	70ea2e4bb2a9c635be3ce2603a273...	{ "rev": "1-13230a9f364a021a02422...
7170def319e06e12e85b74c728897...	7170def319e06e12e85b74c728897...	{ "rev": "1-4bdfcbf4dbb888784fc24d...
7170def319e06e12e85b74c7288b7...	7170def319e06e12e85b74c7288b7...	{ "rev": "1-5b1a46d23a6c259bd5b97...
7170def319e06e12e85b74c7288c7...	7170def319e06e12e85b74c7288c7...	{ "rev": "1-782ab5b4a08aaf22641a1...

The screenshot shows the Cloudant dashboard interface with a document selected. The document ID is '657846f21e0cb8ead462fd89321d28fd'. The document content is displayed in a code editor. The document is a JSON object with keys like 'id', 'rev', 'gas', 'temperature', 'flame', 'fire_status', 'sprinkler_status', 'gas_status', and 'exhaust_fan_status'. The document is currently in a 'Save Changes' state.

```
1 {
2   "_id": "657846f21e0cb8ead462fd89321d28fd",
3   "_rev": "1-1c9683229f242d4133b7fae068107c43",
4   "gas": 267,
5   "temperature": 50,
6   "flame": 931,
7   "fire_status": "Fire is Detected",
8   "sprinkler_status": "Working",
9   "gas_status": "Gas Leakage is Detected",
10  "exhaust_fan_status": "Working"
11 }
```