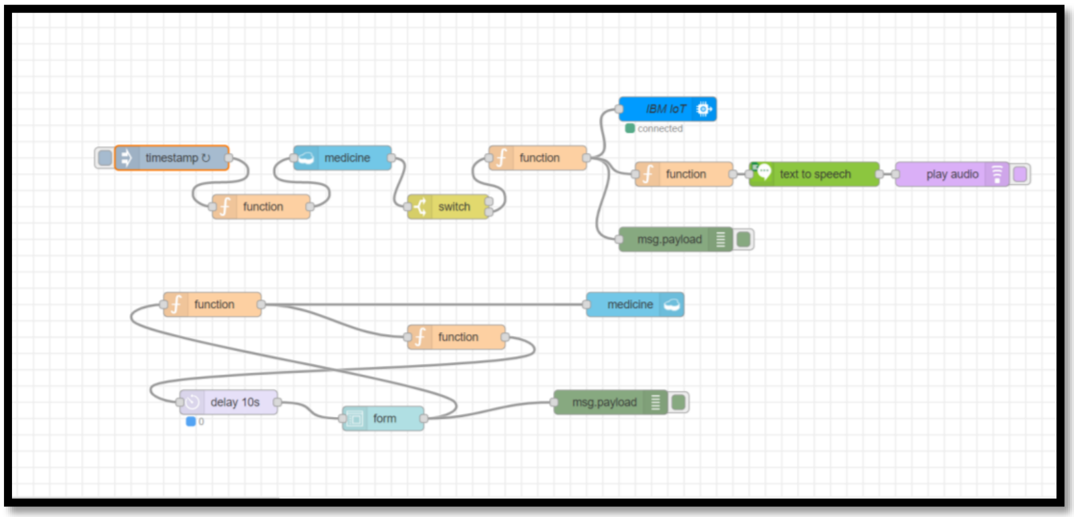
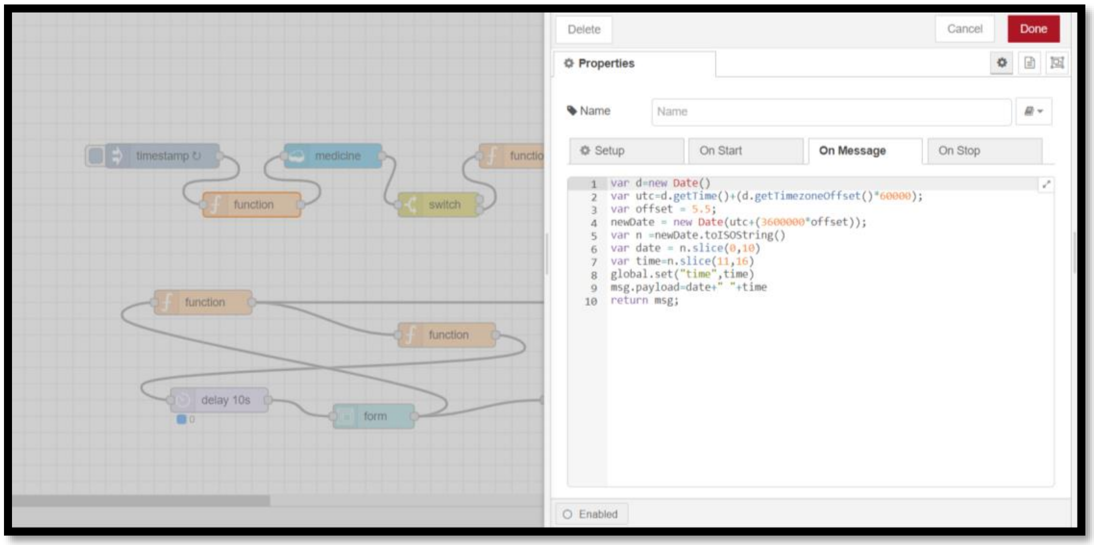
Date	08 November 2022
Team ID	PNT2022TMID54420
Project Name	PERSONAL ASSISTANCE FOR SENIOR WHO ARE SELF-RELIANT

SPRINT-2

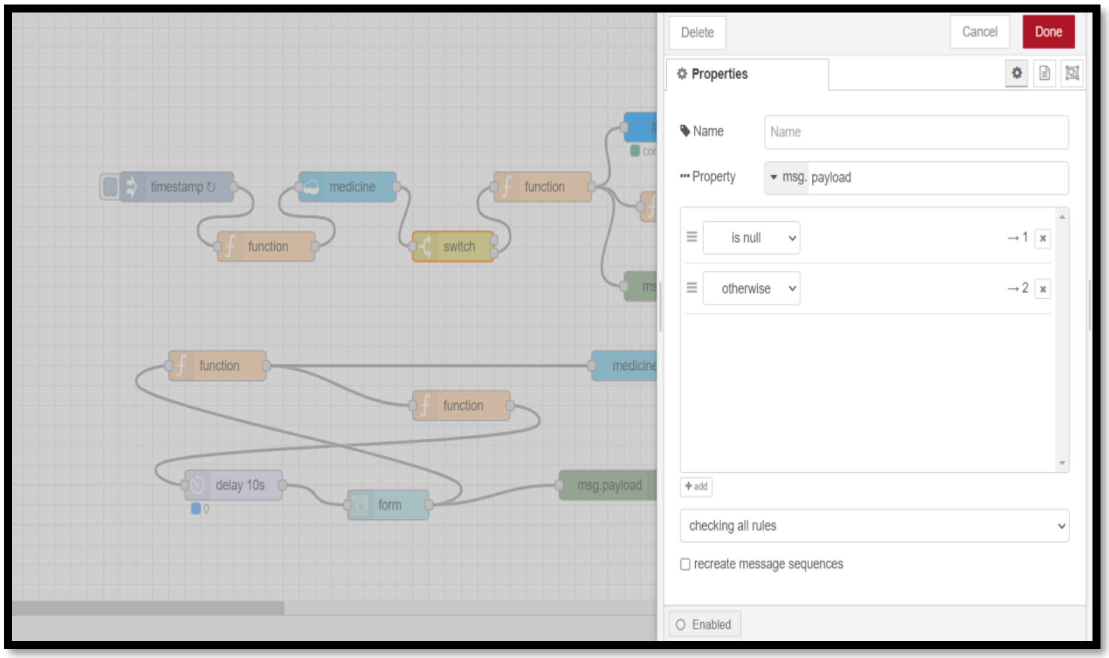
CREATE NODE-RED FORM



ADDING IOT CREDENTIALS



```
var d=new Date()
var utc=d.getTime()+(d.getTimezoneOffset()*60000);
var offset = 5.5;
newDate = new Date(utc+(3600000*offset));
var n =newDate.toISOString()
var date = n.slice(0,10)
var time=n.slice(11,16)
global.set("time",time)
msg.payload=date+" "+time
return msg;
```



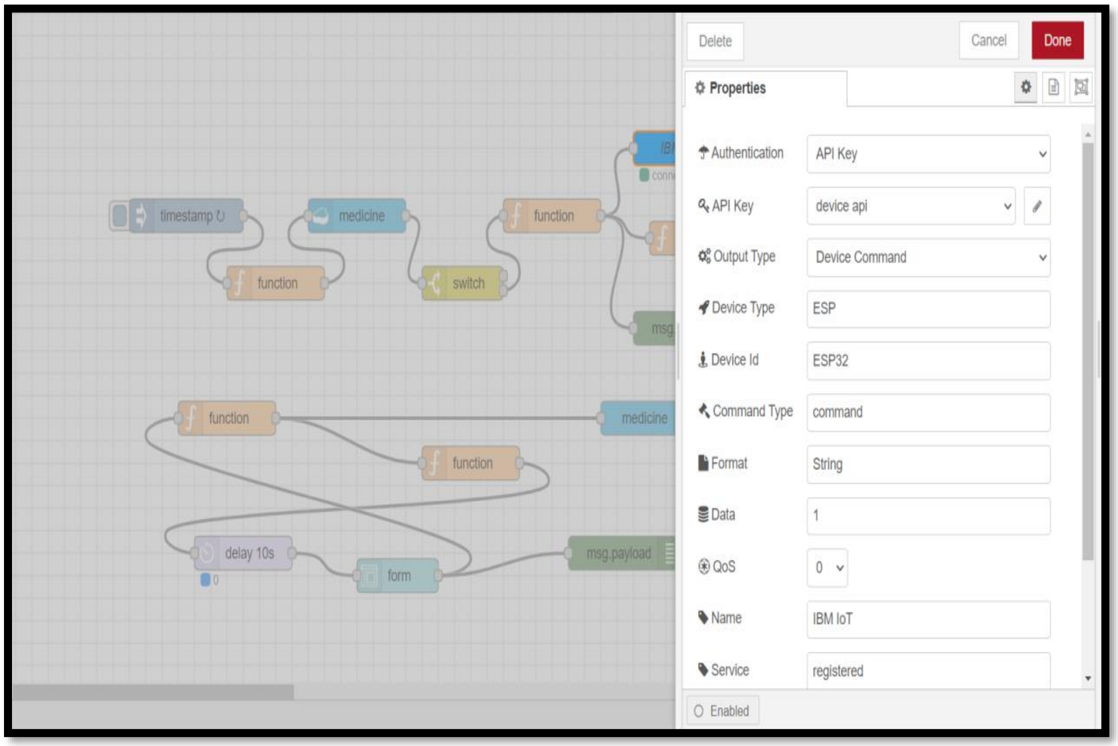
Name: Name
Property: msg. payload

is null		→ 1
otherwise		→ 2

+add

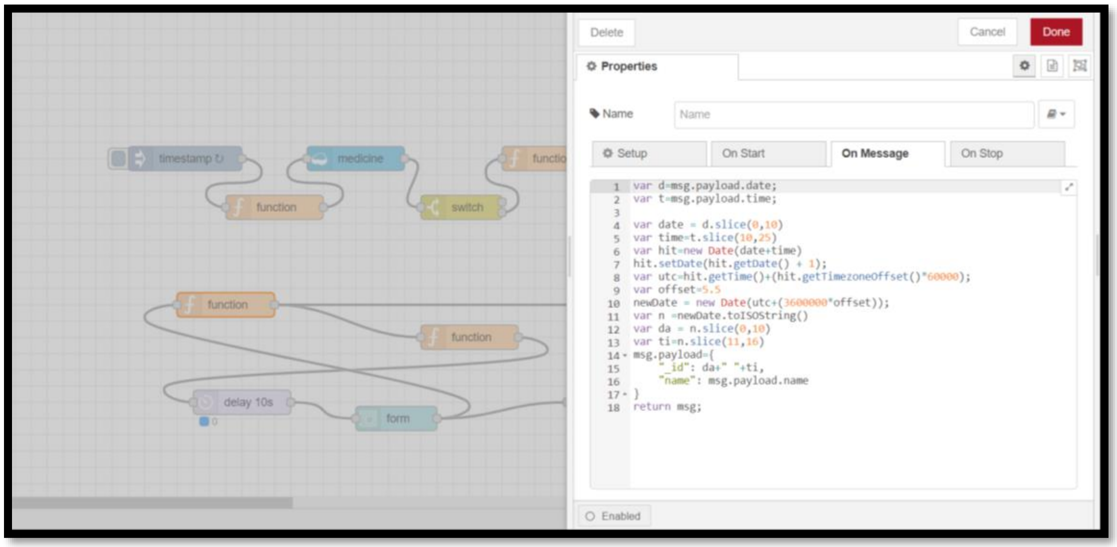
checking all rules

☐ recreate message sequences

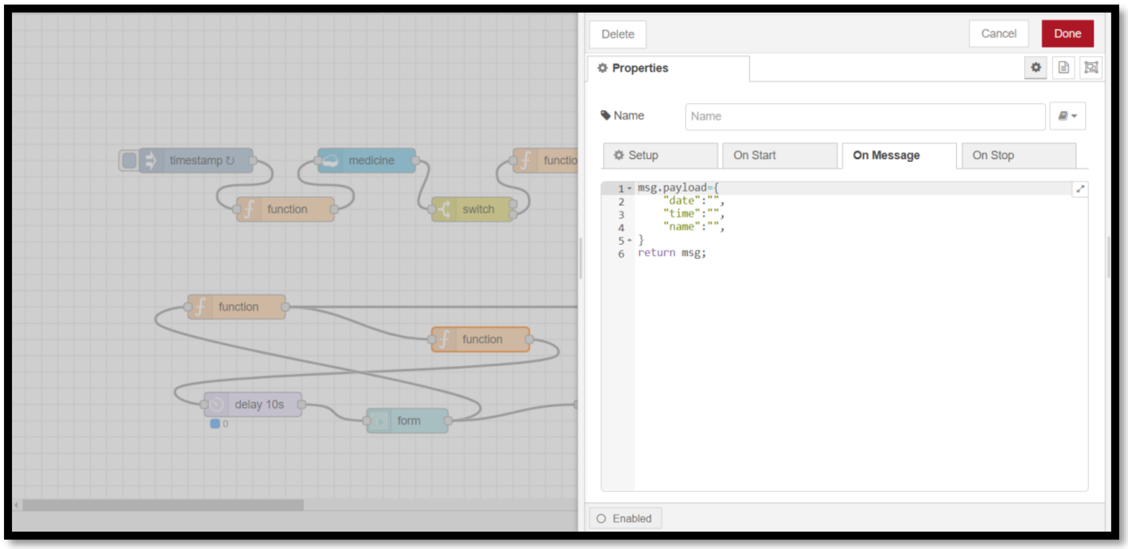


Authentication	API Key
API Key	device api
Output Type	Device Command
Device Type	ESP
Device Id	ESP32
Command Type	command
Format	String
Data	1
QoS	0
Name	IBM IoT
Service	registered

FUNCTION TO GET MEDICINE DETAILS AND TIME



```
var d=msg.payload.date;
var t=msg.payload.time;

var date = d.slice(0,10)
var time=t.slice(10,25)
var hit=new Date(date+time)
hit.setDate(hit.getDate() + 1);
var utc=hit.getTime()+(hit.getTimezoneOffset()*60000);
var offset=5.5
newDate = new Date(utc+(3600000*offset));
var n =newDate.toISOString()
var da = n.slice(0,10)
var ti=n.slice(11,16)
msg.payload={
     "_id": da+" "+ti,
     "name": msg.payload.name
}
return msg;
```



```
msg.payload={
     "date":"",
     "time":"",
     "name":"",
}
return msg;
```

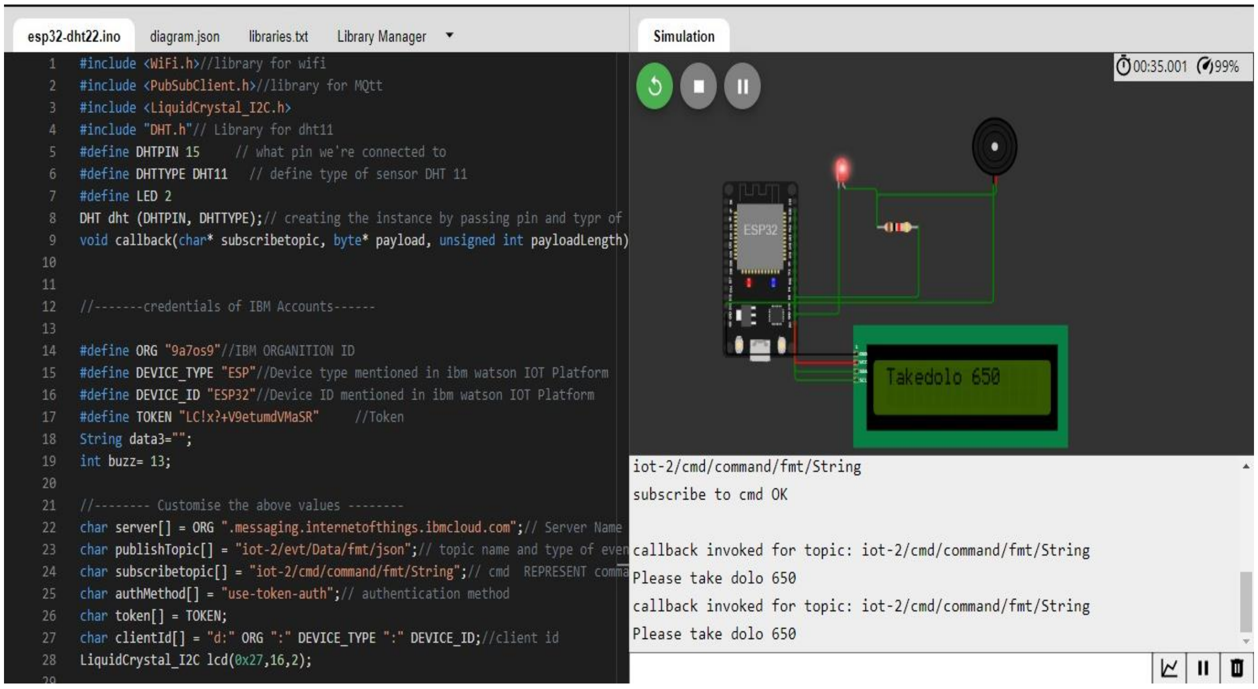
CODE FOR SIMULATION:

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include <LiquidCrystal_I2C.h>
#include "DHT.h"// Library for dht11
#define DHTPIN 15     // what pin we're connected to
#define DHTTYPE DHT11   // define type of sensor DHT 11
#define LED 2
DHT dht (DHTPIN, DHTTYPE);// creating the instance by passing pin and typr of
dht connected
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
```

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

#define ORG "9a7os9"//IBM ORGANITION ID
#define DEVICE_TYPE "ESP"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "ESP32"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "LC!x?+V9etumdVMaSR"     //Token
String data3="";
int buzz= 13;


//-------- 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 and format in which data to be send
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd  REPRESENT
command type 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
LiquidCrystal_I2C lcd(0x27,16,2);

//-----------------------------------------
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient); //calling the
predefined client id by passing parameter like server id,portand wificredential
void setup()// configureing the ESP32
{
  Serial.begin(115200);
  pinMode(LED,OUTPUT);
  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
}

void loop()// Recursive Function
{
  if (!client.loop()) {
    mqttconnect();
  }
}

/*.....................................retrieving to
```

```
Cloud. ............................ */

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 defination for wificonnect
{
  Serial.println(); Serial.print("Connecting
  to ");

  WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish
the connection
  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++) {
    //Serial.print((char)payload[i]);
    data3 += (char)payload[i];
  }

  Serial.println("Please take "+ data3);
  if(data3 != "")
  {
    lcd.init();
    lcd.print("Take"+ data3);

digitalWrite(LED,HIGH);
delay(20000);
digitalWrite(LED,LOW);

  }

  else
  {
digitalWrite(LED,LOW);

  }
data3="";
}
```



```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include <LiquidCrystal_I2C.h>
#include "DHT.h"// Library for dht11
#define DHTPIN 15     // what pin we're connected to
#define DHTTYPE DHT11   // define type of sensor DHT 11
#define LED 2
DHT dht (DHTPIN, DHTTYPE);// creating the instance by passing pin and typr of
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)


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

#define ORG "9a7os9"//IBM ORGANITION ID
#define DEVICE_TYPE "ESP"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "ESP32"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "LC!x?+V9etumdVMaSR"     //Token
String data3="";
int buzz= 13;

//-------- 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 even
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd  REPRESENT comma
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id
LiquidCrystal_I2C lcd(0x27,16,2);
```

```
iot-2/cmd/command/fmt/String
subscribe to cmd OK

callback invoked for topic: iot-2/cmd/command/fmt/String
Please take dolo 650
callback invoked for topic: iot-2/cmd/command/fmt/String
Please take dolo 650
```