

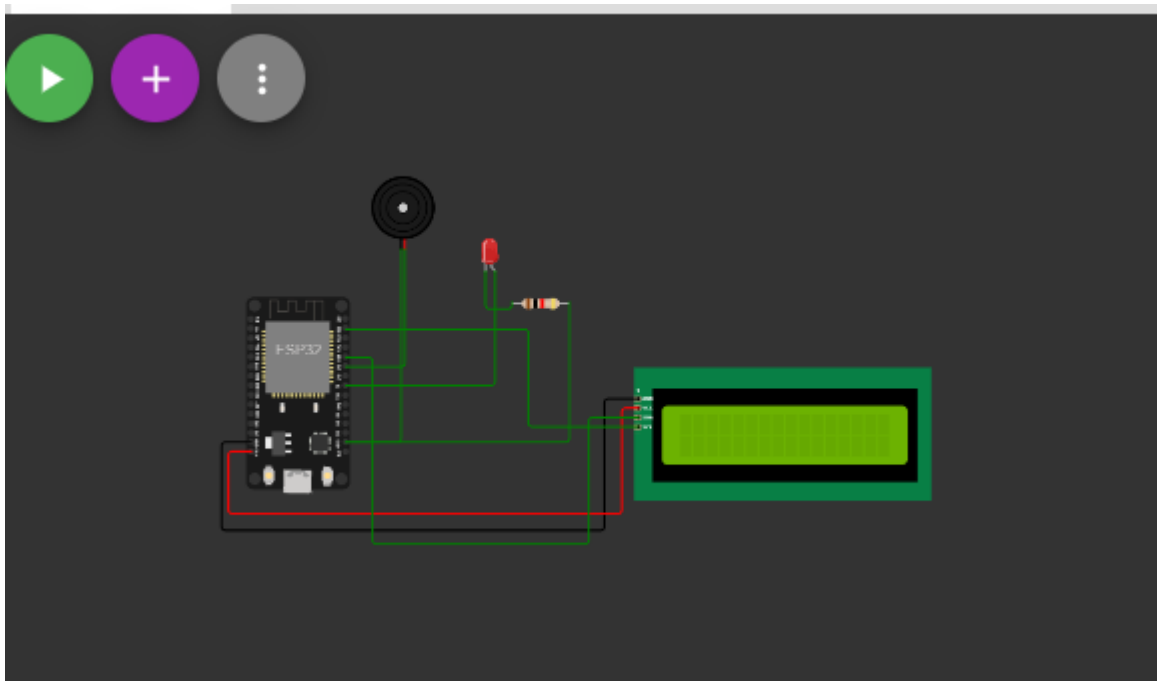
PROJECT DEVELOPMENT PHASE- SPRINT 3

Team ID	PNT2022TMID24841
Project Name	Personal Assistance for Seniors Who Are Self-reliant

SPRINT 3-Creating a IOT device using ESP32 and notify user when the medicine time arrives.

1.DEVICE SETUP

The device consists of EPS32 which is used for the connecting the IBM cloud r to publish and subscribe data. The LED glows when the medicine time arrives. The LCD displays the medicine name and the buzzer rings.



CODE:

```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQTT
#define LED 5
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);
void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);
//-----credentials of IBM Accounts-----
#define ORG "ular9x"//IBM ORGANITION ID
#define DEVICE_TYPE "MedicineRem"//Device type mentioned in ibm watson IOT
Platform
#define DEVICE_ID "dev01"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "ZTt_Cr00t4zB9sBkmX" //Token
String data3,light;
float h, t;
#define BUZZER_PIN 19 // ESP32 GIOP21 pin connected to Buzzer's pin
//----- 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/test/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
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient); //calling the
predefinedclient id by passing parameter like server id,portand wificredential
void setup()// configuring the ESP32
{
  Serial.begin(115200);
  Serial.begin(9600);
  // dht.begin();
  pinMode(LED,OUTPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  delay(10);
  lcd.init();
  lcd.clear();
  lcd.backlight();
  Serial.println();
  wificonnect();
  mqttconnect();
}
void loop()// Recursive Function
{
  digitalWrite(BUZZER_PIN, HIGH);
```

```

delay(1000);
if (!client.loop()) {
  mqttconnect();
}
}

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);
  light=(char)payload[0];
  for (int i = 1; i < payloadLength; i++) {
    Serial.print((char)payload[i]);
    data3 += (char)payload[i];
  }
}

```

```

}
// Make sure backlight is on
Serial.println("data: "+ data3);
if(light=="n")
{
    digitalWrite(BUZZER_PIN, HIGH);
    Serial.println(data3);
    digitalWrite(LED,HIGH);
    // Print a message on both lines of the LCD.
    lcd.setCursor(2,0); //Set cursor to character 2 on line 0
    lcd.print("It's time for");
    lcd.setCursor(2,1); //Move cursor to character 2 on line 1
    lcd.print(data3);
    delay(3000);
    digitalWrite(BUZZER_PIN, LOW);
    digitalWrite(LED,LOW);
    lcd.clear();
}
else
{
    digitalWrite(BUZZER_PIN, LOW);

    Serial.println(data3);
    digitalWrite(LED,LOW);
    lcd.clear();
}
data3="";
}

```

2.WEB DASHBOARD:

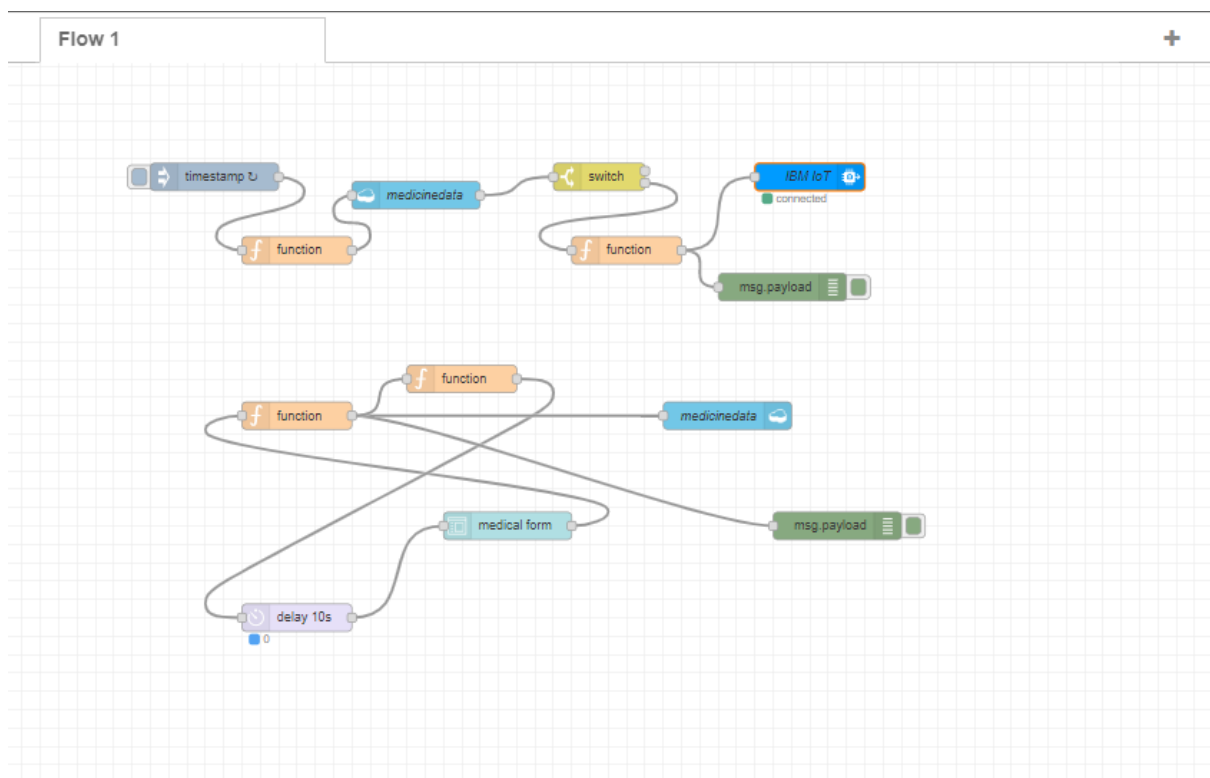
The user can enter the medicine name, data and time when it has to be consumed which will be saved in the cloudant databases and node-red checks in the cloudant database if any medicine has to be taken every minute.

[home](#)

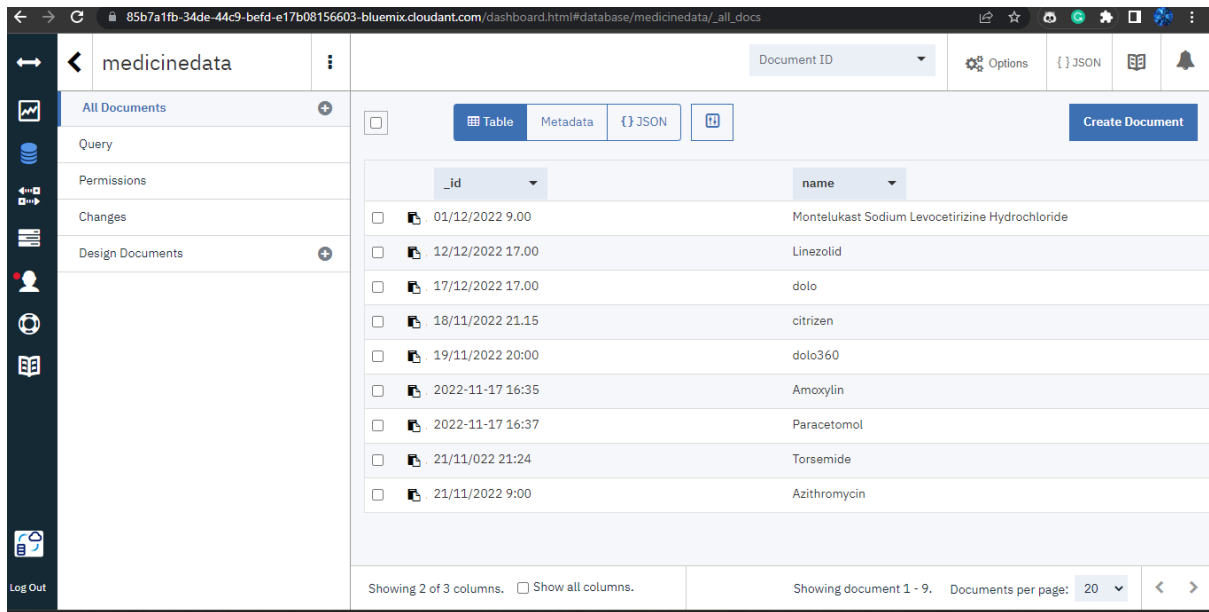
medical form

3.NODE-RED FLOW:

When user adds the medicine into the database the node red flow keeps checking the database every minute and if the medicine has to be taken, it issues a command to the IOT device through IBM IOT Watson platform.



4.MEDICINE DATABASE:

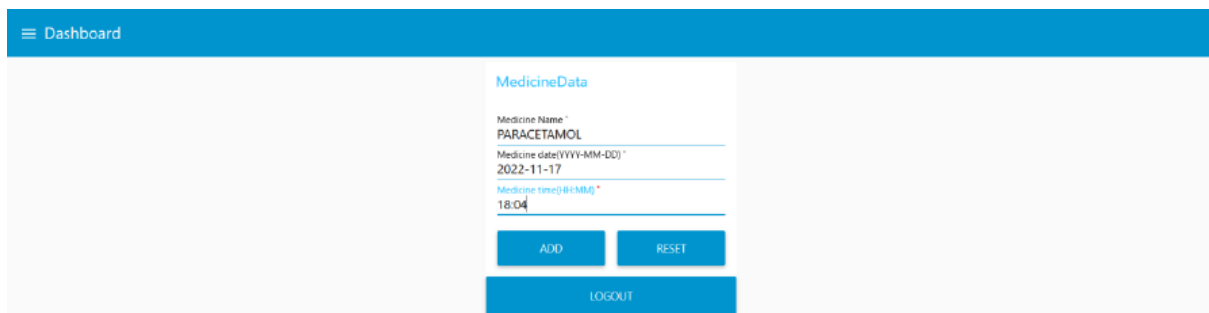


_id	name
01/12/2022 9:00	Montelukast Sodium Levocetirizine Hydrochloride
12/12/2022 17:00	Linezolid
17/12/2022 17:00	dolo
18/11/2022 21:15	citizen
19/11/2022 20:00	dolo360
2022-11-17 16:35	Amoxylin
2022-11-17 16:37	Paracetamol
21/11/2022 21:24	Torsemide
21/11/2022 9:00	Azithromycin

5. WORKING

When the medicine time arrives the node-red flow sends DEVICE COMMAND to the IBM IOT platform. ESP32 which has subscribed to the IBM IOT platform turns the LED and displays the medicine name on the LCD display on receiving the command.

ADDING MEDICINE



MedicineData

Medicine Name *
PARACETAMOL

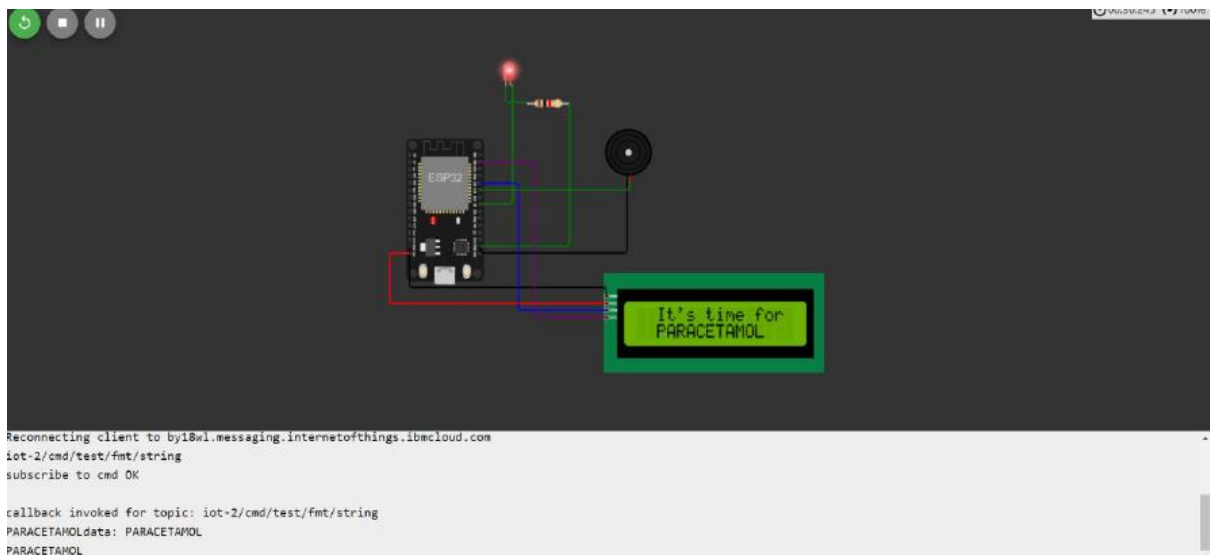
Medicine date(YYYY-MM-DD) *
2022-11-17

Medicine time(H:MM) *
18:04

ADD RESET

LOGOUT

When the medicine time arrives the LED in the devices glows, LCD displays the medicine name and buzzer ring.



Reconnecting client to by18w1.messaging.internetofthings.ibmcloud.com
iot-2/cmd/test/fmt/string
subscribe to cmd OK

callback invoked for topic: iot-2/cmd/test/fmt/string
PARACETAMOLdata: PARACETAMOL
PARACETAMOL