Medicine Reminder For Elderly People Using IBM Cloud

TEAM :

1. K.AKASH : 19R11A0469
2. K.PAVAN PRATHIK : 19R11A0471
3. P.AKSHAYA : 19R11A0483
4. P.HEMANTH : 19R11AO486

INTRODUCTION :

Overview :

Sometimes elderly people forget to take the medicine at the correct time. They also forget which medicine He / She should take at that particular time... And it is difficult for doctors/caretakers to monitor the patients around the clock. To avoid this problem, this medicine remainder system is developed. An is built for the user (caretaker) which enables him to set the desired time and medicine. These details will be stored in the IBM Cloudant DB. if the medicine time arrives the web application will send the medicine name to the IoT Device through the IBM IoT platform. The device will receive the medicine name and notify the user with a voice command

Purpose :

Medicine remainder helps the elderly people by taking medicines at the correct time,so that there will be no delay in taking medicines. Most of the time due to the people as well as regarding age and some disease which leads to forget the basic things among daily routine.If the patient is suffering from the disease where it is compulsory to take medicine the proper time this medicine remainder system helps them a lot.

LITERATURE SURVEY :

Existing problem :

In modern society, busy life has made people forget many things in day to day life . the elderly people and the people victims of chronicle diseases who need to take the medicines timely without missing are suffering from dementia, which is forgetting things in their daily routine. Considering this situation study has been done in this.

Proposed Solution :

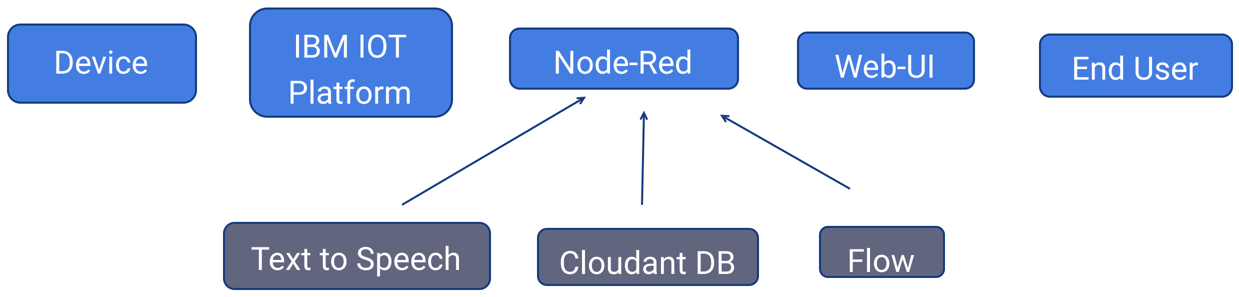
The medicine reminder system will have one duty and that would be to remind the user that he is due for taking the medicine. We are trying to make sure that the users never forgets to take the medicine and hence we do the remainder in three ways.

- One is that we have visual indicator which would be the display.
- In the case that patient is outside, we have a mobile reminder app which will remind using mobile notifications for that time.
- The mobile application can be installed in the android devices. It will add recurring events to the mobile's calendar and will alert the user when he has to take the medicine with the list of medicines and it's prescribed dosage.

<u>THEORETICAL ANALYSIS :</u>

- Users can configure the medicine name, time through a web application.
- All the medicine details will be stored in the IBM Cloudant DB.
- The web application will send the medicine name to the IoT device at the desired time.
- After getting the medicine name the device will speak out the medicine name using IBM text to speech Service to intimate the user to take the medicine

BLOCK DIAGRAM



Hardware:

- Esp32 dev module
- ssd 1306 oled display

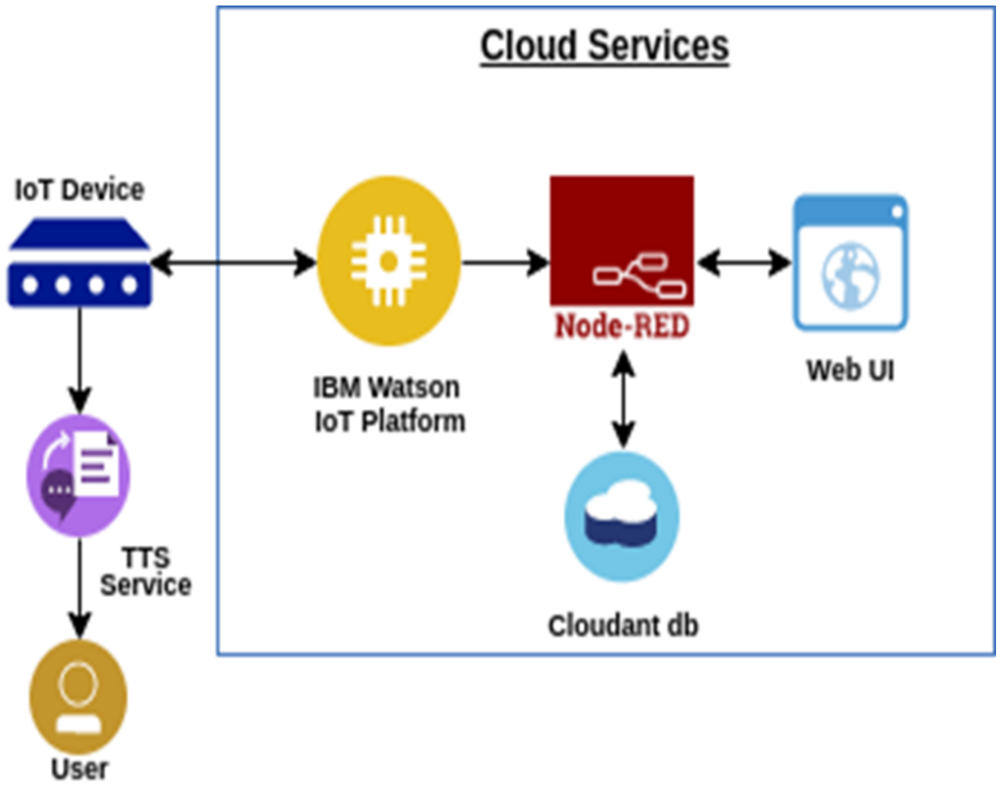
Software Designing :

- Python IDLE
- IBM IOT Platform
- Node-Red Service
- Cloud Database
- Arduino
- TTS service

EXPERIMENTAL INVESTIGATIONS:

With the latest research in the fields of medicine and pharmaceutical sciences, an increasing number of drugs are being invented to cure many fatal diseases and this has enabled people to live healthier and longer. Consequently, the rate of medication usage is increasing, particularly among the elderly. It is important that the medication prescribed by doctors is taken at the correct time of the day and in the correct dosage. This seems to be a significant problem encountered by older people with disabilities. Because of forgetfulness or other problems, many older patients either do not take the correct medicines or the correct dosage at the appropriate times and this reduces the benefits of treatments and increases both healthcare costs and the mortality rates. In order to find a solution to this problem, a IOT and Cloud-based smart application has been developed to improve patient adherence, particularly for older people with disabilities. The novelty of the developed application is the use of a Cloud service to provide two-way communication in the form of feedback between the older patients with disabilities and the doctors so that the medication adherence of the patients can be monitored.Developers should find the developed IOT mobile application based medication reminder application for older people with disabilities a useful example application, and it should help them in developing other mobile applications for older people with disabilities.

FLOWCHART :



RESULT :

1. I was able to store the data in a cloudant database
2. I was able to display the medicine details on the Web User-Interface
3. I was able to alert the users by voice command and through mobile

ADVANTAGES AND DISADVANTAGES :

1. **Receive reminders to take your medications you can get timely reminders for taking your medicine regularly.**
2. **We can only remind the person but not make him take the medicine forcefully**
3. **The user must be able to read, write and understand the basic english language.**
4. **Not only medicine details it can display an other text**

APPLICATIONS :

More and more senior citizens in the world are facing the problem of living alone. People are social and need to communicate with other people. Loneliness is also damaging the health of senior citizens in some ways. Research shows that family companionship is very important for the elderly. It is worth noticing that my product is considered family companionship. I need to put this consideration to seniors' daily life. So, how to alleviate loneliness for senior citizens in their daily life? As we all know, medication is a critical part for senior citizens. There are so many who need to notice when they should take their medicine.

CONCLUSION:

In this paper, we introduce a case study of a medication reminder system that helps, mainly, elderly patients who forget to take their medicines at prescribed time. The system can provide a nearby reminding or remotely by sending SMS message to the distant patient after specific period of time. We used HW/SW Co-design approach to allow the hardware and the software of the system designed and implemented in parallel and make sure that the non-functional properties are met.

FUTURE SCOPE :

As future work, we are planning to improve out medication reminder system by introducing additional features using mobile app and include other medical services such as providing liquid doses and measuring Blood Pressure, Blood Glucose meter and insulin injection.

BIBILOGRAPHY:

- **Smartbridge:Tutorials,Mentors**

SOURCE CODE:

```
import wiotp.sdk.device
import time
import os
import datetime
from ibm_watson import TextToSpeechV1
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
import playsound

authenticator = IAMAuthenticator('1uMWLtaB5gI6HluwS47oy1uqDOmJeLJICi7B-ECWOtT6')
text_to_speech = TextToSpeechV1(
    authenticator=authenticator
    )

text_to_speech.set_service_url('https://api.eu-gb.text-to-speech.watson.cloud.ibm.com/instance
s/43e154dd-ff43-4304-a48b-5ea87abdf7cd')

myConfig = {
    "identity": {
        "orgId": "unneo9",
        "typeId": "ESP32",
        "deviceId": "12345"

    },
    "auth": {
        "token": "12345678"
    }
}
client=wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()


def myCommandCallback(cmd):
    print("Message received from IBM IOT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']
    with open('medicine.mp3', 'wb') as audio_file:
        audio_file.write(
            text_to_speech.synthesize(
                'You have to take '+m+' medicine now',
                voice='en-US_AllisonV3Voice',
                accept='audio/mp3'
            ).get_result() .content)
    playsound.playsound('medicine.mp3')
    os.remove('medicine.mp3')
while True:
        client.commandCallback = myCommandCallback
client.disconnect()
```

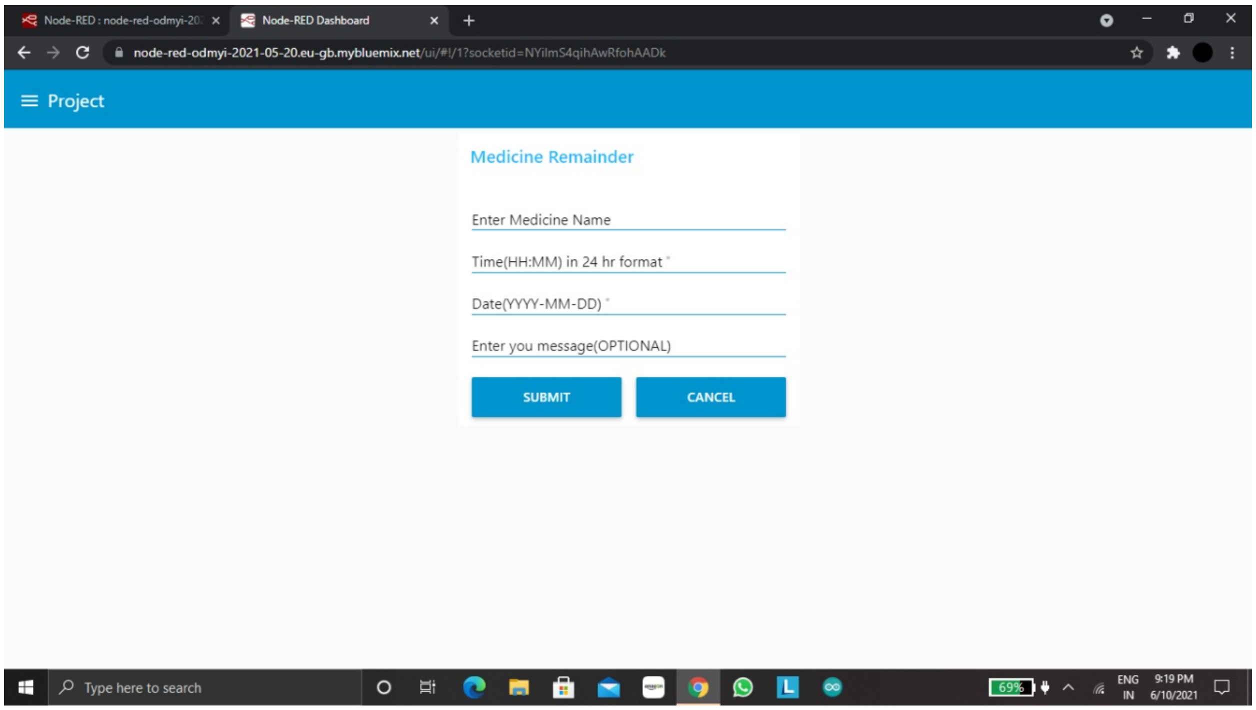
Arduino code:

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

String command;
String data="";

void callback(char* topic, byte* payload, unsigned int payloadLength);

// CHANGE TO YOUR WIFI CREDENTIALS
const char* ssid = "Akshaya";//your wifi ssid
const char* password = "akki@1234";//your password

// CHANGE TO YOUR DEVICE CREDENTIALS AS PER IN IBM BLUMIX
#define ORG "unneo9"
#define DEVICE_TYPE "ESP32"
#define DEVICE_ID "12345"
#define TOKEN "12345678" //  Authentication Token OF THE DEVICE

//  PIN DECLARATIONS

//-------- Customise the above values --------
const char publishTopic[] = "iot-2/evt/Data/fmt/json";
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char topic[] = "iot-2/cmd/home/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);
```

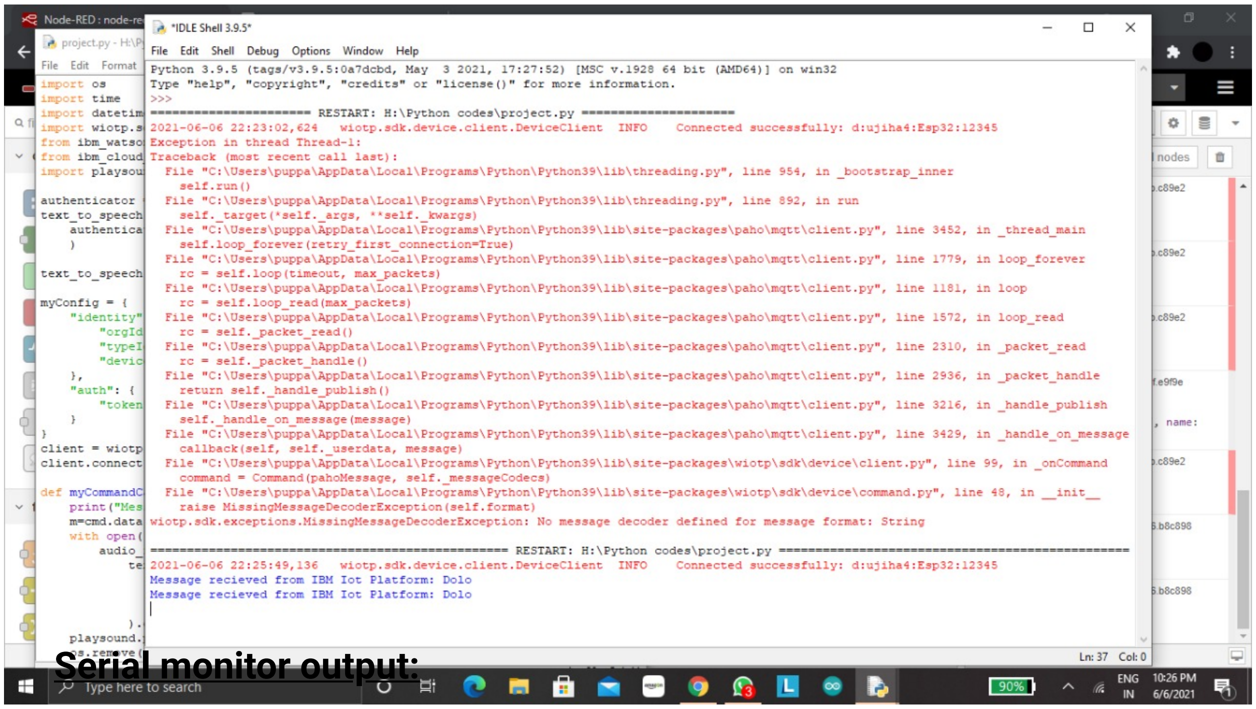
```
int publishInterval = 5000; // 30 seconds
long lastPublishMillis;
void publishData();

void setup() {
  Serial.begin(115200);
  Serial.println();



  wifiConnect();
  mqttConnect();
   delay(2000);
Serial.println("oled test");
  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println("SSD1306 allocation failed");
    for(;;);
  }
}

void loop() {



 if (millis() - lastPublishMillis > publishInterval)
  {
    publishData();
    lastPublishMillis = millis();
  }
  if (!client.loop()) {
    mqttConnect();
  }
}

void wifiConnect() {
  Serial.print("Connecting to "); Serial.print(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.print("WiFi connected, IP address: ");
  Serial.println(WiFi.localIP());
```

```
}

void mqttConnect() {
  if (!client.connected())
  {
    Serial.print("Reconnecting MQTT client to "); Serial.println(server);
    while (!client.connect(clientId, authMethod, token)) {
      Serial.print(".");
      delay(500);
    }
    initManagedDevice();
    Serial.println();
  }
}

void initManagedDevice() {
  if (client.subscribe(topic)) {
    Serial.println("subscribe to cmd OK");
  } else {
    Serial.println("subscribe to cmd FAILED");
  }
}

void callback(char* topic, byte* payload, unsigned int payloadLength) {
  Serial.print("callback invoked for topic: ");
  Serial.println(topic);

  for (int i = 0; i < payloadLength; i++) {
    command+= (char)payload[i];
  }

  control_func();
  command= "";
}

void control_func()
{
  Serial.println("You need to take " +command+ " medicine");

    display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(0, 10);
  // Display static text
  display.print("You need to take " +command+ " medicine");

  display.display();
}

void publishData()
{

}
```

UI output Screenshot


Node-RED : node-red-odmyi-20
Node-RED Dashboard
node-red-odmyi-2021-05-20.eu-gb.mybluemix.net/ui/#!/1?socketid=NYilmS4qihAwRfohAADk
Project
Medicine Remainder
Enter Medicine Name
Time(HH:MM) in 24 hr format *
Date(YYYY-MM-DD) *
Enter you message(OPTIONAL)
SUBMIT
CANCEL
Type here to search
69%
ENG
IN
9:19 PM
6/10/2021

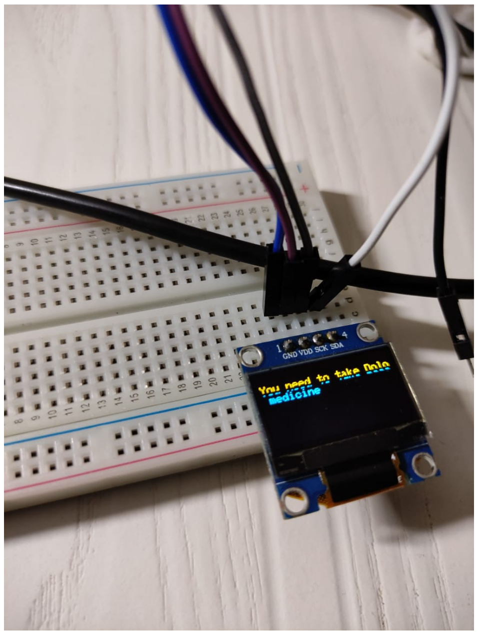
Python output:



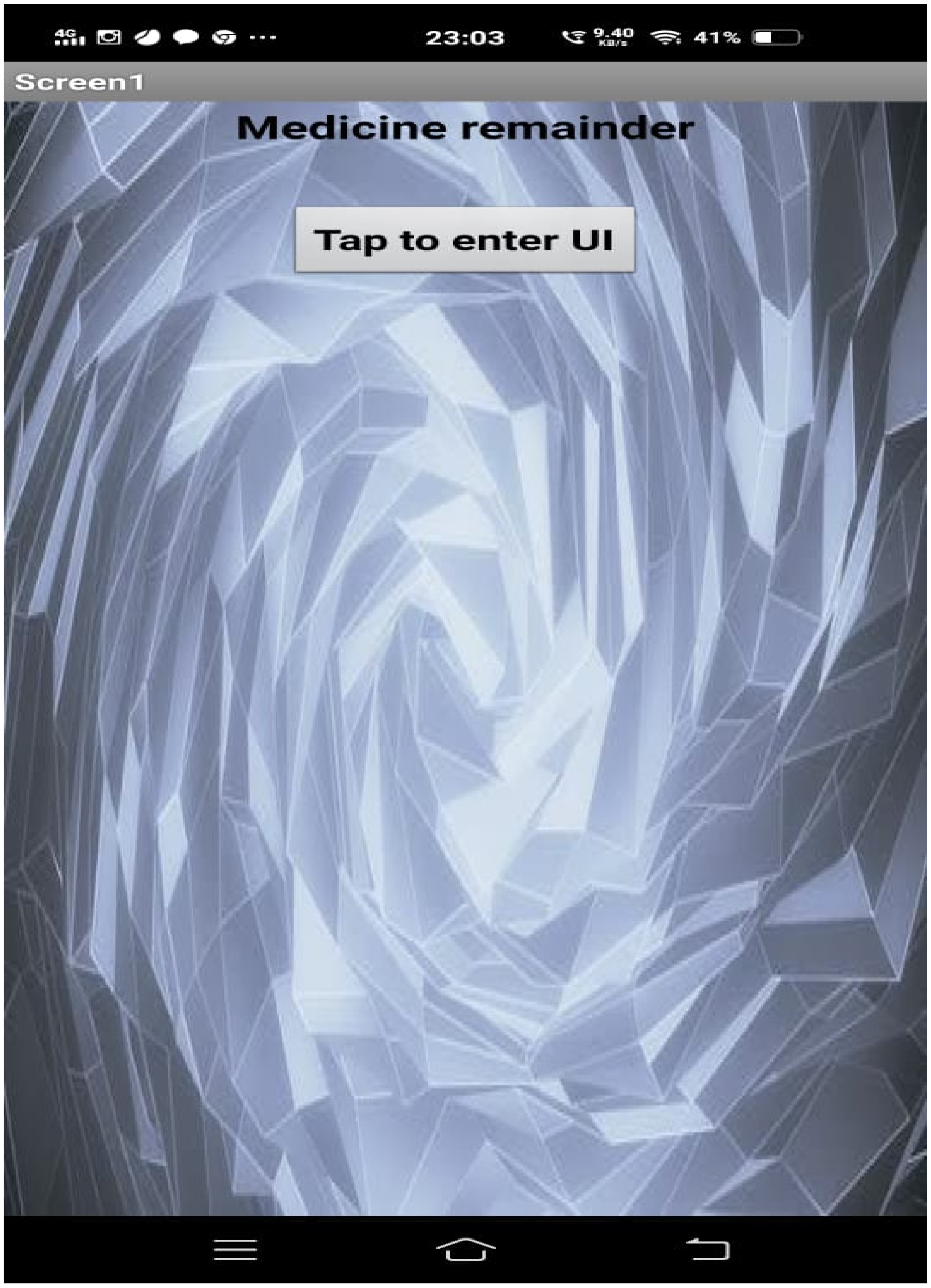
```
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May  3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
====================== RESTART: H:\Python codes\project.py =====================
2021-06-06 22:23:02,624   wiotp.sdk.device.client.DeviceClient  INFO    Connected successfully: d:ujiha4:Esp32:12345
Exception in thread Thread-1:
Traceback (most recent call last):
  File "C:\Users\puppa\AppData\Local\Programs\Python\Python39\lib\threading.py", line 954, in _bootstrap_inner
    self.run()
  File "C:\Users\puppa\AppData\Local\Programs\Python\Python39\lib\threading.py", line 892, in run
    self._target(*self._args, **self._kwargs)
  File "C:\Users\puppa\AppData\Local\Programs\Python\Python39\lib\site-packages\paho\mqtt\client.py", line 3452, in _thread_main
    self.loop_forever(retry_first_connection=True)
  File "C:\Users\puppa\AppData\Local\Programs\Python\Python39\lib\site-packages\paho\mqtt\client.py", line 1779, in loop_forever
    rc = self.loop(timeout, max_packets)
  File "C:\Users\puppa\AppData\Local\Programs\Python\Python39\lib\site-packages\paho\mqtt\client.py", line 1181, in loop
    rc = self.loop_read(max_packets)
  File "C:\Users\puppa\AppData\Local\Programs\Python\Python39\lib\site-packages\paho\mqtt\client.py", line 1572, in loop_read
    rc = self._packet_read()
  File "C:\Users\puppa\AppData\Local\Programs\Python\Python39\lib\site-packages\paho\mqtt\client.py", line 2310, in _packet_read
    rc = self._packet_handle()
  File "C:\Users\puppa\AppData\Local\Programs\Python\Python39\lib\site-packages\paho\mqtt\client.py", line 2936, in _packet_handle
    return self._handle_publish()
  File "C:\Users\puppa\AppData\Local\Programs\Python\Python39\lib\site-packages\paho\mqtt\client.py", line 3216, in _handle_publish
    self._handle_on_message(message)
  File "C:\Users\puppa\AppData\Local\Programs\Python\Python39\lib\site-packages\paho\mqtt\client.py", line 3429, in _handle_on_message
    callback(self, self._userdata, message)
  File "C:\Users\puppa\AppData\Local\Programs\Python\Python39\lib\site-packages\wiotp\sdk\device\client.py", line 99, in _onCommand
    command = Command(pahoMessage, self._messageCodecs)
  File "C:\Users\puppa\AppData\Local\Programs\Python\Python39\lib\site-packages\wiotp\sdk\device\command.py", line 48, in __init__
    raise MissingMessageDecoderException(self.format)
wiotp.sdk.exceptions.MissingMessageDecoderException: No message decoder defined for message format: String

================================================= RESTART: H:\Python codes\project.py =================================================
2021-06-06 22:25:49,136   wiotp.sdk.device.client.DeviceClient  INFO    Connected successfully: d:ujiha4:Esp32:12345
Message recieved from IBM Iot Platform: Dolo
Message recieved from IBM Iot Platform: Dolo
```

Serial monitor output:

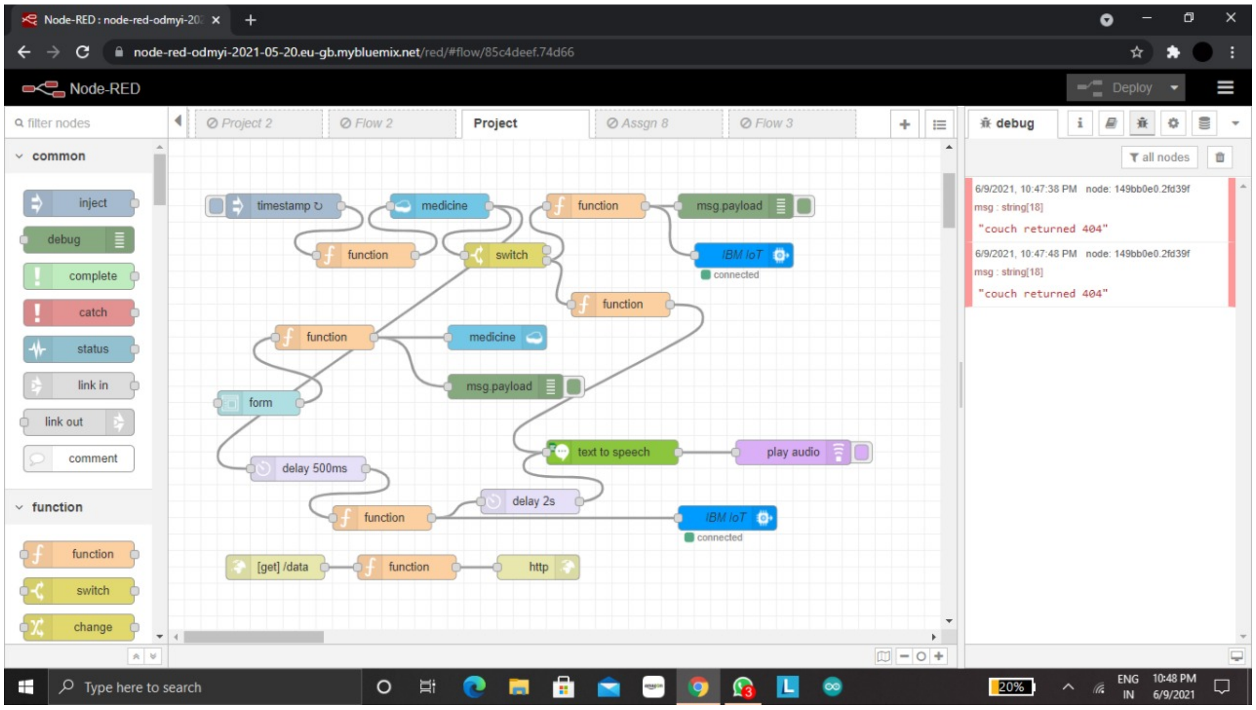
OLED Display:



Mobile app(using MIT app inventor):



NODES:



```
6/9/2021, 10:47:38 PM  node: 149bb0e0.2fd39f
msg : string[18]
"couch returned 404"
6/9/2021, 10:47:48 PM  node: 149bb0e0.2fd39f
msg : string[18]
"couch returned 404"
```