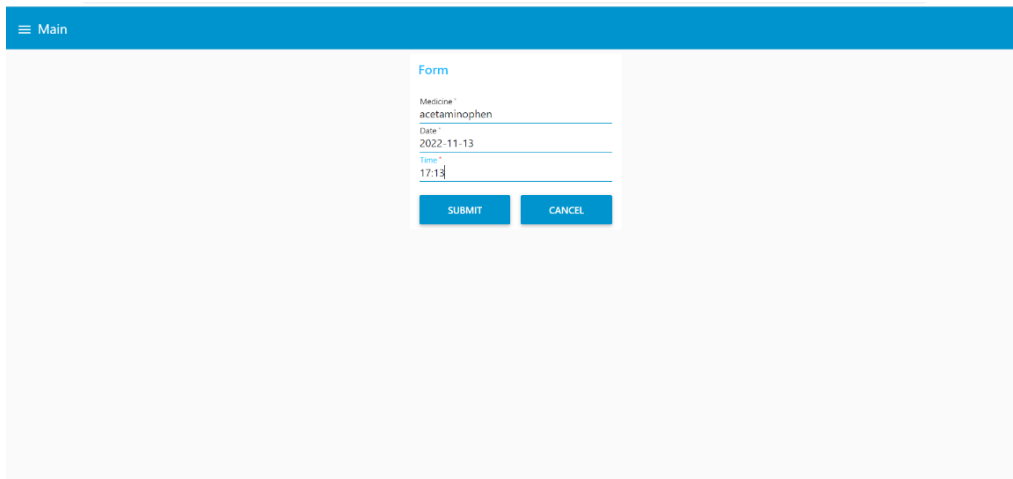


# Final Deliverables

Team ID	PNT2022TMID00479
Project Name	Personal Assistance for Seniors Who Are Self-Reliant.
Team Members	Madhan .S, Madhavan K.C.A, Kevin Gandhi .T, Ponvigneswaran .M

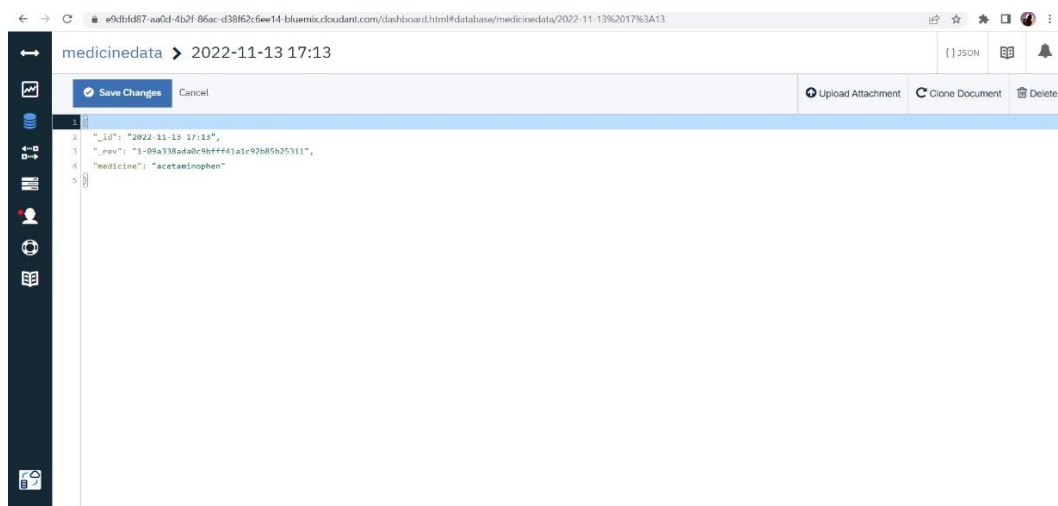
## Web Application

### 1. Get Data From User:

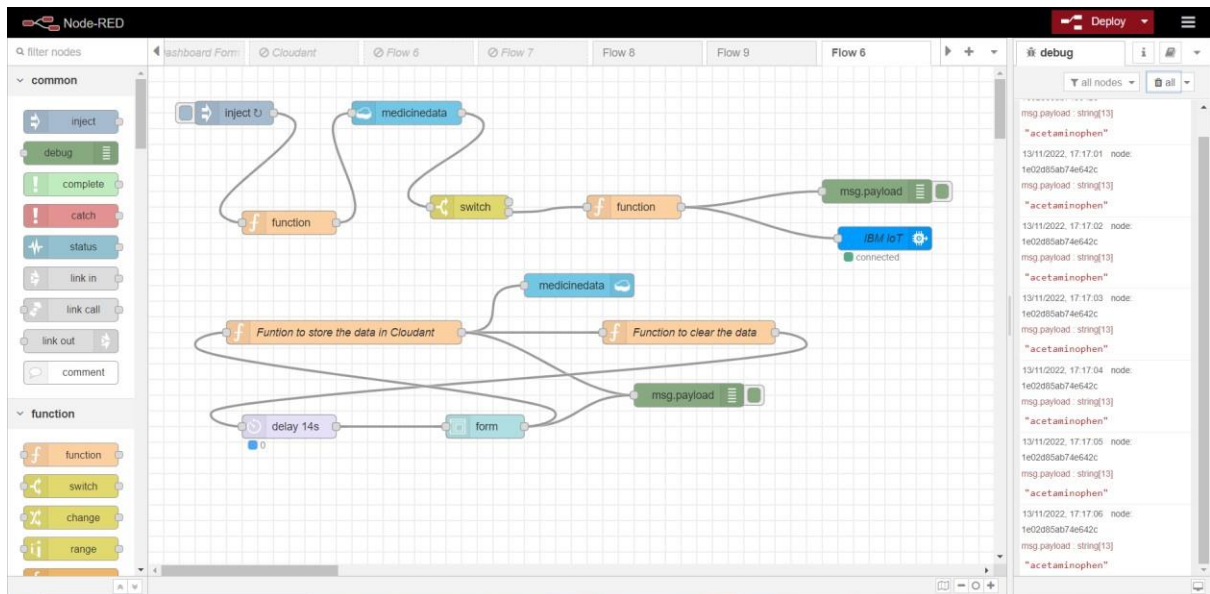


The screenshot shows a web application interface with a blue header bar containing a hamburger menu icon and the text 'Main'. Below the header, there is a light gray background. In the center, a white form titled 'Form' is displayed. The form contains three input fields: 'Medicine' with the value 'acetaminophen', 'Date' with the value '2022-11-13', and 'Time' with the value '17:13'. Below the input fields are two buttons: 'SUBMIT' and 'CANCEL'.

### 2. Stored in Cloudant



### 3. Display in Node-red



## 4. Streaming in Watson IoT Platform

The screenshot displays the IBM Watson IoT Platform interface. At the top, the navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A search bar is present with the text 'Search by Device ID'. The main content area shows a table of devices. The first device, 'b11m3edevicid', is in a 'Connected' state. Below the device list, the 'Recent Events' tab is selected, showing a stream of events. The events table has columns for Event, Value, Format, and Last Received. All events are from an 'IoT Device' and contain the value '{"medicine": "acetaminophen"}' in 'json' format, received 'a few seconds ago'.

Event	Value	Format	Last Received
IoT Device	{"medicine": "acetaminophen"}	json	a few seconds ago
IoT Device	{"medicine": "acetaminophen"}	json	a few seconds ago
IoT Device	{"medicine": "acetaminophen"}	json	a few seconds ago
IoT Device	{"medicine": "acetaminophen"}	json	a few seconds ago
IoT Device	{"medicine": "acetaminophen"}	json	a few seconds ago

## 5. Simulation

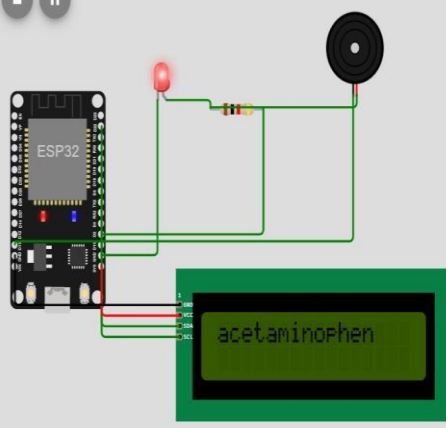
WOKWI SAVE SHARE Medicine Remainder Docs

PNT2022TMD50622.ino diagram.json libraries.txt Library Manager

```
1 #include <WiFi.h> // Library for wifi
2 #include <PubSubClient.h> // Library for MQTT
3 #include <LiquidCrystal_I2C.h>
4 #include "DHT.h" // Library for dht11
5 #define DHTPIN 15 // what pin we're connected to
6 #define DHTTYPE DHT11 // define type of sensor DHT 11
7 #define LED 2
8 DHT dht (DHTPIN, DHTTYPE); // creating the instance by passing pin and type of dht
9 void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
10
11 //-----credentials of IBM Accounts-----
12
13 #define ORG "64yf7x" // IBM ORGANIZATION ID
14 #define DEVICE_TYPE "b1m3edevicetype" // Device type mentioned in ibm watson IOT
15 #define DEVICE_ID "b1m3edeviceid" // Device ID mentioned in ibm watson IOT Platform
16 #define TOKEN "<div>~&Emtr7L-v-Gz2G))e" // Token
17 String data3="";
18 int buzz= 13;
19
20 //----- Customise the above values -----
21 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
22 char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of event
23 char subscribetopic[] = "iot-2/cmd/command/fmt/String"; // cmd REPRESENT command
24 char authMethod[] = "use-token-auth"; // authentication method
25 char token[] = TOKEN;
26 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; // client id
27 LiquidCrystal_I2C lcd(0x27,16,2);
28
29 //-----
30
31 WiFiClient wifiClient; // creating the instance for wifiClient
32 PubSubClient client(server, 1883, callback, wifiClient); // calling the predefined
33
34 void setup() // configuring the ESP32
35 {
```

Simulation

00:21.421 89%



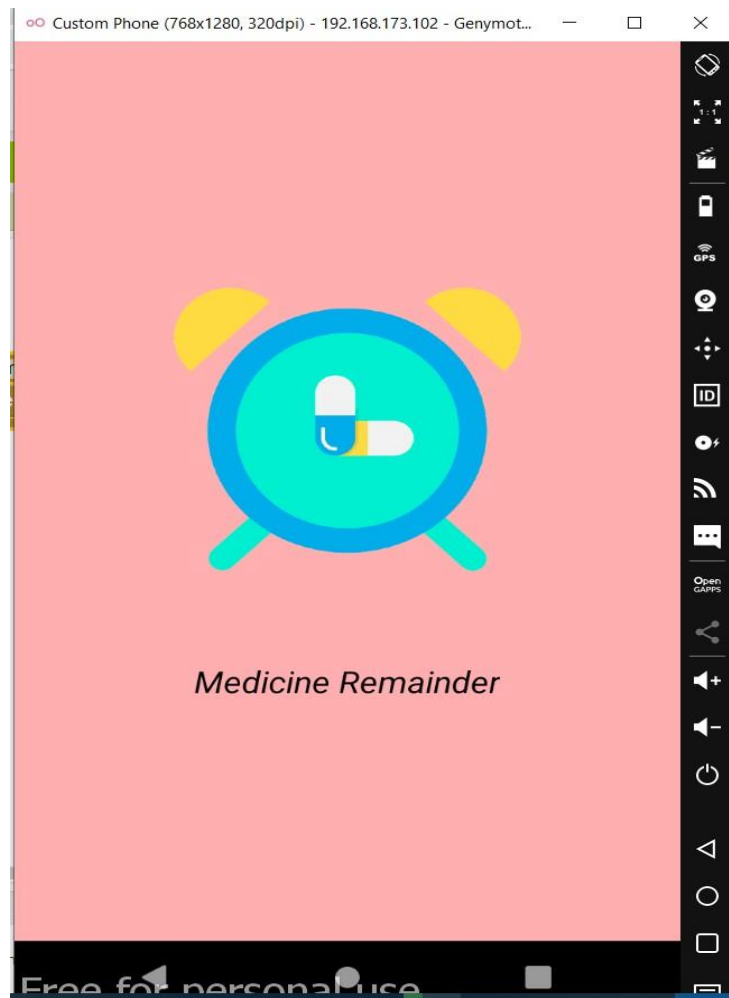
Medicine Name: acetaminophen  
callback invoked for topic: iot-2/cmd/command/fmt/String  
Medicine Name: acetaminophen  
callback invoked for topic: iot-2/cmd/command/fmt/String  
Medicine Name: acetaminophen  
callback invoked for topic: iot-2/cmd/command/fmt/String  
Medicine Name: acetaminophen

**Link:**

<https://wokwi.com/projects/348198638815543891>

# Mobile Application

## 1. Splash Screen



## 2. Get Data From User

1:05

Medicine Antibiotics

Date 2022-11-16

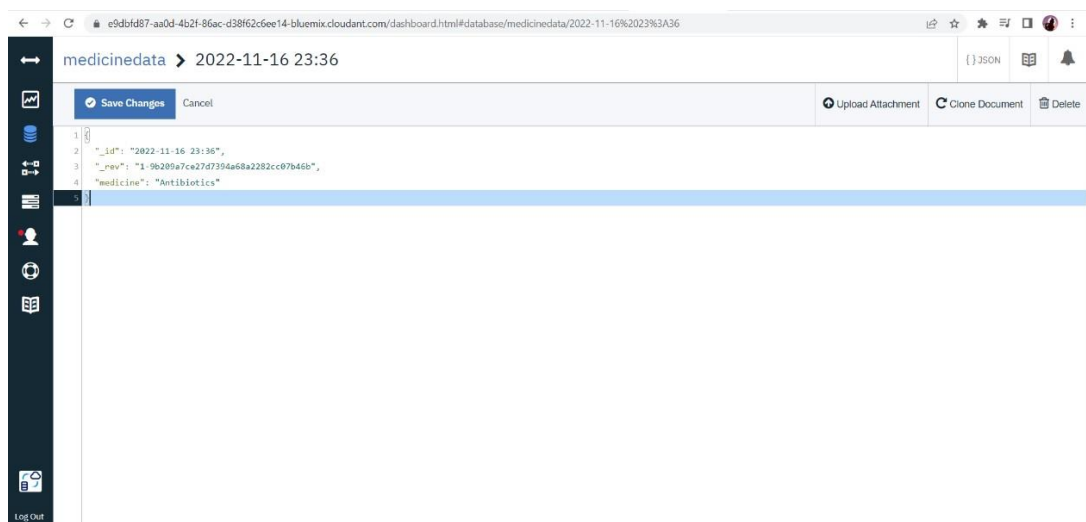
Time 23:36

Save

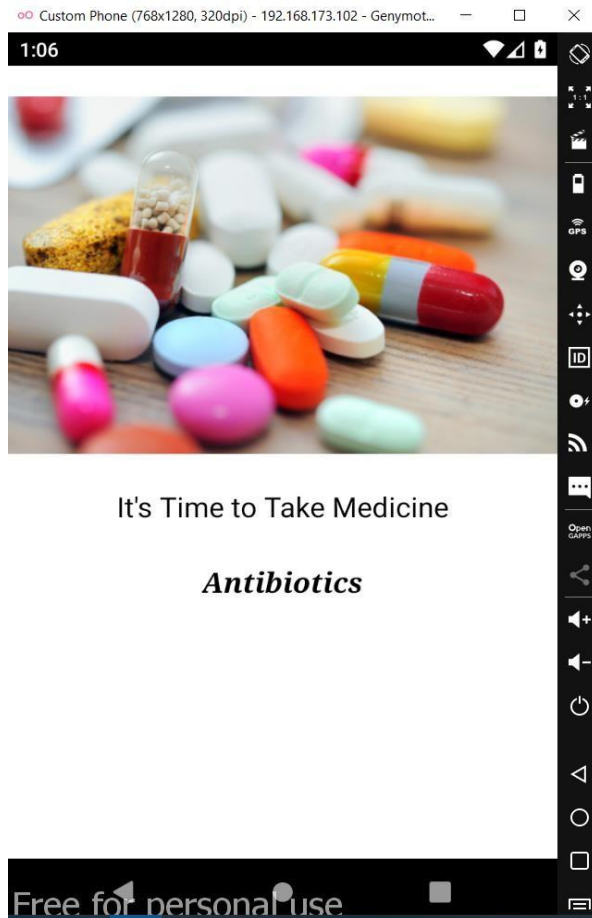
Saved

Free for personal use

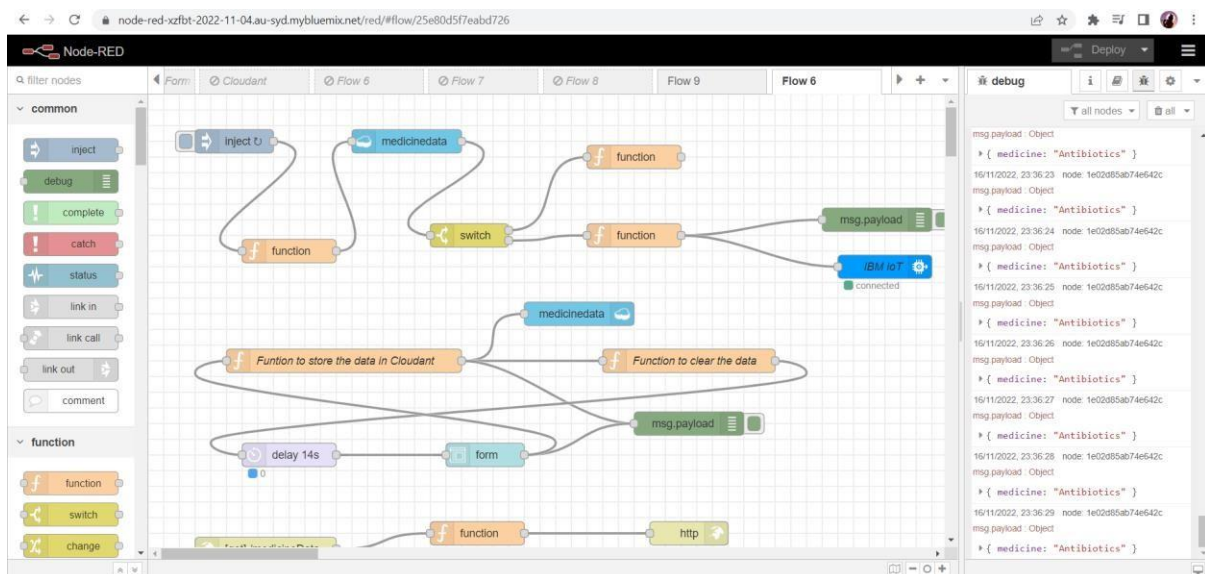
## 3. Store in Cloudant DB



## 4. Display Remainder with audio



## 5. Display in node-red



## 6. Remainder in Simulation

WOKWI

← → ↻ wokwi.com/projects/348198638815543891

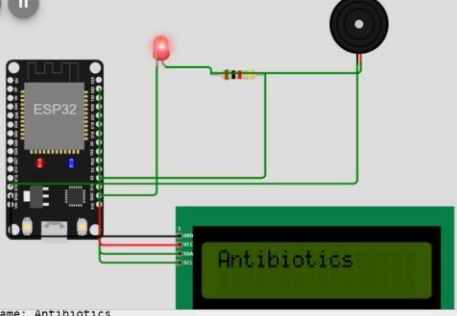
SAVE SHARE Medicine Remainder Docs

PNT2022TMD50622.ino diagram.json libraries.txt Library Manager

```
87 Serial.println("WiFi connected");
88 Serial.println("IP address: ");
89 Serial.println(WiFi.localIP());
90 }
91
92 void initManagedDevice() {
93   if (client.subscribe(subscribetopic)) {
94     Serial.println((subscribetopic));
95     Serial.println("subscribe to cmd OK");
96   } else {
97     Serial.println("subscribe to cmd FAILED");
98   }
99 }
100
101 void callback(char* subscribetopic, byte* payload, unsigned int) {
102   {
103     Serial.print("callback invoked for topic: ");
104     Serial.println(subscribetopic);
105     for (int i = 13; i < payloadlength-2; i++) {
106       //Serial.print((char)payload[i]);
107       data3 += (char)payload[i];
108     }
109   }
110
111   Serial.println("Medicine Name: " + data3);
112   if(data3 != "")
113   {
114     lcd.init();
115
116     lcd.print(data3);
117     digitalWrite(LED,HIGH);
118     tone(buzz, 100, 1000);
119   }
120 }
```

Simulation

00:33.232 90%



Medicine Name: Antibiotics  
callback invoked for topic: iot-2/cmd/command/fmt/String  
Medicine Name: Antibiotics  
callback invoked for topic: iot-2/cmd/command/fmt/String  
Medicine Name: Antibiotics  
callback invoked for topic: iot-2/cmd/command/fmt/String  
Medicine Name: Antibiotics