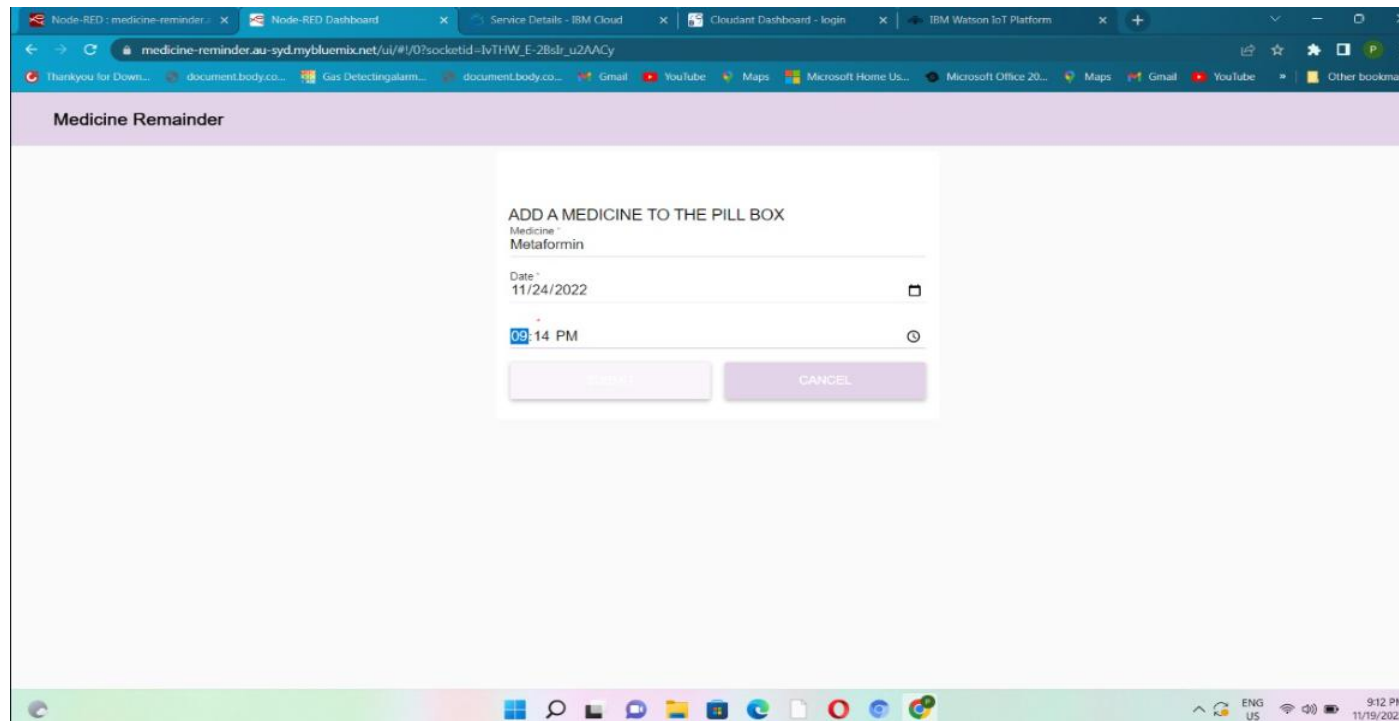


FINAL DELIVERABLES

Date	15 November 2022
Team ID	PNT2022TMID21568
Project Name	Personal Assistance for Seniors Who Are Self Reliant

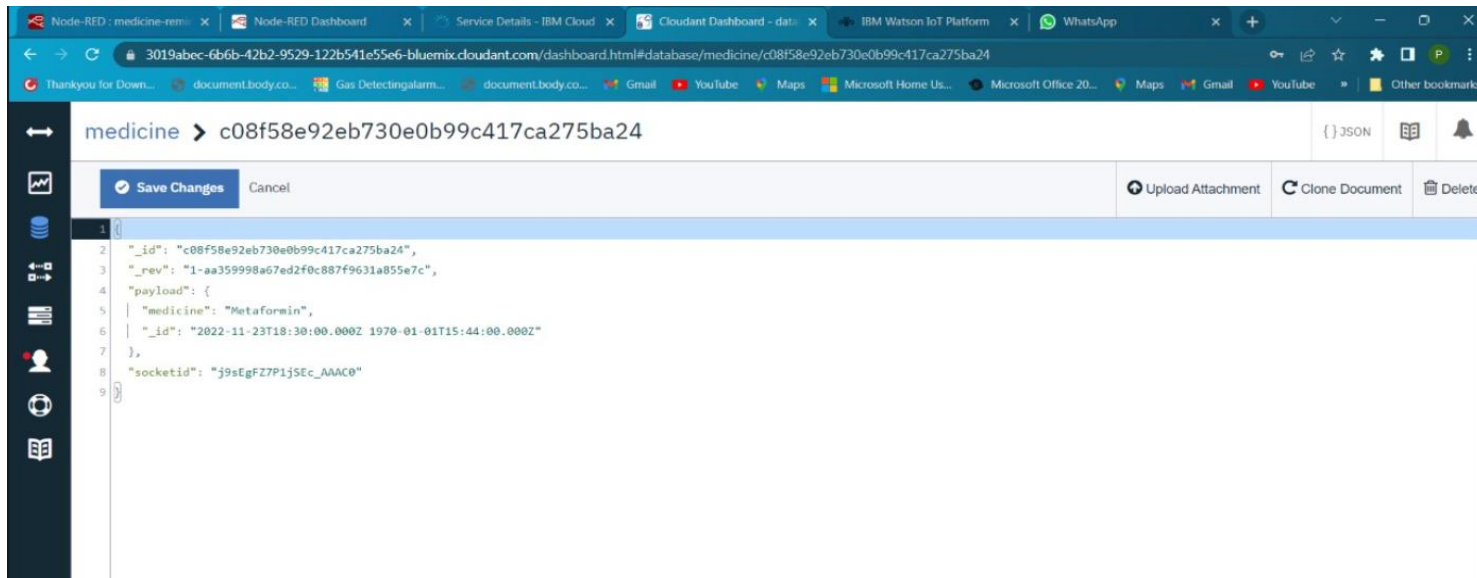
WEB APPLICATION

Getting data from the user:



The screenshot shows a web browser window with multiple tabs. The active tab is 'medicine-reminder.au-syd.mybluemix.net/ui/#/0?socketid=lvTHW_E-2Bslr_u2AACy'. The page title is 'Medicine Remainder'. The main content area displays a form titled 'ADD A MEDICINE TO THE PILL BOX'. The form has three input fields: 'Medicine *' with the value 'Metaformin', 'Date *' with the value '11/24/2022', and a time field with the value '09:14 PM'. There are two buttons at the bottom: 'ADD IT' and 'CANCEL'. The Windows taskbar is visible at the bottom of the screen, showing the date and time as 9:12 PM on 11/19/2022.

Store data in cloudant database:

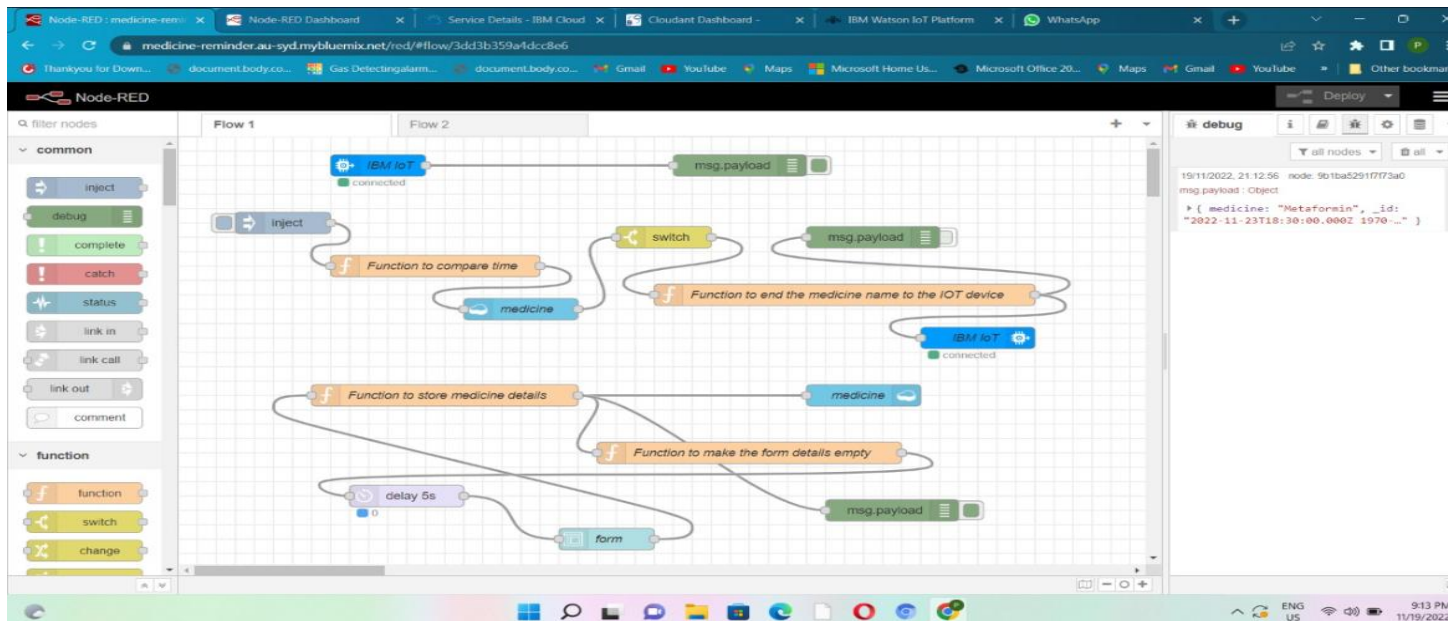


The screenshot shows the Cloudant Dashboard interface for a database named 'medicine'. The document ID is 'c08f58e92eb730e0b99c417ca275ba24'. The document content is a JSON object:

```
{  "_id": "c08f58e92eb730e0b99c417ca275ba24",  "_rev": "1-aa359998a67ed2f0c887f9631a855e7c",  "payload": {    "medicine": "Metaformin",    "_id": "2022-11-23T18:30:00.000Z 1970-01-01T15:44:00.000Z"  },  "socketid": "j9sEgFZ7P1jSEc_AAAC0"}
```

The interface includes buttons for 'Save Changes', 'Cancel', 'Upload Attachment', 'Clone Document', and 'Delete'.

Get the medicine details in Node Red:



The screenshot shows the Node-RED interface with a flow for retrieving medicine details. The flow includes the following nodes:

- inject** (blue)
- Function to compare time** (orange)
- medicine** (blue)
- Function to store medicine details** (orange)
- Function to make the form details empty** (orange)
- form** (blue)
- delay 5s** (purple)
- switch** (yellow)
- Function to end the medicine name to the IOT device** (orange)
- IBM IoT** (blue)
- msg.payload** (green)

The debug console on the right shows the following message:

```
19/11/2022, 21:12:56 node: 90tba5291f773a0  
msg.payload: Object  
> { medicine: "Metaformin", _id: "2022-11-23T18:30:00.000Z 1970-01-01T15:44:00.000Z" }
```

Streaming in IOT Watson Platform:

IBM Watson IoT Platform

abirammi@student.tce.edu
ID: 7kzrri

Browse Action Device Types Interfaces

medicineapp Disconnected nodemcu Device Nov 18, 2022 5:30 PM

Add Device +

Identity Device Information **Recent Events** State Logs

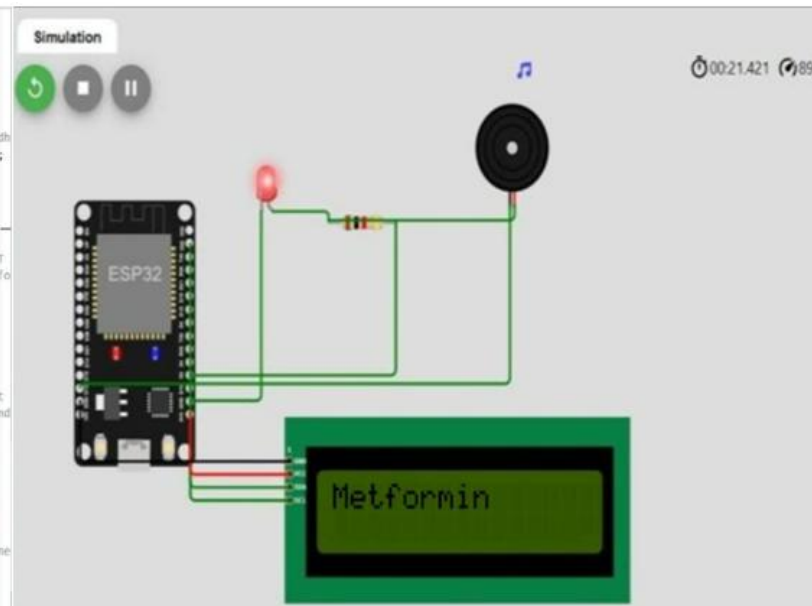
The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
event_1	{"randomNumber":18,"time":18,"date":100}	json	a few seconds ago
event_1	{"randomNumber":50,"time":52,"date":87}	json	a few seconds ago
event_1	{"randomNumber":40,"time":75,"date":98}	json	a few seconds ago

1 Simulation running

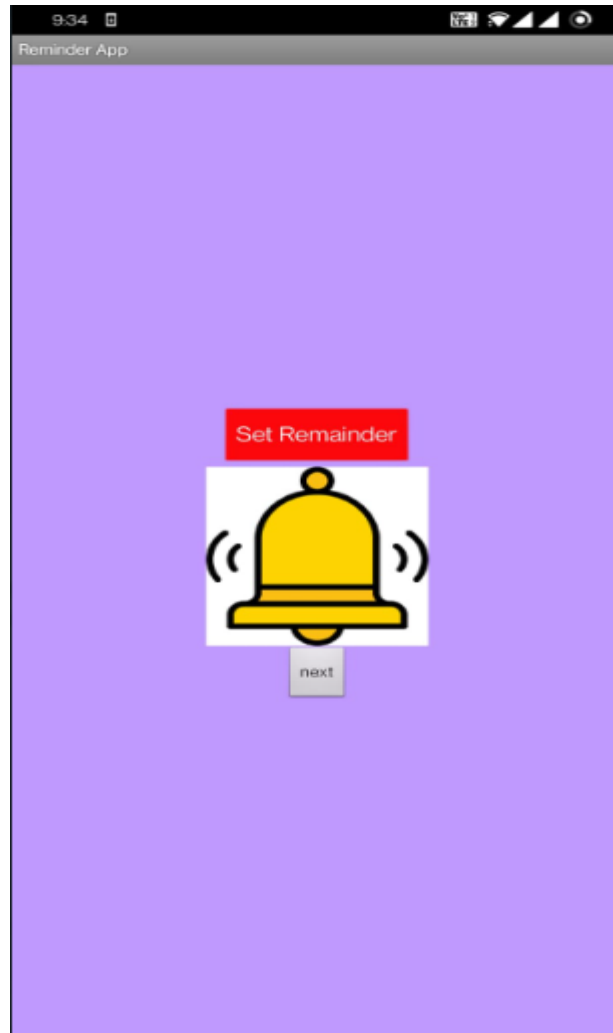
Simulation in Wowki:

```
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* topic, byte* payload, unsigned int payloadlength);
10
11
12 //-----credentials of IBM Accounts-----
13
14 #define ORG "64y7x" //IBM ORGANIZATION ID
15 #define DEVICE_TYPE "blimeidevicetype" //Device type mentioned in ibm watson IOT
16 #define DEVICE_ID "blimeideviceid" //Device ID mentioned in ibm watson IOT Platfo
17 #define TOKEN "&Htr7l-v-6z2G)e" //Token
18 String data="";
19 int buzz= 13;
20
21 //----- Customise the above values -----
22 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
23 char publishTopic[] = "iot-2/evt/Data/fmt/json"; // topic name and type of event
24 char subscribetopic[] = "iot-2/cmd/command/fmt/string"; // cmd REPRESENT command
25 char authMethod[] = "use-token-auth"; // authentication method
26 char token[] = TOKEN;
27 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id
28 LiquidCrystal_I2C lcd(0x27,16,2);
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 {
```

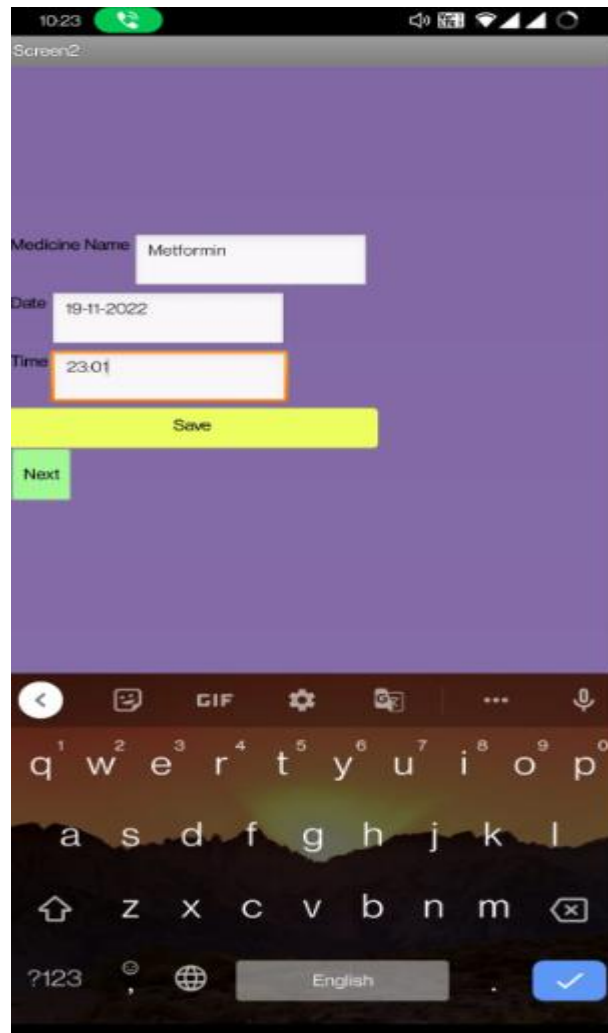


MOBILE APPLICATION

Home page:



Get data from user:



Alarm:

