SPRINT 4

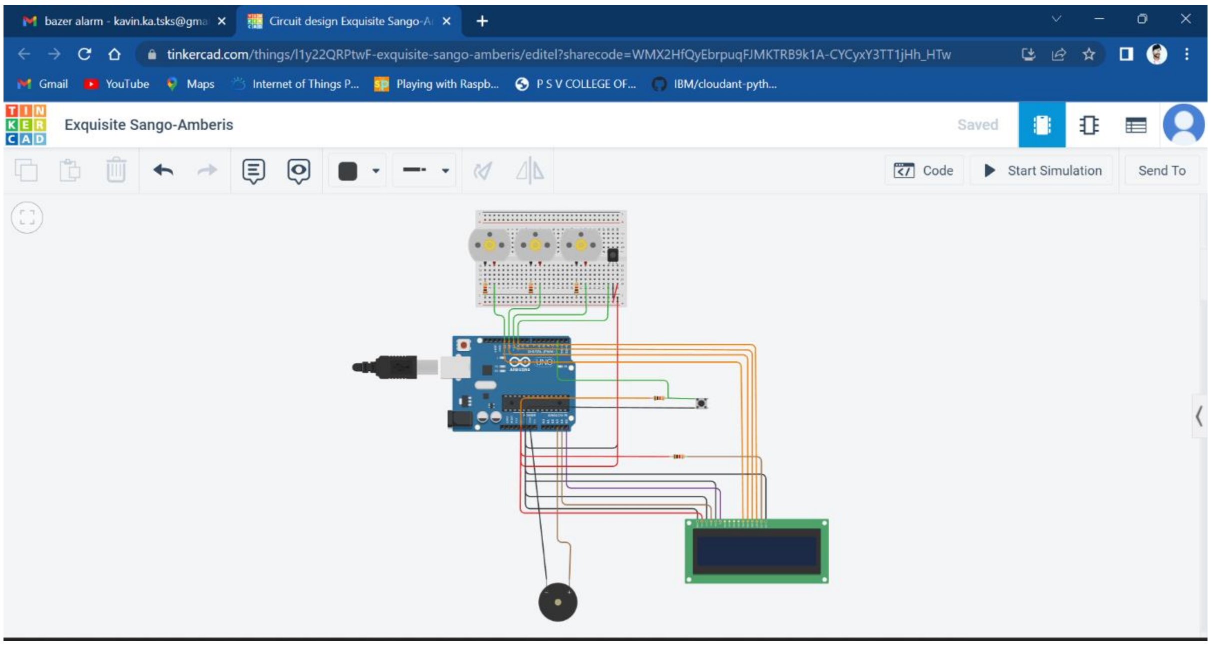
TEAM ID	**PNT2022TMID40888**
PROJECT TITLE	**Personal Assistance for seniors who are self-Reliant**

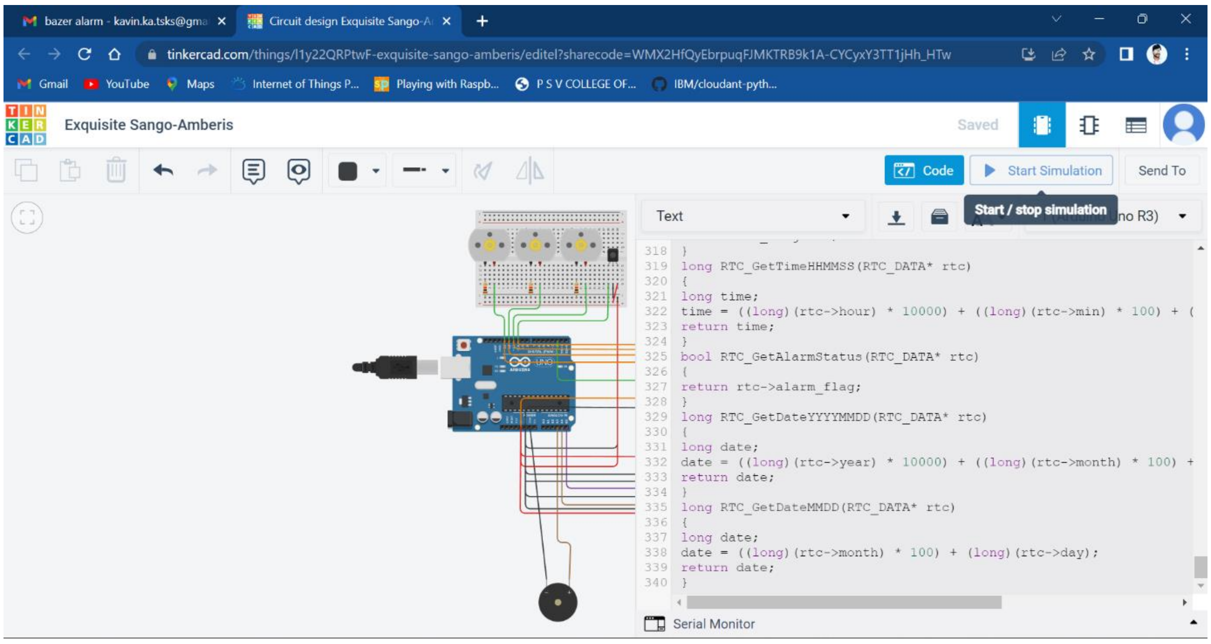
PROGRAM:

```
 #include <LiquidCrystal.h>
#include <stdio.h>
#define mSTATE_1 1
#define mSTATE_2 2
#define mSTATE_3 3
#define mSTATE_4 4
#define mSTATE_5 5
#define mSTATE_6 6
#define mSTATE_7 7
#define mSTATE_8 8
#define mSTATE_9 9
#define ALARM_SWITCH_PIN 2
#define BUZZER_PIN A3
typedef struct
{
bool ALARM_SWITCH;
bool RTC_ALARM;
unsigned long RTC_TIME;
unsigned long RTC_DATE;
char RTC_TIME_C[20];
char RTC_DATE_C[20];
}STATE_VAR;
typedef struct
{
bool alarm_enable;
bool alarm_flag;
int alarm_min;
int alarm_hour;
int tick;
int mls;
int sec;
int min;
int hour;
int month;
int day;
int year;
}RTC_DATA;
void get_input();
void fsm1(STATE_VAR *FSM_VAR);
void printLCDMesgFromStart(LiquidCrystal *lcd, const char* Message, unsigned int row, bool
lcdclear);
void Buzzer(bool ON, int Buzzer_PIN);
float readTemperature(int TempSensor_PIN);
void RTC(RTC_DATA* rtc);
void RTC_Init(RTC_DATA* rtc, int tick);
void RTC_SetTime(RTC_DATA* rtc, int hour, int min, int sec);
void RTC_SetDate(RTC_DATA* rtc, int year, int month, int day);
void RTC_SetAlarm(RTC_DATA* rtc, int hour, int min);
void RTC_EnableAlarm(RTC_DATA* rtc);
void RTC_DisableAlarm(RTC_DATA* rtc);
bool RTC_GetAlarmStatus(RTC_DATA* rtc);
long RTC_GetTimeHHMMSS(RTC_DATA* rtc);
long RTC_GetDateYYYYMMDD(RTC_DATA* rtc);
long RTC_GetDateMMDD(RTC_DATA* rtc);
STATE_VAR FSM1_VAR;
RTC_DATA rtc;
LiquidCrystal lcd(A4, A5, 13, 12, 11, 10);
void setup()
{
Serial.begin(9600);
pinMode(ALARM_SWITCH_PIN, INPUT);
lcd.begin(16, 2);
lcd.setCursor(0, 0);
RTC_Init(&rtc, 100);
RTC_SetTime(&rtc, 11, 00, 0);
RTC_SetDate(&rtc, 2020, 7, 7);
RTC_SetAlarm(&rtc, 11, 36);
RTC_EnableAlarm(&rtc);
}
void get_input()
{
String Date;
String Time;
FSM1_VAR.ALARM_SWITCH = digitalRead(ALARM_SWITCH_PIN);
FSM1_VAR.RTC_ALARM = RTC_GetAlarmStatus(&rtc);
FSM1_VAR.RTC_TIME = RTC_GetTimeHHMMSS(&rtc);
FSM1_VAR.RTC_DATE = RTC_GetDateYYYYMMDD(&rtc);
if(FSM1_VAR.ALARM_SWITCH == 0)
FSM1_VAR.ALARM_SWITCH = true;
else
FSM1_VAR.ALARM_SWITCH = false;
if(rtc.min < 10 && rtc.sec < 10)
Time = String(rtc.hour) + ':' + '0' + String(rtc.min) + ':' + '0' + String(rtc.sec);
else if (rtc.min >= 10 && rtc.sec < 10)
Time = String(rtc.hour) + ':' + String(rtc.min) + ':' + '0' + String(rtc.sec);
else if(rtc.min < 10 && rtc.sec >= 10)
Time = String(rtc.hour) + ':' + '0' + String(rtc.min) + ':' + String(rtc.sec);
else
Time = String(rtc.hour) + ':' + String(rtc.min) + ':' + String(rtc.sec);
Date = String(rtc.day) + '/' + String(rtc.month) + '/' + String(rtc.year);
strcpy(FSM1_VAR.RTC_TIME_C, Time.c_str());
strcpy(FSM1_VAR.RTC_DATE_C, Date.c_str());

}
void loop()
{
RTC(&rtc);
get_input();
fsm1(&FSM1_VAR);
delay(100);
}
void fsm1(STATE_VAR *FSM_VAR)
{
static int MACHINE_STATE;
static bool BUZZER;
bool ALARM_SWITCH = FSM_VAR->ALARM_SWITCH;
bool RTC_ALARM = FSM_VAR->RTC_ALARM;
long RTC_TIME = (FSM_VAR->RTC_TIME / 100);
long RTC_DATE = FSM_VAR->RTC_DATE;
Serial.print("DATE (YYYYMMDD): ");
Serial.print(RTC_DATE);
Serial.print(" TIME (HHMM): ");
Serial.println(RTC_TIME);
switch(MACHINE_STATE)
{
case mSTATE_1:
Serial.println("mSTATE_1");
if(!RTC_ALARM){
printLCDMesgFromStart(&lcd, FSM_VAR->RTC_DATE_C, 1, false);
printLCDMesgFromStart(&lcd, FSM_VAR->RTC_TIME_C, 2, false);
MACHINE_STATE = mSTATE_1 ;
}
else if(RTC_ALARM){
printLCDMesgFromStart(&lcd, "WARNING", 1, true);
printLCDMesgFromStart(&lcd, "ALARM!!", 2, false); BUZZER = true; MACHINE_STATE =
mSTATE_2;
}
else
{}
break;
case mSTATE_2:
Serial. println("mSTATE_2");
if(!ALARM_SWITCH){
MACHINE_STATE = mSTATE_2;
}
else if(ALARM_SWITCH){
RTC_DisableAlarm(&rtc); BUZZER = false; printLCDMesgFromStart(&lcd, FSM_VAR-
>RTC_DATE_C, 1, true);
printLCDMesgFromStart(&lcd, FSM_VAR->RTC_TIME_C, 2, false); MACHINE_STATE =
mSTATE_1;
}
else
{}
break;
default:
BUZZER = false;
MACHINE_STATE = mSTATE_1;
}
Buzzer(BUZZER, BUZZER_PIN);
}
void printLCDMesgFromStart(LiquidCrystal *lcd, const char* Message, unsigned int row, bool
lcdclear)
{
if(lcdclear)
{
lcd->clear();
}
switch(row)
{
case 1:
lcd->setCursor(0, 0);
break;
case 2:
lcd->setCursor(0, 1);
break;
default:
lcd->setCursor(0, 0);
}
if(Message != NULL)
{
lcd->print(Message);
}
}
void Buzzer(bool ON, int Buzzer_PIN)
{
static bool ON_STATE;
if (ON_STATE == false && ON == true)
{
ON_STATE = true;
tone(Buzzer_PIN, 2000);
}
else if (ON == false)
{
ON_STATE = false;
noTone(Buzzer_PIN);
}
else
{}
}
float readTemperature(int TempSensor_PIN)
{
float Temperature;
Temperature = (float) analogRead(TempSensor_PIN);
Temperature = (Temperature * 5.0) / 1024.0;
Temperature = Temperature - 0.5;
Temperature = Temperature * 100;
return Temperature;
}
void RTC(RTC_DATA* rtc)
{
static bool ALARM;
rtc->mls = rtc->mls + rtc->tick;
if(rtc->mls == 1000)
{
rtc->mls = 0;
rtc->sec++;
}
if(rtc->sec >= 60)
{
rtc->sec = 0;
rtc->min++;
}
if(rtc->min >= 60)
{
rtc->sec = 0;
rtc->min = 0;
rtc->hour++;
}
if(rtc->min < 0)
{
rtc->sec = 0;
rtc->min = 59;
rtc->hour--;
}
if(rtc->hour >= 24)
{
rtc->sec = 0;
rtc->min = 0;
rtc->hour = 0;
rtc->day++;
}
if(rtc->month != 2 && (rtc->month % 2) == 1)
{
if(rtc->day == 32)
{
rtc->day = 1;
rtc->month++;
}
if(rtc->day < 1)
{
rtc->day = 30;
rtc->month--;
}
}
if(rtc->month != 2 && (rtc->month % 2) == 0)
{
if(rtc->day == 31)
{rtc->day = 1;
rtc->month++;
}
if(rtc->day < 1)
{
rtc->day = 31;
rtc->month--;
}
}
if(rtc->month == 2)
{
if(rtc->day == 29)
{
rtc->day = 1;
rtc->month++;
}
}
if(rtc->month == 3)
{
if(rtc->day < 1)
{
rtc->day = 28;
rtc->month--;
}
}
if(rtc->hour < 0)
{
rtc->hour = 23;
}
if(rtc->month > 12)
{
rtc->month = 1;
rtc->day = 1;
rtc->year++;
}
if(rtc->alarm_enable == 1 && rtc->alarm_flag == 0)
{
if(rtc->min == rtc->alarm_min && rtc->hour == rtc->alarm_hour)
{
rtc->alarm_flag = 1;
}
}
}
void RTC_Init(RTC_DATA* rtc, int tick)
{
rtc->tick = tick;
rtc->alarm_flag = 0;
rtc->alarm_enable = 0;
}
void RTC_SetTime(RTC_DATA* rtc, int hour, int min, int sec)
{
rtc->sec = sec;
```

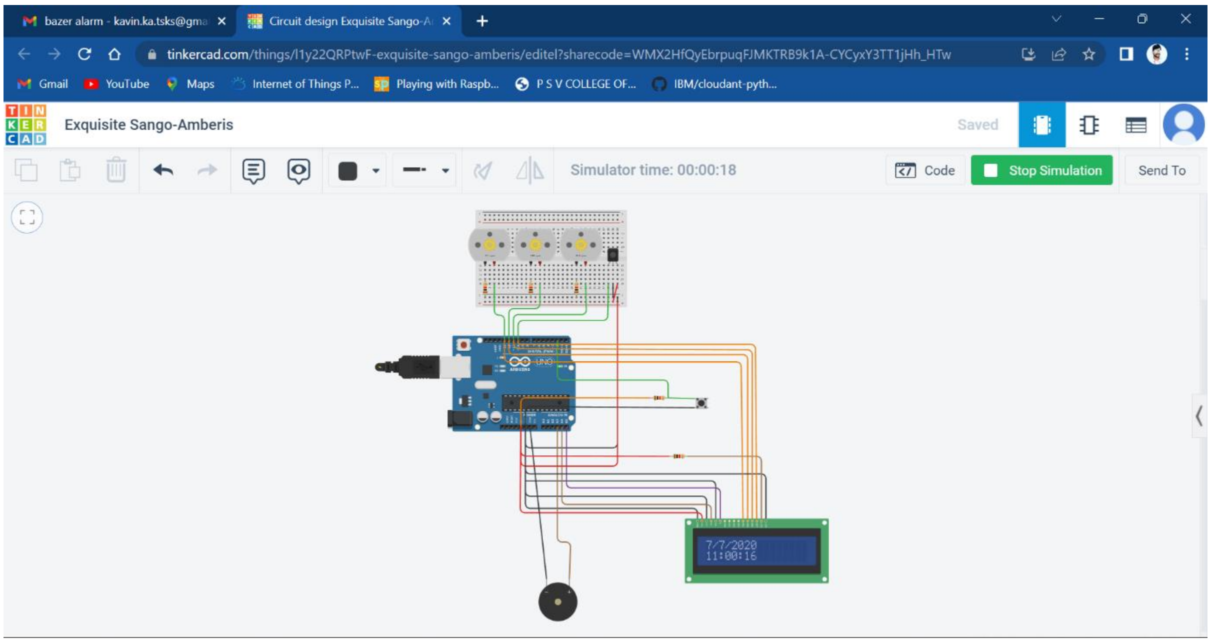
```
rtc->min = min;
rtc->hour = hour;
}
void RTC_SetDate(RTC_DATA* rtc, int year, int month, int day)
{
rtc->day = day;
rtc->month = month;
rtc->year = year;
}
void RTC_SetAlarm(RTC_DATA* rtc, int hour, int min)
{
rtc->alarm_min = min;
rtc->alarm_hour = hour;
}
void RTC_EnableAlarm(RTC_DATA* rtc)
{
rtc->alarm_enable = 1;
rtc->alarm_flag = 0;
}
void RTC_DisableAlarm(RTC_DATA* rtc)
{
rtc->alarm_enable = 0;
rtc->alarm_flag = 0;
}
long RTC_GetTimeHHMMSS(RTC_DATA* rtc)
{
long time;
time = ((long)(rtc->hour) * 10000) + ((long)(rtc->min) * 100) + (long)(rtc->sec);
return time;
}
bool RTC_GetAlarmStatus(RTC_DATA* rtc)
{
return rtc->alarm_flag;
}
long RTC_GetDateYYYYMMDD(RTC_DATA* rtc)
{
long date;
date = ((long)(rtc->year) * 10000) + ((long)(rtc->month) * 100) + (long)(rtc->day);
return date;
}
long RTC_GetDateMMDD(RTC_DATA* rtc)
{
long date;
date = ((long)(rtc->month) * 100) + (long)(rtc->day);
return date;
}
```

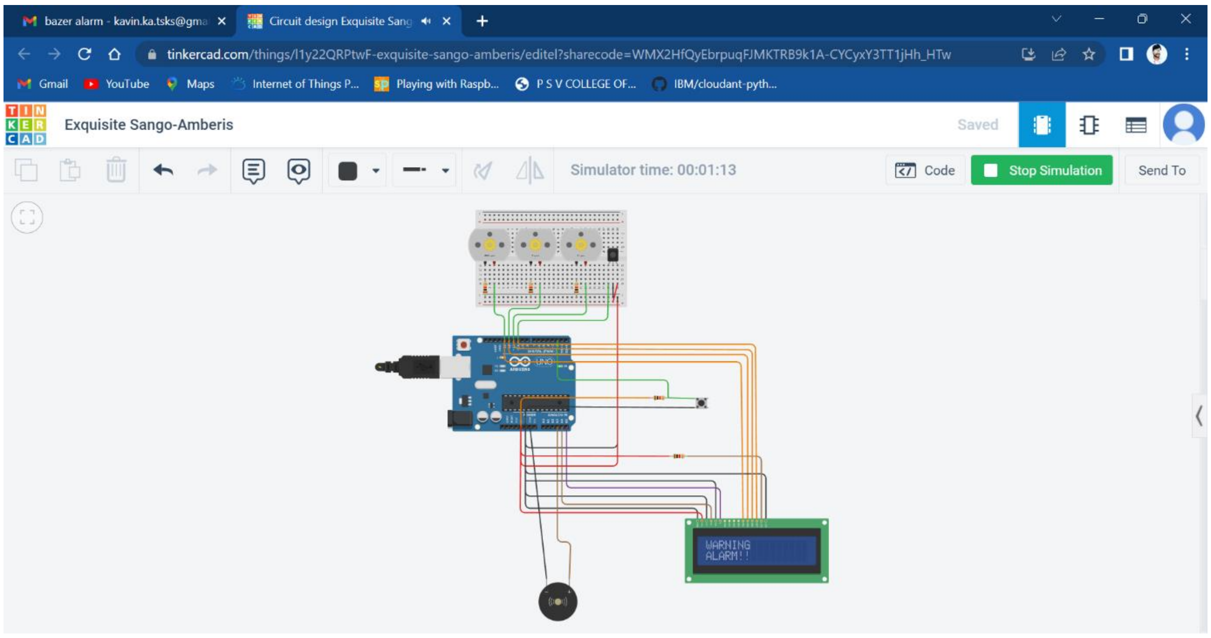
CIRCUIT DIAGRAM:


bazer alarm - kavin.ka.tsks@gma
Circuit design Exquisite Sango-A
tinkercad.com/things/l1y22QRPtwF-exquisite-sango-amberis/editel?sharecode=WMX2HfQyEbrpuqFJMKTRB9k1A-CYCyxY3TT1jHh_HTw
Gmail
YouTube
Maps
Internet of Things P...
Playing with Raspb...
P S V COLLEGE OF...
IBM/cloudant-pyth...
Exquisite Sango-Amberis
Saved
Code
Start Simulation
Send To



```
}
long RTC_GetTimeHHMMSS(RTC_DATA* rtc)
{
long time;
time = ((long)(rtc->hour) * 10000) + ((long)(rtc->min) * 100) + (
return time;
}
bool RTC_GetAlarmStatus(RTC_DATA* rtc)
{
return rtc->alarm_flag;
}
long RTC_GetDateYYYYMMDD(RTC_DATA* rtc)
{
long date;
date = ((long)(rtc->year) * 10000) + ((long)(rtc->month) * 100) +
return date;
}
long RTC_GetDateMMDD(RTC_DATA* rtc)
{
long date;
date = ((long)(rtc->month) * 100) + (long)(rtc->day);
return date;
}
```



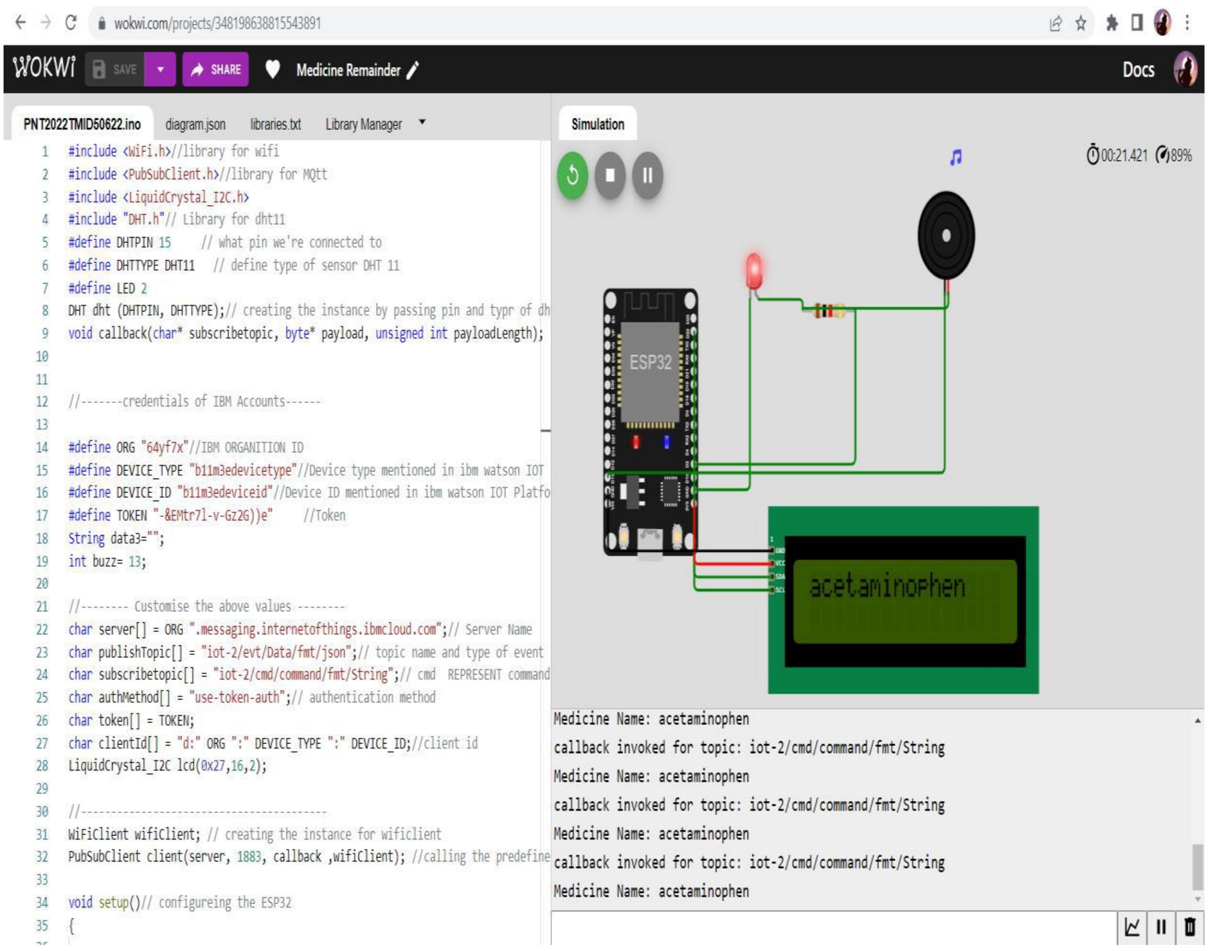


CIRCUIT LINK:

https://www.tinkercad.com/things/l1y22QRPtwF-exquisite-sango-amberis/editel?sharecode=WMX2HfQyEbrpuqFJMKTRB9k1A-CYCyxY3TT1jHh_HTw

Using Wowki:

```
#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 "64yf7x"//IBM ORGANITION ID
#define DEVICE_TYPE "b11m3edevicetype"//Device type mentioned in ibm watson
IOT Platform
#define DEVICE_ID "b11m3edeviceid"//Device ID mentioned in ibm watson IOT
Platform
#define TOKEN "-&EMtr7l-v-Gz2G))e"     //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(0x27,32,2);

//-----------------------------------------
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);
  dht.begin();
  pinMode(buzz, OUTPUT);
  pinMode(LED,OUTPUT);
  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
}

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

}



/*..................................retrieving to
Cloud..............................*/

void PublishData(float temp, float humid) {
  mqttconnect();//function call for connecting to ibm

}
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 = 13; i < payloadLength-2; i++) {
    //Serial.print((char)payload[i]);
    data3 += (char)payload[i];
  }

  Serial.println("Medicine Name: "+ data3);
  if(data3 != "")
  {
    lcd.init();

    lcd.print(data3);
    digitalWrite(LED,HIGH);
    tone(buzz, 100, 1000);
    delay(2000);
    digitalWrite(LED,LOW);
   noTone(buzz);
   delay(1000);

  }

  else
  {
digitalWrite(LED,LOW);

  }
data3="";
}
```

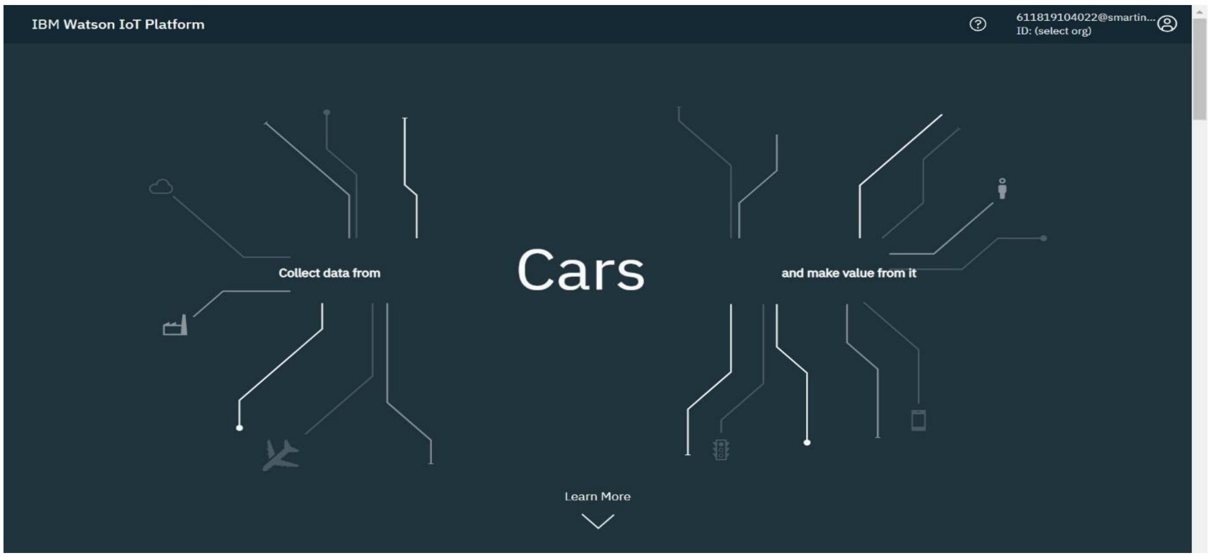


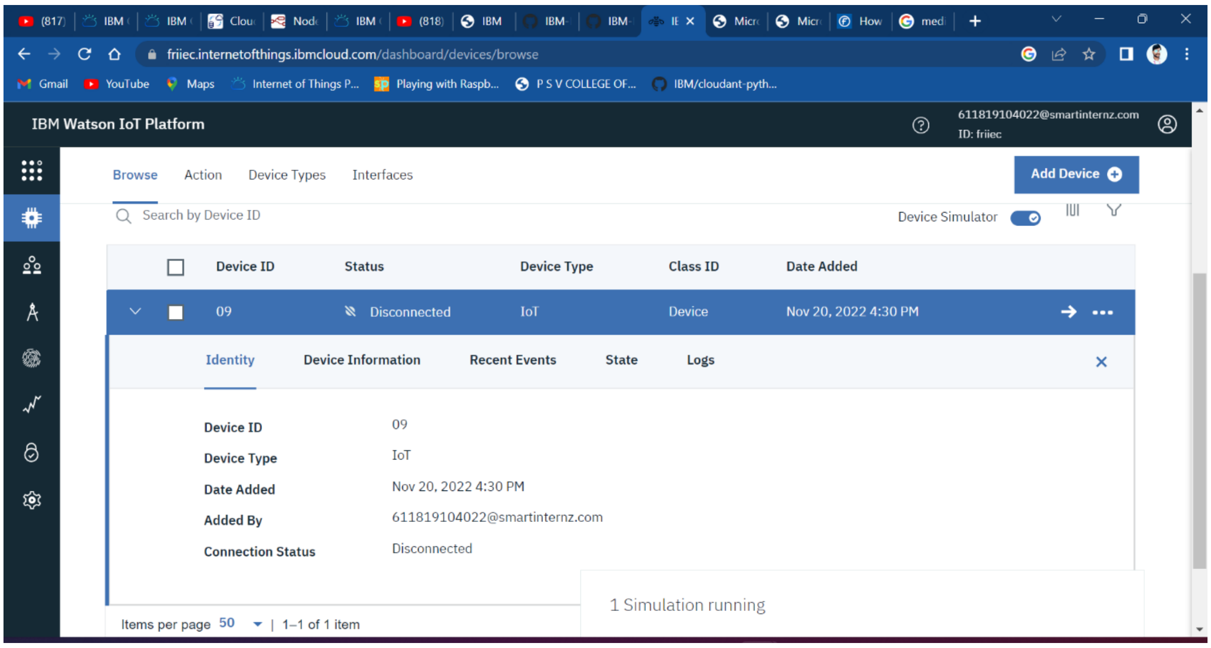
```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include <LiquidCrystal_I2C.h>
#include "DHT.h"// Library for dht11
#define DHTPIN 15     // what pin we're connected to
#define DHTTYPE DHT11   // define type of sensor DHT 11
#define LED 2
DHT dht (DHTPIN, DHTTYPE);//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


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

#define ORG "64yf7x"//IBM ORGANITION ID
#define DEVICE_TYPE "b11m3edevicetype"//Device type mentioned in ibm watson IOT
#define DEVICE_ID "b11m3edeviceid"//Device ID mentioned in ibm watson IOT Platfo
#define TOKEN "-&EMtr7l-v-Gz2G))e"     //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
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd  REPRESENT command
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 predefine

void setup()// configureing the ESP32
{
```

```
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
```

- software (create device in the iot watson platform, workflow for iot scenarios using local node red).

create device in the iot watson platform:



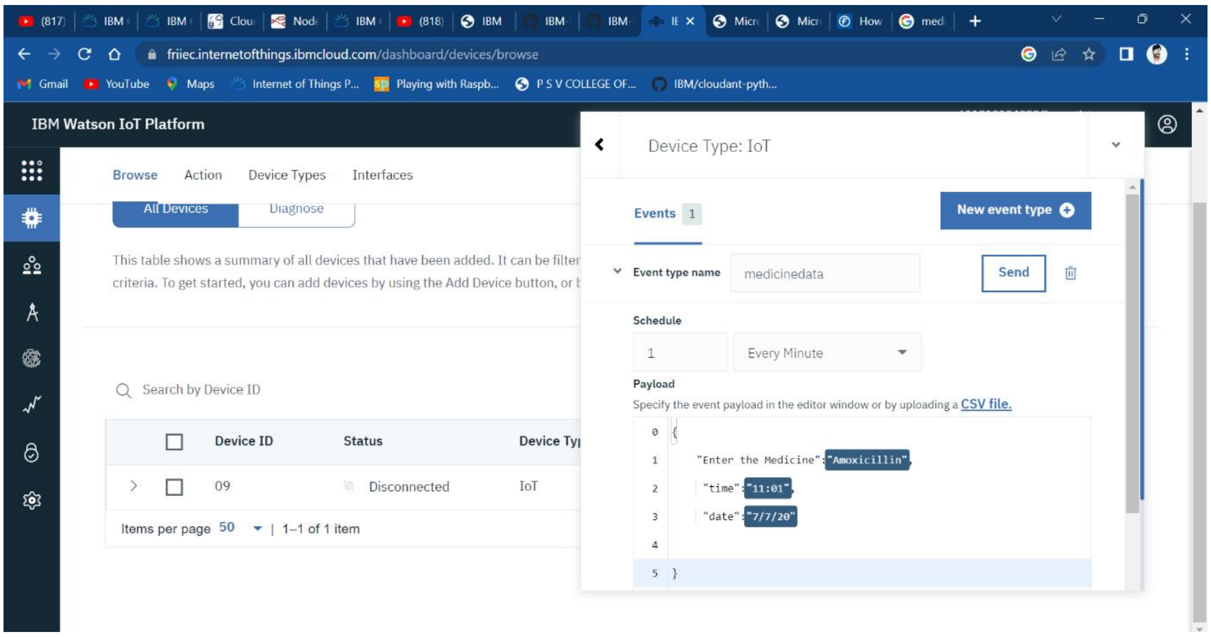


Device ID	Status	Device Type	Class ID	Date Added
09	Disconnected	IoT	Device	Nov 20, 2022 4:30 PM

Device ID	09
Device Type	IoT
Date Added	Nov 20, 2022 4:30 PM
Added By	611819104022@smartinternz.com
Connection Status	Disconnected

Items per page 50 | 1–1 of 1 item

1 Simulation running



This table shows a summary of all devices that have been added. It can be filter criteria. To get started, you can add devices by using the Add Device button, or t

Device ID	Status	Device Typ
09	Disconnected	IoT

Items per page 50 | 1–1 of 1 item

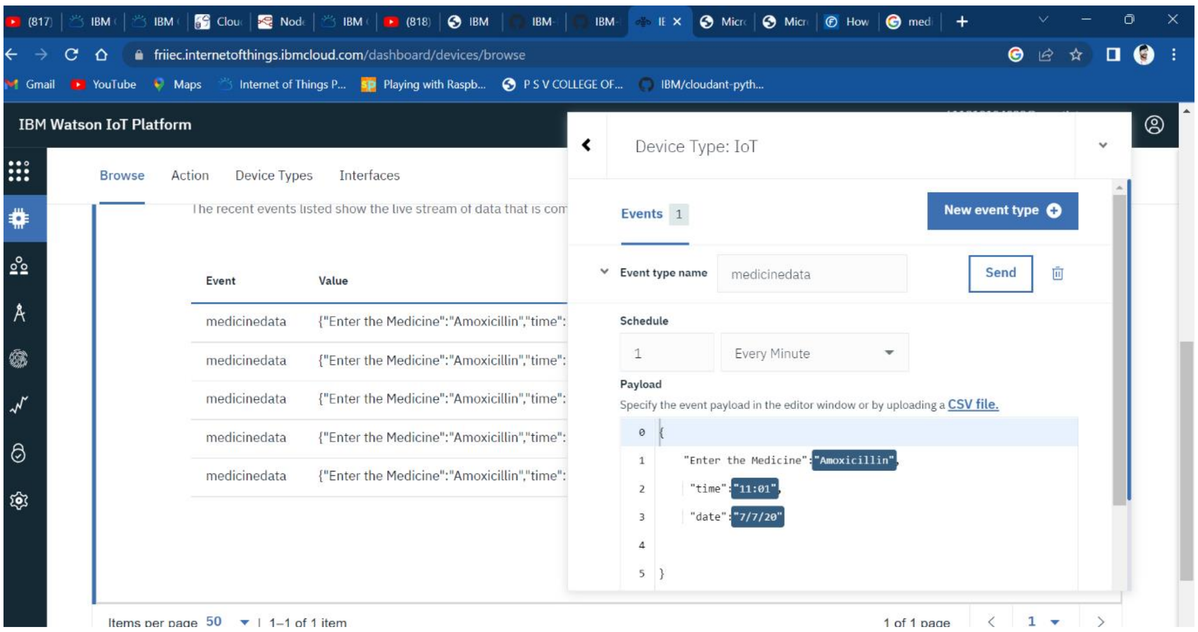
Event type name: medicinedata

Schedule: 1 Every Minute

Payload

Specify the event payload in the editor window or by uploading a CSV file.

```
{
  "Enter the Medicine":"Amoxicillin",
   "time":"11:01",
   "date":"7/7/20"

}
```



friiec.internetofthings.ibmcloud.com/dashboard/devices/browse

Gmail · YouTube · Maps · Internet of Things P... · Playing with Raspb... · P S V COLLEGE OF... · IBM/cloudant-pyth...

IBM Watson IoT Platform

Browse · Action · Device Types · Interfaces

The recent events listed show the live stream of data that is com

Event	Value
medicinedata	{"Enter the Medicine":"Amoxicillin","time":
medicinedata	{"Enter the Medicine":"Amoxicillin","time":
medicinedata	{"Enter the Medicine":"Amoxicillin","time":
medicinedata	{"Enter the Medicine":"Amoxicillin","time":
medicinedata	{"Enter the Medicine":"Amoxicillin","time":

Items per page 50 | 1–1 of 1 item 1 of 1 page 1

Device Type: IoT

Events 1 New event type

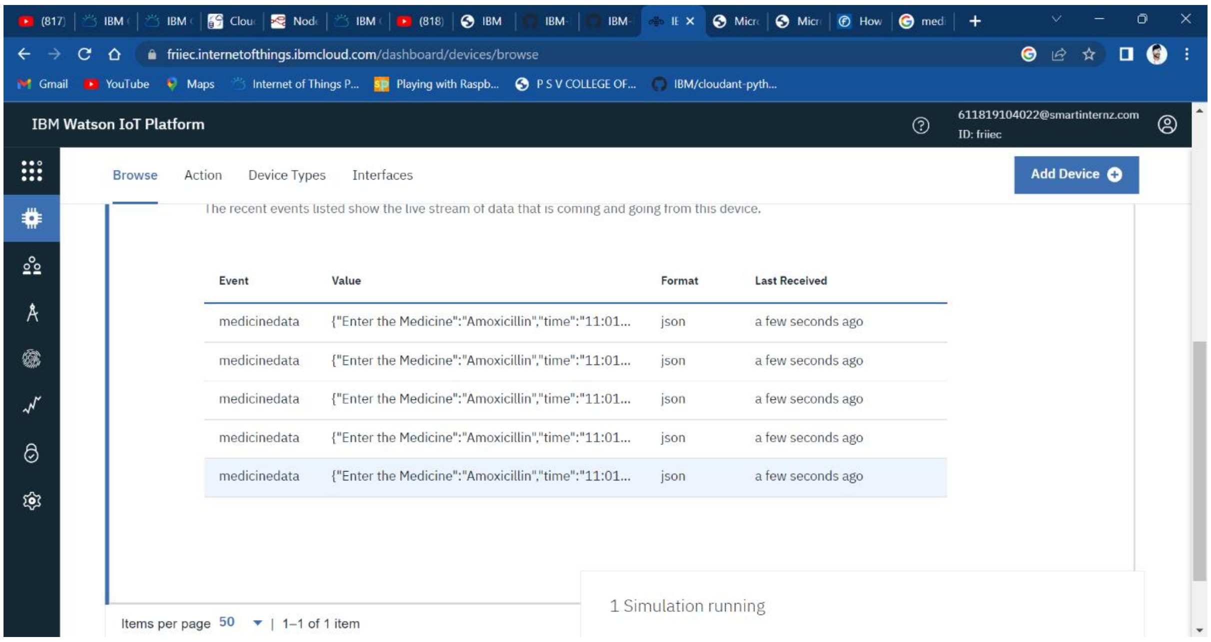
Event type name: medicinedata Send

Schedule: 1 Every Minute

Payload

Specify the event payload in the editor window or by uploading a CSV file.

```
{
   "Enter the Medicine":"Amoxicillin",
    "time":"11:01",
    "date":"7/7/20"

}
```



friiec.internetofthings.ibmcloud.com/dashboard/devices/browse

Gmail · YouTube · Maps · Internet of Things P... · Playing with Raspb... · P S V COLLEGE OF... · IBM/cloudant-pyth...

IBM Watson IoT Platform 611819104022@smartinternz.com ID: friiec

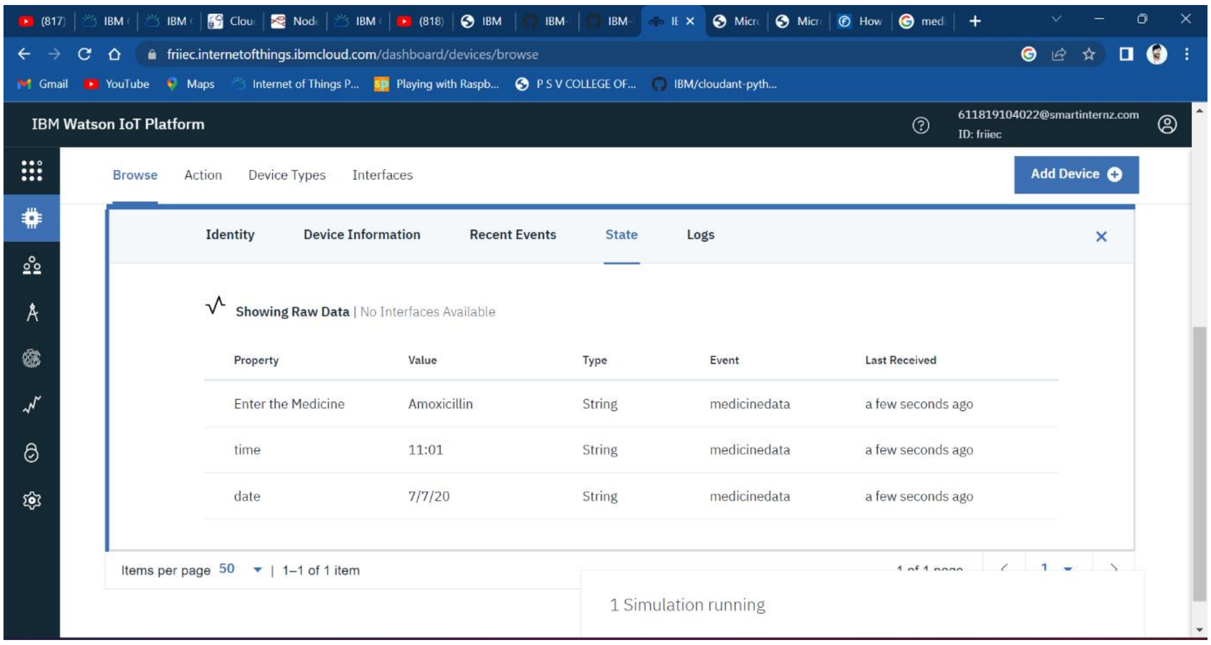
Browse · Action · Device Types · Interfaces Add Device

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

Event	Value	Format	Last Received
medicinedata	{"Enter the Medicine":"Amoxicillin","time":"11:01...	json	a few seconds ago
medicinedata	{"Enter the Medicine":"Amoxicillin","time":"11:01...	json	a few seconds ago
medicinedata	{"Enter the Medicine":"Amoxicillin","time":"11:01...	json	a few seconds ago
medicinedata	{"Enter the Medicine":"Amoxicillin","time":"11:01...	json	a few seconds ago
medicinedata	{"Enter the Medicine":"Amoxicillin","time":"11:01...	json	a few seconds ago

1 Simulation running

Items per page 50 | 1–1 of 1 item



IBM Watson IoT Platform

611819104022@smartinternz.com
ID: friiec

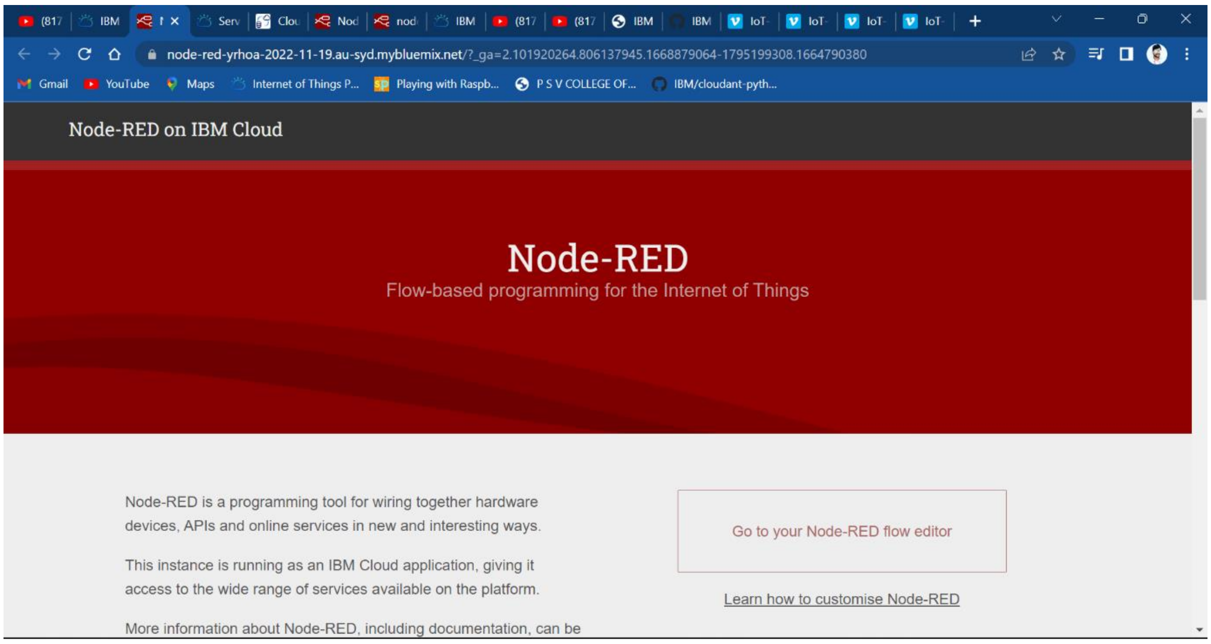
Showing Raw Data | No Interfaces Available

Property	Value	Type	Event	Last Received
Enter the Medicine	Amoxicillin	String	medicinedata	a few seconds ago
time	11:01	String	medicinedata	a few seconds ago
date	7/7/20	String	medicinedata	a few seconds ago

Items per page 50 | 1–1 of 1 item

1 Simulation running

- Thus we created a device in that IoT platform, in this platform we store some medicine details, time and date.
- Then using the node-red, we connect IoT platform.



Node-RED on IBM Cloud

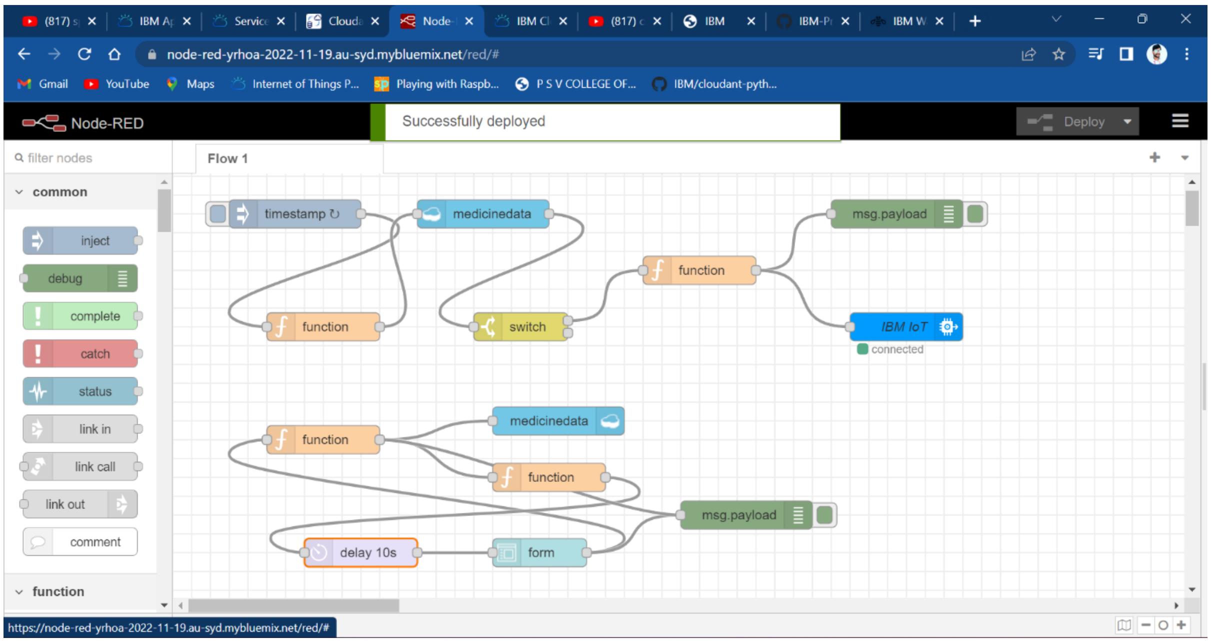
Node-RED
Flow-based programming for the Internet of Things

Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.

This instance is running as an IBM Cloud application, giving it access to the wide range of services available on the platform.

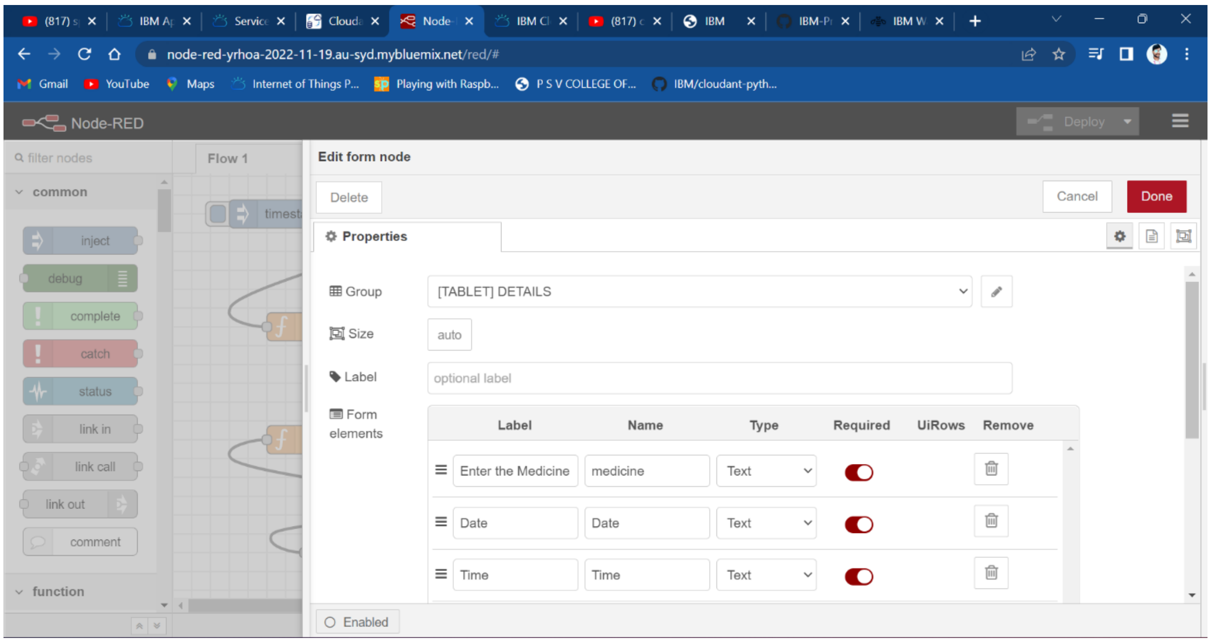
More information about Node-RED, including documentation, can be

Go to your Node-RED flow editor
Learn how to customise Node-RED


node-red-yrhoa-2022-11-19.au-syd.mybluemix.net/red/#

Successfully deployed

https://node-red-yrhoa-2022-11-19.au-syd.mybluemix.net/red/#


node-red-yrhoa-2022-11-19.au-syd.mybluemix.net/red/#

Edit form node

Delete Cancel Done

Properties

Group: [TABLET] DETAILS

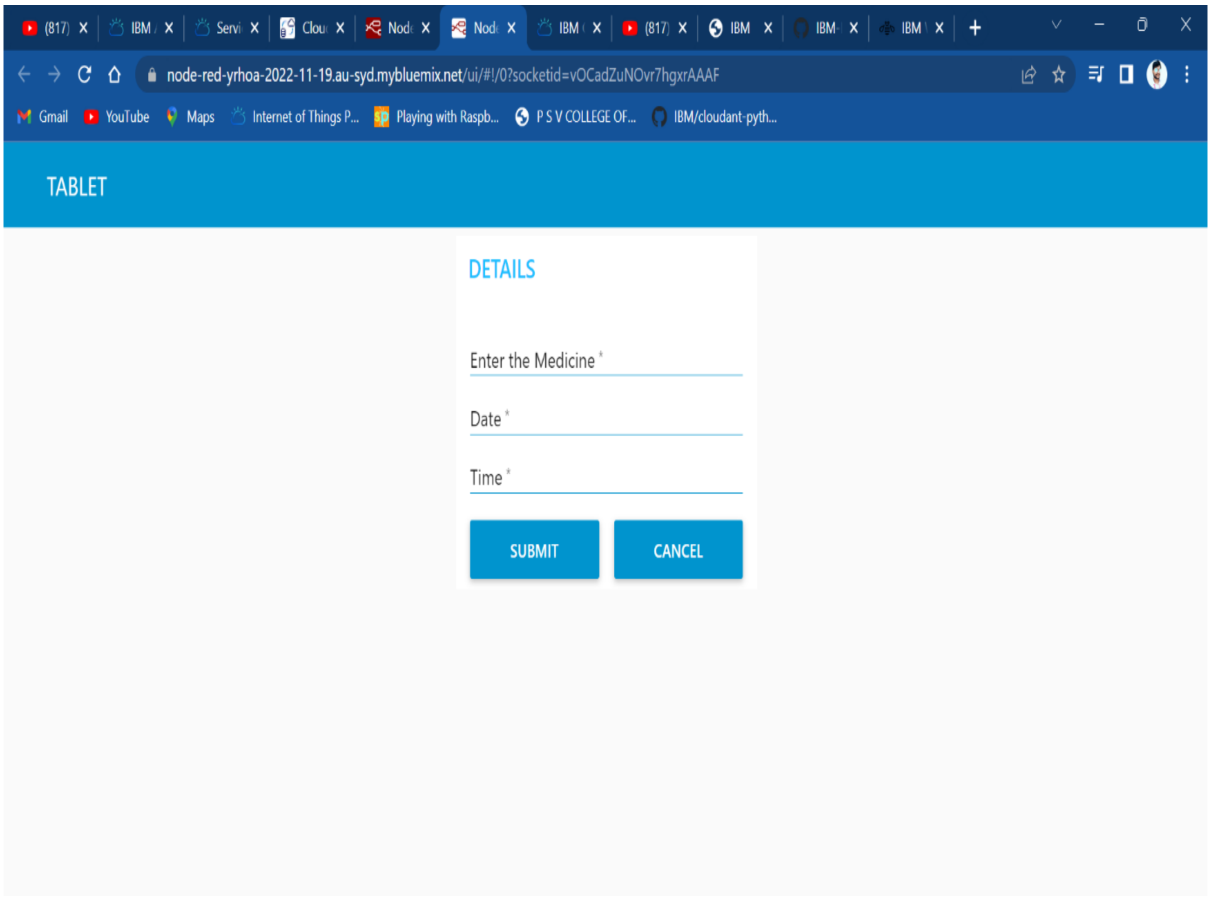
Size: auto

Label: optional label

Form elements

Label	Name	Type	Required	UiRows	Remove
Enter the Medicine	medicine	Text	●		🗑
Date	Date	Text	●		🗑
Time	Time	Text	●		🗑

Enabled



node-red-yrhoa-2022-11-19.au-syd.mybluemix.net/ui/#!/0?socketid=vOCadZuNOvr7hgxrAAAF

TABLET

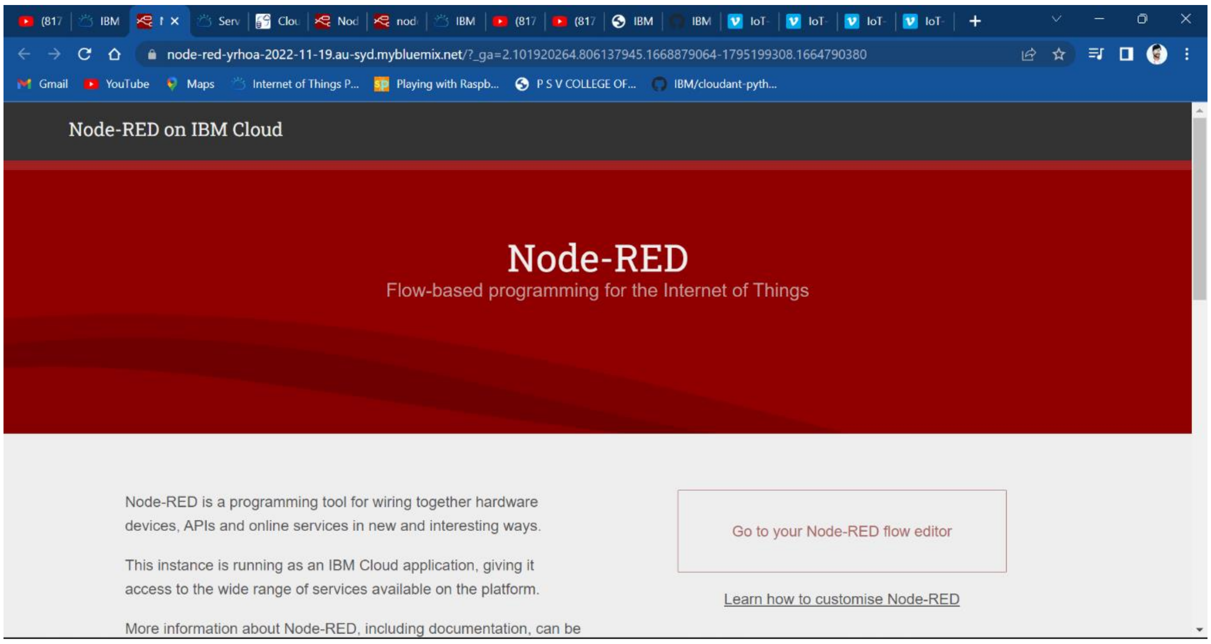
DETAILS

Enter the Medicine *

Date *

Time *

Node-red and Cloud database that store information:



node-red-yrhoa-2022-11-19.au-syd.mybluemix.net/?_ga=2.101920264.806137945.1668879064-1795199308.1664790380

Node-RED on IBM Cloud

Node-RED

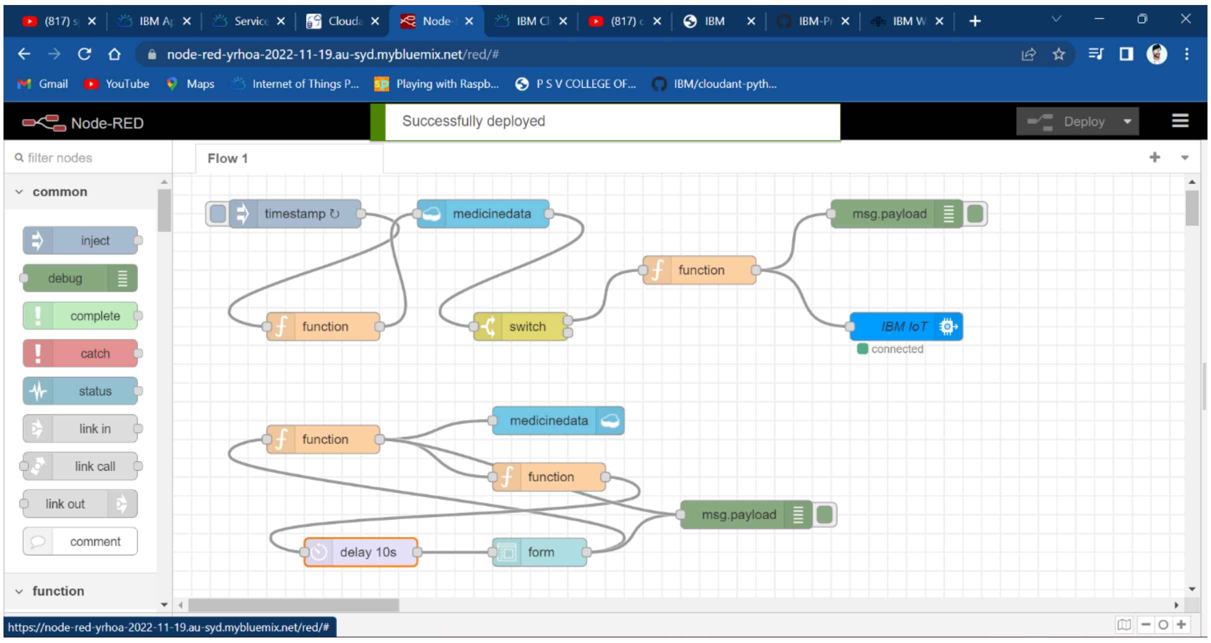
Flow-based programming for the Internet of Things

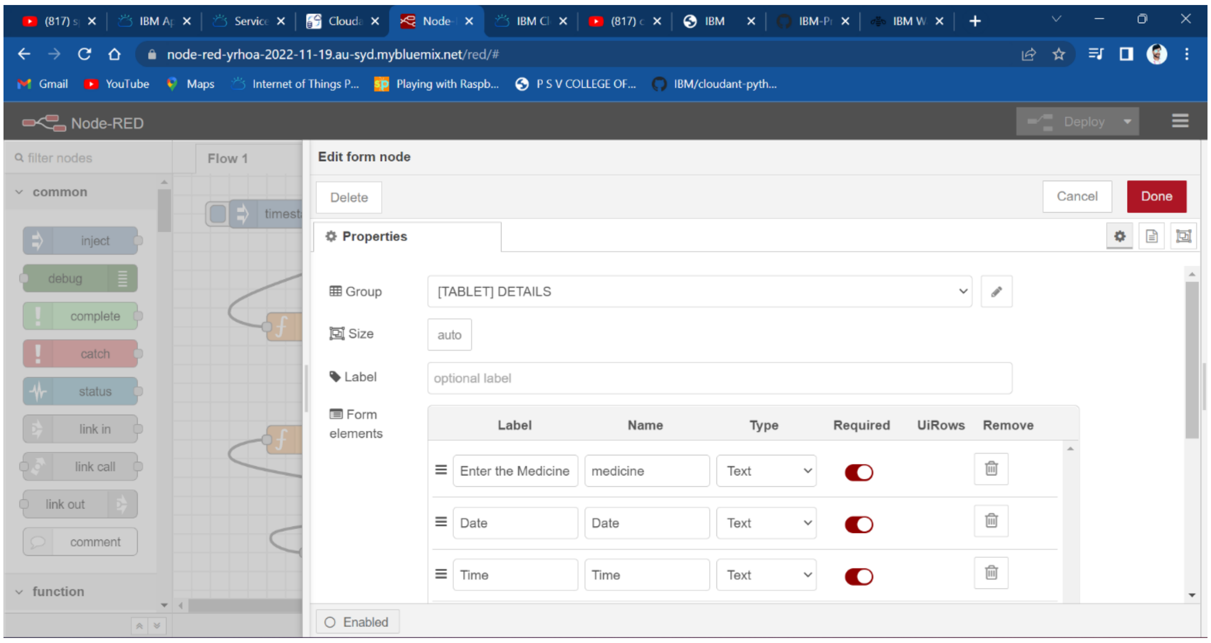
Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.

This instance is running as an IBM Cloud application, giving it access to the wide range of services available on the platform.

More information about Node-RED, including documentation, can be

Go to your Node-RED flow editor
Learn how to customise Node-RED




Edit form node

Delete | Cancel | Done

Properties

Group: [TABLET] DETAILS

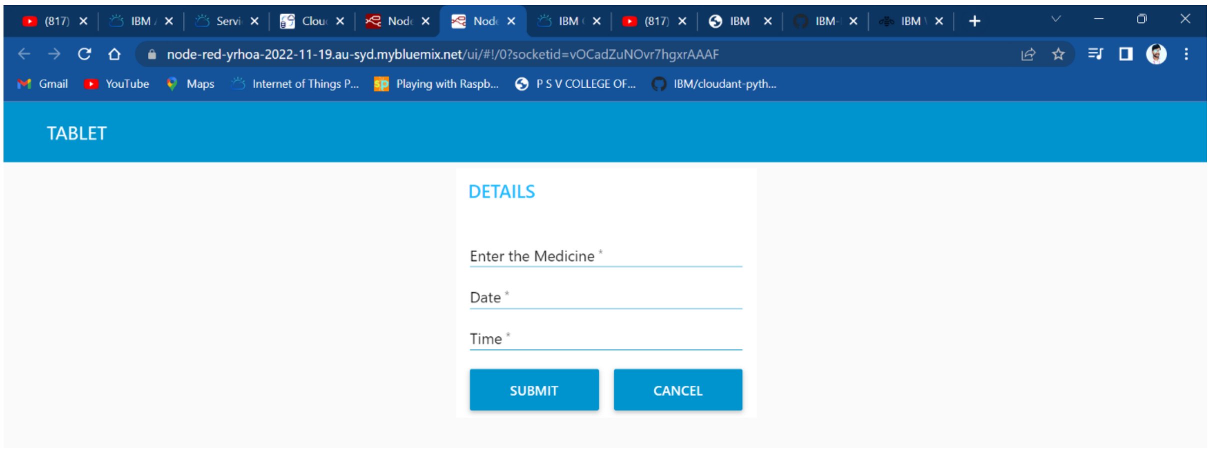
Size: auto

Label: optional label

Form elements

Label	Name	Type	Required	UiRows	Remove
Enter the Medicine	medicine	Text	●		🗑
Date	Date	Text	●		🗑
Time	Time	Text	●		🗑

Enabled


node-red-yrhoa-2022-11-19.au-syd.mybluemix.net/ui/#!/0?socketid=vOCadZuNOvr7hgxrAAAF

TABLET

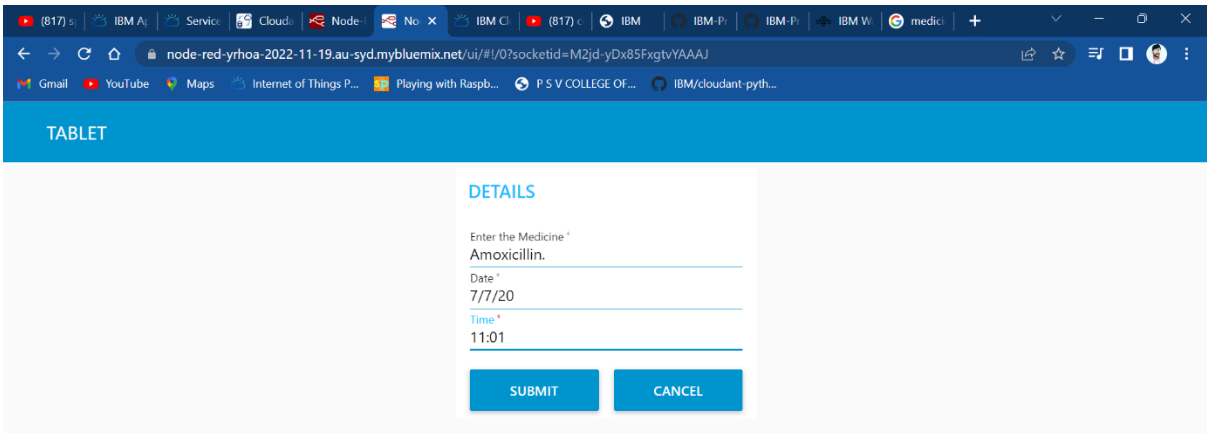
DETAILS

Enter the Medicine *

Date *

Time *

SUBMIT CANCEL


node-red-yrhoa-2022-11-19.au-syd.mybluemix.net/ui/#!/0?socketid=M2jd-yDx85FxgtvYAAAJ

TABLET

DETAILS

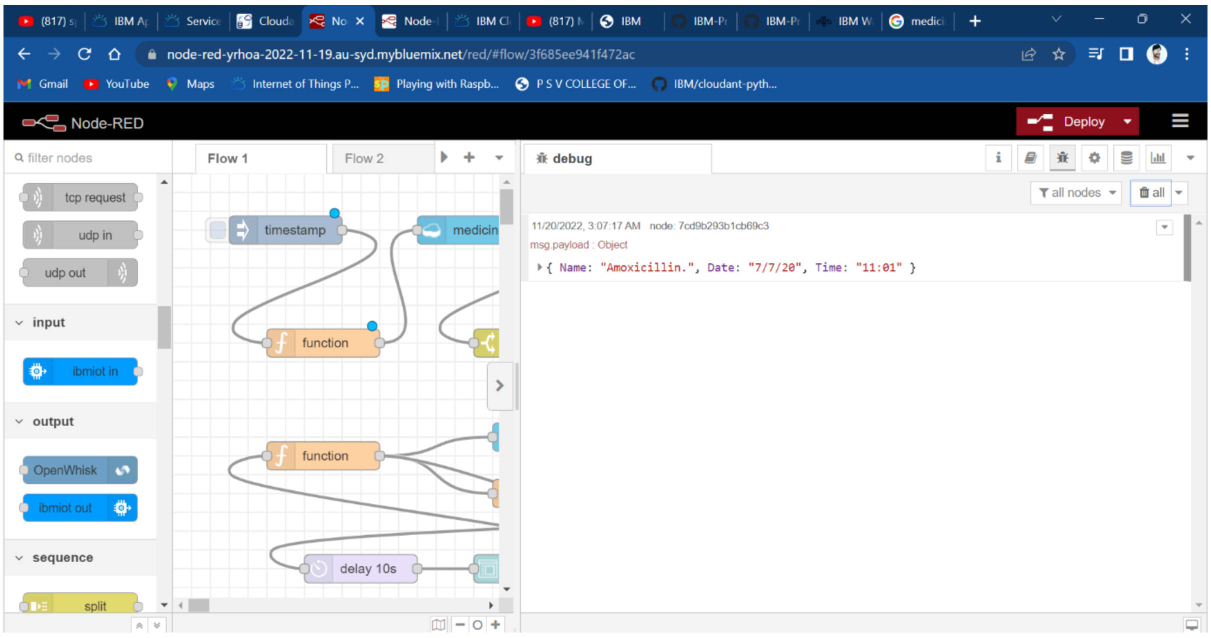
Enter the Medicine *
Amoxicillin.

Date *
7/7/20

Time *
11:01

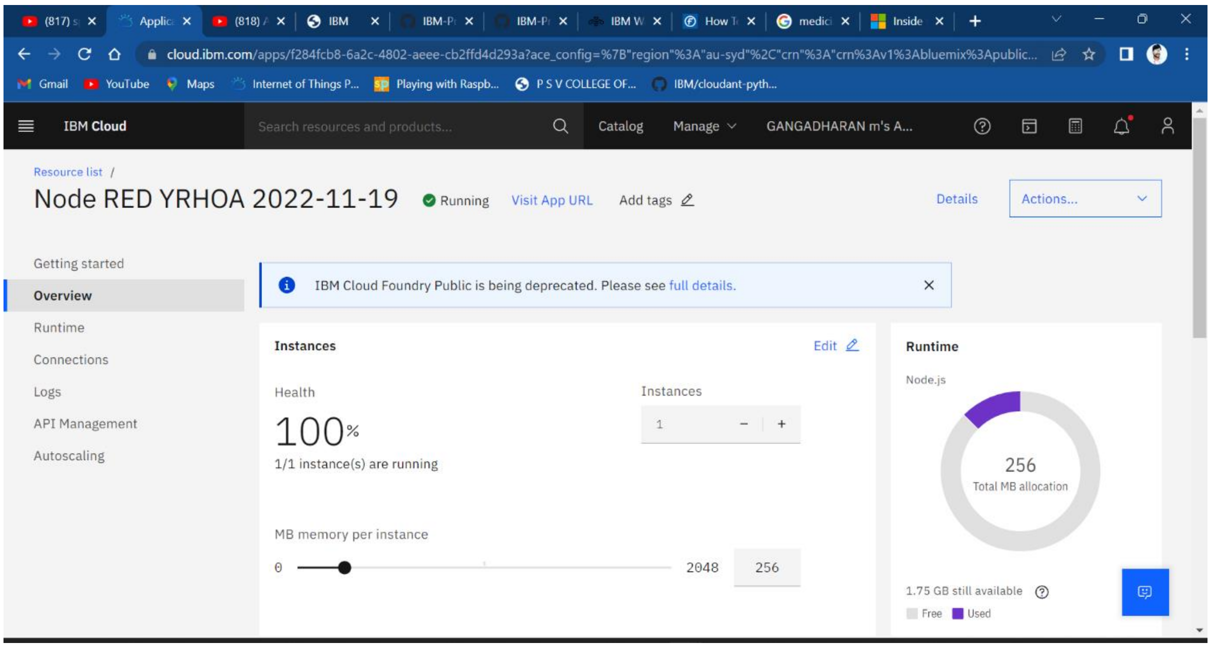
SUBMIT CANCEL

- Here we created a form to store details of the medicine and what time we want to take medicine.


node-red-yrhoa-2022-11-19.au-syd.mybluemix.net/red/#flow/3f685ee941f472ac

```
11/20/2022, 3:07:17 AM  node: 7cd9b293b1cb69c3
msg.payload : Object
▸{ Name: "Amoxicillin.", Date: "7/7/20", Time: "11:01" }
```

- The information are stored in database.



Node RED YRHOA 2022-11-19 Running Visit App URL Add tags

IBM Cloud Foundry Public is being deprecated. Please see full details.

Instances

Health
100%
1/1 instance(s) are running

Instances
1

MB memory per instance
0 2048 256

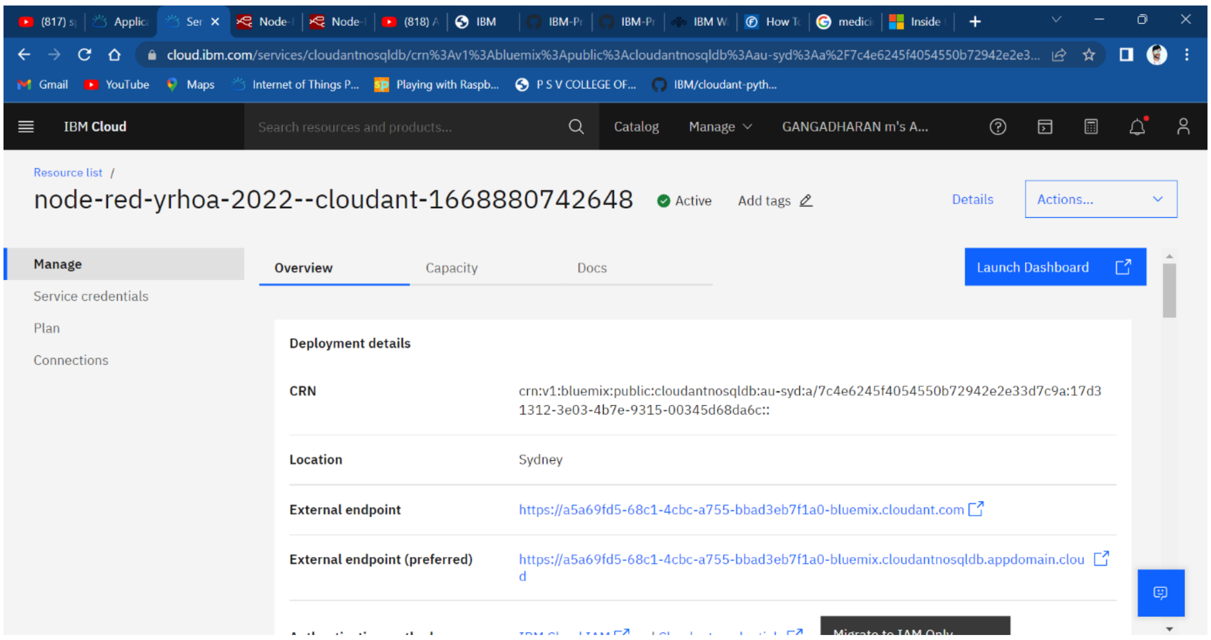
Runtime

Node.js

256
Total MB allocation

1.75 GB still available

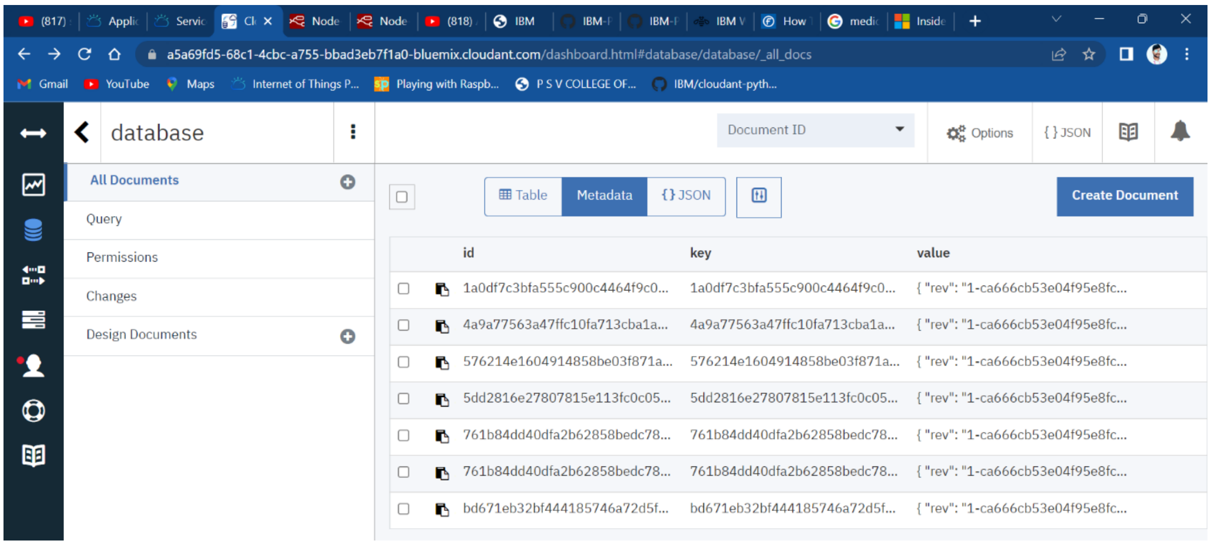
Free Used



node-red-yrhoa-2022--cloudant-1668880742648 Active Add tags

Deployment details

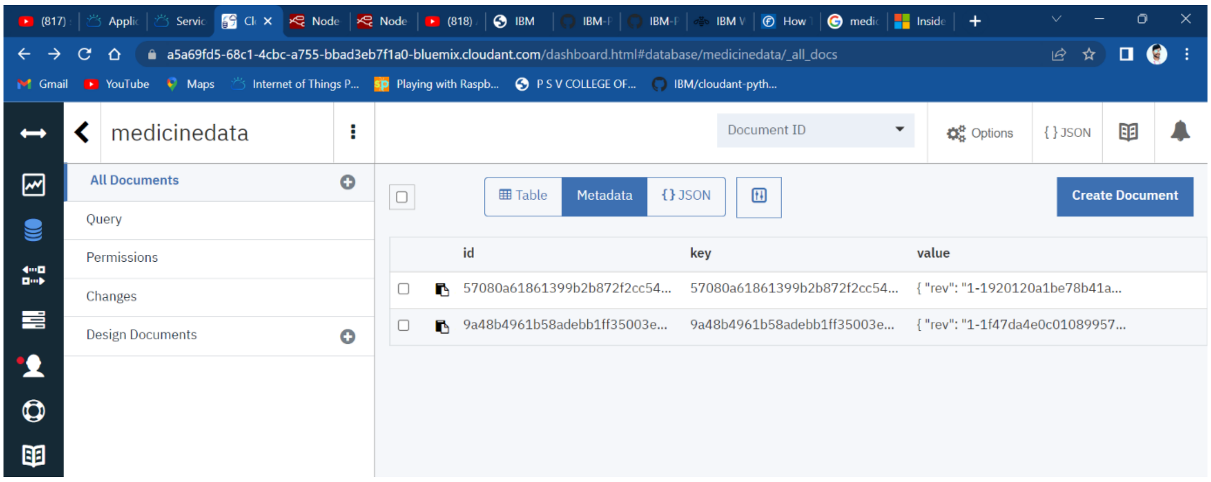
CRN	crn:v1:bluemix:public:cloudantnosqldb:au-syd:a/7c4e6245f4054550b72942e2e33d7c9a:17d31312-3e03-4b7e-9315-00345d68da6c::
Location	Sydney
External endpoint	https://a5a69fd5-68c1-4cbc-a755-bbad3eb7f1a0-bluemix.cloudant.com
External endpoint (preferred)	https://a5a69fd5-68c1-4cbc-a755-bbad3eb7f1a0-bluemix.cloudantnosqldb.appdomain.cloud


a5a69fd5-68c1-4cbc-a755-bbad3eb7f1a0-bluemix.cloudant.com/dashboard.html#database/database/_all_docs

database

All Documents

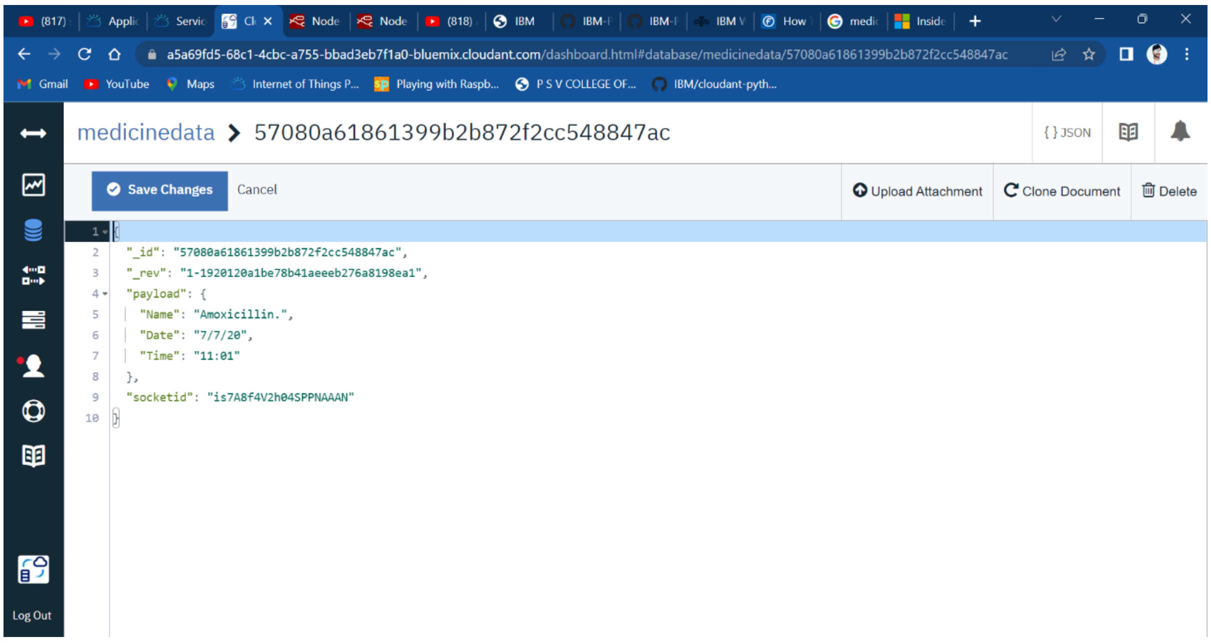
id	key	value
1a0df7c3bfa555c900c4464f9c0...	1a0df7c3bfa555c900c4464f9c0...	{ "rev": "1-ca666cb53e04f95e8fc...
4a9a77563a47ffc10fa713cba1a...	4a9a77563a47ffc10fa713cba1a...	{ "rev": "1-ca666cb53e04f95e8fc...
576214e1604914858be03f871a...	576214e1604914858be03f871a...	{ "rev": "1-ca666cb53e04f95e8fc...
5dd2816e27807815e113fc0c05...	5dd2816e27807815e113fc0c05...	{ "rev": "1-ca666cb53e04f95e8fc...
761b84dd40dfa2b62858bedc78...	761b84dd40dfa2b62858bedc78...	{ "rev": "1-ca666cb53e04f95e8fc...
761b84dd40dfa2b62858bedc78...	761b84dd40dfa2b62858bedc78...	{ "rev": "1-ca666cb53e04f95e8fc...
bd671eb32bf444185746a72d5f...	bd671eb32bf444185746a72d5f...	{ "rev": "1-ca666cb53e04f95e8fc...


a5a69fd5-68c1-4cbc-a755-bbad3eb7f1a0-bluemix.cloudant.com/dashboard.html#database/medicinedata/_all_docs

medicinedata

All Documents

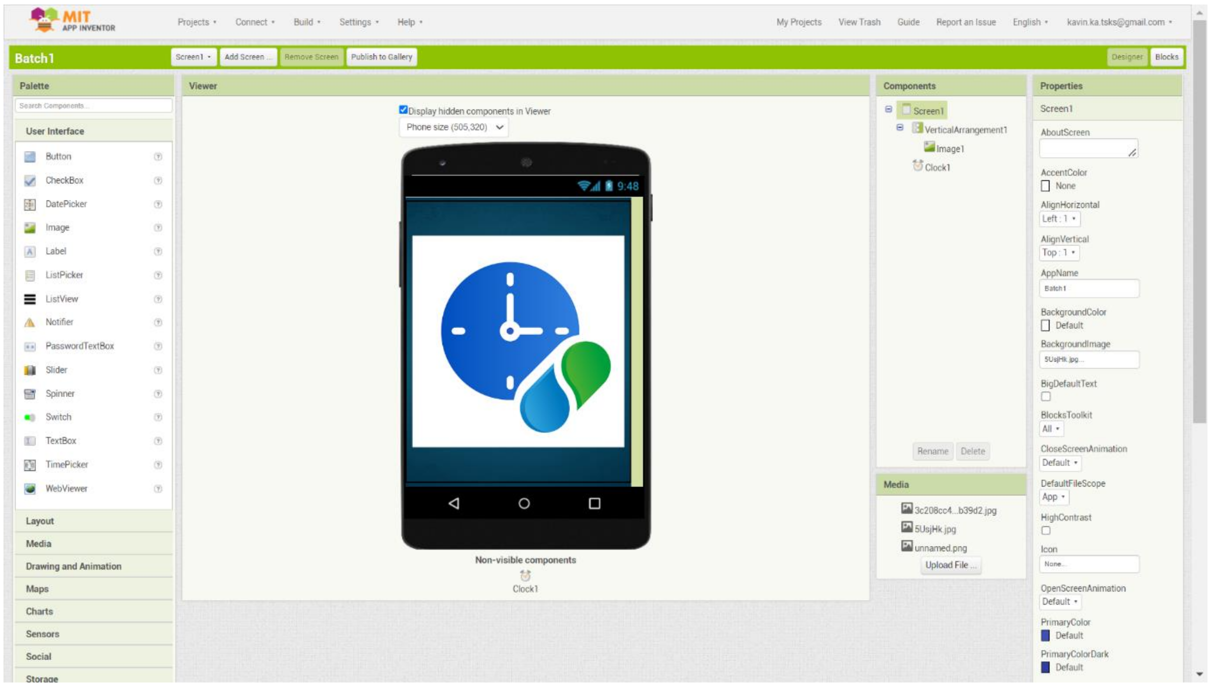
id	key	value
57080a61861399b2b872f2cc54...	57080a61861399b2b872f2cc54...	{ "rev": "1-1920120a1be78b41a...
9a48b4961b58adebb1ff35003e...	9a48b4961b58adebb1ff35003e...	{ "rev": "1-1f47da4e0c01089957...

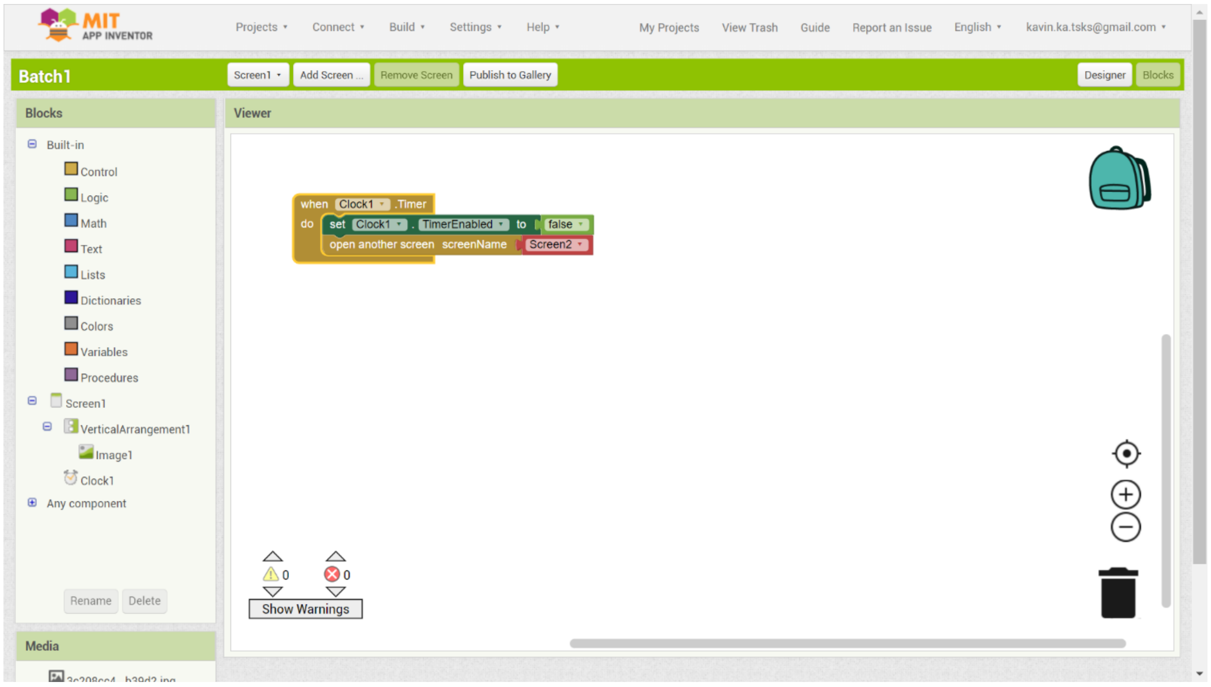

a5a69fd5-68c1-4cbc-a755-bbad3eb7f1a0-bluemix.cloudant.com/dashboard.html#database/medicinedata/57080a61861399b2b872f2cc548847ac

medicinedata > 57080a61861399b2b872f2cc548847ac

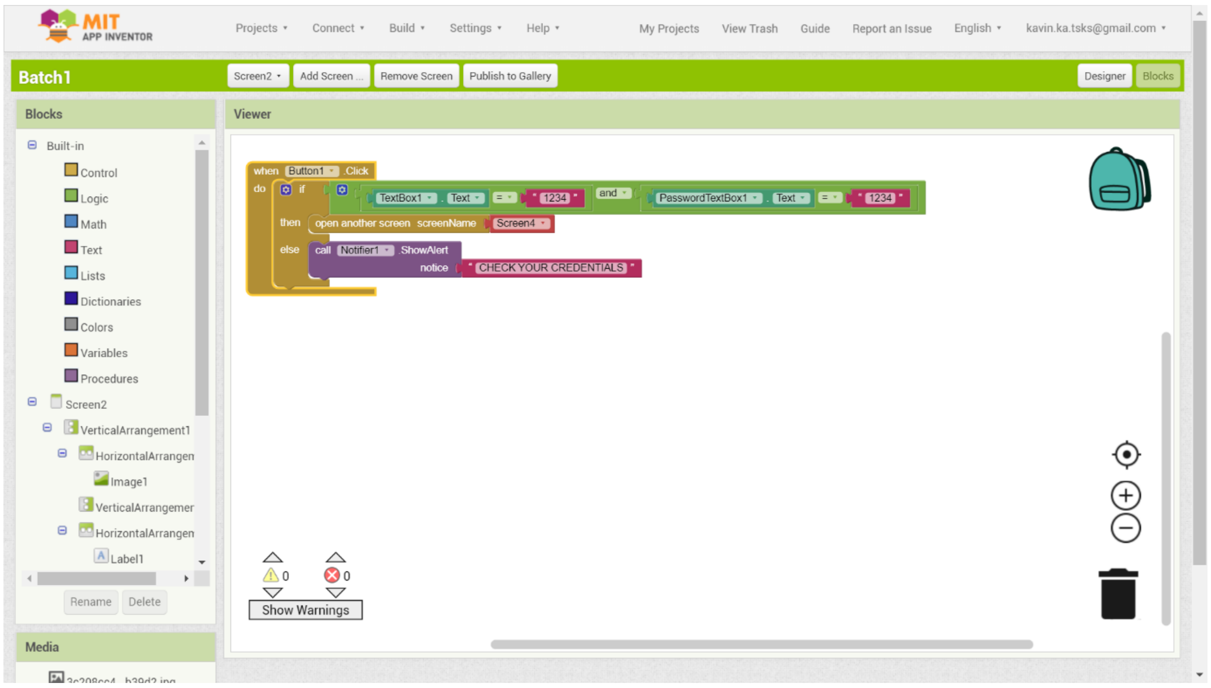
```
{
  "_id": "57080a61861399b2b872f2cc548847ac",
  "_rev": "1-1920120a1be78b41aeeeb276a8198ea1",
  "payload": {
    "Name": "Amoxicillin.",
    "Date": "7/7/20",
    "Time": "11:01"
  },
  "socketid": "is7A8f4V2h04SPPNAAAN"
}
```

Application for User:

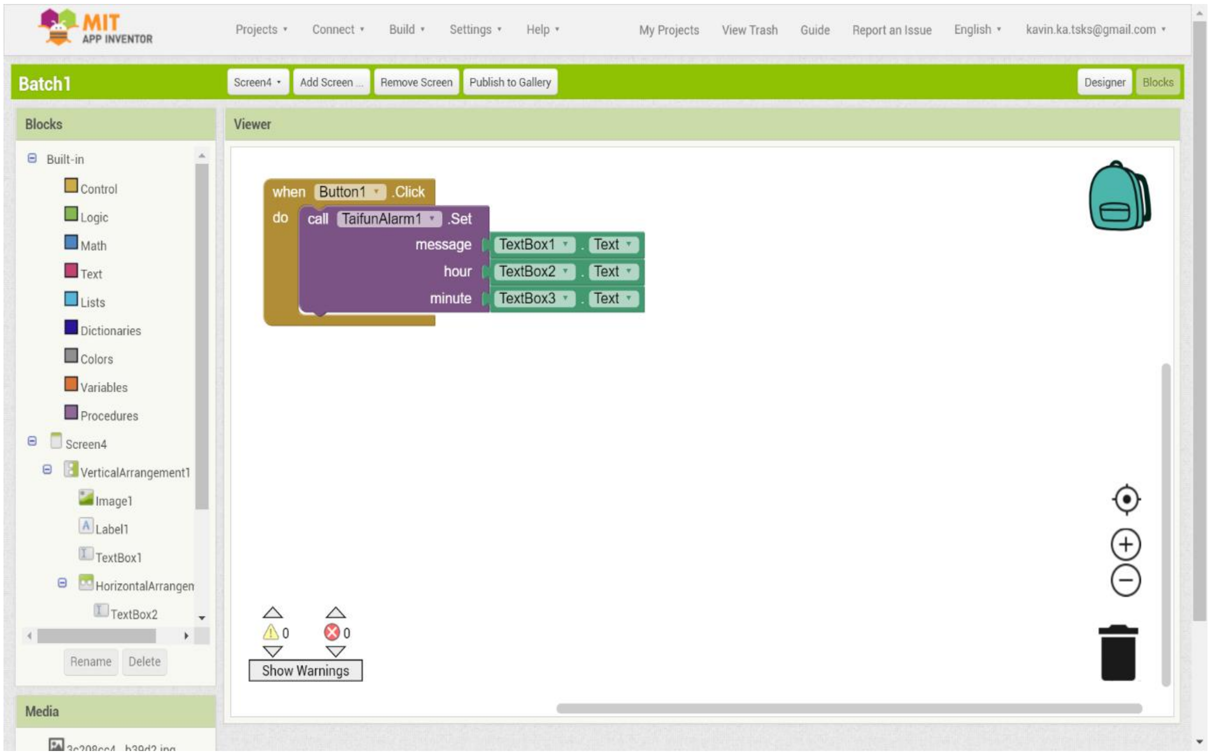




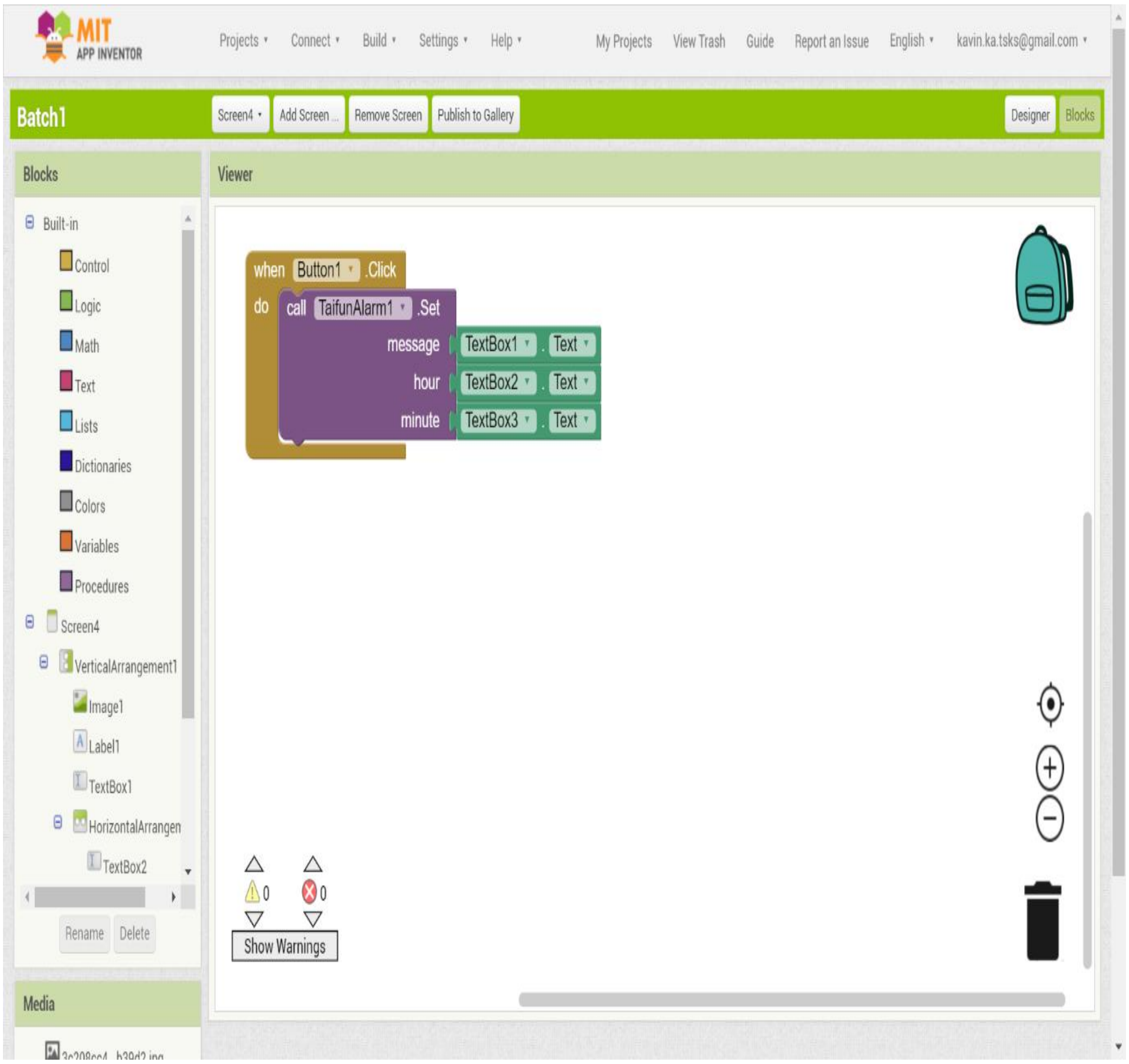
```
when Clock1 .Timer
do  set Clock1 . TimerEnabled to false
    open another screen  screenName  "Screen2"
```



```
when Button1 .Click
do  if  TextBox1 . Text = "1234" and PasswordTextBox1 . Text = "1234"
    then  open another screen  screenName  Screen4
    else  call Notifier1 .ShowAlert
                          notice  "CHECK YOUR CREDENTIALS"
```

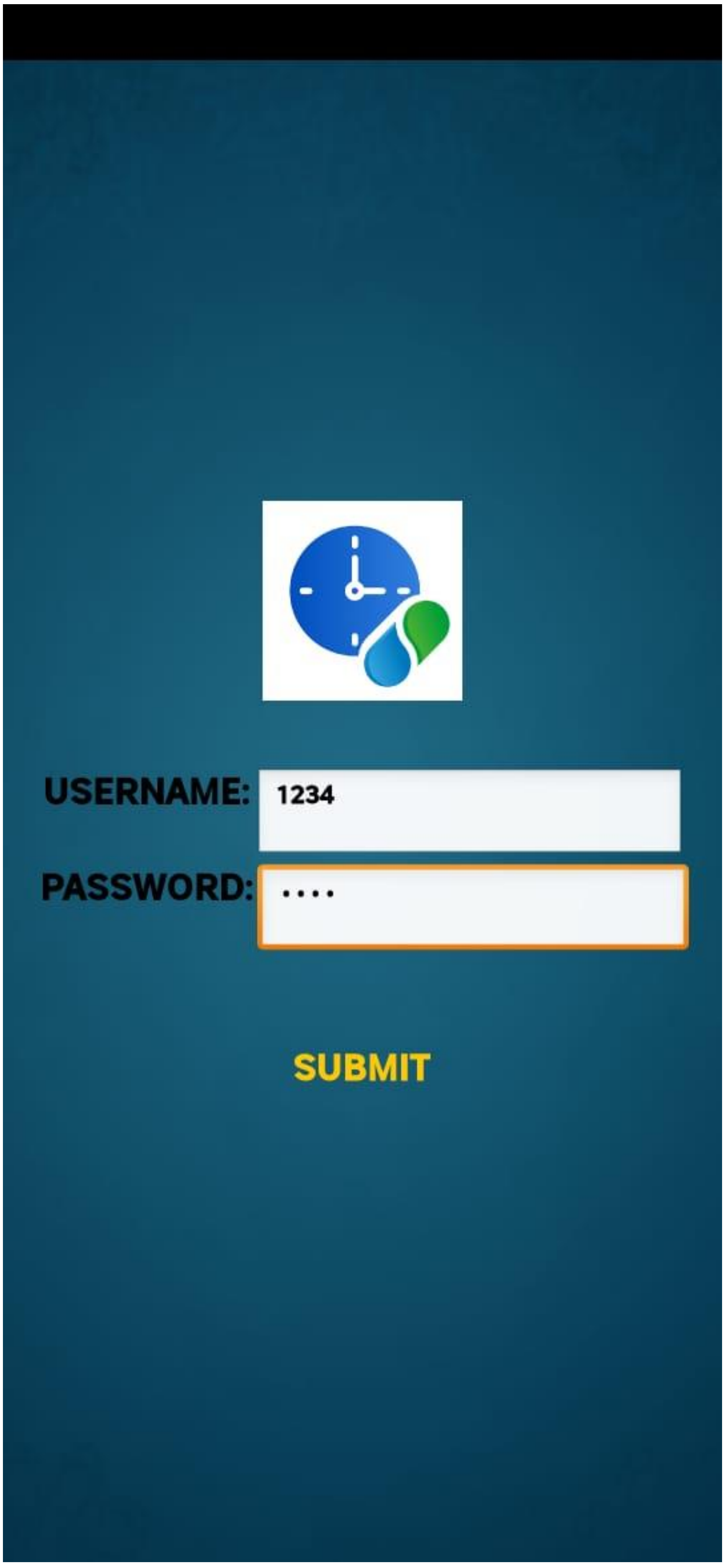


```
when Button1 .Click
do  call TaifunAlarm1 .Set
                  message  TextBox1 . Text
                     hour  TextBox2 . Text
                   minute  TextBox3 . Text
```



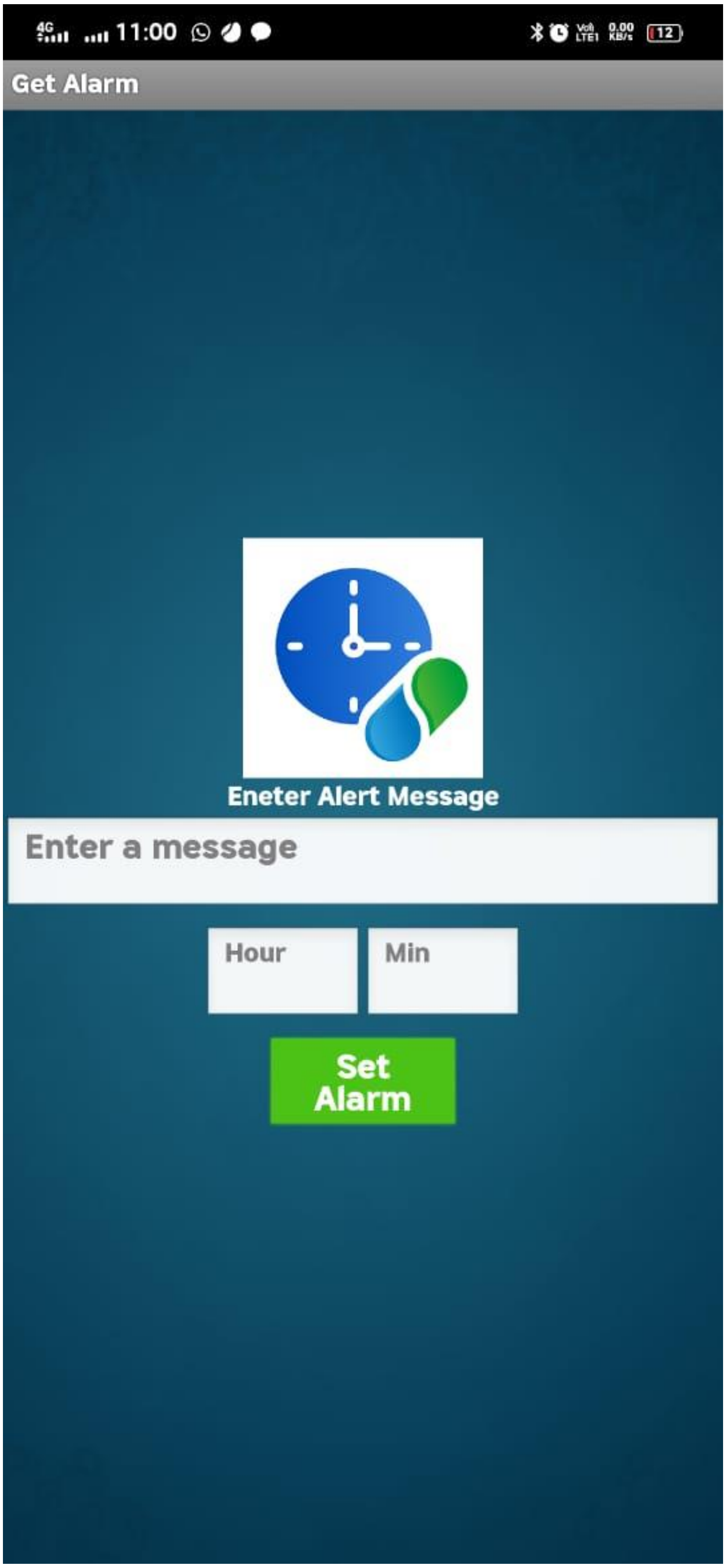
```
when Button1 .Click
do  call TaifunAlarm1 .Set
        message  TextBox1 . Text
        hour     TextBox2 . Text
        minute   TextBox3 . Text
```

4G
11:00
0.00 KB/s
12

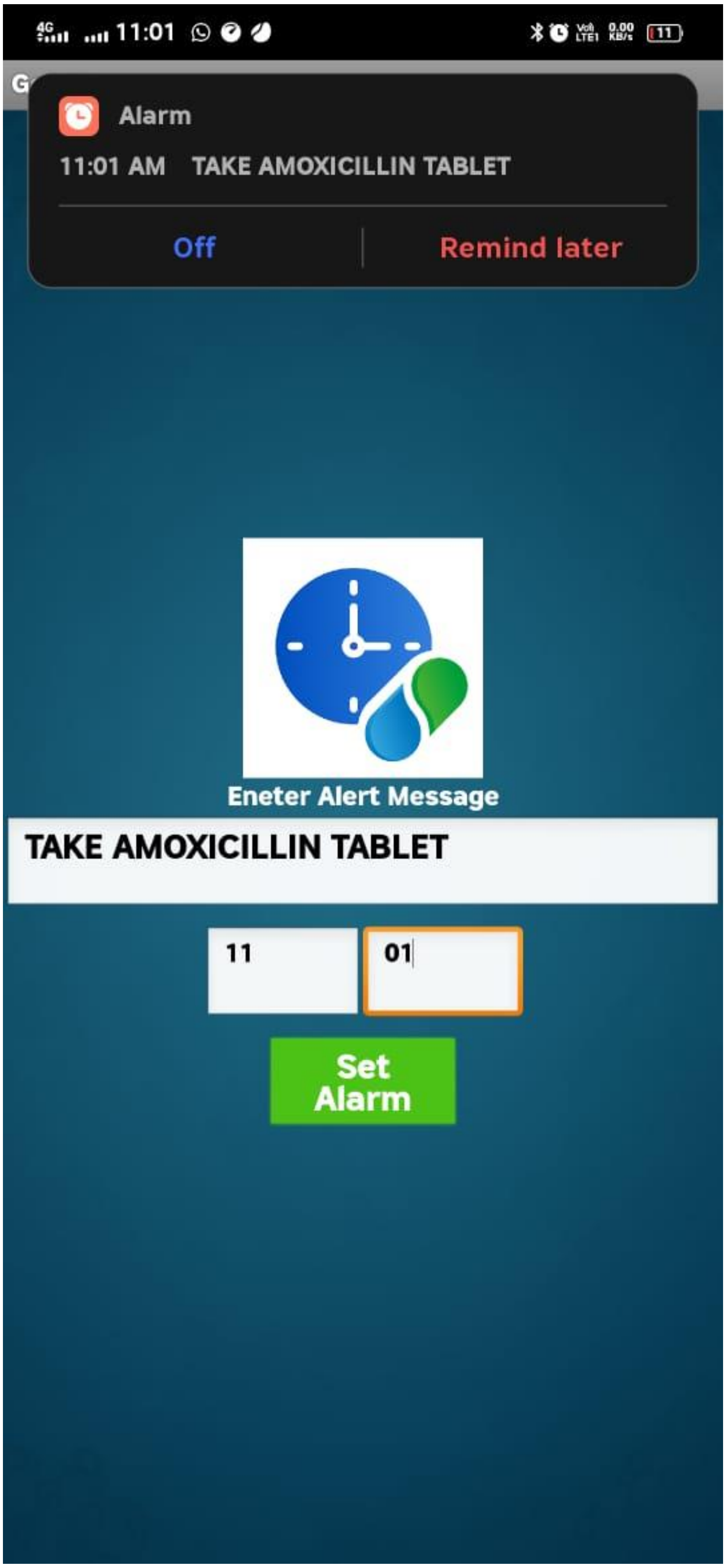


USERNAME: 1234

PASSWORD: ••••



11:00
Get Alarm
Eneter Alert Message
Enter a message
Hour
Min
Set Alarm



Alarm

11:01 AM TAKE AMOXICILLIN TABLET

TAKE AMOXICILLIN TABLET