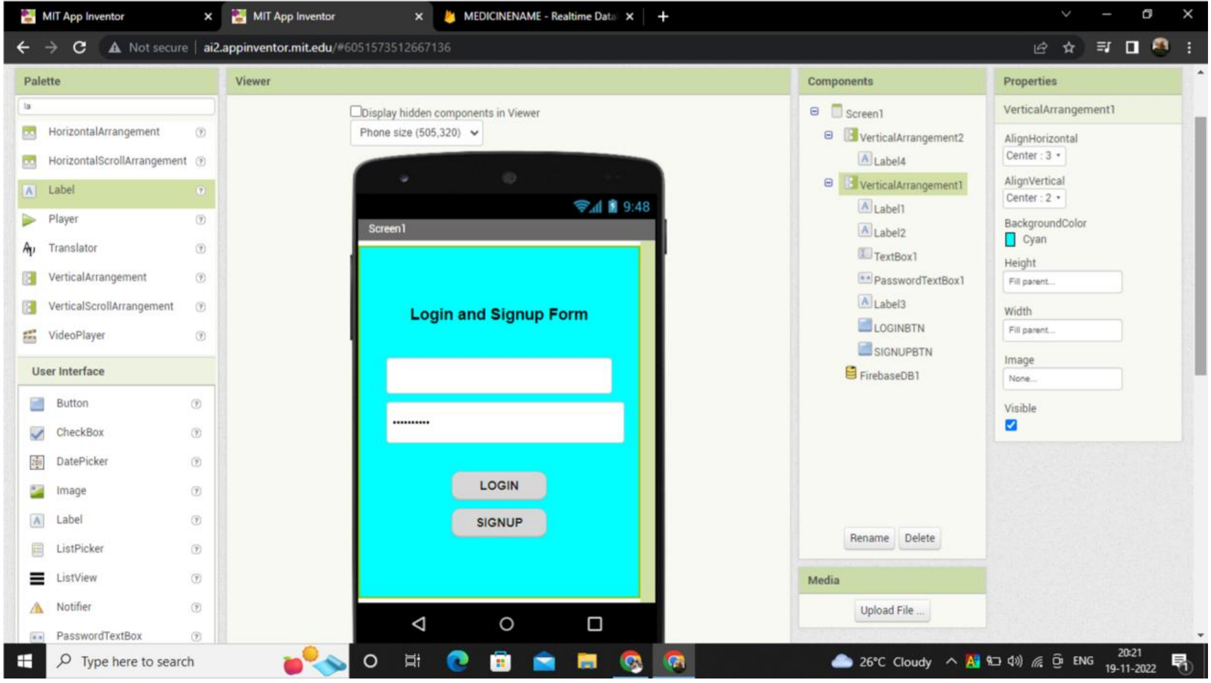
FINAL DELIVERABLES

TEAM ID: PNT2022TIMID29810

PROJECT NAME: PERSONAL ASSISTANCE FOR SENIORS WHO ARE SELF-RELIANT

Using MIT app inventor login and sign up form is created



ai2.appinventor.mit.edu/#6051573512667136

Screen1

Login and Signup Form

LOGIN

SIGNUP

```
Screen1
  VerticalArrangement2
    Label4
  VerticalArrangement1
    Label1
    Label2
    TextBox1
    PasswordTextBox1
    Label3
    LOGINBTN
    SIGNUPBTN
  FirebaseDB1
```

VerticalArrangement1

AlignHorizontal: Center : 3

AlignVertical: Center : 2

BackgroundColor: Cyan

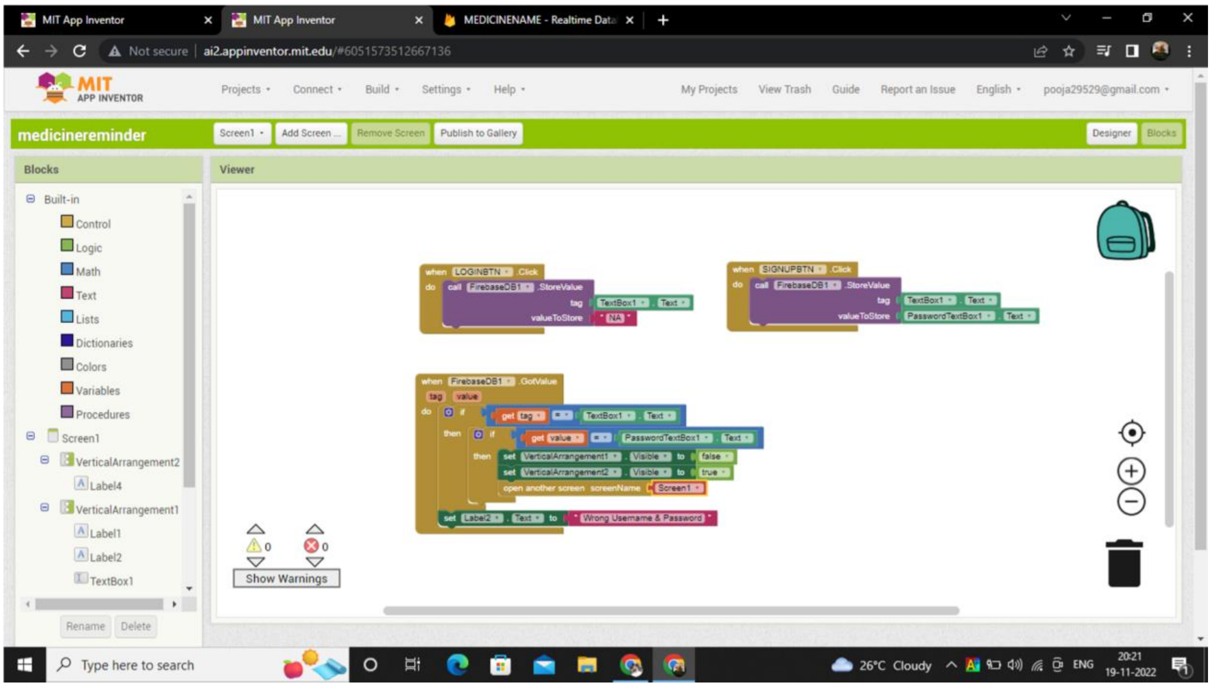
Height: Fill parent...

Width: Fill parent...

Image: None...

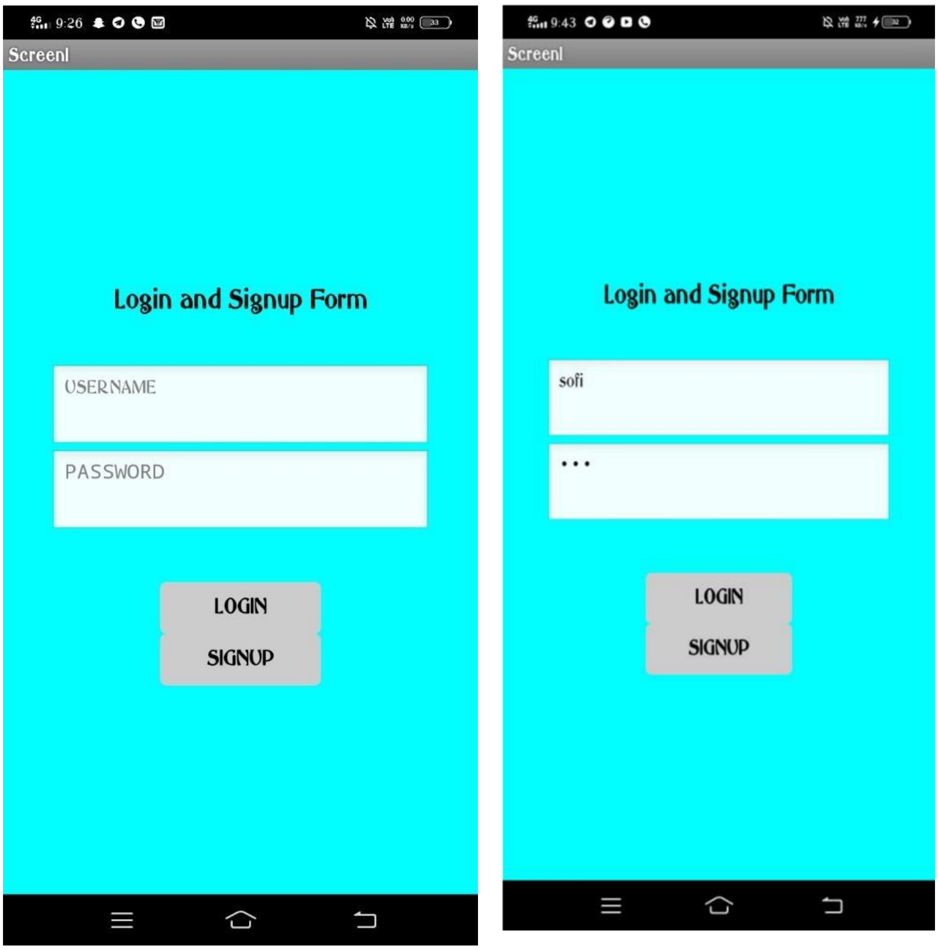
Visible: ☑

Blocks created



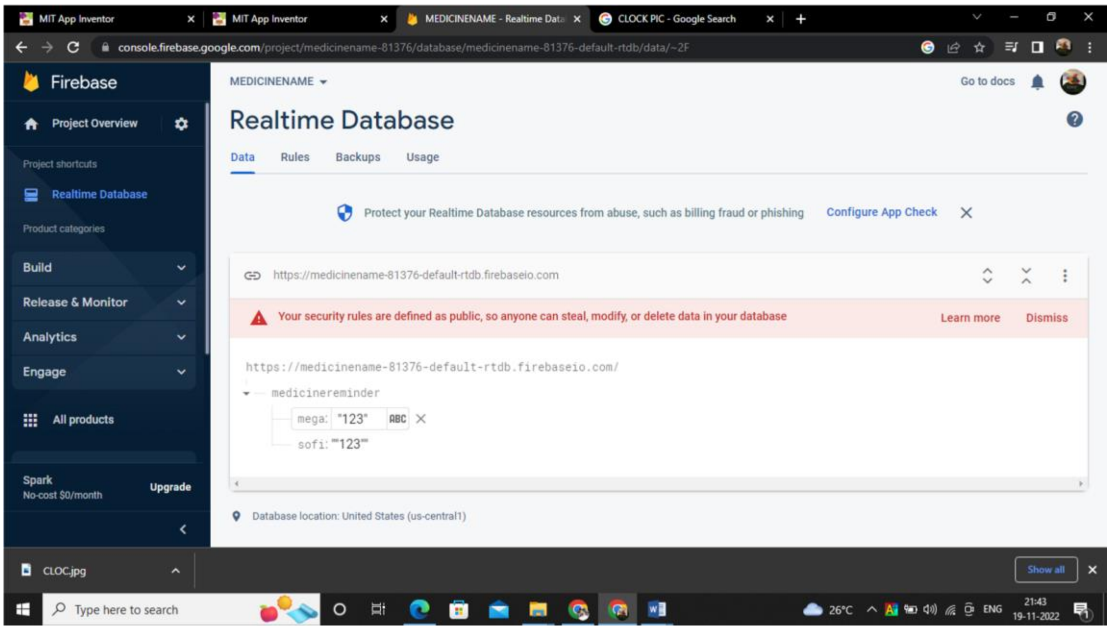
medicinereminder

```
when LOGINBTN .Click
do  call FirebaseDB1 .StoreValue
                 tag  TextBox1 . Text
         valueToStore  " NA "

when SIGNUPBTN .Click
do  call FirebaseDB1 .StoreValue
                 tag  TextBox1 . Text
         valueToStore  PasswordTextBox1 . Text

when FirebaseDB1 .GotValue
  tag  value
do  if  get tag = TextBox1 . Text
    then  if  get value = PasswordTextBox1 . Text
          then  set VerticalArrangement1 . Visible to false
                set VerticalArrangement2 . Visible to true
                open another screen screenName "Screen1"
    set Label2 . Text to " Wrong Username & Password "
```



App login page is created

The details which we give in this page will automatically store it in the firbaase database.



console.firebase.google.com/project/medicinename-81376/database/medicinename-81376-default-rtdb/data/~2F

Realtime Database

Protect your Realtime Database resources from abuse, such as billing fraud or phishing

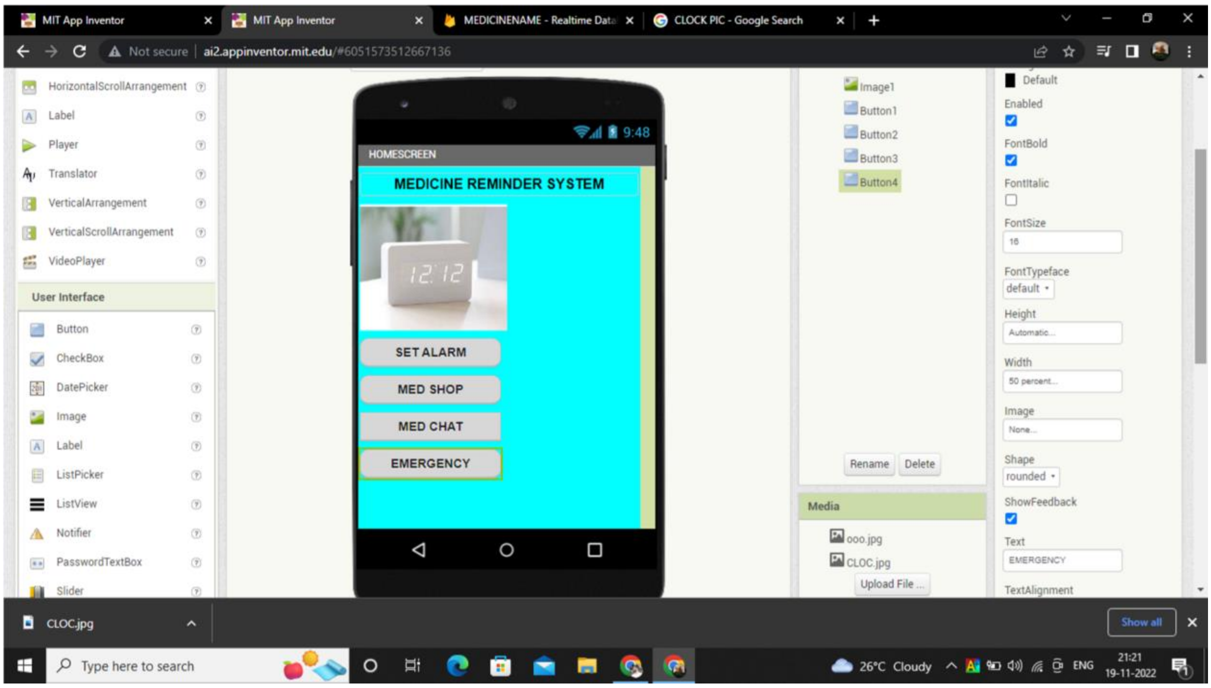
https://medicinename-81376-default-rtdb.firebaseio.com

Your security rules are defined as public, so anyone can steal, modify, or delete data in your database

```
https://medicinename-81376-default-rtdb.firebaseio.com/
▾ — medicinereminder
      mega: "123"
      sofi: ""123""
```

Database location: United States (us-central1)

After successful login, user can be redirected to home screen.



Default

Enabled
☑

FontBold
☑

FontItalic
□

FontSize
16

FontTypeface
default

Height
Automatic...

Width
50 percent...

Image
None...

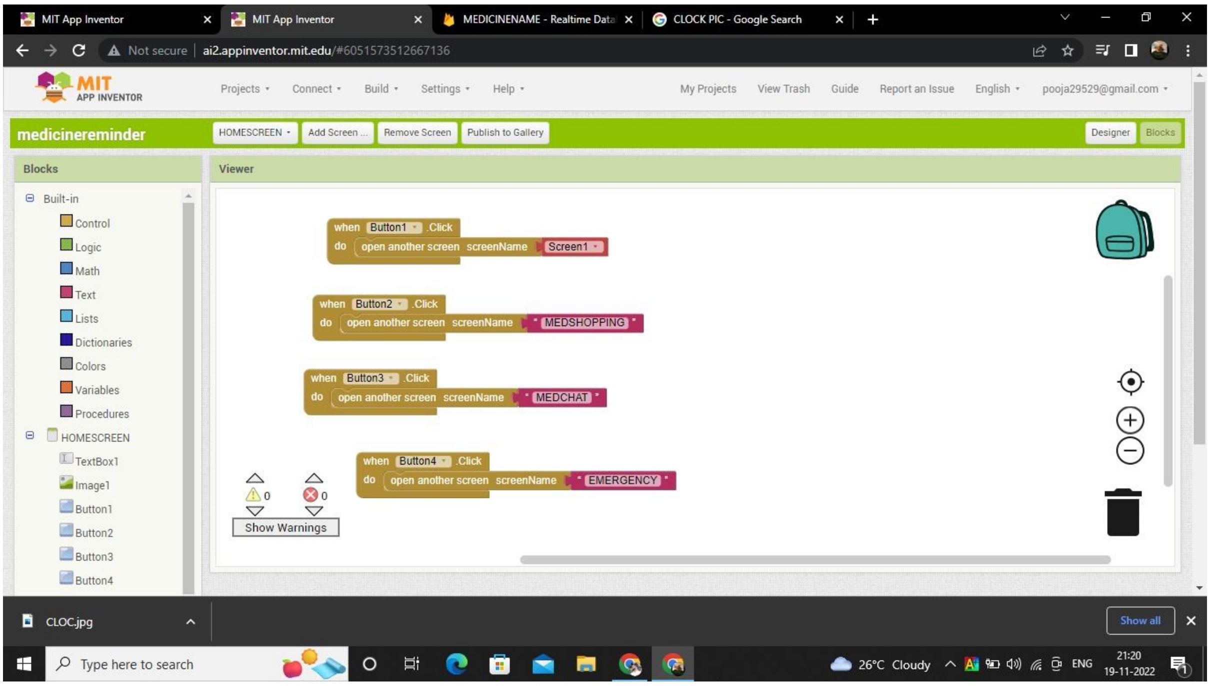
Shape
rounded

ShowFeedback
☑

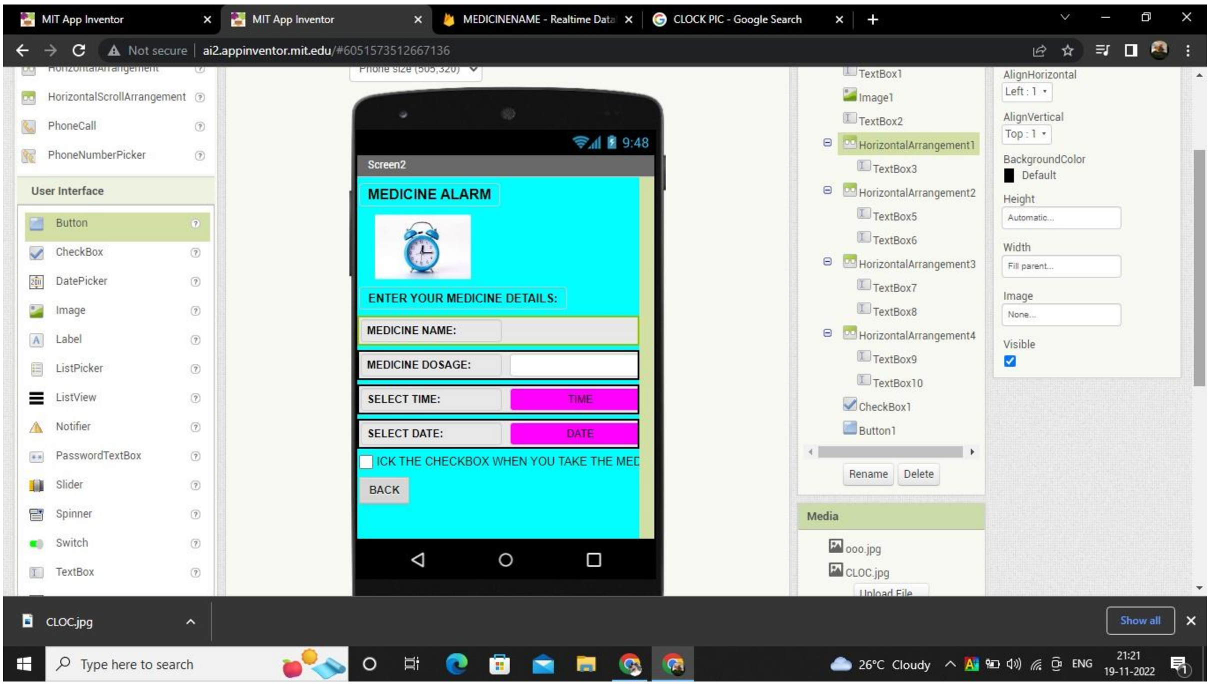
Text
EMERGENCY

TextAlignment

BLOCK CREATION



```
when Button1 .Click
do  open another screen  screenName  " Screen1 "

when Button2 .Click
do  open another screen  screenName  " MEDSHOPPING "

when Button3 .Click
do  open another screen  screenName  " MEDCHAT "

when Button4 .Click
do  open another screen  screenName  " EMERGENCY "
```



Screen2

MEDICINE ALARM

ENTER YOUR MEDICINE DETAILS:

MEDICINE NAME:

MEDICINE DOSAGE:

SELECT TIME: TIME

SELECT DATE: DATE

□ ICK THE CHECKBOX WHEN YOU TAKE THE MED

BACK

AlignHorizontal: Left : 1

AlignVertical: Top : 1

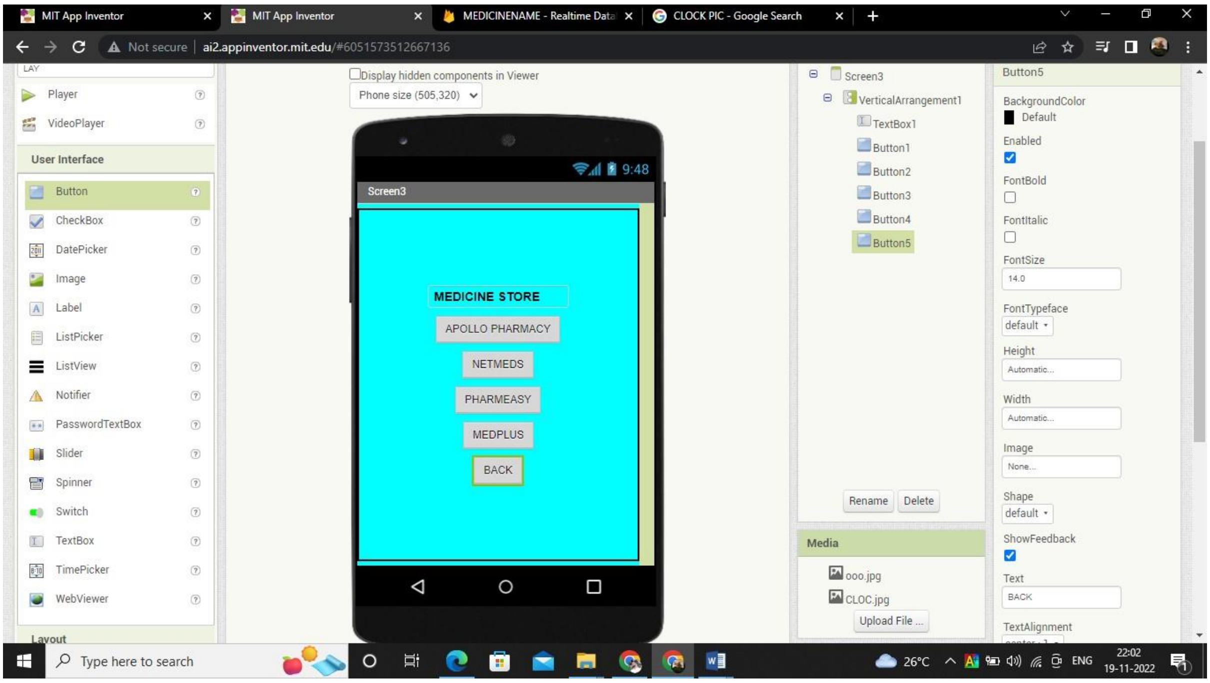
BackgroundColor: Default

Height: Automatic...

Width: Fill parent...

Image: None...

Visible: ☑



Screen3

MEDICINE STORE

APOLLO PHARMACY

NETMEDS

PHARMEASY

MEDPLUS

BACK

Button5

BackgroundColor: Default

Enabled: ☑

FontBold: □

FontItalic: □

FontSize: 14.0

FontTypeface: default

Height: Automatic...

Width: Automatic...

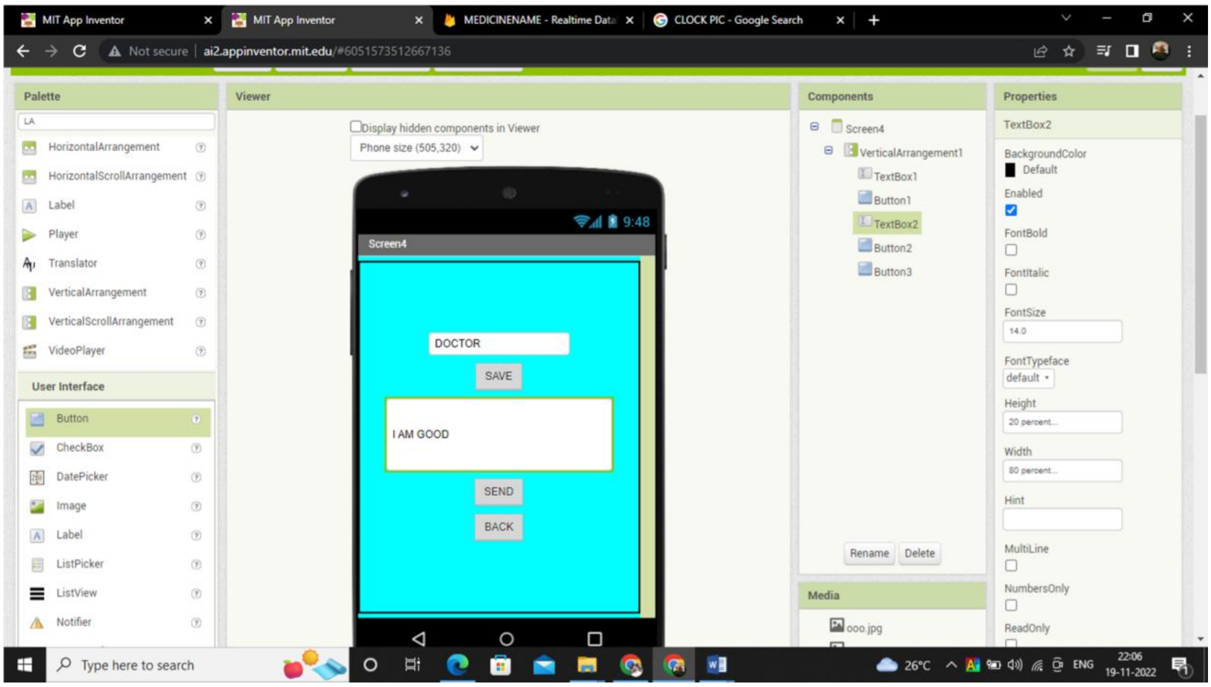
Image: None...

Shape: default

ShowFeedback: ☑

Text: BACK

TextAlignment: center : 1



Palette (search: LA): HorizontalArrangement, HorizontalScrollArrangement, Label, Player, Translator, VerticalArrangement, VerticalScrollArrangement, VideoPlayer

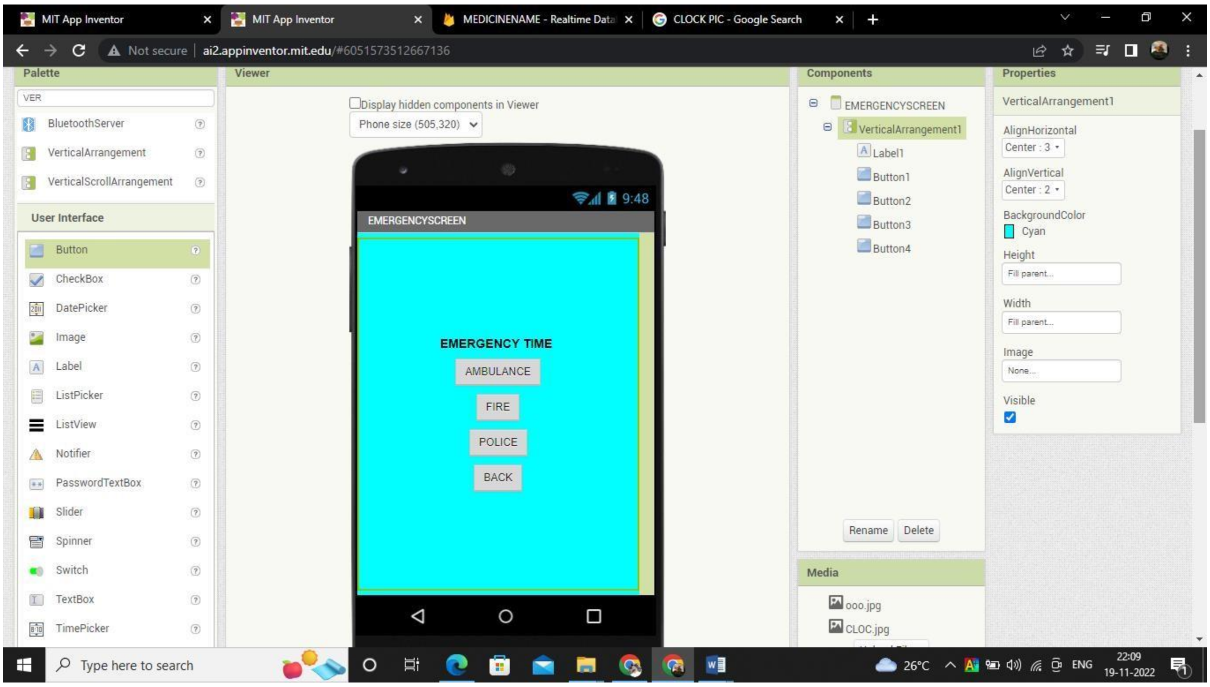
User Interface: Button, CheckBox, DatePicker, Image, Label, ListPicker, ListView, Notifier

Viewer – Screen4: DOCTOR, SAVE, I AM GOOD, SEND, BACK

Components: Screen4 > VerticalArrangement1 > TextBox1, Button1, TextBox2, Button2, Button3

Media: ooo.jpg

Properties – TextBox2:
BackgroundColor: Default
Enabled: ☑
FontBold: □
FontItalic: □
FontSize: 14.0
FontTypeface: default
Height: 20 percent...
Width: 80 percent...
Hint:
MultiLine: □
NumbersOnly: □
ReadOnly: □



Palette (search: VER): BluetoothServer, VerticalArrangement, VerticalScrollArrangement

User Interface: Button, CheckBox, DatePicker, Image, Label, ListPicker, ListView, Notifier, PasswordTextBox, Slider, Spinner, Switch, TextBox, TimePicker

Viewer – EMERGENCYSCREEN: EMERGENCY TIME, AMBULANCE, FIRE, POLICE, BACK

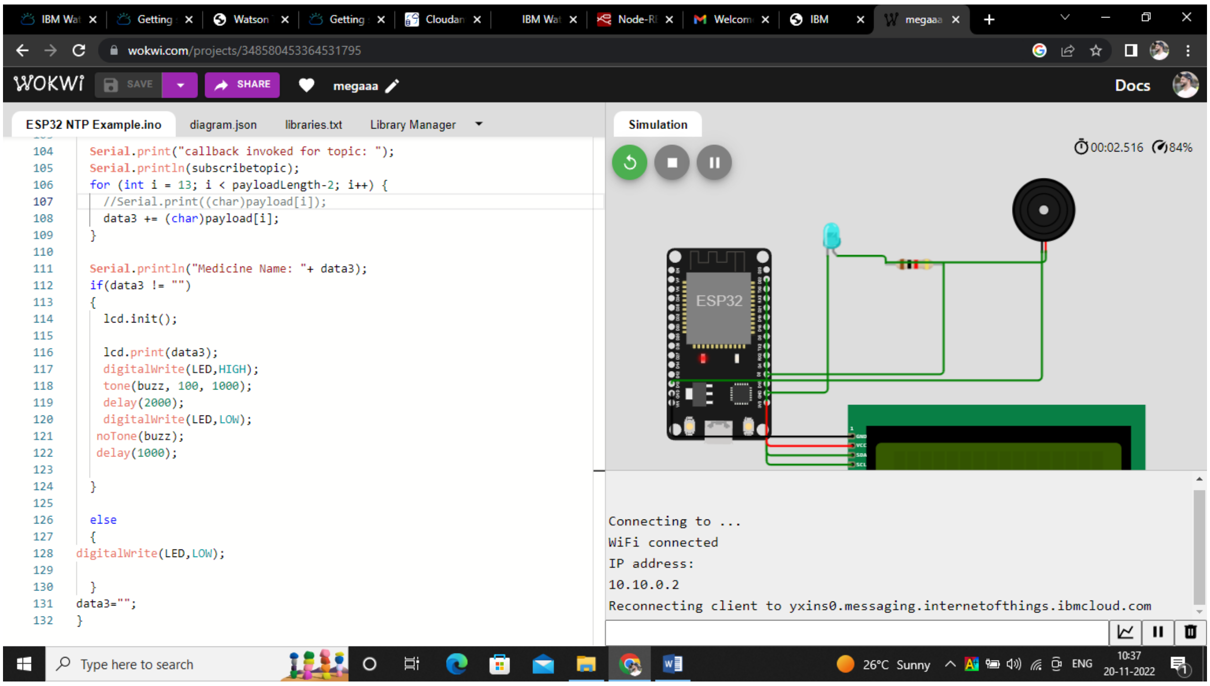
Components: EMERGENCYSCREEN > VerticalArrangement1 > Label1, Button1, Button2, Button3, Button4

Media: ooo.jpg, CLOC.jpg

Properties – VerticalArrangement1:
AlignHorizontal: Center : 3
AlignVertical: Center : 2
BackgroundColor: Cyan
Height: Fill parent...
Width: Fill parent...
Image: None...
Visible: ☑

CODE FOR SIMULATION

```
#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 "yxins0"//IBM ORGANITION ID
#define DEVICE_TYPE "b11m32deviceid"//Device type mentioned in ibm watson IOT
Platform
#define DEVICE_ID "123456"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678"     //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[] = "token";// 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 definion 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="";
}
```



```
  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="";
}
```

```
Connecting to ...
WiFi connected
IP address:
10.10.0.2
Reconnecting client to yxins0.messaging.internetofthings.ibmcloud.com
```