MIT AI2 204

IoT with MIT App Inventor

Fundamental

X. Tang

IoT with MIT App Inventor

Start 7:05PM

Please update your name with studentID

In Zoom meeting

Raspberry Pi Wi-Fi based projects

Project 1 - Getting and displaying the local WiFi parameters

Description: In this project, local Wi-Fi parameters, such as the IP address, MAC address, and signal strength of the Android device are obtained and displayed on its screen.

The Taifun WiFi component extension is used to get the local Wi-Fi parameters.

Getting and Displaying Local Wi-Fi Parameters

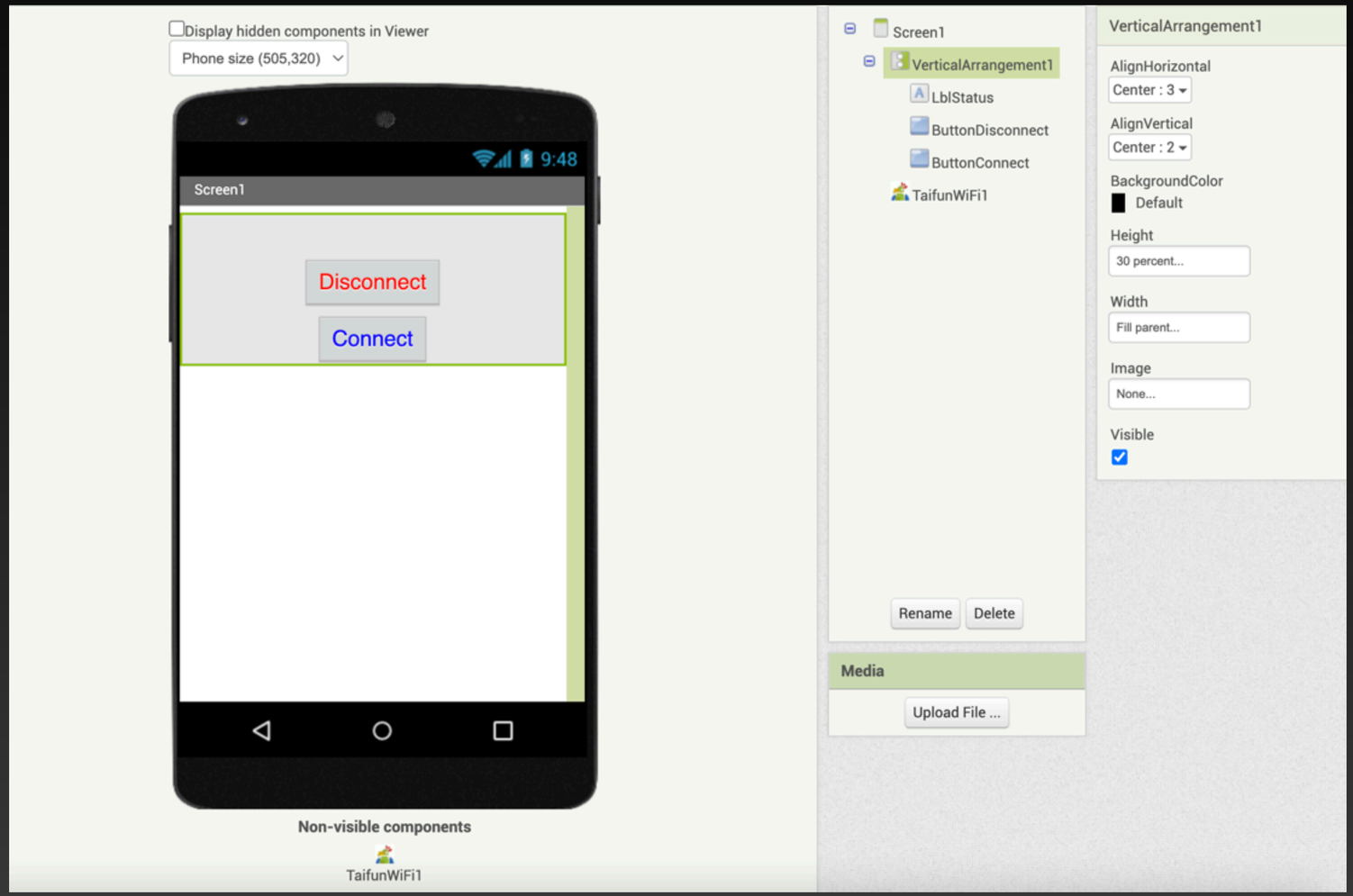
- Create a new project and name it as WIFI_TEST
- Insert a VerticalArrangement with the following configuration:

  AlignHorizontal: Center: 3

  AlignVertical: Center: 2

  Height: 30 percent

  Width: Fill parent



Getting and Displaying Local Wi-Fi Parameters

- Insert a Label with the following configuration. This Label will display the local Wi-Fi parameters:

  Name: LblStatus

  BackgroudColor: None

  FontBold: ticked

  FontSize: 20

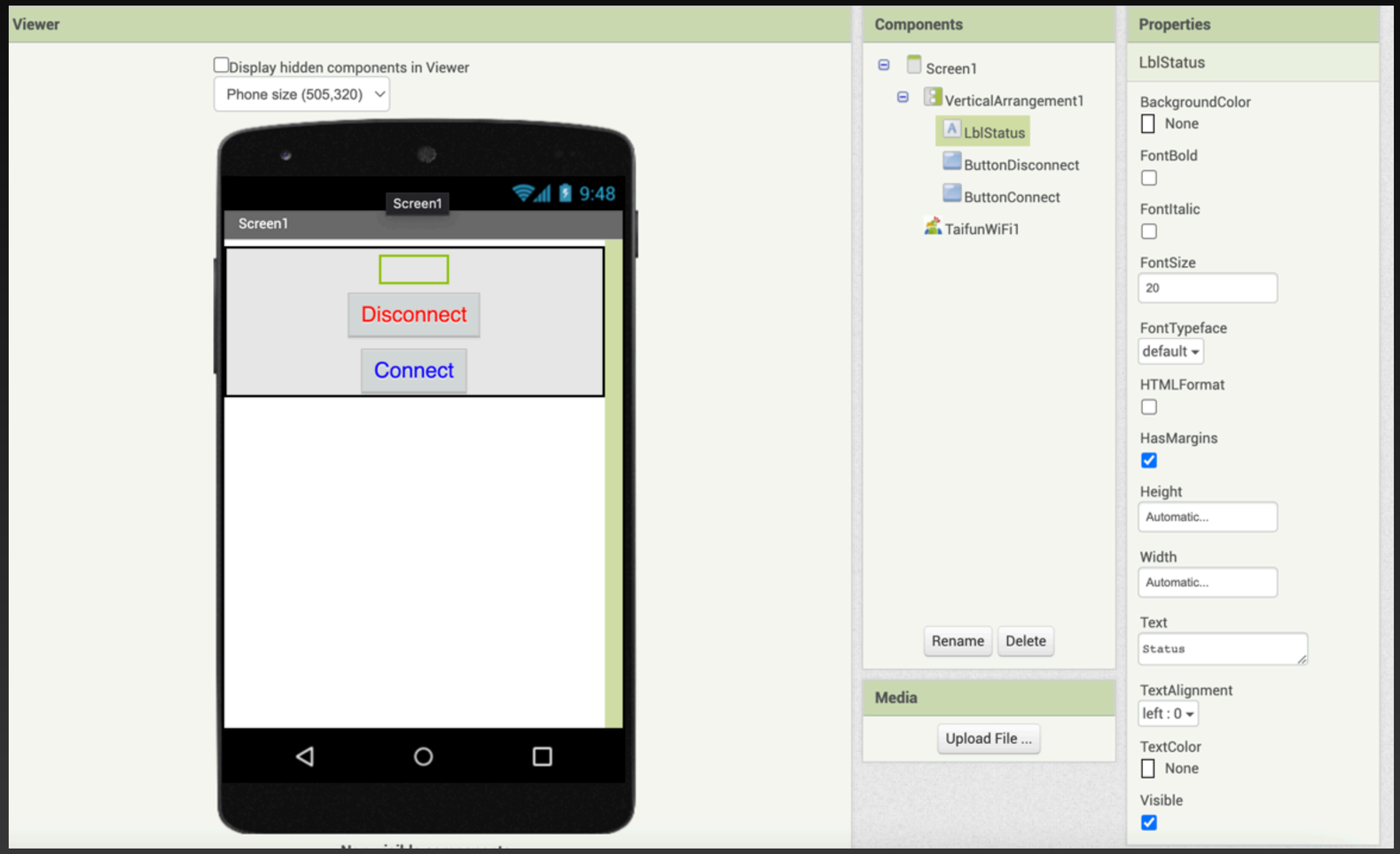
  Height: Automatic

  Width: Automatic

  Text: Status

  TextColor: None

  Visible: ticked



Getting and Displaying Local Wi-Fi Parameters

- Insert a Button with the following configuration. This button will disconnect from Wi-Fi when clicked:

  Name: ButtonDisconnect

  BackgroudColor: Default

  FontBold: ticked

  FontSize: 20

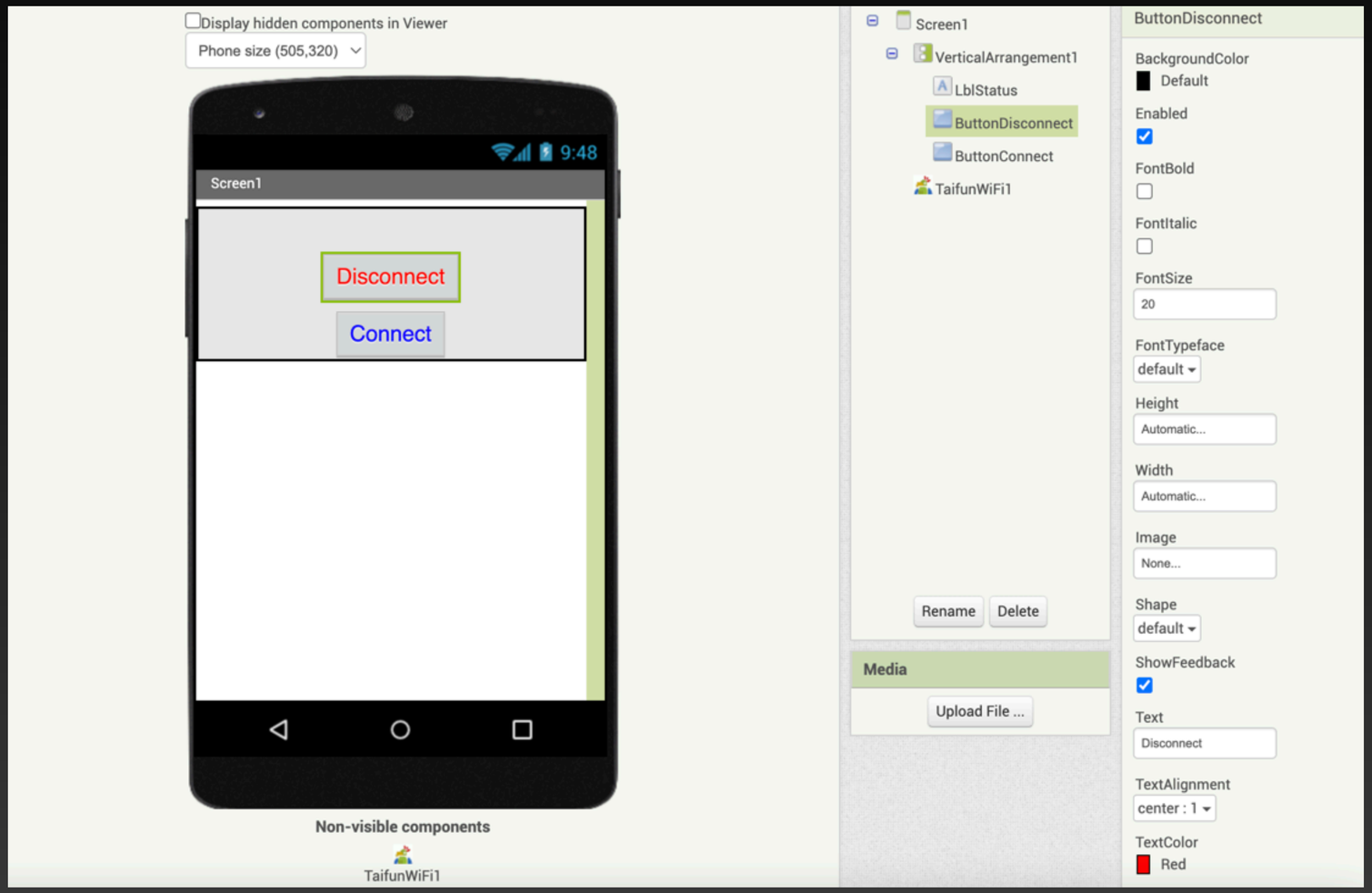
  Height: Automatic

  Width: Automatic

  Text: Disconnect

  TextColor: Red

  Visible: ticked



Getting and Displaying Local Wi-Fi Parameters

- Insert a Button with the following configuration. This button will connect to Wi-Fi when clicked:

  Name: ButtonConnect

  BackgroudColor: Default

  FontBold: ticked

  FontSize: 20

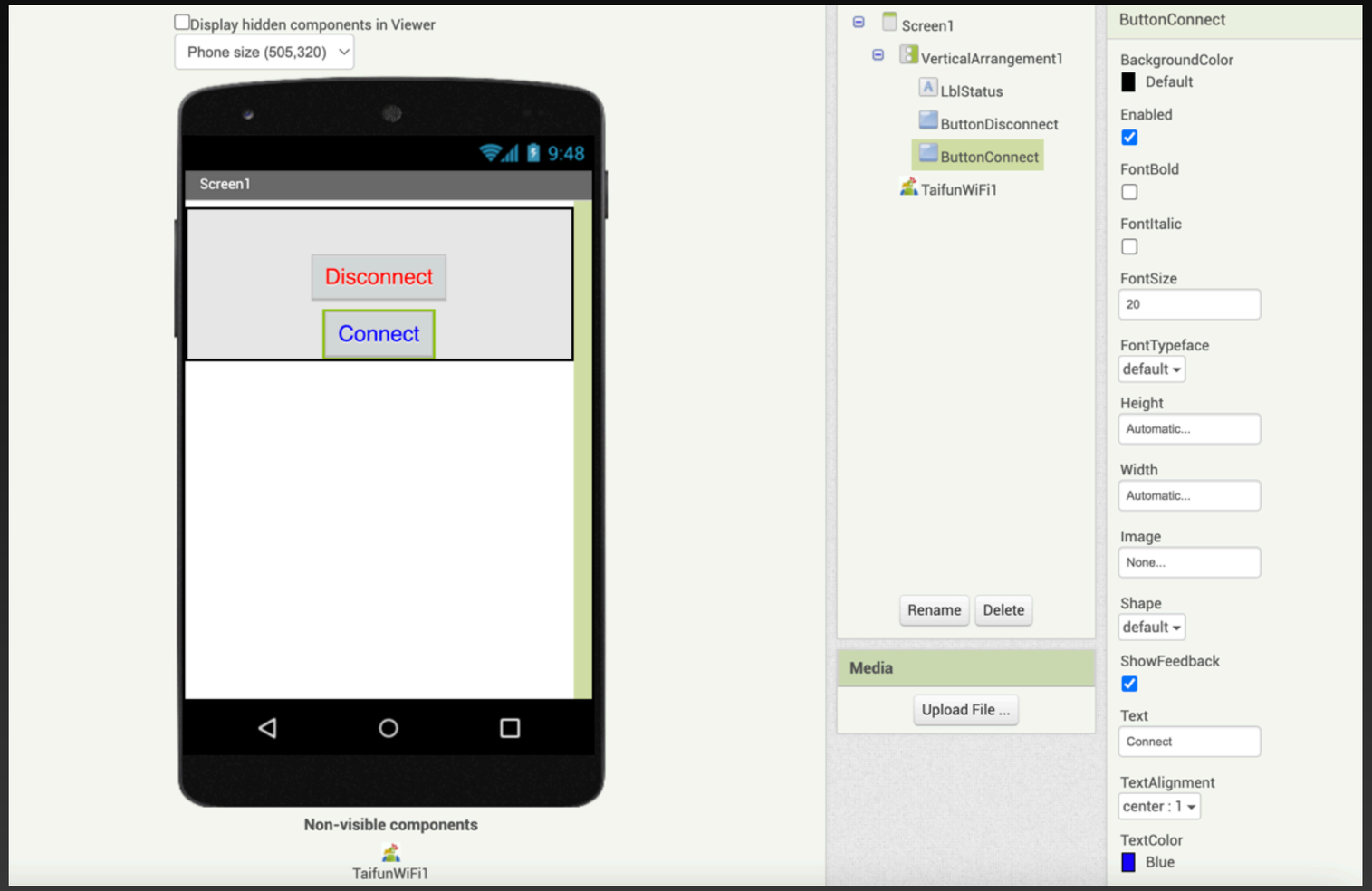
  Height: Automatic

  Width: Automatic

  Text: Connect

  TextColor: Blue

  Visible: ticked

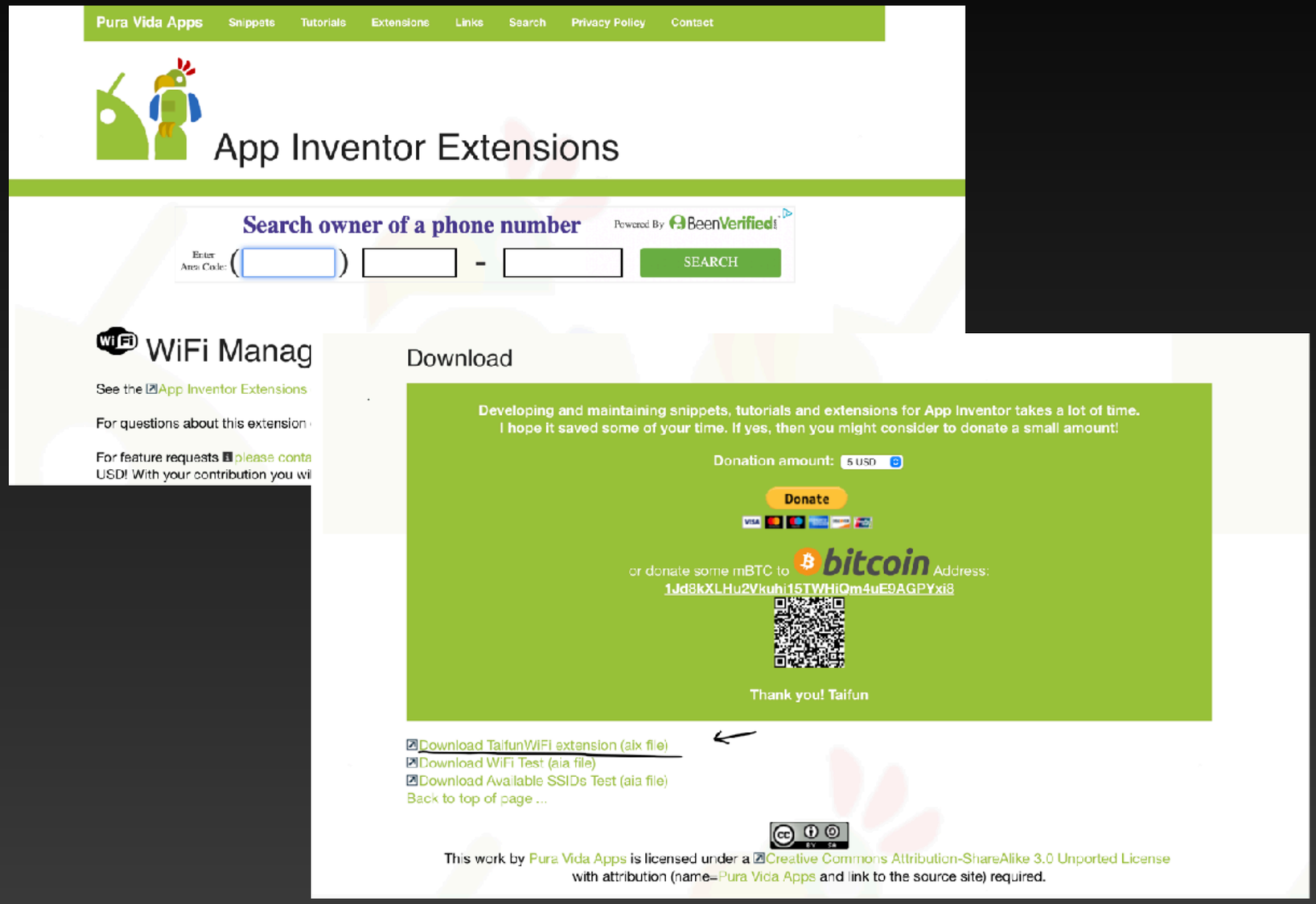


Getting and Displaying Local Wi-Fi Parameters

- Go to the following website:

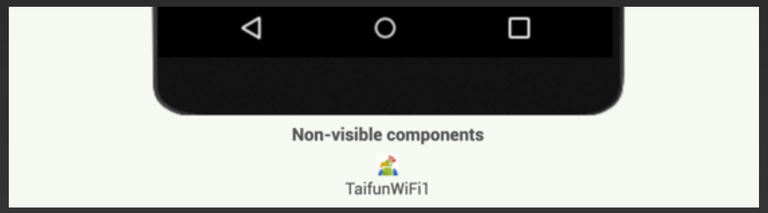
  https://puravidaapps.com/wifi.php

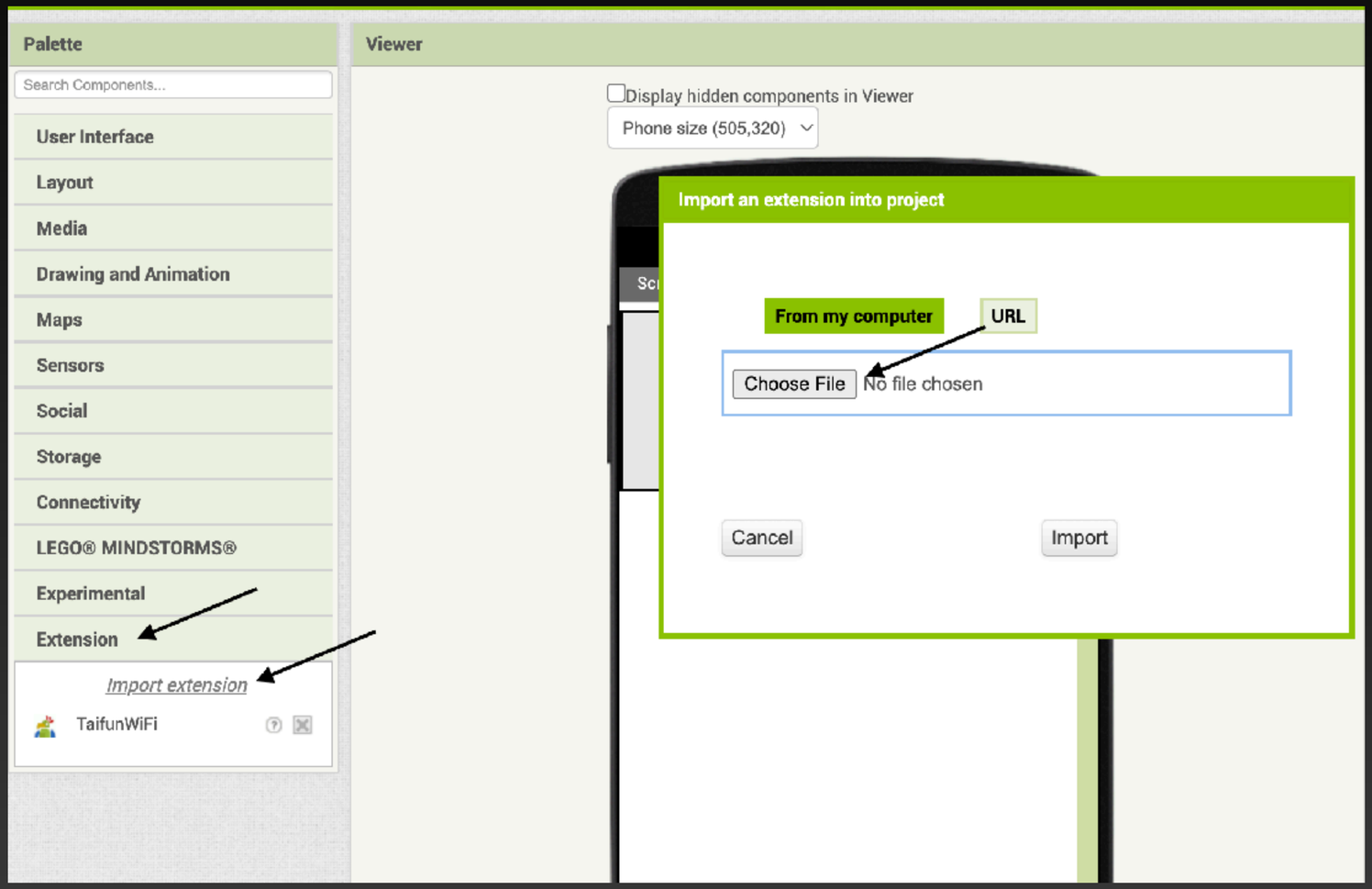
- Go to the end of the site and click Download TaifunWifi extension (aix file), download it to a folder



Getting and Displaying Local Wi-Fi Parameters

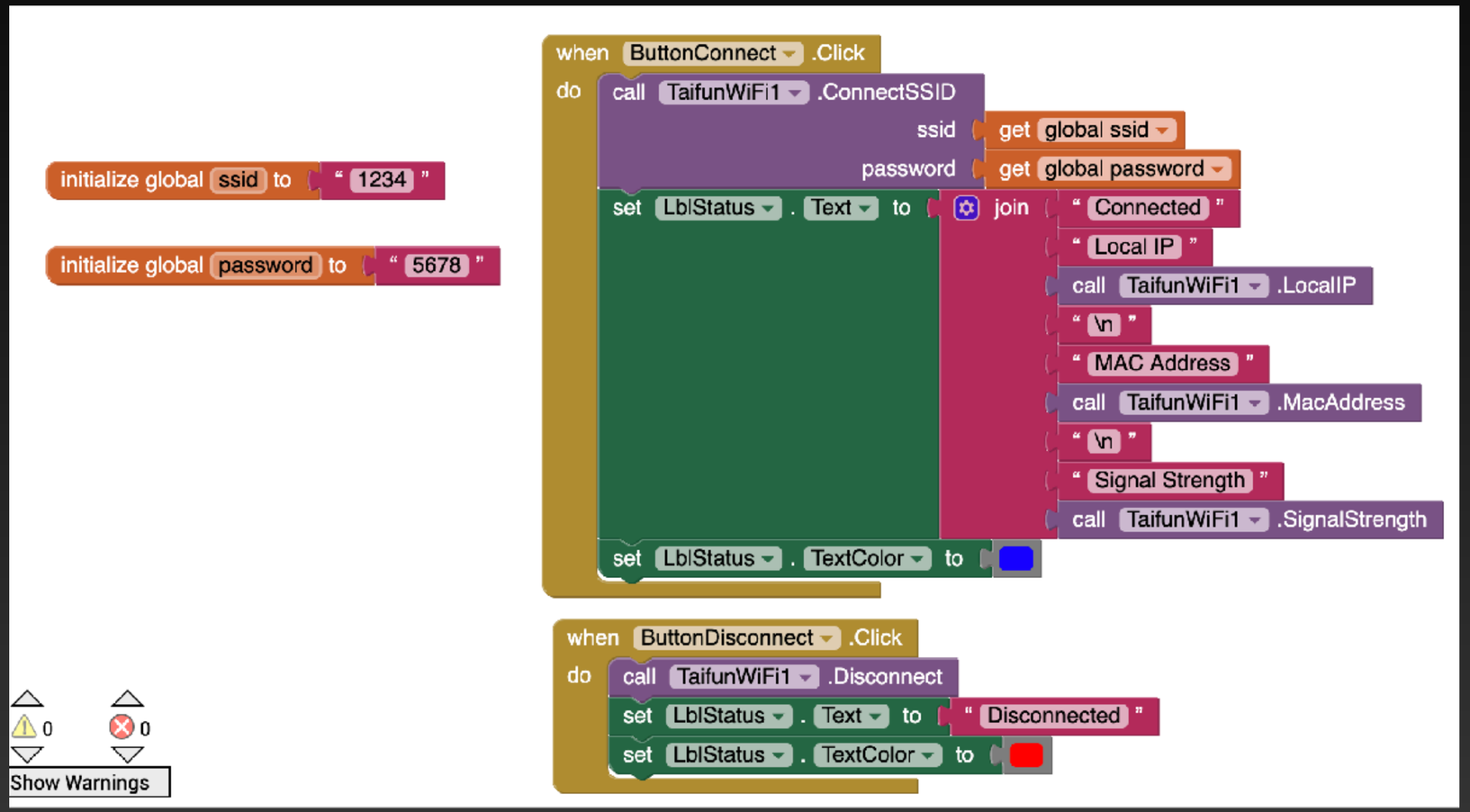
- Click Extension tab on the left-hand side of your app inventor project
- Click Import extension and browse to the file you just downloaded
- You should see a new component called TaifunWiFi under Extension. Click and drop it on your Viewer. This is a hidden component and will only be displayed under the phone image as TaifunWiFi1





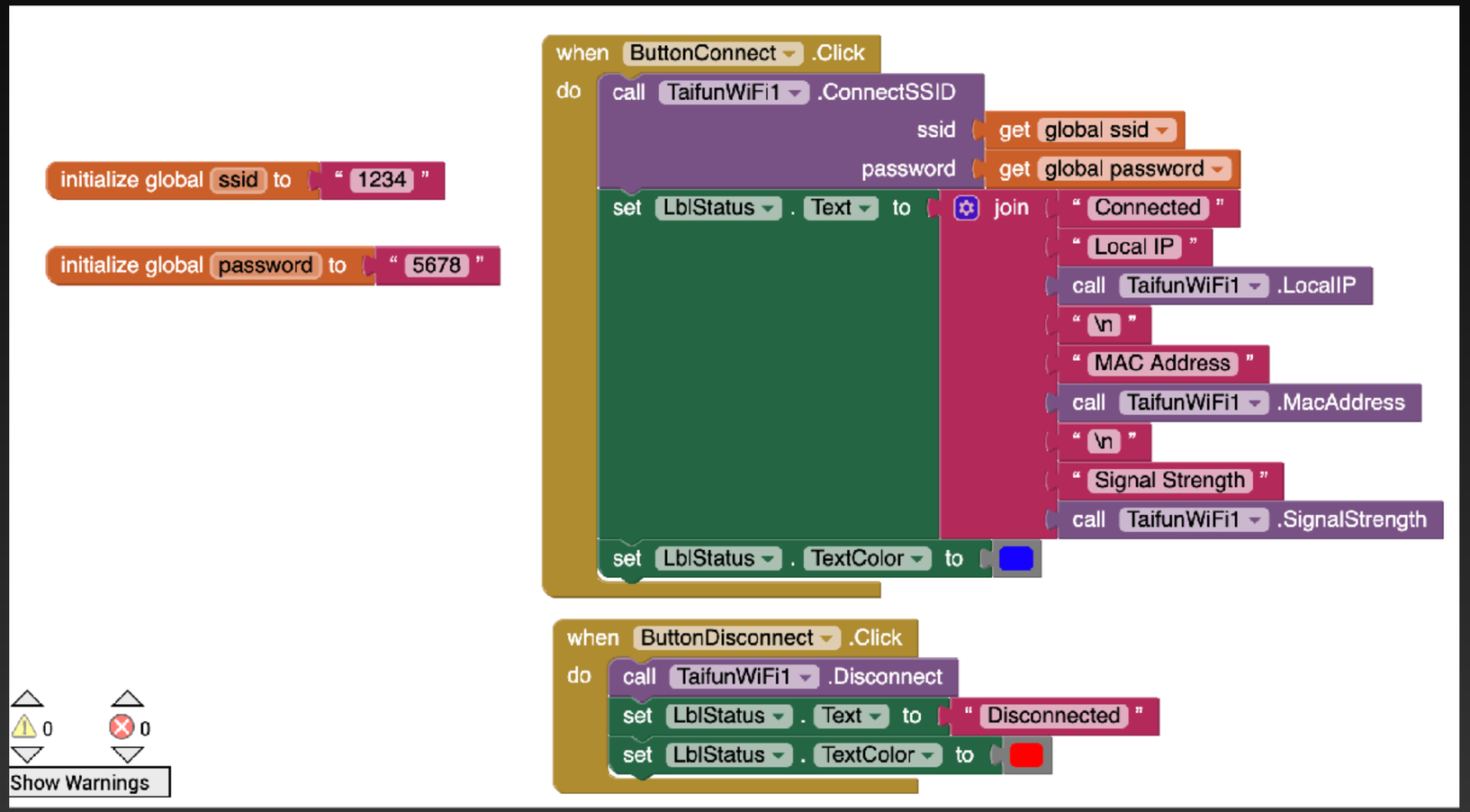
Getting and Displaying Local Wi-Fi Parameters

- Initialize two variables named said and password, Enter YOUR Wi-Fi SSID name and password into theses blocks respectively.
- Click ButtonConnect and select when ButtonConnect.Click do. This block will be executed when button Connect is clicked.
- Click on TaifunWiFi1 and select call TaifunWiFi1.ConnectSSID and join blocks ssid and password to this block. This block will establish a connection to the local Wi-Fi router.



Getting and Displaying Local Wi-Fi Parameters

- Inset Join block and extend it to 8 connectors. Enter blocks as shown to display the IP address, MAC address, and the signal strength.
- Click Button Disconnect disconnects from the Wi-Fi



Getting and Displaying Local Wi-Fi Parameters

- Connect your program through AI companion, and click on connect in your app.

- What did you see on your mobile screen?

Raspberry Pi Wi-Fi based projects

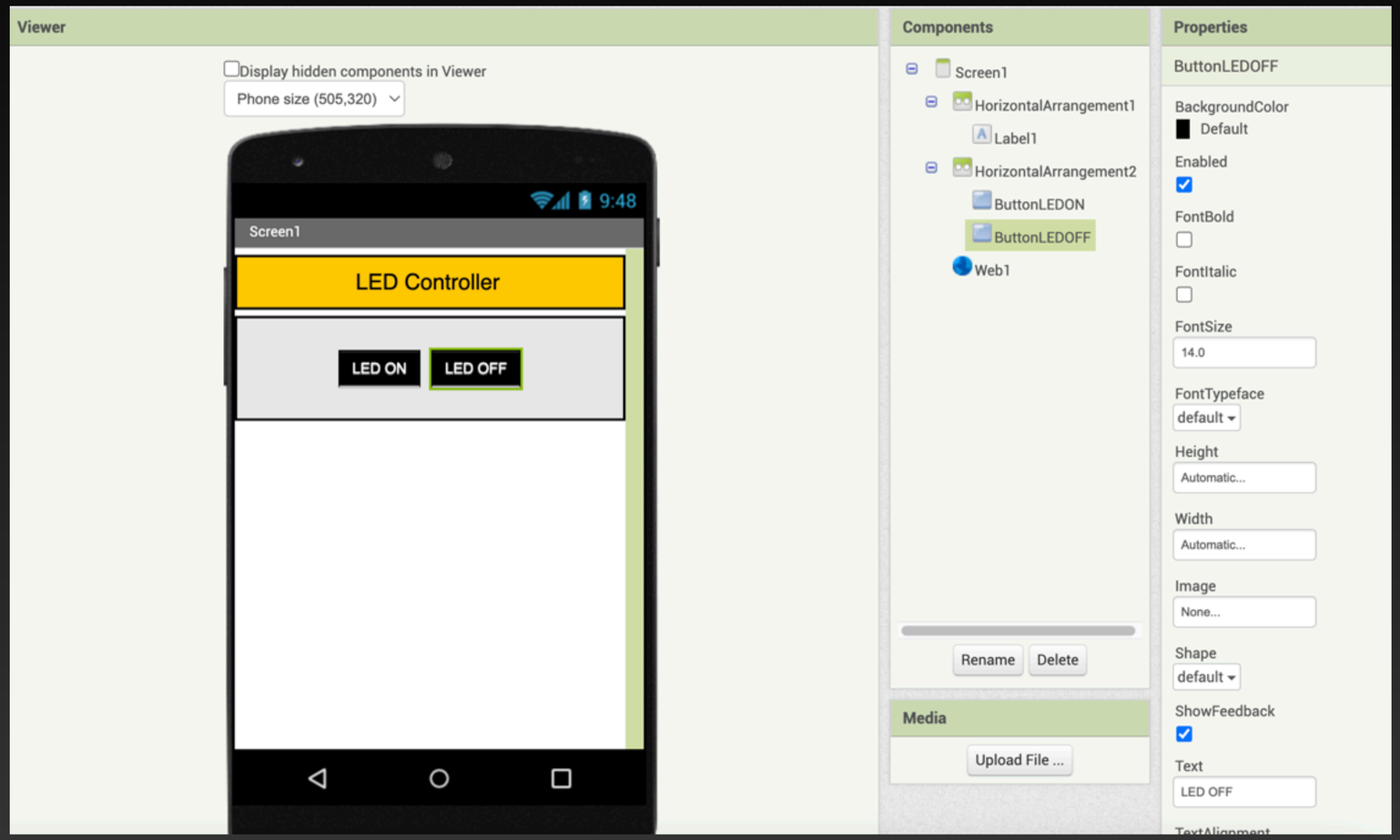
Project 2 - Web server to control LED

Description: In this project, an LED is connected to the Raspberry Pi and is controlled from an Android mobile phone using a web server application.

Use the same circuit program from previous project to setup LED on raspberry Pi.

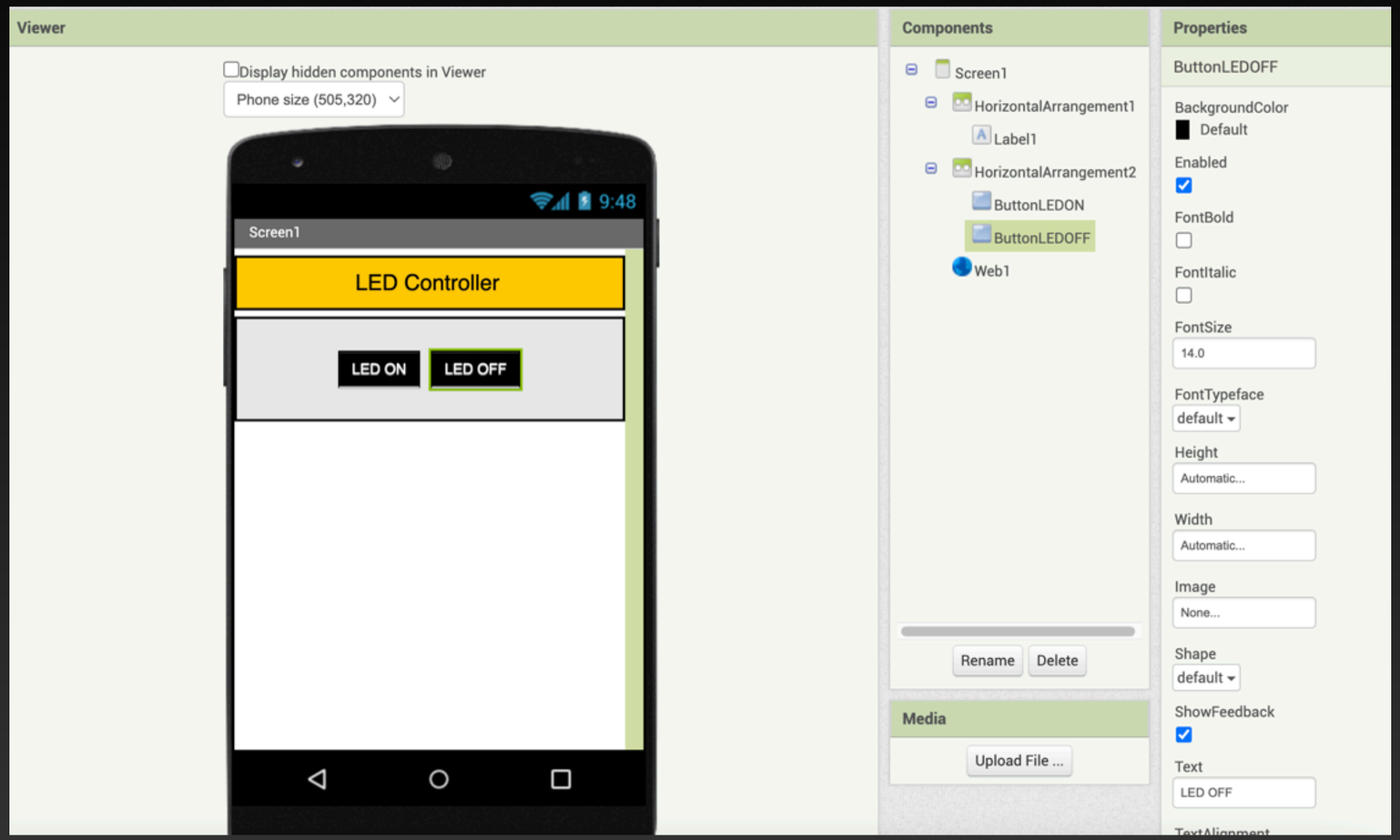
Web Server to Control LED

- Create a new project and name it as WEB_LED
- Insert a HorizontalAggrangement and insert a Label on it with its Text set to LED CONTROLLER
- Insert another HorizontalArrangement



Web Server to Control LED

- Insert two buttons on the HorizontalArrangement with the names ButtonON and ButtonOFF, with their texts set to LED ON and LED OFF respectively
- Click the Connectivity tab and insert a Web component on the Viewer. This is a hidden component.
- Initialize a variable called RaspberryPi and set it to the IP address of your RasspberryPi

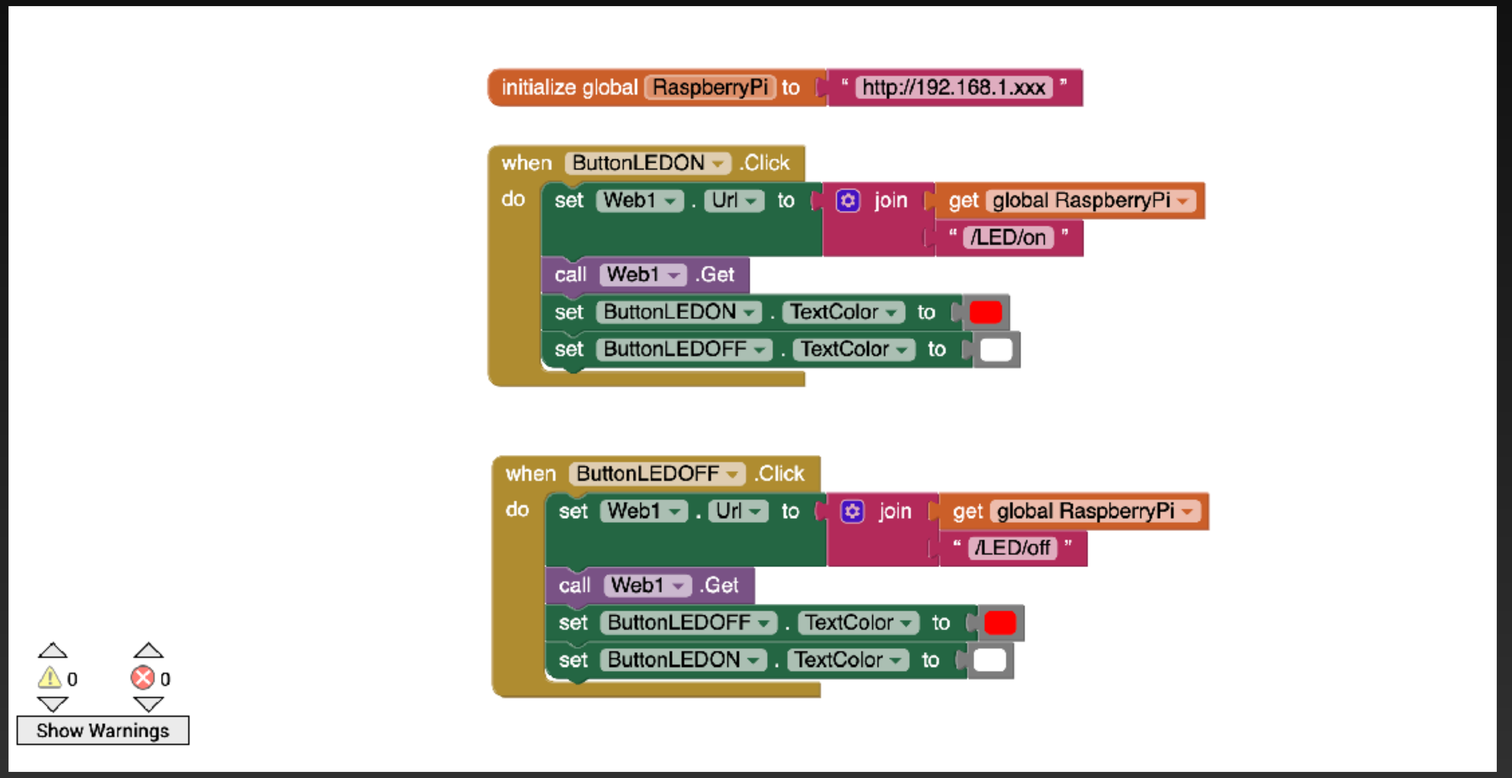


Web Server to Control LED

- Initialize a variable called RaspberryPi and set it to the IP address of your RasspberryPi
- Click ButtonON and select when ButtonON.Click do. This block will be executed when button LED ON is clicked.
- Click on Web1 and select set Web1.Url to.
- Insert a Join block and set the URL to the IP address of your URL and add string /LED/on to this block.

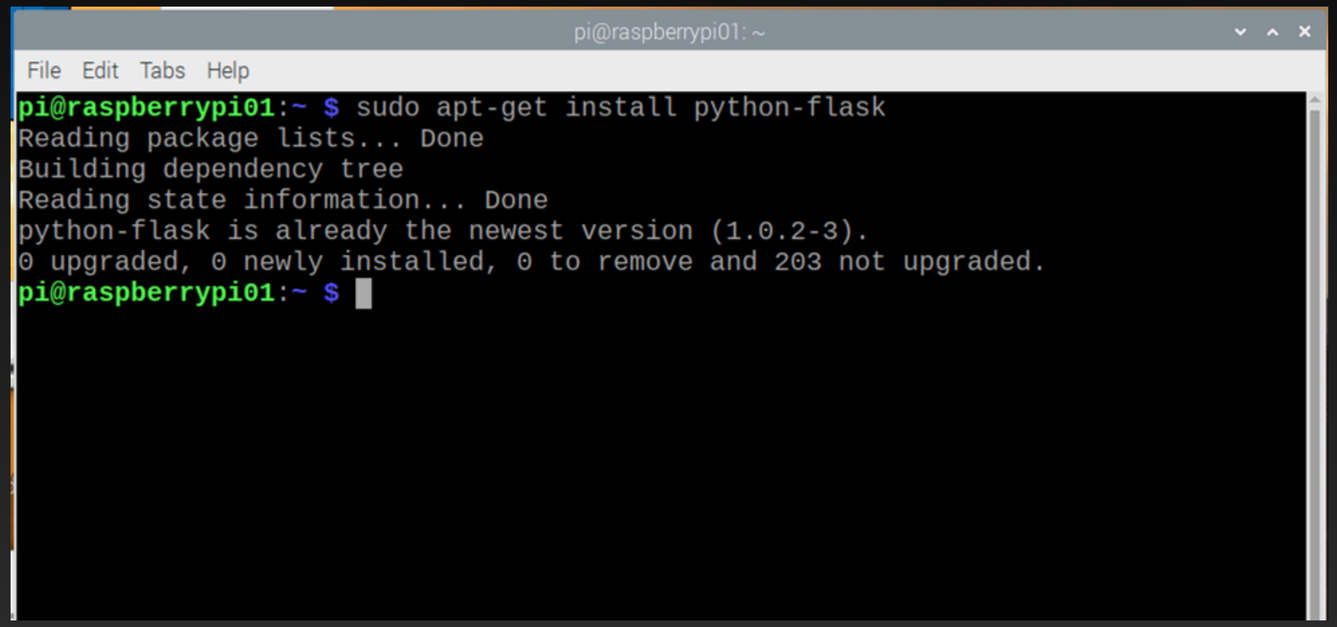
Web Server to Control LED

- Click on Web1 and select call Web1.Get. In this project, the URL is set to http://192.168.1.xxx/LED/on
- Set the cooler of ButtonON to red , and the color of ButtonOFF to white
- Repeat for the ButtonOFF as shown in the second group of blocks. But this time set the URL to http://192.168.1.xxx/LED/off



Web Server to Control LED

Python program: Python program is based on Flask which is a simple micro-framework written in Python. It is free of charge and can be used to create a web server on Raspberry Pi to control GPIO ports over the Internet.



Flask should already be available in Python on your Raspberry Pi, but if it isn't, it can be installed using command below:

```
sudo apt-get install python3-flask
```

Web Server to Control LED

Python program:

Import flask and gpiozero library

```
from flask import Flask,render_template

From gpiozero import LED


app=Flask(__name__)
```

```
13 #------------------------------------------------------------------
14
15 from flask import Flask, render_template   # import flask web server library
16 from gpiozero import LED
17
18 app = Flask(__name__)
19
20
21 led = LED(2)
22
23 #
24 #start the main program loop, read command from android app
25 #through local web server, decode the msg and control LED
26 #
```

Web Server to Control LED

Python program:

Set LED to 2 which is the GPIO 2 port it is connected to. This port is configured as an output and the LED is turned OFF at the beginning of the program.

```
Led = LED(2)
```

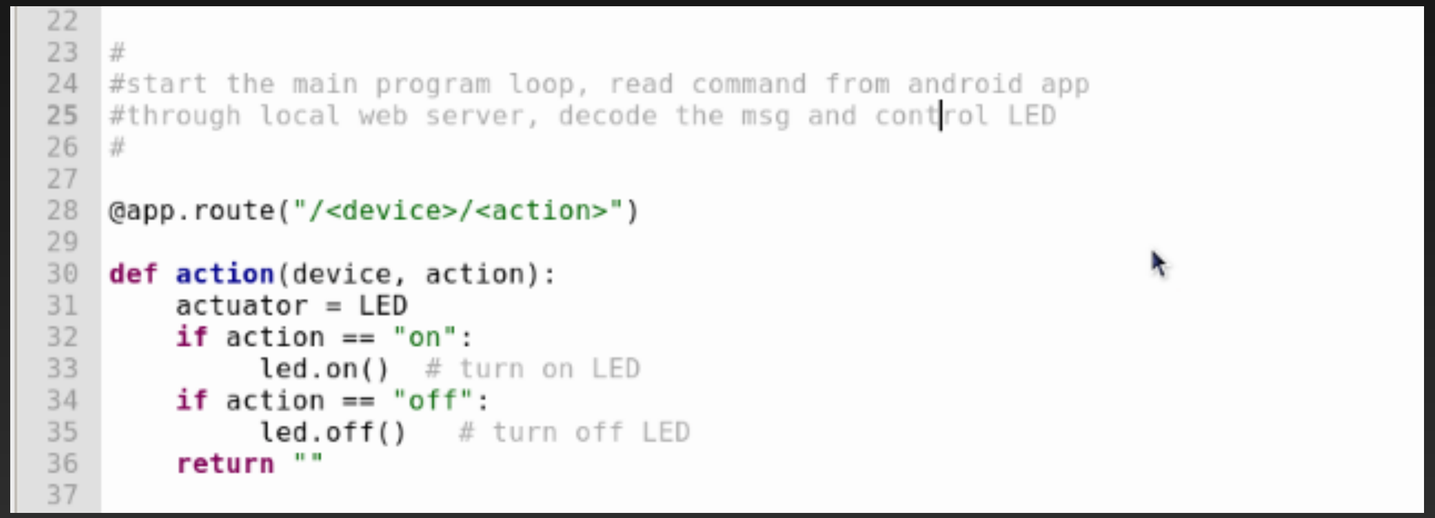
```
13 #------------------------------------------------------------------
14
15 from flask import Flask, render_template   # import flask web server library
16 from gpiozero import LED
17
18 app = Flask(__name__)
19
20
21 led = LED(2)
22
23 #
24 #start the main program loop, read command from android app
25 #through local web server, decode the msg and control LED
26 #
27
```

Web Server to Control LED

Python program:

An app.route is created with parameters device and action. For every actuator, we must have an action. If the action is on, the LED is turned ON, otherwise, if the action is off, the LED is turned OFF.

```
@app.route("/<device>/<action>")
def action(device, action):
    actuator = LED
    if action == "on":
        led.on()
    if action == "off":
        led.off()
   return ""
```



```
22
23  #
24  #start the main program loop, read command from android app
25  #through local web server, decode the msg and control LED
26  #
27
28  @app.route("/<device>/<action>")
29
30  def action(device, action):
31      actuator = LED
32      if action == "on":
33          led.on()  # turn on LED
34      if action == "off":
35          led.off()   # turn off LED
36      return ""
37
```

Web Server to Control LED

Python program:

Notice that the LED can be turned ON by entering the URL: http://192.168.1.xxx/LED/on to our web browser. Similarly, the LED can be turned OFF by the URL: http://192.168.1.xxx/LED/off

```
if __name__ == '__main__':

    app.run(debug=True, port=80,
host='0.0.0.0', use_reloader=False)
```

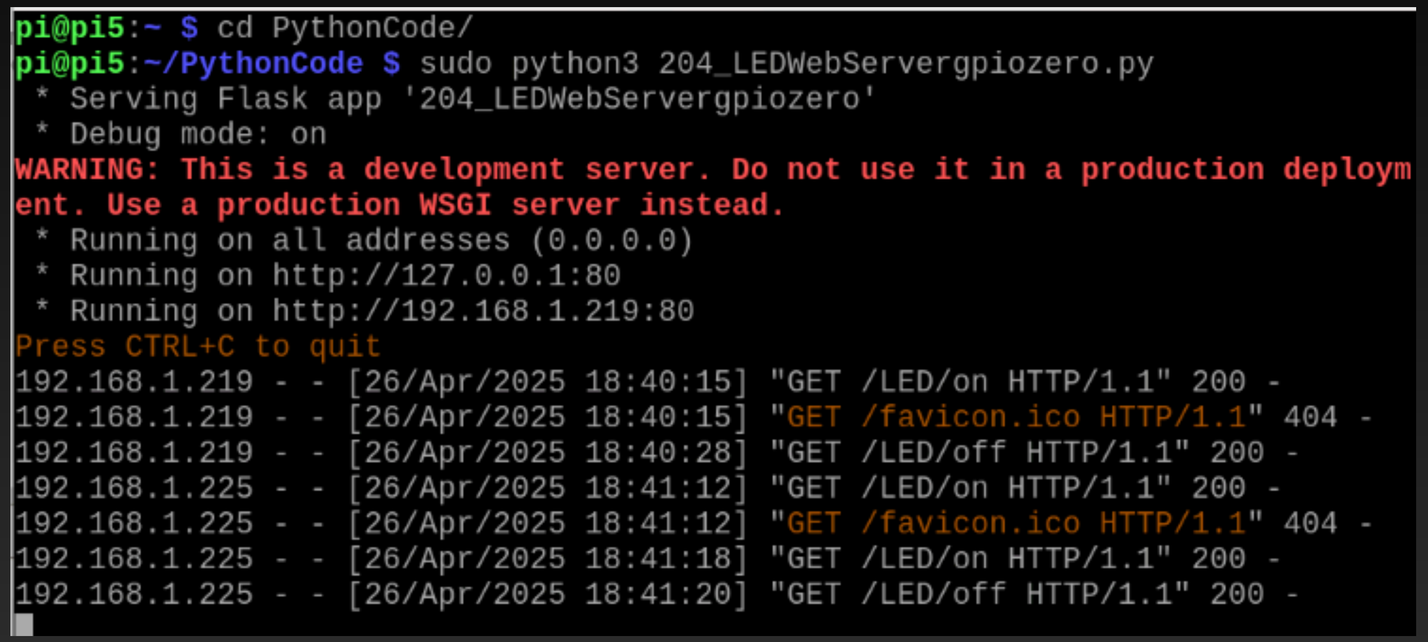
```
23 #
24 #start the main program loop, read command from android app
25 #through local web server, decode the msg and control LED
26 #
27
28 @app.route("/<device>/<action>")
29
30 def action(device, action):
31     actuator = LED
32     if action == "on":
33         led.on()  # turn on LED
34     if action == "off":
35         led.off()   # turn off LED
36     return ""
37
38 if __name__ == '__main__':
39     app.run(debug=True, port=80, host='0.0.0.0', use_reloader=False)
40
```

Web Server to Control LED

Python program:

To test your program, go to the folder where you project file stored, start python program on your Raspberry Pi from command line as:

```
sudo python3 204_LEDWebServergpiozero.py
```



Python Program - LED control through bluetooth

Home project1 - Develop app to control LED through speech commands

Home project2 - Develop app to control 3 LEDs through WIFI