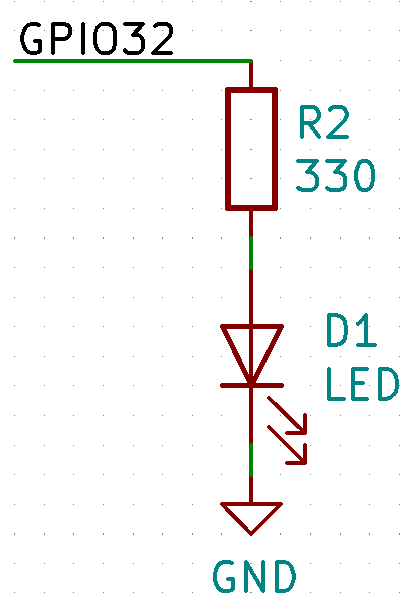
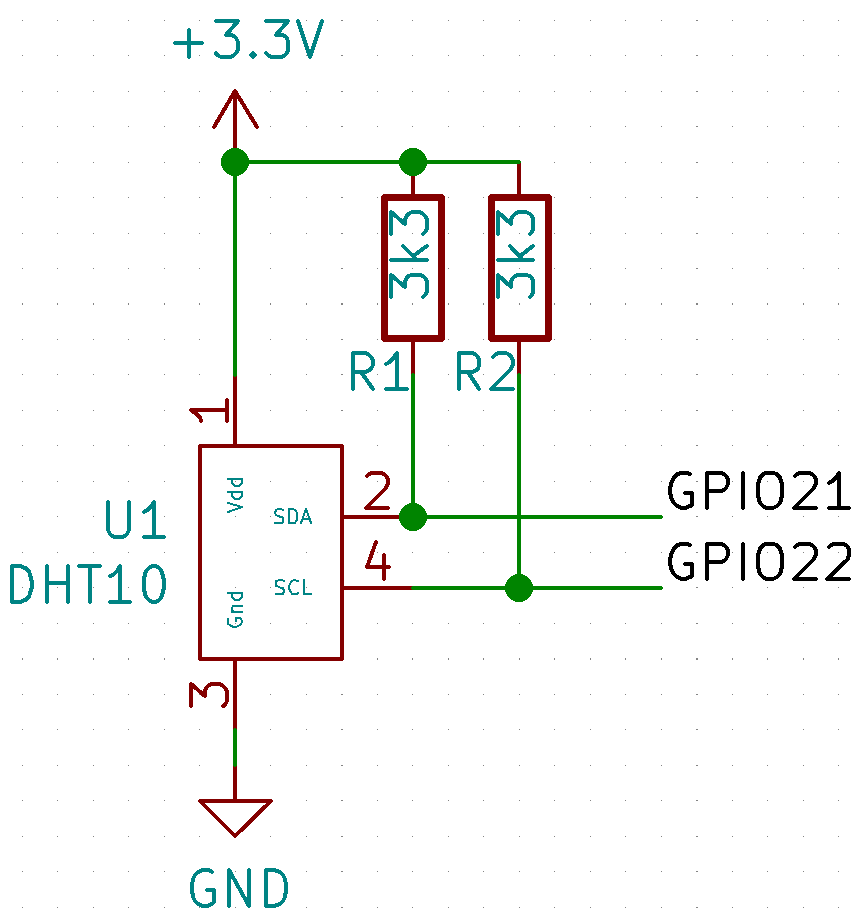
**Lab 13**

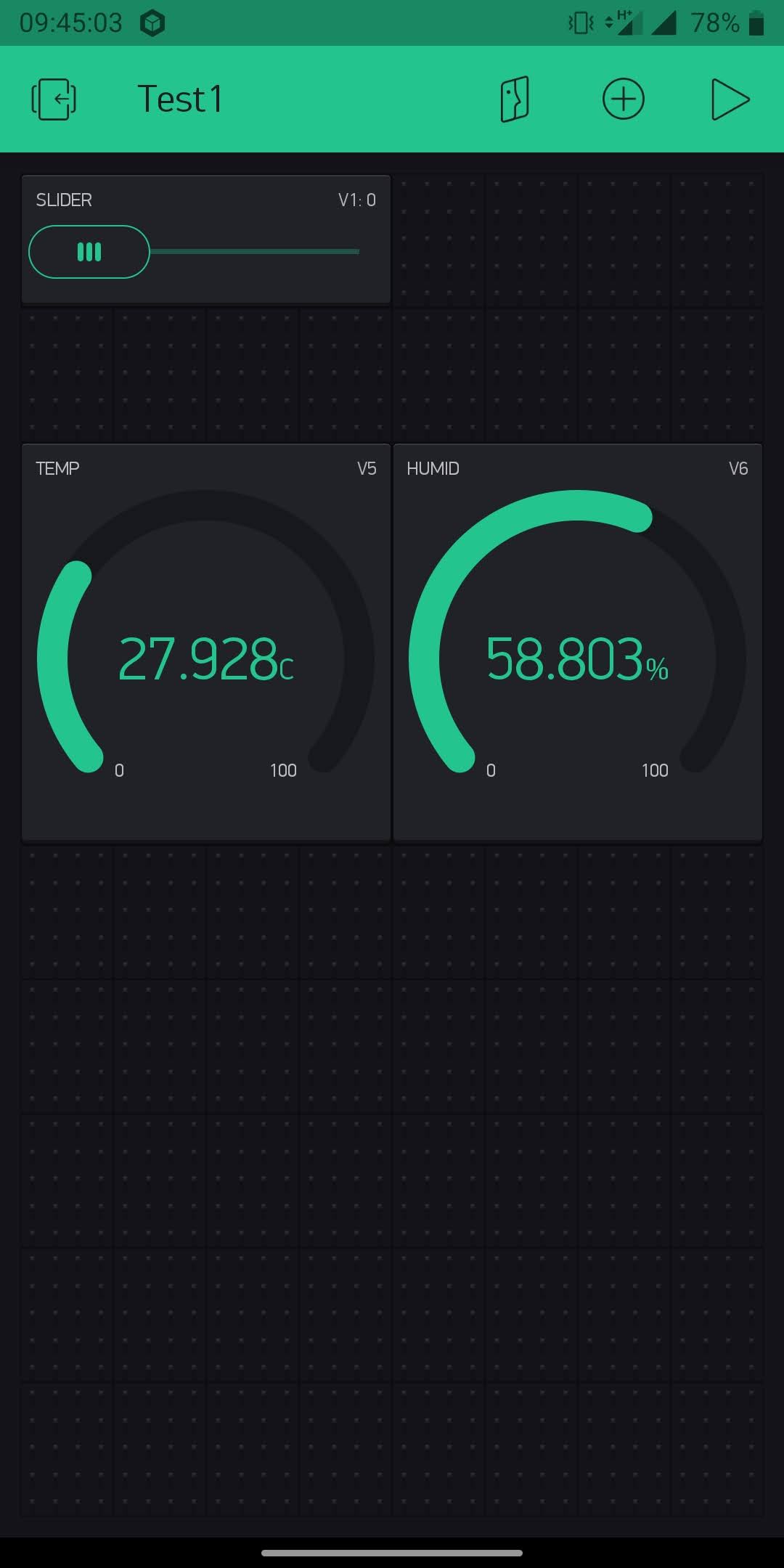
**PART I**

**Exercise 1**

**The above figure shows the architecture of this exercise.**

****

* 1. **Wire up the above circuit on ESP32.**
  2. **Install Blynk application on your mobile phone and create an account according to** [**this instruction**](https://docs.blynk.cc/#getting-started)**.**
  3. **On your Arduino IDE, install Blynk library via Manage Libraries.**
  4. **In Blynk application on your mobile phone, create the following widgets:**
     1. **Slider: to control the brightness of the LED on ESP32; it binds to virtual pin V1 and has the range from 0 to 1023.**
     2. **Gauge: to display the temperature from DHT10 on ESP32; it binds to virtual pin V5 and has the range from 0 to 100.**
     3. **Gauge: to display the humidity from DHT10 on ESP32; it binds to virtual pin V6 and has the range from 0 to 100.**

****

* 1. **In Blynk application, go to the menu in the red rectangle above, then send AUTH TOKENS to your registered email. You will use the token in your ESP32 program.**
  2. **Upload** DHT10-LED-Blynk.ino **to ESP32. Do not forget to fill in your Wi-Fi SSID, password, Blynk authentication token, and MQTT server IP address. The MQTT broker server can be your RPi but it is not used now, instead it will be used in the next exercise.**
  3. **Make a video clip of your finished product then put your video shared URL below as a submission. In the video should show temperature and humidity changing on the application, and the physical LED fading according to the slider adjustment on Blynk.**

**Exercise 2**

**2.1 Enable Mosquitton on your RPi and install MongoDB according to** [**this instruction**](https://pimylifeup.com/mongodb-raspberry-pi/)**.**

**2.2 Log the temperature and humidity sent from ESP32 to MongoDB. You can use** [**this Python library**](https://github.com/David-Lor/MQTT2MongoDB)**.**

**2.3 Zip your program folder including your used libraries that gets the data from MQTT broker and write it to MongoDB. Then submit according with this document.**

**https://drive.google.com/file/d/1ZDWsJBmGrn\_VUYRpg-IQafkOjA247ZM5/view?usp=sharing**

**Then you should get the following flow.**

* 1. **Deploy your flow then capture your finished dashboard and put it below.**

