Web Infrastructure
Assignment #2

Introduction:

In this assignment, using ESP32 you will setup a web server & access point which will switch LEDs on/off

Note:

- Professor will provide you 2 LEDs. Pick them up during class.
- You must follow the instructions precisely (SSID name, URLs) as your assignment will be tested in class and there will be no accommodations for different URLS or SSIDs with passwords.
- We will test your assignment before-class, please come in early on the designated day.

Steps: part#1

1) Read the note: “Getting started with ESP32 WiFi Module” on Piazza (https://piazza.com/class/ivvcz3pop7x4xu?cid=9)
2) Follow instruction how to install Arduino ESP32 support on your OS https://github.com/espressif/arduino-esp32/tree/master/docs/arduino-ide
   tip: if you are using ESP32 from hiletgo, you’ll need to find and install drivers first, and when choosing a board from the options in Arduino IDE, pick Node32s
   tip: useful video for the configuration: https://www.espressif.com/en/content/iot-college-videos
   from 2:30 to 4:00 -> setting up Arduino IDE with ESP32
3) Using Arduino development software (1.8.8), GoTo file->examples->Wifi->WiFiAccessPoint
4) In the code, **set SSID to your first name and no password** (NULL)
5) Upload the code to your ESP32 and test if your website is working (connect to you access point, and use your browser to go to 192.168.4.1

Steps: part#2

1) Connect the LED to your board as follows
   a. (Note the long wire is +ve, short one is –ve)
   b. See the pin diagram and info in the “Getting started with ESP8266 WiFi Module” on Piazza
   c. Attach the LED +ve wire to GPIO4,
   d. Attach -ve wire to the GND pin (use a breadboard)
2) In the setup() function init your pin to output mode: **`pinMode(4, OUTPUT);`**
3) In the loop() function, find the condition: **`if (currentLine.length() == 0)`**
4) Inside this condition, we will add two HTTP responses. One to link to 192.168.4.1/on and another for 192.168.4.1/off
   `client.print("Click <a href=\"/on\">here</a> to turn ON the LED.<br>");`
5) Outside of if block: **`if (c == '\n')`** check to see what the request was

```
if (currentLine.endsWith("GET /on")) {
  digitalWrite(4, HIGH);
}
```

6) Write similar code for OFF
7) Upload the code to your ESP32 and test if your website is working (connect to you access point, and use your browser to go to 192.168.4.1
8) The following URL will set the LED on : http://192.168.4.1/on
9) The following URL will set the LED off : http://192.168.4.1/off

tip: detailed tutorial how to set up WiFi Access Point
https://randomnerdtutorials.com/esp32-access-point-ap-web-server/

Steps: part#3

1) At this point you should be able to switch on/off an LED attached to GPIO4
2) Attach the 2nd LED to an available GPIO and write new code to switch it on/off using /on2 /off2 URLS (instead of /on /off in part#2)

Steps: part#4

3) Create an HTML code that gets served when accessing http://192.168.4.1 and allow the LEDs to be controlled from the web page.
4) Your web page MUST HAVE:
   a) Buttons to switch each LEDs on/off (choice of buttons/styles is up to you)
   b) A High-Resolution Background Image for the web page

Deliverables:

1) Bring your ESP32 with everything ready to class and have it ready for testing
2) When asked to do so, power it up, Instructor will connect and test the web page and functionality
3) Upload your code & screenshots to Camino as per assignment guidelines.