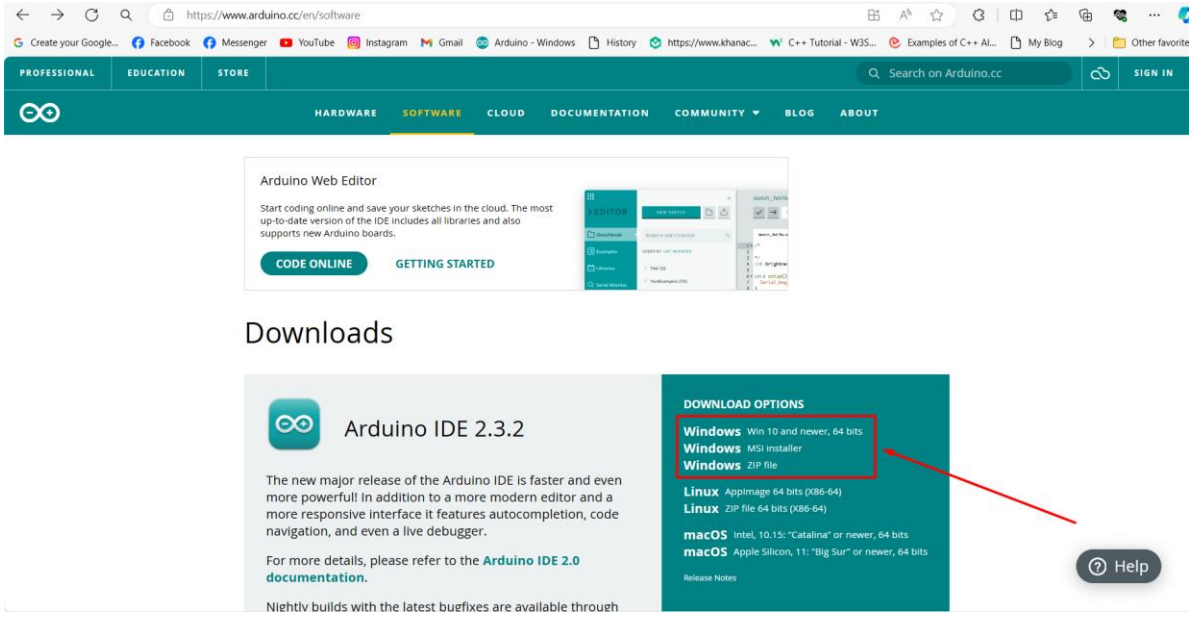
ESP32CAM WITH REACT JS Documentation

1. ESP32CAM SETUP

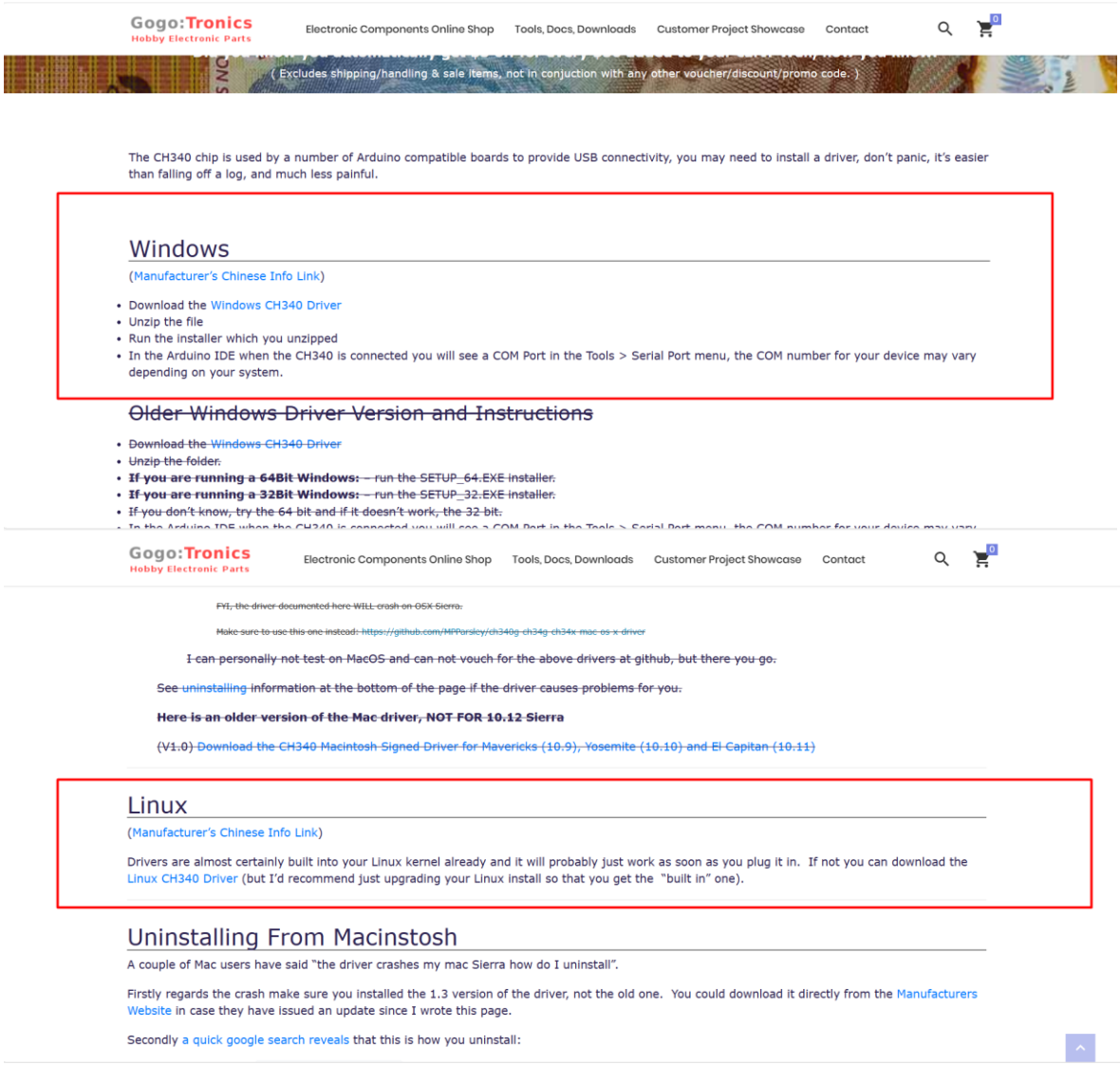
- Install the requirements:

1. Arduino IDE: You can download from this website: [Software | Arduino]



You can choose the latest IDE and installer based on your preference and if it is compatible with your computer.

2. Download and install CH340 Drivers on your computer. Kindly go to this link: [CH340 Drivers for Windows, Mac and Linux (gogo.co.nz)]



Gogo:Tronics
Hobby Electronic Parts

Electronic Components Online Shop | Tools, Docs, Downloads | Customer Project Showcase | Contact

(Excludes shipping/handling & sale items, not in conjuction with any other voucher/discount/promo code.)

The CH340 chip is used by a number of Arduino compatible boards to provide USB connectivity, you may need to install a driver, don't panic, it's easier than falling off a log, and much less painful.

Windows

(Manufacturer's Chinese Info Link)

- Download the Windows CH340 Driver
- Unzip the file
- Run the installer which you unzipped
- In the Arduino IDE when the CH340 is connected you will see a COM Port in the Tools > Serial Port menu, the COM number for your device may vary depending on your system.

~~Older Windows Driver Version and Instructions~~

- ~~Download the Windows CH340 Driver~~
- ~~Unzip the folder.~~
- ~~**If you are running a 64Bit Windows:** – run the SETUP_64.EXE installer.~~
- ~~**If you are running a 32Bit Windows:** – run the SETUP_32.EXE installer.~~
- ~~If you don't know, try the 64 bit and if it doesn't work, the 32 bit.~~

Gogo:Tronics
Hobby Electronic Parts

Electronic Components Online Shop | Tools, Docs, Downloads | Customer Project Showcase | Contact

~~FYI, the driver documented here WILL crash on OSX Sierra.~~

~~Make sure to use this one instead: https://github.com/MPorsley/ch340g-ch34g-ch34x-mac-os-x-driver~~

~~I can personally not test on MacOS and can not vouch for the above drivers at github, but there you go.~~

~~See uninstalling information at the bottom of the page if the driver causes problems for you.~~

~~**Here is an older version of the Mac driver, NOT FOR 10.12 Sierra**~~

~~(V1.0) Download the CH340 Macintosh Signed Driver for Mavericks (10.9), Yosemite (10.10) and El Capitan (10.11)~~

Linux

(Manufacturer's Chinese Info Link)

Drivers are almost certainly built into your Linux kernel already and it will probably just work as soon as you plug it in. If not you can download the Linux CH340 Driver (but I'd recommend just upgrading your Linux install so that you get the "built in" one).

Uninstalling From Macinstosh

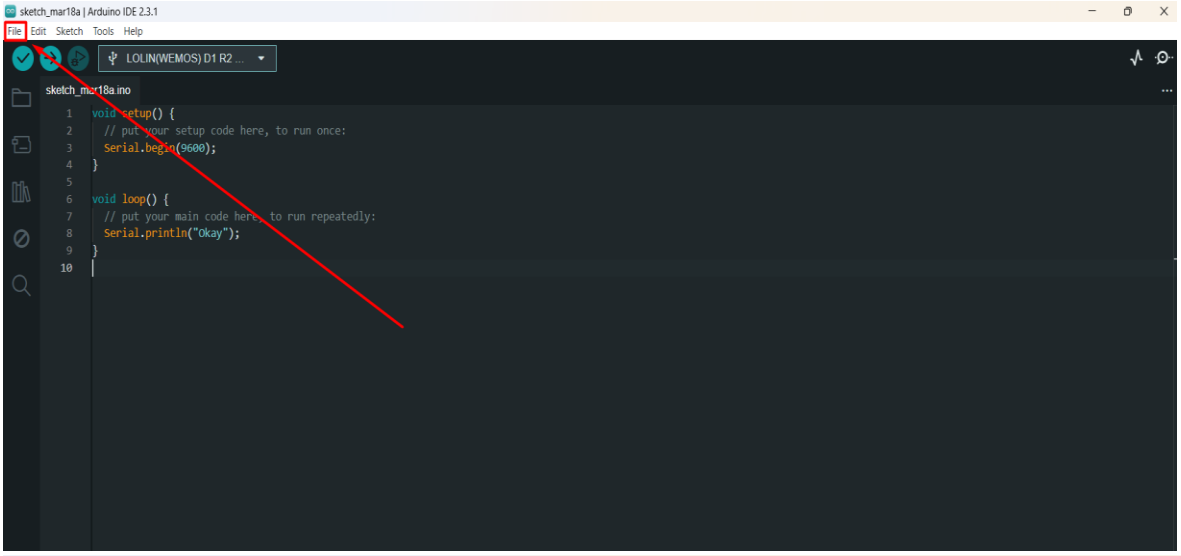
A couple of Mac users have said "the driver crashes my mac Sierra how do I uninstall".

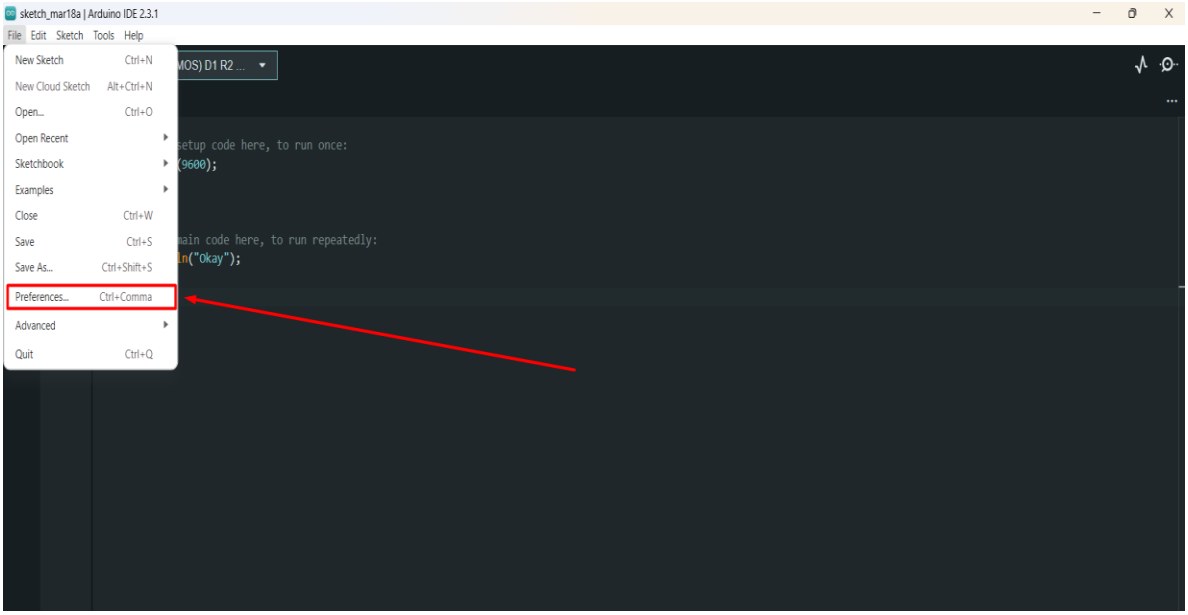
Firstly regards the crash make sure you installed the 1.3 version of the driver, not the old one. You could download it directly from the Manufacturers Website in case they have issued an update since I wrote this page.

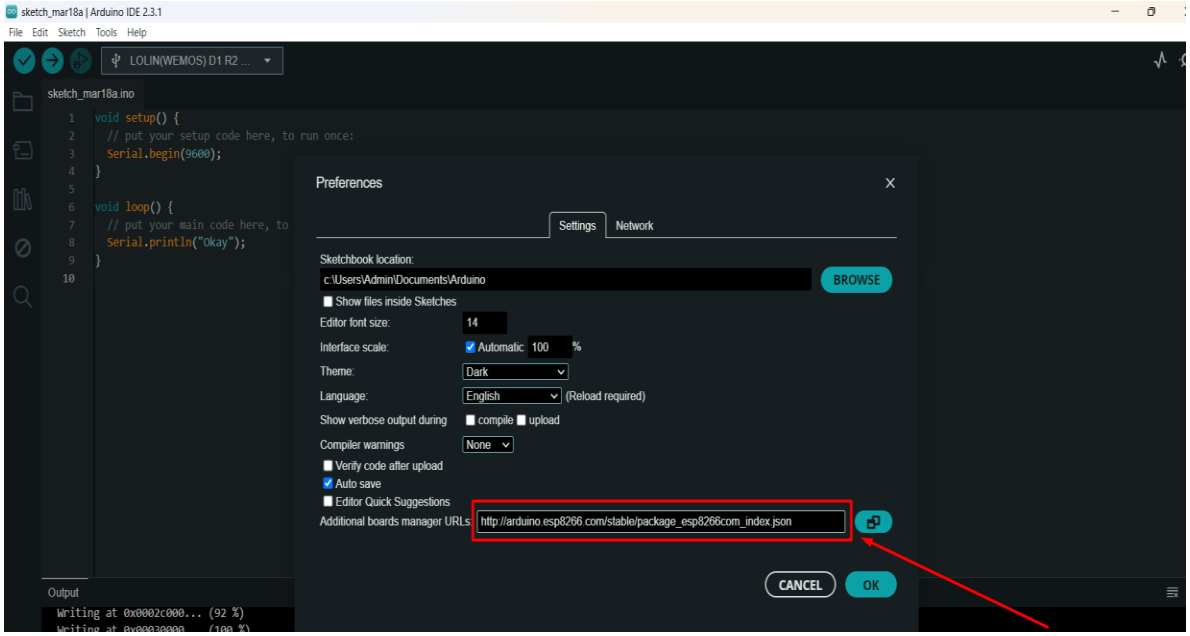
Secondly a quick google search reveals that this is how you uninstall:

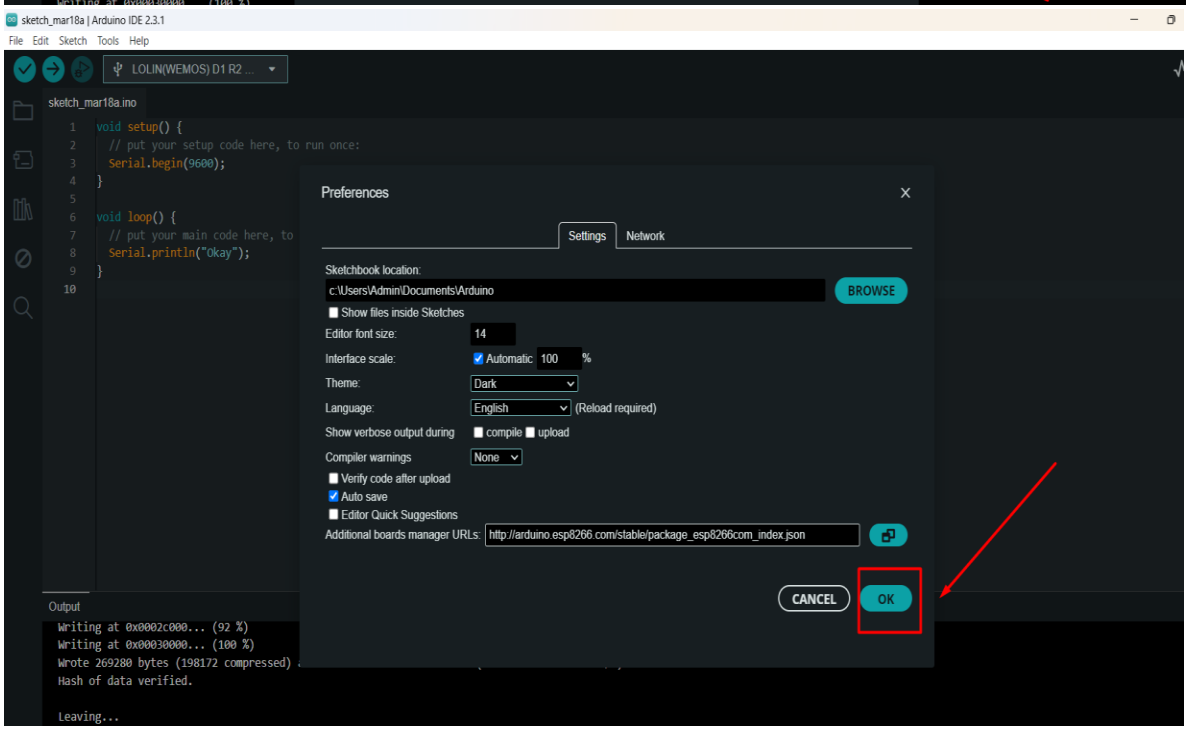
You can choose based on your operating system either Windows or Linux. After that, follow the instructions given below.

3. Install your esp32 on your Arduino IDE. Kindy open your Arduino IDE and go to File > Preferences > Copy and paste the following URL into the box: "https://dl.espressif.com/dl/package_esp32_index.json"

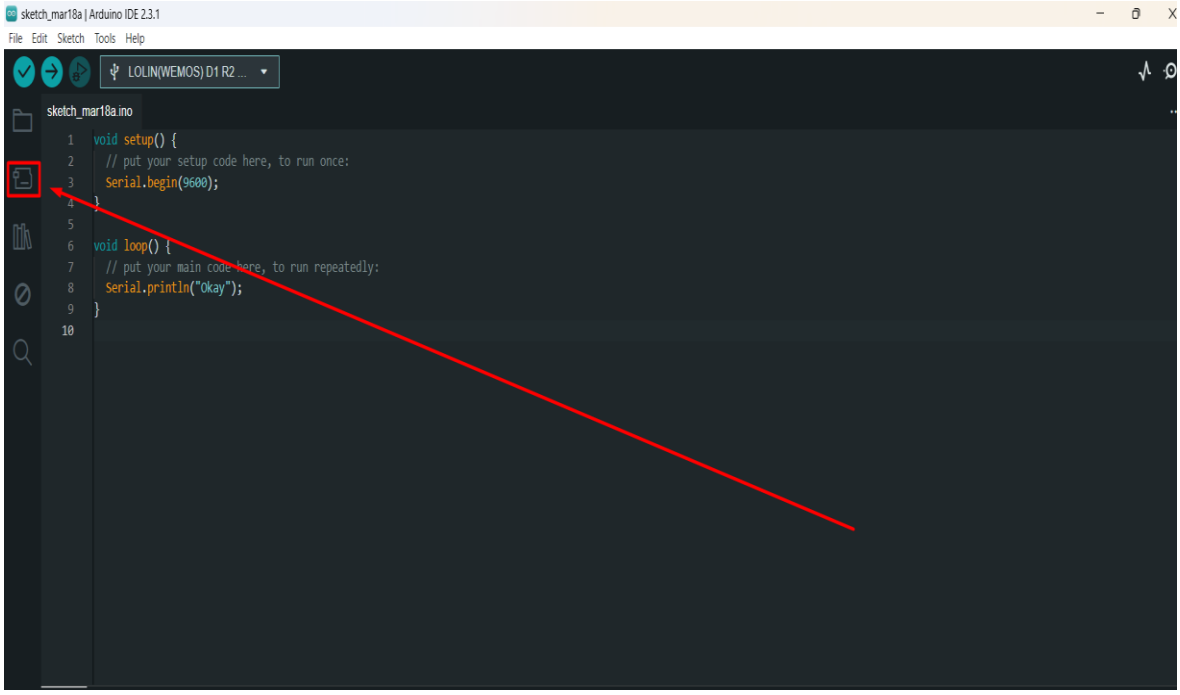

sketch_mar18a | Arduino IDE 2.3.1
File Edit Sketch Tools Help
LOLIN(WEMOS) D1 R2 ...
sketch_mar18a.ino
void setup() {
// put your setup code here, to run once:
Serial.begin(9600);
}
void loop() {
// put your main code here, to run repeatedly:
Serial.println("Okay");
}

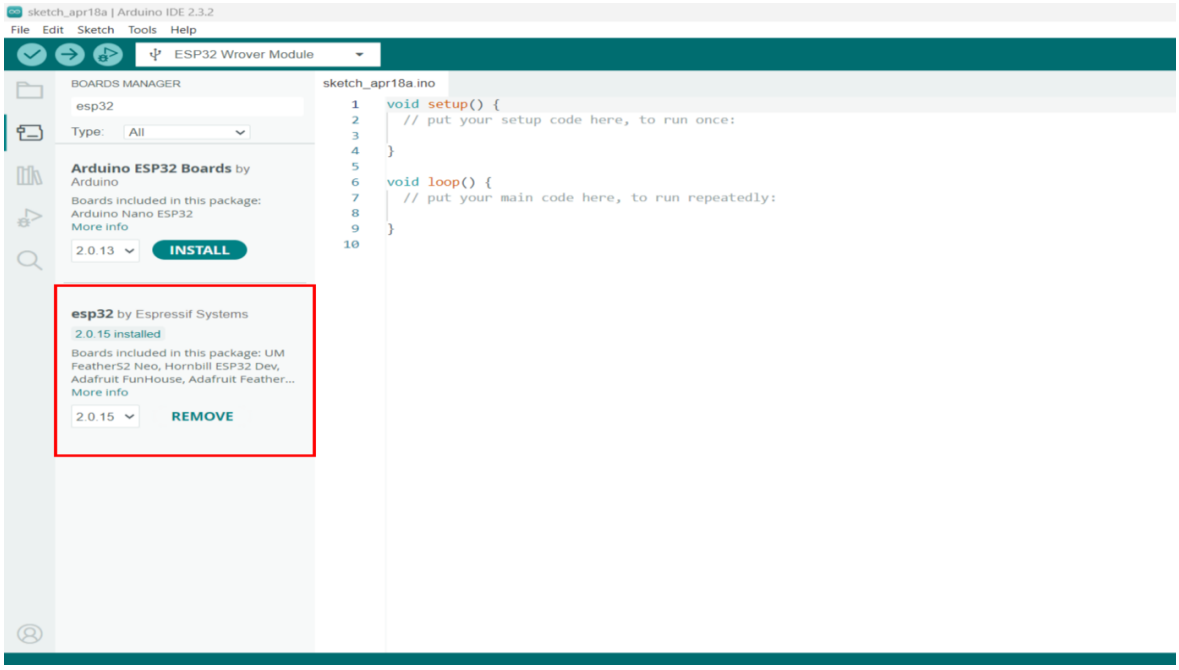

sketch_mar18a | Arduino IDE 2.3.1
File Edit Sketch Tools Help
New Sketch Ctrl+N
New Cloud Sketch Alt+Ctrl+N
Open... Ctrl+O
Open Recent
Sketchbook
Examples
Close Ctrl+W
Save Ctrl+S
Save As... Ctrl+Shift+S
Preferences... Ctrl+Comma
Advanced
Quit Ctrl+Q





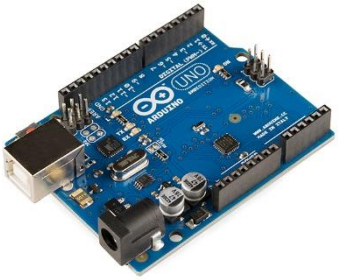
4. Go to the Board Manager search “ESP32” and click install.



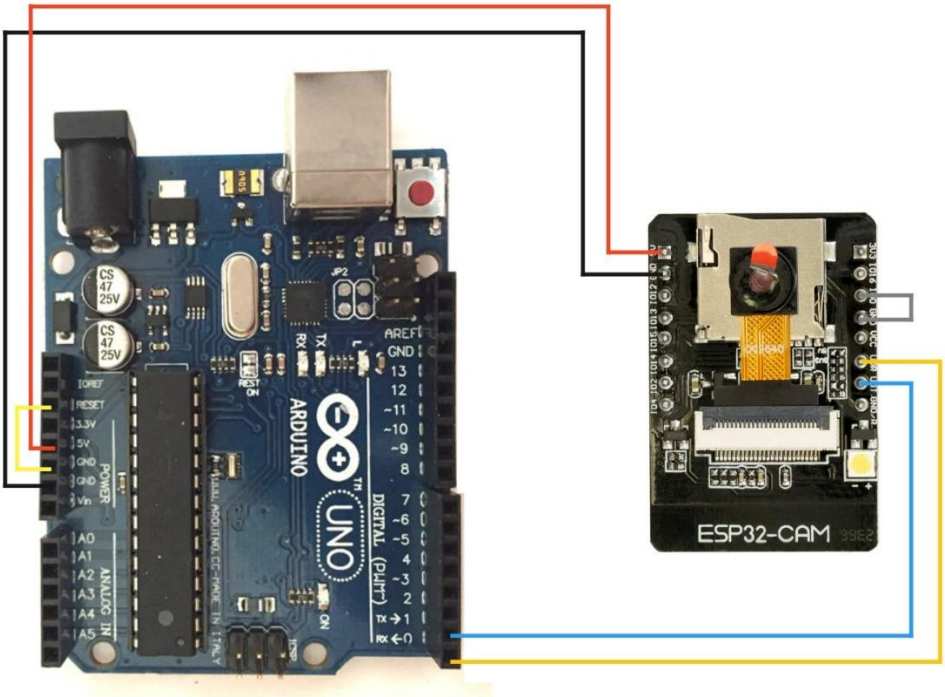


If you see the “REMOVE”, it means the board is already downloaded on your Arduino IDE.

- Materials:

Quantity	Materials	Picture
1	ESP32CAM	
1	ARDUINO UNO	

- Circuit Diagram



Modules/Sensors	Pins	Arduino UNO Pins
ESP32 CAM	5V	5V
	GND	GND
	GND	
	IO0	
	UOR	1 - TX
	UOT	0 - RX

- ## Code

You can download the code at this link: https://drive.google.com/drive/folders/1-UpWp-PX1EHekZiBcduuwuJ1kru_TbLk

Or you can copy and paste.

```
#include "esp_camera.h"
#include <WiFi.h>
#include "DHT.h"
#include <ArduinoWebsockets.h>
#define CAMERA_MODEL_AI_THINKER
#include <stdio.h>
#include "camera_pins.h"

#define DHT_PIN 2
#define FLASH_PIN 4

const char* ssid = "SSID"; // Your wifi name like "MyWifi"
const char* password = "PASSWORD"; // Your password to the wifi
network like "password"
const char* websocket_server_host = "SERVER HOST";
const uint16_t websocket_server_port1 = SERVER PORT;

float hmem = 0;
float tmem = 0;
int flashlight = 0;

DHT dht(DHT_PIN, DHT11);
using namespace websockets;
WebsocketsClient client;

void onEventsCallback(WebsocketsEvent event, String data) {
    if(event == WebsocketsEvent::ConnectionOpened) {
        Serial.println("Connection Opened");
    } else if(event == WebsocketsEvent::ConnectionClosed) {
        Serial.println("Connection Closed");
        ESP.restart();
```

```
    } else if(event == WebsocketsEvent::GotPing) {
        Serial.println("Got a Ping!");
    } else if(event == WebsocketsEvent::GotPong) {
        Serial.println("Got a Pong!");
    }
}

void onMessageCallback(WebsocketsMessage message) {
    String data = message.data();
    int index = data.indexOf("=");
    if(index != -1) {
        String key = data.substring(0, index);
        String value = data.substring(index + 1);

        if(key == "ON_BOARD_LED_1") {
          if(value.toInt() == 1) {
            flashlight = 1;
            digitalWrite(FLASH_PIN, HIGH);
          } else {
            flashlight = 0;
            digitalWrite(FLASH_PIN, LOW);
          }
        }

        Serial.print("Key: ");
        Serial.println(key);
        Serial.print("Value: ");
        Serial.println(value);
    }
}

void setup()
{
  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
```

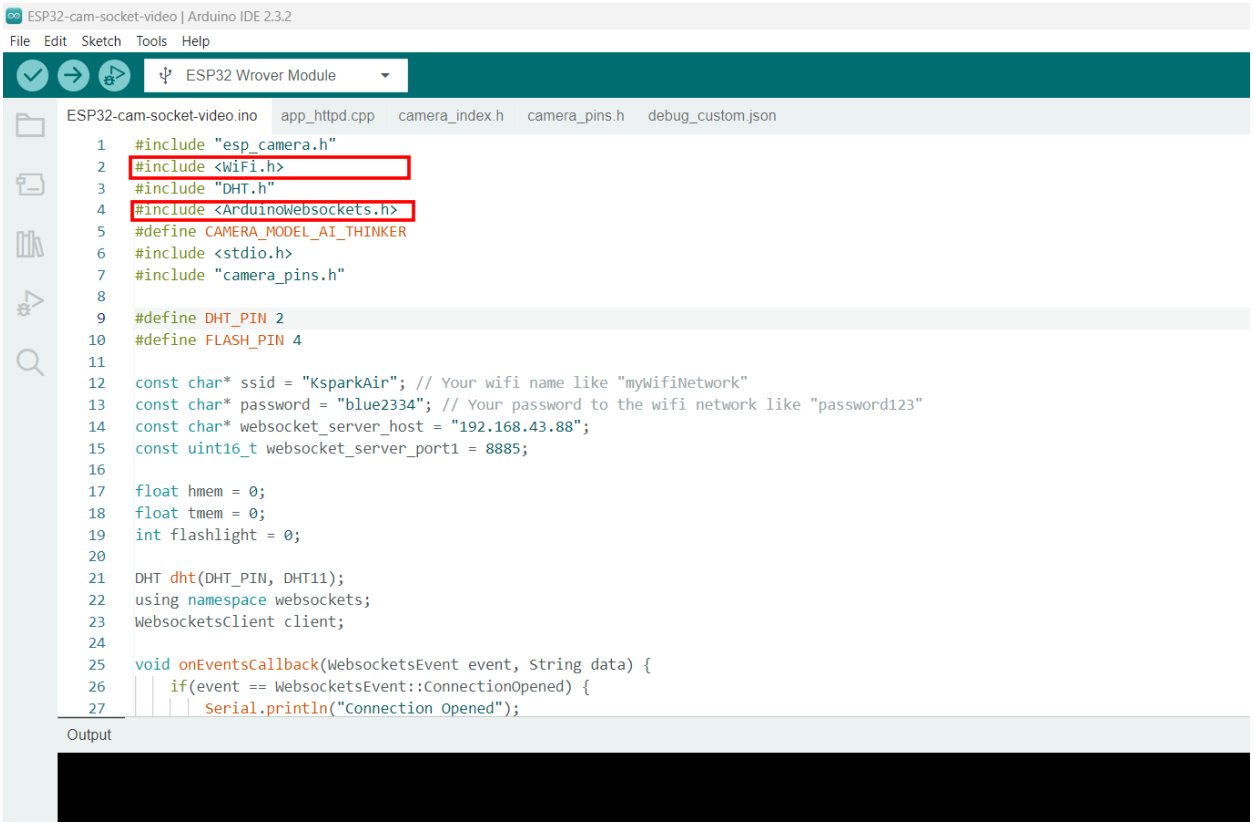
```
config.pin_d3 = Y5_GPIO_NUM;
config.pin_d4 = Y6_GPIO_NUM;
config.pin_d5 = Y7_GPIO_NUM;
config.pin_d6 = Y8_GPIO_NUM;
config.pin_d7 = Y9_GPIO_NUM;
config.pin_xclk = XCLK_GPIO_NUM;
config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sscb_sda = SIOD_GPIO_NUM;
config.pin_sscb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;

config.xclk_freq_hz = 10000000;
config.pixel_format = PIXFORMAT_JPEG;
config.frame_size = FRAMESIZE_SVGA;
config.jpeg_quality = 40;
config.fb_count = 2;

// camera init
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) { return; }

sensor_t * s = esp_camera_sensor_get();

s->set_contrast(s, 0);
s->set_raw_gma(s, 1);

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) { delay(500); }

dht.begin();
pinMode(FLASH_PIN, OUTPUT);
client.onMessage(onMessageCallback);
client.onEvent(onEventsCallback);

Serial.begin(115200);
```

```
  while(!client.connect(websocket_server_host,
websocket_server_port1, "/")) { delay(500); }
}

void loop()
{
  client.poll();
  camera_fb_t *fb = esp_camera_fb_get();
  if(!fb)
  {
    esp_camera_fb_return(fb);
    return;
  }

  if(fb->format != PIXFORMAT_JPEG) { return; }

  client.sendBinary((const char*) fb->buf, fb->len);
  esp_camera_fb_return(fb);

  float h = dht.readHumidity();
  float t = dht.readTemperature();

  if(isnan(h)) {
    h = hmem;
  } else {
    hmem = h;
  }

  if(isnan(t)) {
    t = tmem;
  } else {
    tmem = t;
  }

  String output = "temp=" + String(t, 2) + ",hum=" + String(h, 2)
+ ",light=12;state:ON_BOARD_LED_1=" + String(flashlight);
  Serial.println(output);

  client.send(output);
```

```
}
```

- # Instructions

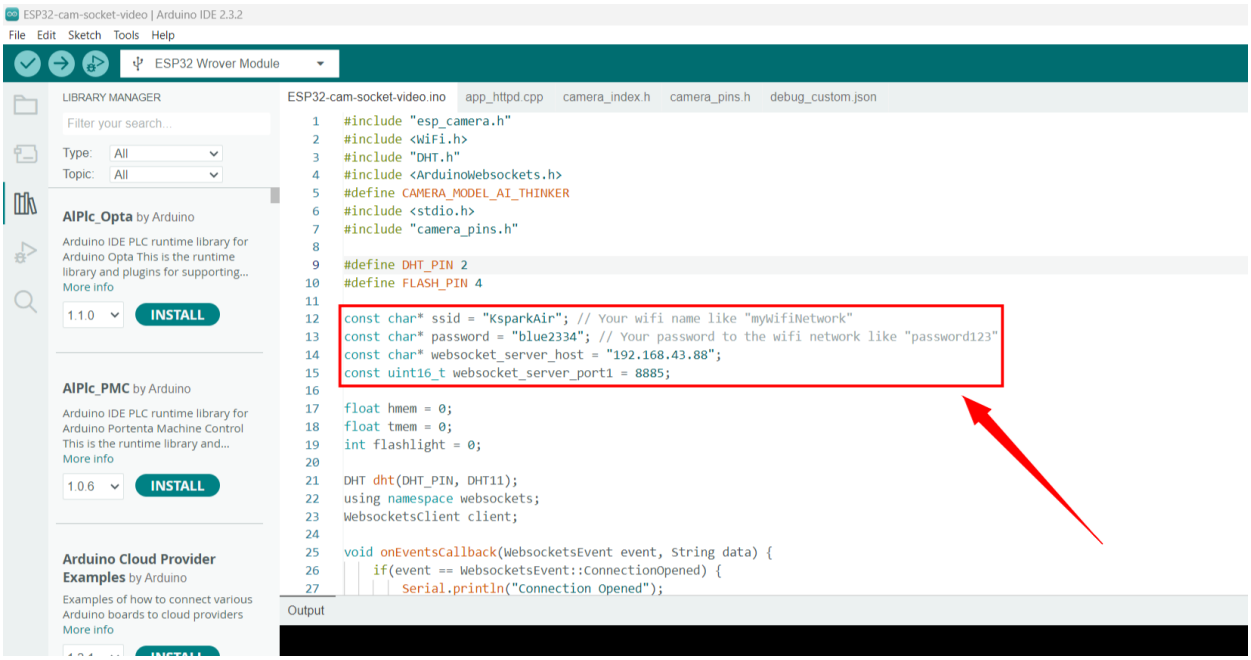
1. Add libraries:
    1. DHT.h (You can add others,not necessary)
    2. Wifi.h (Necessary)
    3. ArduinoWebsockets.h (Necessary)



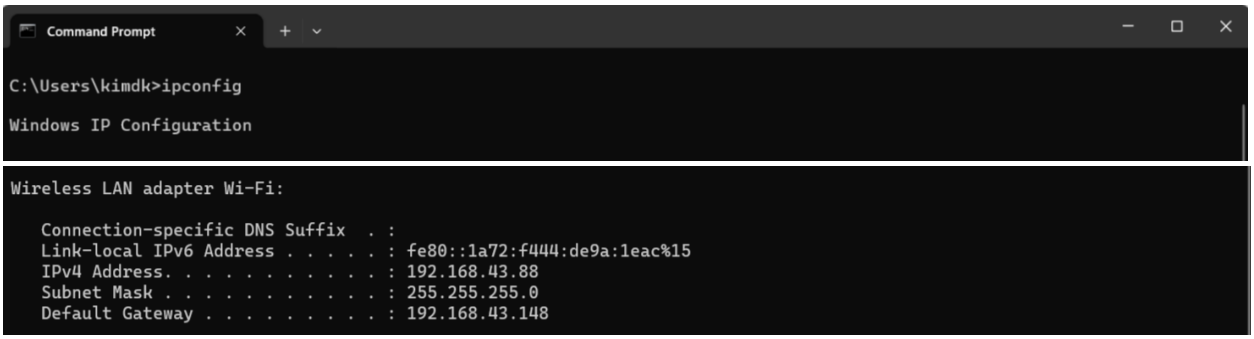
You can find some libraries in this link:
https://drive.google.com/drive/folders/1Cf2vV2rFE9zKqbL8USwOPdw8hjU6dfhN?usp=sharing

Or you can search in the Library Manager in the Arduino IDE.

2. Before uploading the code, you need to set up some things. First, you need to change the ssid[ ] and pass[ ] based on the WiFi to be connected.



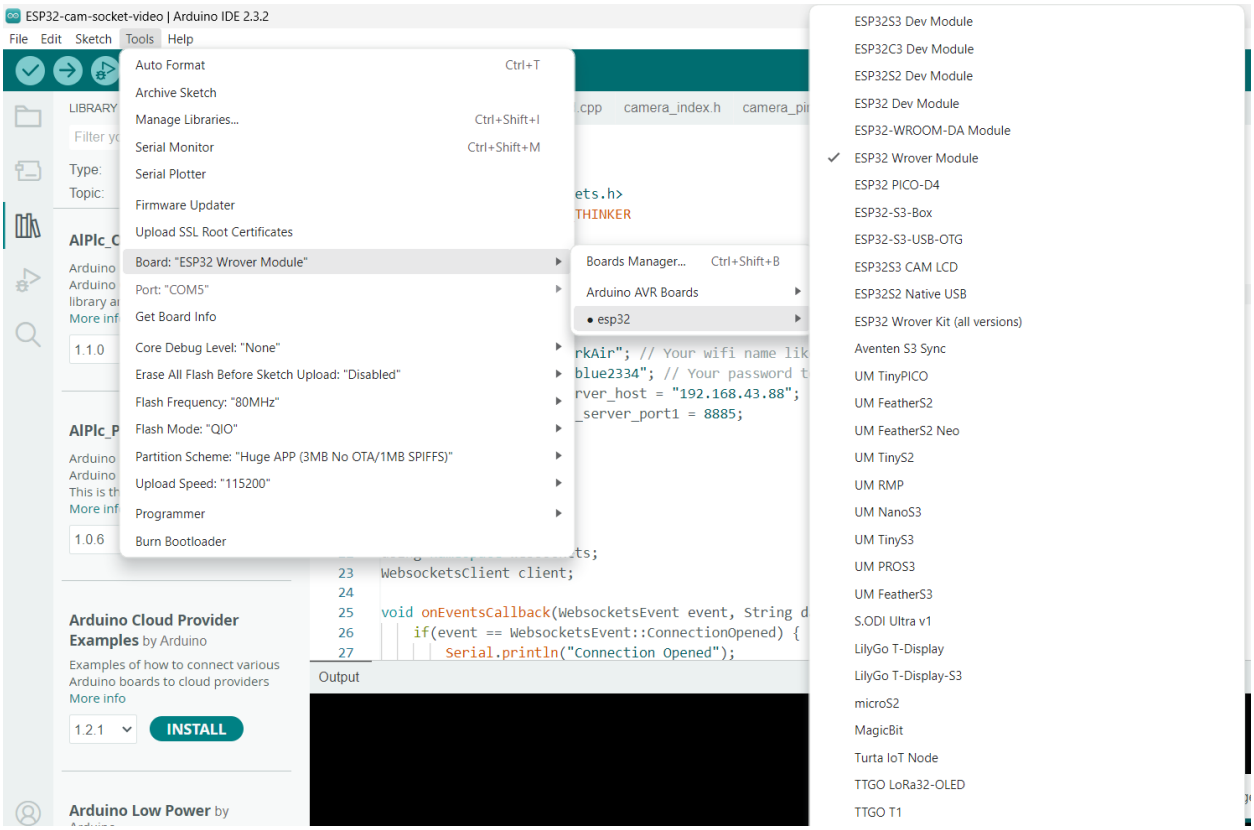
Note: Your websocket server host can be found in your network connection by simply typing “ipconfig” in your cmd and copy the IPV4 address in the following list.



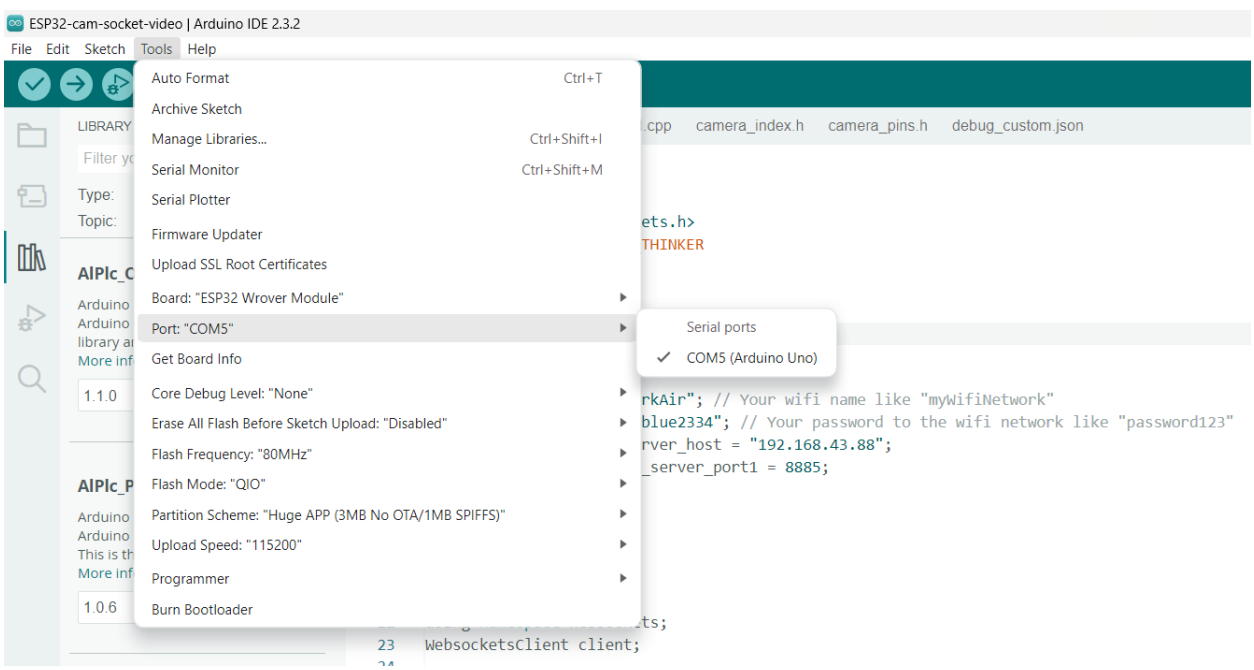
- Upload the code to the microcontroller.

After the set up the WiFi. Go to **Tools** > **Board** > **ESP32** > **ESP32 WROVER**

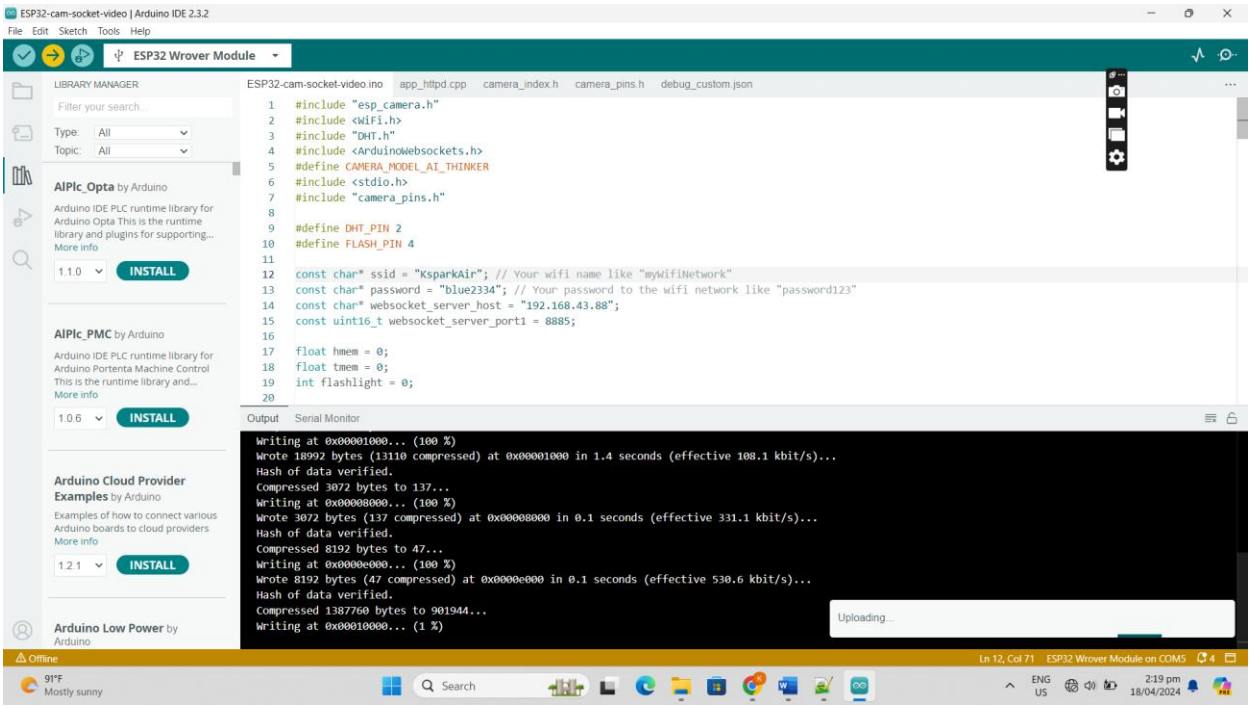
MODULE

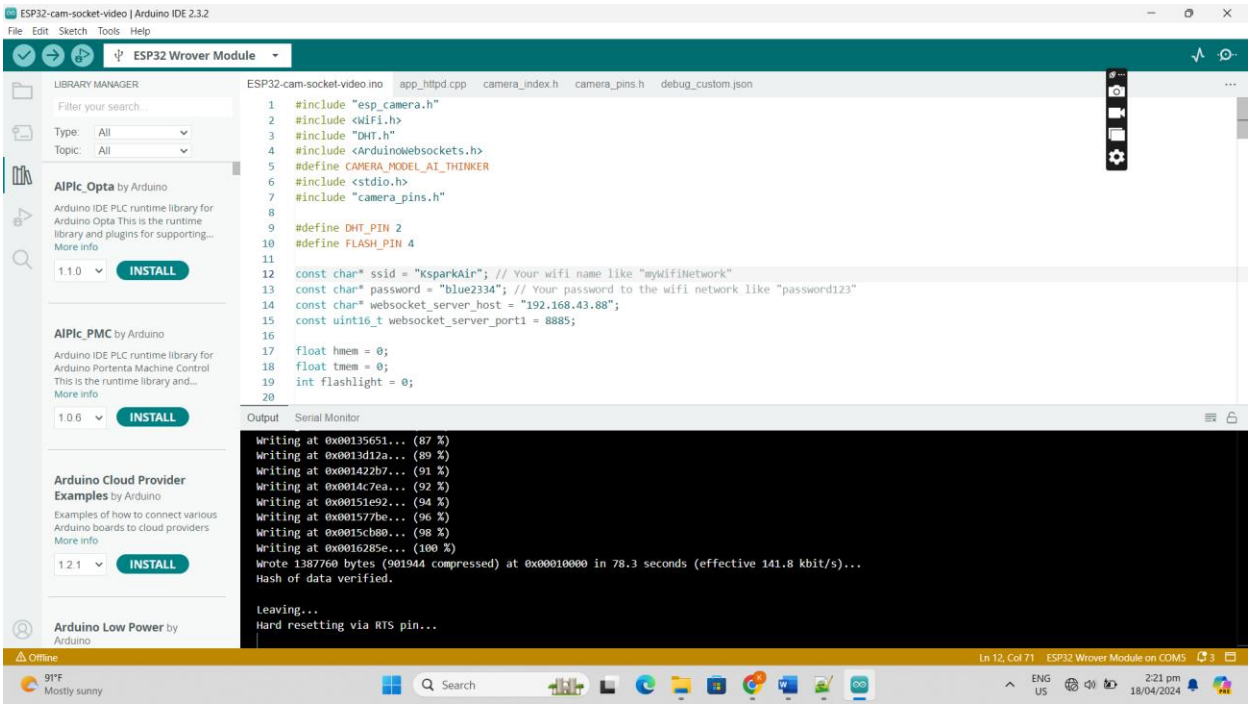


Next, go to **Tool** > **Port** > **Serial ports** and select the com(number) for the Arduino Uno microcontroller.



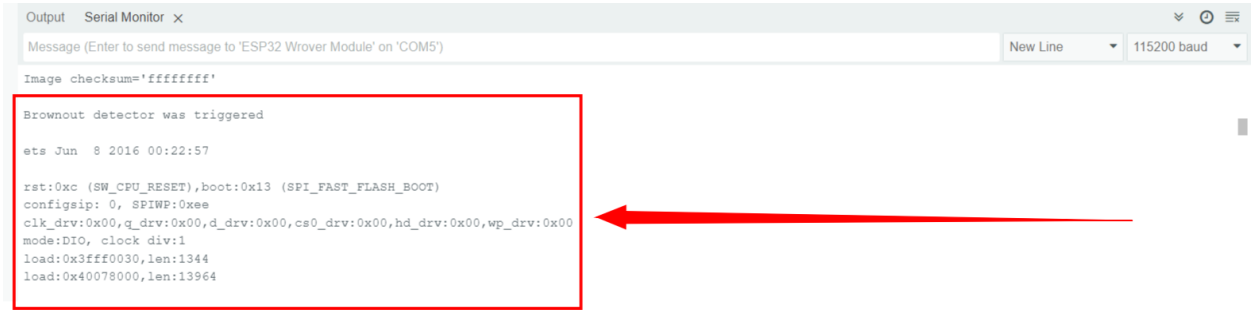
- Result


ESP32-cam-socket-video | Arduino IDE 2.3.2
File Edit Sketch Tools Help
ESP32 Wrover Module
LIBRARY MANAGER
Filter your search...
Type: All
Topic: All
AIPlc_Opta by Arduino
Arduino IDE PLC runtime library for Arduino Opta This is the runtime library and plugins for supporting...
More info
1.1.0 INSTALL
AIPlc_PMC by Arduino
Arduino IDE PLC runtime library for Arduino Portenta Machine Control This is the runtime library and...
More info
1.0.6 INSTALL
Arduino Cloud Provider Examples by Arduino
Examples of how to connect various Arduino boards to cloud providers
More info
1.2.1 INSTALL
Arduino Low Power by Arduino
ESP32-cam-socket-video.ino app_httpd.cpp camera_index.h camera_pins.h debug_custom.json
#include "esp_camera.h"
#include <WiFi.h>
#include "DHT.h"
#include <ArduinoWebsockets.h>
#define CAMERA_MODEL_AI_THINKER
#include <stdio.h>
#include "camera_pins.h"
#define DHT_PIN 2
#define FLASH_PIN 4
const char* ssid = "KsparkAir"; // Your wifi name like "myWifiNetwork"
const char* password = "blue2334"; // Your password to the wifi network like "password123"
const char* websocket_server_host = "192.168.43.88";
const uint16_t websocket_server_port1 = 8885;
float hmem = 0;
float tmem = 0;
int flashlight = 0;
Output Serial Monitor
Writing at 0x00001000... (100 %)
Wrote 18992 bytes (13110 compressed) at 0x00001000 in 1.4 seconds (effective 108.1 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 137...
Writing at 0x00008000... (100 %)
Wrote 3072 bytes (137 compressed) at 0x00008000 in 0.1 seconds (effective 331.1 kbit/s)...
Hash of data verified.
Compressed 8192 bytes to 47...
Writing at 0x0000e000... (100 %)
Wrote 8192 bytes (47 compressed) at 0x0000e000 in 0.1 seconds (effective 530.6 kbit/s)...
Hash of data verified.
Compressed 1387760 bytes to 901944...
Writing at 0x00010000... (1 %)
Uploading...
Offline
Ln 12, Col 71 ESP32 Wrover Module on COM5
91°F Mostly sunny
Search
ENG US
2:19 pm 18/04/2024


ESP32-cam-socket-video | Arduino IDE 2.3.2
File Edit Sketch Tools Help
ESP32 Wrover Module
LIBRARY MANAGER
Filter your search...
Type: All
Topic: All
AIPlc_Opta by Arduino
Arduino IDE PLC runtime library for Arduino Opta This is the runtime library and plugins for supporting...
More info
1.1.0 INSTALL
AIPlc_PMC by Arduino
Arduino IDE PLC runtime library for Arduino Portenta Machine Control This is the runtime library and...
More info
1.0.6 INSTALL
Arduino Cloud Provider Examples by Arduino
Examples of how to connect various Arduino boards to cloud providers
More info
1.2.1 INSTALL
Arduino Low Power by Arduino
ESP32-cam-socket-video.ino app_httpd.cpp camera_index.h camera_pins.h debug_custom.json
#include "esp_camera.h"
#include <WiFi.h>
#include "DHT.h"
#include <ArduinoWebsockets.h>
#define CAMERA_MODEL_AI_THINKER
#include <stdio.h>
#include "camera_pins.h"
#define DHT_PIN 2
#define FLASH_PIN 4
const char* ssid = "KsparkAir"; // Your wifi name like "myWifiNetwork"
const char* password = "blue2334"; // Your password to the wifi network like "password123"
const char* websocket_server_host = "192.168.43.88";
const uint16_t websocket_server_port1 = 8885;
float hmem = 0;
float tmem = 0;
int flashlight = 0;
Output Serial Monitor
Writing at 0x00135651... (87 %)
Writing at 0x0013d12a... (89 %)
Writing at 0x001422b7... (91 %)
Writing at 0x0014c7ea... (92 %)
Writing at 0x00151e92... (94 %)
Writing at 0x001577be... (96 %)
Writing at 0x0015cb80... (98 %)
Writing at 0x0016285e... (100 %)
Wrote 1387760 bytes (901944 compressed) at 0x00010000 in 78.3 seconds (effective 141.8 kbit/s)...
Hash of data verified.
Leaving...
Hard resetting via RTS pin...
Offline
Ln 12, Col 71 ESP32 Wrover Module on COM5
91°F Mostly sunny
Search
ENG US
2:21 pm 18/04/2024

Now after the *“Hard resetting via RTS pin”*, remove the jumper wire connected IO0 and GND, and press reset button in the ESP32CAM for it to re-connect to the Wi-Fi network.

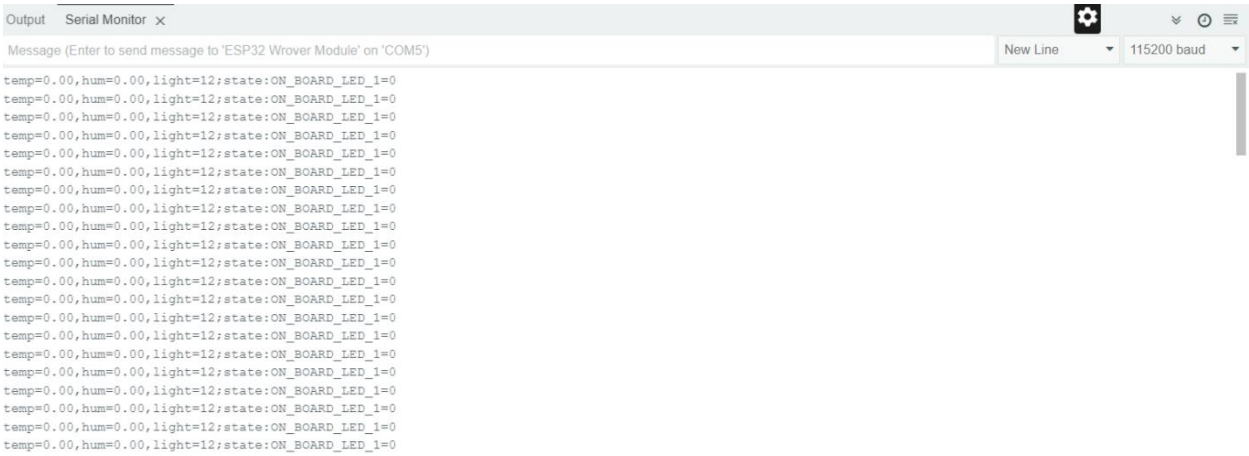
If the display of arduino serial port is this:



```
Image checksum='ffffffff'

Brownout detector was triggered

ets Jun  8 2016 00:22:57

rst:0xc (SW_CPU_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0030,len:1344
load:0x40078000,len:13964
```

It means that you have pressed the reset button and the ESP32CAM have connect to the Wi-Fi network attempts to established a connection to the node.js server.

The example display below on Arduino will only show if the connection between ESP32CAM and node server.js established a connection.



```
temp=0.00,hum=0.00,light=12;state:ON_BOARD_LED_1=0
temp=0.00,hum=0.00,light=12;state:ON_BOARD_LED_1=0
temp=0.00,hum=0.00,light=12;state:ON_BOARD_LED_1=0
temp=0.00,hum=0.00,light=12;state:ON_BOARD_LED_1=0
temp=0.00,hum=0.00,light=12;state:ON_BOARD_LED_1=0
temp=0.00,hum=0.00,light=12;state:ON_BOARD_LED_1=0
temp=0.00,hum=0.00,light=12;state:ON_BOARD_LED_1=0
temp=0.00,hum=0.00,light=12;state:ON_BOARD_LED_1=0
temp=0.00,hum=0.00,light=12;state:ON_BOARD_LED_1=0
temp=0.00,hum=0.00,light=12;state:ON_BOARD_LED_1=0
temp=0.00,hum=0.00,light=12;state:ON_BOARD_LED_1=0
temp=0.00,hum=0.00,light=12;state:ON_BOARD_LED_1=0
temp=0.00,hum=0.00,light=12;state:ON_BOARD_LED_1=0
temp=0.00,hum=0.00,light=12;state:ON_BOARD_LED_1=0
temp=0.00,hum=0.00,light=12;state:ON_BOARD_LED_1=0
temp=0.00,hum=0.00,light=12;state:ON_BOARD_LED_1=0
temp=0.00,hum=0.00,light=12;state:ON_BOARD_LED_1=0
temp=0.00,hum=0.00,light=12;state:ON_BOARD_LED_1=0
temp=0.00,hum=0.00,light=12;state:ON_BOARD_LED_1=0
temp=0.00,hum=0.00,light=12;state:ON_BOARD_LED_1=0
temp=0.00,hum=0.00,light=12;state:ON_BOARD_LED_1=0
```

2.VS CODE SETUP FOR REACT w/ NODE.JS

Install the requirements:

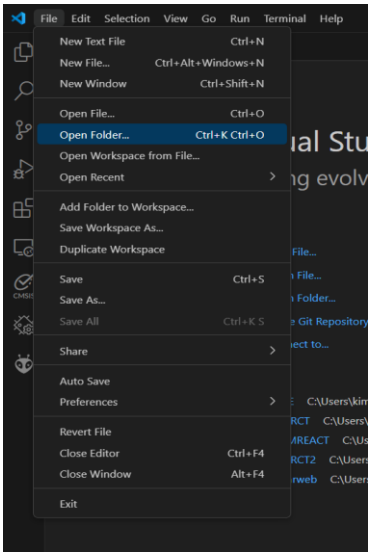
VS CODE DOWNLOAD: https://code.visualstudio.com/download
NODE.JS DOWNLOAD: https://nodejs.org/en/download (better to download the latest or current version)

A. NODE JS SETUP

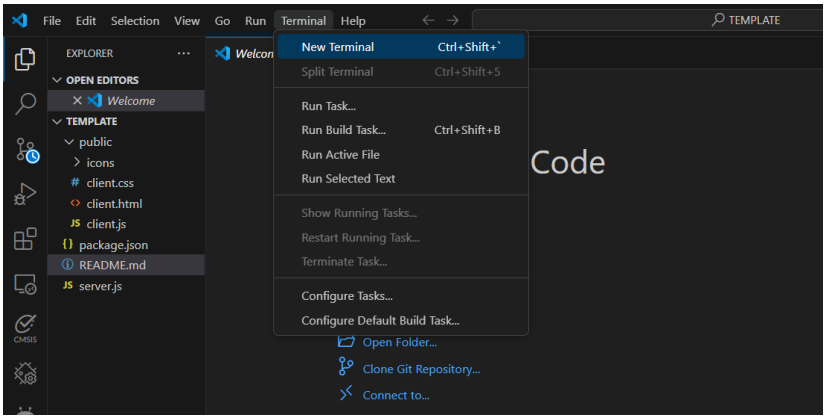
1. Download the template folder in the gdrive link below:

   https://drive.google.com/drive/folders/1yxbzzZZi7A591ECYjhxDYMLLoS7ZlAbY?usp=drive_link

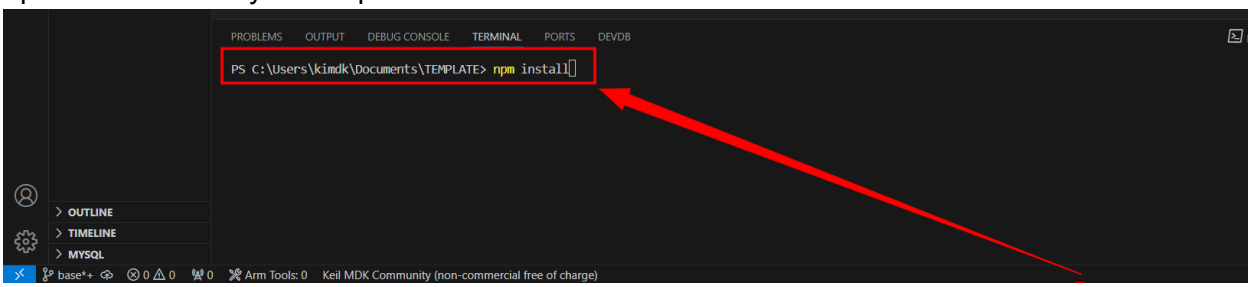
2. Now in VS code studio, open the download folder,



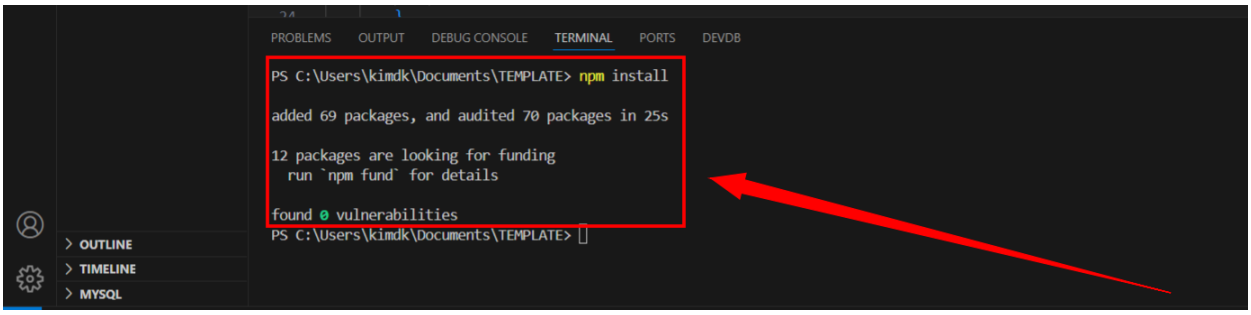
3. Click new Terminal



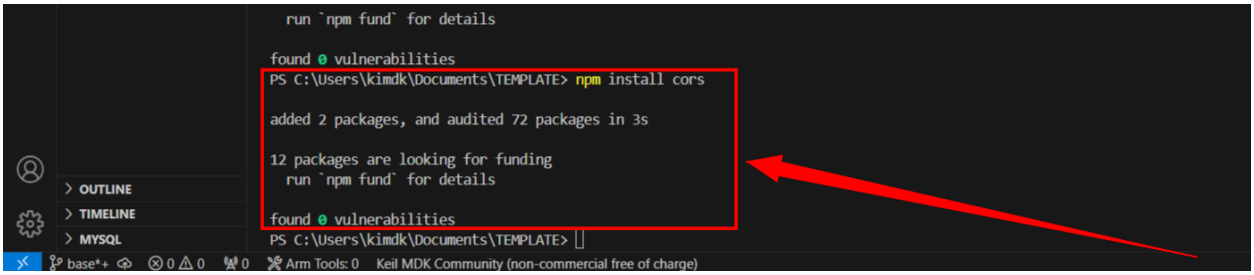
4. Now check your path in the terminal if it is correct , but in any case if not, type " cd <path of the folder you've opened>"



5. Type "npm install" (it would automatically download the necessary modules for node js server).



6. Type "npm install cors" (to prevent encountering Cross-Origin Resource Sharing (CORS) issues when fetching resources from the server running the React app.



7. Now modify the code in "client.js" and use the web server host IP address used in your ESP32CAM

```
const ws = new WebSocket('ws:///192.168.43.88:8999');

ws.addEventListener('open', (event) => {
    ws.send(JSON.stringify({
        'client': '8999',
        'operation': 'connecting',
        'data': {}
    }));
});

ws.onmessage = message => {
    let md = JSON.parse(message.data);

    for (const device in md.devices) {
        if (!document.querySelector('#' + device)) {
            document.querySelector('#main-wrapper')
                .appendChild(createElement('div',{ id: device, class: md.devices[device].class + ' item' }))
                .appendChild(createElement('h2',{ id: device + '-header', class: 'sensors-header' }, md.devices[device].display));
            document.querySelector('#'+device)
                .appendChild(createElement('div',{ id:'wrap-' + device + '-image', class: 'image-wrapper' }))
                .appendChild(createElement('img',{ id:'img-' + device }));
            document.querySelector('#'+device)
                .appendChild(createElement('div',{ id:'wrap-' + device + '-commands' }));
```

```
PS C:\Users\kimdk\Documents\TEMPLATE> npm install

added 69 packages, and audited 70 packages in 25s

12 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\kimdk\Documents\TEMPLATE>
```

8. Also modify the code in “server.js” and use the web server port used in your ESP32CAM

```
const path = require('path');
const express = require('express');
const WebSocket = require('ws');
const cors = require('cors');

const app = express();

app.use(cors());
app.use('/static', express.static(path.join(__dirname, 'public')));

let clients = [];

const HTTP_PORT = 8000;
let devices = {
    test1: {port:8885, class: 'cam-instance', display: 'Cam #1', view: 'overlay'},
};

process.on('uncaughtException', (error, origin) => {
    console.log('----- Uncaught exception -----');
    console.log(error);
    console.log('----- Exception origin -----');
    console.log(origin);
    console.log('----- Status -----');
```

```
PS C:\Users\kimdk\Documents\TEMPLATE> npm install

added 69 packages, and audited 70 packages in 25s

12 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\kimdk\Documents\TEMPLATE>
```

9. Now type “node server.js” or “npm start” to know if the server.js is running in the node js server.

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    DEVDB

added 2 packages, and audited 72 packages in 3s

12 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\kimdk\Documents\TEMPLATE> node server.js
```

10. Please check the web server port and host displayed in the terminal.

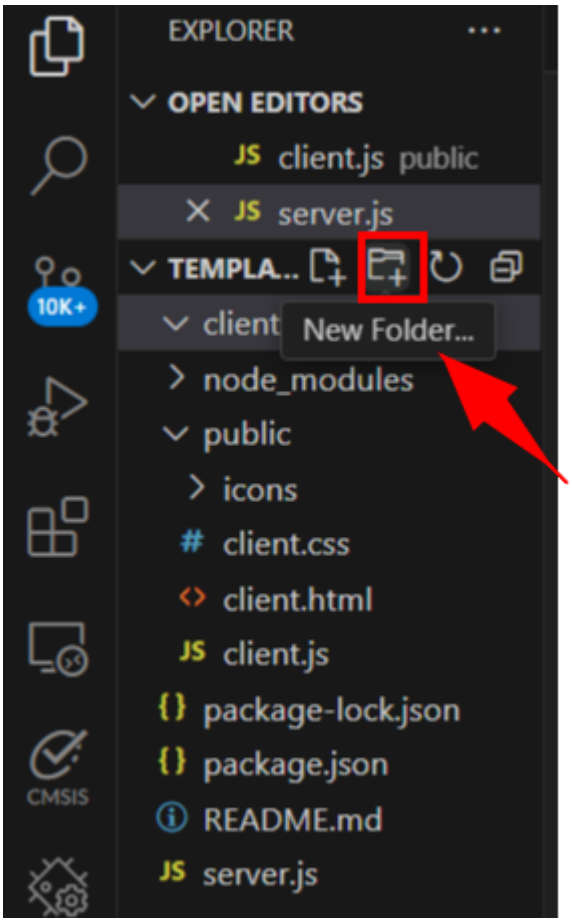
```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    DEVDB

added 2 packages, and audited 72 packages in 3s

12 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\kimdk\Documents\TEMPLATE> node server.js
WS Server is listening at 8999
WS Server is listening at 8885
HTTP server starting on 8000
```

11. Now to exit press “Ctrl + c “ to terminate or stop the server from running.

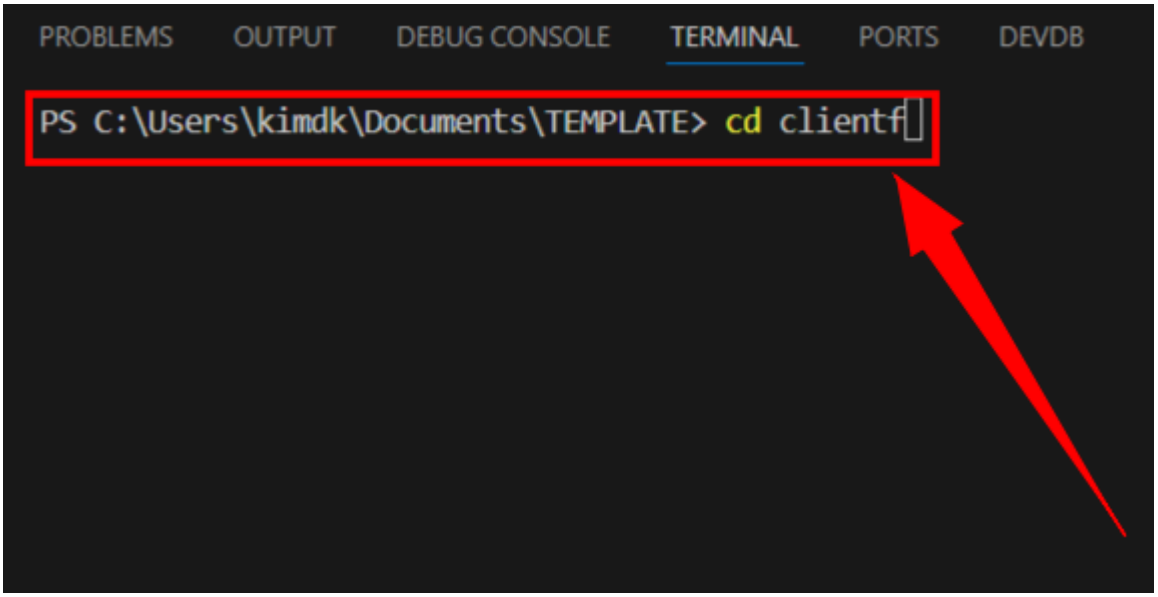
- Now you have already setup the NODE.JS in your folder
- Note: the wifi connection of your ESP32CAM must have the same wifi network on where your Node server.js is running.

B. **REACT SETUP**

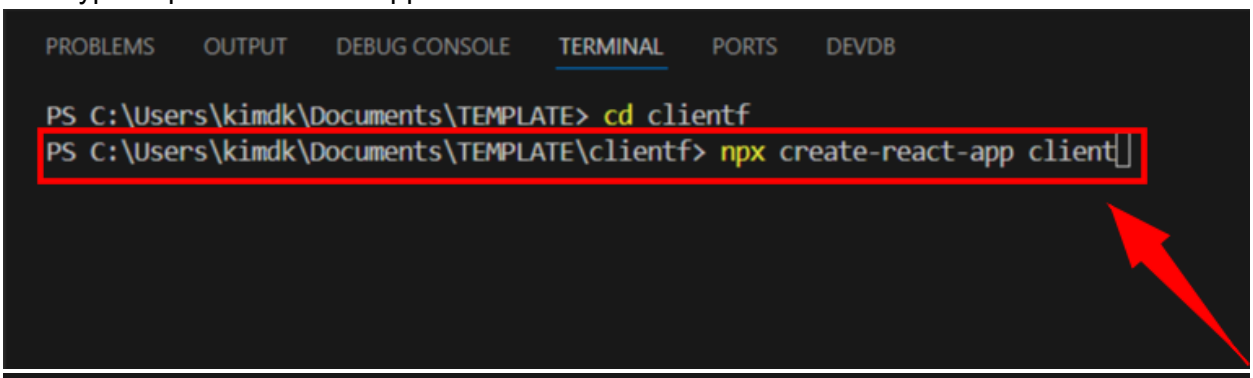
1. Create a folder named “Clientf” or any name you want to use.

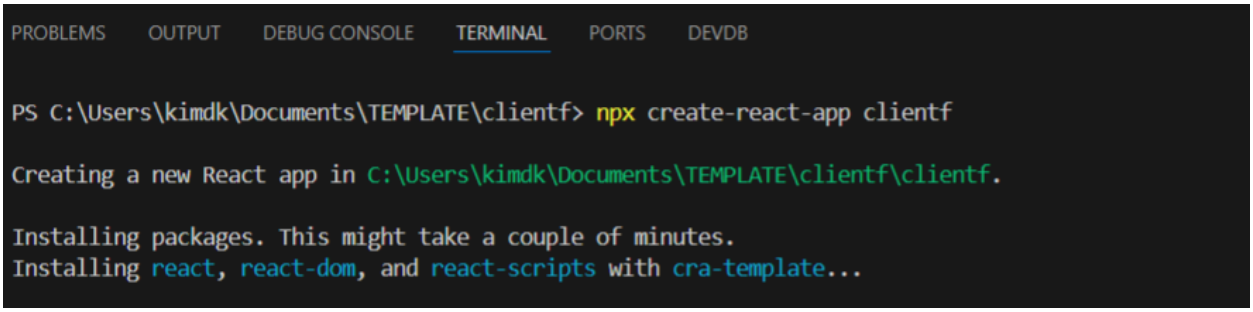


2. Click in new Terminal , type “cd clientf” (will change the path to the new created folder clientf)



3. Now type “ npx create-react-app client “





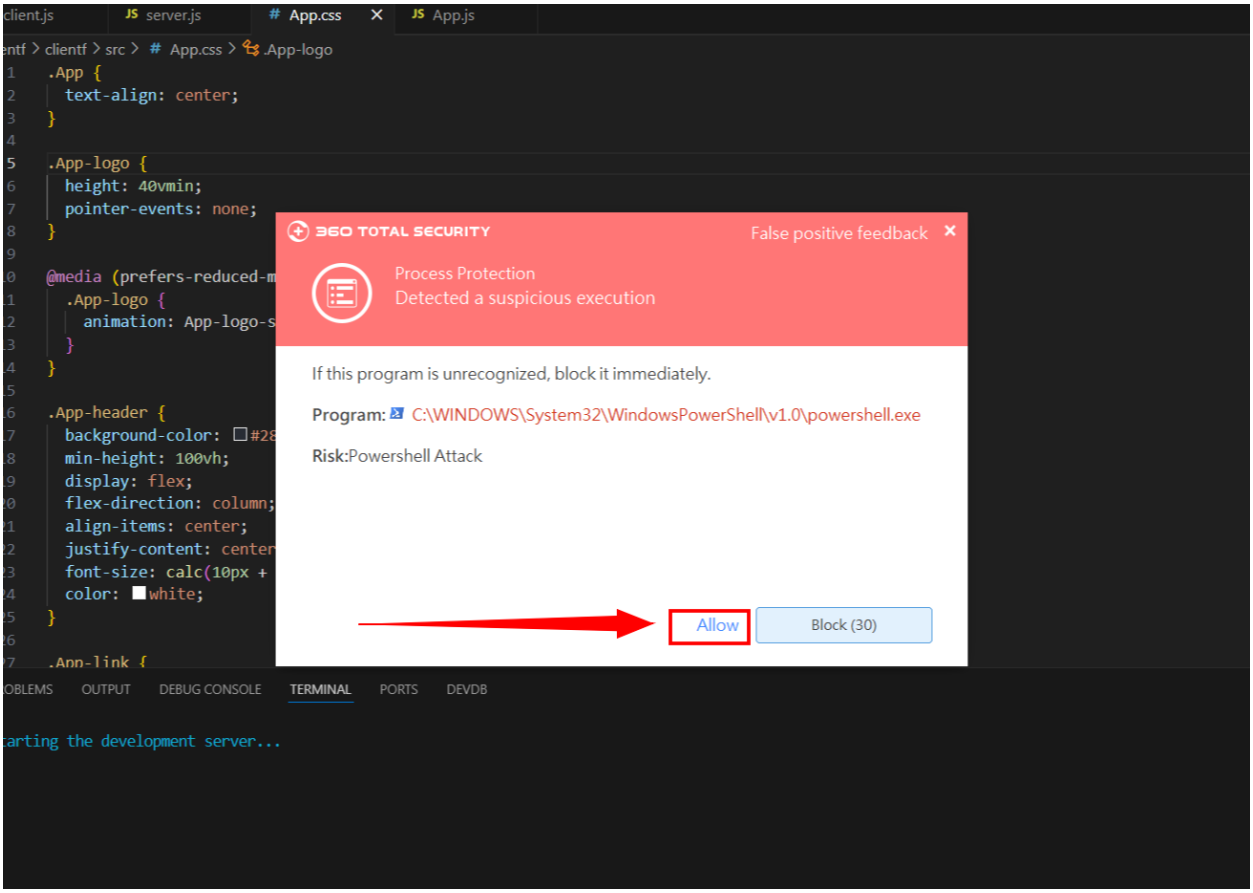
4. Now follow the picture shown below:

```
Run `npm audit` for details.

Success! Created clientf at C:\Users\kimdk\Documents\TEMPLATE\clientf\clientf
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

  cd clientf
  npm start

Happy hacking!
PS C:\Users\kimdk\Documents\TEMPLATE\clientf> cd clientf
```

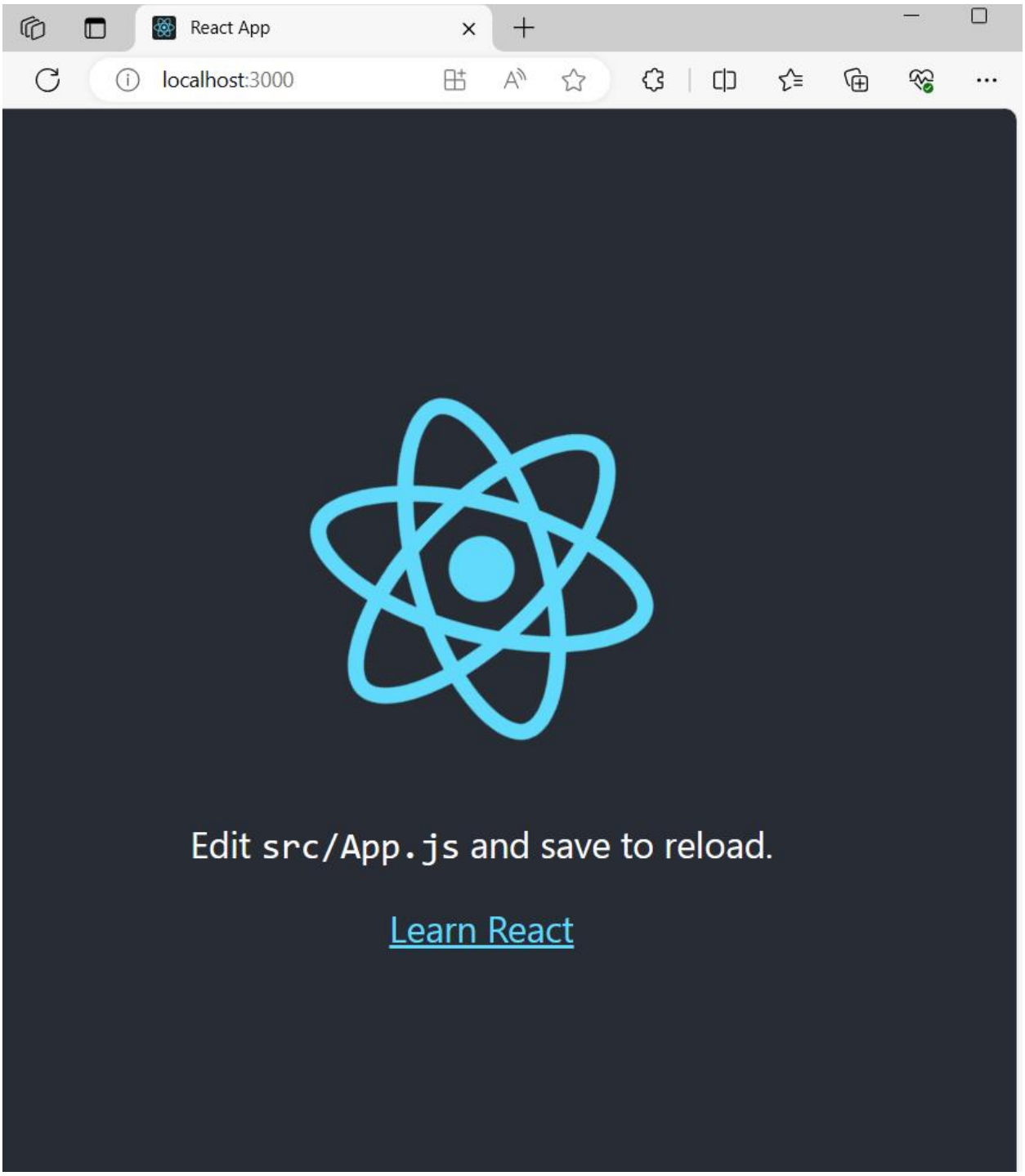
Result:

```
Compiled successfully!

You can now view clientf in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://10.10.33.216:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

Note: If you see this window (due to the antivirus software protection), click allow in order for the react app to run:

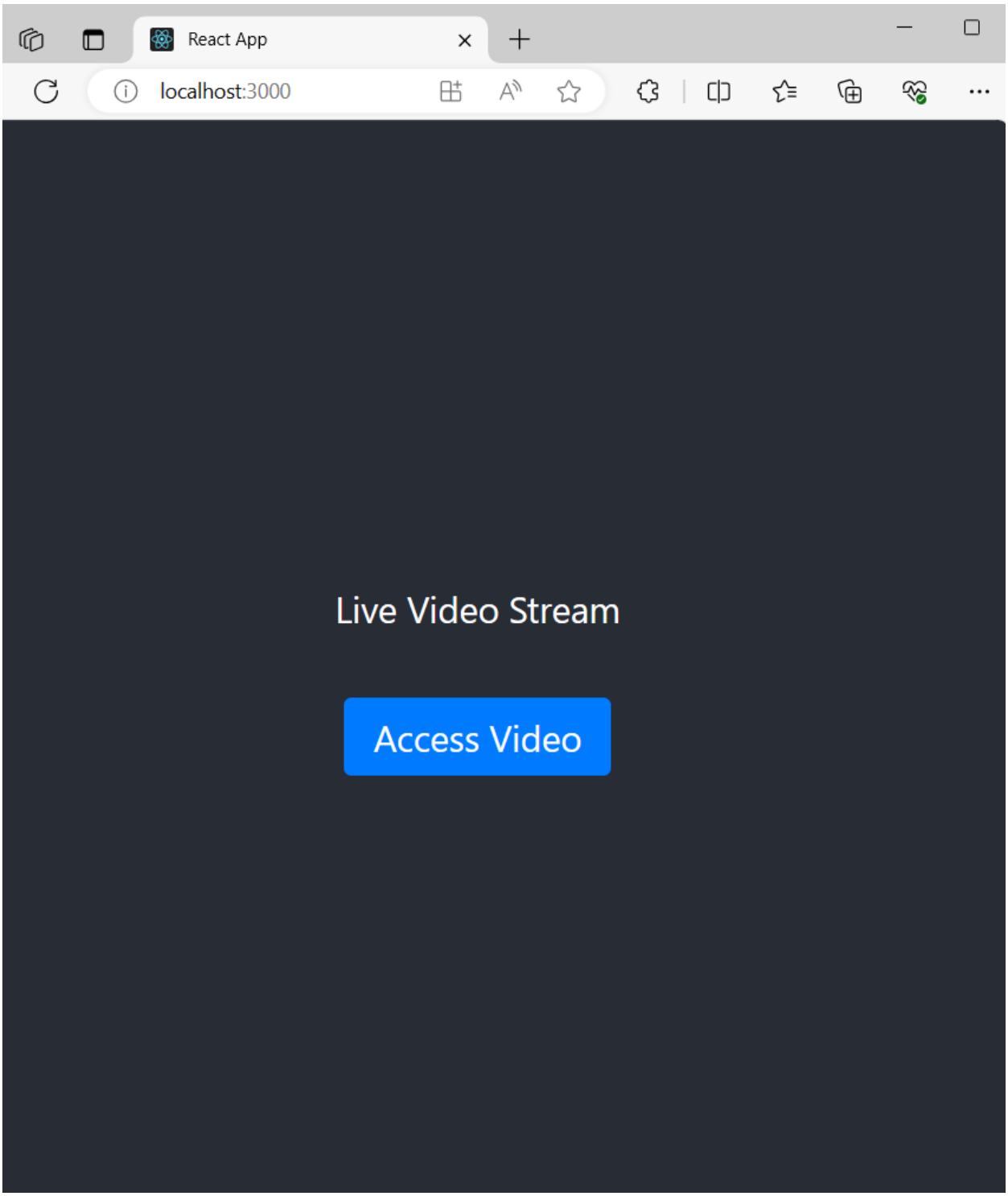


5. Now copy and paste the localhost to your browser and it display this:

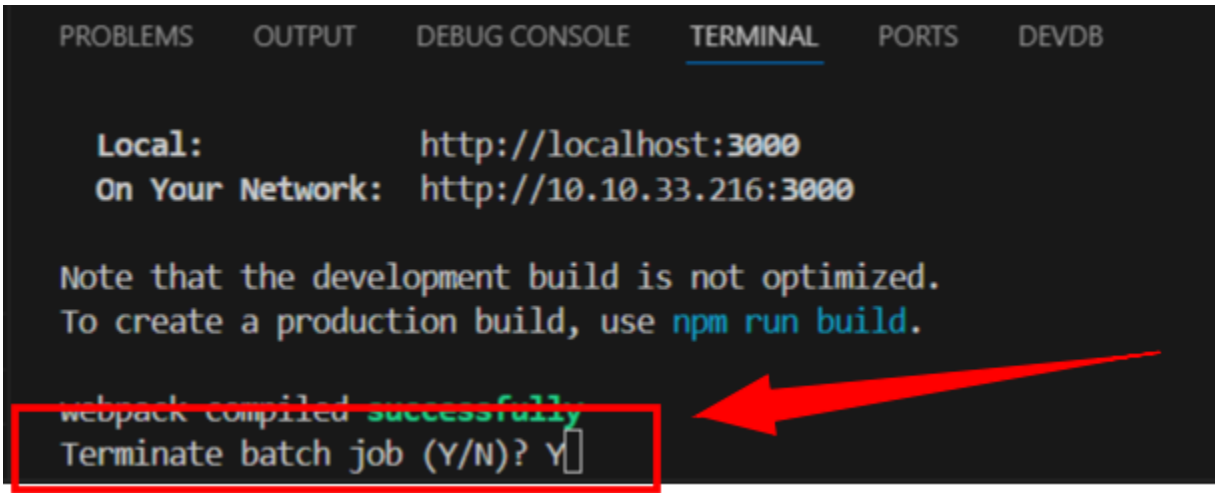


6. Now Copy and move the downloaded files in this link to your new created folder under "src" replacing the current app.js and app.css:
   gdrive link:
   https://drive.google.com/drive/folders/1hN7ObwEwSDOlJ_tzo8t8G5d_ioMnu_-O?usp=drive_link

   Now the updated display will show this:

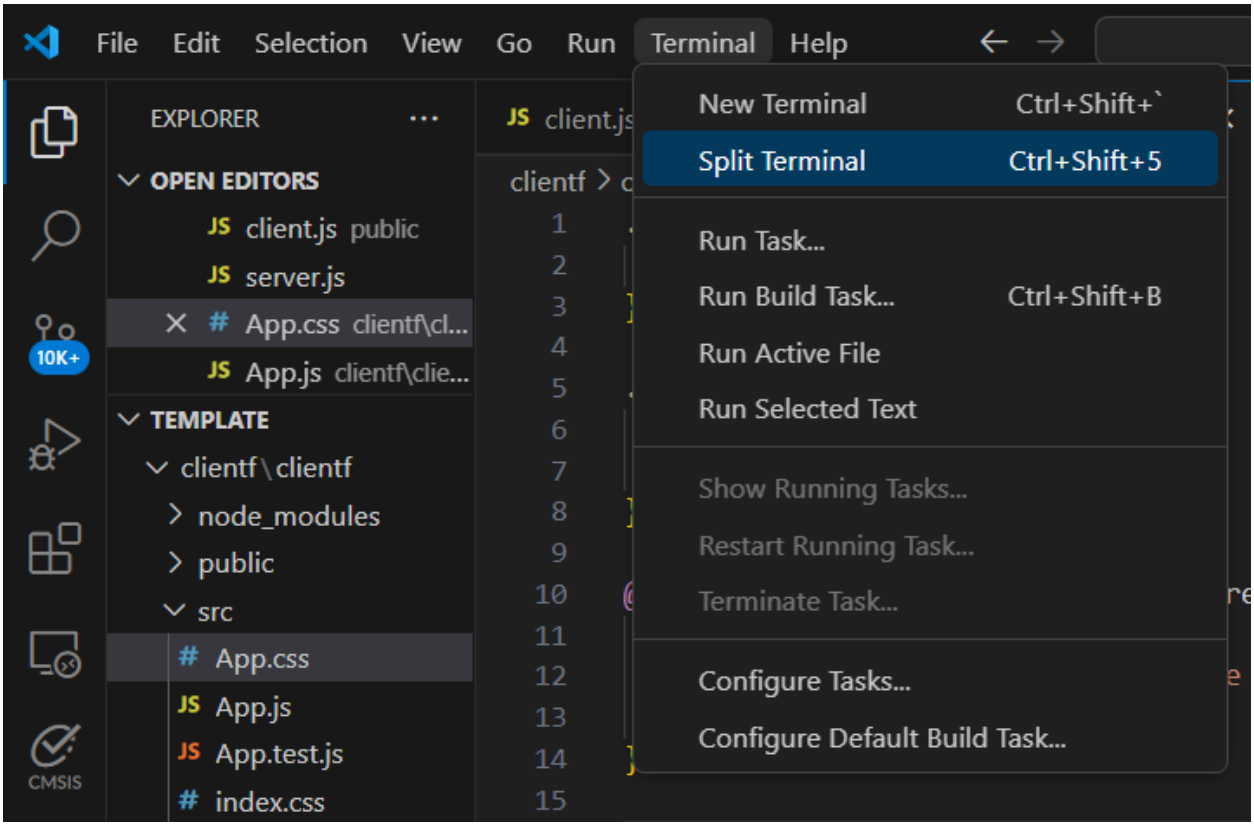


7. To stop and terminate the server running on react app press “Ctrl + c “ and it will show “terminate batch (Y/N)” , press “Y or y “ to terminate or exit.

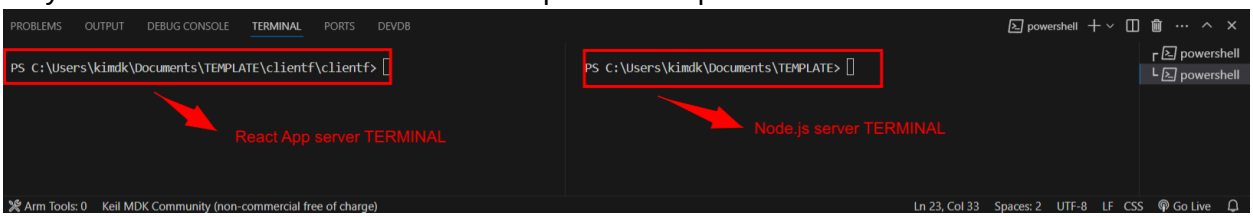


C. **Node and React running**

1. Now click split terminal and in the first terminal , proceed to type “npm start” as this will make your server.js to run in node.js server and also to provide a communication to your ESP32CAM.

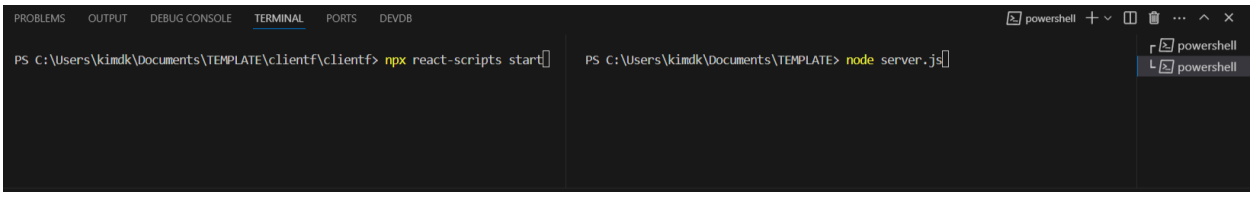


As you have observed the terminals are split into two parts:

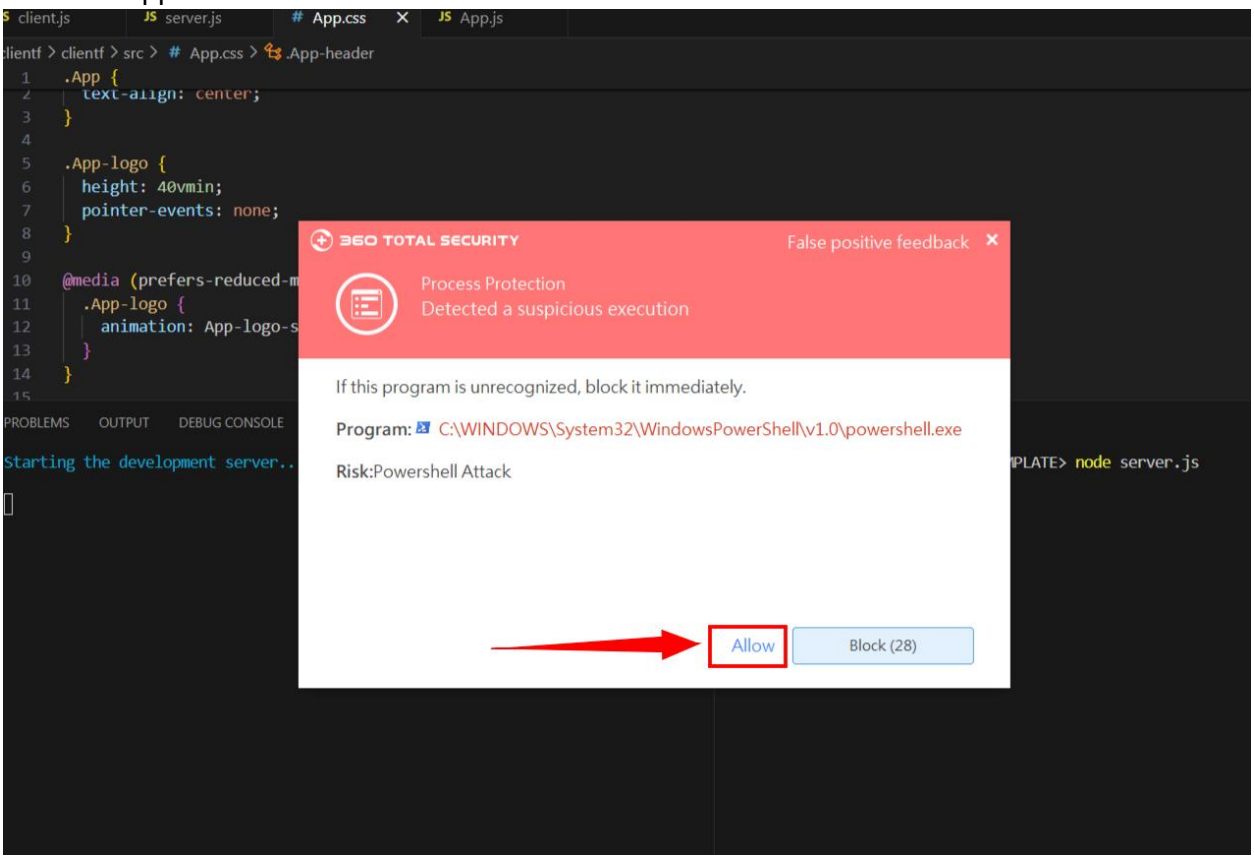


On the other hand, if the terminal is not on the right path especially in react app path, proceed to type “cd <name>” to make sure your path is in the client folder or <name> folder to run the react app.

2. Type “npx react-script start” or “npm start” to run the react app in client folder and type “node server.js” or “npm start” on other terminal to run the node.js server

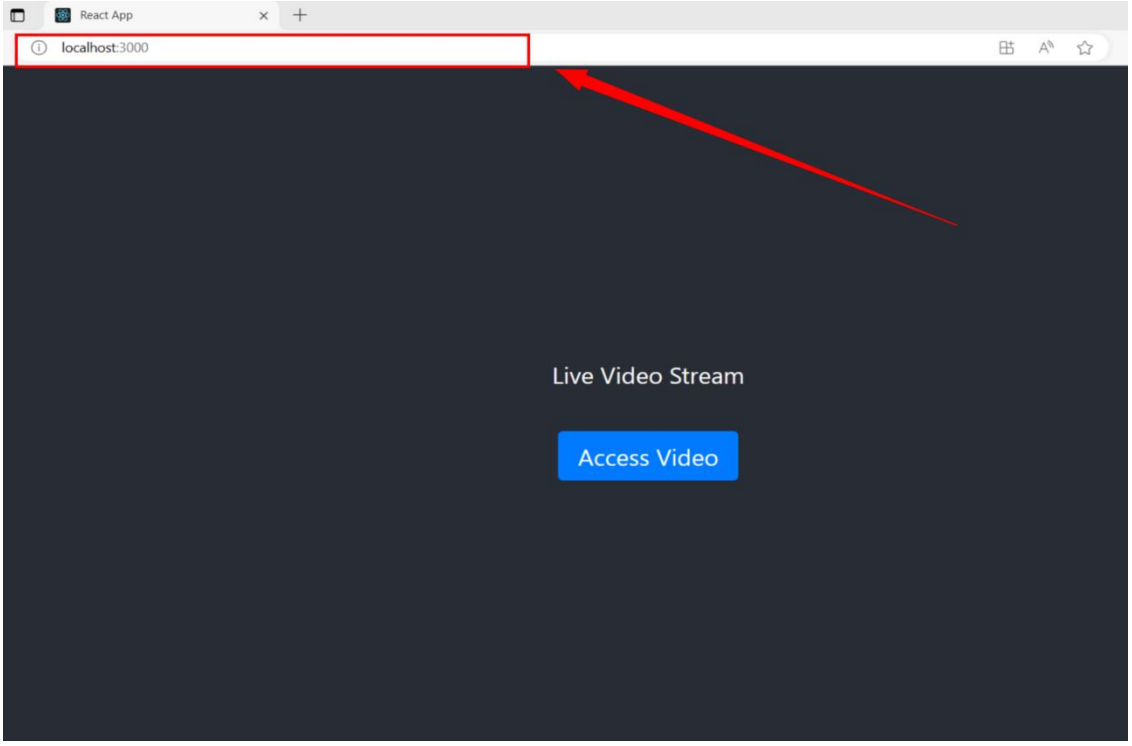


3. If you see this window (due to the antivirus software protection), click allow in order for the react app to run.

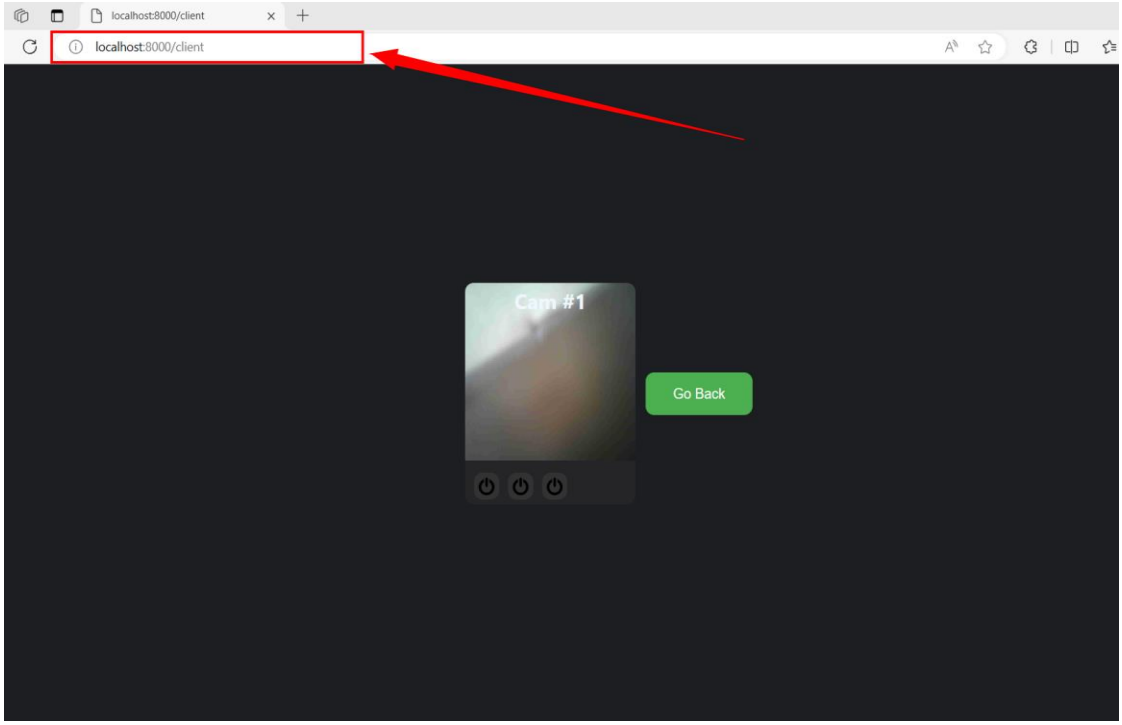


DESIRED RESULTS IN THIS DOCUMENTATION:

1. The *React app* will serve as a client side running on localhost:3000 to access the ESP32CAM video running on localhost:8000/client.



2. The ESP32CAM video stream will run on the node server.js side.



Note: if you press go back it will navigate back to the react app having a connection between server side where ESP32CAM video stream is displayed and react app as the client side.