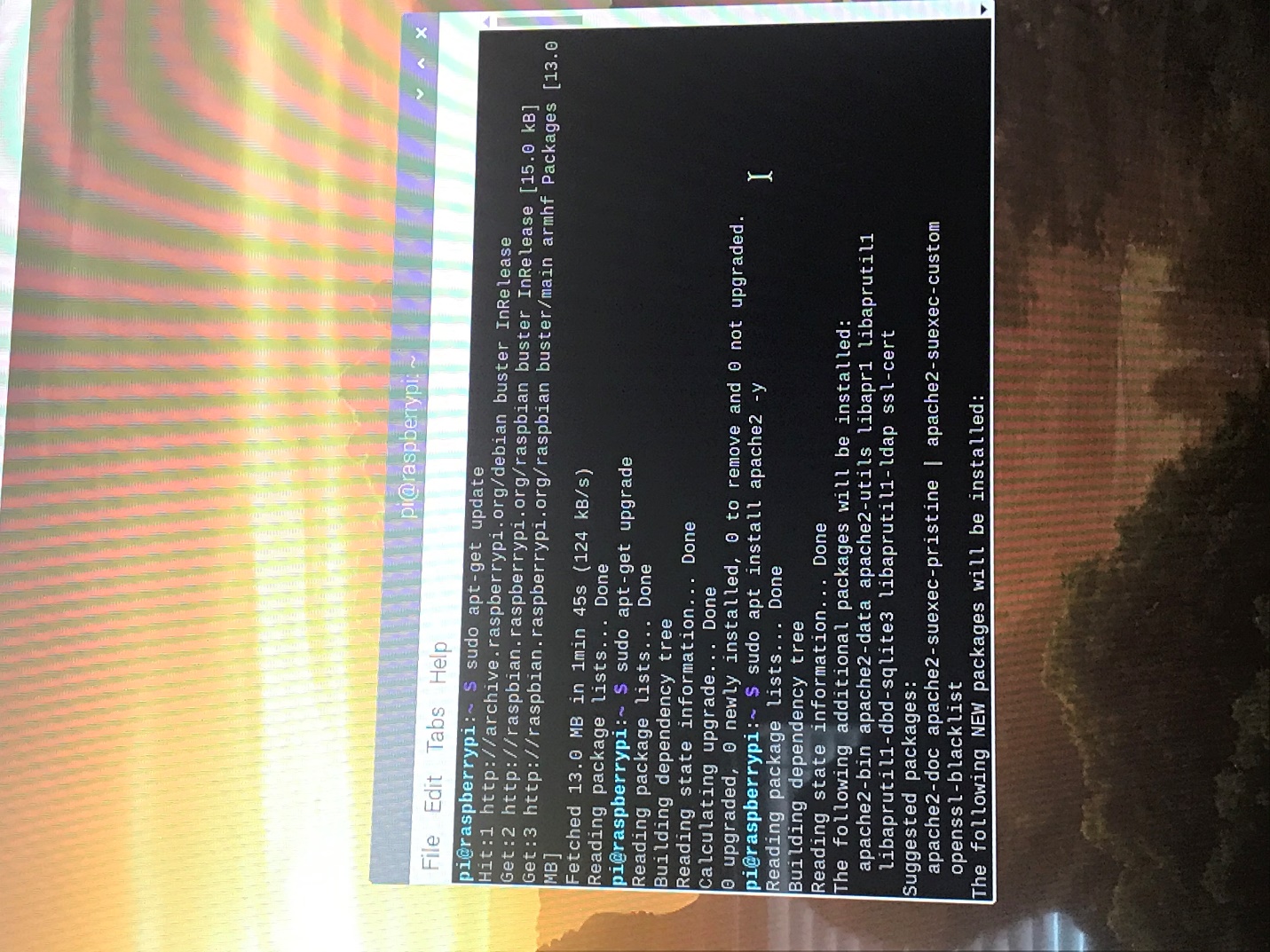
Raspberry Pi – Apache Webserver

Jonathan Franck

Objective: In this project, I will explain and show you how to install/setup Apache2 on your Raspberry Pi to be able to host a webserver.

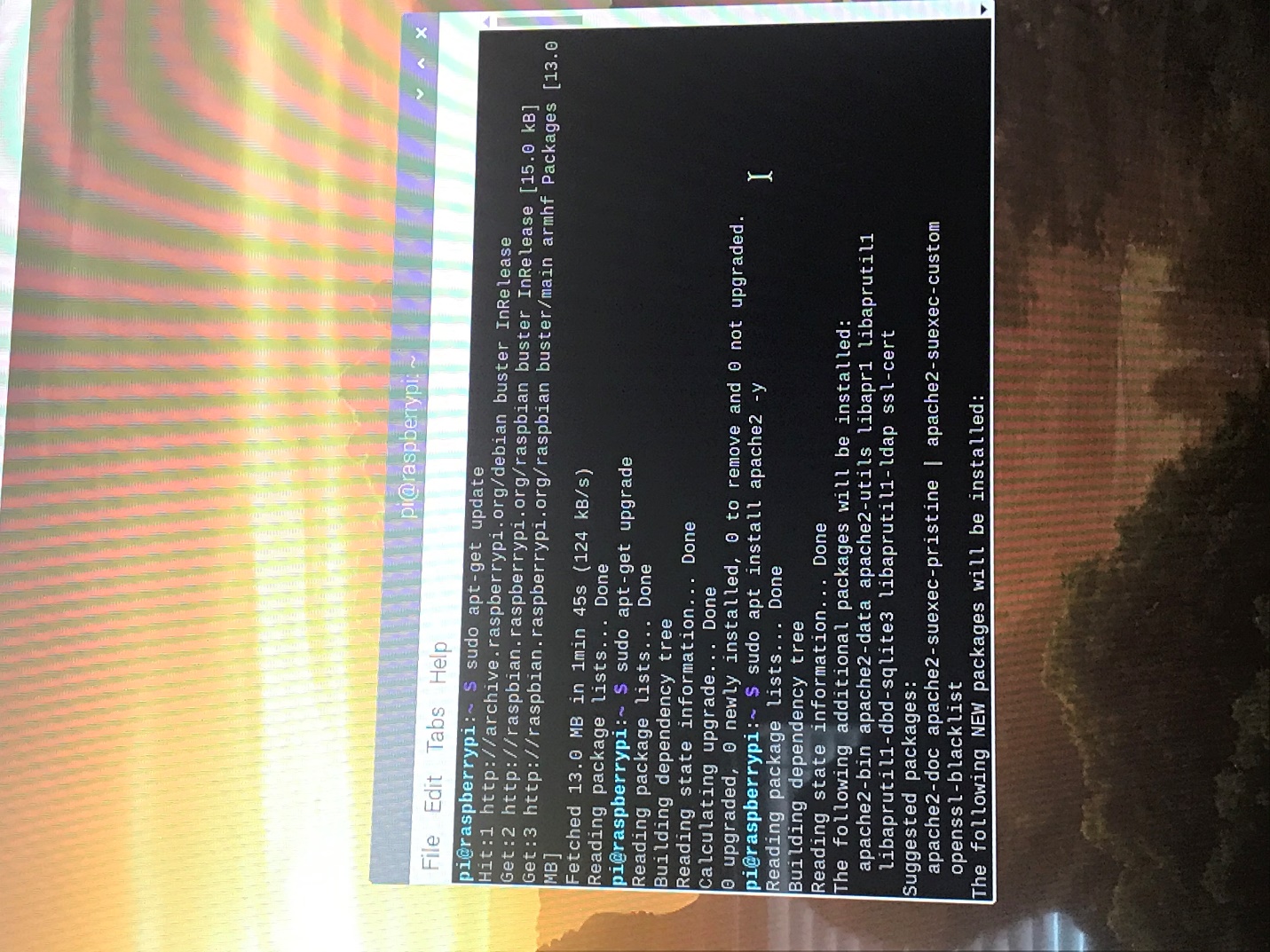
# Step 1: Update your Pi

Just like any other project, start by using the commands “sudo apt-get update” and “sudo apt-get upgrade” in the terminal to make sure all of your software is up to date (I had just updated, so mine did not install any new updates).

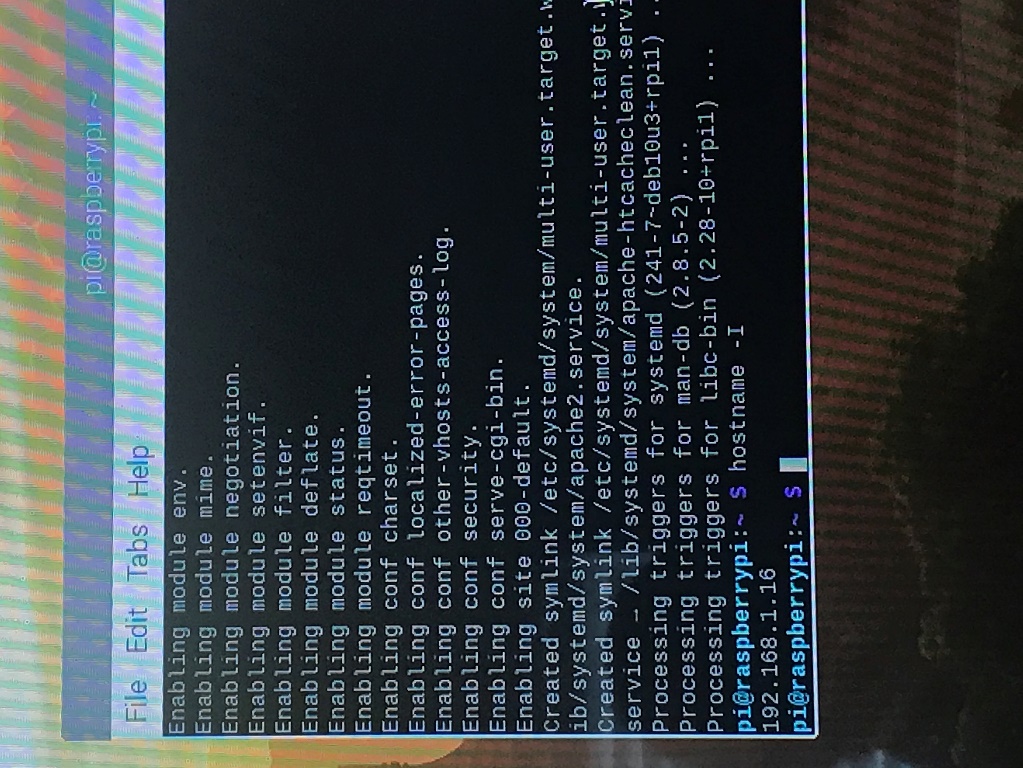


# Step 2: Install Apache2

Use “sudo apt install apache2 -y” to download Apache



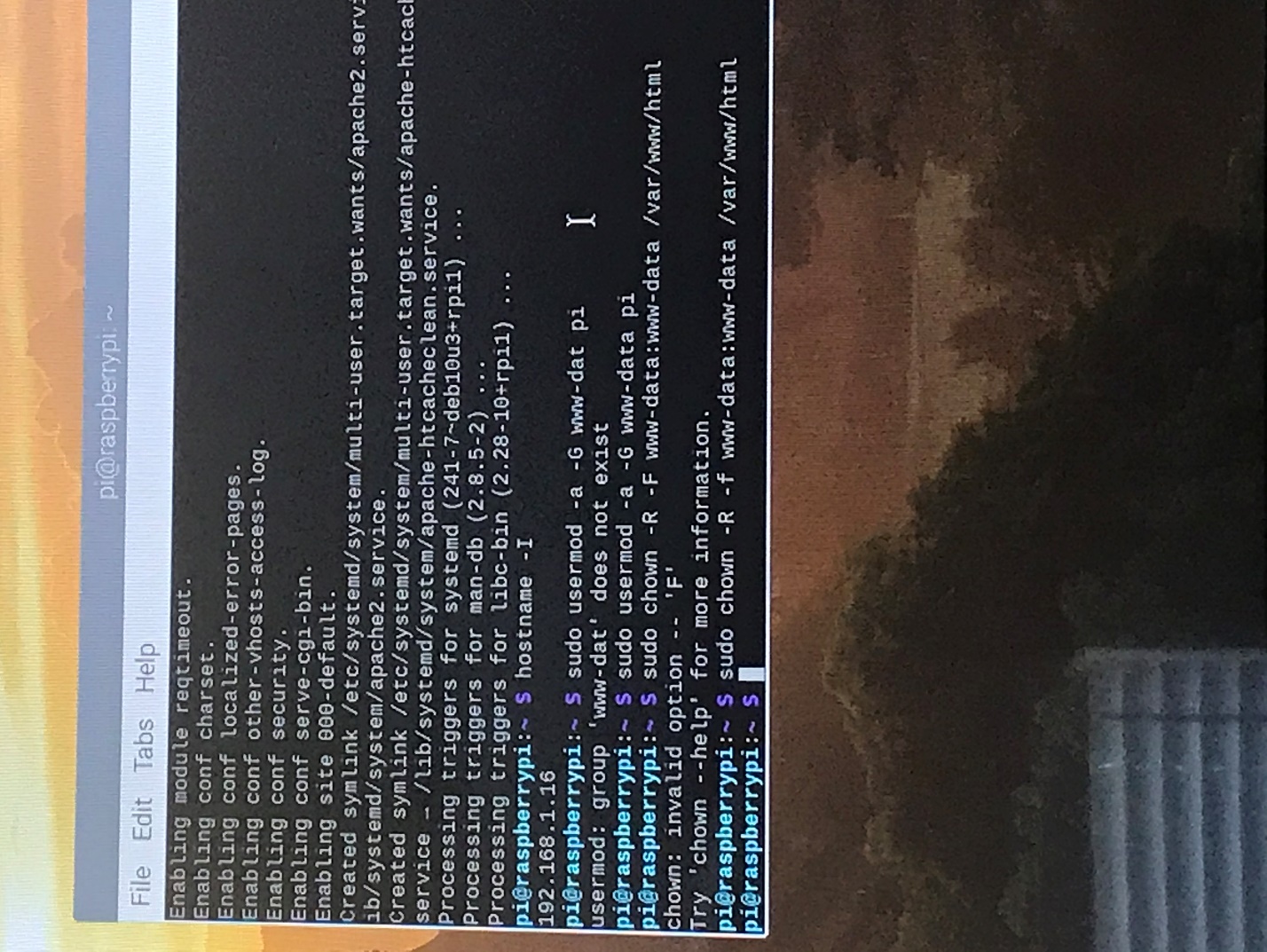
# Step 3: Check to make sure that Apache is working

Once it is installed, you can type the IP address of your Raspberry Pi into the web-browser of another device. If you don’t know the IP, you can check with the command “hostname -I”. 



# Step 4: Add user to group and give ownership

Well, now we know that Apache is working, but there are some more things that we can do. For example, if you want to be able to make changes to the files, you either have to use root or you can just give permissions to your user. Enter the following two commands in the terminal:



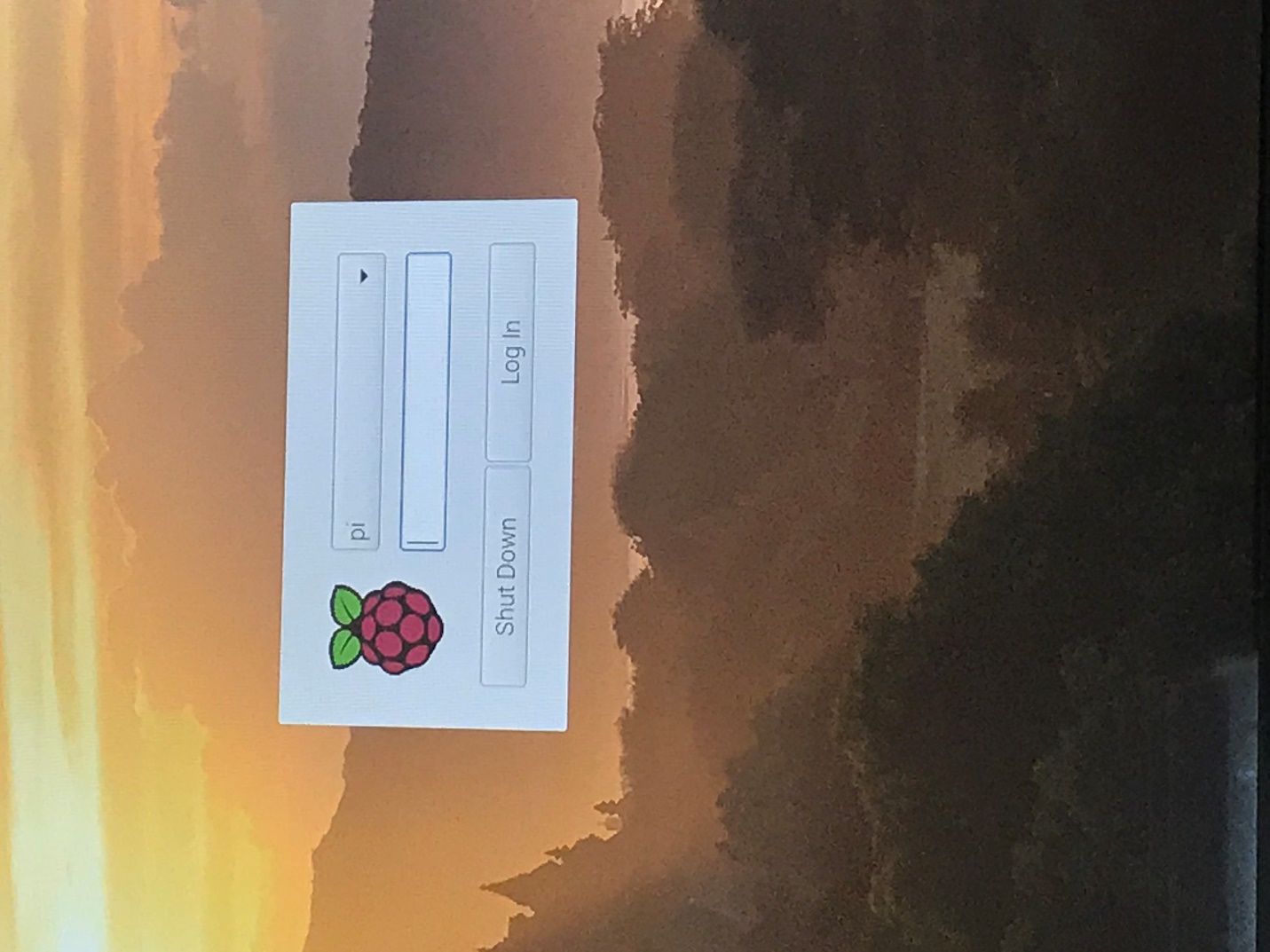
sudo usermod -a -G www-data pi

(This adds the user “pi”, which should be the name of the main use on the Raspberry Pi, to the default group in Apache)

sudo chown -R -f www-data:www-data /var/www/html

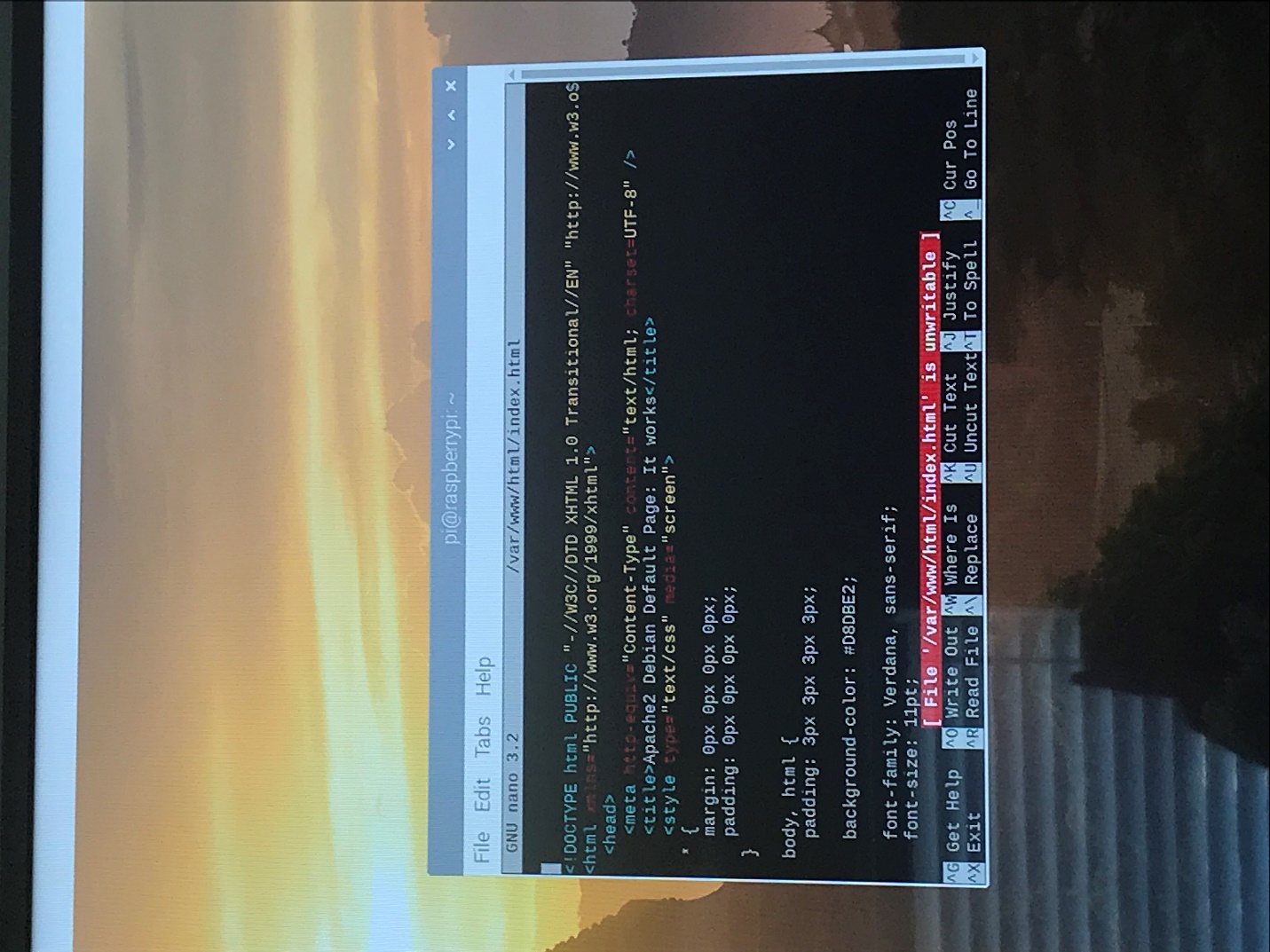
(This one gives that default group, www-data, ownership of the files in the /var/www/html directory)

Next, Logout and then log back in as user:pi again.



You should now be able to edit the index.html page by using the command:

sudo nano /var/www/html/index.html

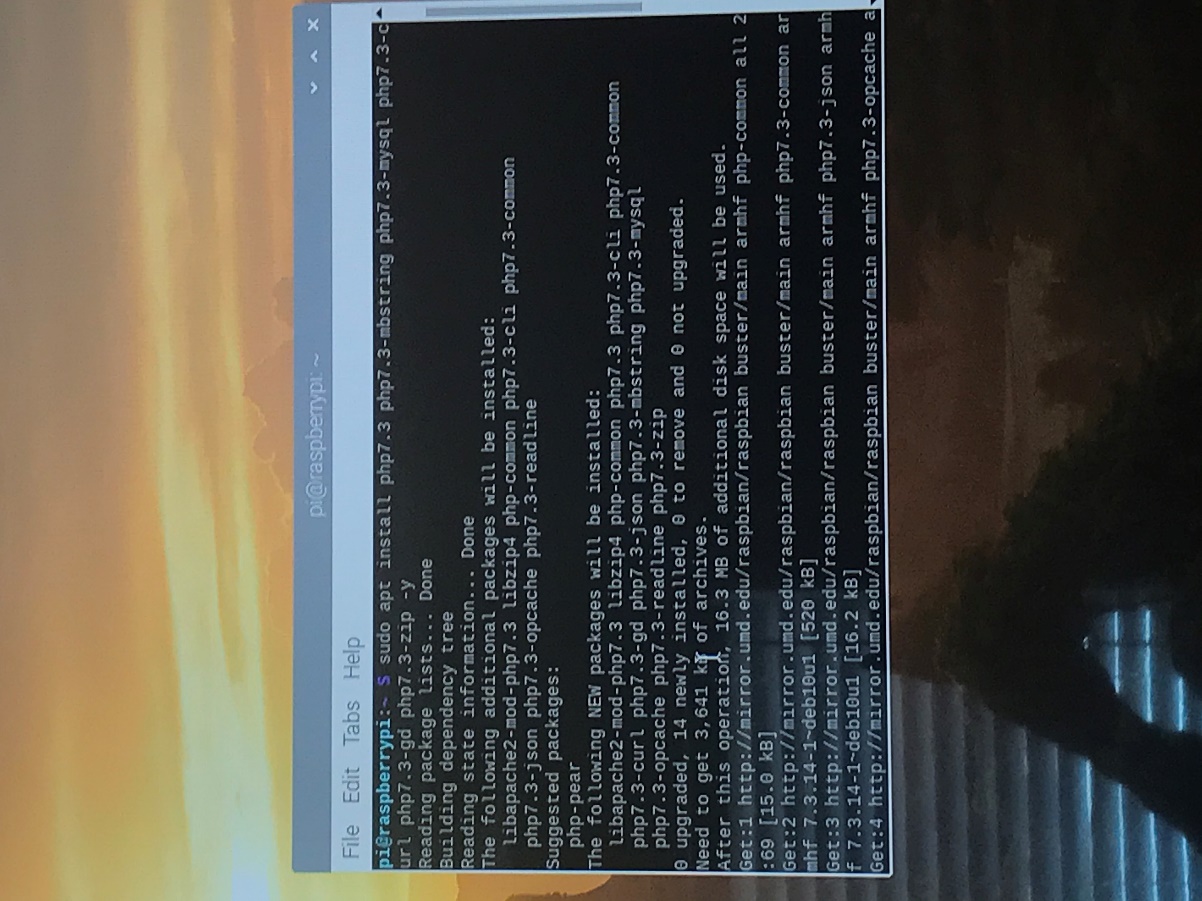


To exit press CTR + X, then Y and ENTER to save.

# Step 5: Install Php

To setup Apcahe for more dynamic websites, we can even install PHP. To do this, use the “sudo apt install” command to install the following PHP packages:

php7.3 php7.3-mbstring php7.3-mysql php7.3-curl php7.3-gd php7.3-zip -y



# Step 6: Create a new Php script to display the date

Now that PHP is installed let’s test it out. Create a new PHP script using the nano:

“sudo nano /var/www/html/sample.php”

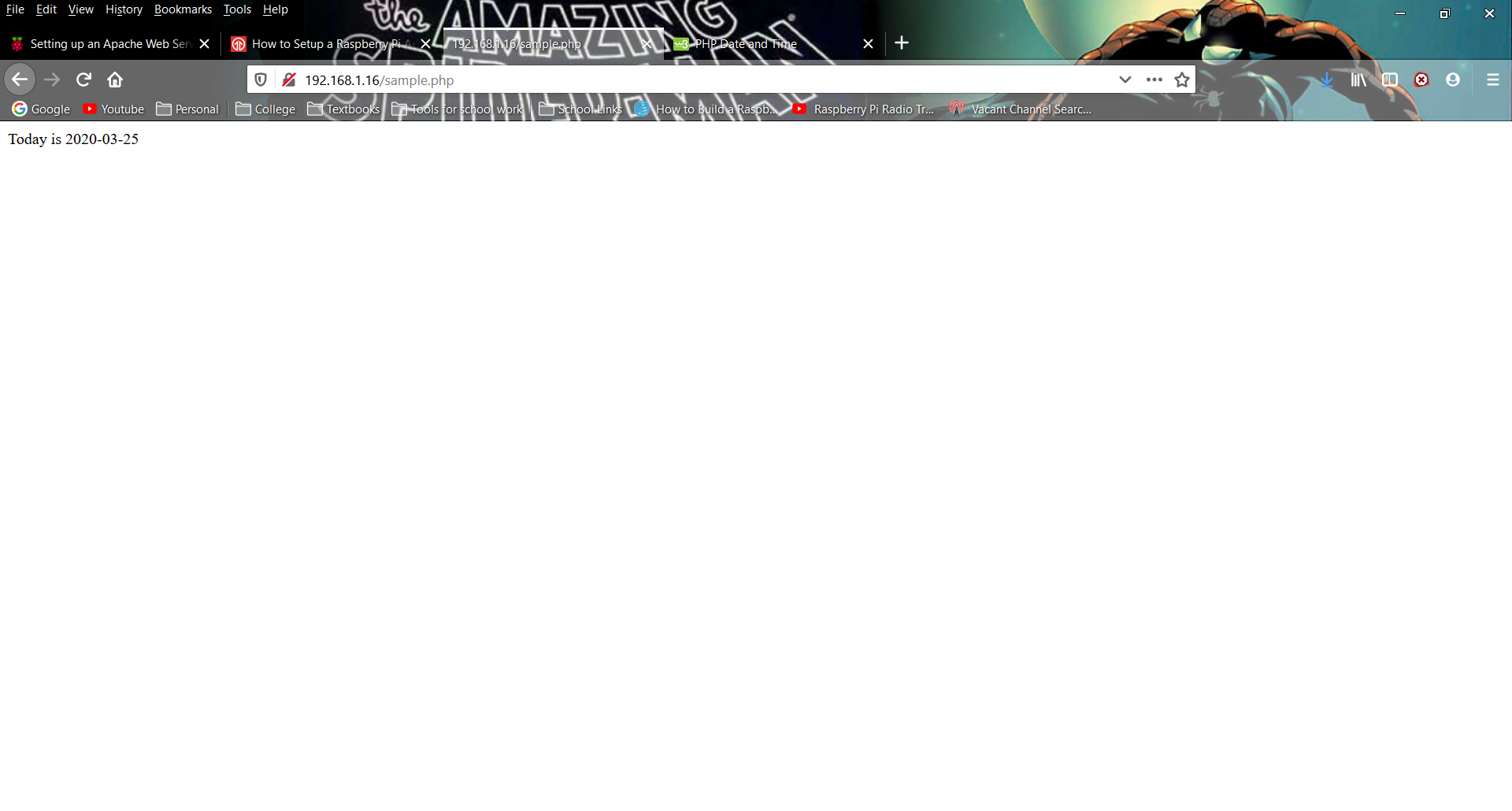
Then type the following lines of code in the file:

<?php

echo "Today's date is ".date('Y-m-d H:i:s');



Again, use CTRL + X to exit, and Y and ENTER to save the changes. To test it, open up a browser and type in <http://raspberryip/sample.php>, replacing “raspberryip” with your IP address.



# Step 7: Setup a Virtual Host

Another thing that we can do now is setup a new Virtual Host. In Apache, Virtual Hosts are necessary for managing multiple websites. Let’s create a new VirtualHost file by typing “sudo nano /etc/apache2/sites-available/example.com.conf” in the terminal. Now we can edit the VirtualHost by entering lines of code to configure it. To create a basic Virtual Host, add the following text:

<VirtualHost \*:80>

(This is a header that specifies that the Virtual Host should listen on Port 80, the port for http)

ServerName example.com

ServerAlias [www.example.com](http://www.example.com)

(These two lines designate the base domain. The server name matches the VirtualHost with the base domain, and the server alias lists any other domains that will be matched.)

DocumentRoot /var/www/example.com/public\_html

(The document root specifies the directory that Apache will serve the files from)

ErrorLog ${APACHE\_LOG\_DIR}/example.com\_error.log

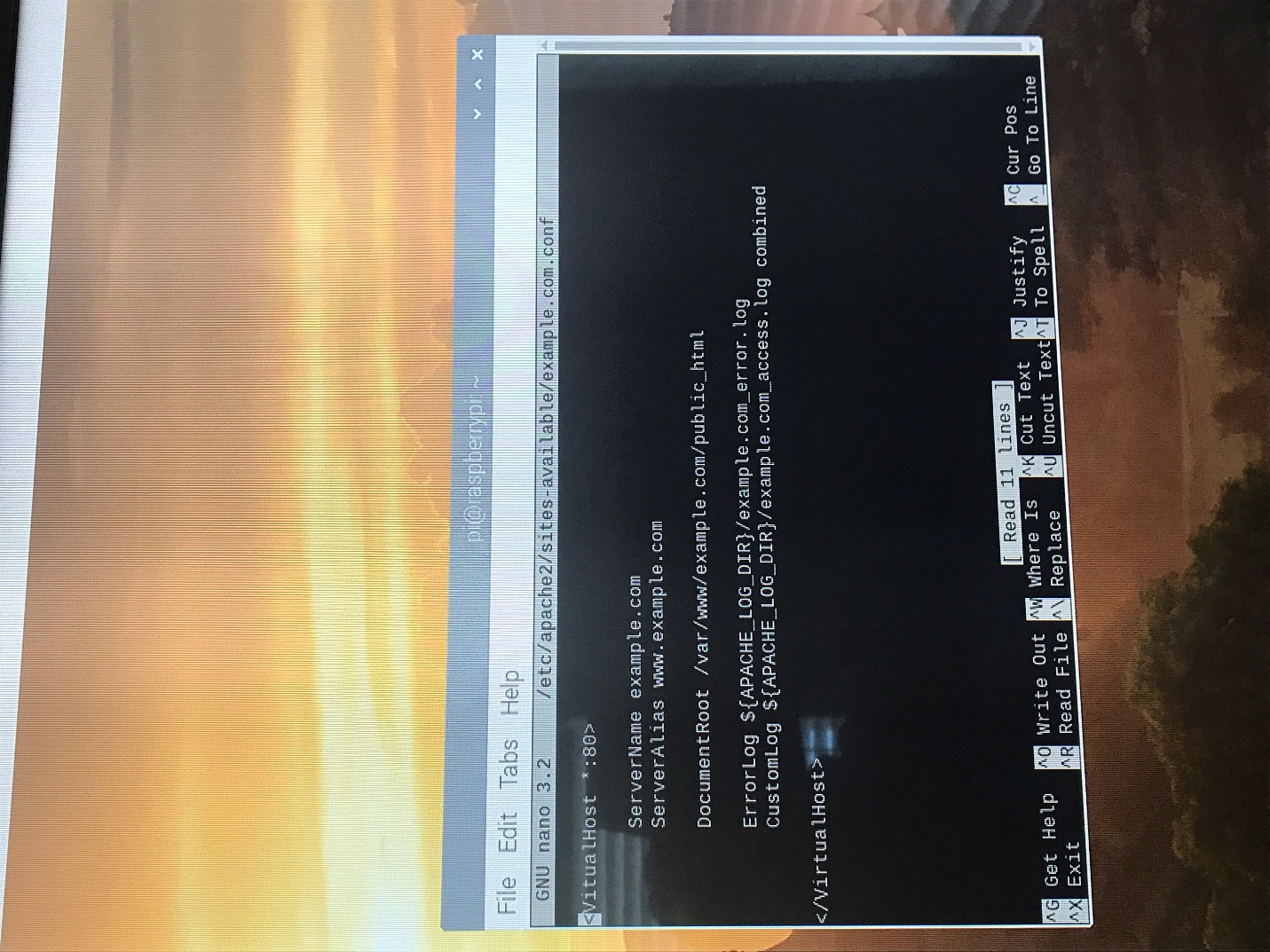
CustomLog ${APACHE\_LOG\_DIR}/example.com\_access.log combined

(The two lines above detail where (which directories) we want the logs to be recorded)

</VirtualHost>

(This just closes the VirtualHost)

You should end up with a file that looks like this:



One more time, CTRL + X to exit, and Y and ENTER to save.

Now we want to give the www-data group ownership of this file, as well. Use the following commands in terminal:

sudo mkdir -p /var/www/example.com/public\_html

sudo chown -R www-data:www-data /var/www/example.com/public\_html



Now we just have to activate the Virtual Host and reload Apache 2 for it to take effect. Use these commands: “sudo a2ensite example.com.conf” and “sudo systemctl reload apache2”



Your Virtual Host should now be running. Congratulations! From here you could even give your server a domain name by setting up port forwarding and pointing a DNS to your Raspberry PI’s IP address (that will not be covered in this project).

# Challenges:

Now that I am no longer on PBA’s WiFi, I did not have nearly as many issues. One standing issue I had to resolve before starting this project was from an incomplete update from a previously. Essentially, my PI would not boot to the desktop, and when I tried to connect to PBA’s WiFi I had no way to open a browser for the authentication page. Without internet there was no way to run updates to fix the issue, so I decided to just reformat my SD card and reload NOOBS onto it. It took me a couple tries to figure out how to reformat the drive properly as there were about four partitions, but I formatted the first one (:/D) and set the volume to default. After copying the NOOBS files over and starting the Raspberry Pi, I was able to reinstall Raspbian, and continue working on my PI as if the issue never happened.