**Name of Title:** Learning Nginx

**Video Name:** LEMP Stack Demonstration

**Estimated Length:**

**Author Name:** Michael Jenkins

**Chapter\_Section\_Video:**

**Video Objective:**

At the end of this video the learner will be able to demonstrate an application that uses all levels of the LEMP stack.

**Script:**

At this point, we’re readying to add a some dynamic content to our static site.

SLIDE: Appointments Web App

(Picture/video of the web app)

We’ll be setting up a PHP application that lists the appointments for the patients of Wisdom Pet Medicine. The application is a PHP script that connects to the appointment database, reads the appointments, and lists them in a table.

It’s a simple application, but it’s perfect for demonstrating how the components of the LEMP stack work together.

I’m connected to our development server where nginx, mariadb, and php are installed and configured. Let’s change to the root user and do a quick check to make sure all the systems are running as expected.

CTRL+L  
 sudo su -

We’ll use systemctl to check the status of each service: nginx, mysql, and php-FPM.

We could check each one individually but fortunately, we can use one call to systemctl and check them all at the same time:

CTRL+L

Systemctl status nginx mysqld php7.2-fpm

To get the actual info we need, let’s pipe the output to grep and look for Loaded and Active:

Systemctl status nginx mysqld php7.2-fpm | grep -E “(Loaded|Active)”

This output shows us that each service is running and active. We’re good to go!

If you’re following along and your output differs from this, review the installation for each service and make any corrections.

Now let’s install the PHP script. I’m using the exercise files so I can find the script in the slash vagrant directory which is synchronized with the directory where i started the VM with vagrant up.

CTRL+L

Cd /vagrant

Ls -1

The script is in the file named index.php.

We won’t be going into too much detail around the code, but let’s take a quick look at a few lines inside the script:

vim +97 /vagrant/index.php

I wanted to point out the lines where two components of the stack come together, the PHP and the database. On this line we see where PHP is using mysqli\_connect function to connect the database on localhost. The user and password are there along with the name of the database.

A few lines down (102), we see where PHP uses the mysqli\_query function to get all of the rows from the data table of the appointment database.

And on this line (110), PHP prints the data as HTML. Of course, this is the content we’ll see when the page is served by nginx and loaded in our browser.

CLOSE FILE; GO BACK TO CLI

So now let’s put this script where nginx can access it. We’ll make a directory inside the root of the website:

CTRL+L

mkdir /var/www/wisdomdpetmed.local/appointments

And copy the index.php file there:

cp /vagrant/index.php /var/www/wisdomdpetmed.local/appointments

We also need to set the permissions on the file to make sure the nginx process can read it:

chmod +r /var/www/wisdomdpetmed.local/appointments/index.php

Ls -ltr /var/www/wisdomdpetmed.local/appointments/index.php

OK everything looks good.

Now is a good time to point out why the file for our application is named index.php.

Nginx is configured to use index.php as one of the files that gets loaded by default when nginx serves a directory. By naming our script index.php and then placing it in a directory named appointments, anytime someone requests the page /appointments, the app will get loaded automatically. Pretty slick, right?

We could have used another name for the script, appointments.php for example, but using index.php make things easier to manage if the appointment locations needs other directives applied to it.

Let’s take a look at our web app now.

OPEN BROWSER TO:

<http://192.168.0.3/appointments> (or <http://wisdompetmed.local/appointments>)

# **WISDOM PET MEDICINE APPOINTMENTS**

Today is 06/13/2018

SQL Error: Table 'appointments.data' doesn't exist

| **Pet Name** | **Owner Name** | **Appointment Date** | **Reason for Appointment** |
| --- | --- | --- | --- |

We can see our application but there’s something missing. And this error message tells us right away: There’s no data in our database! Let’s fix that.

**STOP**

GO BACK TO THE TERMINAL

Cd /vagrant

Ls -1

In the /vagrant directory, there’s a file named appointment\_database.sql that contains the data we need and the commands to insert it into the database. Before we do that, let’s take a look inside.

view appointment\_database.sql

In the SQL file, we can see on this line where the table gets created along with the names of each column. And this line (13) is where the data insertion begins. The rest of the lines in the file are the rows of data that make up the database.

Exit the file

Now let’s insert the data. We’ll use the “standard in” redirect character to send the contents of the SQL file into a mysql command:

mysql -u admin -padmin appointments < /vagrant/appointment\_database.sql

Now let’s refresh the appointments page:

GO TO BROWSER AND REFRESH PAGE

There’s our data! Now that the database has some content, the page loads without an error.

Now we can say we’ve successfully applied each component of the LEMP stack. We added some dynamic content to our static site by applying some PHP code and some database content.

But we’re still not done. There are other things we can consider for our site with security being the top priority.

**Exercise Files:**

**Basement:**

1. Shrink pictures

Our development VM is configured now and we’re logged in. Let’s become the root user:

If you’re following along with the exercise files, you can use the Vagrantfile in the folder for this chapter. It will boot the VM and install all the components of the LEMP stack along with the demo site. The MySQL database will also be configured with the admin user and the appointment database.

If you're not using the exercise files, you can follow along with a VM running Ubuntu 18.04 LTS. You’ll need root access and you’ll need to install and configure each component of the LEMP stack.