6. Installation and configuration of common software frame works such as Laravel.

Installing Laravel

Procedure:

Step 1: Install Apache web server

To install apache2, type:

```
sudo apt install apache2
```

Once installed, Apache should be running. If it's not, for whatever reason, start it:

```
sudo systemctl start apache2
```

Then enable it to start on boot time.

```
sudo systemctl enable apache2
```

To verify the status of Apache, execute:

```
sudo systemctl status apache2
```

Step 2: Install PHP and additional PHP extensions

Laravel 8 requires PHP 7.3 or above.

PHP 7.4 is available in Ubuntu repositories. So, install PHP and the following PHP extensions.

```
sudo apt install php libapache2-mod-php php-mbstring php-cli php-bcmath php-json php-xml php-
```

When the installation is complete, verify the PHP version.

```
php –v
```

Step 3: Create Database for Laravel Application

Next up, we will create a database for the Laravel application. But first, we need to install a database server. Laravel supported database systems are MariaDB, MySQL, SQLite, Postgres, or SQL Server.

We will go with the MariaDB database engine.

```
sudo apt install mariadb-server
```

Once the database server is installed, log into the MariaDB prompt:

```
sudo  mysql -u root -p
```

Once logged in create the database, database user, and grant all privileges to the database user.

```
CREATE DATABASE laravel_db;CREATE USER 'laravel_user'@'localhost' IDENTIFIED BY 'secretpassword';GRANT ALL ON laravel_db.* TO 'laravel_user'@'localhost';FLUSH PRIVILEGES;QUIT;
```

Step 4: Install Composer

Composer is a dependency package manager for PHP. It provides a framework for managing libraries and dependencies and required dependencies. To use Laravel, first install composer.

To download Composer, invoke the command shown.

```
 curl -sS https://getcomposer.org/installer | php
```

Next, move the composer file to the /usr/local/bin path.

```
 sudo mv composer.phar  /usr/local/bin/composer
```

Assign execute permission:

```
 sudo chmod +x   /usr/local/bin/composer
```

Verify the Composer version installed:

```
composer -–version
```

Composer version 2.1.3 is installed.

Step 5: Install Laravel 8 on Ubuntu

With Composer installed, the next course of action is to install Laravel.

Navigate to the webroot directory, type:

```
cd /var/www/html
```

Now, install Laravel using the composer command, type:

```
 sudo composer create-project laravel/laravel laravelapp
```

The command creates a new directory called laravelapp and installs all the files and directories for Laravel.

Step 6: Configure Apache to serve Laravel site

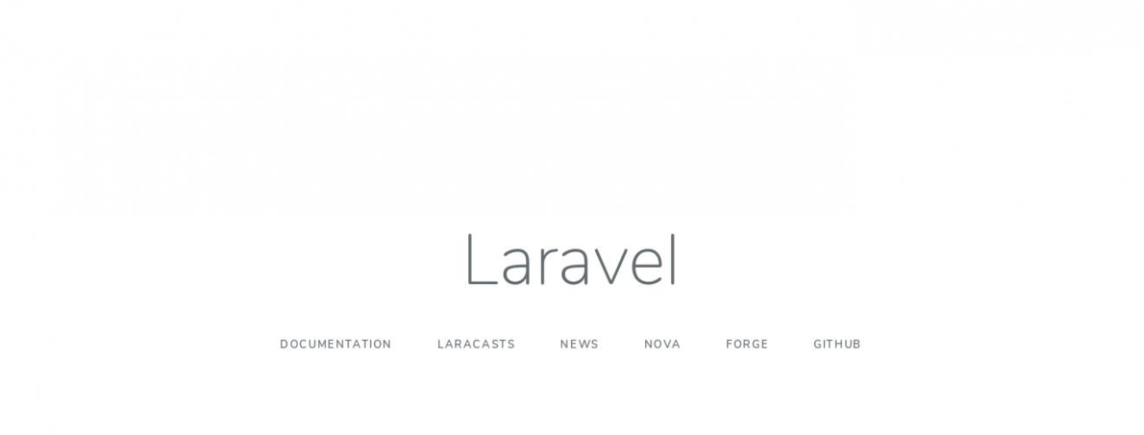
Lastly, we need to set up the Apache webserver to host the Laravel site. For that to happen, we need to create a virtual host file.

```
 sudo vim /etc/apache2/sites-available/laravel.conf
```

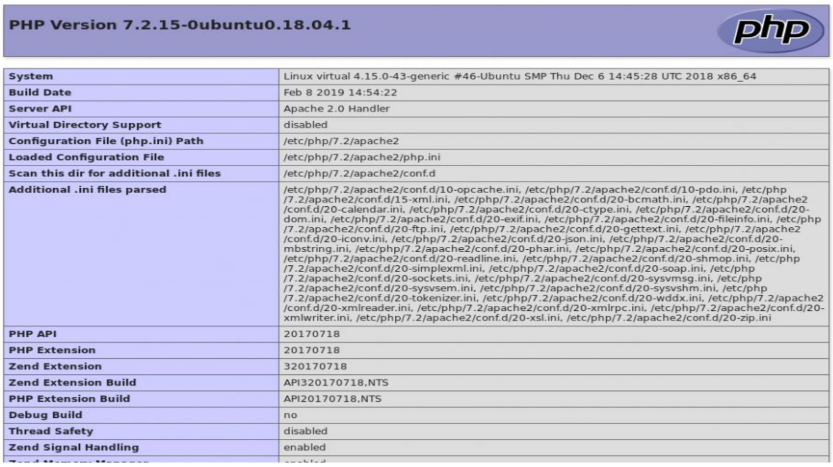
Step 7: Access Laravel from a browser

sudo apt Laravel

OUTPUT



PHPMYADMIN


PHP Version 7.2.15-0ubuntu0.18.04.1

System	Linux virtual 4.15.0-43-generic #46-Ubuntu SMP Thu Dec 6 14:45:28 UTC 2018 x86_64
Build Date	Feb 8 2019 14:54:22
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-opcache.ini, /etc/php/7.2/apache2/conf.d/10-pdo.ini, /etc/php/7.2/apache2/conf.d/15-xml.ini, /etc/php/7.2/apache2/conf.d/20-bcmath.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-dom.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-mbstring.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-simplexml.ini, /etc/php/7.2/apache2/conf.d/20-soap.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, /etc/php/7.2/apache2/conf.d/20-wddx.ini, /etc/php/7.2/apache2/conf.d/20-xmlreader.ini, /etc/php/7.2/apache2/conf.d/20-xmlrpc.ini, /etc/php/7.2/apache2/conf.d/20-xmlwriter.ini, /etc/php/7.2/apache2/conf.d/20-xsl.ini, /etc/php/7.2/apache2/conf.d/20-zip.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled