

|  |
| --- |
| **Project 1- Buyer/Seller matchmaking website**  Tools Setup Guide |



To run Buyer/Seller matchmaking program in a development environment the following are required:

* **Composer** - a tool for dependency management in PHP
* **Laravel** - a PHP framework, recommended version(5.4)
* **Local web host** - ie. XAMPP, WAMP, MAMP with PHP version (7.\*.\*)
* **GitHub client** - ie. Git, SourceTree, Tower
* **Project files** from GitHub

This is a guide for setting up Xampp local server with laravel installed, and to teach you how to import an existing laravel project on my GitHub repository

Note: if you are familiar with basic laravel project installation, you can skip the first 3 steps and move to step 4 directly. If you are having trouble running the project, please read the instruction again from step 1.

**Step1 (installation)**

You have to download and install the following app:

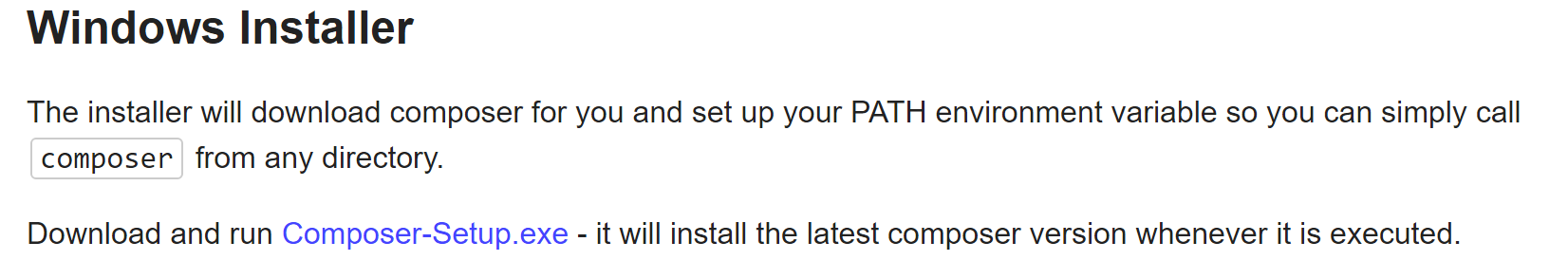
1. **Xampp with PHP 7.1.1**

<https://www.apachefriends.org/download.html>

1. **Composer**

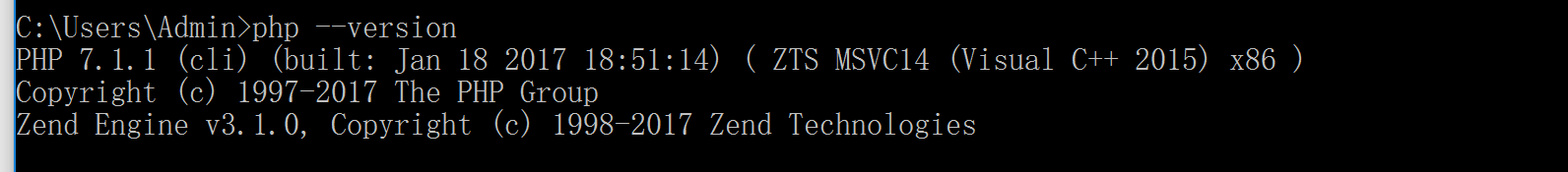
<https://getcomposer.org/download/>

(if you are using windows, just click “composer-setup.exe” to start the download)

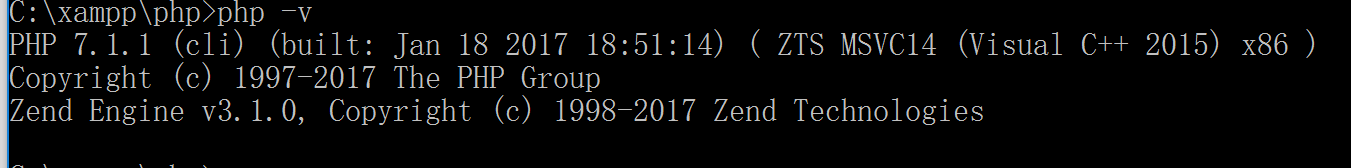


Please install Xampp first, don’t worry about the warning prompt at the start of the installation

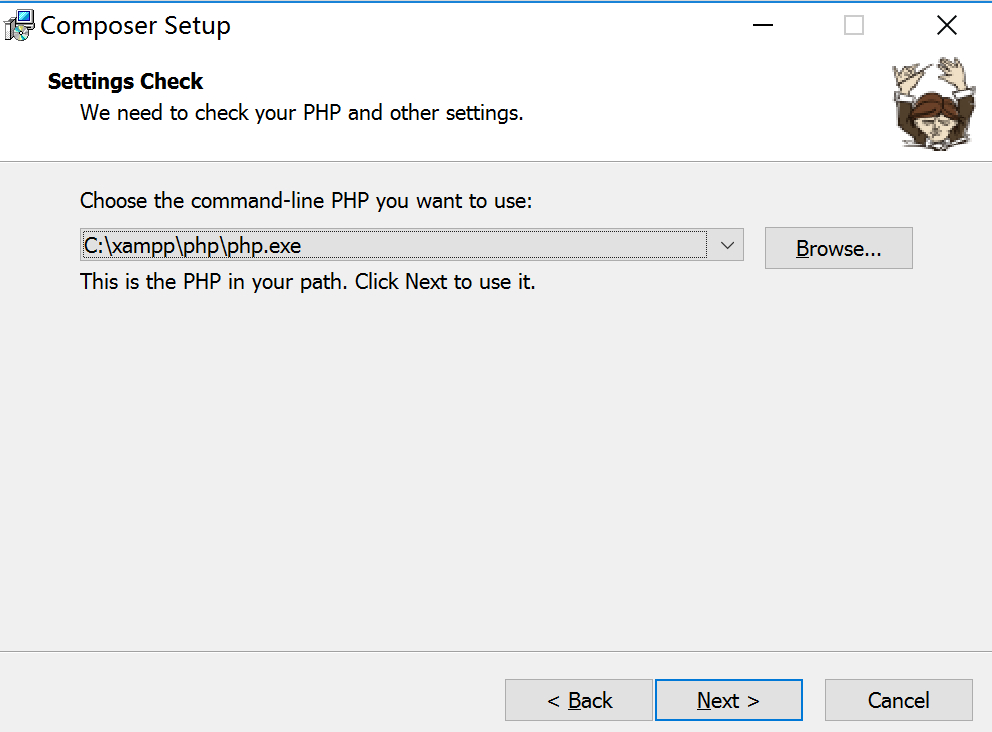
After you have installed Xampp, check that your php has successfully installed by running “php --version” in CMD



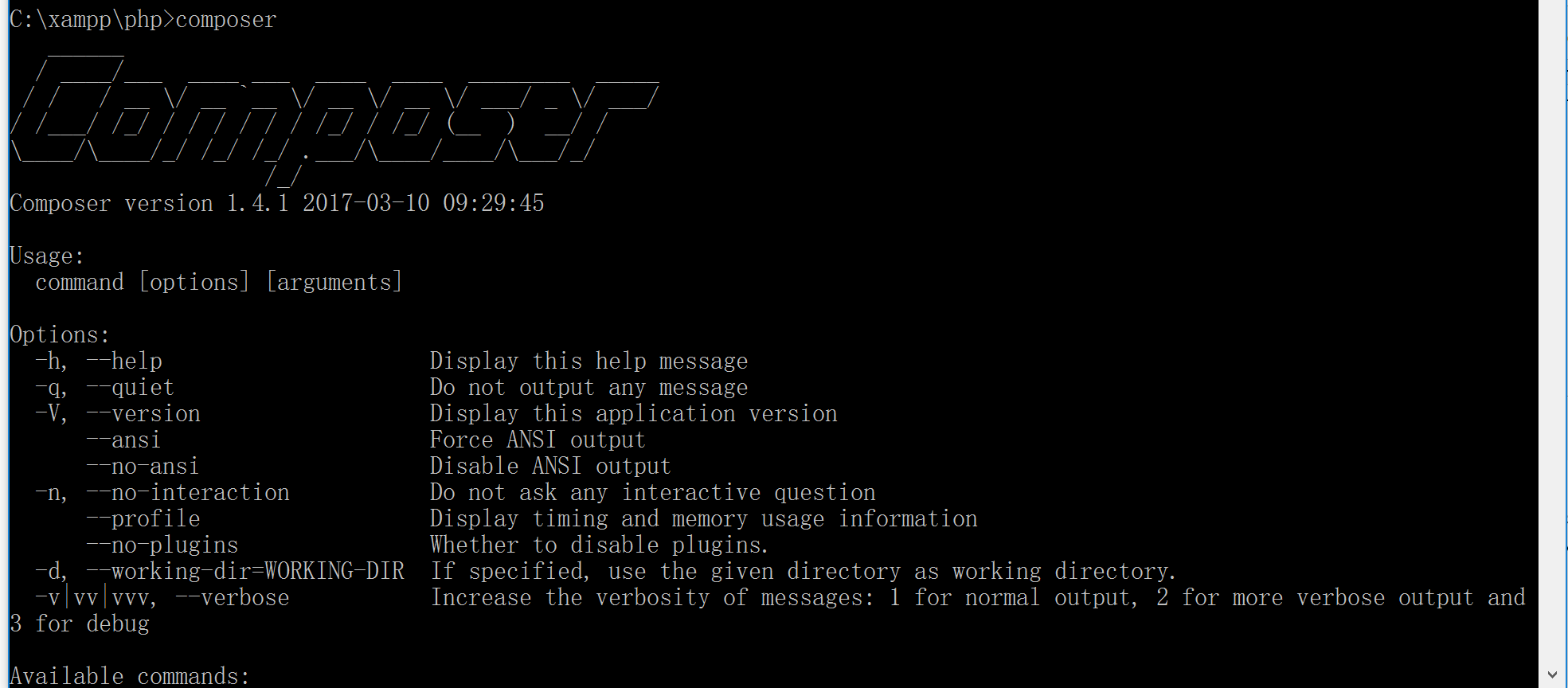
(if it doesn’t work, go to the path where your php.exe is, which is in “Xampp/php” and run the command again, it should work)



Next, install composer, and make sure to locate to the path where your php.exe is in



After finished installing, check that you have successfully installed composer by running “composer” in CMD



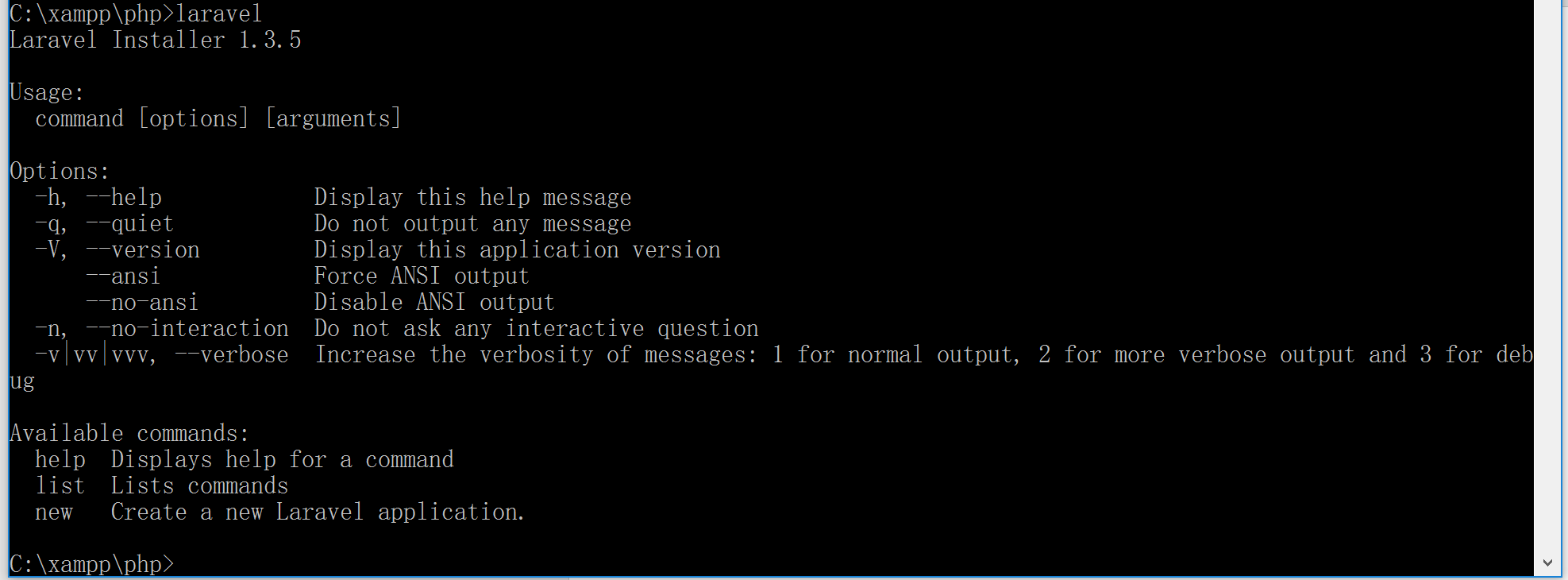
**Step 2 (Install laravel)**

Now you can install laravel by running

composer global require "laravel/installer"

in CMD, it may take a while to install.

After that, try run “Laravel” in CMD and it should display this:



That means you have installed laravel onto your computer.

You can now create a new laravel project by running “laravel new projectname ”, notice that the project will be created under the current path your CMD is locating

For example:

C:\Users\Admin\AppData\Local\Microsoft\Windows\INetCache\Content.Word\laravelnewblog.png

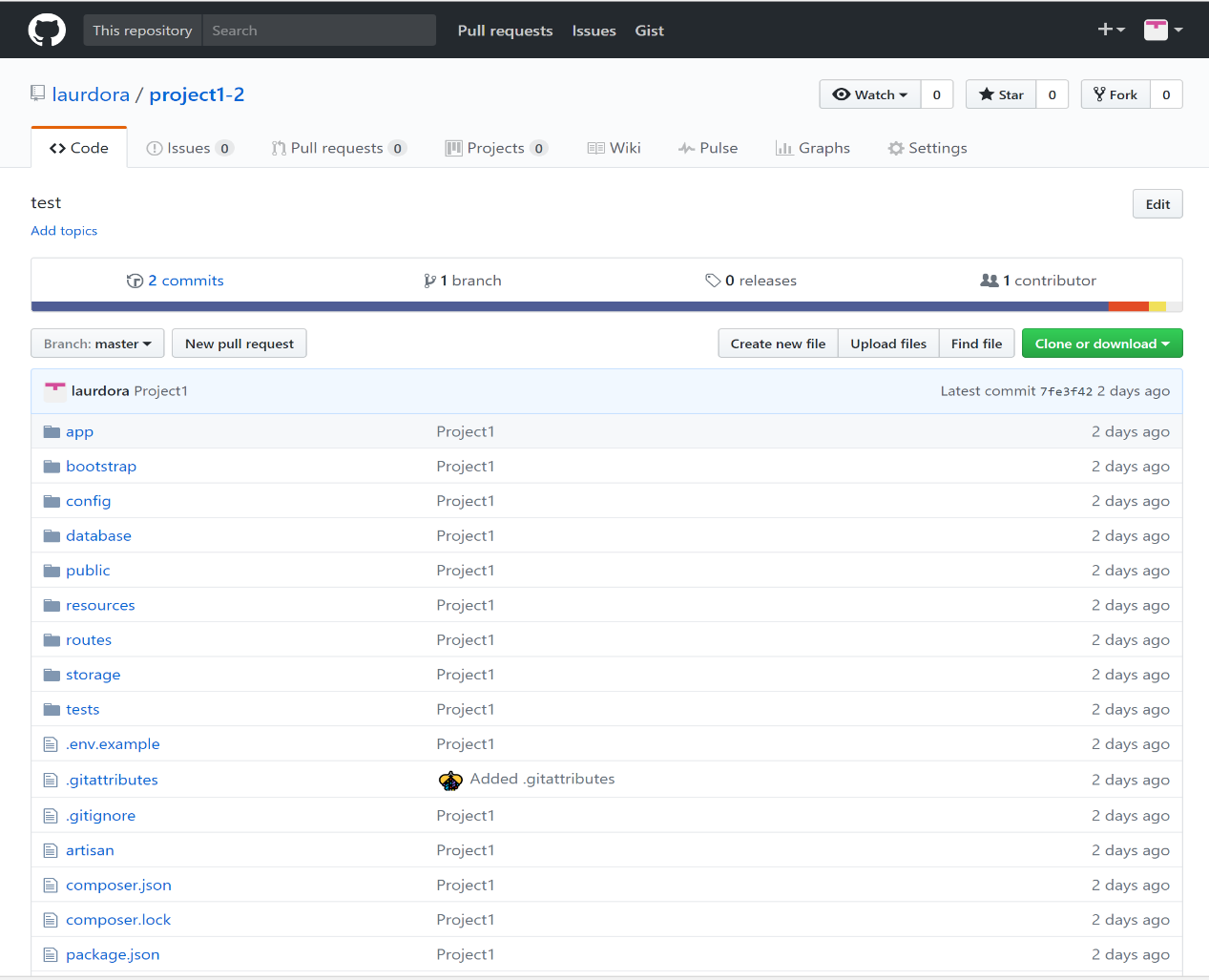
the following command will create a new laravel project named “blog” under “c:\xampp/htdocs” directory.

**Step 3 (Import laravel project on GitHub)**

Since we need to work on the same project folder on GitHub, therefore, you will need to clone or download the laravel project master folder on our GitHub master repository, make a branch of the master repository and start working your located work, later on you can load your changed file to your repository and we can check what you have done, so that we can include your code to our repository. At the end, we will combine all our branch repository and test the project application.

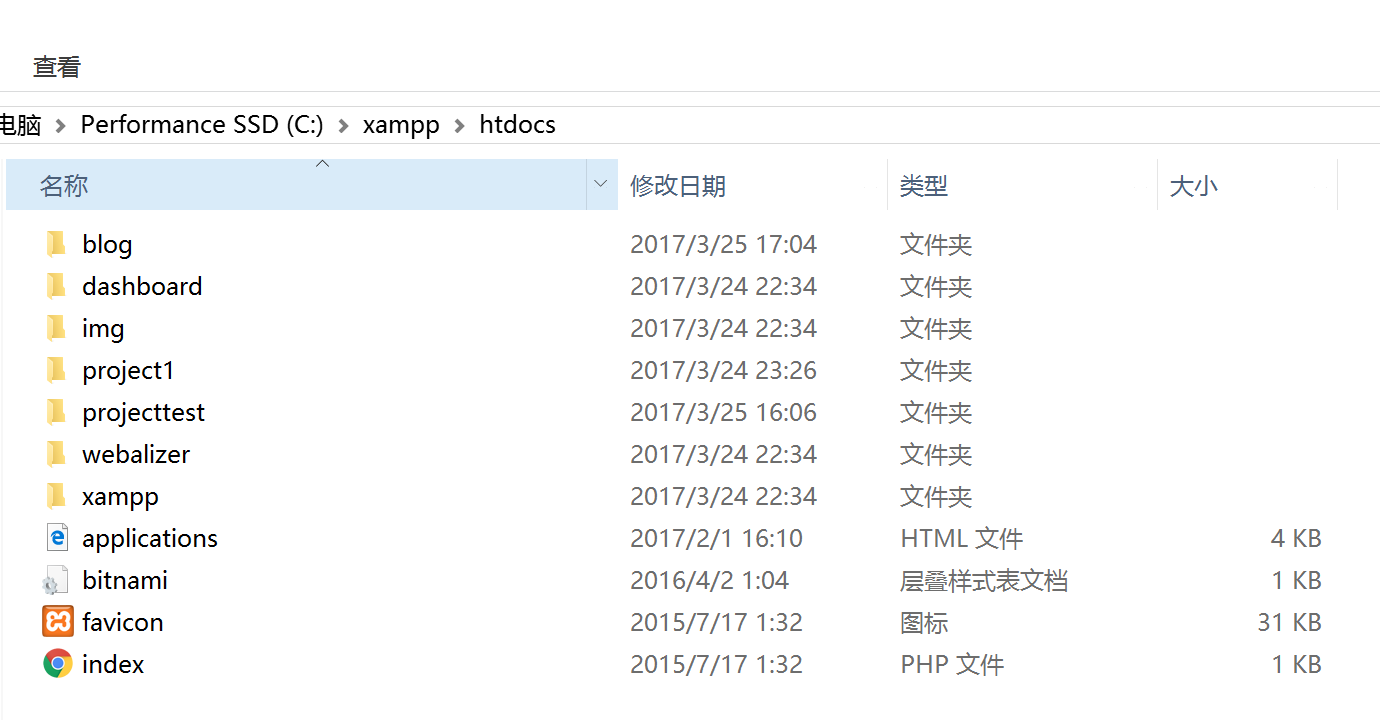
I have uploaded a fresh copy of a new laravel project onto my GitHub

<https://github.com/laurdora/project1-2>



To import the project to your computer up and running, first click on the green button “clone or download” – “download zip”

Then, extract “project1-2-master” folder to the “htdocs” file under your xampp directory ,which in my case is at c:\xampp\htdocs. And change the name of the folder to a name that is easier to remember, I named my project to “project1”.



Inside my “project1” folder, you will see a .env.example file

In your CMD, locate to this folder (in my case which is ”c:\xampp\htdocs\project1”), try to run “copy .env.example .env”, hopefully you will get a new file “.env”

Next, under your project path, run “composer install”, It may take a while to set up your project file up and ready to use.



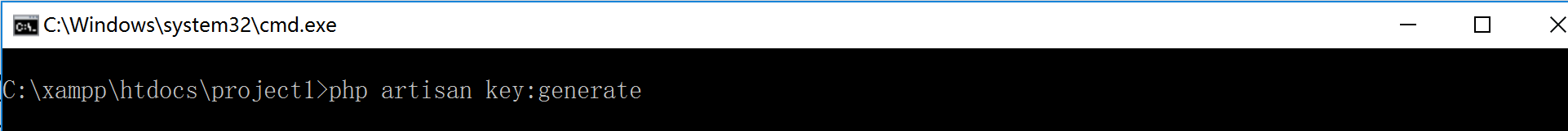
Note: Before installation, make sure the following PHP extension module is enabled:

* PDO PHP Extension
* SQLite PHP Extension
* mbstring PHP Extension

To enable these extension, edit your “php.ini” file under “c://Xampp/php” folder, and uncomment the following lines:

* extension=php\_mbstring.dll
* extension=php\_pdo\_sqlite.dll
* extension=php\_pdo\_mysql.dll

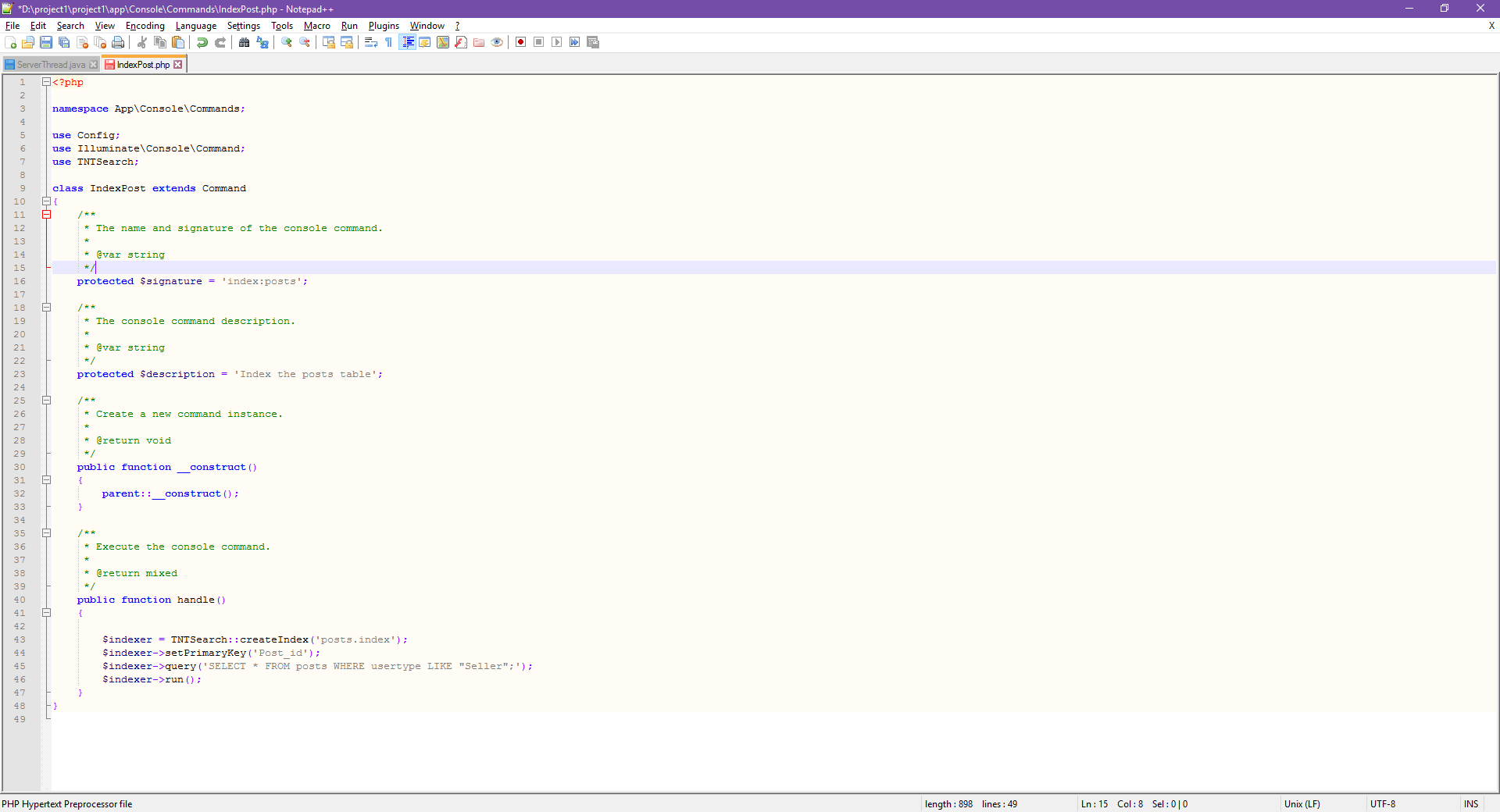
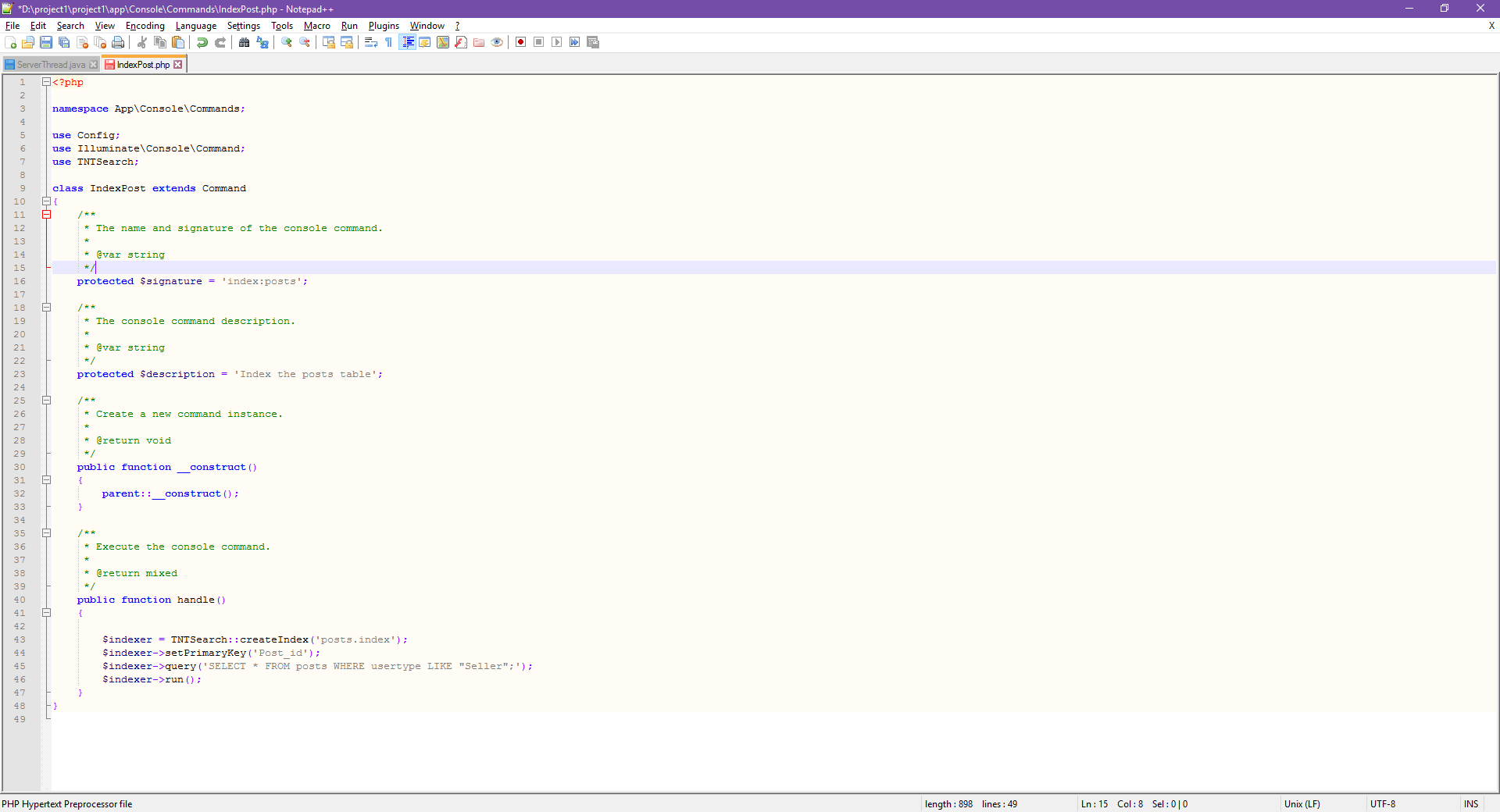
then next, run “php artisan key:generate” it will generate an encryption key for this laravel project, without this your browser will not run laravel framework properly.

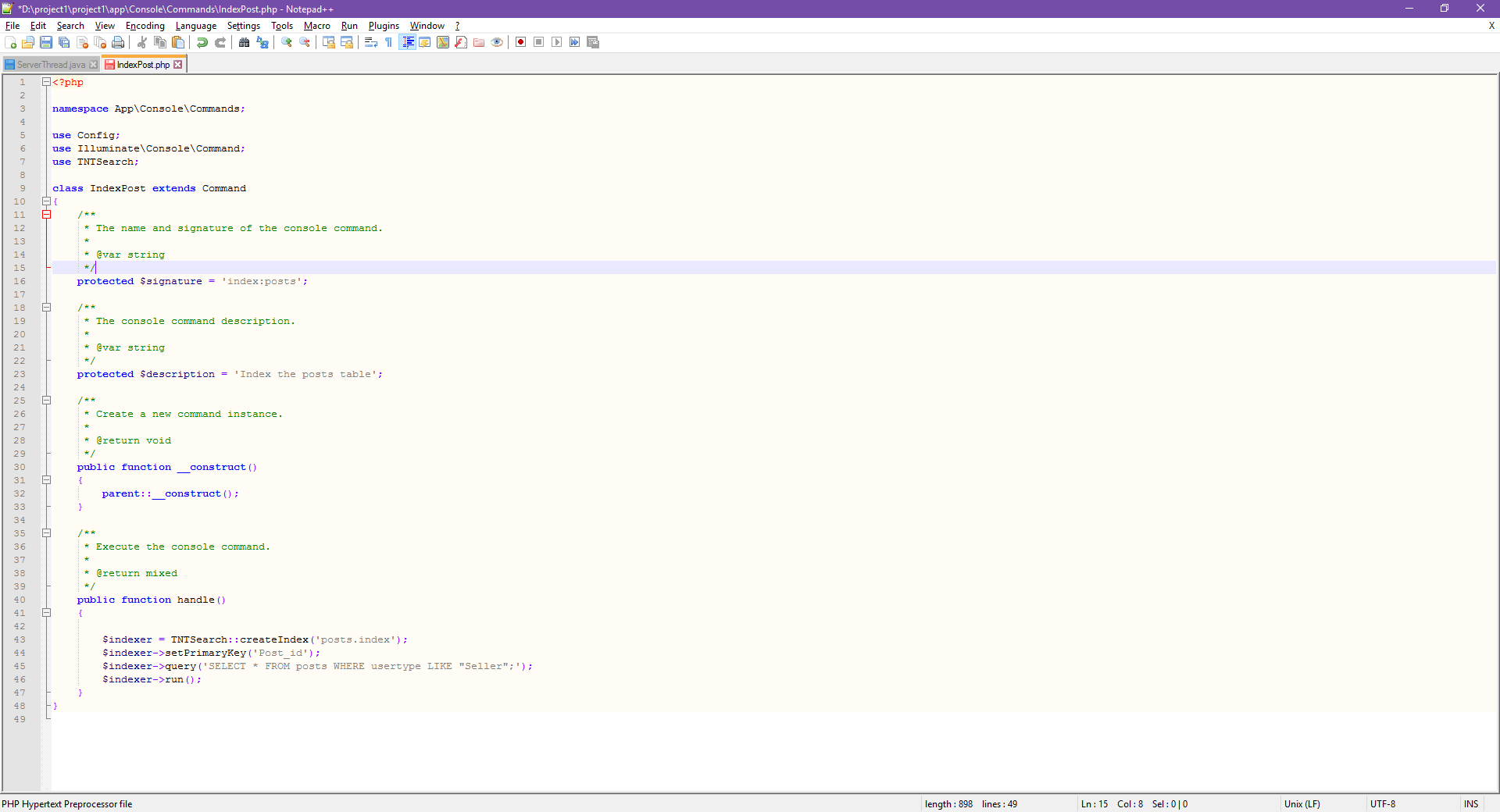


Next, create a link in the public folder from storage folder in order to access image files from storage path. “php artisan storage:link”

Now your laravel project is imported, there are few configurations you need before you can run laravel project on your browser.

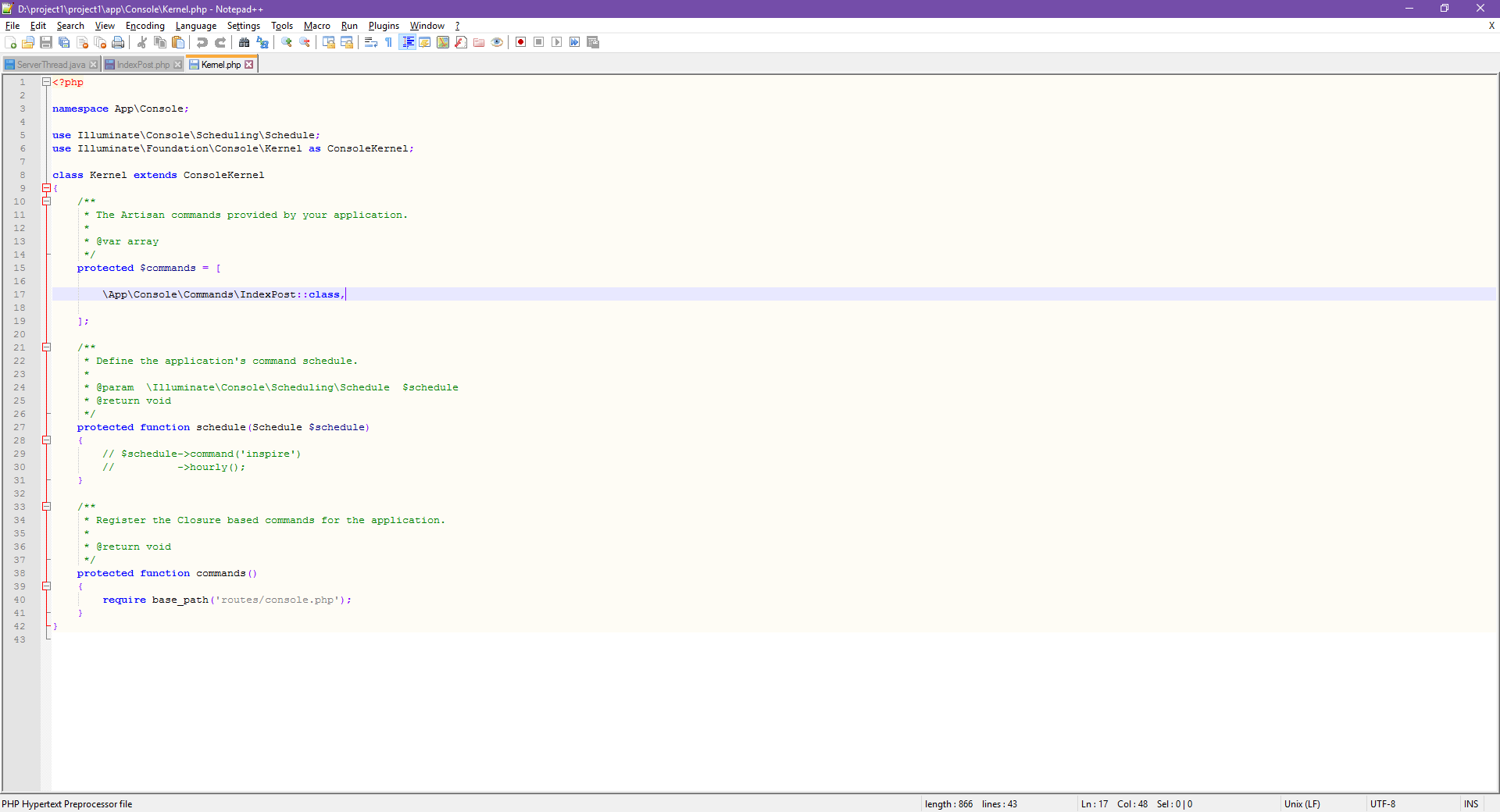
**Step 4 (TNTsearch package configuration)**

1. Installing SQLite extension which was required to run this package.
   1. Type sudo apt-get install -y php5-sqlite into apache command line
   2. Restart apache server
2. Open console and have the path set as the project path
   1. Type “composer require teamtnt/tntsearch”
3. After composer has installed tntsearch
   1. Open app.php file located in the config folder (config/app.php)
   2. Under Service Provider uncomment the following line:
      1. TeamTNT\TNTSearch\TNTSearchServiceProvider::class,
   3. Under Aliases uncomment the following line:
      1. ‘TNTSearch’ => TeamTNT\TNTSearch\Facades\TNTSearch::class,
4. Open console and create an indexer to allow for indexing of a table, in our case we are indexing the posts table to make it searchable.
   1. Type php artisan make:command IndexPost
5. An IndexPost.php file will be made inside App/Console/Commands folder.
   1. Open IndexPost.php and edit the following
      1. Add ‘use’ commands Config; and TNTSearch;
      2. Change protected $signature to ‘index:posts’; and protected $description to ‘Index the posts table’;
      3. Add the handle() function to gather all the information from columns to turn it into one searchable index.



6. Open Kernal.php located in App\Console\Commands folder.

1. Add into the protected $commands section
   1. \App\Console\Commands\IndexPost::class,

7. Run php artisan index:posts in console to create searchable index which then can be found in the storage folder

8. Copy teamtntbackup folder from Github repository and paste into vendor folder.

1. Rename teamtntbackup to teamtnt

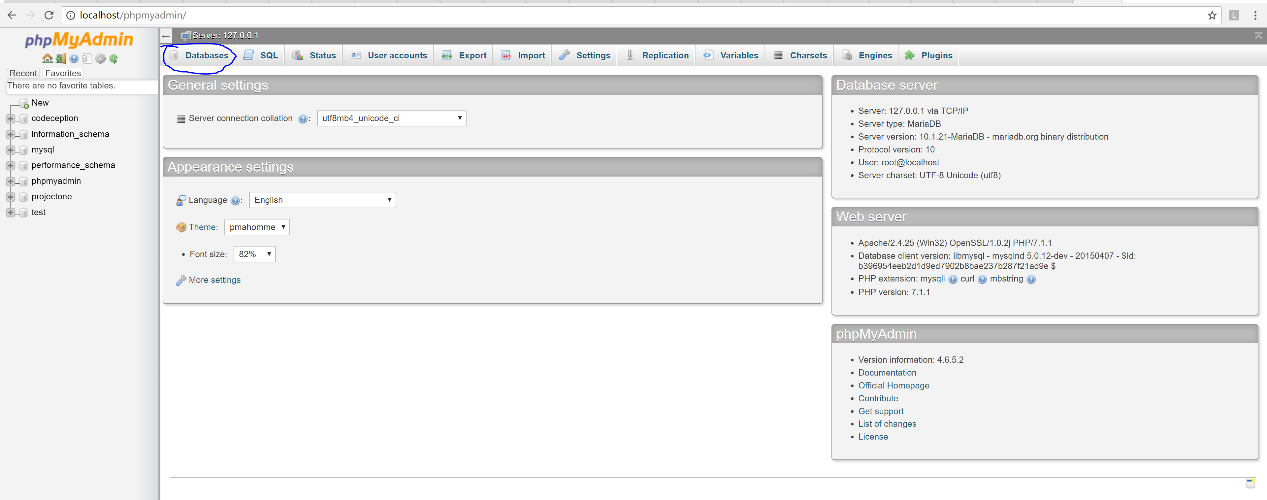
9. Ensure database config port is set to the appropriate port to connect to desired SQL database.

10. Open console to enable read and write permissions for teamtnt folder along with posts.index and IndexPost.php.

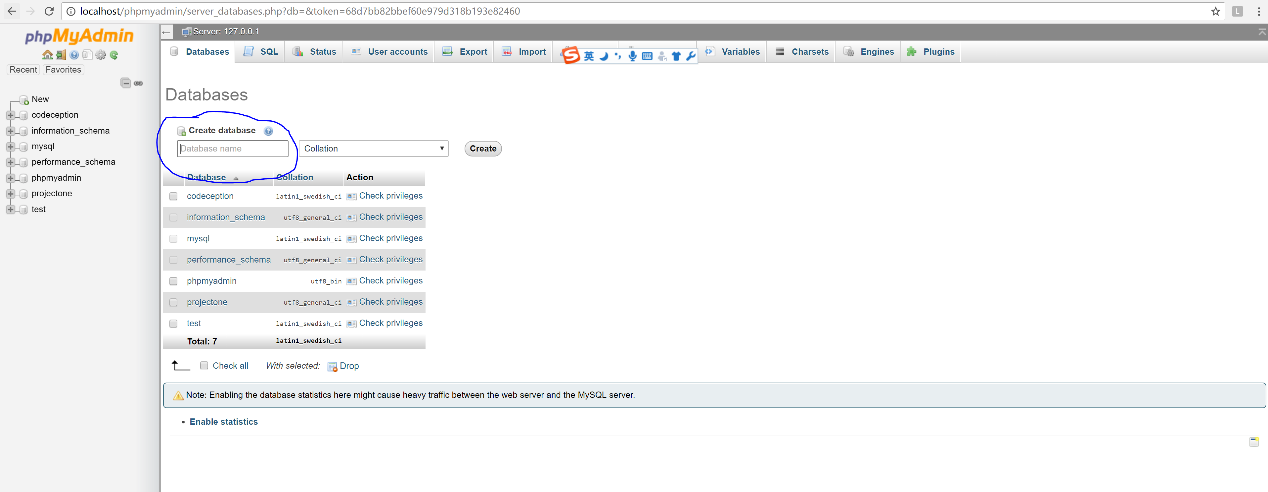
**Step 5 (Set up mysql connection in laravel)**

First of all open your browser, go to “yourservername/phpmyadmin” or

“localhost/phpmyadmin”, you will see this.

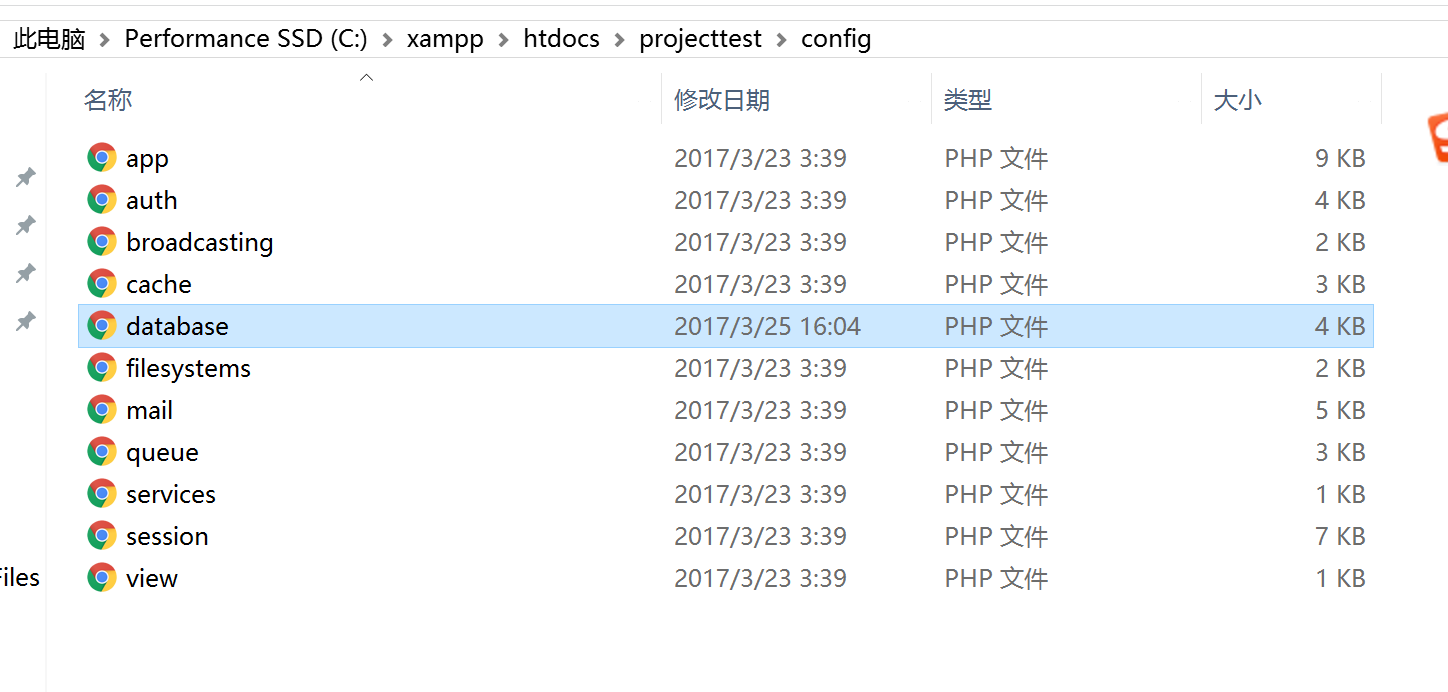


Lets click on database, where I circled above.



Create a database, name your database and press create.

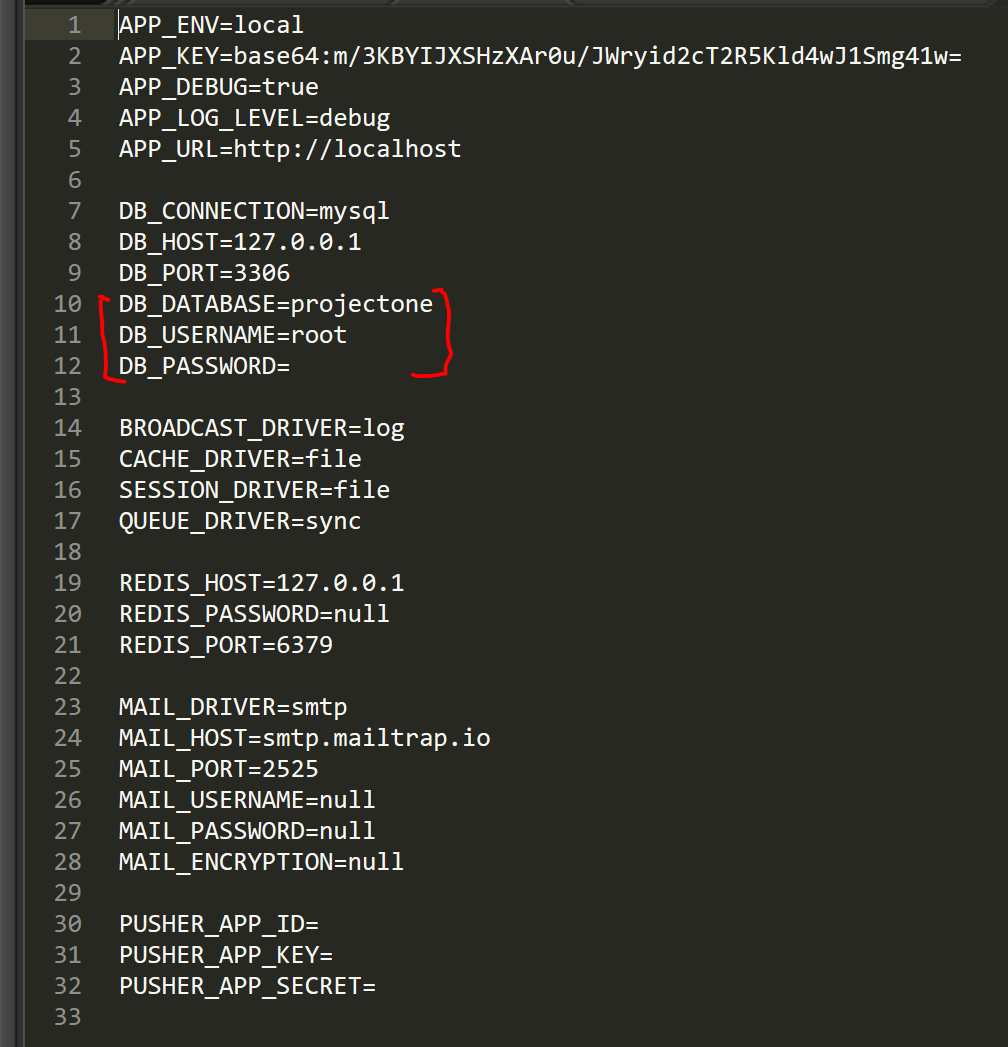
Next, go to your laravel project directory, under “/config” folder and edit database.php file



Over the following line, change the database name to the database you created from phpmyadmin, with username “root”, leave password blank ‘’ and save your change.



Next, go back to your laravel project directory, edit “.env” file



Change the database name to the one you create on phpmyadmin, username as “root” and no password, then save file.

Finally, in you cmd, under laravel project path, run “php artisan migrate”, it should insert two tables (laravel default tables) to your database.

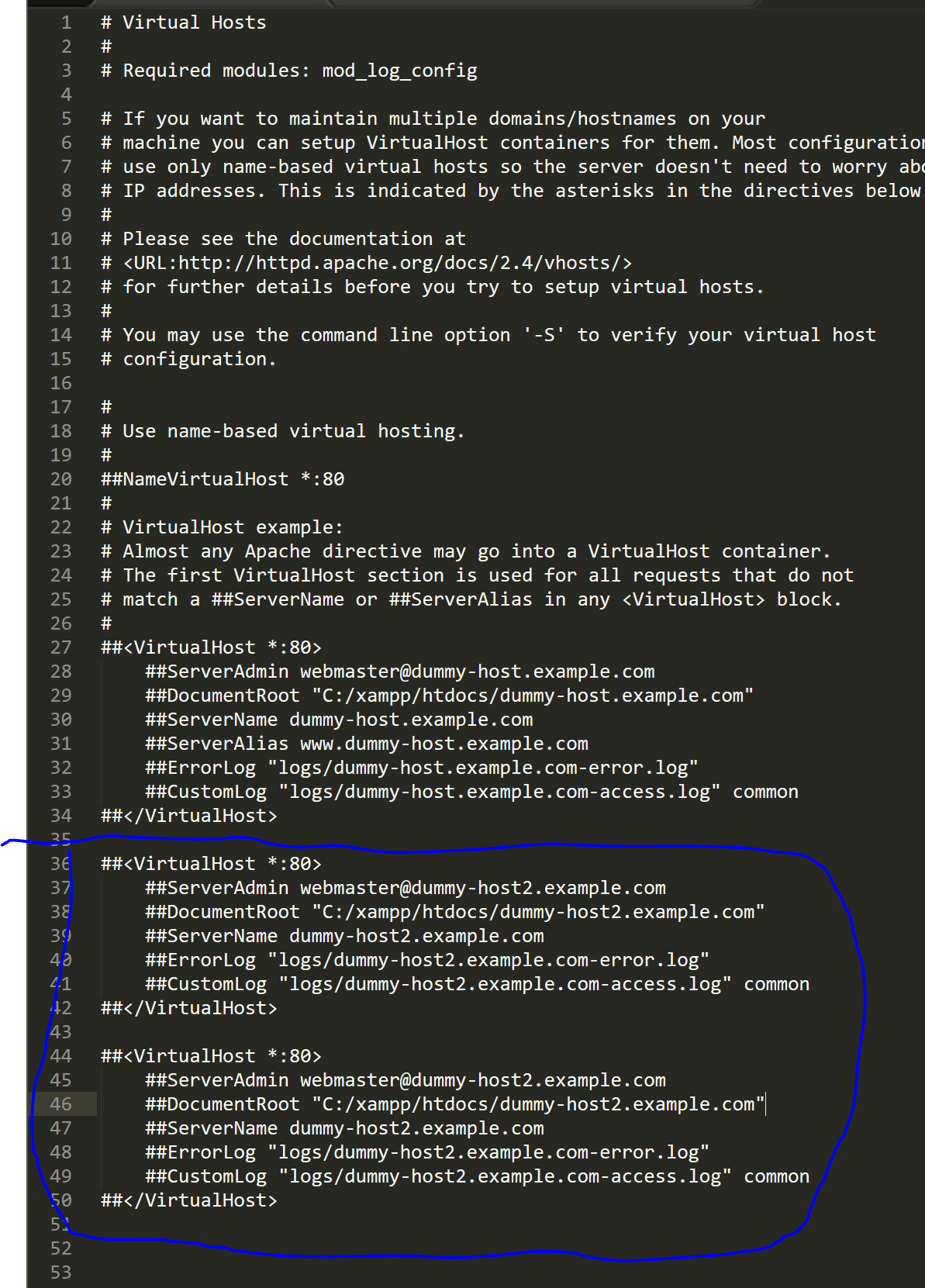
if error occur, run “composer dump-autoload” then “php artisan migrate”

go back to phpmyadmin and check if your database included two laravel default tables, if it does, then your database has successfully connected with laravel.

**Step 6 (virtualhost configuration)**

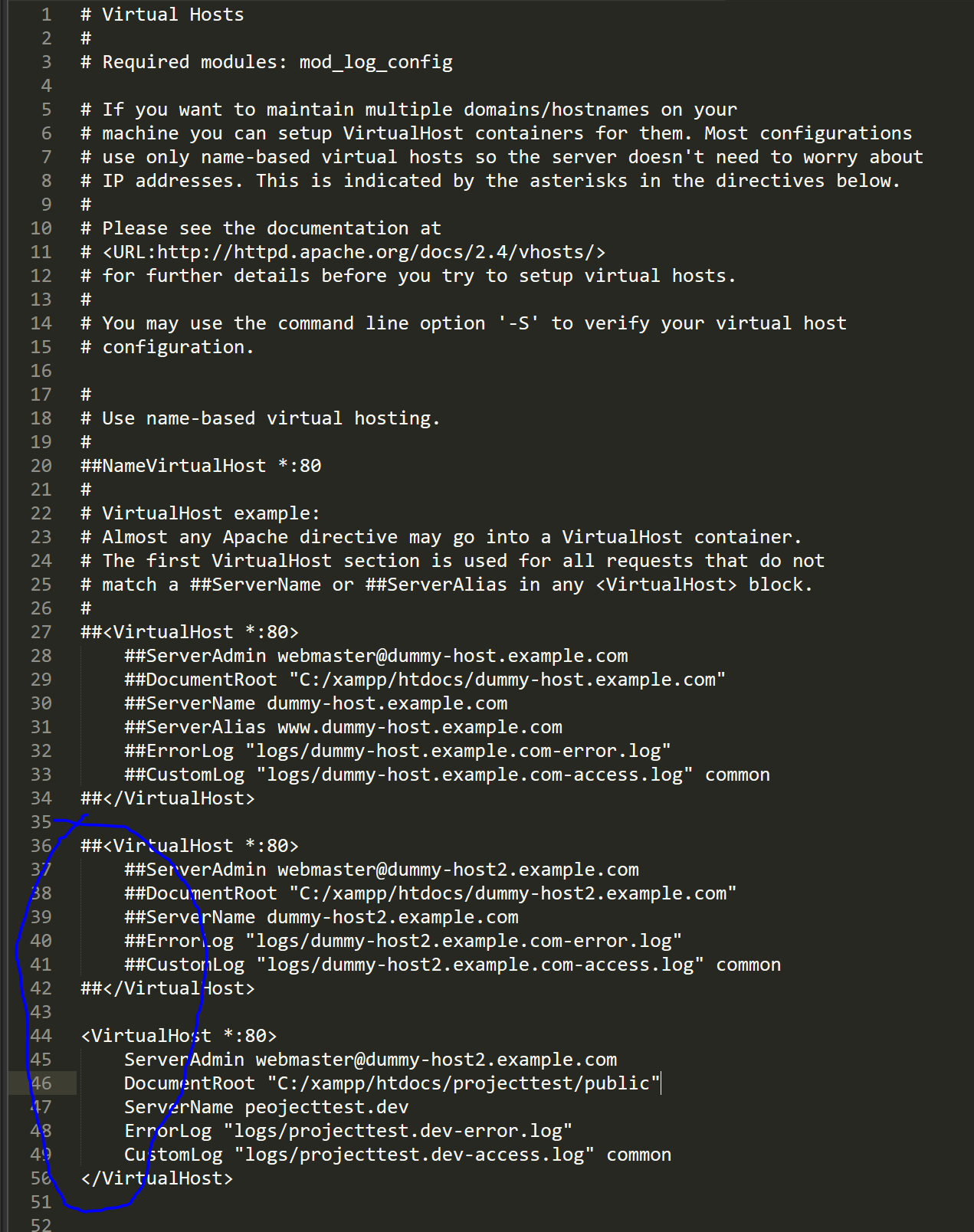
First, you need to configure your virtual host, go to your “xampp” directory, inside “apache/conf/extra “ folder, find “httpd-vhosts.conf” file and edit this file using notepad or anyother editing tool.

Scroll all the way down, copy and paste everything in <VirtualHost\*:80></VirtualHost> to a new line like the following



On the last <virtual \*:80> block

Get rid of “##” for each line, it will execute the virtual host configuration.



We need to change DocumentRoot to your laravel project, public directory, in my case, is in “c:/xampp/htdocs/project1/public” (don’t worry about the projecttest, type in the name of your laravel project you set, then /public).

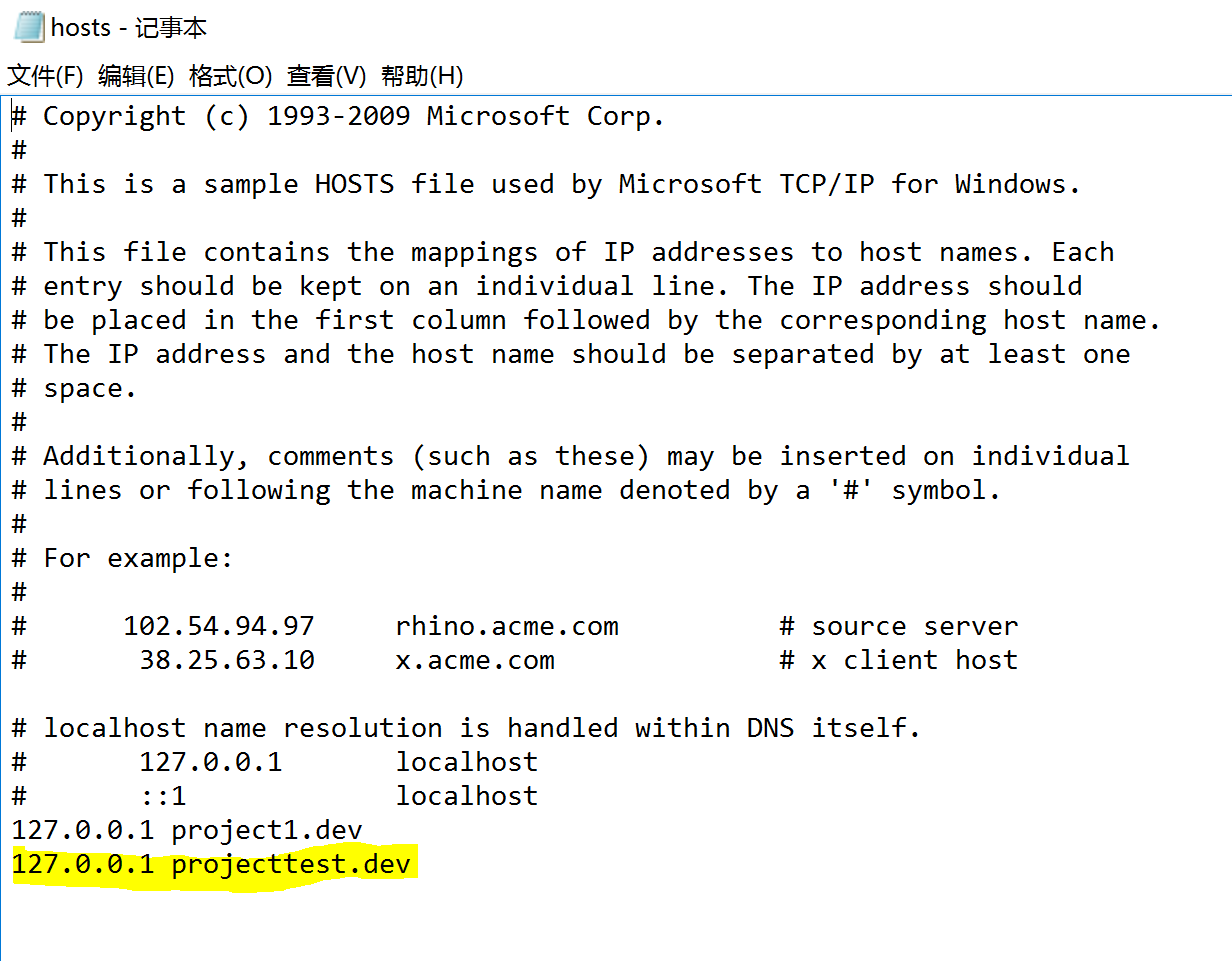
Name your server, any name would work, for example, laraveltest.dev, or test.dev etc…

Then, on ErrorLog and CustomLog, replace “dummy-host2.example.com” to

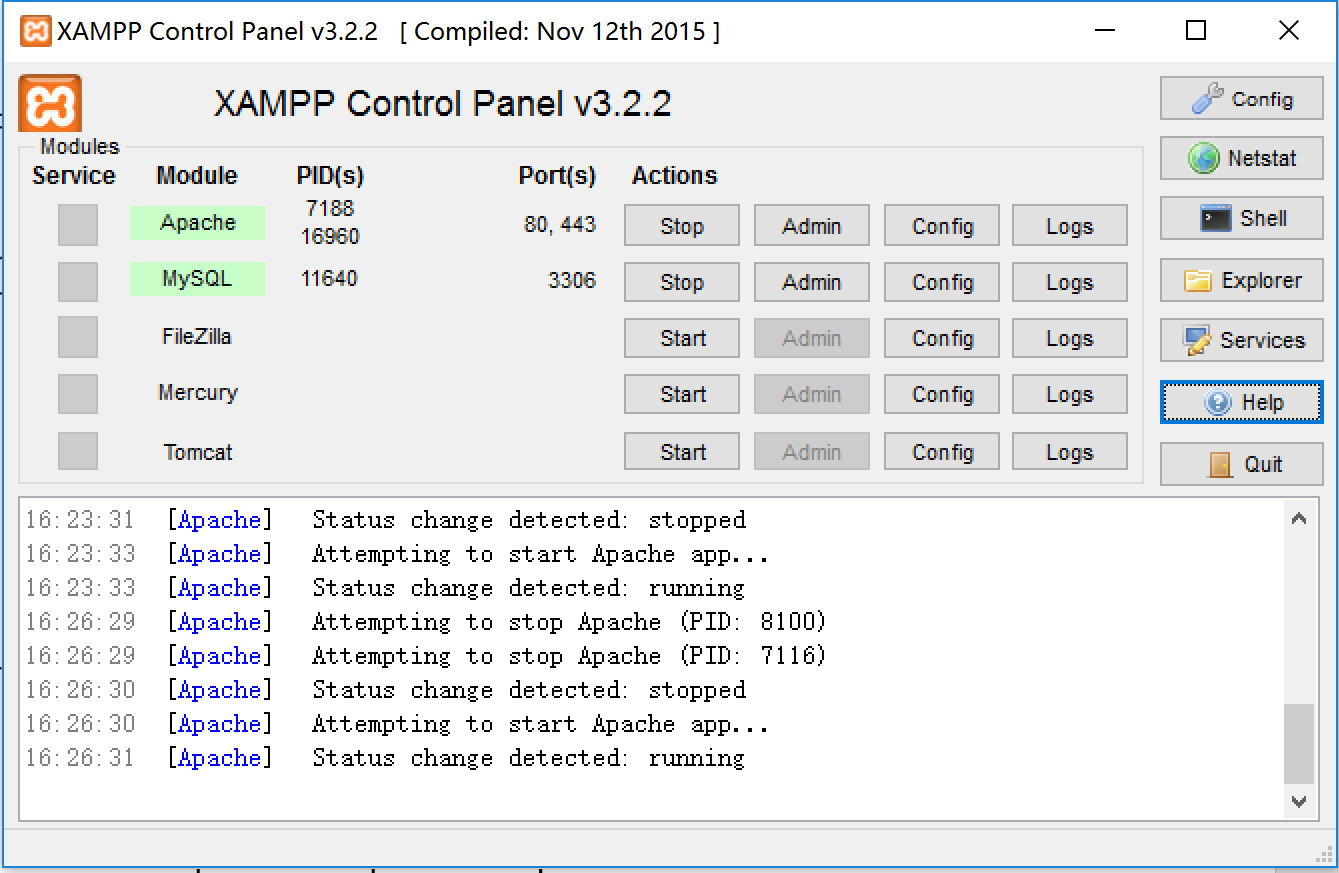
“(servername).dev”, just like the format above, then save.

Next, go to this directory “c:/windows/system32/drivers/etc”, you should be able to see a file call “hosts”, run the file as admin and open with notepad, and add “127.0.0.1 (servername)” at the end, then save.

In my case:



Then go to you xampp panel, stop apache server and run again. (if you haven’t run xampp, run xampp and press run apache and run mysql)



Now you should be able to run your laravel project on your browser by entering server name, or localhost as URL.

Later if you want to import another laravel project and test it on your browser, pls redo everything from step 3.

*Refer to the User Guide for usage information.*