**Project1 - Buyer/Seller match making website**

**Tools - research material**

**Laravel research and instructions** by Jiajun He

* open-source PHP framework for rapid development
* Uses composer to manage all dependencies and ensures they are up to date
* has powerful command line functions that allow for easy setup (eg. php artisan make:auth = makes login and registration views)
* fully supports the MVC design pattern in the PHP environment
* allows for easy addition of other third party frameworks that can also be managed through the composer application
* built in support for SQL relational databases
* built in URL routing features for loading new views and controllers
* very easy to set up and start development

Instructions from

1.<https://laravel.com/docs/5.4/installation>

2.See our “Laravel + Xampp installation guide”

1. Download and install Composer. [https://getcomposer.org/download](https://getcomposer.org/download/)
2. Add the Composer path to your system’s environment variables. This will vary

between systems. On Windows 10 it was *C:\Users/USERNAME\AppData\Roaming\Composer\vendor\bin*

Once you know this path add it to your environment variables.

System Properties > Advanced > Environment Variables > Path

1. Open cmd and run -

composer global require "laravel/installer"

1. You need to clone the BitBucket repository to a new folder. If you are using XAMPP for your local web server then do this to the htdocs folder. I recommend naming this folder *clevo-rmit* so we are all using the same name.
2. In CMD navigate to this folder.

cd c:/xampp/htdocs/project1

1. Run the following command

composer update

OR

composer install

1. With XAMPP running you should now be able to see a Laravel landing page at http://localhost/project1/public/

**Google reCAPTCHA (Human Verification) API** by Jiajun He

* A google service to protect website from spam and abuse
* Uses advanced risk analysis engine and adaptive CAPTCHAs to keep automated software from engaging in abusive activities on client-server site.
* Easy to use on human, hard on bots
* Free to sue
* Must register a reCAPTCHA API key pair with a domain name
* Can return JSON

**Usage:**

* **Client site integration:**

To create a reCAPTCHA widget, a <div> field with class=”g-recaptcha” and datasitekey=”6Lc2RCAUAAAAAAr3sx1t\_cSUeyMufoebNU6CvbZc”(Here is the example of our site key) is added onto your preferred html file, and make sure the google api java script is loaded on the same html page as well.

<script src='<https://www.google.com/recaptcha/api.js>'></script>

* **Server site integration**

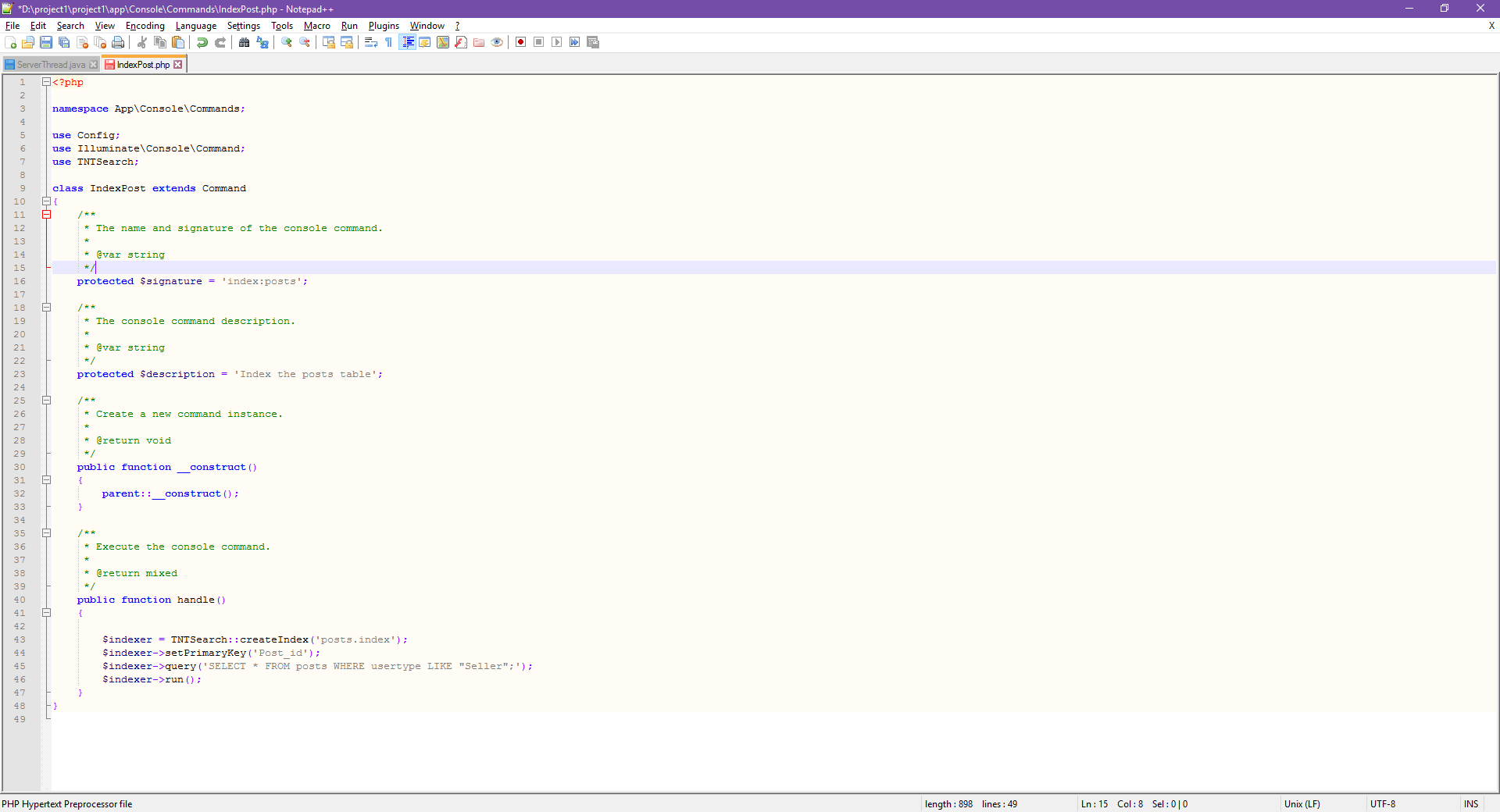
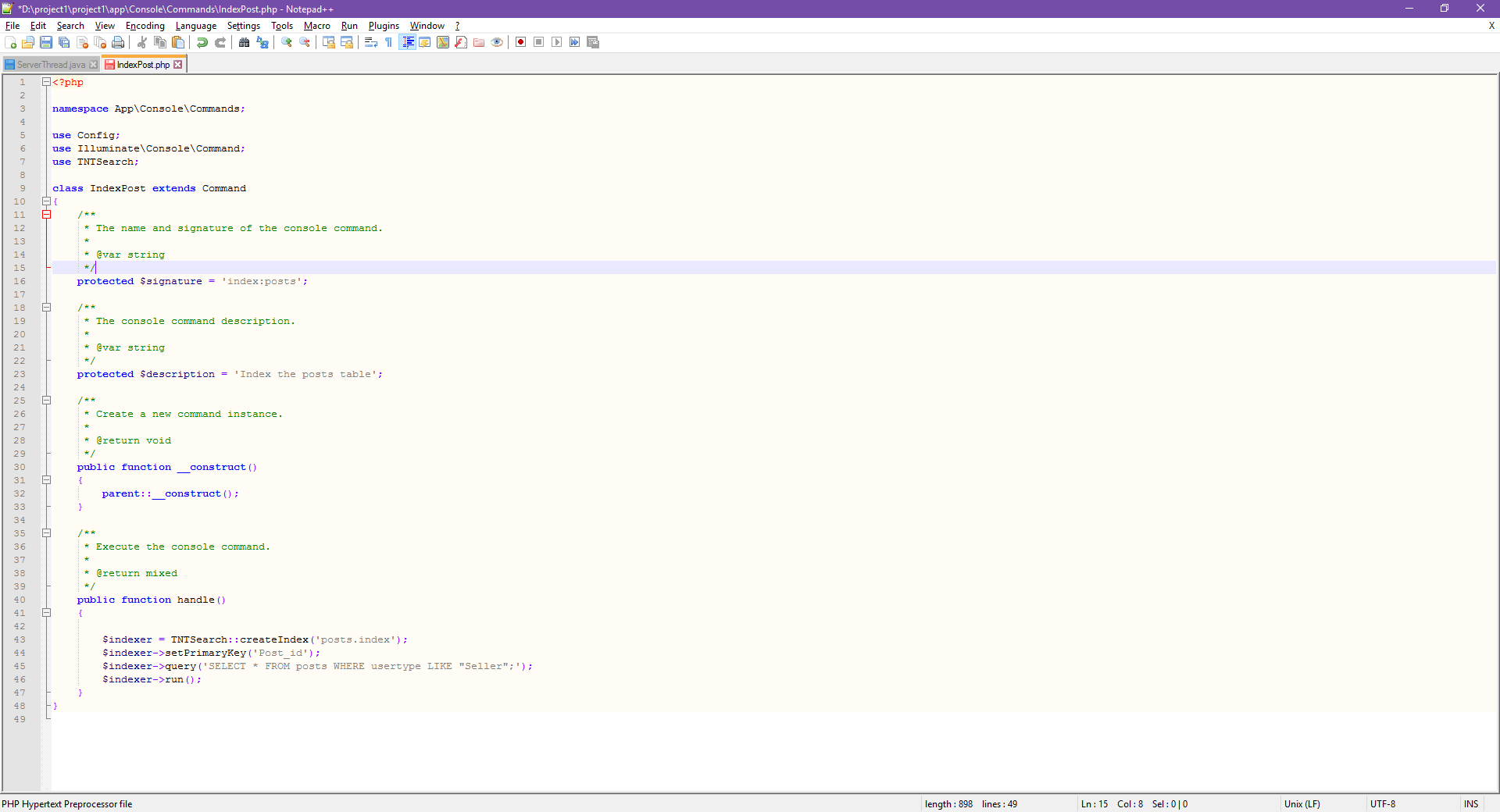
When users submit the form where you integrated reCAPTCHA, they will get as part of the payload a string with the name "g-recaptcha-response". In order to check whether Google has verified that user, send a POST request with these parameters:

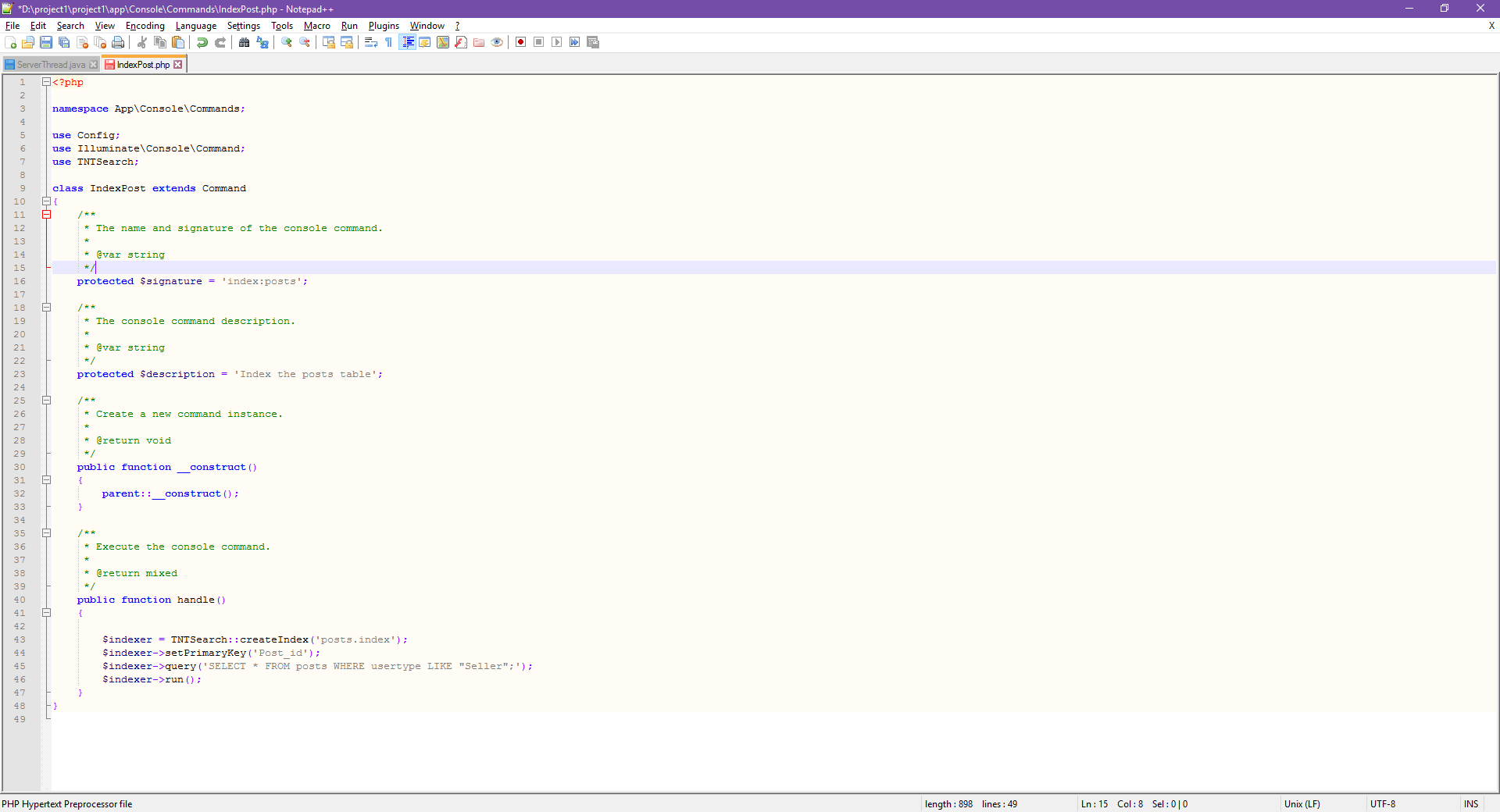
URL: https://www.google.com/recaptcha/api/siteverify

|  |  |
| --- | --- |
| secret (required) | 6Lc2RCAUAAAAAP73clpEH6artwHRESrvZbz209SX (example) |
| response (required) | The value of 'g-recaptcha-response'. |
| remoteip | The end user's ip address. (optional) |

Installing TNTStudio search package and indexing

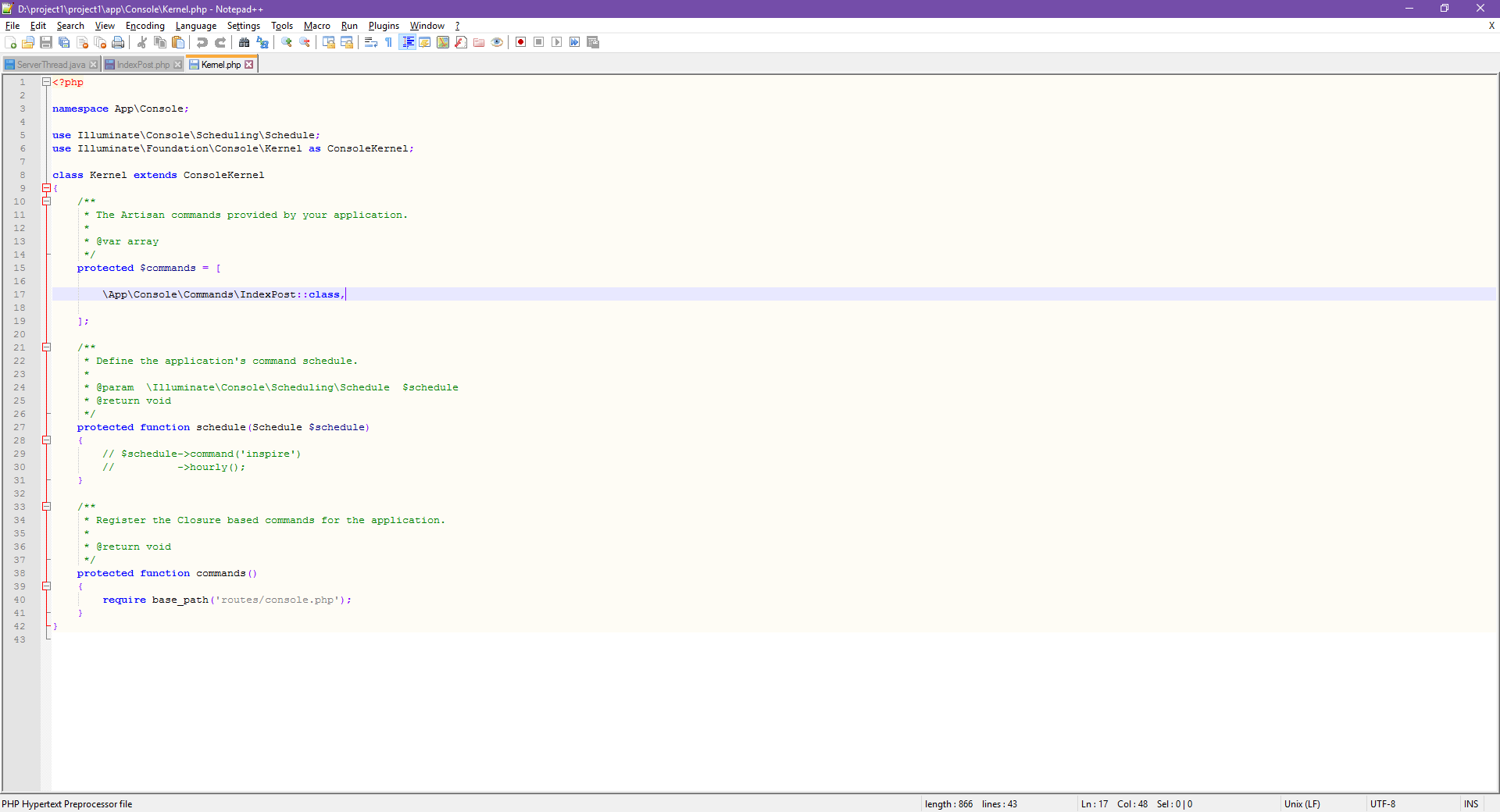
The use of the TNTStudio search package was installed and implemented in helping improve search functionality on the website. This package allowed us to search multiple keywords rather than locking in the search to just one keyword.

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 add:
      1. TeamTNT\TNTSearch\TNTSearchServiceProvider::class,
   3. Under Aliases add:
      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.