RANTS TUTORIAL

Demo 4, CMSC 129 LAB

Prepared by: Ara Abigail E. Ambita

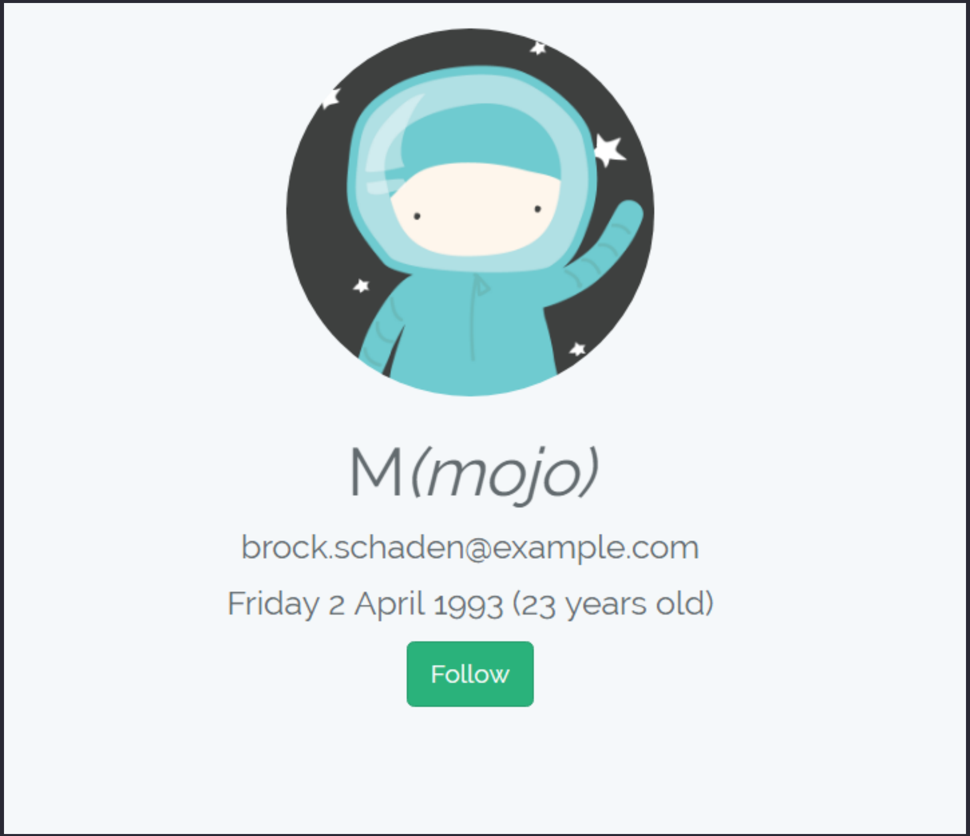
Last demo session…

- Update
- Delete

Next Task: Add rants to my profile

My rants

Currently the profile design



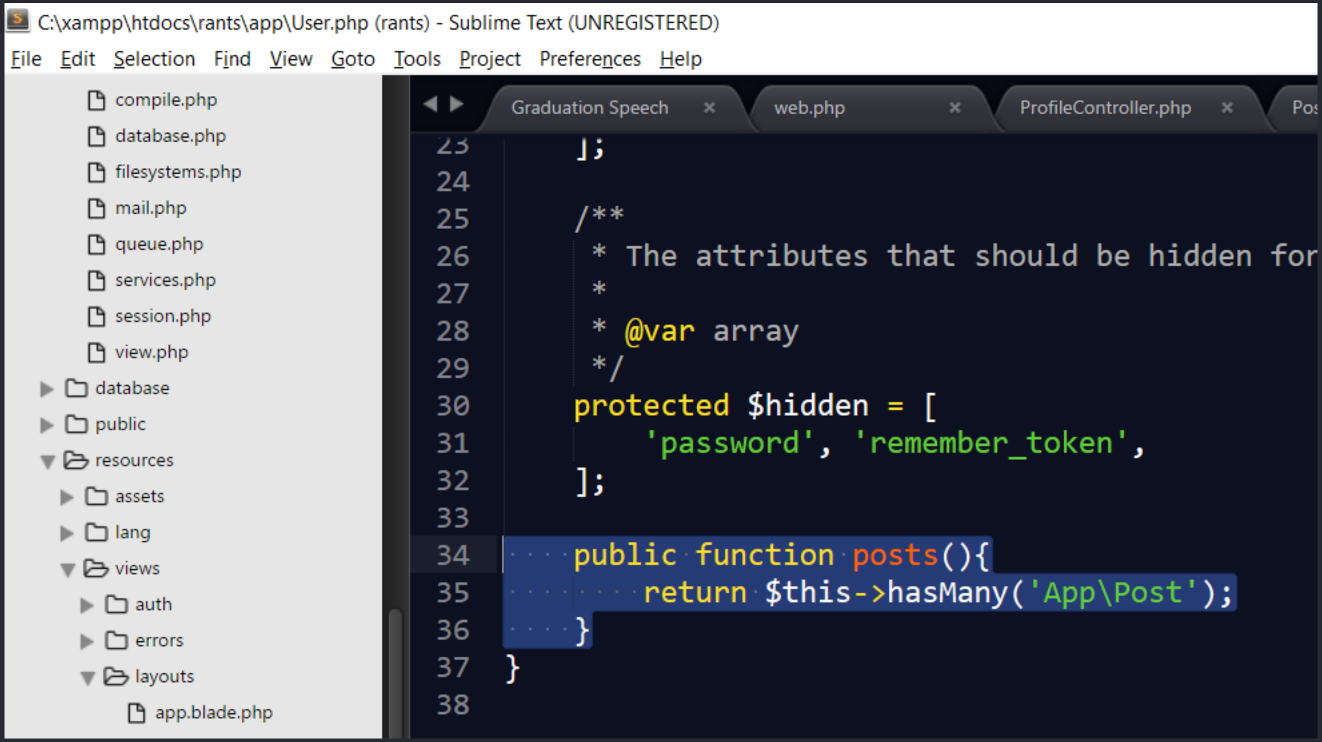
M*(mojo)*

brock.schaden@example.com

Friday 2 April 1993 (23 years old)

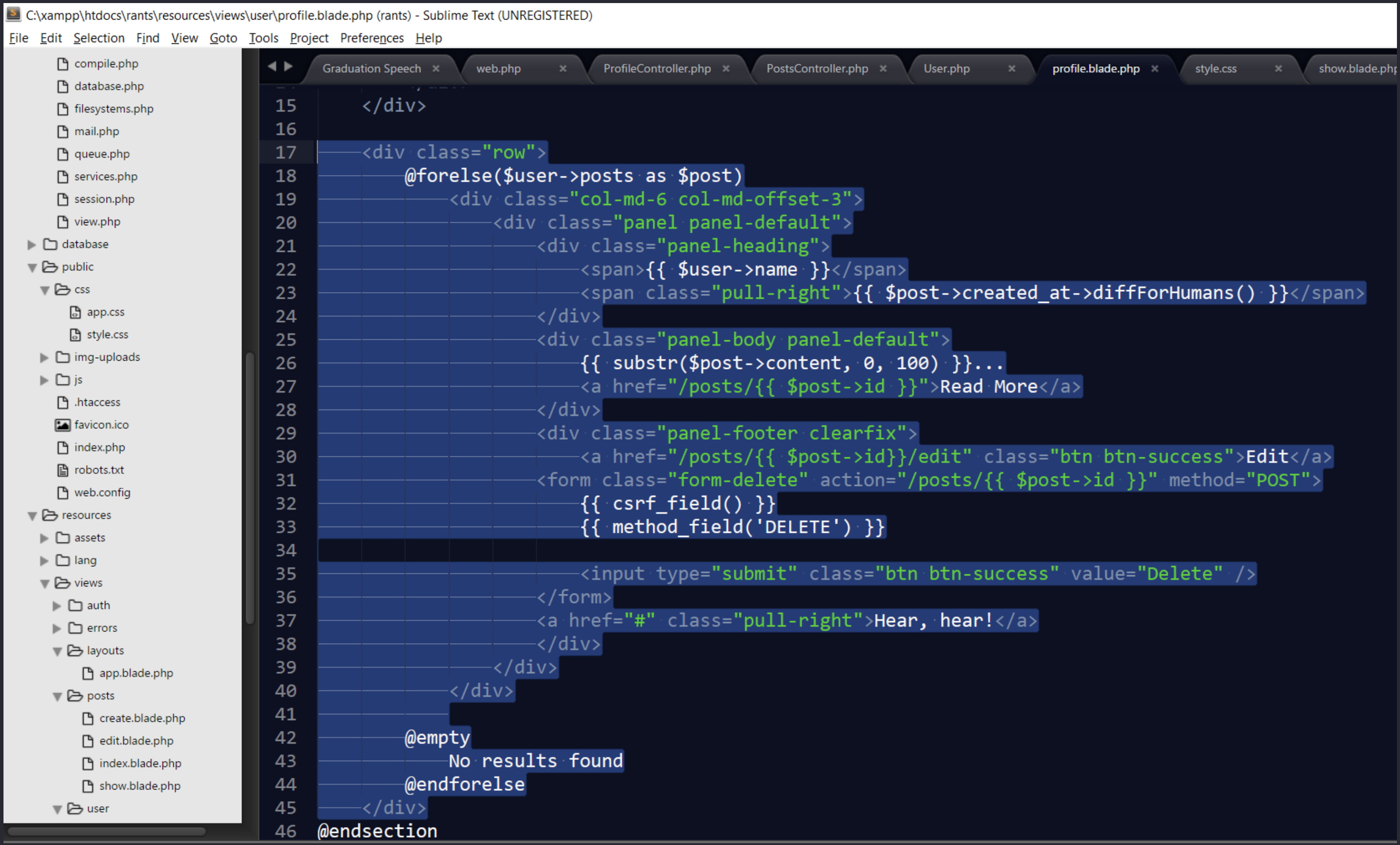
Open User.php, model for users table

Add these lines of codes



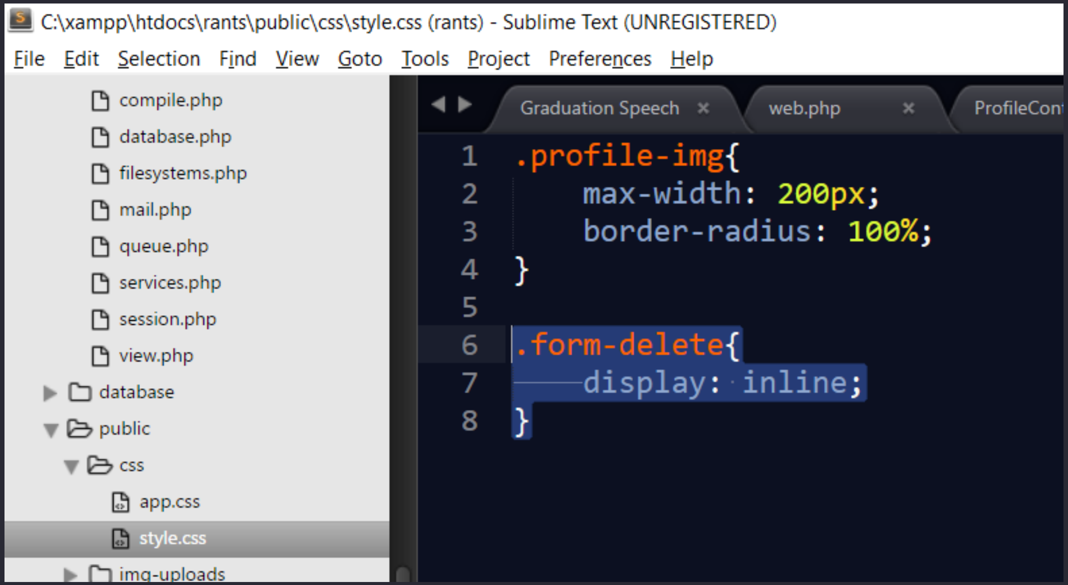
```
    ];

    /**
     * The attributes that should be hidden for
     *
     * @var array
     */
    protected $hidden = [
        'password', 'remember_token',
    ];

    public function posts(){
        return $this->hasMany('App\Post');
    }
}
```

Modify profile.blade.php



```
    </div>

    <div class="row">
        @forelse($user->posts as $post)
            <div class="col-md-6 col-md-offset-3">
                <div class="panel panel-default">
                    <div class="panel-heading">
                        <span>{{ $user->name }}</span>
                        <span class="pull-right">{{ $post->created_at->diffForHumans() }}</span>
                    </div>
                    <div class="panel-body panel-default">
                        {{ substr($post->content, 0, 100) }}...
                        <a href="/posts/{{ $post->id }}">Read More</a>
                    </div>
                    <div class="panel-footer clearfix">
                        <a href="/posts/{{ $post->id}}/edit" class="btn btn-success">Edit</a>
                    <form class="form-delete" action="/posts/{{ $post->id }}" method="POST">
                        {{ csrf_field() }}
                        {{ method_field('DELETE') }}

                        <input type="submit" class="btn btn-success" value="Delete" />
                    </form>
                    <a href="#" class="pull-right">Hear, hear!</a>
                    </div>
                </div>
            </div>

        @empty
            No results found
        @endforelse
    </div>
@endsection
```

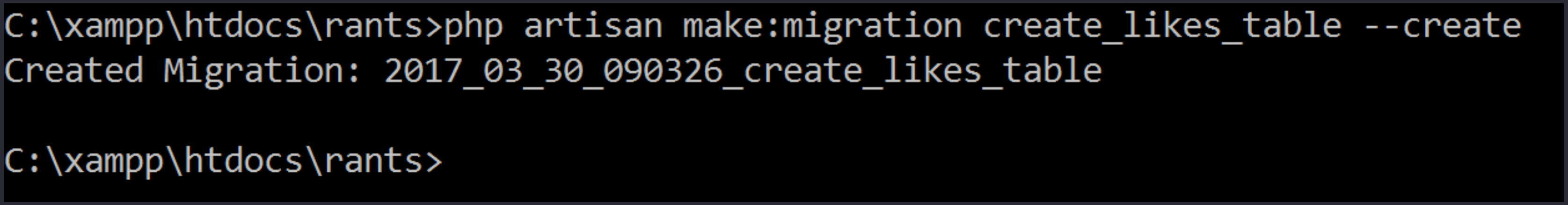
Some changes with style.css



```
.profile-img{
    max-width: 200px;
    border-radius: 100%;
}

.form-delete{
    display: inline;
}
```

Next Task: Like!

Create a migration table for Likes

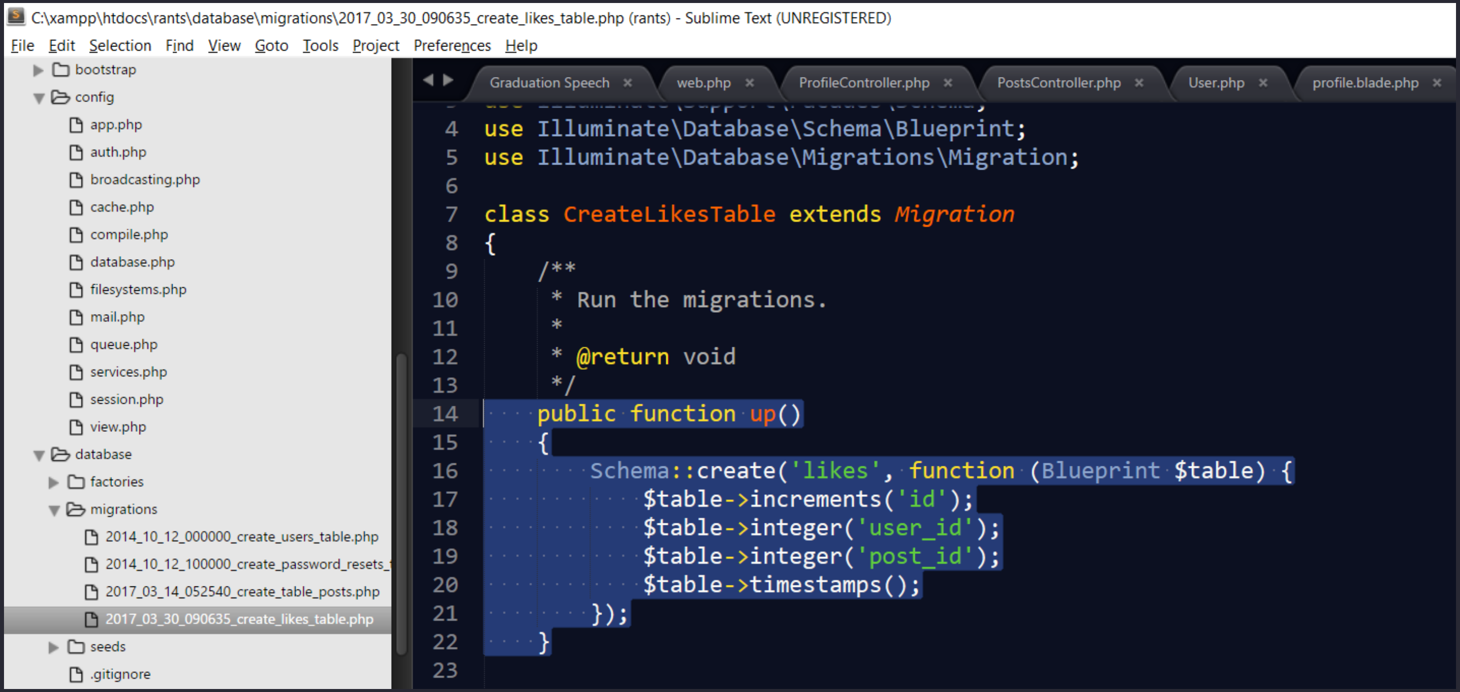


```
C:\xampp\htdocs\rants>php artisan make:migration create_likes_table --create
Created Migration: 2017_03_30_090326_create_likes_table

C:\xampp\htdocs\rants>
```

- There are different ways on how we can implement this

Open the migration file we have created

Add these codes

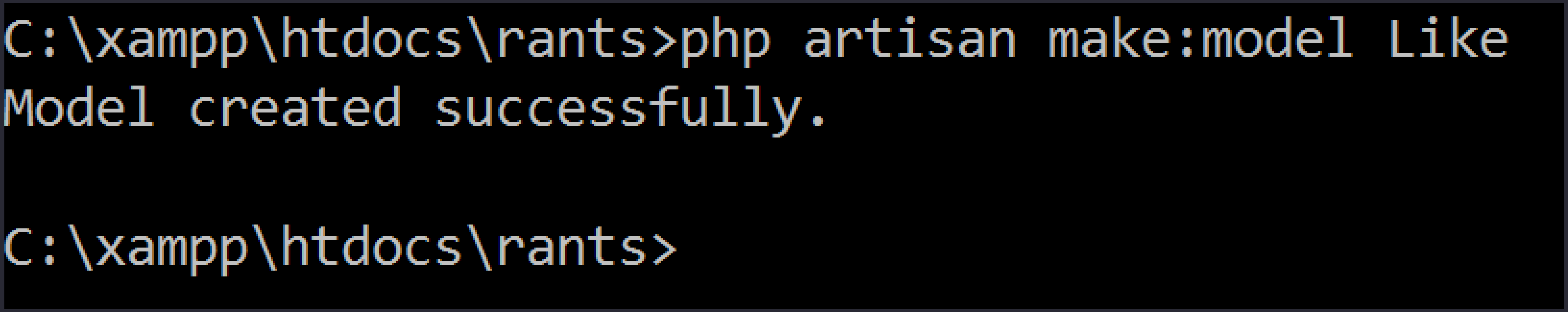

C:\xampp\htdocs\rants\database\migrations\2017_03_30_090635_create_likes_table.php (rants) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
bootstrap
config
app.php
auth.php
broadcasting.php
cache.php
compile.php
database.php
filesystems.php
mail.php
queue.php
services.php
session.php
view.php
database
factories
migrations
2014_10_12_000000_create_users_table.php
2014_10_12_100000_create_password_resets_
2017_03_14_052540_create_table_posts.php
2017_03_30_090635_create_likes_table.php
seeds
.gitignore
Graduation Speech
web.php
ProfileController.php
PostsController.php
User.php
profile.blade.php

```php
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreateLikesTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('likes', function (Blueprint $table) {
            $table->increments('id');
            $table->integer('user_id');
            $table->integer('post_id');
            $table->timestamps();
        });
    }

```

Create a model for likes table

Type in this command in terminal

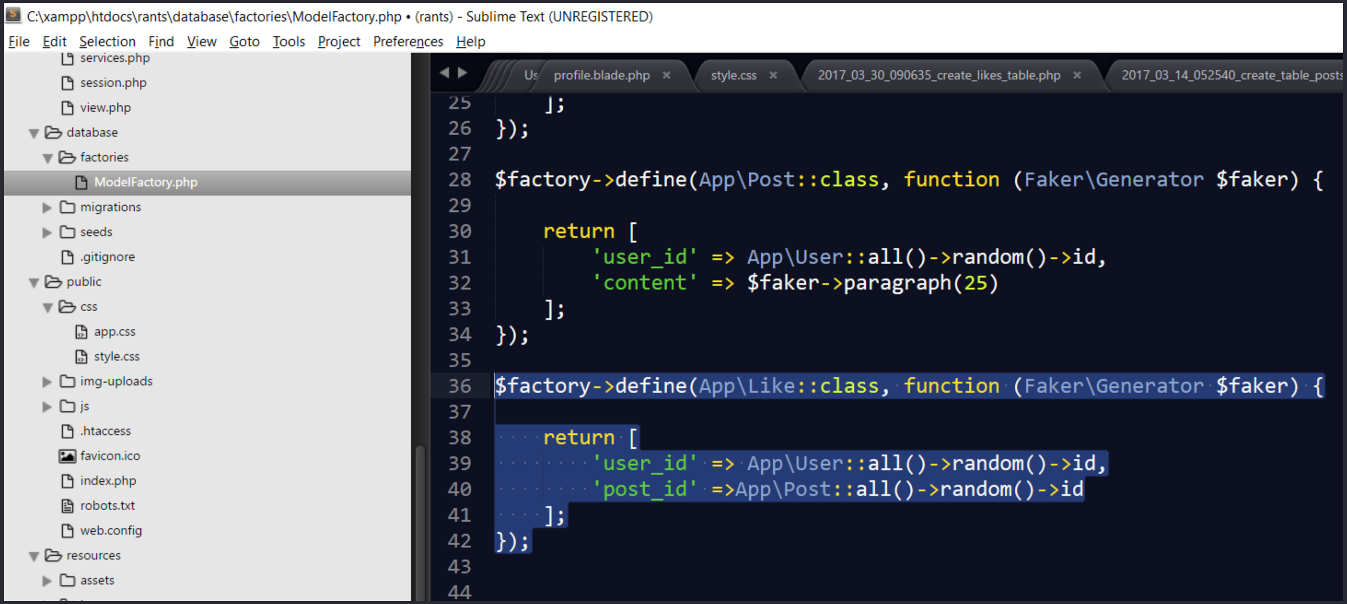
(or you can manually create the file)



```
C:\xampp\htdocs\rants>php artisan make:model Like
Model created successfully.

C:\xampp\htdocs\rants>
```

Now, let's create fake data for our likes table

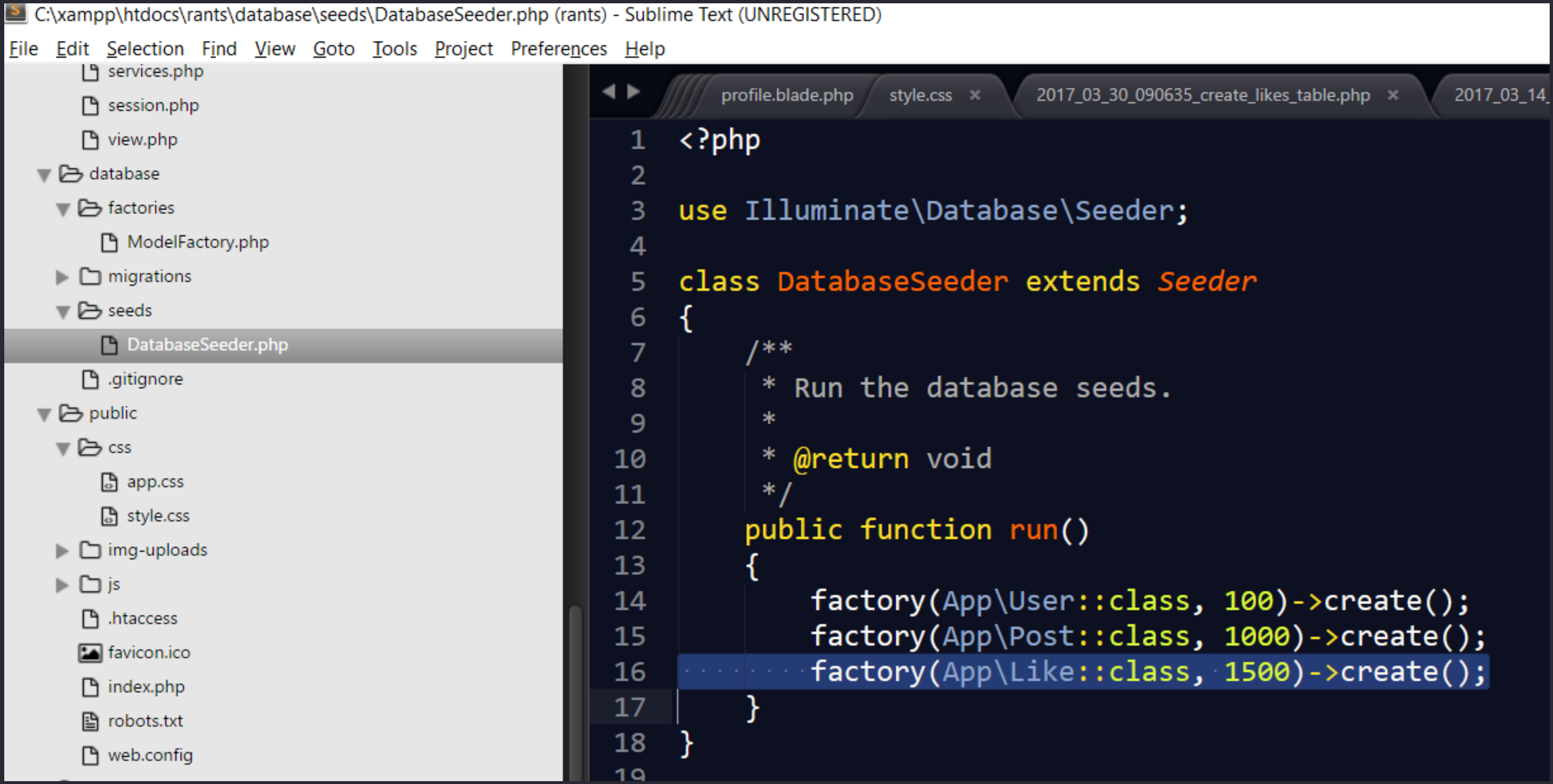
Define the factory for Likes Table in ModelFactory.php


C:\xampp\htdocs\rants\database\factories\ModelFactory.php • (rants) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
services.php
session.php
view.php
database
factories
ModelFactory.php
migrations
seeds
.gitignore
public
css
app.css
style.css
img-uploads
js
.htaccess
favicon.ico
index.php
robots.txt
web.config
resources
assets
profile.blade.php
style.css
2017_03_30_090635_create_likes_table.php
2017_03_14_052540_create_table_posts

```php
    ];
});

$factory->define(App\Post::class, function (Faker\Generator $faker) {

    return [
        'user_id' => App\User::all()->random()->id,
        'content' => $faker->paragraph(25)
    ];
});

$factory->define(App\Like::class, function (Faker\Generator $faker) {

    return [
        'user_id' => App\User::all()->random()->id,
        'post_id' =>App\Post::all()->random()->id
    ];
});
```

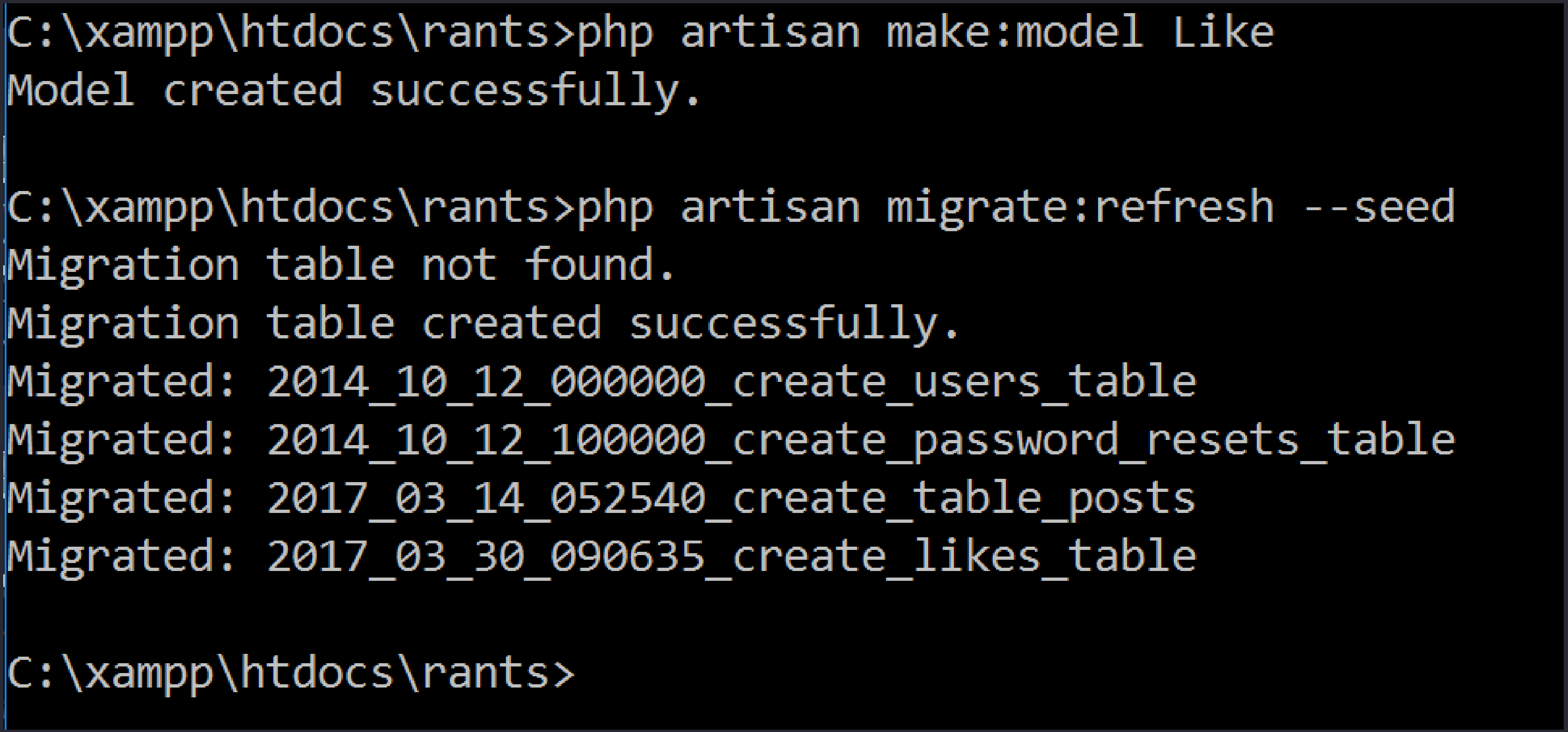
Then seed the data.

Open DatabaseSeeder.php and this line



```php
<?php

use Illuminate\Database\Seeder;

class DatabaseSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        factory(App\User::class, 100)->create();
        factory(App\Post::class, 1000)->create();
        factory(App\Like::class, 1500)->create();
    }
}
```

Migrate again.

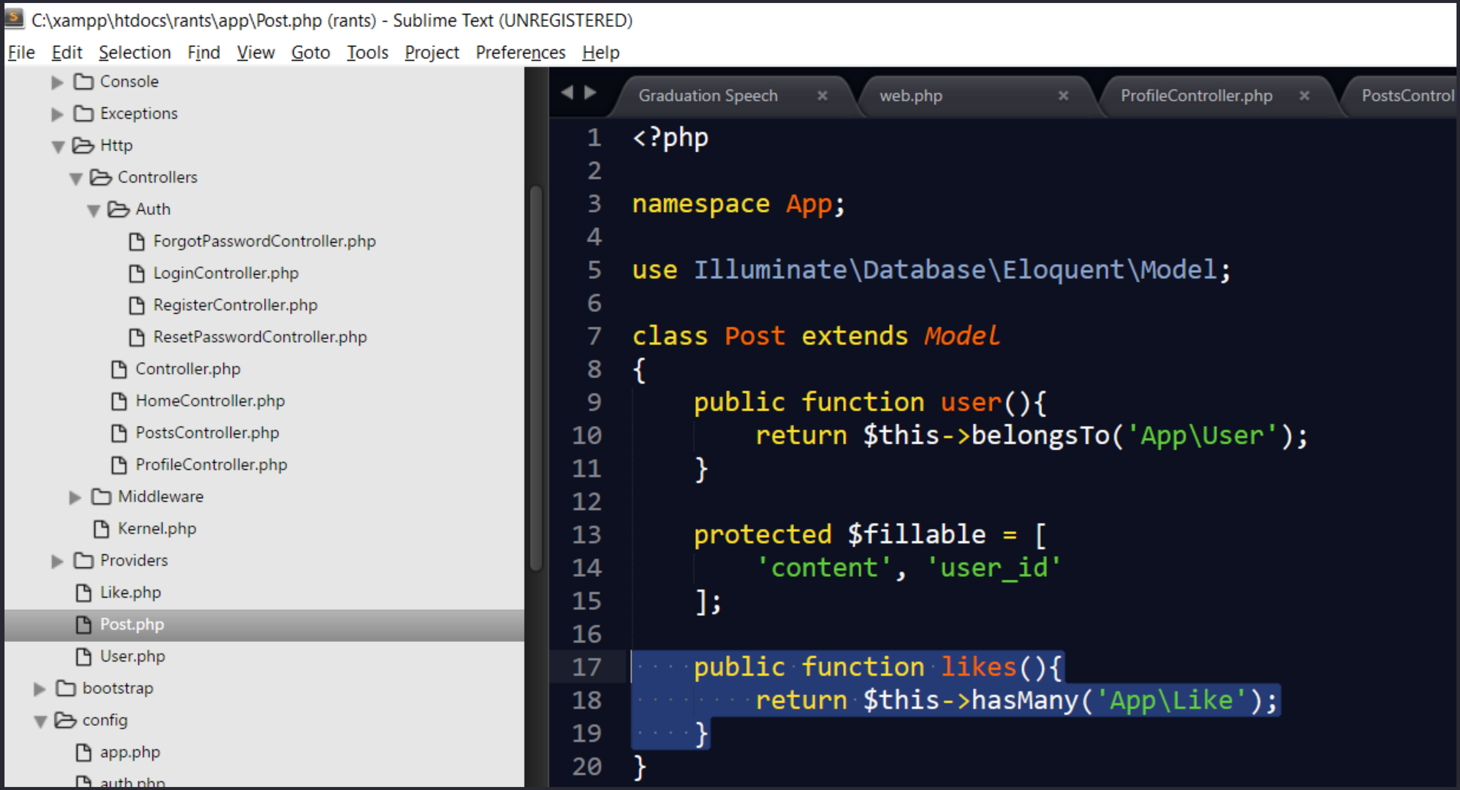


```
C:\xampp\htdocs\rants>php artisan make:model Like
Model created successfully.

C:\xampp\htdocs\rants>php artisan migrate:refresh --seed
Migration table not found.
Migration table created successfully.
Migrated: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_100000_create_password_resets_table
Migrated: 2017_03_14_052540_create_table_posts
Migrated: 2017_03_30_090635_create_likes_table

C:\xampp\htdocs\rants>
```

- If an error is generated, try to drop the tables first.

Reflect the number of likes in each post.

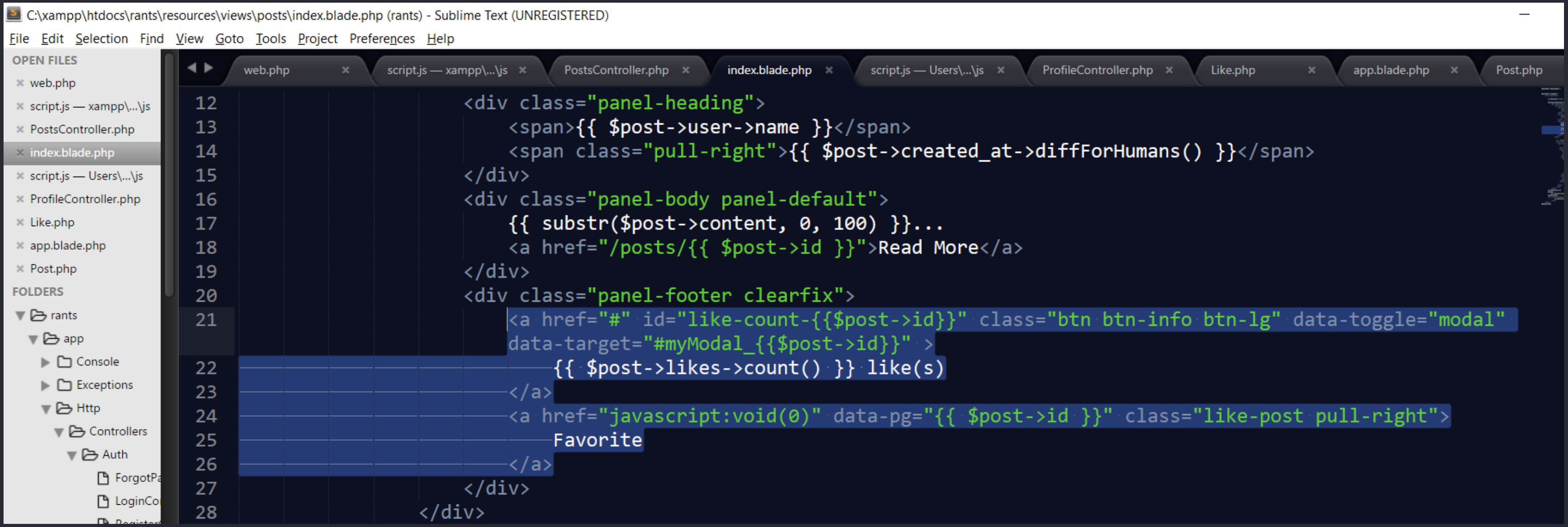
- Connect the table of Posts to Likes. Each post has many likes.
- Edit the model of Post.php



```php
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Post extends Model
{
    public function user(){
        return $this->belongsTo('App\User');
    }

    protected $fillable = [
        'content', 'user_id'
    ];

    public function likes(){
        return $this->hasMany('App\Like');
    }
}
```

Modify index.blade.php

Print the number of likes

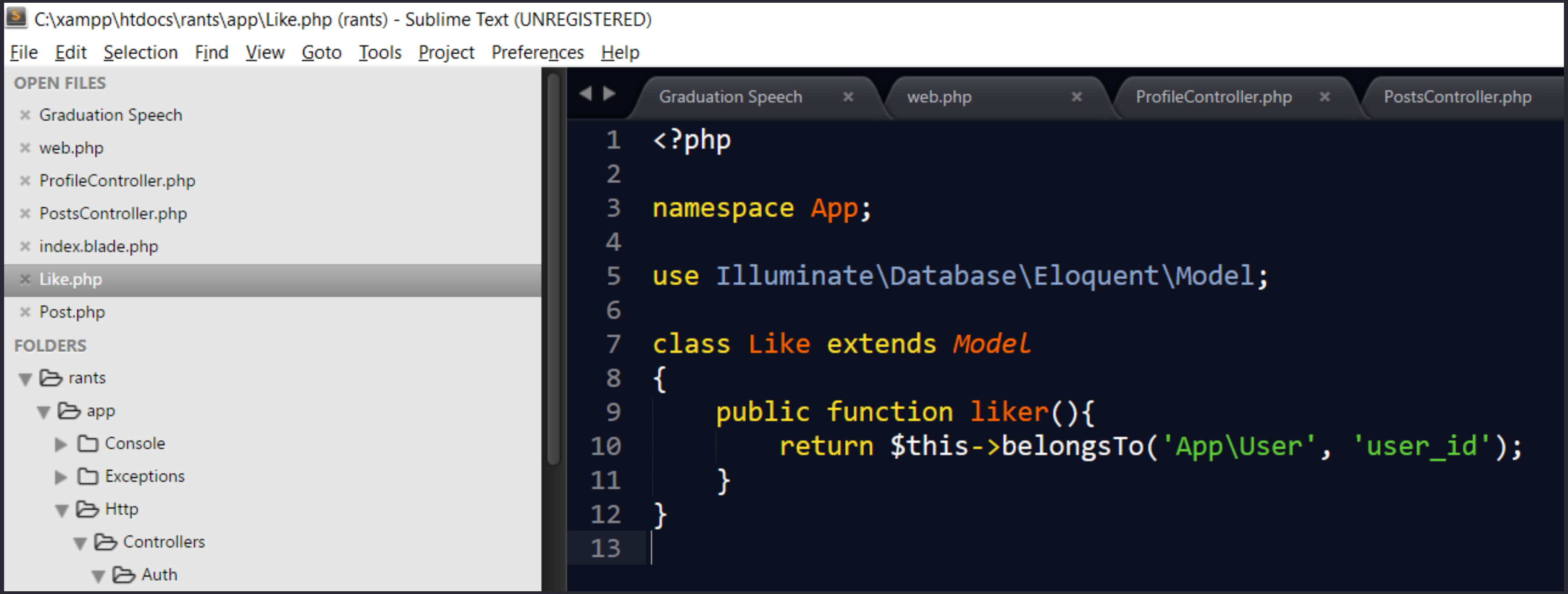


C:\xampp\htdocs\rants\resources\views\posts\index.blade.php (rants) - Sublime Text (UNREGISTERED)

```
            <div class="panel-heading">
                <span>{{ $post->user->name }}</span>
                <span class="pull-right">{{ $post->created_at->diffForHumans() }}</span>
            </div>
            <div class="panel-body panel-default">
                {{ substr($post->content, 0, 100) }}...
                <a href="/posts/{{ $post->id }}">Read More</a>
            </div>
            <div class="panel-footer clearfix">
                <a href="#" id="like-count-{{$post->id}}" class="btn btn-info btn-lg" data-toggle="modal"
                data-target="#myModal_{{$post->id}}" >
                    {{ $post->likes->count() }} like(s)
                </a>
                <a href="javascript:void(0)" data-pg="{{ $post->id }}" class="like-post pull-right">
                    Favorite
                </a>
            </div>
        </div>
```

Set the relationship between users and likes table

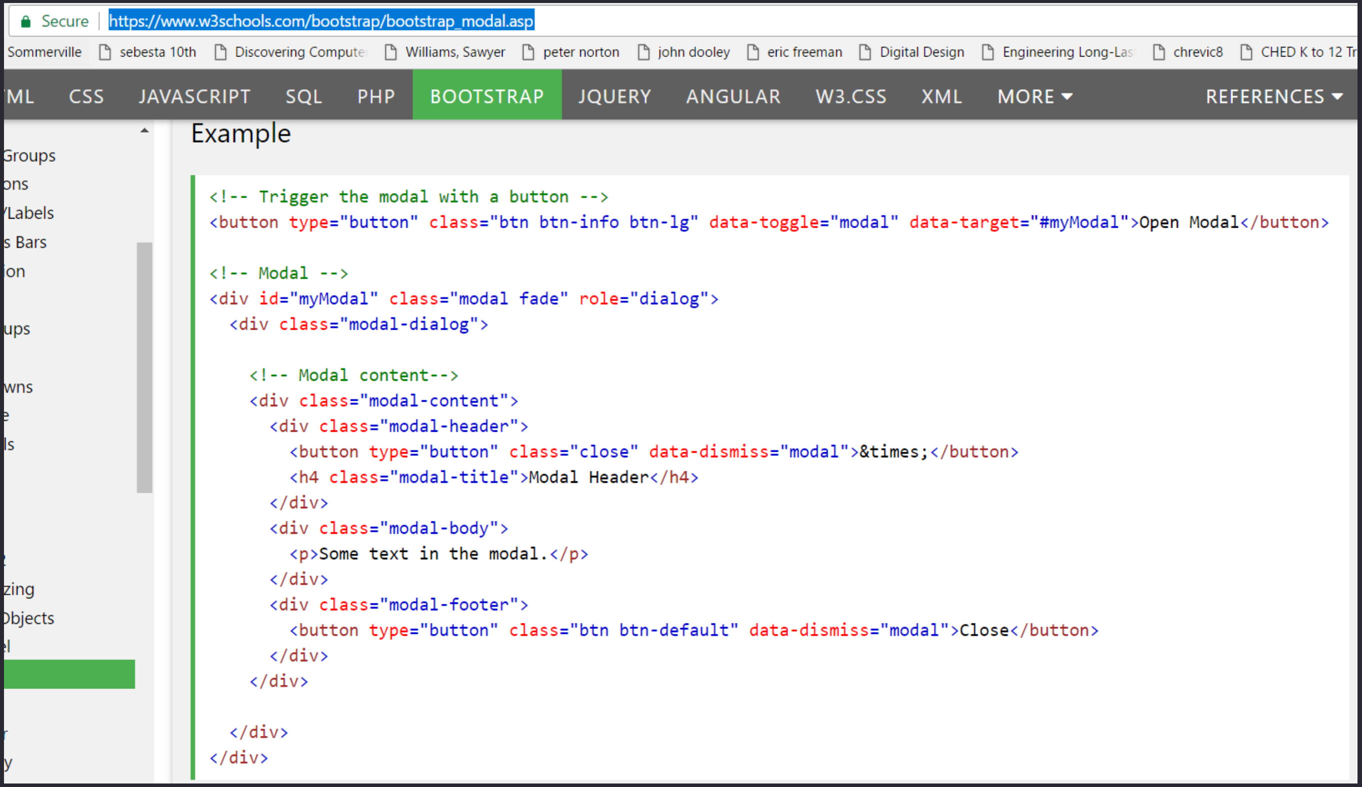
A like belongs to a user.



```php
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Like extends Model
{
    public function liker(){
        return $this->belongsTo('App\User', 'user_id');
    }
}
```

Now, if the count is clicked, a modal pops up with the names of the likers

- I'll just copy modal code from
- https://www.w3schools.com/bootstrap/bootstrap_modal.asp
- And incorporate it in our code



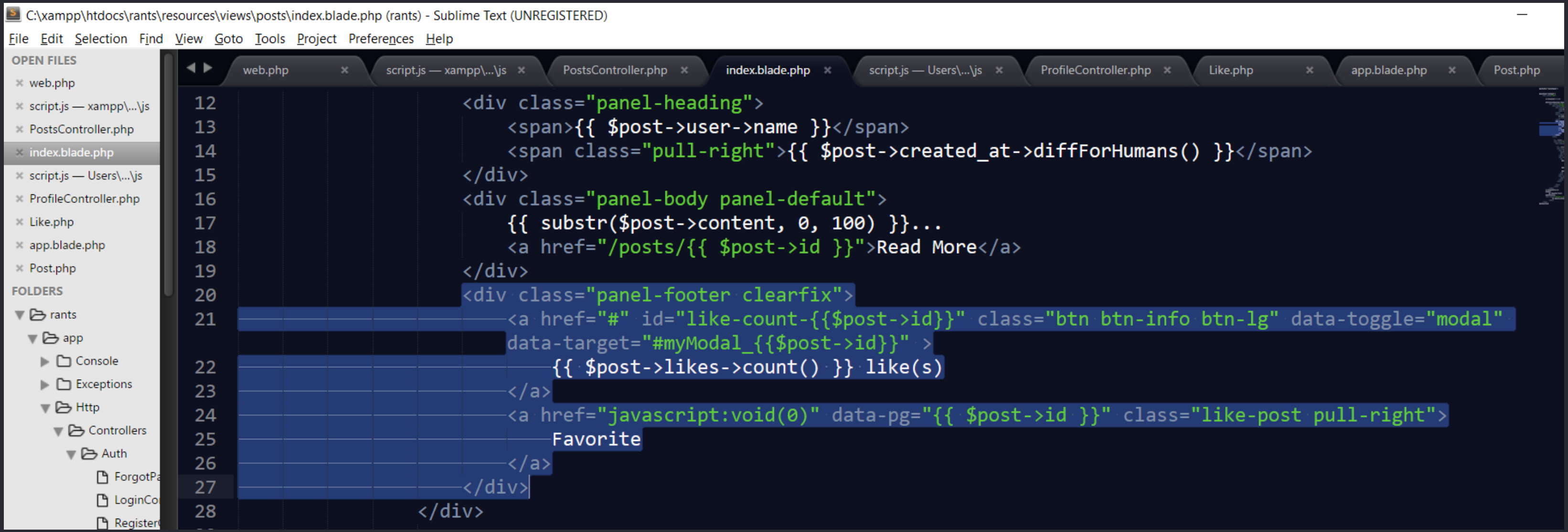
Secure
https://www.w3schools.com/bootstrap/bootstrap_modal.asp
Sommerville | sebesta 10th | Discovering Compute | Williams, Sawyer | peter norton | john dooley | eric freeman | Digital Design | Engineering Long-Las | chrevic8 | CHED K to 12 Tr
ML CSS JAVASCRIPT SQL PHP BOOTSTRAP JQUERY ANGULAR W3.CSS XML MORE REFERENCES
Groups
ons
Labels
s Bars
on
ups
wns
e
ls
zing
Objects
el
y

Example

```
<!-- Trigger the modal with a button -->
<button type="button" class="btn btn-info btn-lg" data-toggle="modal" data-target="#myModal">Open Modal</button>

<!-- Modal -->
<div id="myModal" class="modal fade" role="dialog">
  <div class="modal-dialog">

    <!-- Modal content-->
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal">&times;</button>
        <h4 class="modal-title">Modal Header</h4>
      </div>
      <div class="modal-body">
        <p>Some text in the modal.</p>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-default" data-dismiss="modal">Close</button>
      </div>
    </div>

  </div>
</div>
```

Time to tweak the view :D

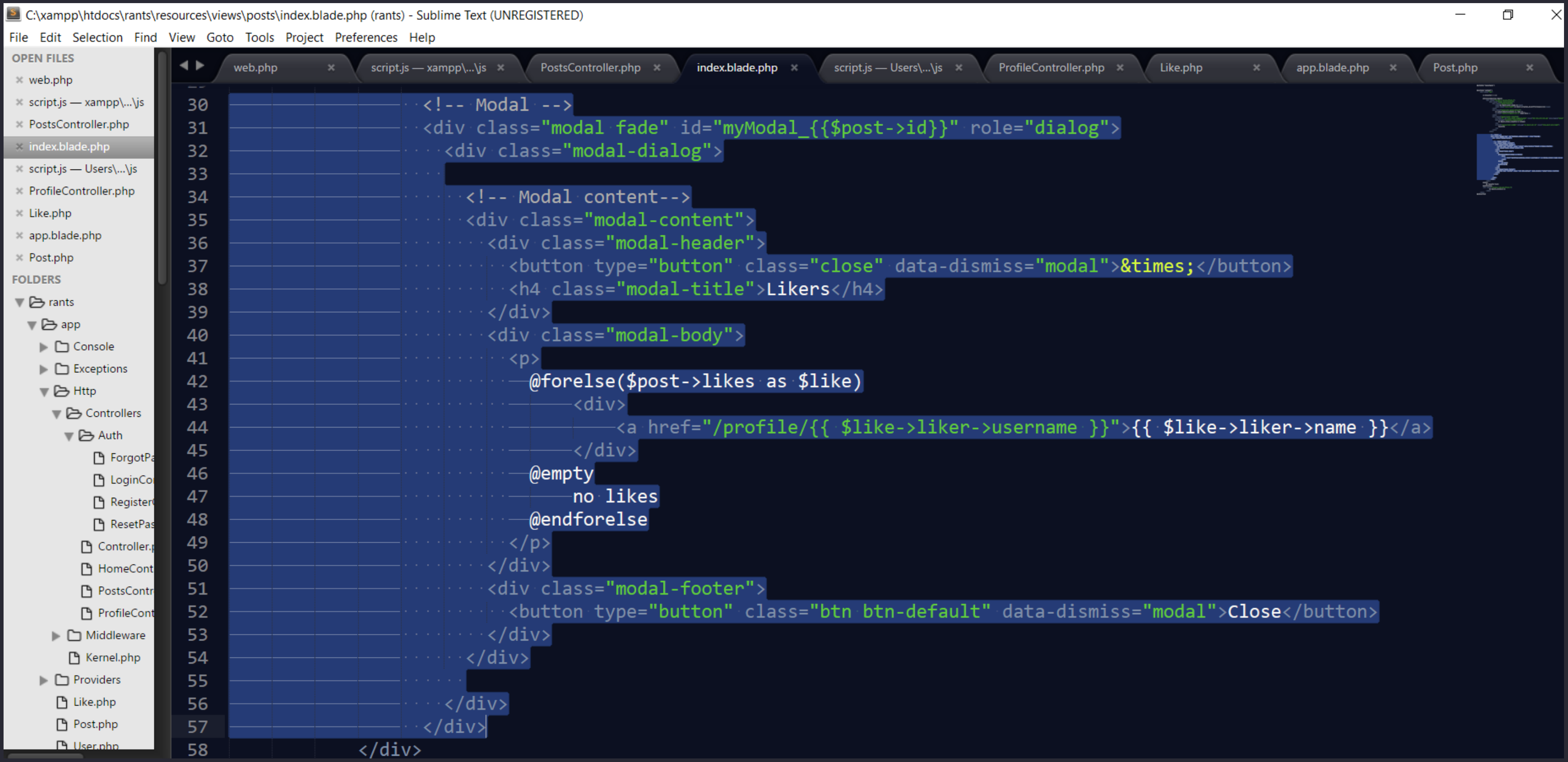
- Add the printing of names of the likers



```
            <div class="panel-heading">
                <span>{{ $post->user->name }}</span>
                <span class="pull-right">{{ $post->created_at->diffForHumans() }}</span>
            </div>
            <div class="panel-body panel-default">
                {{ substr($post->content, 0, 100) }}...
                <a href="/posts/{{ $post->id }}">Read More</a>
            </div>
            <div class="panel-footer clearfix">
                <a href="#" id="like-count-{{$post->id}}" class="btn btn-info btn-lg" data-toggle="modal"
                data-target="#myModal_{{$post->id}}" >
                    {{ $post->likes->count() }} like(s)
                </a>
                <a href="javascript:void(0)" data-pg="{{ $post->id }}" class="like-post pull-right">
                    Favorite
                </a>
            </div>
        </div>
```

Still on the view.. ☹

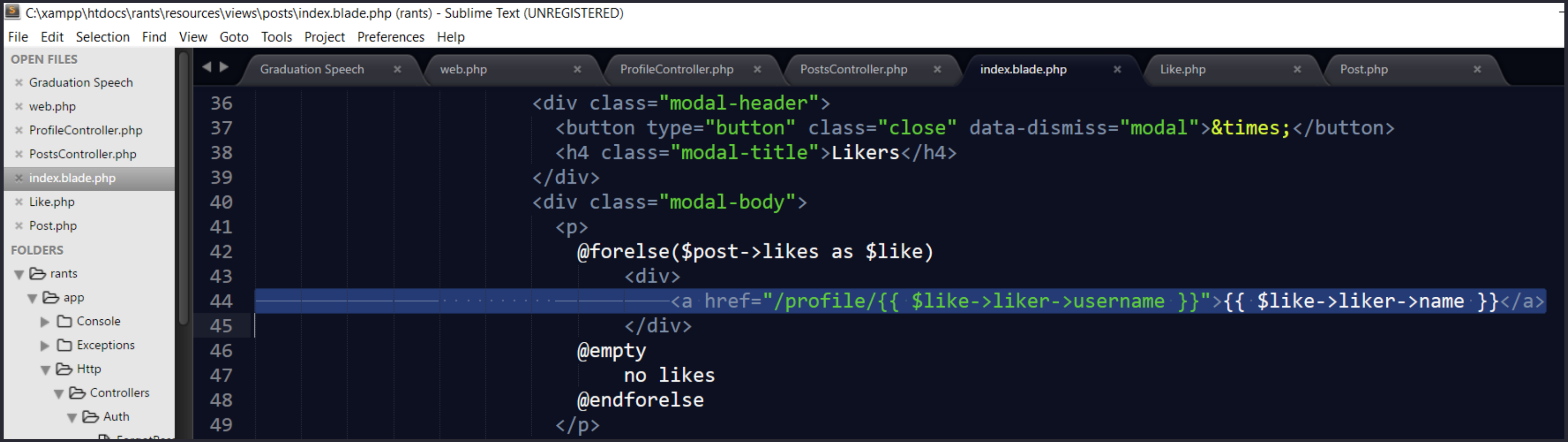
Add the div for the modal



```
   <!-- Modal -->
   <div class="modal fade" id="myModal_{{$post->id}}" role="dialog">
    <div class="modal-dialog">

     <!-- Modal content-->
     <div class="modal-content">
      <div class="modal-header">
       <button type="button" class="close" data-dismiss="modal">&times;</button>
       <h4 class="modal-title">Likers</h4>
      </div>
      <div class="modal-body">
       <p>
        @forelse($post->likes as $like)
          <div>
            <a href="/profile/{{ $like->liker->username }}">{{ $like->liker->name }}</a>
          </div>
        @empty
          no likes
        @endforelse
       </p>
      </div>
      <div class="modal-footer">
       <button type="button" class="btn btn-default" data-dismiss="modal">Close</button>
      </div>
     </div>

    </div>
   </div>
</div>
```

Do a little bit more modification.

Make the names of the likers a link so if you click the name, it redirects you to their profile.



C:\xampp\htdocs\rants\resources\views\posts\index.blade.php (rants) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
OPEN FILES
Graduation Speech
web.php
ProfileController.php
PostsController.php
index.blade.php
Like.php
Post.php
FOLDERS
rants
app
Console
Exceptions
Http
Controllers
Auth

```
<div class="modal-header">
  <button type="button" class="close" data-dismiss="modal">&times;</button>
  <h4 class="modal-title">Likers</h4>
</div>
<div class="modal-body">
  <p>
    @forelse($post->likes as $like)
        <div>
        <a href="/profile/{{ $like->liker->username }}">{{ $like->liker->name }}</a>
        </div>
    @empty
        no likes
    @endforelse
  </p>
```

Ok. We’re done for now! Haha.

Joke. One more task!

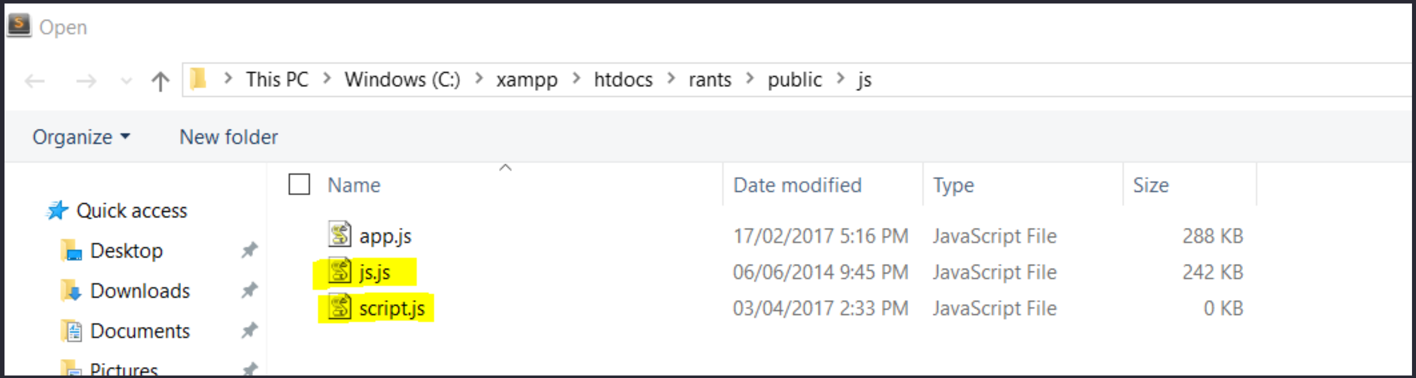
- What we did so far is to initially show the likes that are saved in the database.

- Next should be inserting a row in the likes table if the like button is clicked.

- So how?

We will use ajax. (Wew.)

- Download *jquery* library.
- Save it in the *public/js* folder
- Create a new file in *public/js folder*, filename.js
- I set the filename=script.js

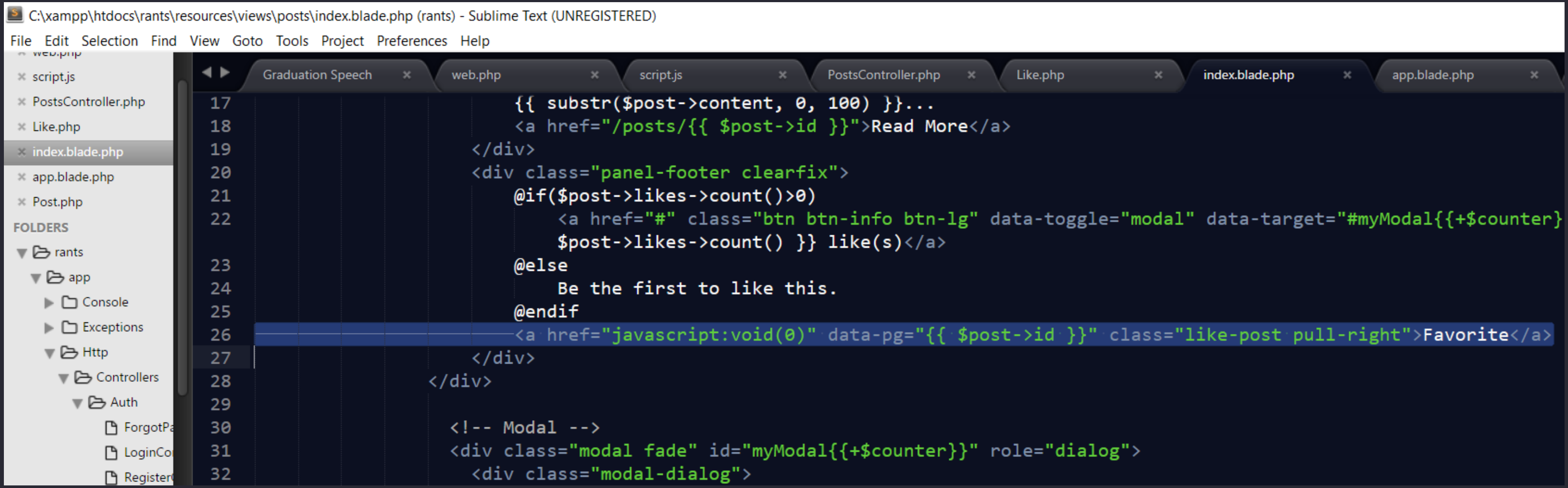

Open

This PC > Windows (C:) > xampp > htdocs > rants > public > js

Name	Date modified	Type	Size
app.js	17/02/2017 5:16 PM	JavaScript File	288 KB
js.js	06/06/2014 9:45 PM	JavaScript File	242 KB
script.js	03/04/2017 2:33 PM	JavaScript File	0 KB

Modify index.blade.php

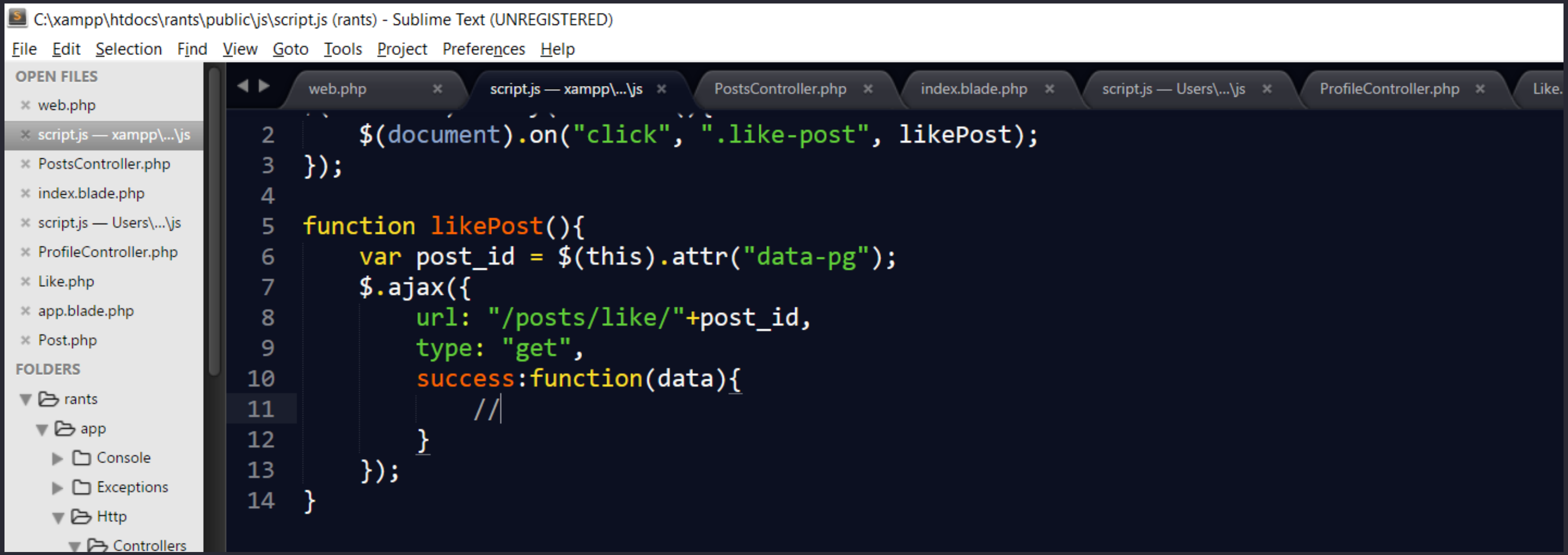
Add class to the like button. In this case, I used *like-post*



```
    {{ substr($post->content, 0, 100) }}...
    <a href="/posts/{{ $post->id }}">Read More</a>
  </div>
  <div class="panel-footer clearfix">
    @if($post->likes->count()>0)
      <a href="#" class="btn btn-info btn-lg" data-toggle="modal" data-target="#myModal{{+$counter}
      $post->likes->count() }} like(s)</a>
    @else
      Be the first to like this.
    @endif
    <a href="javascript:void(0)" data-pg="{{ $post->id }}" class="like-post pull-right">Favorite</a>
  </div>
</div>

 <!-- Modal -->
 <div class="modal fade" id="myModal{{+$counter}}" role="dialog">
  <div class="modal-dialog">
```

Now, the script!

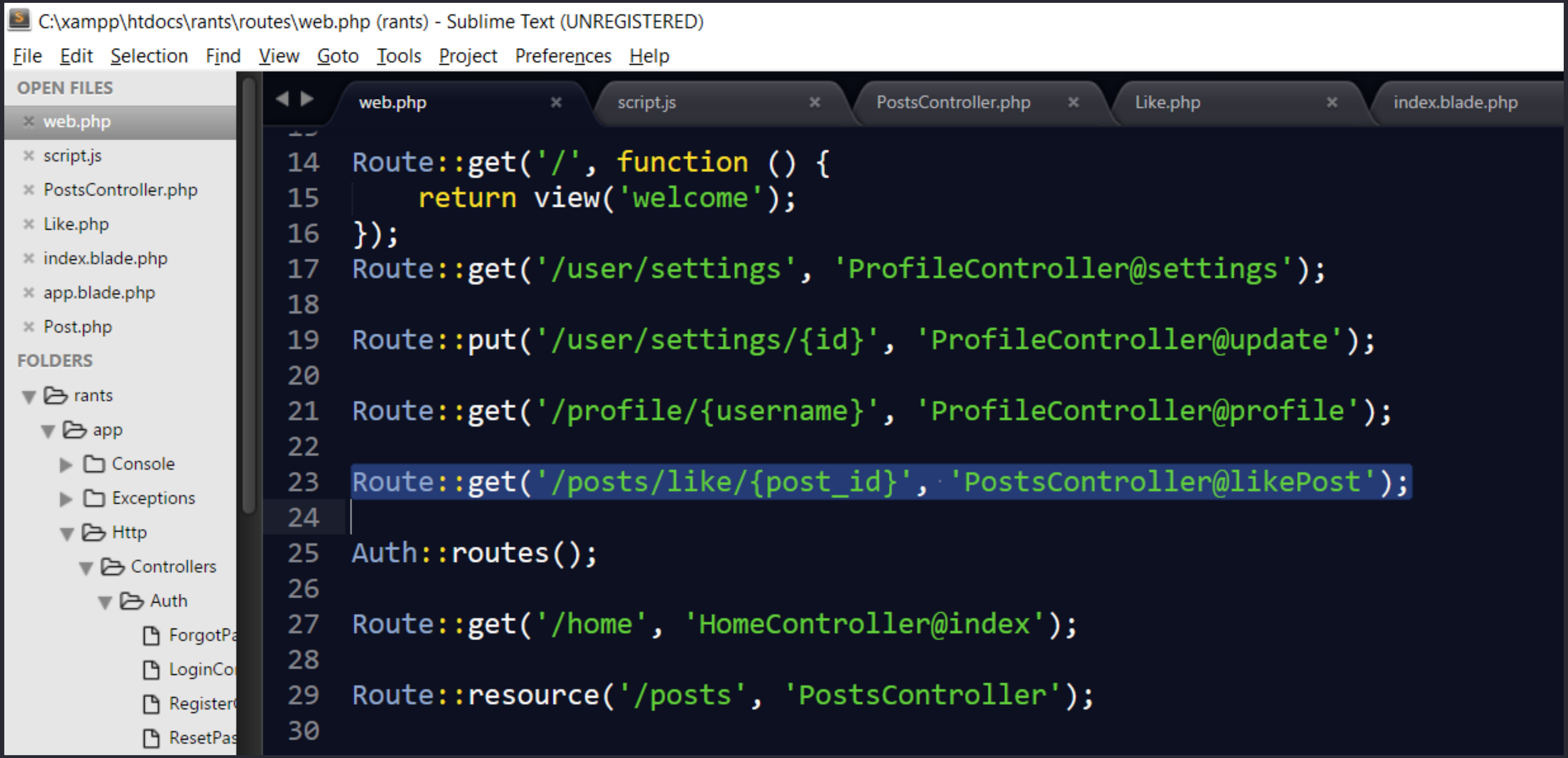
- Edit script.js
- Add a function *likePost* triggered when *like-post* is clicked.



```
    $(document).on("click", ".like-post", likePost);
});

function likePost(){
    var post_id = $(this).attr("data-pg");
    $.ajax({
        url: "/posts/like/"+post_id,
        type: "get",
        success:function(data){
            //
        }
    });
}
```

Oops, error! Why?

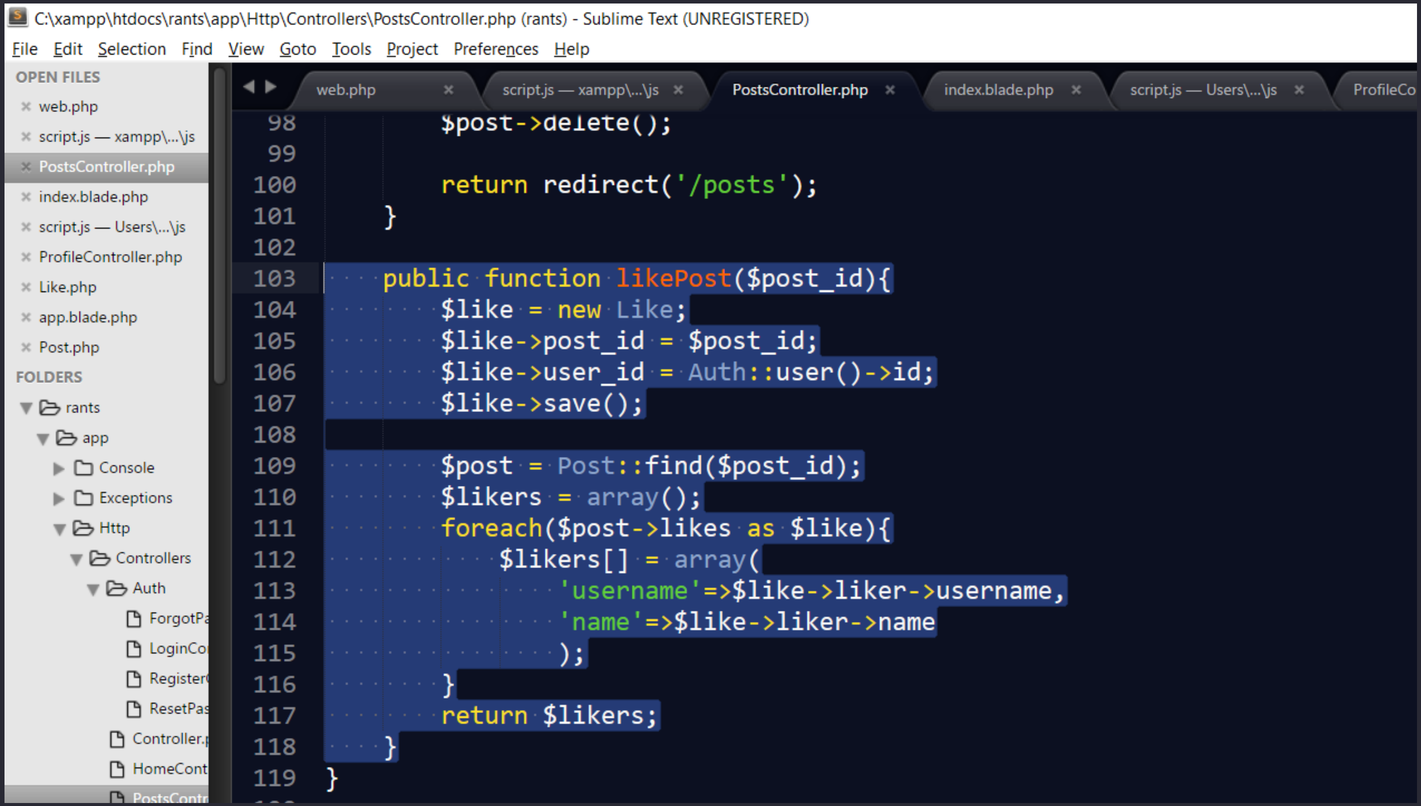
- We have to add a new route.



```php
Route::get('/', function () {
    return view('welcome');
});
Route::get('/user/settings', 'ProfileController@settings');

Route::put('/user/settings/{id}', 'ProfileController@update');

Route::get('/profile/{username}', 'ProfileController@profile');

Route::get('/posts/like/{post_id}', 'PostsController@likePost');


Auth::routes();

Route::get('/home', 'HomeController@index');

Route::resource('/posts', 'PostsController');

```

Another error! Why?

- No function *likePost* exists in *PostController*.
- Create function *likePost*.



```
        $post->delete();

        return redirect('/posts');
    }

    public function likePost($post_id){
        $like = new Like;
        $like->post_id = $post_id;
        $like->user_id = Auth::user()->id;
        $like->save();

        $post = Post::find($post_id);
        $likers = array();
        foreach($post->likes as $like){
            $likers[] = array(
                'username'=>$like->liker->username,
                'name'=>$like->liker->name
                );
        }
        return $likers;
    }
}
```

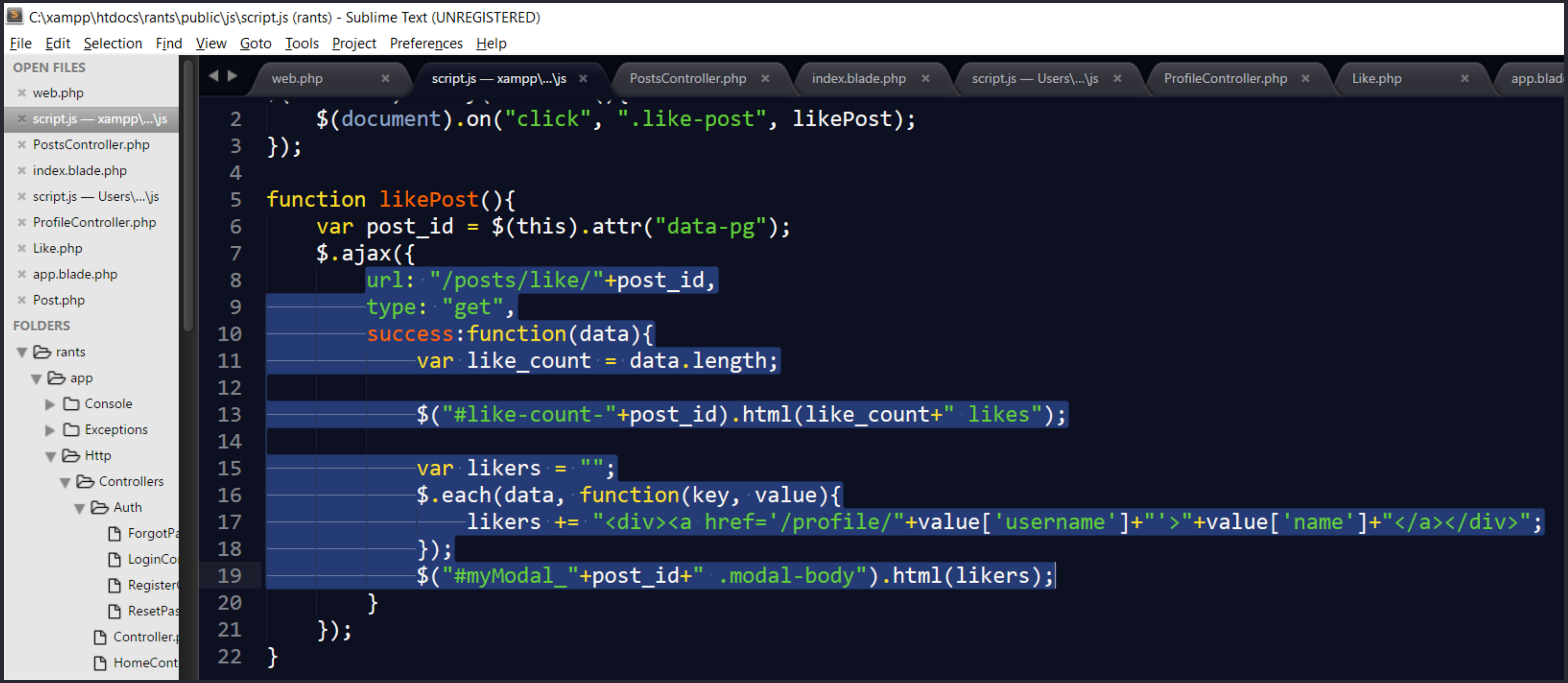
Note!

- We have returned the likers so the values inside the modal will also be changed
- Remember that 1 should be added to the count of likes.
- Also, the names will be updated with the name of the new liker

- Next, edit the *likePost* function in *script.js*

Print the new count of likes.

- Also, update the names.



```
    $(document).on("click", ".like-post", likePost);
});

function likePost(){
    var post_id = $(this).attr("data-pg");
    $.ajax({
        url: "/posts/like/"+post_id,
        type: "get",
        success:function(data){
            var like_count = data.length;

            $("#like-count-"+post_id).html(like_count+" likes");

            var likers = "";
            $.each(data, function(key, value){
                likers += "<div><a href='/profile/"+value['username']+"'>"+value['name']+"</a></div>";
            });
            $("#myModal_"+post_id+" .modal-body").html(likers);
        }
    });
}
```

Next: Edit the profile.blade.php

So…

- I JUST VIOLATED SRP.
- Don’t follow what I did.
- Optimize the code.
- Such that repetitions are lessened.

Thank you for listening!

- We still have a lot to do though.
- Restrictions for likes (only one like per post), middleware, etc. ☹

Next Lab Activity

- Implement the follows functionality.

- Breakdown of tasks:
  - Create a follows table in the database
  - Add a model for follows table.
  - You can use the user controller for your functions.
  - Add a restriction that if you have already followed a user, you can no longer follow them again.
  - Add a link/button in your profile. One for your followers. Another link for those you follow. (Like in twitter)
  - Once the link is clicked, it will show you the clickable names of the people you follow and follow you.