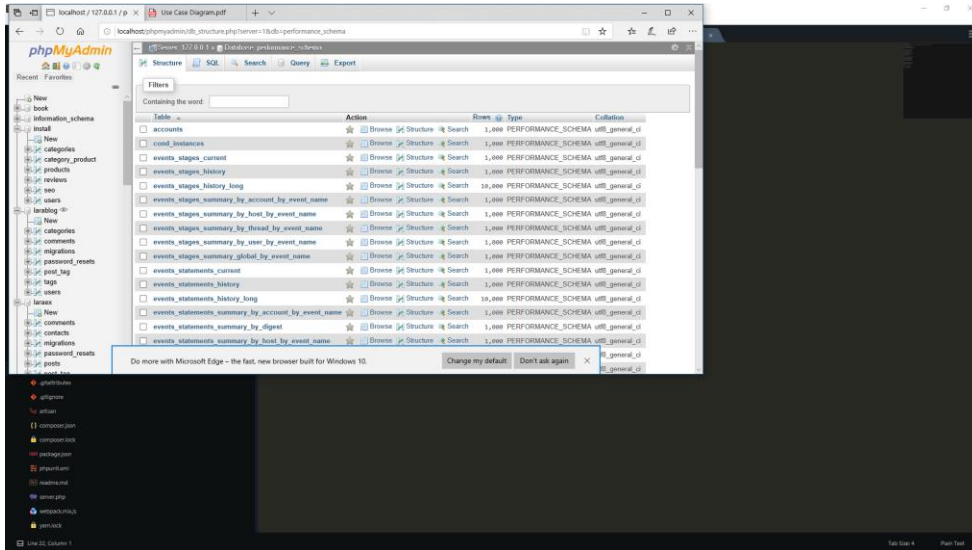
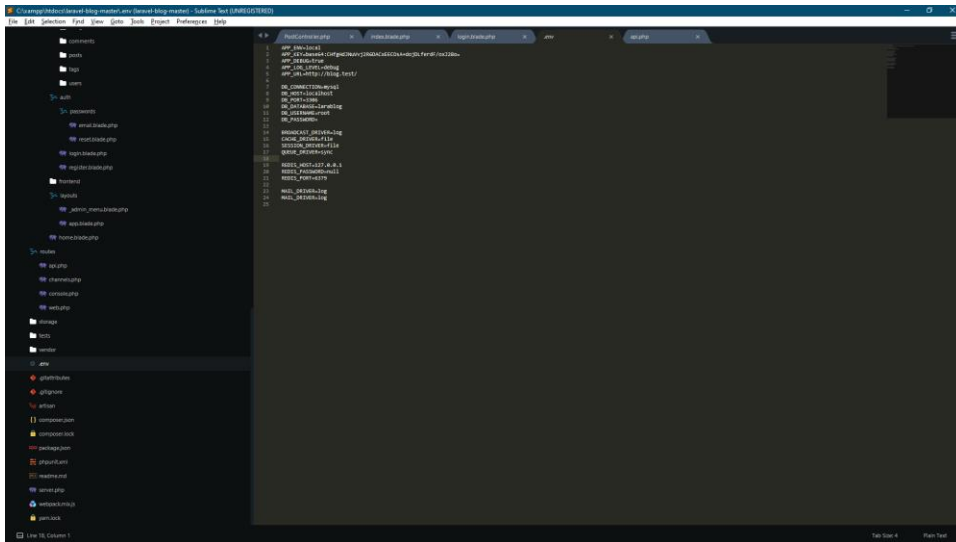


Laravel Blog

- Create empty db in your sqlserver



- cd Project name
- composer install
- cp .env.example .env (you can rename with out command!!)



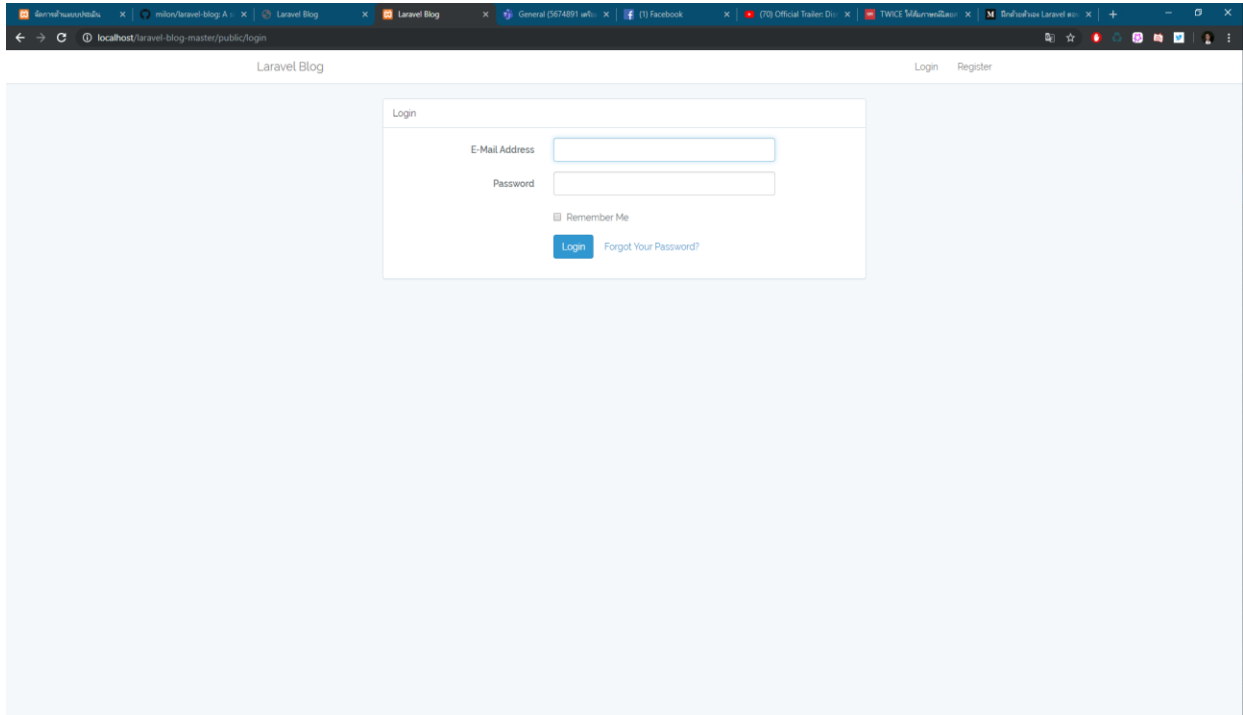
rename DB_DATABASE as “your created empty db name” If your DB have username and password or host name pls custom your DB_USERNAME, DB_PASSWORD, DB_HOST

- php artisan key:generate
- php artisan migrate
- php artisan db:seed

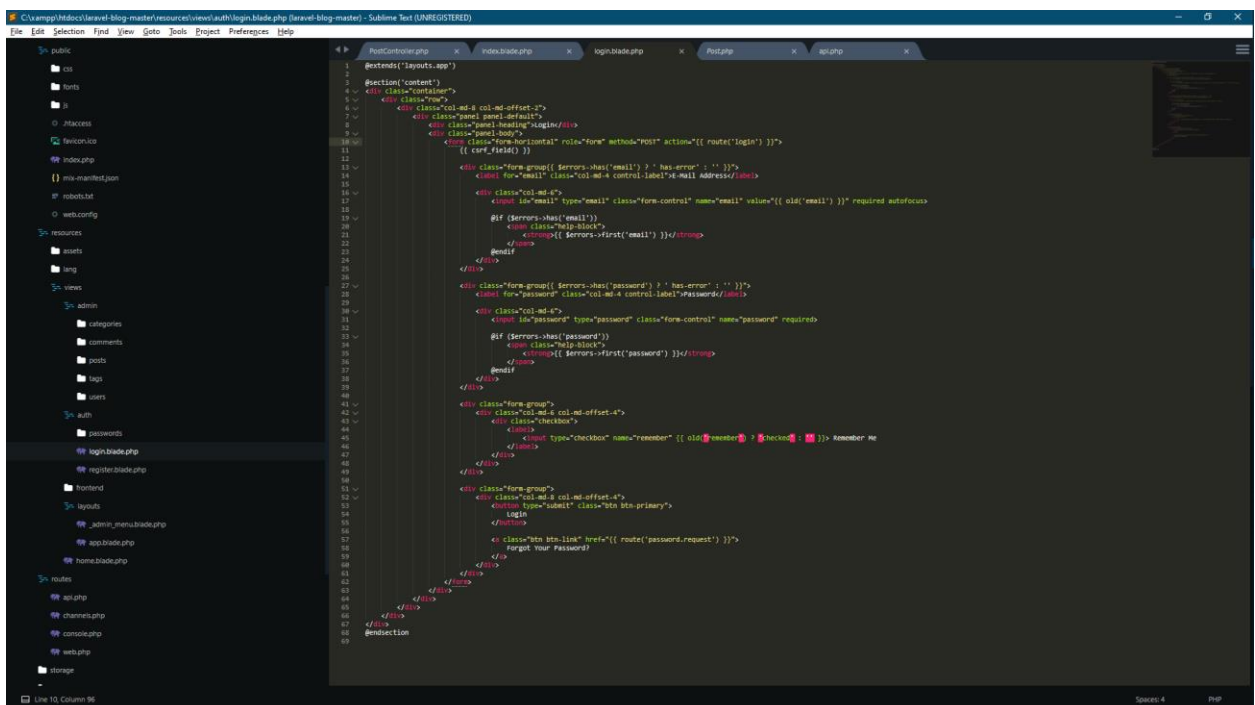
Login & Register

<https://www.youtube.com/watch?v=VoGL0tit1do&t=38s>

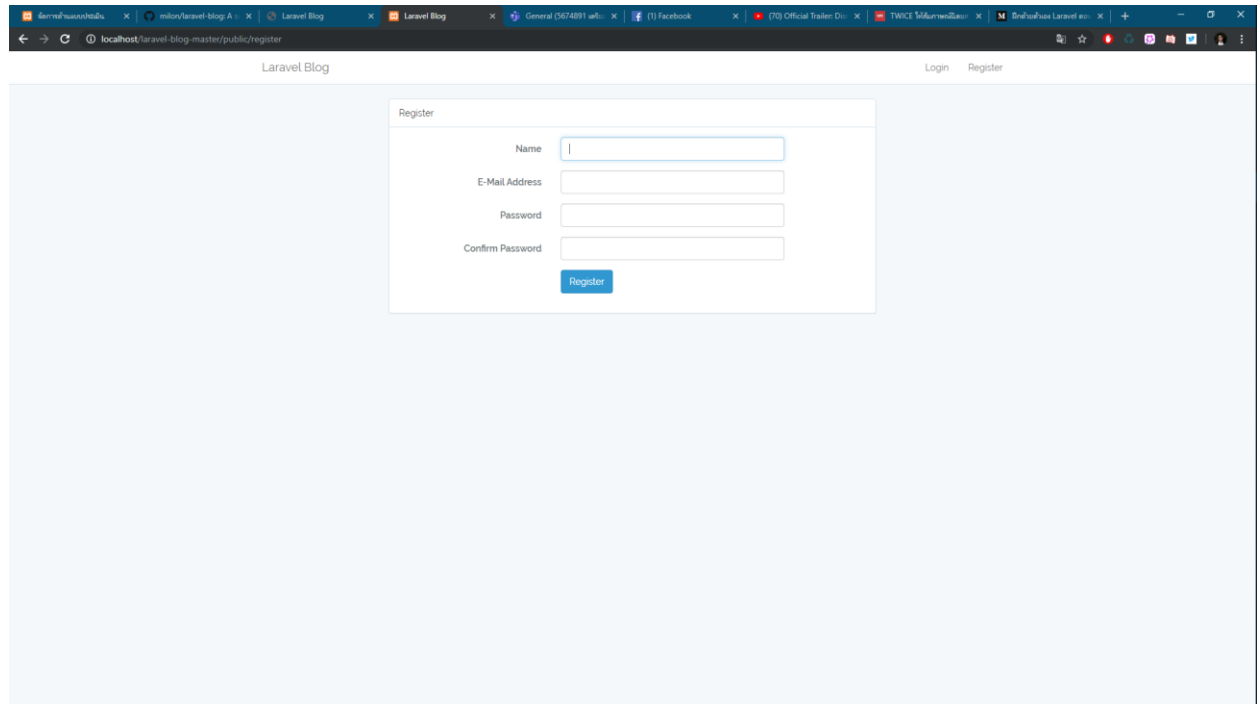
Form Login



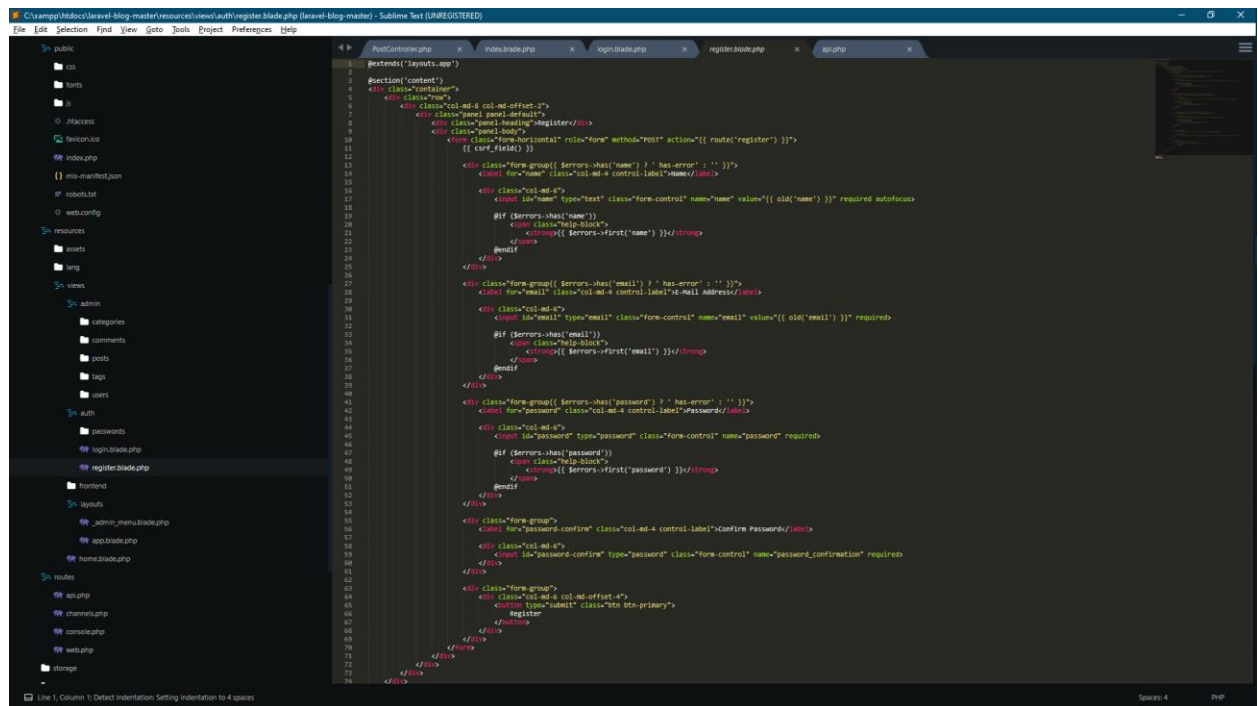
Login.blade.php



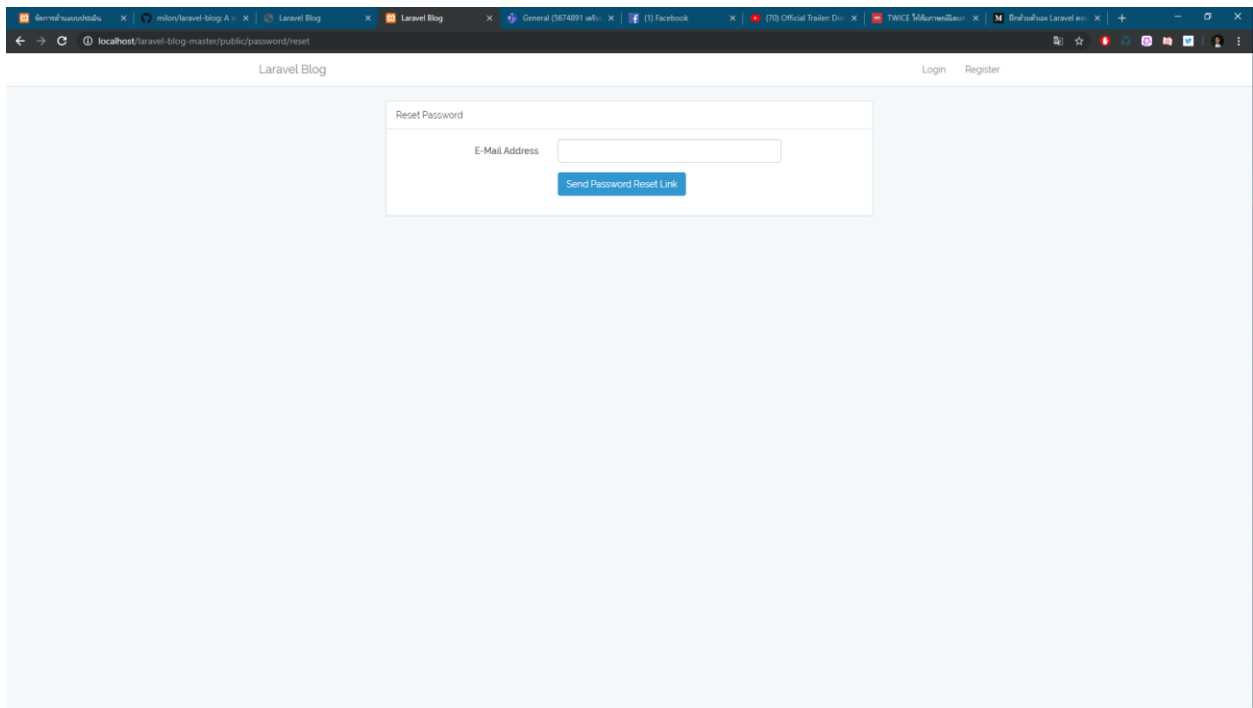
Form Register



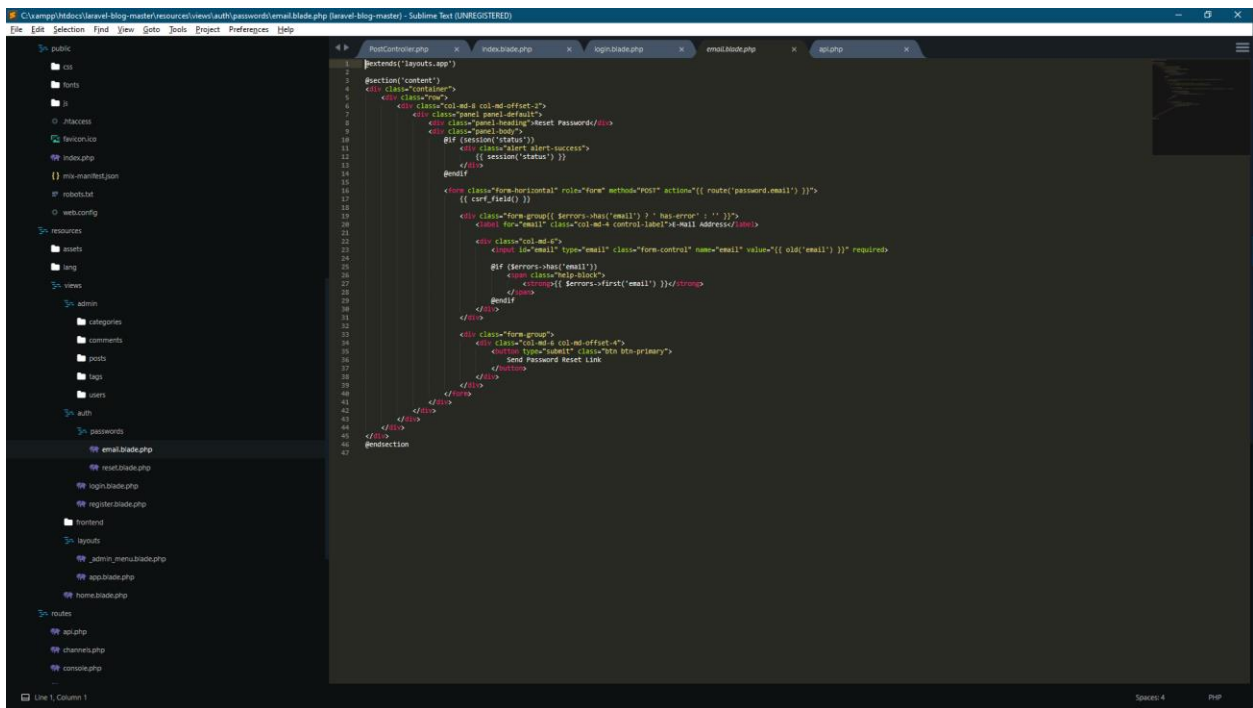
Register.blade.php



Form forget password



Email.blade.php



Index.blade.php

[illegible]

_search.blade.php

The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a directory structure with folders like assets, lang, views, and routes. The code editor shows a PHP file named _search.blade.php with a form search structure.

File Explorer Structure:

- resources
 - assets
 - lang
 - views
 - admin
 - categories
 - comments
 - posts
 - tags
 - users
 - index.blade.php
 - auth
 - passwords
 - login.blade.php
 - register.blade.php
 - frontend
 - comments.blade.php
 - form.blade.php
 - index.blade.php
 - post.blade.php
 - layouts
 - admin_menu.blade.php
 - app.blade.php
 - home.blade.php
 - routes
 - api.php
 - channels.php
 - console.php
 - web.php
 - storage
 - tests
 - vendor

Code Editor Content (resources/views/frontend/_search.blade.php):

```

1 <div class="row form-search">
2     <!-- Form: open ([method] => 'GET', 'role' => 'form') -->
3     <div class="col-md-12">
4         <!-- Form: text ('search', request()->get('search'), ['class' => 'form-control', 'placeholder' => 'Search...']) -->
5         </div>
6         <div class="col-md-2">
7             <!-- Form: submit ('Submit', ['class' => 'btn btn-block btn-default']) -->
8             </div>
9         <!-- Form: close() -->
10    </div>
11

```

app.blade.php

```
C:\xampp\htdocs\laravel-blog-master\resources\views\layouts\app.blade.php (laravel-blog-master) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

x web.php x app.blade.php

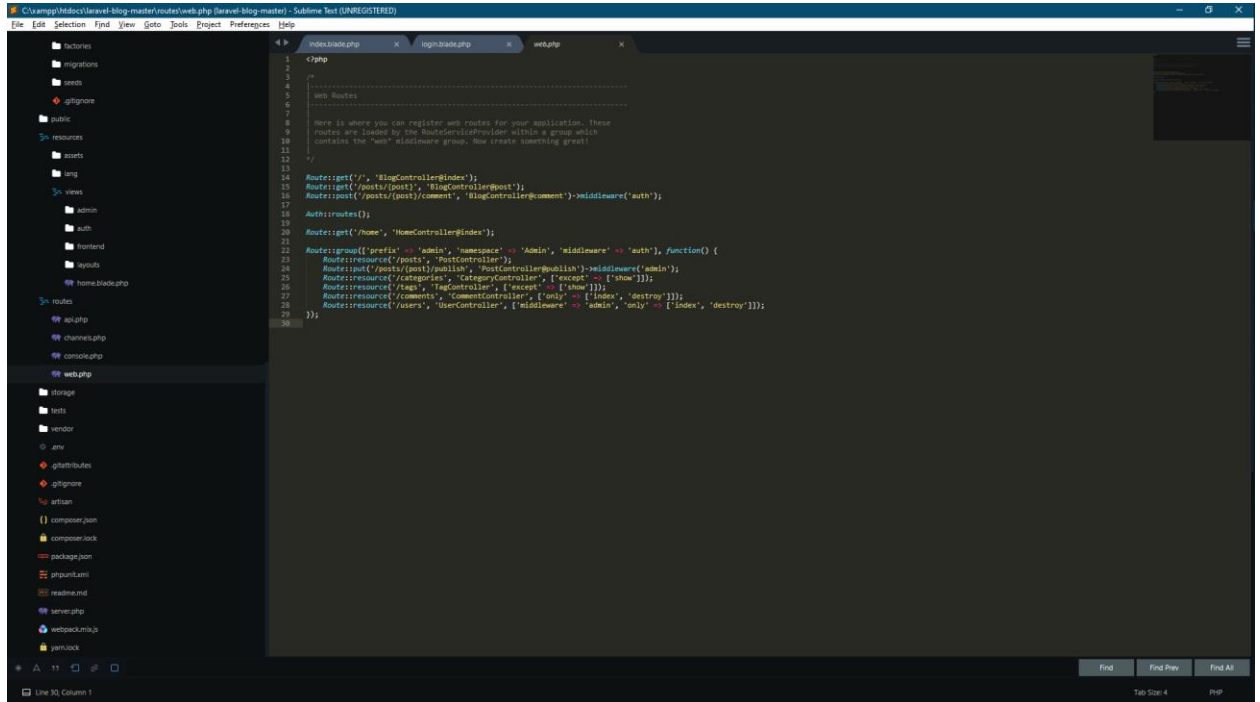
1 <!DOCTYPE html>
2 <html lang="{{ config('app.locale') }}">
3 <head>
4   <meta charset="utf-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7
8   <!-- CSRF Token -->
9   <meta name="csrf-token" content="{{ csrf_token() }}">
10
11   <title>{{ config('app.name', 'Laravel') }}</title>
12
13   <!-- Styles -->
14   <link href="{{ asset('css/app.css') }}" rel="stylesheet">
15
16   <!-- Scripts -->
17   <script>
18     window.Laravel = {!! json_encode([
19       'csrfToken' => csrf_token(),
20     ]) !!};
21   </script>
22 </head>
23 <body>
24   <div id="app">
25     <nav class="navbar navbar-default navbar-static-top">
26       <div class="container">
27         <div class="navbar-header">
28           <!-- Collapsed Hamburger -->
29           <button types="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#app-navbar-collapse">
30             <span class="sr-only">Toggle Navigation</span>
31             <span class="icon-bar"></span>
32             <span class="icon-bar"></span>
33             <span class="icon-bar"></span>
34           </button>
35
36           <!-- Branding Image -->
37           <a class="navbar-brand" href="{{ url('/') }}">
38             {{ config('app.name', 'Laravel') }}
39           </a>
40         </div>
41
42         <div class="collapse navbar-collapse" id="app-navbar-collapse">
43           <!-- Left Side Of Navbar -->
44
45           @includeWhen(Auth::user(), 'layouts._admin_menu')
46         </div>
47       </div>
48     </nav>
49
50     <div class="container">
51       <div class="row">
52         <div class="col-md-12">
53           @yield('content')
54         </div>
55       </div>
56     </div>
57
58     <!-- Scripts -->
59     <script src="{{ asset('js/app.js') }}"></script>
60   </body>
61 </html>
```

```
C:\xampp\htdocs\laravel-blog-master\resources\views\layouts\app.blade.php (laravel-blog-master) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

x web.php x app.blade.php x

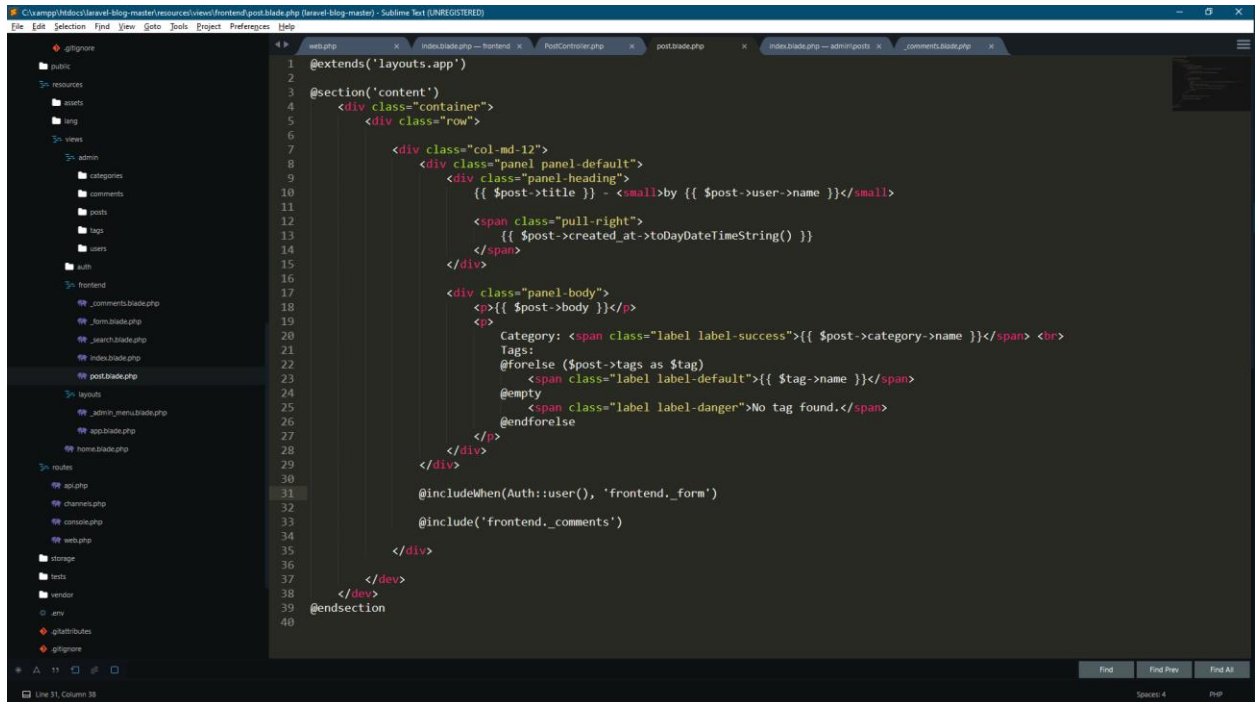
43 <div class="collapse navbar-collapse" id="app-navbar-collapse">
44   <!-- Left Side Of Navbar -->
45
46   @includeWhen(Auth::user(), 'layouts._admin_menu')
47
48   <!-- Right Side Of Navbar -->
49   <div class="nav navbar-nav navbar-right">
50     <!-- Authentication Links -->
51     @if (Auth::guest())
52       <a href="{{ route('login') }}">Login</a>
53       <a href="{{ route('register') }}">Register</a>
54     @else
55       <a href="{{ route('logout') }}"
56         class="dropdown-toggle" data-toggle="dropdown" role="button" aria-expanded="false"
57         >{{ Auth::user()->name }} <span class="caret"></span>
58       </a>
59       <div class="dropdown-menu" role="menu">
60         <a href="{{ route('logout') }}"
61           onclick="event.preventDefault();
62             document.getElementById('logout-form').submit();"
63         >Logout
64         </a>
65         <form id="logout-form" action="{{ route('logout') }}" method="POST" style="display: none;"
66         >{{ csrf_field() }}
67         </form>
68       </div>
69     @endif
70   </div>
71 </div>
72
73 <div class="container">
74   <div class="row">
75     <div class="col-md-12">
76       @include('flash::message')
77     </div>
78   </div>
79   @yield('content')
80 </div>
81
82 <!-- Scripts -->
83 <script src="{{ asset('js/app.js') }}"></script>
84 </body>
85 </html>
```

web.php



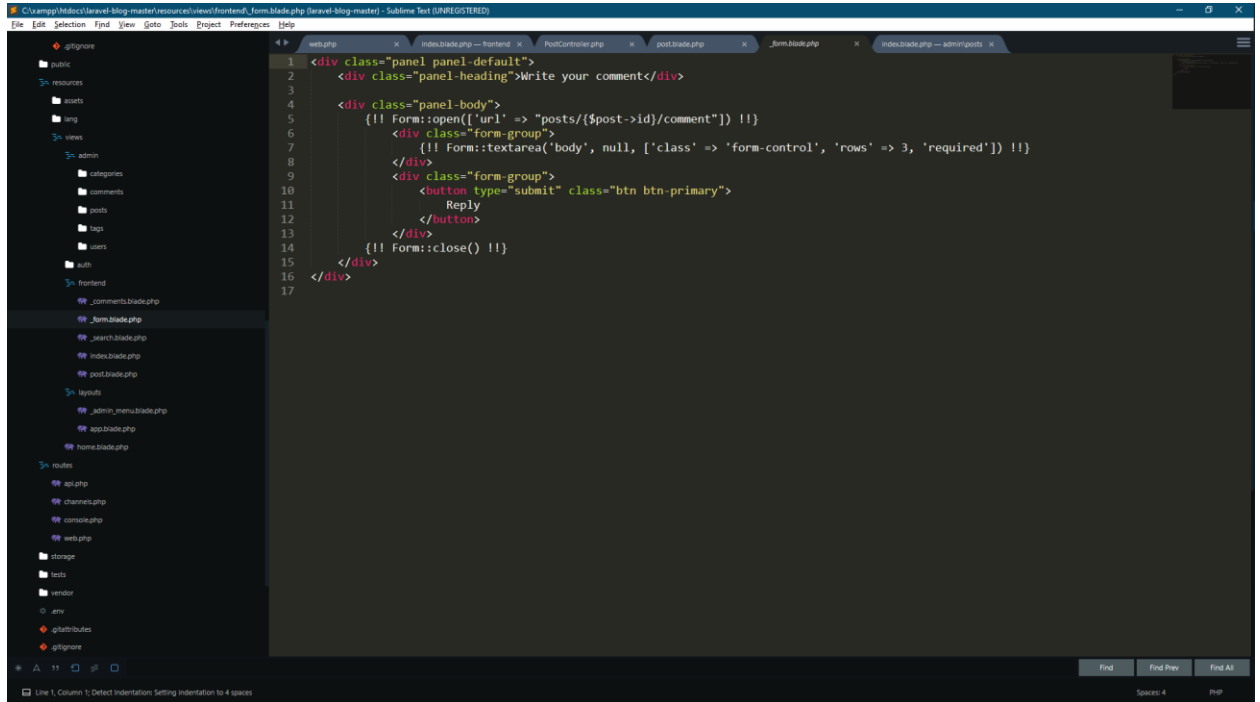
```
1 <?php
2
3 /*
4  * Here you can register the application routes. These
5  * routes are loaded by the RouteServiceProvider within a group which
6  * contains the "web" middleware group. Now create something great!
7  */
8
9
10
11
12
13
14 Route::get('/', 'BlogController@index');
15 Route::get('/posts/{post}', 'BlogController@post');
16 Route::post('/posts/{post}/comment', 'BlogController@comment')->middleware('auth');
17
18 Auth::routes();
19
20 Route::get('/home', 'HomeController@index');
21
22 Route::group(['prefix' => 'admin', 'middleware' => 'Admin', 'middleware' => 'auth'], function() {
23     Route::resource('posts', 'PostController');
24     Route::post('/posts/{post}/publish', 'PostController@publish')->middleware('admin');
25     Route::resource('categories', 'CategoryController', ['except' => ['show']]);
26     Route::resource('tags', 'TagController', ['except' => ['show']]);
27     Route::resource('comments', 'CommentController', ['only' => ['index', 'destroy']]);
28     Route::resource('users', 'UserController', ['middleware' => 'admin', 'only' => ['index', 'destroy']]);
29 });
```

Post.blade.php



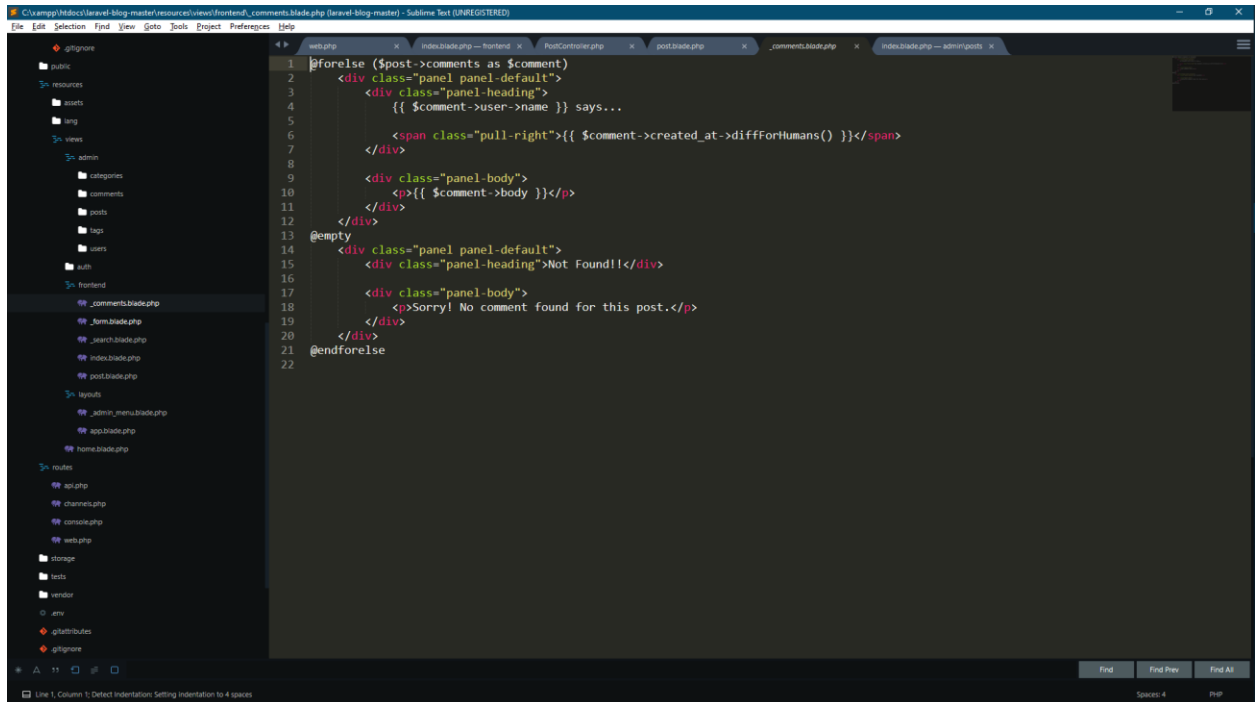
```
1 @extends('layouts.app')
2
3 @section('content')
4     <div class="container">
5         <div class="row">
6
7             <div class="col-md-12">
8                 <div class="panel panel-default">
9                     <div class="panel-heading">
10                         {{ $post->title }} - <small>by {{ $post->user->name }}</small>
11                     </div>
12                     <div class="panel-body">
13                         <div class="pull-right">
14                             {{ $post->created_at->toDayDateTimeString() }}
15                         </div>
16                         <div class="panel-body">
17                             <div class="panel-body">
18                                 <div class="panel-body">
19                                     <div class="panel-body">
20                                         Category: <span class="label label-success">{{ $post->category->name }}</span> <br>
21                                         Tags:
22                                         @forelse ($post->tags as $tag)
23                                             <span class="label label-default">{{ $tag->name }}</span>
24                                         @empty
25                                             <span class="label label-danger">No tag found.</span>
26                                         @endforelse
27                                     </div>
28                                 </div>
29                             </div>
30                         </div>
31                     </div>
32                 </div>
33             </div>
34         </div>
35     @endsection
36
37 @includeWhen(Auth::user(), 'frontend._form')
38
39 @include('frontend._comments')
```

_form.blade.php



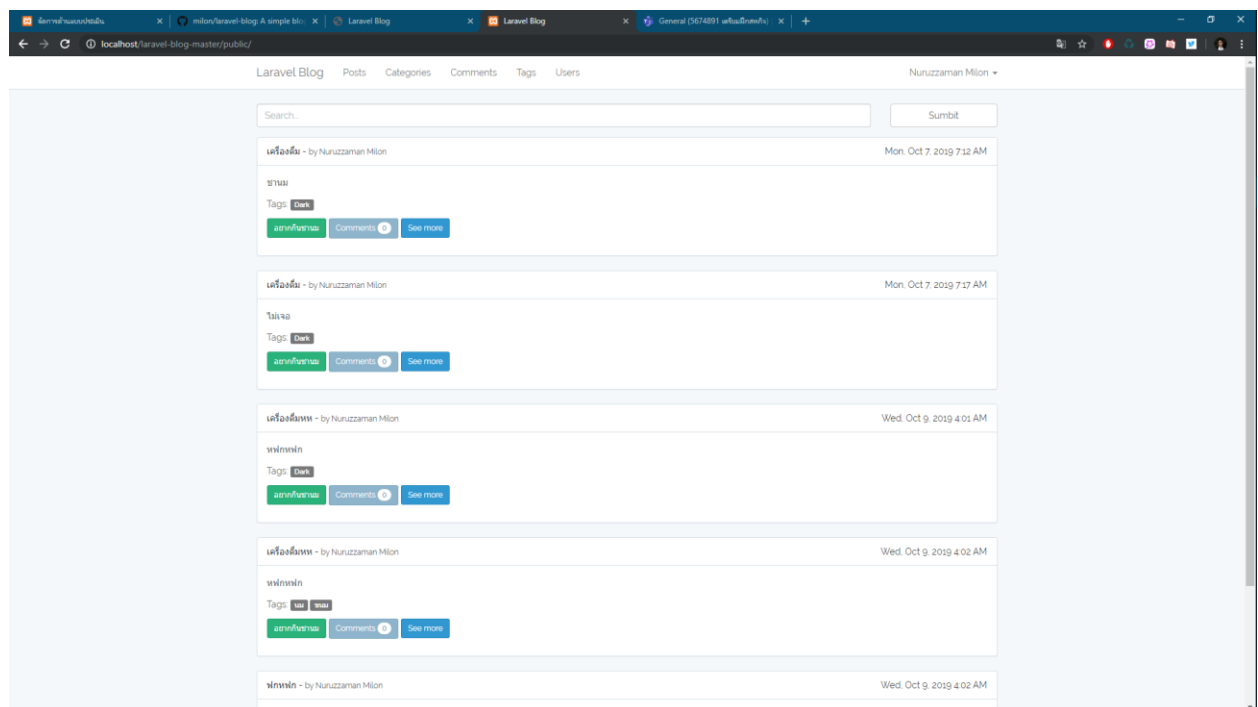
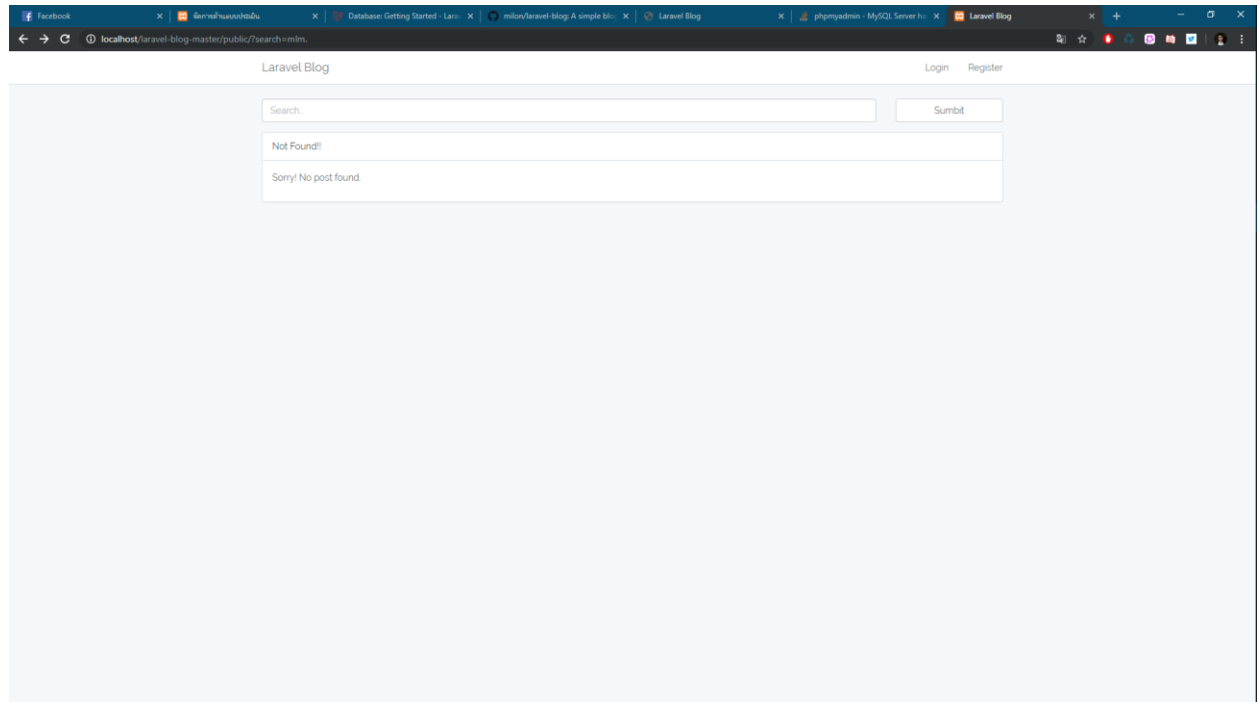
```
1 <div class="panel panel-default">
2   <div class="panel-heading">Write your comment</div>
3
4   <div class="panel-body">
5     {!! Form::open(['url' => "posts/{$post->id}/comment"]) !!}
6     <div class="form-group">
7       {!! Form::textarea('body', null, ['class' => 'form-control', 'rows' => 3, 'required']) !!}
8     </div>
9     <div class="form-group">
10      <button type="submit" class="btn btn-primary">
11        Reply
12      </button>
13    </div>
14    {!! Form::close() !!}
15  </div>
16 </div>
17
```

_comment.blade.php

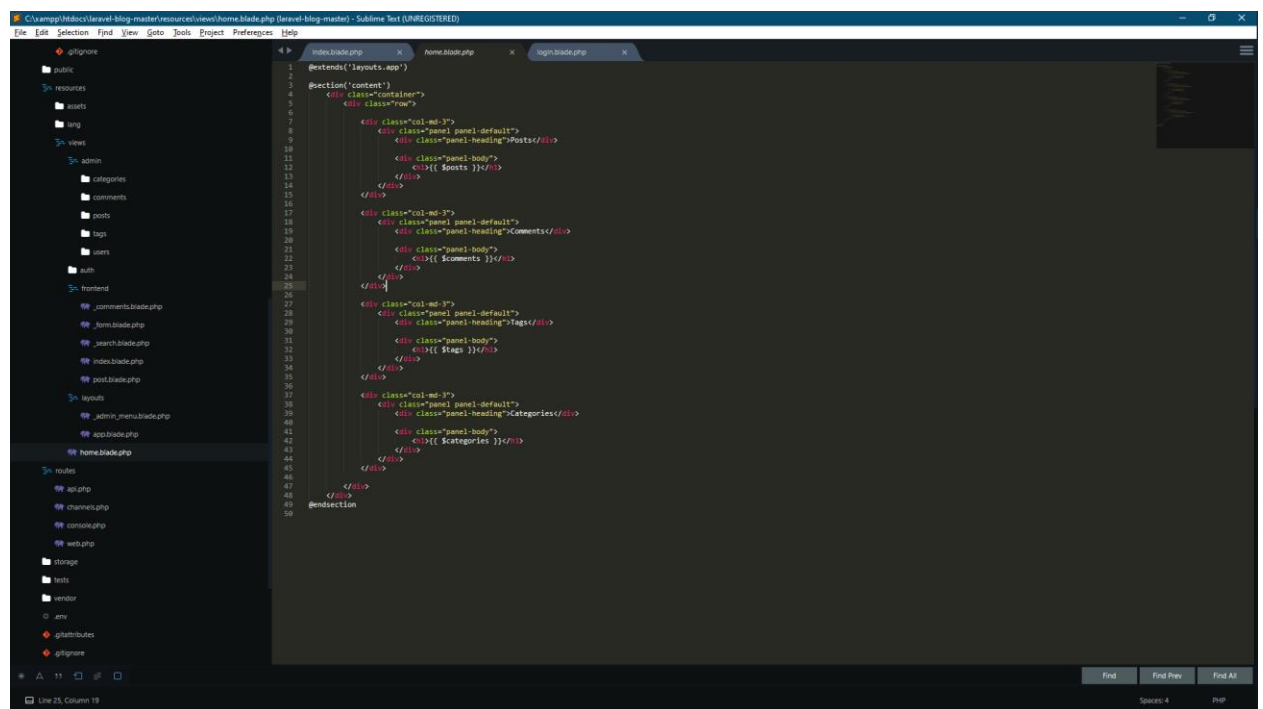


```
1 @foreach ($post->comments as $comment)
2   <div class="panel panel-default">
3     <div class="panel-heading">
4       {{ $comment->user->name }} says...
5     </div>
6     <span class="pull-right">{{ $comment->created_at->diffForHumans() }}</span>
7   </div>
8   <div class="panel-body">
9     <p>{{ $comment->body }}</p>
10   </div>
11 @empty
12   <div class="panel panel-default">
13     <div class="panel-heading">Not Found!</div>
14     <div class="panel-body">
15       <p>Sorry! No comment found for this post.</p>
16     </div>
17   </div>
18 @endforeach
```


Form index



- หน้า index.blade เป็นหน้า template แรกของเว็บโดยมีการ extends app.php ในไฟล์เตอร์ layer มาเรียกใช้ _admin_manu.blade เมื่อค่า status = admin 0t หรือ status = user โดยที่จะเรียกเพิ่มหน้า home.blade เป็นหน้าโชว์ข้อมูลทั้งหมด โดยทำการตรวจ user ไม่ใช่ admin จะไม่ใช่ manu ที่ชื่อว่า users บน admin_manu.blade แต่ถ้าเป็น admin จะมี users ถ้าหากเป็น guest จะมีแค่ manu แค่ login และ register
- มีการเรียกหน้า frontend_search ในการแสดง form สำหรับ search ข้อมูล โดยจะรับค่าที่ posts ที่รับมาจาก search.blade มาเก็บเป็น post โดยเช็คค่าว่าถ้ามี \$posts as post ให้สร้างหัวข้อที่ด้านซ้ายเป็น title – by user->name และด้านซ้ายเป็นวันที่ทำการเพิ่มข้อมูลเข้า
- body มีการกำหนดจำนวนตัวอักษรไม่เกิน 200 ตัวอักษร และเมื่อมี \$post->tags มาเก็บไว้ใน \$tag ทำการแสดง \$tag->name แต่ถ้าไม่มีให้ทำการใส่ No tag found ทำการแสดง \$post->category->name โดยดึงจาก route ชื่อ \$posts โดยการทำการดึงค่า \$post->category->name จาก web.php ที่มีการ route ไปที่ PostController และส่งค่า post ไปด้วย โดยมีการเชื่อมต่อ model ที่ชื่อว่า post ฟังก์ชัน category มาเรียก category->name ใน db โดย button comment และ button see more โดย see more a href {{ url("/posts/{ \$post->id}") }} เป็นการลิงค์ไปหน้า post.blade
- post.blade แสดง form เหมือนเดิมตามหน้า index แต่จะมี @includeWhen(Auth::user(), 'frontend._form') เป็นฟอร์มในการใช้กรอก comment โดยมีการ @include('frontend._comments') ที่เป็นตัวฟอร์มในการแสดง comment
- ถ้าไม่มี \$posts as post ให้ทำการสร้างหัวข้อเป็น not found และ ตัว body เป็น No post found



- เมื่อทำการ login สำเร็จ ทั้ง user และ admin จะเข้าสู่ฟอร์มหน้า home

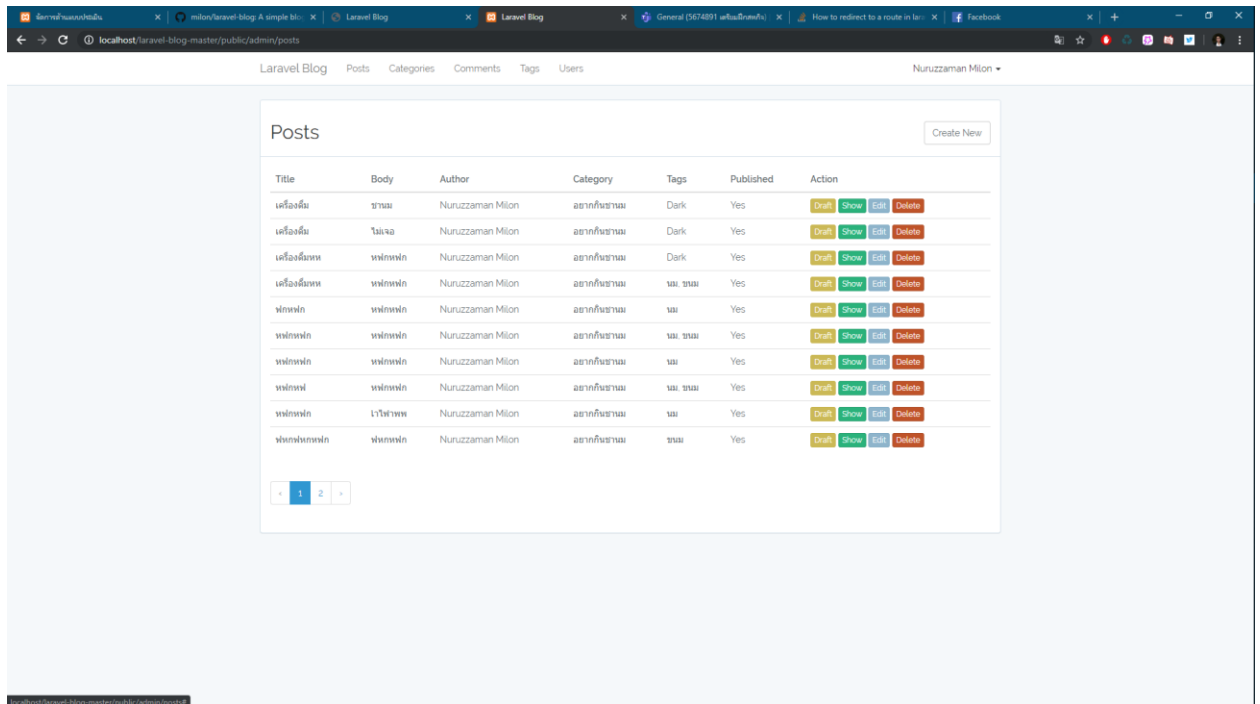
Index.blade.php

```

1 @extends('layouts.app')
2
3 @section('content')
4     <div class="container">
5         <div class="row">
6             <div class="col-md-12">
7                 <div class="panel panel-default">
8                     <div class="panel-heading">
9                         Posts
10                     </div>
11                     <div class="panel-body">
12                         <table class="table">
13                             <thead>
14                                 <tr>
15                                     <th>Title</th>
16                                     <th>Body</th>
17                                     <th>Author</th>
18                                     <th>Category</th>
19                                     <th>Tags</th>
20                                     <th>Published</th>
21                                     <th>Actions</th>
22                                 </tr>
23                             </thead>
24                             <tbody>
25                                 @foreach ($posts as $post)
26                                     <tr>
27                                         <td>{{ $post->title }}</td>
28                                         <td>{{ $post->body }}</td>
29                                         <td>{{ $post->user->name }}</td>
30                                         <td>{{ $post->category->name }}</td>
31                                         <td>{{ $post->tags->implode(' ', ' ') }}</td>
32                                         <td>{{ $post->published }}</td>
33                                         <td>
34                                             @if ($post->published == 'yes')
35                                                 <a href="{{ url('admin/posts/{id}/publish') }}" data-method="PUT" data-token="{{ csrf_token() }}" data-confirm="Are you sure?" class="btn btn-xs">Publish</a>
36                                             @elseif ($post->published == 'no')
37                                                 <a href="{{ url('admin/posts/{id}/draft') }}" class="btn btn-xs btn-success" data-method="PATCH" data-token="{{ csrf_token() }}" data-confirm="Are you sure?">Draft</a>
38                                             @endif
39                                             <a href="{{ url('admin/posts/{id}/edit') }}" class="btn btn-xs btn-info" data-method="PATCH" data-token="{{ csrf_token() }}" data-confirm="Are you sure?">Edit</a>
40                                             <a href="{{ url('admin/posts/{id}/delete') }}" data-method="DELETE" data-token="{{ csrf_token() }}" data-confirm="Are you sure?" class="btn btn-xs btn-danger" data-method="DELETE" data-token="{{ csrf_token() }}">Delete</a>
41                                         </td>
42                                     </tr>
43                                 @endforeach
44                             </tbody>
45                         </table>
46                     </div>
47                 </div>
48             </div>
49         </div>
50     </div>
51
52 @if ($posts->isEmpty())
53     <div class="text-center">
54         No posts available.
55     </div>
56 @endif
57
58 @if ($posts->isEmpty())
59     <div class="text-center">
60         No posts available.
61     </div>
62 @endif
63
64 @if ($posts->isEmpty())
65     <div class="text-center">
66         No posts available.
67     </div>
68 @endif
69
70 @if ($posts->isEmpty())
71     <div class="text-center">
72         No posts available.
73     </div>
74 @endif
75
76 @if ($posts->isEmpty())
77     <div class="text-center">
78         No posts available.
79     </div>
80 @endif
81
82 @if ($posts->isEmpty())
83     <div class="text-center">
84         No posts available.
85     </div>
86 @endif
87
88 @if ($posts->isEmpty())
89     <div class="text-center">
90         No posts available.
91     </div>
92 @endif
93
94 @if ($posts->isEmpty())
95     <div class="text-center">
96         No posts available.
97     </div>
98 @endif
99
100 @if ($posts->isEmpty())
101     <div class="text-center">
102         No posts available.
103     </div>
104 @endif
105
106 @if ($posts->isEmpty())
107     <div class="text-center">
108         No posts available.
109     </div>
110 @endif
111
112 @if ($posts->isEmpty())
113     <div class="text-center">
114         No posts available.
115     </div>
116 @endif
117
118 @if ($posts->isEmpty())
119     <div class="text-center">
120         No posts available.
121     </div>
122 @endif
123
124 @if ($posts->isEmpty())
125     <div class="text-center">
126         No posts available.
127     </div>
128 @endif
129
130 @if ($posts->isEmpty())
131     <div class="text-center">
132         No posts available.
133     </div>
134 @endif
135
136 @if ($posts->isEmpty())
137     <div class="text-center">
138         No posts available.
139     </div>
140 @endif
141
142 @if ($posts->isEmpty())
143     <div class="text-center">
144         No posts available.
145     </div>
146 @endif
147
148 @if ($posts->isEmpty())
149     <div class="text-center">
150         No posts available.
151     </div>
152 @endif
153
154 @if ($posts->isEmpty())
155     <div class="text-center">
156         No posts available.
157     </div>
158 @endif
159
160 @if ($posts->isEmpty())
161     <div class="text-center">
162         No posts available.
163     </div>
164 @endif
165
166 @if ($posts->isEmpty())
167     <div class="text-center">
168         No posts available.
169     </div>
170 @endif
171
172 @if ($posts->isEmpty())
173     <div class="text-center">
174         No posts available.
175     </div>
176 @endif
177
178 @if ($posts->isEmpty())
179     <div class="text-center">
180         No posts available.
181     </div>
182 @endif
183
184 @if ($posts->isEmpty())
185     <div class="text-center">
186         No posts available.
187     </div>
188 @endif
189
190 @if ($posts->isEmpty())
191     <div class="text-center">
192         No posts available.
193     </div>
194 @endif
195
196 @if ($posts->isEmpty())
197     <div class="text-center">
198         No posts available.
199     </div>
200 @endif
201
202 @if ($posts->isEmpty())
203     <div class="text-center">
204         No posts available.
205     </div>
206 @endif
207
208 @if ($posts->isEmpty())
209     <div class="text-center">
210         No posts available.
211     </div>
212 @endif
213
214 @if ($posts->isEmpty())
215     <div class="text-center">
216         No posts available.
217     </div>
218 @endif
219
220 @if ($posts->isEmpty())
221     <div class="text-center">
222         No posts available.
223     </div>
224 @endif
225
226 @if ($posts->isEmpty())
227     <div class="text-center">
228         No posts available.
229     </div>
230 @endif
231
232 @if ($posts->isEmpty())
233     <div class="text-center">
234         No posts available.
235     </div>
236 @endif
237
238 @if ($posts->isEmpty())
239     <div class="text-center">
240         No posts available.
241     </div>
242 @endif
243
244 @if ($posts->isEmpty())
245     <div class="text-center">
246         No posts available.
247     </div>
248 @endif
249
250 @if ($posts->isEmpty())
251     <div class="text-center">
252         No posts available.
253     </div>
254 @endif
255
256 @if ($posts->isEmpty())
257     <div class="text-center">
258         No posts available.
259     </div>
260 @endif
261
262 @if ($posts->isEmpty())
263     <div class="text-center">
264         No posts available.
265     </div>
266 @endif
267
268 @if ($posts->isEmpty())
269     <div class="text-center">
270         No posts available.
271     </div>
272 @endif
273
274 @if ($posts->isEmpty())
275     <div class="text-center">
276         No posts available.
277     </div>
278 @endif
279
280 @if ($posts->isEmpty())
281     <div class="text-center">
282         No posts available.
283     </div>
284 @endif
285
286 @if ($posts->isEmpty())
287     <div class="text-center">
288         No posts available.
289     </div>
290 @endif
291
292 @if ($posts->isEmpty())
293     <div class="text-center">
294         No posts available.
295     </div>
296 @endif
297
298 @if ($posts->isEmpty())
299     <div class="text-center">
300         No posts available.
301     </div>
302 @endif
303
304 @if ($posts->isEmpty())
305     <div class="text-center">
306         No posts available.
307     </div>
308 @endif
309
310 @if ($posts->isEmpty())
311     <div class="text-center">
312         No posts available.
313     </div>
314 @endif
315
316 @if ($posts->isEmpty())
317     <div class="text-center">
318         No posts available.
319     </div>
320 @endif
321
322 @if ($posts->isEmpty())
323     <div class="text-center">
324         No posts available.
325     </div>
326 @endif
327
328 @if ($posts->isEmpty())
329     <div class="text-center">
330         No posts available.
331     </div>
332 @endif
333
334 @if ($posts->isEmpty())
335     <div class="text-center">
336         No posts available.
337     </div>
338 @endif
339
340 @if ($posts->isEmpty())
341     <div class="text-center">
342         No posts available.
343     </div>
344 @endif
345
346 @if ($posts->isEmpty())
347     <div class="text-center">
348         No posts available.
349     </div>
350 @endif
351
352 @if ($posts->isEmpty())
353     <div class="text-center">
354         No posts available.
355     </div>
356 @endif
357
358 @if ($posts->isEmpty())
359     <div class="text-center">
360         No posts available.
361     </div>
362 @endif
363
364 @if ($posts->isEmpty())
365     <div class="text-center">
366         No posts available.
367     </div>
368 @endif
369
370 @if ($posts->isEmpty())
371     <div class="text-center">
372         No posts available.
373     </div>
374 @endif
375
376 @if ($posts->isEmpty())
377     <div class="text-center">
378         No posts available.
379     </div>
380 @endif
381
382 @if ($posts->isEmpty())
383     <div class="text-center">
384         No posts available.
385     </div>
386 @endif
387
388 @if ($posts->isEmpty())
389     <div class="text-center">
390         No posts available.
391     </div>
392 @endif
393
394 @if ($posts->isEmpty())
395     <div class="text-center">
396         No posts available.
397     </div>
398 @endif
399
400 @if ($posts->isEmpty())
401     <div class="text-center">
402         No posts available.
403     </div>
404 @endif
405
406 @if ($posts->isEmpty())
407     <div class="text-center">
408         No posts available.
409     </div>
410 @endif
411
412 @if ($posts->isEmpty())
413     <div class="text-center">
414         No posts available.
415     </div>
416 @endif
417
418 @if ($posts->isEmpty())
419     <div class="text-center">
420         No posts available.
421     </div>
422 @endif
423
424 @if ($posts->isEmpty())
425     <div class="text-center">
426         No posts available.
427     </div>
428 @endif
429
430 @if ($posts->isEmpty())
431     <div class="text-center">
432         No posts available.
433     </div>
434 @endif
435
436 @if ($posts->isEmpty())
437     <div class="text-center">
438         No posts available.
439     </div>
440 @endif
441
442 @if ($posts->isEmpty())
443     <div class="text-center">
444         No posts available.
445     </div>
446 @endif
447
448 @if ($posts->isEmpty())
449     <div class="text-center">
450         No posts available.
451     </div>
452 @endif
453
454 @if ($posts->isEmpty())
455     <div class="text-center">
456         No posts available.
457     </div>
458 @endif
459
460 @if ($posts->isEmpty())
461     <div class="text-center">
462         No posts available.
463     </div>
464 @endif
465
466 @if ($posts->isEmpty())
467     <div class="text-center">
468         No posts available.
469     </div>
470 @endif
471
472 @if ($posts->isEmpty())
473     <div class="text-center">
474         No posts available.
475     </div>
476 @endif
477
478 @if ($posts->isEmpty())
479     <div class="text-center">
480         No posts available.
481     </div>
482 @endif
483
484 @if ($posts->isEmpty())
485     <div class="text-center">
486         No posts available.
487     </div>
488 @endif
489
490 @if ($posts->isEmpty())
491     <div class="text-center">
492         No posts available.
493     </div>
494 @endif
495
496 @if ($posts->isEmpty())
497     <div class="text-center">
498         No posts available.
499     </div>
500 @endif

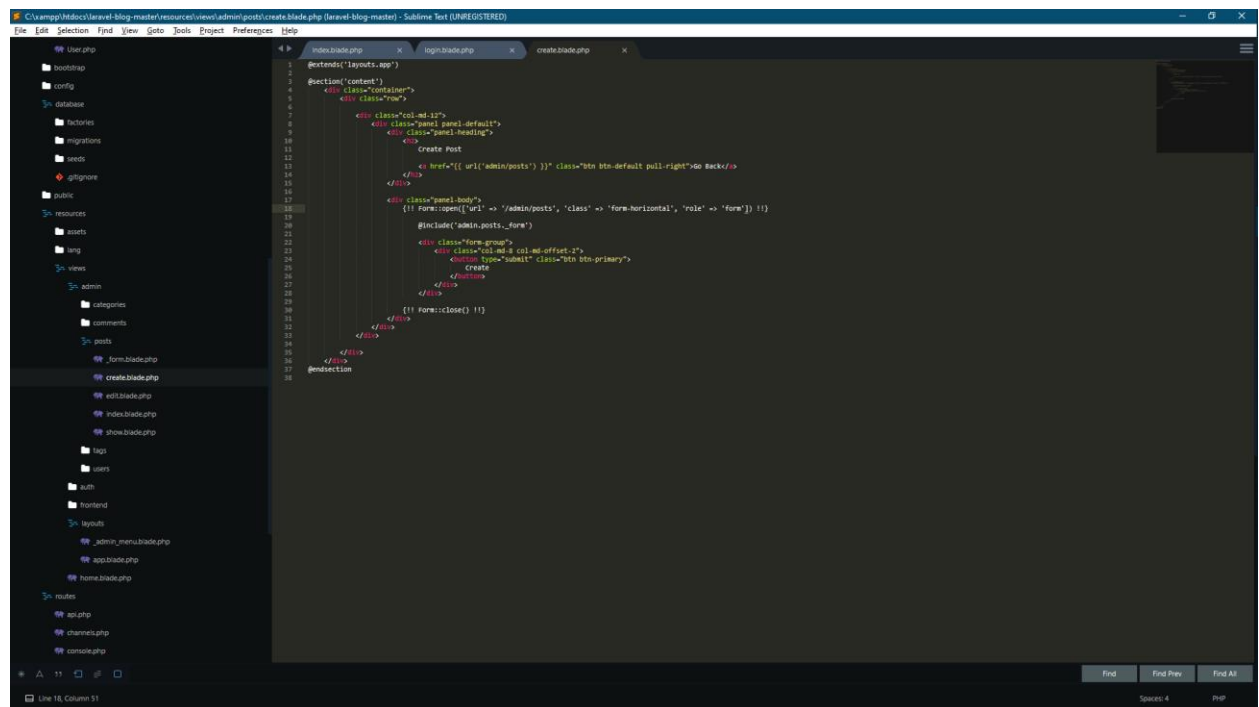
```

Form Post



- เมื่อทำการกด menu ที่ชื่อว่า post บน navbar จะวิ่งเข้าสู่ route {{ url('admin/posts') }} ในคอนโทรเลอร์ โดยรันฟังก์ชันที่ชื่อว่า index โดยฟังก์ชันส่งค่า posts เป็นตัวเก็บ Post::with(['user', 'category', 'tags', 'comments'])->paginate(10) และรันหน้า 'admin.posts.index' โดยหน้าจะแสดงตารางของค่า post ทั้งหมด โดยมีการเช็คค่า publish จากปุ่ม <a href="{{ url('/admin/posts/{<!-- \$post-->id }}/publish") }}" data-method="PUT" data-token="{{ csrf_token() }}" data-confirm="Are you sure?" class="btn btn-xs btn-warning">{{ \$label }} }}" โดยมี data-method="PUT", data-token="{{ csrf_token() }}" และ data-confirm="Are you sure?" โดยเมื่อมีค่า yes เพื่อยืนยันการเปิด publish โดยค่า publish จะมาจาก route {{ url("/admin/posts/{<!-- \$post-->id }}/publish") }}" โดยการเปิดจะเป็น dialog ขึ้นมาให้กดยืนยันเพื่อเปิด publish
- ปุ่ม show เป็นการส่งค่า \$post->id ไปที่คอนโทรเลอร์ฟังก์ชัน show โดย \$post = \$post->load(['user', 'category', 'tags', 'comments']); และ return ค่าไป 'admin.posts.show' แล้วรันหน้า 'admin.posts.show'
- ปุ่ม Edit เป็นการส่งค่า \$post->id ไปที่ route posts ส่งค่าส่งไปที่คอนโทรเลอร์ฟังก์ชัน edit ของ Postcontroller โดยทำการเช็คสถานะว่าเป็น admin หรือไม่ถ้าไม่ให้ขึ้นข้อความ You can't edit other peoples post แล้ว return redirect('/admin/posts'); โดยเก็บค่า \$categories = Category::pluck('name', 'id')->all(); และ \$tags = Tag::pluck('name', 'name')->all(); โดยสั่งให้ return ค่าไปหน้า view('admin.posts.edit', โดยมี compact('post', 'categories', 'tags'));

Crete.blade.php



```

1 @extends('layouts.app')
2
3 @section('content')
4     <div class="container">
5         <div class="row">
6             <div class="col-md-12">
7                 <div class="panel panel-default">
8                     <div class="panel-heading">
9                         Create Post
10                    </div>
11                    <div class="panel-body">
12                        <a href="{{ url('admin/posts') }}" class="btn btn-default pull-right">Go back</a>
13                    </div>
14                </div>
15                <div class="panel-body">
16                    {!! Form::open(['url' => 'admin/posts', 'class' => 'form-horizontal', 'role' => 'form']) !!}
17                    @include('admin.posts._form')
18                </div>
19            </div>
20        </div>
21    </div>
22
23    <div class="form-group">
24        <div class="col-md-8 col-md-offset-2">
25            <input type="submit" class="btn btn-primary">Create
26        </div>
27    </div>
28
29    {!! Form::close() !!}
30 </div>
31 @endsection

```

The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'bootstrap', 'config', 'database', 'factories', 'migrations', 'models', 'resources', 'views', and 'admin'. The code editor displays a Blade template file named 'form.blade.php' with the following content:

```

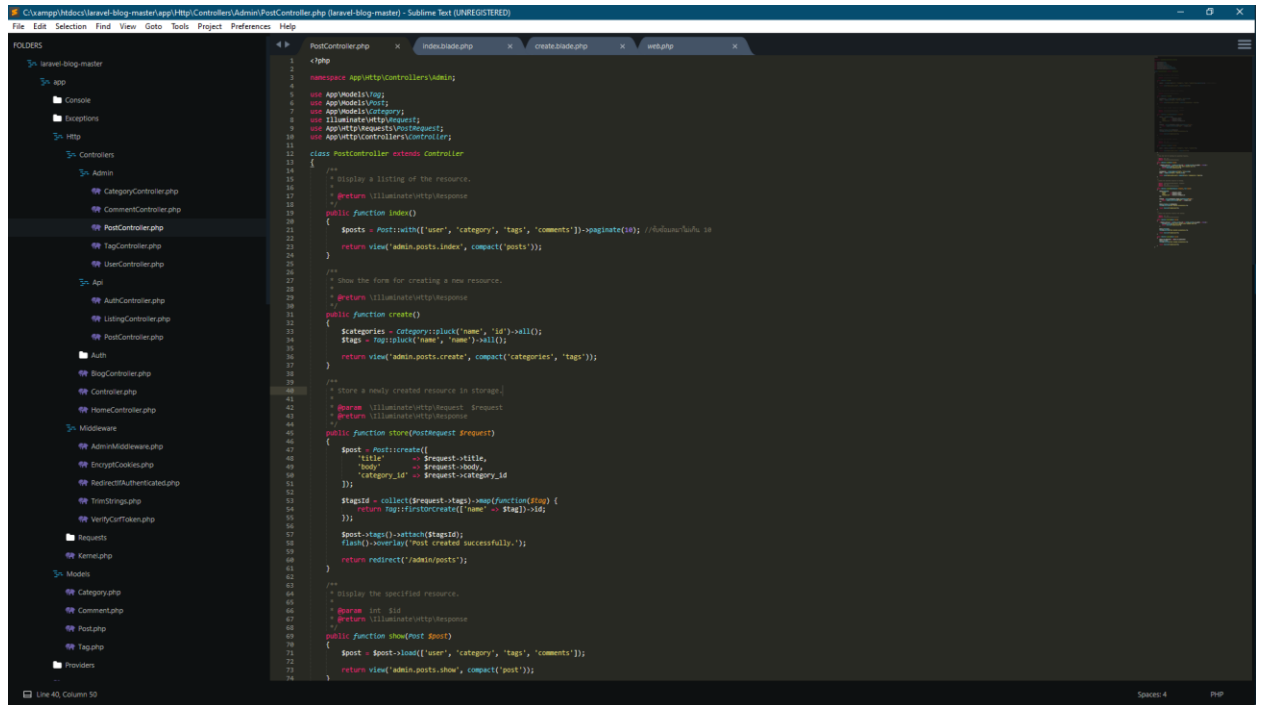
1 <div class="form-group" {{ $errors->has('title') ? ' has-error' : '' }}>
2     {{ Form::label('title', 'Title', ['class' => 'col-md-2 control-label']) }}
3
4     <div class="col-md-8">
5         {{ Form::text('title', null, ['class' => 'form-control', 'required', 'autofocus']) }}
6
7         <span class="help-block">
8             <strong>{{ $errors->first('title') }}</strong>
9         </span>
10    </div>
11
12    <div class="form-group" {{ $errors->has('body') ? ' has-error' : '' }}>
13        {{ Form::label('body', 'Body', ['class' => 'col-md-2 control-label']) }}
14
15        <div class="col-md-8">
16            {{ Form::textarea('body', null, ['class' => 'form-control', 'required']) }}
17
18            <span class="help-block">
19                <strong>{{ $errors->first('body') }}</strong>
20            </span>
21        </div>
22    </div>
23
24    <div class="form-group" {{ $errors->has('category_id') ? ' has-error' : '' }}>
25        {{ Form::label('category_id', 'Category', ['class' => 'col-md-2 control-label']) }}
26
27        <div class="col-md-8">
28            {{ Form::select('category_id', $categories, null, ['class' => 'form-control', 'required']) }}
29
30            <span class="help-block">
31                <strong>{{ $errors->first('category_id') }}</strong>
32            </span>
33        </div>
34    </div>
35
36    @if(isset($tags))
37        <div class="form-group" {{ $errors->has('tags') ? ' has-error' : '' }}>
38            {{ Form::label('tags', 'Tag', ['class' => 'col-md-2 control-label']) }}
39
40            <div class="col-md-8">
41                {{ Form::select('tags[]', $tags, $tag, ['class' => 'form-control select-tags', 'required', 'multiple']) }}
42
43                <span class="help-block">
44                    <strong>{{ $errors->first('tags') }}</strong>
45                </span>
46            </div>
47        </div>
48    @endif

```

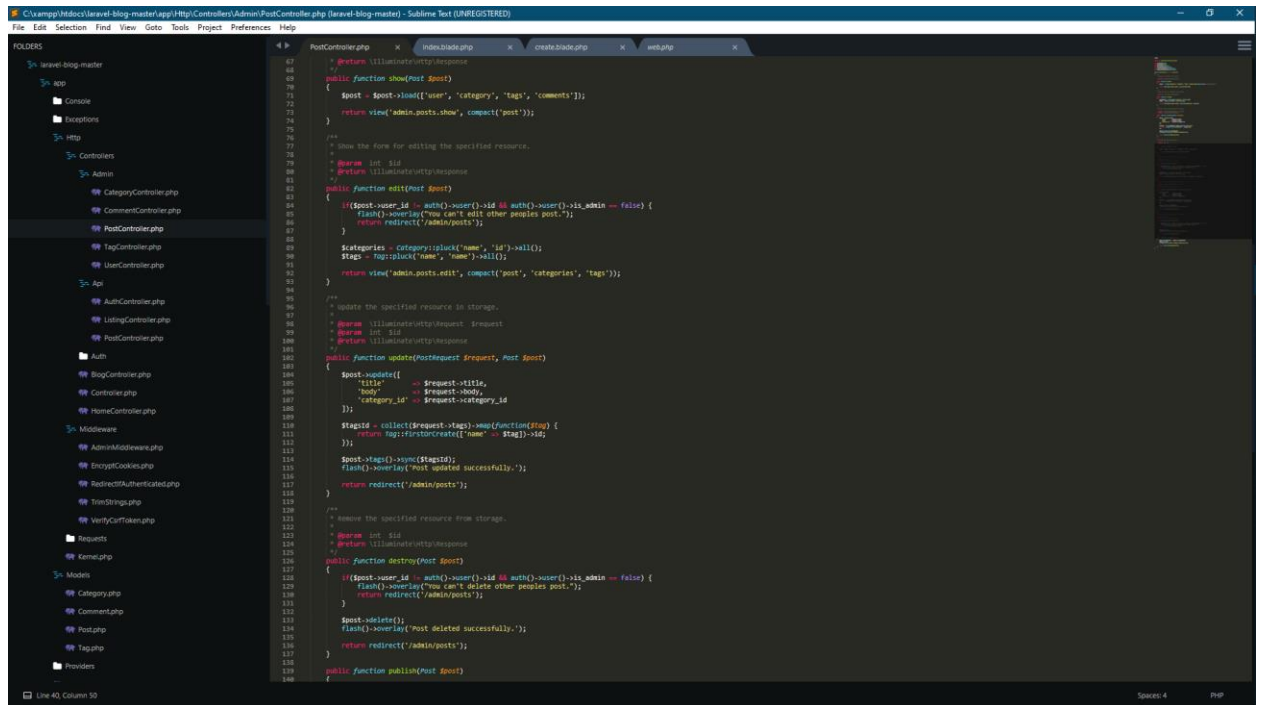
The status bar at the bottom indicates 'Line 1, Column 1: Detect indentation: Setting indentation to 4 spaces'.

The screenshot displays a web browser window with multiple tabs. The active tab is titled 'Laravel Blog' and shows a URL of 'localhost:8080/laravel-blog-master/public/admin/posts/create'. The browser's address bar and navigation icons are visible at the top. Below the browser window, a navigation bar for the 'Laravel Blog' application is shown, featuring links for 'Posts', 'Categories', 'Comments', and 'Tags'. The main content area is a 'Create Post' form. This form contains a 'Title' input field, a 'Body' text area, a 'Category' dropdown menu (currently displaying 'เลือกหมวดหมู่'), and a 'Tag' input field. A 'Go Back' button is located in the top right corner of the form, and a blue 'Create' button is positioned at the bottom left.

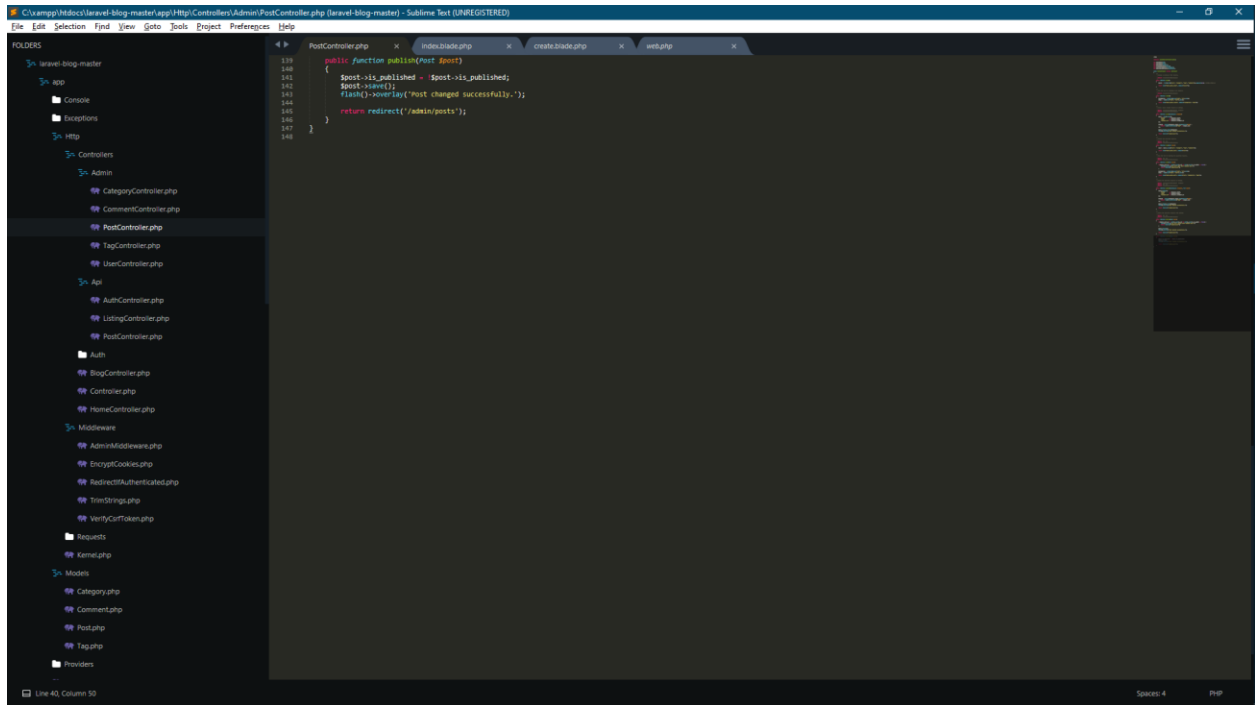
Postcontroller



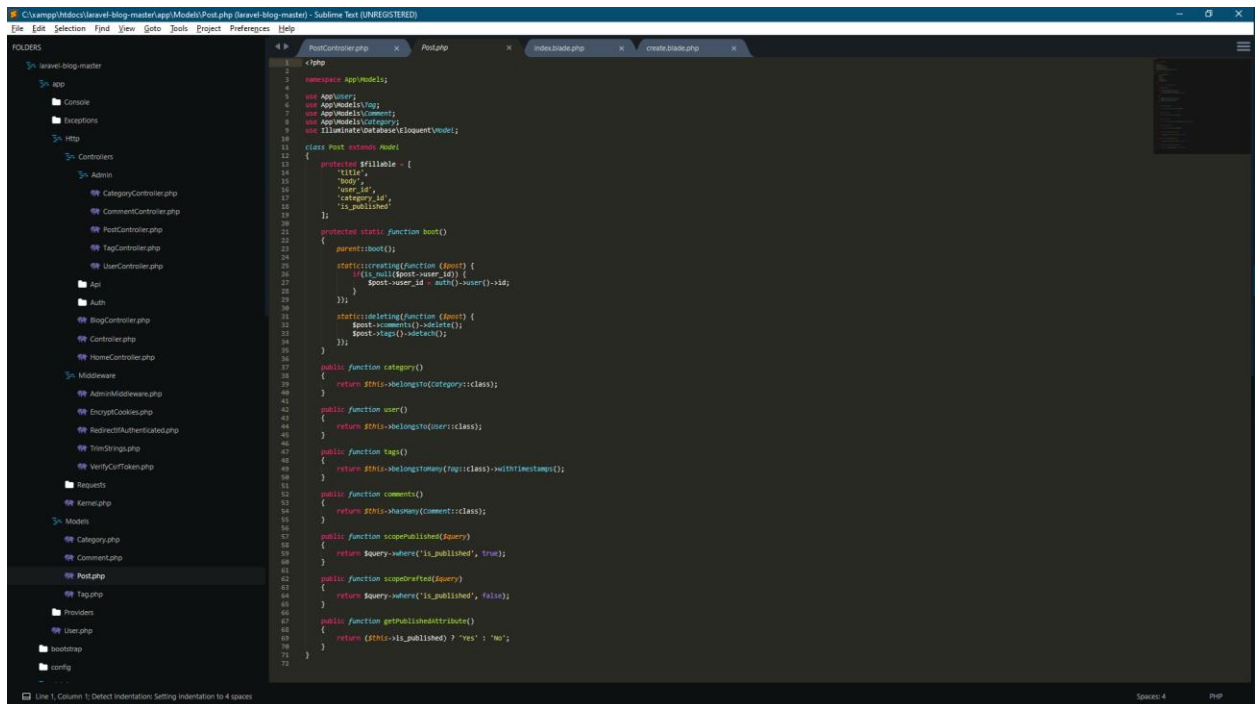
```
1 <?php
2
3 namespace App\Http\Controllers\Admin;
4
5 use App\Models\Post;
6 use App\Models\Category;
7 use App\Models\Tag;
8 use Illuminate\Http\Request;
9 use App\Http\Requests\PostRequest;
10 use App\Http\Controllers\Controller;
11
12 class PostController extends Controller
13 {
14     /**
15      * Display a listing of the resource.
16      *
17      * @return \Illuminate\Http\Response
18      */
19     public function index()
20     {
21         $posts = Post::with(['user', 'category', 'tags', 'comments'])->paginate(10); // default 10
22         return view('admin.posts.index', compact('posts'));
23     }
24
25     /**
26      * Show the form for creating a new resource.
27      *
28      * @return \Illuminate\Http\Response
29      */
30     public function create()
31     {
32         $categories = Category::pluck('name', 'id')->all();
33         $tags = Tag::pluck('name', 'name')->all();
34         return view('admin.posts.create', compact('categories', 'tags'));
35     }
36
37     /**
38      * Store a newly created resource in storage.
39      *
40      * @param \Illuminate\Http\Request $request
41      * @return \Illuminate\Http\Response
42      */
43     public function store(PostRequest $request)
44     {
45         $post = Post::create([
46             'title' => $request->title,
47             'body' => $request->body,
48             'category_id' => $request->category_id
49         ]);
50         $tagsid = collect($request->tags)->map(function($tag) {
51             return Tag::firstOrCreate(['name' => $tag])->id;
52         });
53         $post->tags()->sync($tagsid);
54         flash()->success('Post created successfully.')->toSession();
55         return redirect('/admin/posts');
56     }
57
58     /**
59      * Display the specified resource.
60      *
61      * @param int $id
62      * @return \Illuminate\Http\Response
63      */
64     public function show(Post $post)
65     {
66         $post = $post->load(['user', 'category', 'tags', 'comments']);
67         return view('admin.posts.show', compact('post'));
68     }
69 }
```



```
69
70     /**
71      * Show the form for editing the specified resource.
72      *
73      * @param int $id
74      * @return \Illuminate\Http\Response
75      */
76     public function edit(Post $post)
77     {
78         if($post->user_id != auth()->user()->id && auth()->user()->is_admin == false) {
79             flash()->warning('You can't edit other peoples post.')->toSession();
80             return redirect('/admin/posts');
81         }
82         $categories = Category::pluck('name', 'id')->all();
83         $tags = Tag::pluck('name', 'name')->all();
84         return view('admin.posts.edit', compact('post', 'categories', 'tags'));
85     }
86
87     /**
88      * Update the specified resource in storage.
89      *
90      * @param \Illuminate\Http\Request $request
91      * @param int $id
92      * @return \Illuminate\Http\Response
93      */
94     public function update(PostRequest $request, Post $post)
95     {
96         $post->update([
97             'title' => $request->title,
98             'body' => $request->body,
99             'category_id' => $request->category_id
100         ]);
101         $tagsid = collect($request->tags)->map(function($tag) {
102             return Tag::firstOrCreate(['name' => $tag])->id;
103         });
104         $post->tags()->sync($tagsid);
105         flash()->success('Post updated successfully.')->toSession();
106         return redirect('/admin/posts');
107     }
108
109     /**
110      * Remove the specified resource from storage.
111      *
112      * @param int $id
113      * @return \Illuminate\Http\Response
114      */
115     public function destroy(Post $post)
116     {
117         if($post->user_id != auth()->user()->id && auth()->user()->is_admin == false) {
118             flash()->warning('You can't delete other peoples post.')->toSession();
119             return redirect('/admin/posts');
120         }
121         $post->delete();
122         flash()->success('Post deleted successfully.')->toSession();
123         return redirect('/admin/posts');
124     }
125
126     /**
127      * Publish the specified resource.
128      *
129      * @param Post $post
130      */
131     public function publish(Post $post)
132     {
133         $post->update(['status' => 'published']);
134         flash()->success('Post published successfully.')->toSession();
135         return redirect('/admin/posts');
136     }
137 }
```

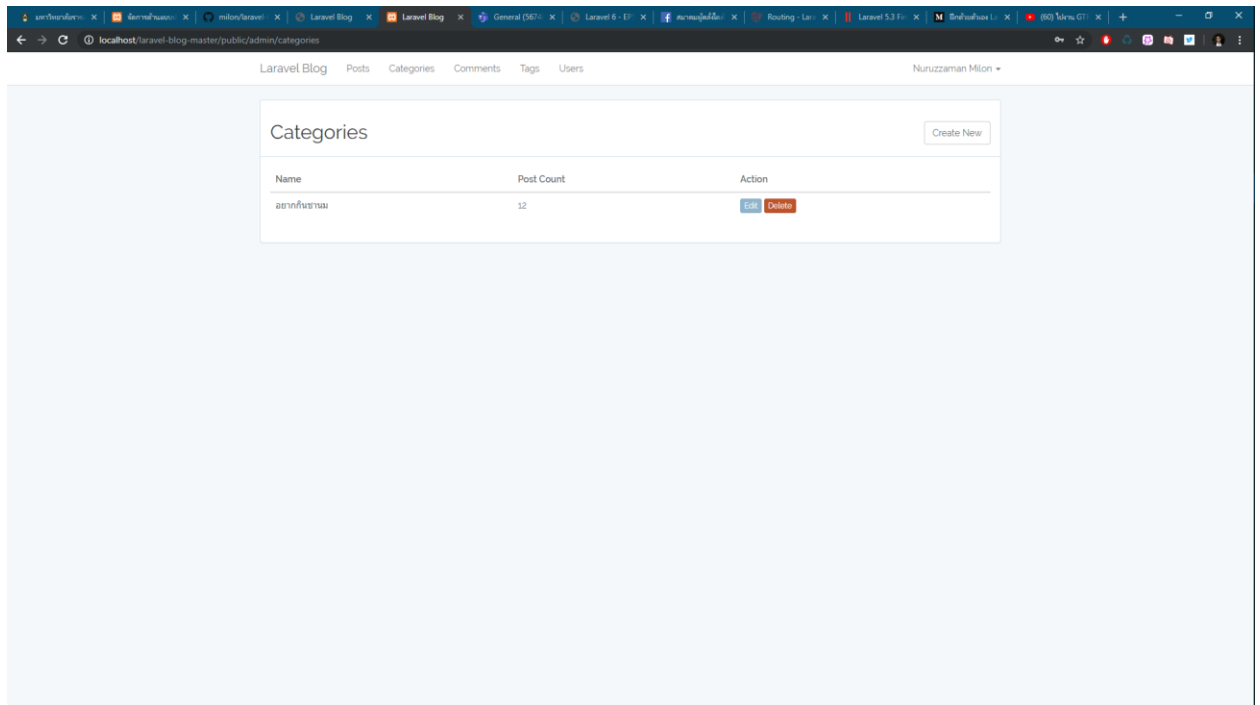


Post model

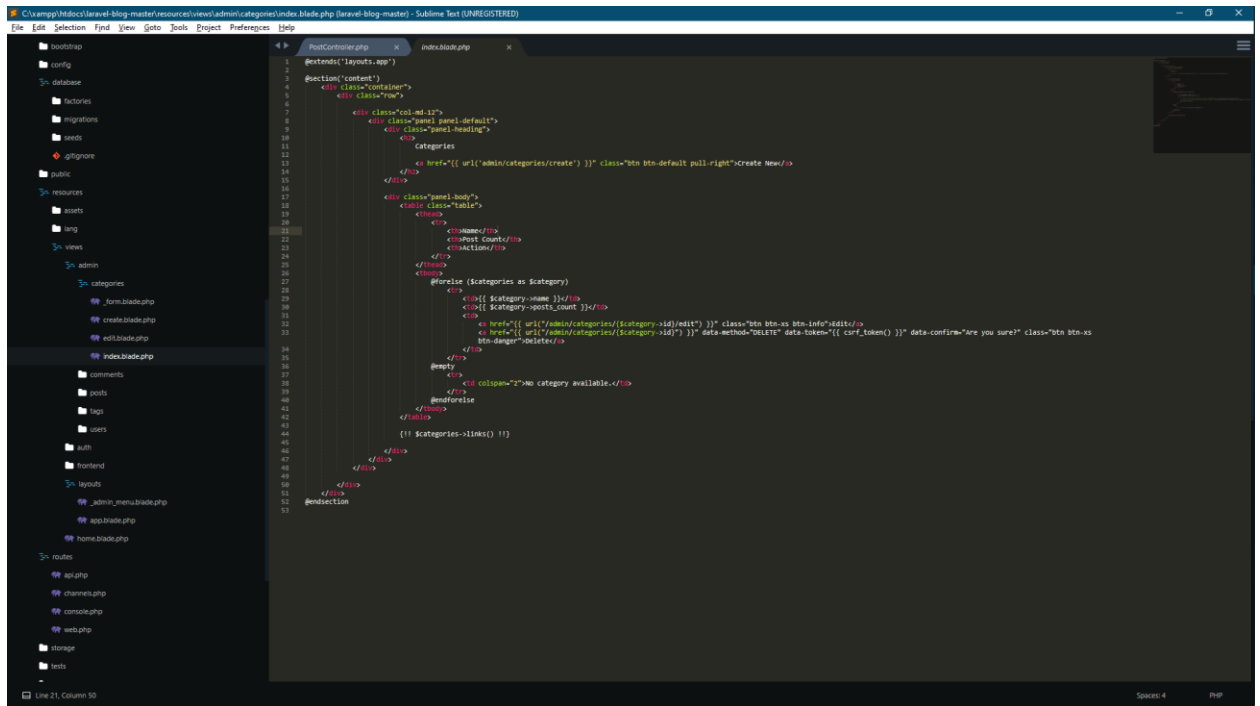


- เมื่อกดปุ่ม create now จะทำการวิ่งเข้าสู่ route {{ url('admin/posts/create') }} โดย create เป็นพารามิเตอร์ไปยังคอนโทรลเลอร์ที่ชื่อว่า postcontroller โดยจะมี url ของฟอร์มคือ '/admin/posts' โดยมีการ @include('admin.posts._form') เป็นฟอร์มตัว input ค่าต่างๆ โดยจะมีการ alert ค่าต่างๆโดย \$errors และมีการเช็คค่า \$post มีค่าส่งมาหรือไม่ โดยเก็บค่า tags ต่างๆ
- เมื่อทำการ input ค่าที่ต้องการ ให้กด create จะทำการเข้าสู่ route posts ไปยังคอนโทรลเลอร์ ชื่อว่า Postcontroller ฟังก์ชัน store ที่ทำการ เก็บค่าที่ post มาไว้ใน \$request โดย \$post เป็นค่า \$post = Post::create(['title' => \$request->title, 'body' => \$request->body, 'category_id' => \$request->category_id]);
- Delete จะทำการส่งค่า \$post->id โดย route ไปที่ postcontroller ฟังก์ชัน destroy เช็คว่าเป็น admin เท่านั้นถึงจะลบได้ โดยทำการลบโดยใช้ db ที่ชื่อ delete เมื่อลบสำเร็จแสดง Post deleted successfully

Form category

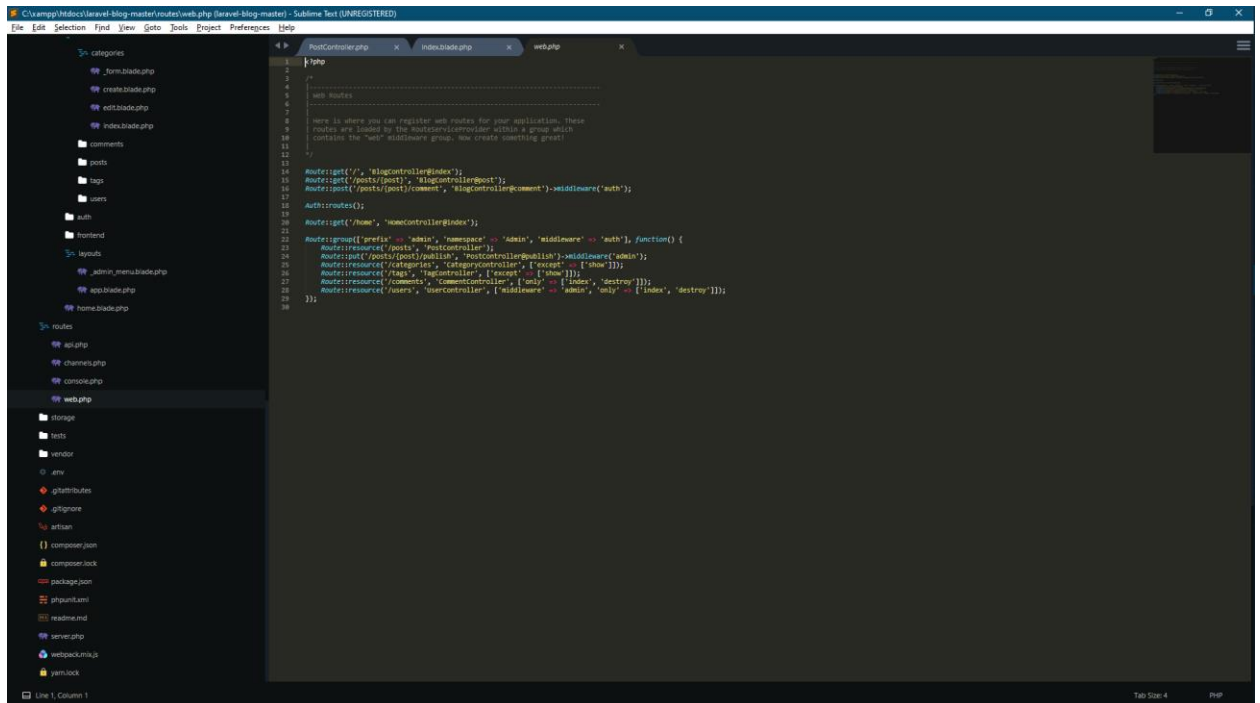


Index.blade.php



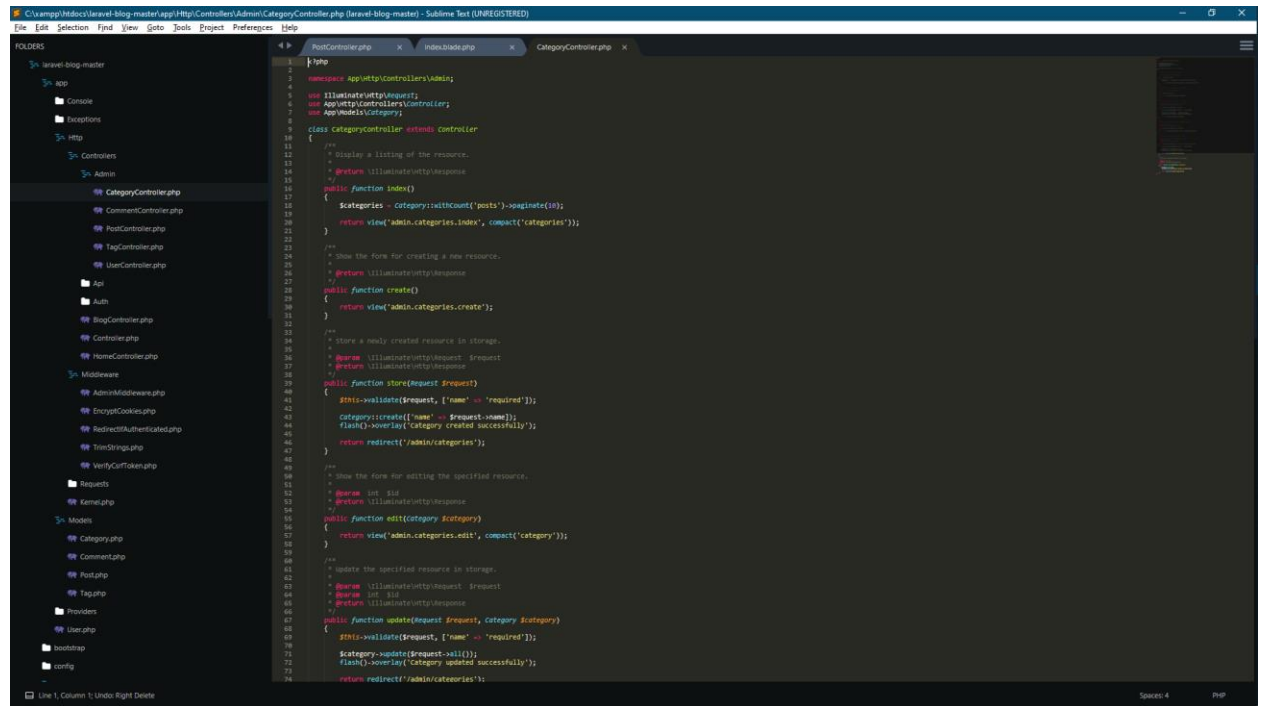
```
1 @extends('layouts.app')
2
3 @section('content')
4     <div class="container">
5         <div class="row">
6
7             <div class="col-md-12">
8                 <div class="panel panel-default">
9                     <div class="panel-heading">
10                         Categories
11                     </div>
12                     <div>
13                         <a href="{{ url('admin/categories/create') }}" class="btn btn-default pull-right">Create New</a>
14                     </div>
15                     <div class="panel-body">
16                         <table class="table">
17                             <thead>
18                                 <tr>
19                                     <th>Name</th>
20                                     <th>Post Count</th>
21                                 </tr>
22                             </thead>
23                             <tbody>
24                                 @foreach ($categories as $category)
25                                     <tr>
26                                         <td>{{ $category->name }}</td>
27                                         <td>{{ $category->posts_count }}</td>
28                                     </tr>
29                                     <tr>
30                                         <td colspan="2">
31                                             <a href="{{ url('admin/categories/{category-id}/edit') }}" class="btn btn-xs btn-info">Edit</a>
32                                             <a href="{{ url('admin/categories/{category-id}') }}" data-method="DELETE" data-token="{{ csrf_token() }}" data-confirm="Are you sure?" class="btn btn-xs btn-danger">Delete</a>
33                                         </td>
34                                     </tr>
35                                 </tbody>
36                             </table>
37                         </div>
38                         <div class="text-center">
39                             <div class="colspan=2">No category available.</div>
40                         </div>
41                     </div>
42                 </div>
43             </div>
44         </div>
45     </div>
46 @endsection
```

web.php

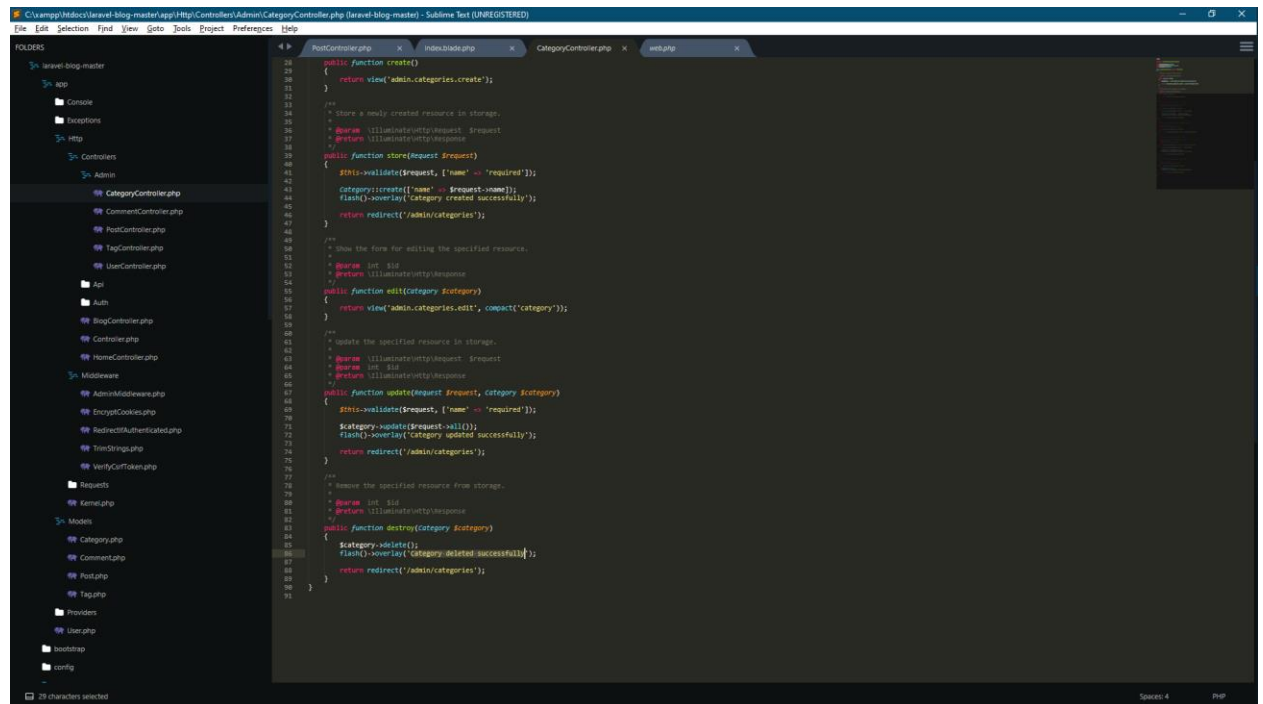


```
1 <?php
2
3 //
4
5 web Routes
6
7 Here is where you can register web routes for your application. These
8 routes are loaded by the RouteServiceProvider within a group which
9 contains the "web" middleware group. Now create something great!
10
11
12
13
14 Route::get('/', 'BlogController@index');
15 Route::get('posts/{post}', 'BlogController@post');
16 Route::post('posts/{post}/comment', 'BlogController@comment')->middleware('auth');
17
18 Auth::routes();
19
20 Route::get('/home', 'HomeController@index');
21
22 Route::group(['prefix' => 'admin', 'namespace' => 'admin', 'middleware' => 'auth'], function() {
23     Route::resource('posts', 'PostController');
24     Route::post('posts/{post}/publish', 'PostController@publish')->middleware('admin');
25     Route::resource('categories', 'CategoryController', ['except' => ['show']]);
26     Route::resource('tags', 'TagController', ['except' => ['show']]);
27     Route::resource('comments', 'CommentController', ['only' => ['index', 'destroy']]);
28     Route::resource('users', 'UserController', ['middleware' => 'admin', 'only' => ['index', 'destroy']]);
29 });
```

Category.blade.php

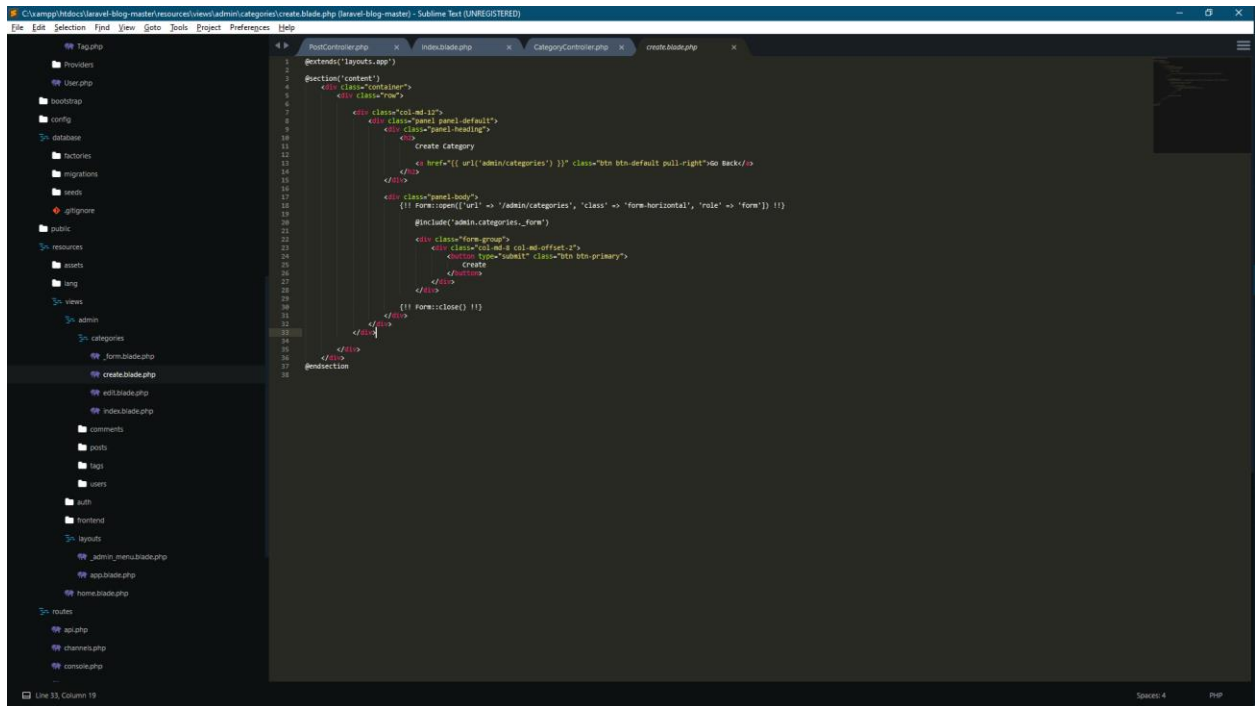


```
1 <?php
2 namespace App\Http\Controllers\Admin;
3
4 use Illuminate\Http\Request;
5 use App\Http\Controllers\Controller;
6 use App\Models\Category;
7
8 class CategoryController extends Controller
9 {
10     /**
11      * Display a listing of the resource.
12      */
13     public function index()
14     {
15         $categories = Category::withCount('posts')->paginate(10);
16         return view('admin.categories.index', compact('categories'));
17     }
18
19     /**
20      * Show the form for creating a new resource.
21      */
22     public function create()
23     {
24         return view('admin.categories.create');
25     }
26
27     /**
28      * Store a newly created resource in storage.
29      */
30     public function store(Request $request)
31     {
32         $this->validate($request, ['name' => 'required']);
33         $category = Category::create(['name' => $request->name]);
34         flash()>overlay('Category created successfully');
35         return redirect('admin/categories');
36     }
37
38     /**
39      * Show the form for editing the specified resource.
40      */
41     public function edit(Category $category)
42     {
43         return view('admin.categories.edit', compact('category'));
44     }
45
46     /**
47      * Update the specified resource in storage.
48      */
49     public function update(Request $request, Category $category)
50     {
51         $this->validate($request, ['name' => 'required']);
52         $category->update($request->all());
53         flash()>overlay('Category updated successfully');
54         return redirect('admin/categories');
55     }
56
57     /**
58      * Remove the specified resource from storage.
59      */
60     public function destroy(Category $category)
61     {
62         $category->delete();
63         flash()>overlay('Category deleted successfully');
64         return redirect('admin/categories');
65     }
66 }
```



```
26 public function create()
27 {
28     return view('admin.categories.create');
29 }
30
31 /**
32  * Store a newly created resource in storage.
33  */
34 public function store(Request $request)
35 {
36     $this->validate($request, ['name' => 'required']);
37     $category = Category::create(['name' => $request->name]);
38     flash()>overlay('Category created successfully');
39     return redirect('admin/categories');
40 }
41
42 /**
43  * Show the form for editing the specified resource.
44  */
45 public function edit(Category $category)
46 {
47     return view('admin.categories.edit', compact('category'));
48 }
49
50 /**
51  * Update the specified resource in storage.
52  */
53 public function update(Request $request, Category $category)
54 {
55     $this->validate($request, ['name' => 'required']);
56     $category->update($request->all());
57     flash()>overlay('Category updated successfully');
58     return redirect('admin/categories');
59 }
60
61 /**
62  * Remove the specified resource from storage.
63  */
64 public function destroy(Category $category)
65 {
66     $category->delete();
67     flash()>overlay('Category deleted successfully');
68     return redirect('admin/categories');
69 }
70 }
```

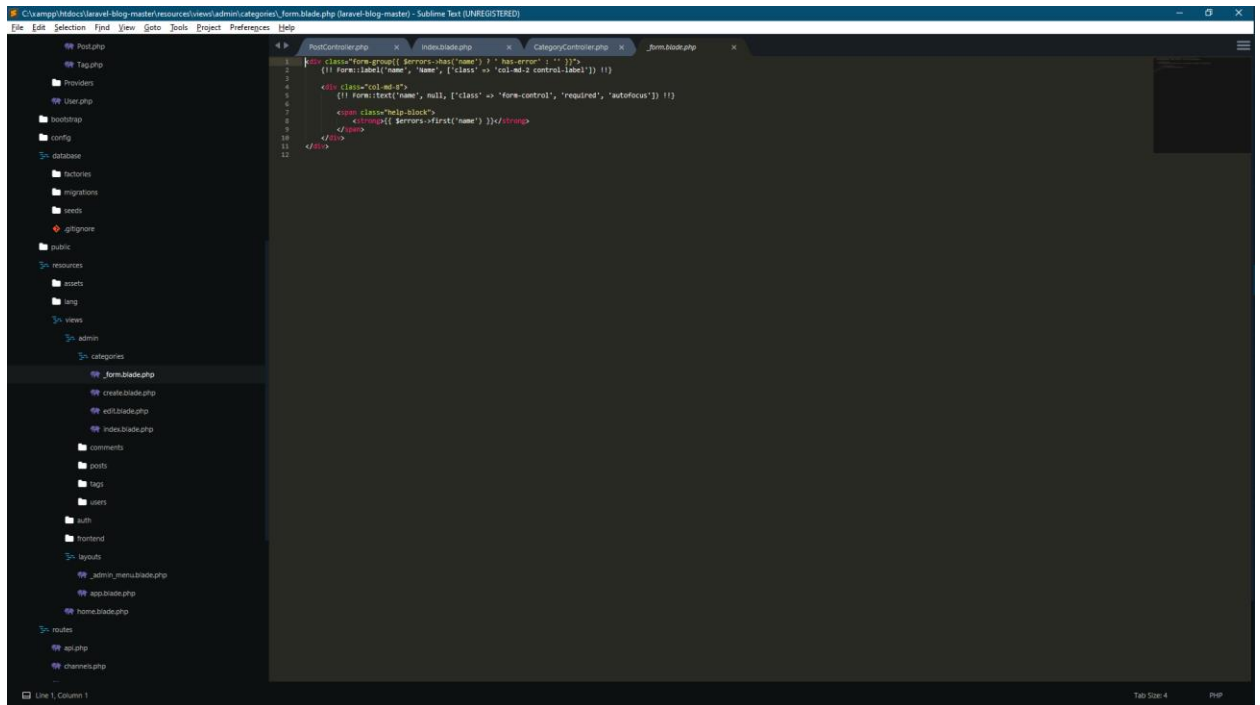
Create.blade.php



The screenshot shows a code editor with the file `create.blade.php` open. The file contains Blade template code for creating a new category. The code is as follows:

```
1 @extends('layouts.app')
2
3 @section('content')
4     <div class="container">
5         <div class="row">
6             <div class="col-md-12">
7                 <div class="panel panel-default">
8                     <div class="panel-heading">
9                         Create Category
10                     </div>
11                     <div class="panel-body">
12                         <div class="form-group">
13                             <input type="text" class="form-control" value="{{ old('name') }}" />
14                         </div>
15                         <div class="form-group">
16                             <input type="text" class="form-control" value="{{ old('description') }}" />
17                         </div>
18                         <div class="form-group">
19                             <input type="button" value="Create" class="btn btn-primary" />
20                         </div>
21                     </div>
22                 </div>
23             </div>
24         </div>
25     </div>
26 @endsection
```

_form.blade.php

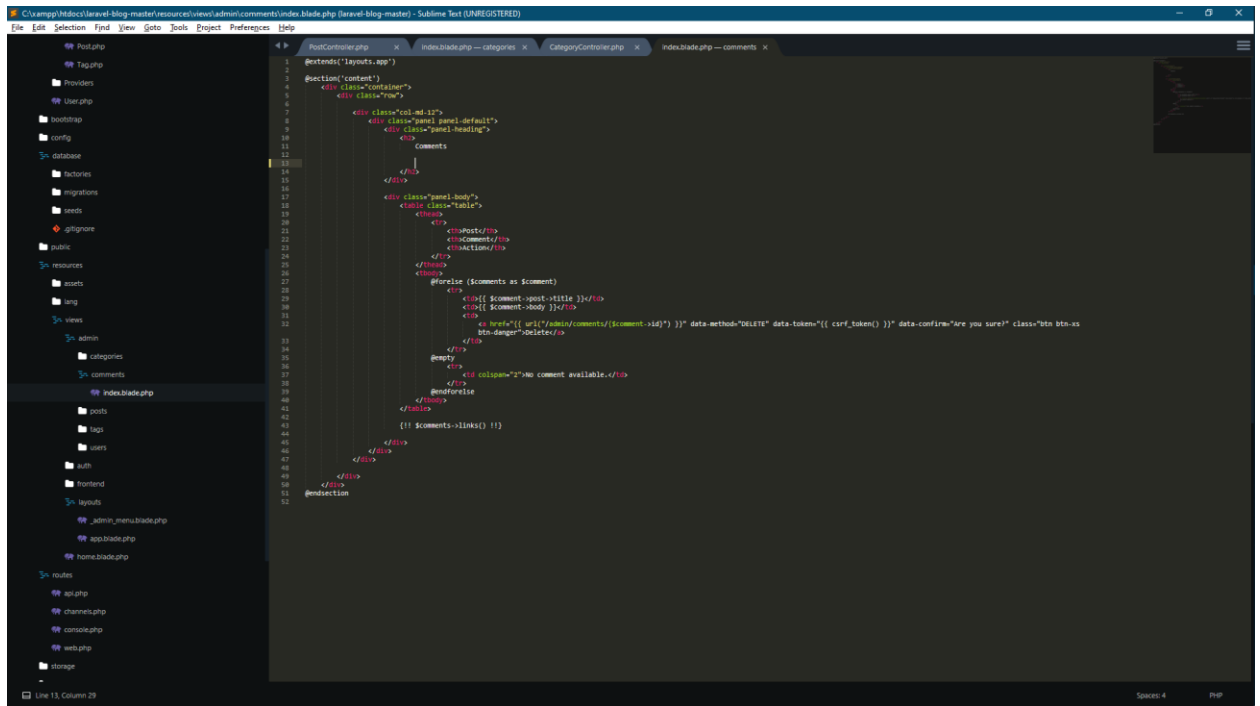


The screenshot shows a code editor with the file `_form.blade.php` open. The file contains Blade template code for a form. The code is as follows:

```
1 <div class="form-group">
2     <input type="text" class="form-control" value="{{ old('name') }}" />
3 </div>
4
5 <div class="form-group">
6     <input type="text" class="form-control" value="{{ old('description') }}" />
7 </div>
8
9 <div class="form-group">
10     <input type="button" value="Create" class="btn btn-primary" />
11 </div>
```

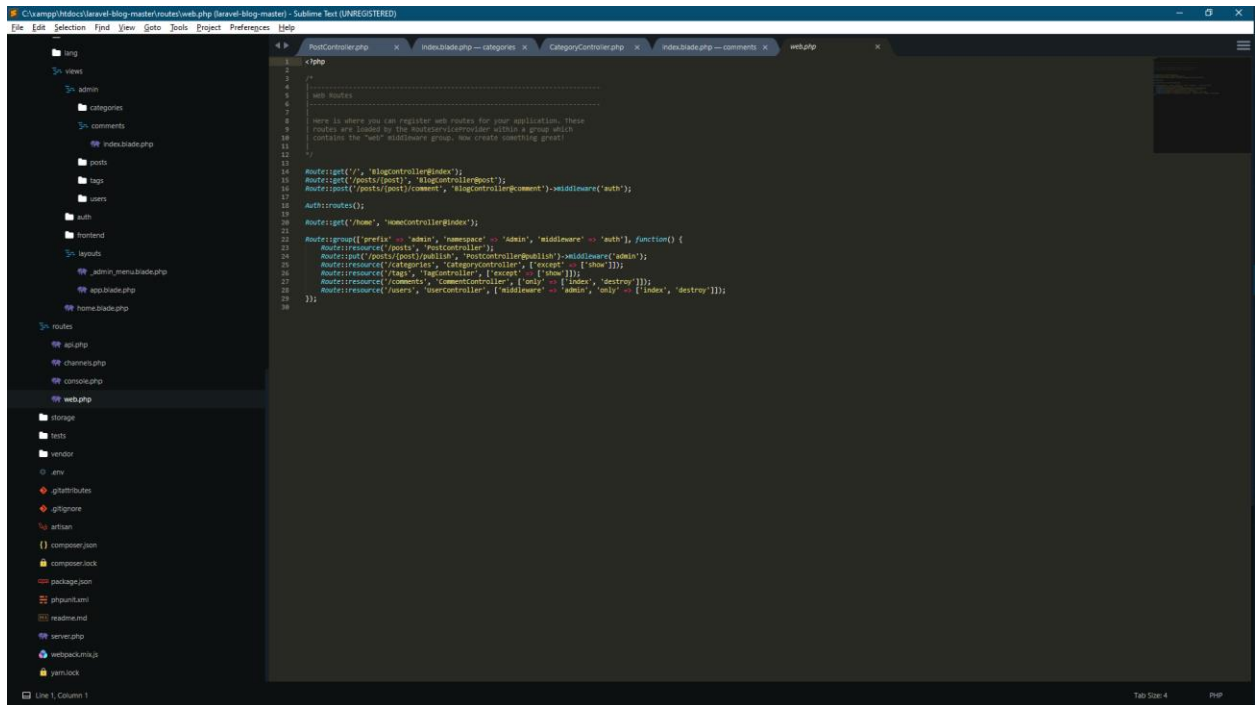
- Menu category เมื่อกดจะเข้าสู่หน้า index ของ category เป็นหน้าที่เป็นฟอร์มไว้ใช้แสดงข้อมูล category ทั้งหมด
- ปุ่ม create now เป็นการส่งไปหน้าฟอร์มที่ใช้เพิ่ม category โดยใช้ @include('admin.categories._form') เป็นฟอร์ม input และมี route {{ url('admin/categories/create') }} โดยที่ route จะส่งค่าไปที่คอนโทรลเลอร์ categorycontroller ฟังก์ชัน create โดยฟังก์ชันจะสั่งทำการ return view('admin.categories.create'); ไปที่ หน้า create ใน admin.category
- เมื่อทำการ input ค่าที่ต้องการ ให้กด create จะทำการเข้าสู่ route category ไปยังคอนโทรลเลอร์ ชื่อว่า categorycontroller ฟังก์ชัน store ที่ทำการ เก็บค่าที่ post มาไว้ใน \$request โดย \$post เป็นค่า Category::create(['name' => \$request->name]); ถ้าอัปสำเร็จแสดง 'Category created successfully' แล้วรีเทิร์นค่าไป /admin/categories ที่เป็นคอนโทรลเลอร์เริ่มที่เมื่อกดเข้ามาจะทำคำสั่งให้แสดงหน้า show
- ปุ่ม Edit จะใช้ @include('admin.categories._form') เป็นฟอร์ม input เหมือนเดิม จะเป็นการส่งค่า \$category->id ไปที่ route category ส่งคำสั่งไปที่คอนโทรลเลอร์ฟังก์ชัน edit ของ categorycontroller โดยคำสั่งจะทำการเรียกหน้า ฟอร์มพร้อมทั้งใส่ข้อมูลมาให้แล้ว
- เมื่อแก้ไขกดปุ่มแก้ไขจะทำการนำ \$category->update(\$request->all()); ที่มีคำสั่งโมเดล category ที่ชื่อ update ในการสั่งในอัปเดต \$request ที่ส่งมาลงฐานข้อมูล
- Delete จะทำการส่งค่า \$category->id โดย route ไปที่ categorycontroller ฟังก์ชัน destroy เชื่อกว่าสแตตัสเป็น admin เท่านั้นถึงจะลบได้ โดยทำการลบโดยใช้ db ที่ชื่อ delete เมื่อลบสำเร็จแสดง Category deleted successfully รีเทิร์นค่าไปหน้า show

Index.blade.php



```
1 @extends('layouts.app')
2
3 @section('content')
4     <div class="container">
5         <div class="row">
6             <div class="col-md-12">
7                 <div class="panel panel-default">
8                     <div class="panel-heading">
9                         Comments
10                     </div>
11                     <div class="panel-body">
12                         <table class="table">
13                             <thead>
14                                 <tr>
15                                     <th>ID</th>
16                                     <th>Body</th>
17                                     <th>Actions</th>
18                                 </tr>
19                             </thead>
20                             <tbody>
21                                 @foreach($comments as $comment)
22                                     <tr>
23                                         <td>{{ $comment->id }}</td>
24                                         <td>{{ $comment->body }}</td>
25                                         <td>
26                                             <a href="{{ url('/admin/comments/' . $comment->id . 'edit') }}" data-method="DELETE" data-token="{{ csrf_token() }}" data-confirm="Are you sure?" class="btn btn-danger">Delete</a>
27                                             <a href="#">Edit</a>
28                                         </td>
29                                     </tr>
30                                 </tbody>
31                             </table>
32                         </div>
33                     </div>
34                 </div>
35             </div>
36         </div>
37     </section>
38 @endsection
```

web.php



```
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4
5 /*
6  * Here you can register web routes for your application. These
7  * routes are loaded by the RouteServiceProvider within a group which
8  * contains the "web" middleware group. Now create something great!
9  */
10
11 Route::get('/', 'BlogController@index');
12 Route::get('/posts/{post}', 'BlogController@post');
13 Route::get('/posts/{post}/comment', 'BlogController@comment')->middleware('auth');
14
15 Auth::routes();
16
17 Route::get('/home', 'HomeController@index');
18
19 Route::group(['prefix' => 'admin', 'namespace' => 'Admin', 'middleware' => 'auth'], function() {
20     Route::resource('posts', 'PostController');
21     Route::post('posts/{post}/publish', 'PostController@publish')->middleware('admin');
22     Route::resource('categories', 'CategoryController', ['except' => ['show']]);
23     Route::resource('tags', 'TagController', ['except' => ['show']]);
24     Route::resource('comments', 'CommentController', ['only' => ['index', 'destroy']]);
25     Route::resource('users', 'UserController', ['middleware' => 'admin', 'only' => ['index', 'destroy']]);
26 });
```

- Menu comment เมื่อเข้ามาจะทำคำสั่ง route ไปที่ commentcontroller ในการสั่งให้ใช้ฟังก์ชัน index เพื่อแสดงหน้าแสดง comment ทั้งหมด
- Delete จะทำการส่งค่า \$category->id โดย route ไปที่ commentcontroller ฟังก์ชัน destroy เช็คว่าสแตตัสเป็น admin เท่านั้นถึงจะลบได้ โดยทำการลบโดยใช้ db ที่ชื่อ delete เมื่อลบสำเร็จแสดง comment deleted successfully รีเธรนค่าไปหน้า show

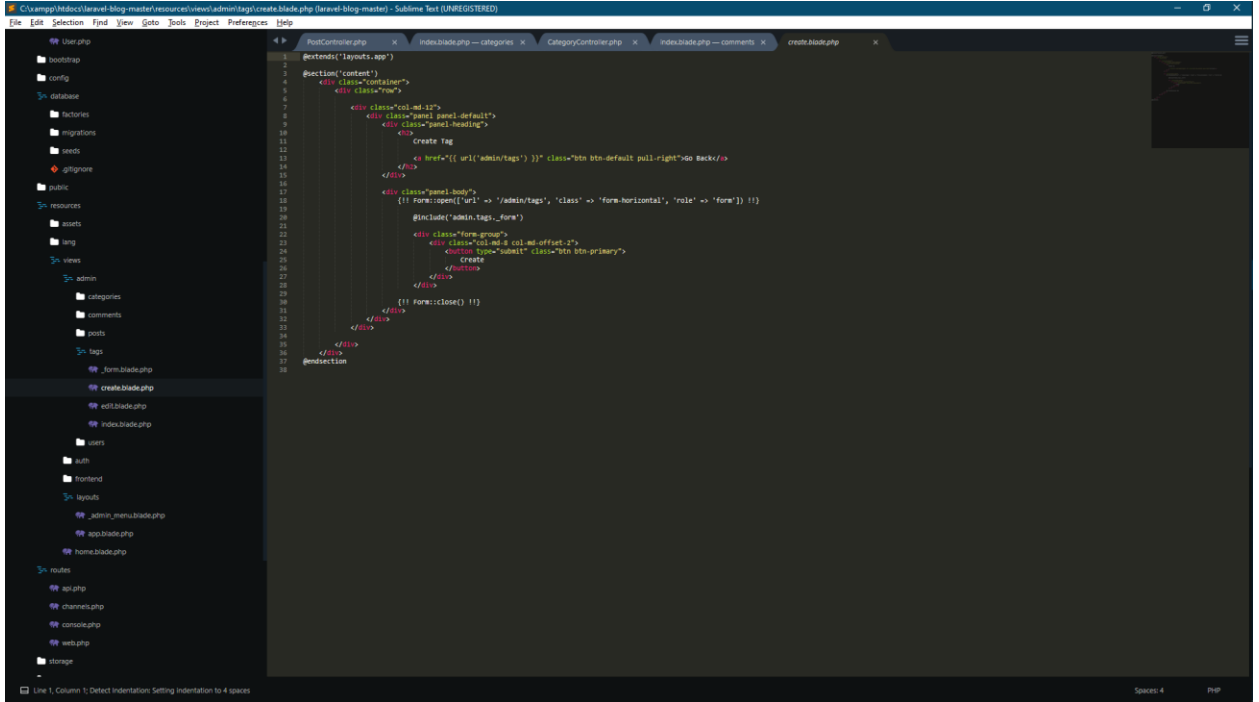
Index.blade.php

```

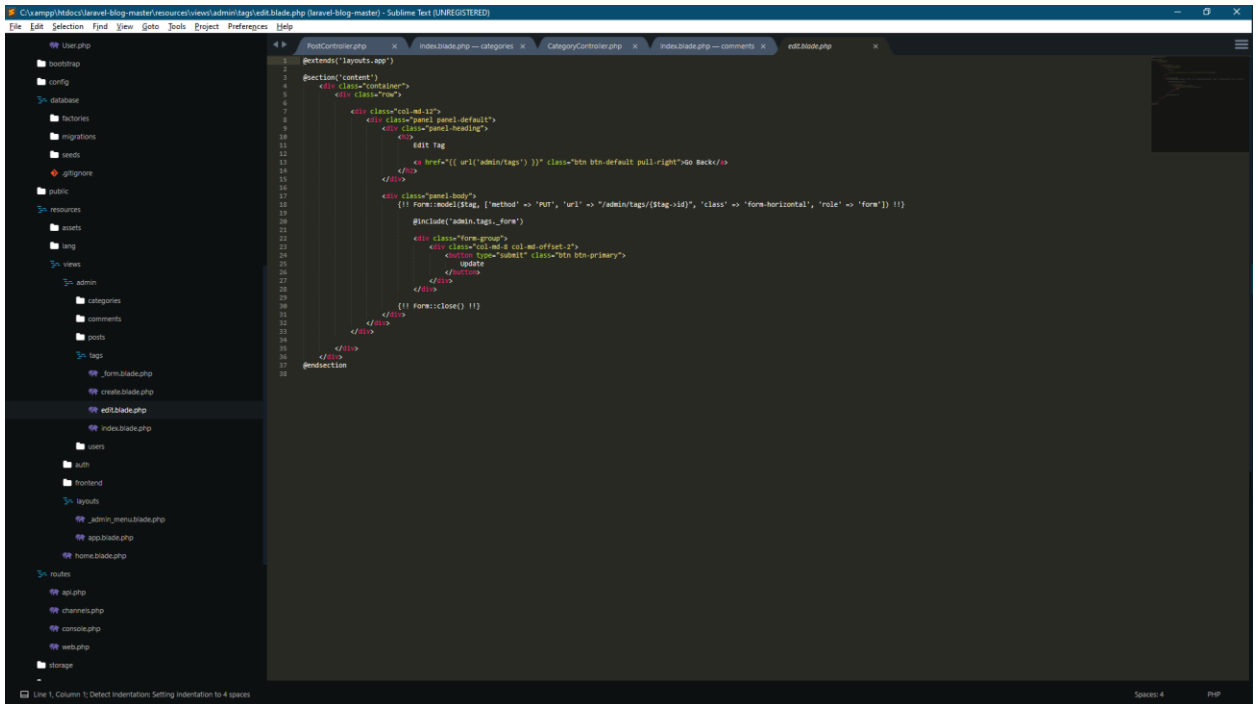
1 @extends('layouts.app')
2
3 @section('content')
4     <div class="container">
5         <div class="row">
6
7             <div class="col-md-12">
8                 <div class="panel panel-default">
9                     <div class="panel-heading">
10                         Tags
11                     </div>
12                     <div class="panel-body">
13                         <div class="table">
14                             <table>
15                                 <thead>
16                                     <tr>
17                                         <th>Name</th>
18                                         <th>Action</th>
19                                     </tr>
20                                 </thead>
21                                 <tbody>
22                                     @foreach ($tags as $tag)
23                                         <tr>
24                                             <td>{{ $tag->name }}</td>
25                                             <td>
26                                                 <a href="{{ url('/admin/tags/edit/' . $tag->id) }}" class="btn btn-xs btn-info">Edit</a>
27                                                 <a href="{{ url('/admin/tags/' . $tag->id) }}" class="btn btn-xs btn-danger" data-confirm="Are you sure?" data-method="DELETE" data-token="{{ csrf_token() }}">Delete</a>
28                                             </td>
29                                         </tr>
30                                     </tbody>
31                                 </table>
32                             </div>
33                             @if ($tags->isEmpty())
34                                 <div class="text-center">No tag available.</div>
35                             @endif
36                         </div>
37                     </div>
38                 </div>
39             </div>
40         </div>
41     </div>
42 @endsection

```

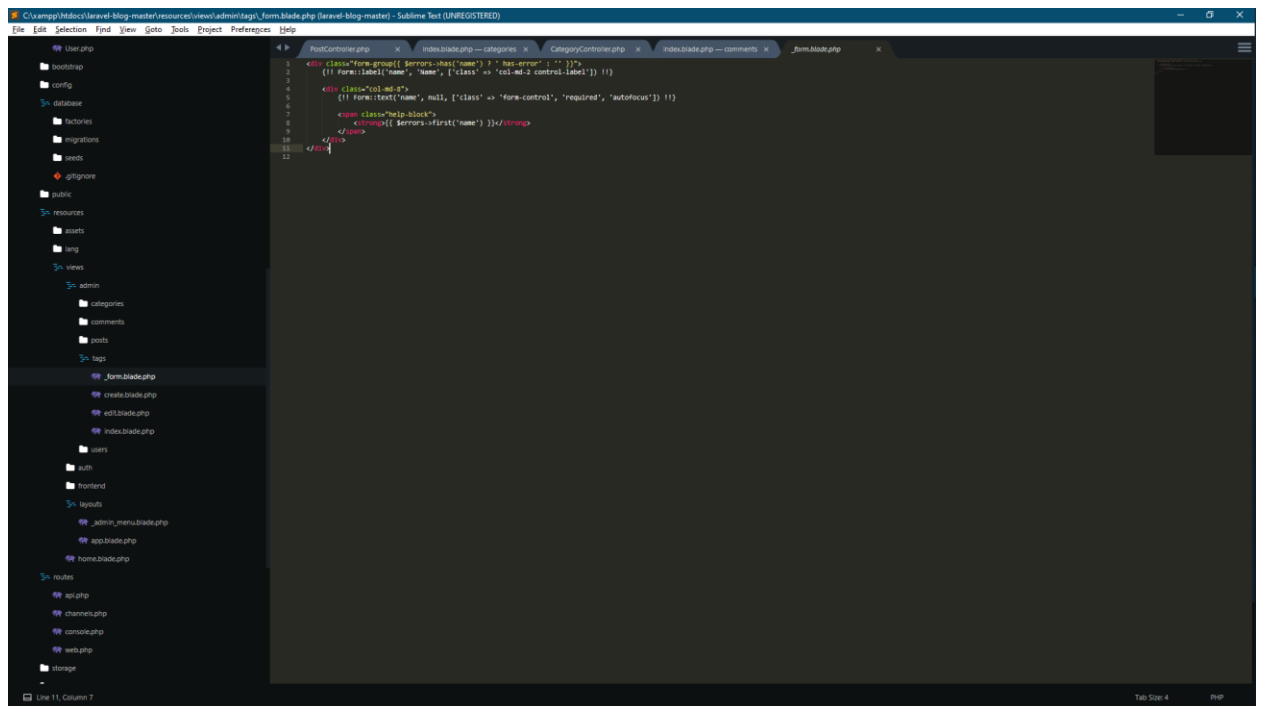
Create.blade.php



Edit.blade.php

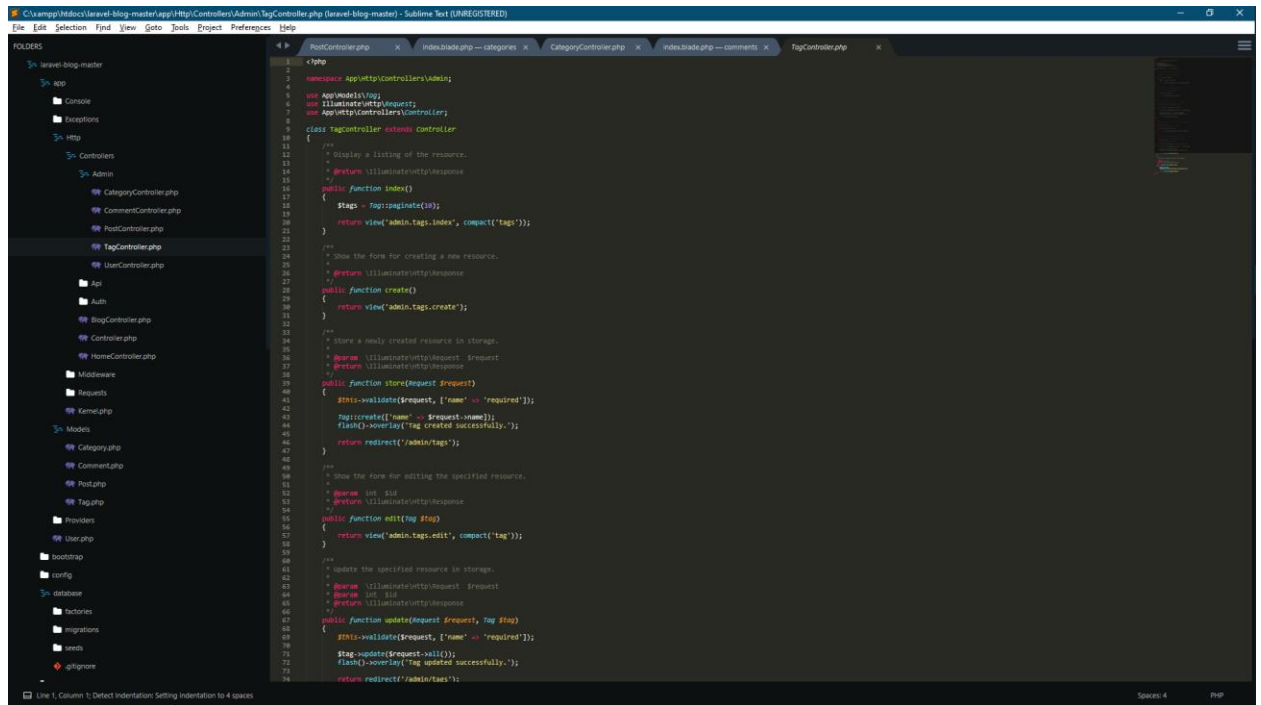


_form.blade.php

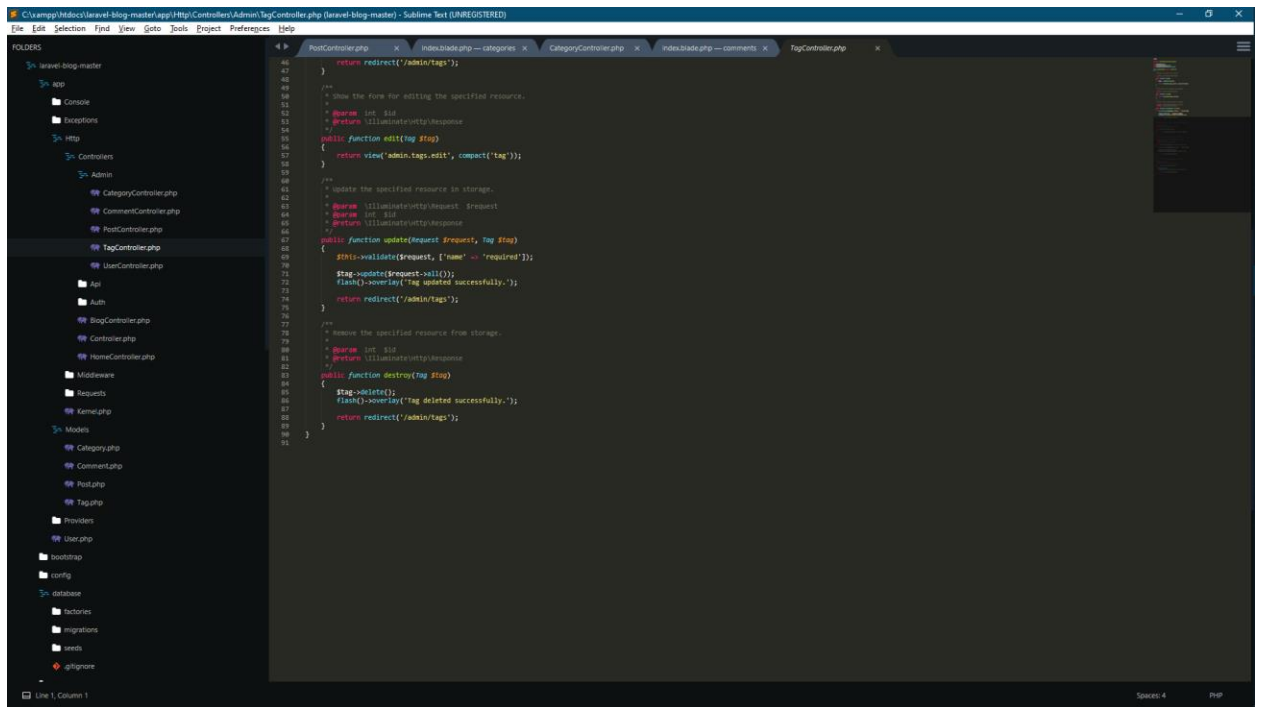


```
1 <div class="form-group{{ $errors->has('name') ? ' has-error' : '' }}">
2   <!-- Form:label('name', 'name', ['class' => 'col-md-2 control-label']) -->
3
4   <div class="col-md-8">
5     <!-- Form:text('name', null, ['class' => 'form-control', 'required', 'autofocus']) -->
6     <div class="help-block">
7       <strong{{ $errors->first('name') }}</strong>
8     </div>
9     <input type="text" class="form-control" value="{{ old('name') }}" />
10   </div>
11 </div>
```

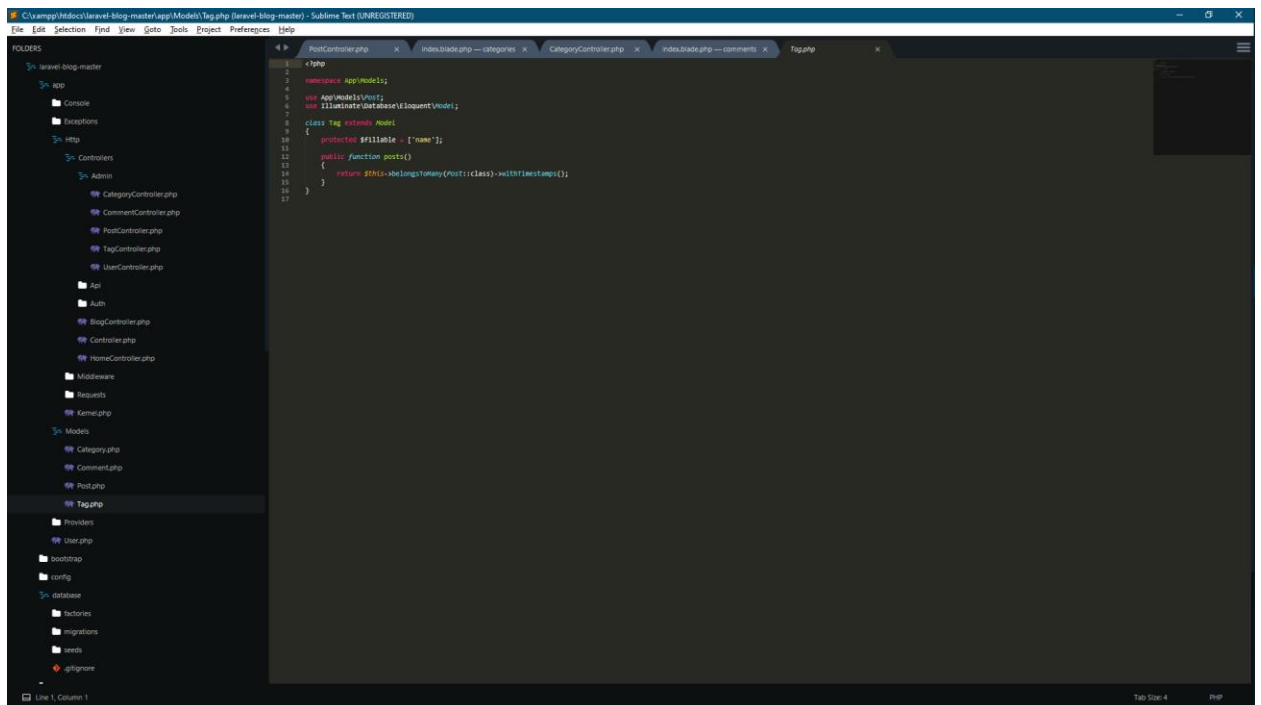
TagController



```
1 <?php
2 namespace App\Http\Controllers\Admin;
3 use App\Models\Tag;
4 use Illuminate\Http\Request;
5 use App\Http\Controllers\Controller;
6 class TagController extends Controller
7 {
8     /**
9      * Display a listing of the resource.
10     */
11     public function index()
12     {
13         $tags = Tag::paginate(10);
14         return view('admin.tags.index', compact('tags'));
15     }
16
17     /**
18      * Show the form for creating a new resource.
19     */
20     public function create()
21     {
22         return view('admin.tags.create');
23     }
24
25     /**
26      * Store a newly created resource in storage.
27     */
28     public function store(Request $request)
29     {
30         $this->validate($request, ['name' => 'required']);
31         Tag::create(['name' => $request->name]);
32         flash()->overlay('Tag created successfully.')->toRoute();
33         return redirect('/admin/tags');
34     }
35
36     /**
37      * Show the form for editing the specified resource.
38     */
39     public function edit($tag $tag)
40     {
41         return view('admin.tags.edit', compact('tag'));
42     }
43
44     /**
45      * Update the specified resource in storage.
46     */
47     public function update(Request $request, Tag $tag)
48     {
49         $this->validate($request, ['name' => 'required']);
50         $tag->update($request->all());
51         flash()->overlay('Tag updated successfully.')->toRoute();
52         return redirect('/admin/tags');
53     }
54 }
```

Tag model



- Menu tags เมื่อกดจะเข้าสู่หน้า index ของ tags เป็นหน้าที่เป็นฟอร์มไว้ใช้แสดงข้อมูล tags ทั้งหมด
- ปุ่ม create now เป็นการส่งไปหน้าฟอร์มที่ใช้เพิ่ม tags โดยใช้ @include('admin.tags._form') เป็นฟอร์ม input และมี route {{ url('admin/ tags /create') }} โดยที่ route จะส่งค่าไปที่คอนโทรลเลอร์ tagcontroller ฟังก์ชัน create โดยฟังก์ชันจะสั่งทำการ return view('admin.tags.create'); ไปที่หน้า create ใน admin.category
- เมื่อทำการ input ค่าที่ต้องการ ให้กด create จะทำการเข้าสู่ route tags ไปยังคอนโทรลเลอร์ ชื่อว่า Tagcontroller ฟังก์ชัน store ที่ทำการ เก็บค่าที่ post มาไว้ใน \$request โดย \$request เป็นค่า Tag::create(['name' => \$request->name]); ถ้าสำเร็จแสดง Tag created successfully แล้วรีเทิร์นค่าไป /admin/tags ที่เป็นคอนโทรลเลอร์เริ่มที่เมื่อกดเข้ามาจะทำคำสั่งให้แสดงหน้า show
- ปุ่ม Edit จะใช้ @include('admin.tags._form') เป็นฟอร์ม input เหมือนเดิม จะเป็นการส่งค่า \$tags->id ไปที่ route tags ส่งค่าส่งไปที่คอนโทรลเลอร์ฟังก์ชัน edit ของ Tagtcontroller โดยคำสั่งจะทำการเรียกหน้า ฟอร์มพร้อมทั้งใส่ข้อมูลมาให้แล้ว
- เมื่อแก้ไขกดปุ่มแก้ไขจะทำการนำ \$tags->update(\$request->all()); ที่มีคำสั่งโมเดล tag ที่ชื่อ update ในการส่งใน อัปเดต \$request ที่ส่งมาลงฐานข้อมูล
- Delete จะทำการส่งค่า \$tags->id โดย route ไปที่ Tagcontroller ฟังก์ชัน destroy เช็คว่าสแตตัสเป็น admin เท่านั้น ถึงจะลบได้ โดยทำการลบโดยใช้ db ที่ชื่อ delete เมื่อลบสำเร็จแสดง Tag deleted successfully รีเทิร์นค่าไปหน้า show

Index.blade.php

```

1 @extends('layouts.app')
2 @section('content')
3     <div class="container">
4         <div class="row">
5             <div class="col-md-4">
6                 <div class="panel panel-default">
7                     <div class="panel-heading">
8                         Users
9                     </div>
10                    <div class="panel-body">
11                        <table class="table">
12                            <thead>
13                                <tr>
14                                    <th>Name</th>
15                                    <th>Email</th>
16                                    <th>Delete</th>
17                                </tr>
18                            </thead>
19                            <tbody>
20                                @foreach ($users as $user)
21                                    <tr>
22                                        <td>{{ $user->name }}</td>
23                                        <td>{{ $user->email }}</td>
24                                        <td>
25                                            <a href="#">Delete</a>
26                                        </td>
27                                    </tr>
28                                </tbody>
29                            </table>
30                        </div>
31                    </div>
32                </div>
33            </div>
34            <div class="col-md-8">
35                <div class="panel panel-default">
36                    <div class="panel-heading">
37                        Posts
38                    </div>
39                    <div class="panel-body">
40                        <table class="table">
41                            <thead>
42                                <tr>
43                                    <th>Name</th>
44                                    <th>Email</th>
45                                    <th>Delete</th>
46                                </tr>
47                            </thead>
48                            <tbody>
49                                @foreach ($posts as $post)
50                                    <tr>
51                                        <td>{{ $post->name }}</td>
52                                        <td>{{ $post->email }}</td>
53                                        <td>
54                                            <a href="#">Delete</a>
55                                        </td>
56                                    </tr>
57                                </tbody>
58                            </table>
59                        </div>
60                    </div>
61                </div>
62            </div>
63        </div>
64    </section>

```

- Menu user เป็นของ admin เท่านั้นเมื่อกดจะเข้าสู่หน้า index ของ user โดยผ่าน route ที่ชื่อ '/users', 'UserController' โดยเรียกใช้ฟังก์ชัน index เป็นหน้าที่เป็นฟอร์มไว้ใช้แสดงข้อมูล user ทั้งหมด
- Delete จะทำการส่งค่า \$users->id โดย route ไปที่ Usercontroller ฟังก์ชัน destroy เช็ควาสเตตัสเป็น admin เท่านั้นถึงจะลบได้ โดยทำการลบโดยใช้ db ที่ชื่อ delete เมื่อลบสำเร็จแสดง User deleted successfully รีเทิร์นค่าไปหน้า index