

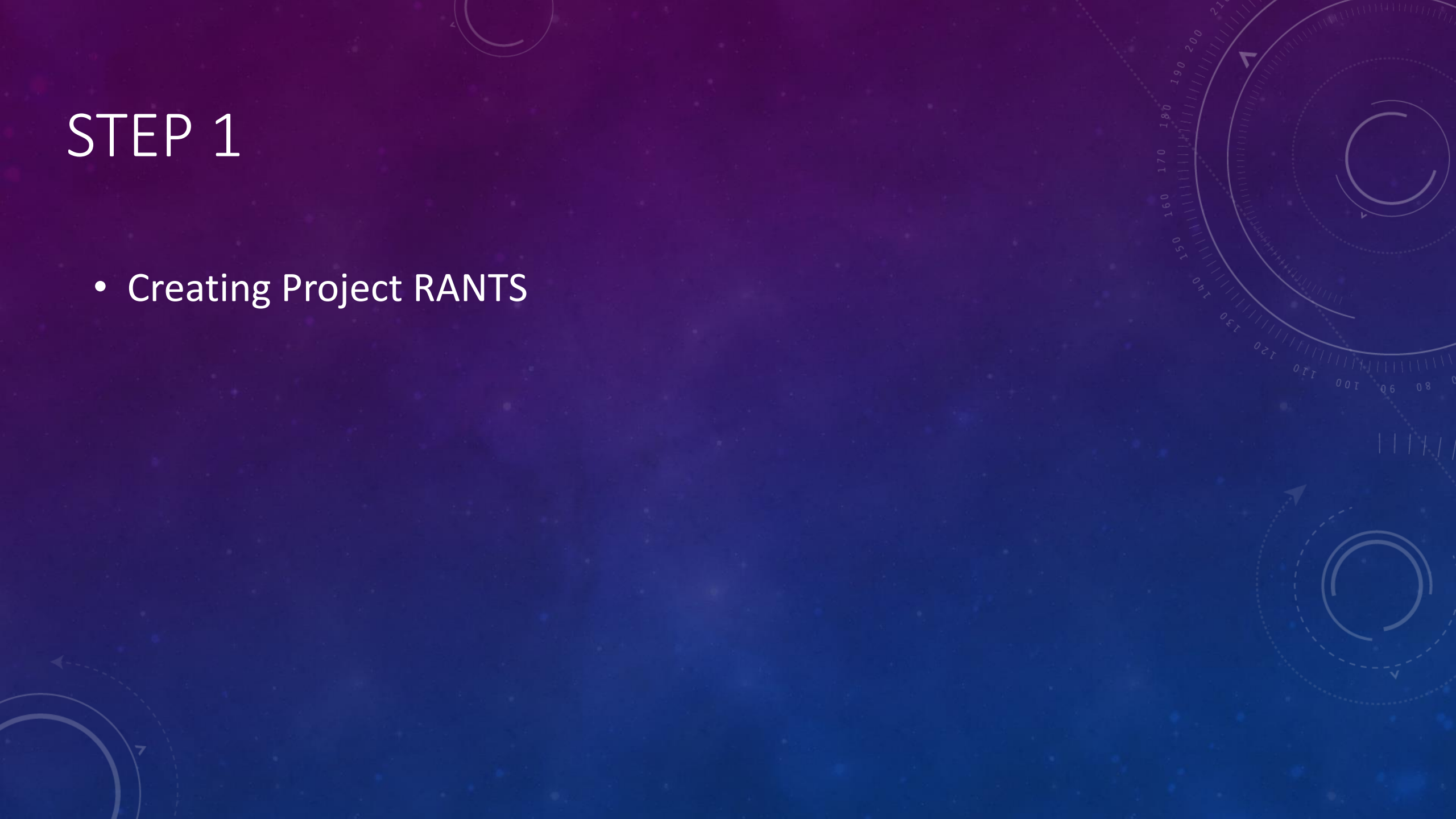
The background features a dark blue gradient with faint, light blue geometric patterns. On the left side, there are several concentric circles and arcs, some of which are marked with degree values ranging from 40 to 260. These markings are arranged in a way that suggests a circular scale or a compass rose. The overall aesthetic is technical and modern.

# RANTS TUTORIAL

LAB 1, CMSC 129

# STEP 1

- Creating Project RANTS

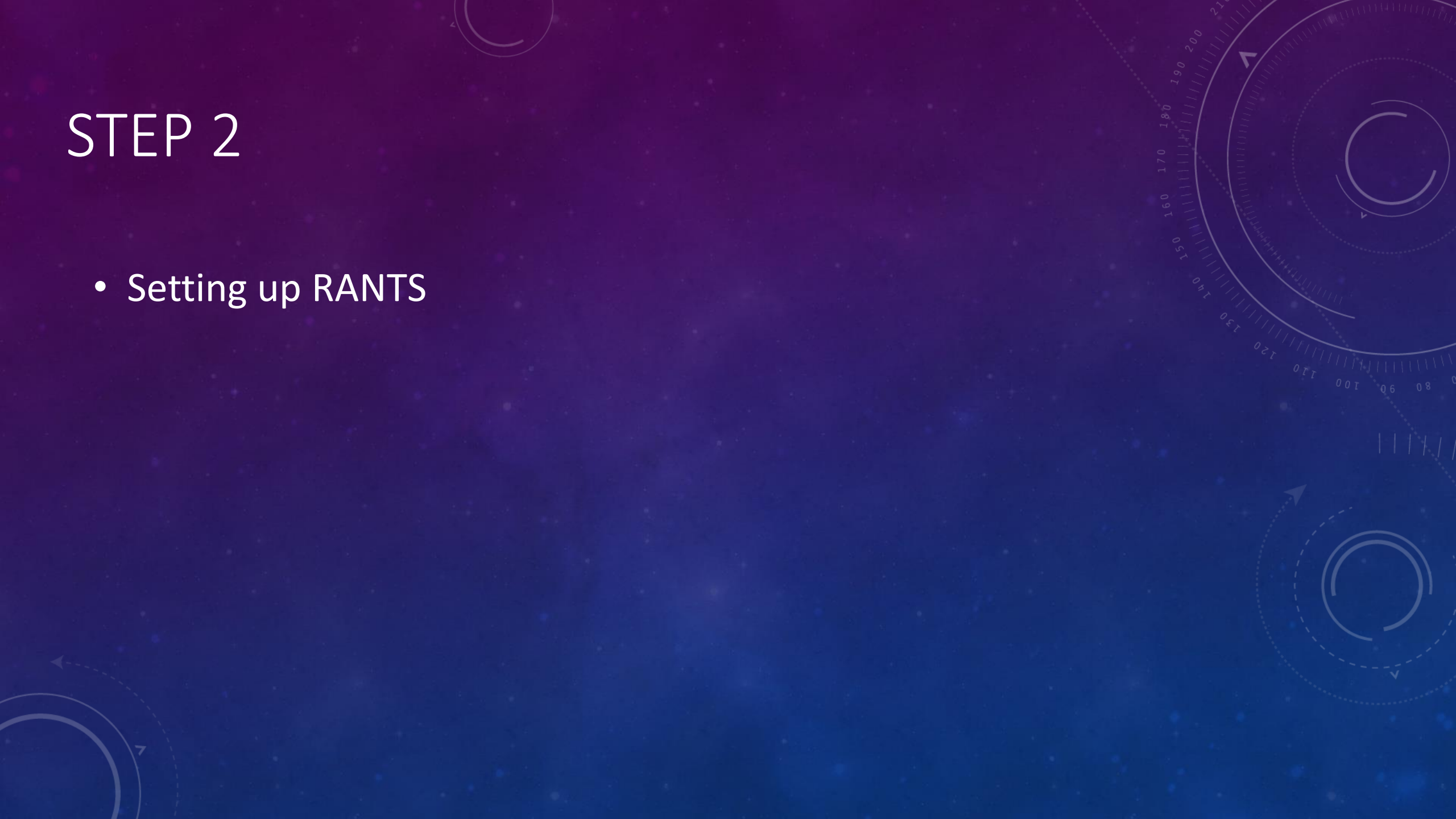


# STEP 1

- Open CMD and go to this directory :
  - ..XAMPP/htdocs
- Type in : `composer create-project --prefer-dist laravel/laravel rants`

# STEP 2

- Setting up RANTS



# STEP 2

Creating a database:

- Open a web browser and type in “localhost/phpmyadmin” as address (make sure your xampp server is on)
- Create your prescribed database in phpmyadmin



# STEP 2

Server: 127.0.0.1

Databases SQL Status User accounts Export Import

## Databases

Create database ?

rants Collation Create

	Database	Collation	Action
<input type="checkbox"/>	application	latin1_swedish_ci	Check privileges
<input type="checkbox"/>	directory	latin1_swedish_ci	Check privileges
<input type="checkbox"/>	enrollment	latin1_swedish_ci	Check privileges
<input type="checkbox"/>	information_schema	utf8_general_ci	Check privileges
<input type="checkbox"/>	instapics	latin1_swedish_ci	Check privileges

# CONNECT THE DATABASE TO OUR PROJECT

- Open folder rants
- Open the .env file in C://xampp/htdocs/
- Change the DB\_DATABASE entry to your created database name instead of the default “homestead”
- Also, Change the DB\_USERNAME to root and DB\_PASSWORD to blank
- Overwrite the file

C:\xampp\htdocs\rants\.env (rants) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

OPEN FILES

× untitled

× .env

FOLDERS

▼ rants

▶ app

▶ bootstrap

▶ config

▶ database

▶ public

▶ resources

▶ routes

▶ storage

▶ tests

▶ vendor

• .env

• .env.example

• .gitattributes

• .gitignore

untitled

.env

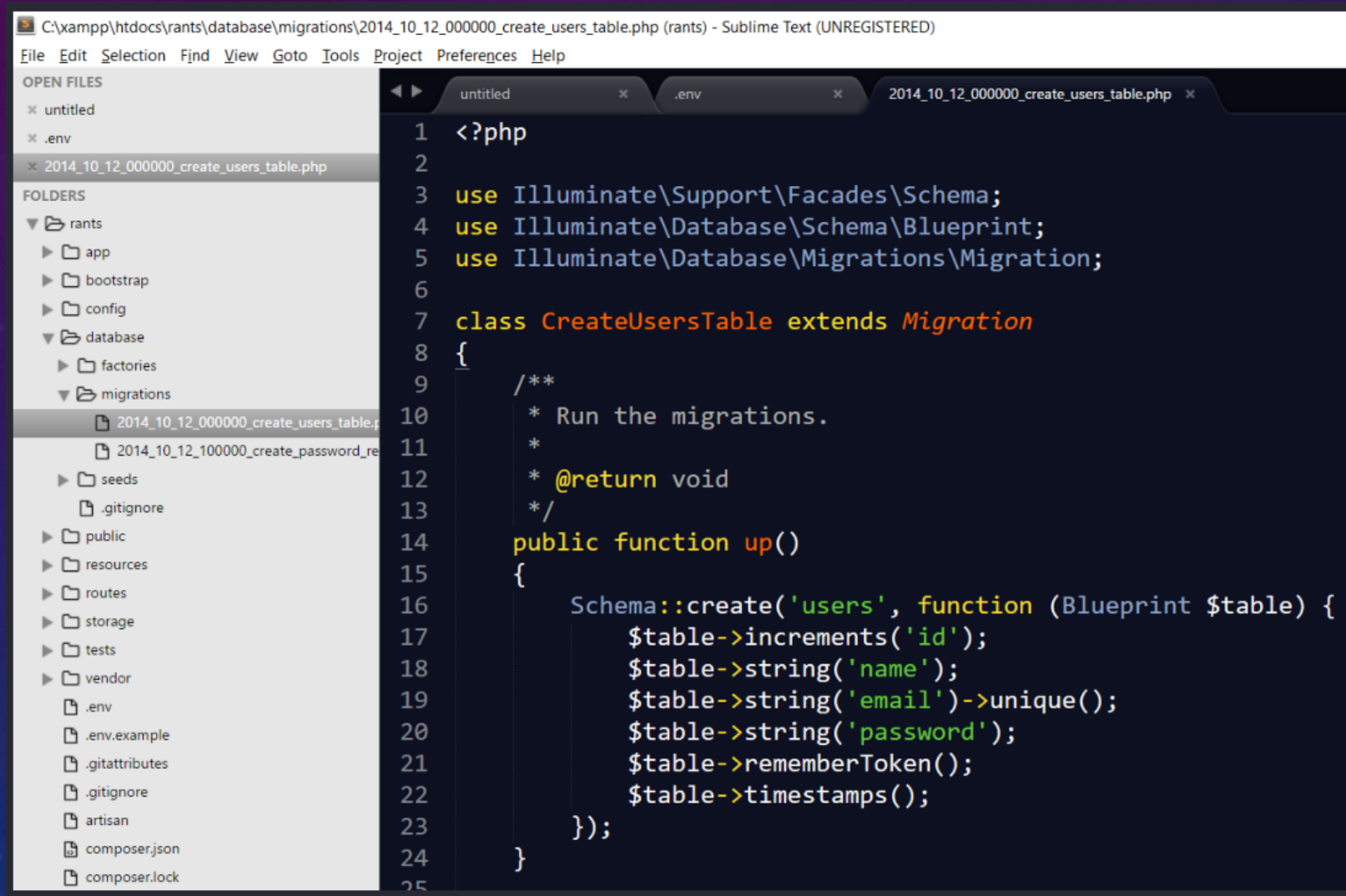
```
1 APP_ENV=local
2 APP_KEY=base64:5JAAi01ZNN09/C890HV3Mp65CPwzjXEC/1I7J8PuImU=
3 APP_DEBUG=true
4 APP_LOG_LEVEL=debug
5 APP_URL=http://localhost
6
7 DB_CONNECTION=mysql
8 DB_HOST=localhost
9 DB_PORT=3306
10 DB_DATABASE=rants
11 DB_USERNAME=root
12 DB_PASSWORD=
13
14 BROADCAST_DRIVER=log
15 CACHE_DRIVER=file
```



# SETTING UP ACCOUNTS

- Creating a table for accounts
  - Open create\_users\_table.php file in:  
C://xampp/htdocs/rants/database/migrations

# CONVERT THIS CODE..



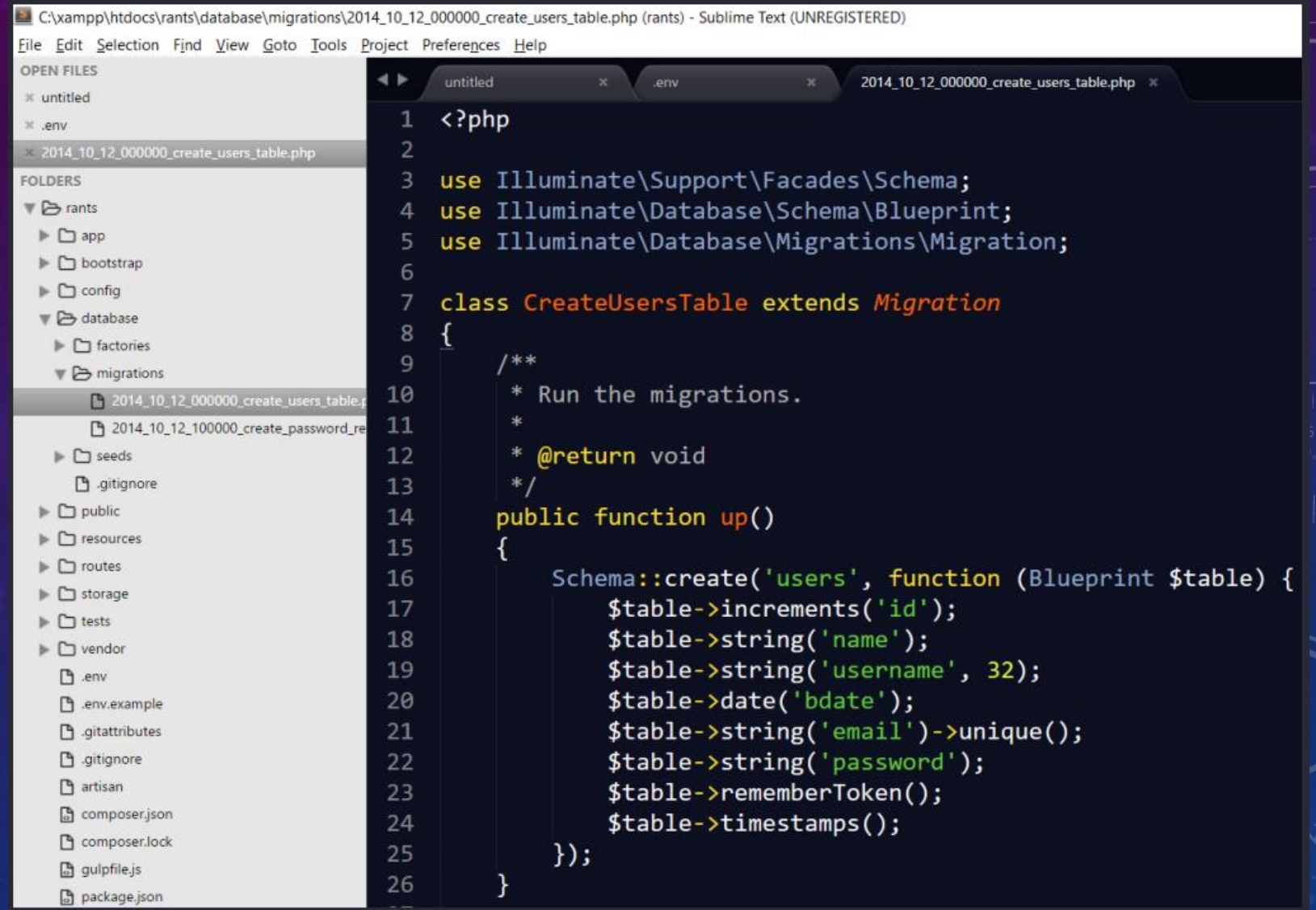
```
C:\xampp\htdocs\rants\database\migrations\2014_10_12_000000_create_users_table.php (rants) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

OPEN FILES
x untitled
x .env
x 2014_10_12_000000_create_users_table.php

FOLDERS
v rants
  > app
  > bootstrap
  > config
  v database
    > factories
    v migrations
      2014_10_12_000000_create_users_table.php
      2014_10_12_100000_create_password_resets_table.php
    > seeds
    .gitignore
  > public
  > resources
  > routes
  > storage
  > tests
  > vendor
  .env
  .env.example
  .gitattributes
  .gitignore
  artisan
  composer.json
  composer.lock

1 <?php
2
3 use Illuminate\Support\Facades\Schema;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Database\Migrations\Migration;
6
7 class CreateUsersTable extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('users', function (Blueprint $table) {
17             $table->increments('id');
18             $table->string('name');
19             $table->string('email')->unique();
20             $table->string('password');
21             $table->rememberToken();
22             $table->timestamps();
23         });
24     }
25 }
```

# TO THIS CODE..



```
C:\xampp\htdocs\rants\database\migrations\2014_10_12_000000_create_users_table.php (rants) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

OPEN FILES
* untitled
* .env
* 2014_10_12_000000_create_users_table.php

FOLDERS
▼ rants
  ► app
  ► bootstrap
  ► config
  ▼ database
    ► factories
    ▼ migrations
      2014_10_12_000000_create_users_table.php
      2014_10_12_100000_create_password_resets_table.php
    ► seeds
    .gitignore
  ► public
  ► resources
  ► routes
  ► storage
  ► tests
  ► vendor
  .env
  .env.example
  .gitattributes
  .gitignore
  artisan
  composer.json
  composer.lock
  gulpfile.js
  package.json

1 <?php
2
3 use Illuminate\Support\Facades\Schema;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Database\Migrations\Migration;
6
7 class CreateUsersTable extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('users', function (Blueprint $table) {
17             $table->increments('id');
18             $table->string('name');
19             $table->string('username', 32);
20             $table->date('bdate');
21             $table->string('email')->unique();
22             $table->string('password');
23             $table->rememberToken();
24             $table->timestamps();
25         });
26     }
27 }
```

# SETTING UP ACCOUNTS

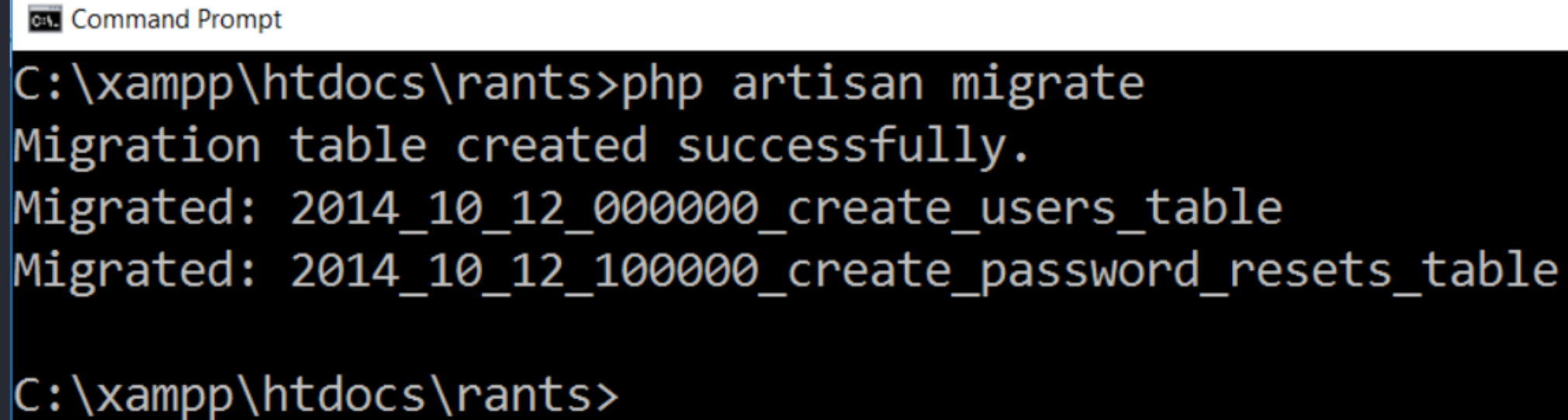
Now run a migration to import table contents:

- In cmd (Make sure you're in the right directory), type  
in:

```
php artisan migrate
```



# RUNNING THE COMMAND “PHP ARTISAN MIGRATE” WILL LEAD YOU TO THIS:

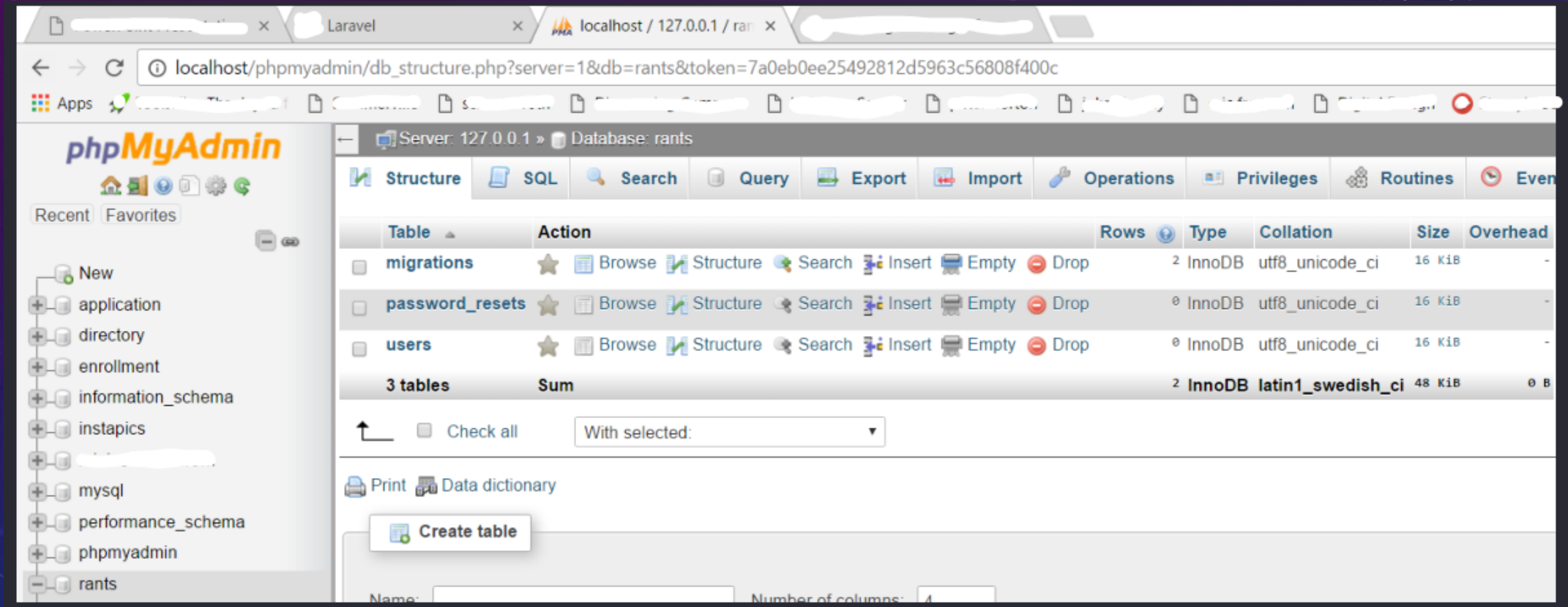


```
Command Prompt
C:\xampp\htdocs\rants>php artisan migrate
Migration table created successfully.
Migrated: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_100000_create_password_resets_table

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



# GO TO PHPMYADMIN



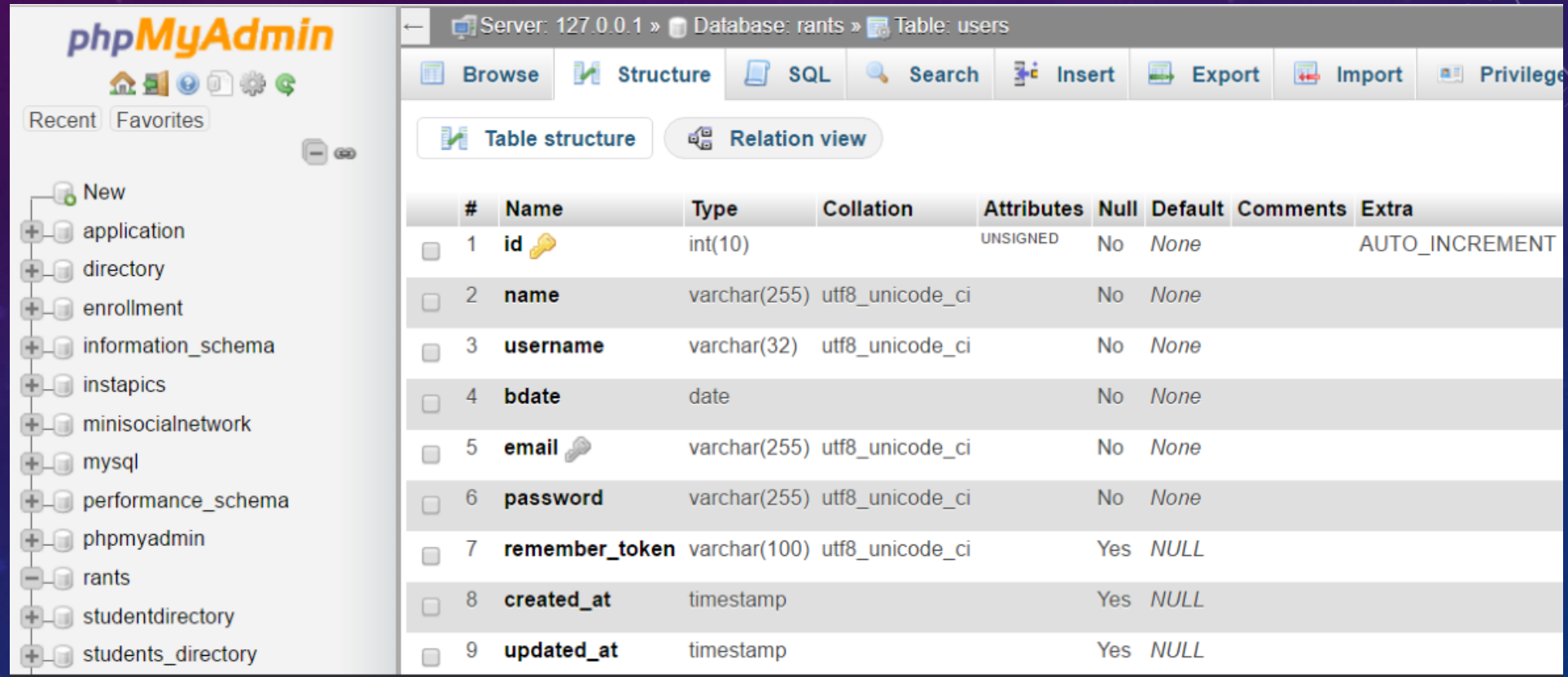
The screenshot displays the phpMyAdmin interface in a web browser. The address bar shows the URL: `localhost/phpmyadmin/db_structure.php?server=1&db=rants&token=7a0eb0ee25492812d5963c56808f400c`. The interface is for the 'rants' database on a server at 127.0.0.1.

The left sidebar shows a tree view of databases, with 'rants' selected. The main panel shows the 'Structure' tab for the 'rants' database. It lists three tables: 'migrations', 'password\_resets', and 'users'. Each table has a star icon, a 'Browse' button, and links for 'Structure', 'Search', 'Insert', 'Empty', and 'Drop'. The table 'migrations' has 2 rows, while 'password\_resets' and 'users' have 0 rows. The table 'users' is highlighted.

Table	Action	Rows	Type	Collation	Size	Overhead
migrations	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8_unicode_ci	16 KiB	-
password_resets	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_unicode_ci	16 KiB	-
users	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_unicode_ci	16 KiB	-
3 tables	Sum	2	InnoDB	latin1_swedish_ci	48 KiB	0 B

Below the table list, there is a 'Check all' checkbox and a 'With selected:' dropdown menu. At the bottom, there is a 'Create table' button and a form with 'Name:' and 'Number of columns:' fields.

# INSIDE THE USERS TABLE...



The screenshot shows the phpMyAdmin interface. On the left is a sidebar with a tree view of databases, including 'rants'. The main panel displays the 'Table structure' for the 'users' table. The table has 9 columns: id, name, username, bdate, email, password, remember\_token, created\_at, and updated\_at. The 'id' column is the primary key and is auto-incrementing. The 'email' column is also marked as a unique key. The 'remember\_token' column is nullable. The 'created\_at' and 'updated\_at' columns are timestamps and are nullable.

Server: 127.0.0.1 » Database: rants » Table: users

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(10)		UNSIGNED	No	None		AUTO_INCREMENT
2	name	varchar(255)	utf8_unicode_ci		No	None		
3	username	varchar(32)	utf8_unicode_ci		No	None		
4	bdate	date			No	None		
5	email	varchar(255)	utf8_unicode_ci		No	None		
6	password	varchar(255)	utf8_unicode_ci		No	None		
7	remember_token	varchar(100)	utf8_unicode_ci		Yes	NULL		
8	created_at	timestamp			Yes	NULL		
9	updated_at	timestamp			Yes	NULL		

# SETTING UP ACCOUNTS

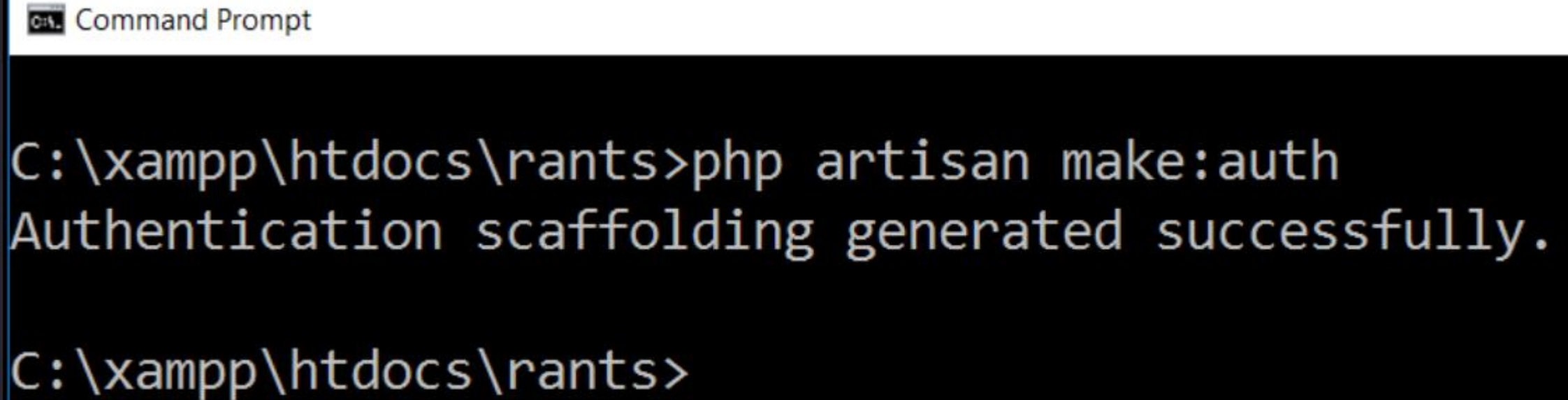
## Make an authentication

- In cmd (Make sure you're in the right directory) type in:

```
php artisan make:auth
```

- This code auto-creates a login and register page

# AFTER TYPING THE COMMAND..

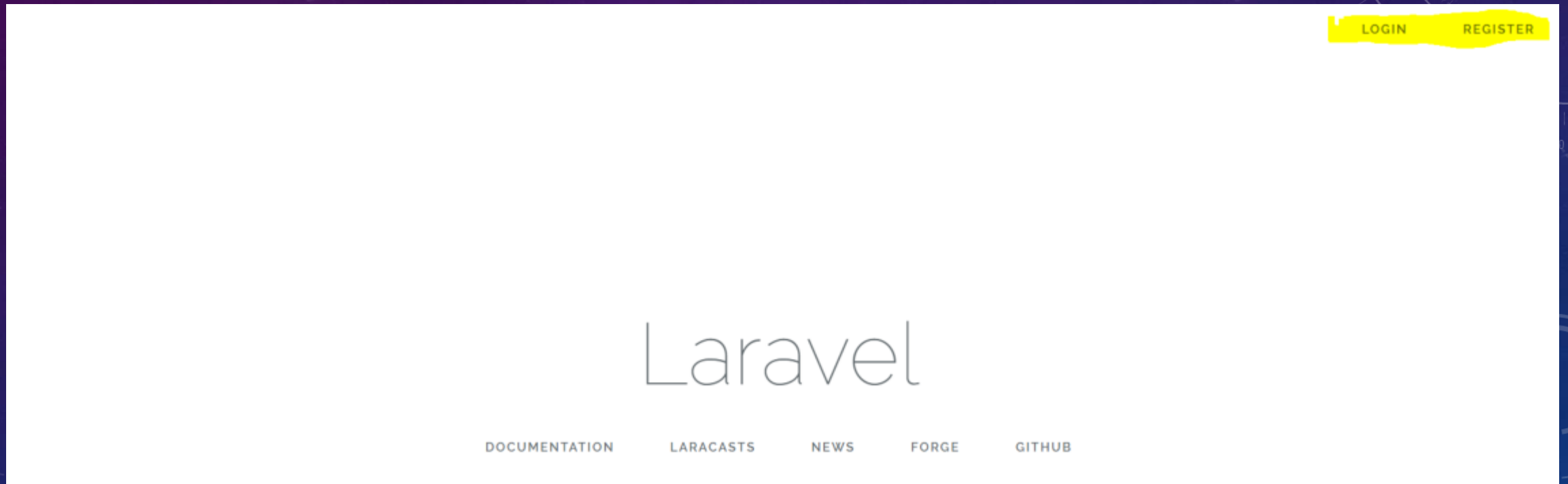


The screenshot shows a Windows Command Prompt window with a white title bar that reads "C:\> Command Prompt". The main area has a black background with white text. The first line shows the command `C:\xampp\htdocs\rants>php artisan make:auth` being entered. The second line shows the output: `Authentication scaffolding generated successfully.`. The third line shows the prompt `C:\xampp\htdocs\rants>` again, indicating the command has finished.

```
C:\xampp\htdocs\rants>php artisan make:auth
Authentication scaffolding generated successfully.

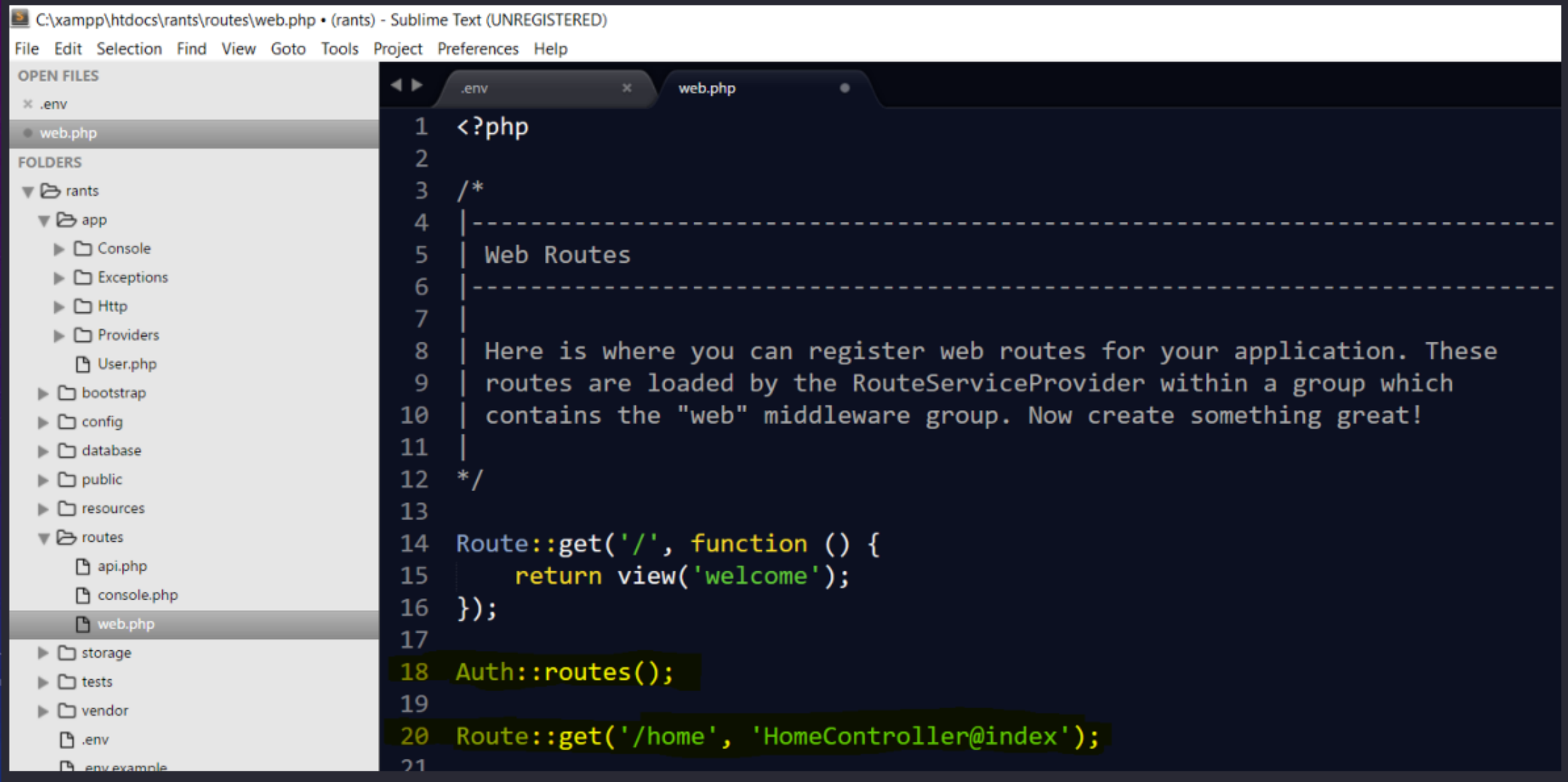
C:\xampp\htdocs\rants>
```

# RUNNING PHP ARTISAN SERVE





# IN OUR MAIN ROUTE FILE..



```
C:\xampp\htdocs\rants\routes\web.php • (rants) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

OPEN FILES
x .env
web.php

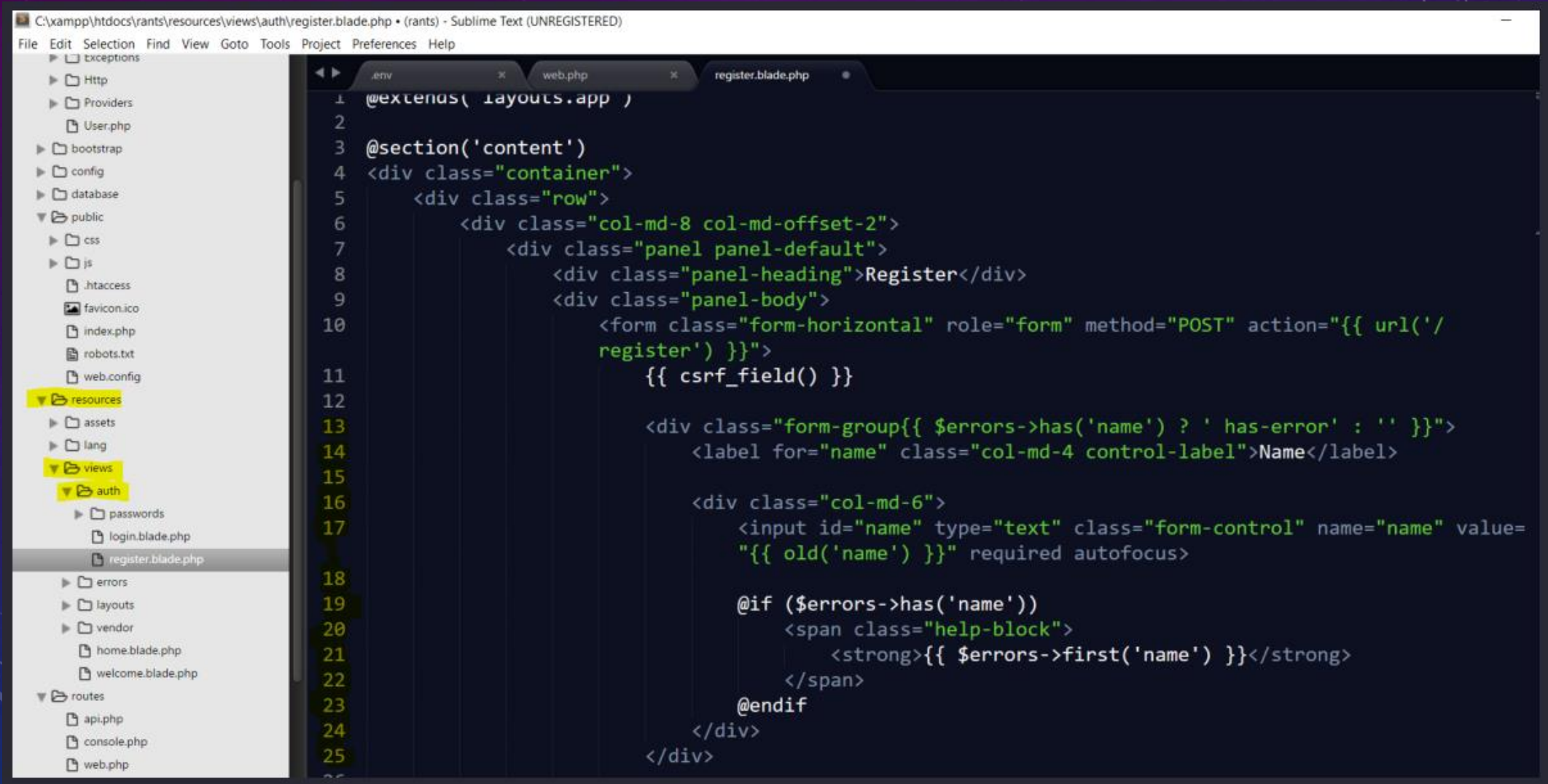
FOLDERS
▼ rants
  ▼ app
    ► Console
    ► Exceptions
    ► Http
    ► Providers
      User.php
  ► bootstrap
  ► config
  ► database
  ► public
  ► resources
  ▼ routes
    api.php
    console.php
    web.php
  ► storage
  ► tests
  ► vendor
  .env
  env.example

1 <?php
2
3 /*
4 |-----
5 | Web Routes
6 |-----
7 |
8 | Here is where you can register web routes for your application. These
9 | routes are loaded by the RouteServiceProvider within a group which
10 | contains the "web" middleware group. Now create something great!
11 |
12 */
13
14 Route::get('/', function () {
15     return view('welcome');
16 });
17
18 Auth::routes();
19
20 Route::get('/home', 'HomeController@index');
21
```

# TWEAKING REGISTER PAGE

`C://xampp/htdocs/rants/resources/views/auth/register.blade.php`

# COPY THESE LINES OF CODE..



```
C:\xampp\htdocs\rants\resources\views\auth\register.blade.php • (rants) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

.env web.php register.blade.php
1 @extends('layouts.app')
2
3 @section('content')
4 <div class="container">
5     <div class="row">
6         <div class="col-md-8 col-md-offset-2">
7             <div class="panel panel-default">
8                 <div class="panel-heading">Register</div>
9                 <div class="panel-body">
10                     <form class="form-horizontal" role="form" method="POST" action="{{ url('/register') }}">
11                         {{ csrf_field() }}
12
13                     <div class="form-group{{ $errors->has('name') ? ' has-error' : '' }}">
14                         <label for="name" class="col-md-4 control-label">Name</label>
15
16                         <div class="col-md-6">
17                             <input id="name" type="text" class="form-control" name="name" value=
18                                 "{{ old('name') }}" required autofocus>
19
20                             @if ($errors->has('name'))
21                                 <span class="help-block">
22                                     <strong>{{ $errors->first('name') }}</strong>
23                                 </span>
24                             @endif
25                         </div>
26                     </div>
27                 </div>
28             </div>
29         </div>
30     </div>
31 </div>
32 @endsection
```

# REPLACE THE NAME WITH USERNAME

```
.env x web.php x register.blade.php x
26
27 <div class="form-group{{ $errors->has('username') ? ' has-error' : '' }}">
28   <label for="username" class="col-md-4 control-label">Username</label>
29
30   <div class="col-md-6">
31     <input id="username" type="text" class="form-control" name="username"
32       " value="{{ old('username') }}" required autofocus>
33
34     @if ($errors->has('username'))
35       <span class="help-block">
36         <strong>{{ $errors->first('username') }}</strong>
37       </span>
38     @endif
39   </div>
40 </div>
41
```

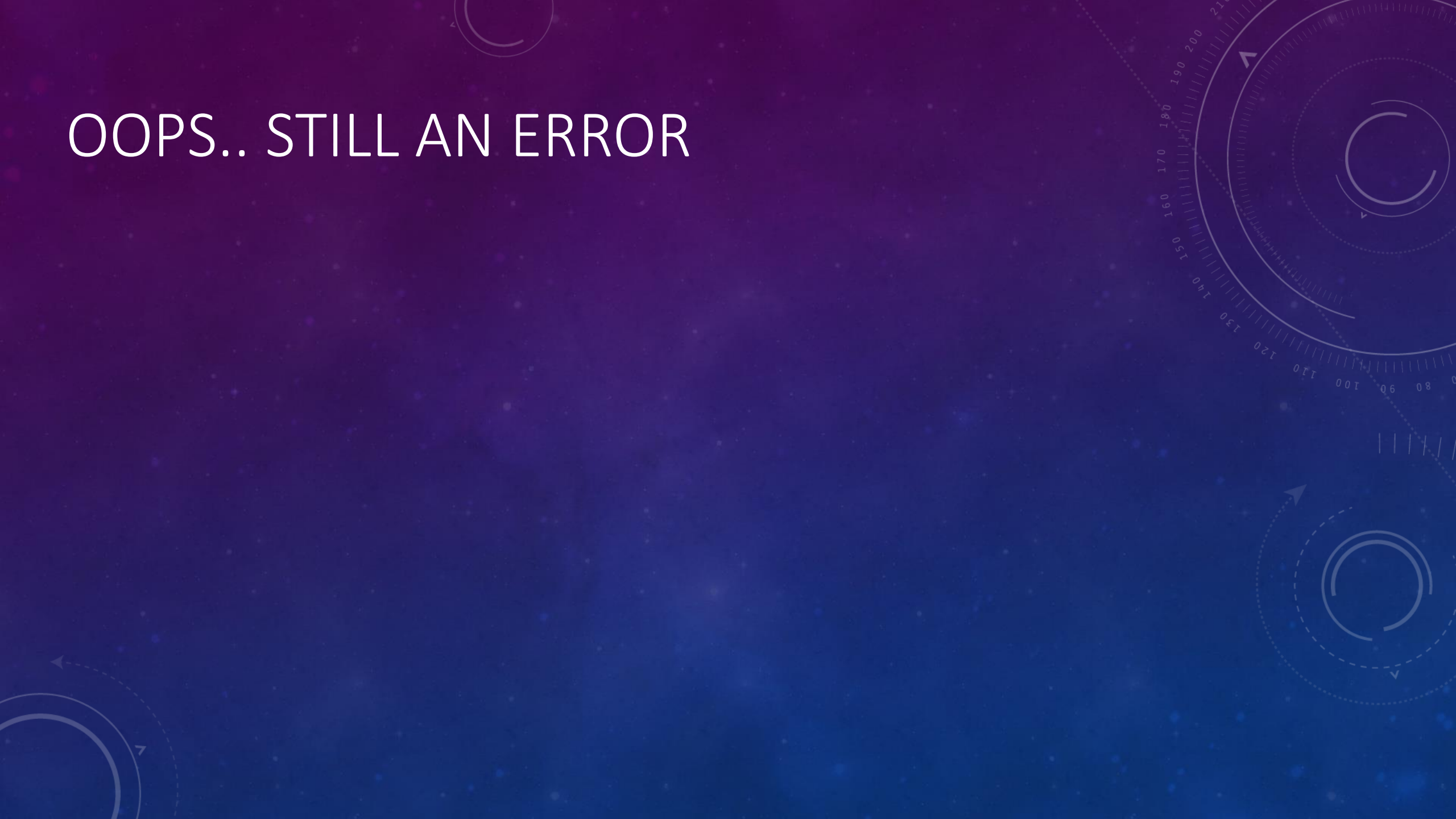


# DO THE SAME THING WITH BDATE

```
.env x web.php x register.blade.php x
41
42 <div class="form-group{{ $errors->has('bdate') ? ' has-error' : '' }}">
43   <label for="bdate" class="col-md-4 control-label">Birthday</label>
44
45   <div class="col-md-6">
46     <input id="bdate" type="text" class="form-control" name="bdate" value="{{
47       old('bdate') }}" required autofocus>
48
49     @if ($errors->has('bdate'))
50       <span class="help-block">
51         <strong>{{ $errors->first('bdate') }}</strong>
52       </span>
53     @endif
54   </div>
</div>
```



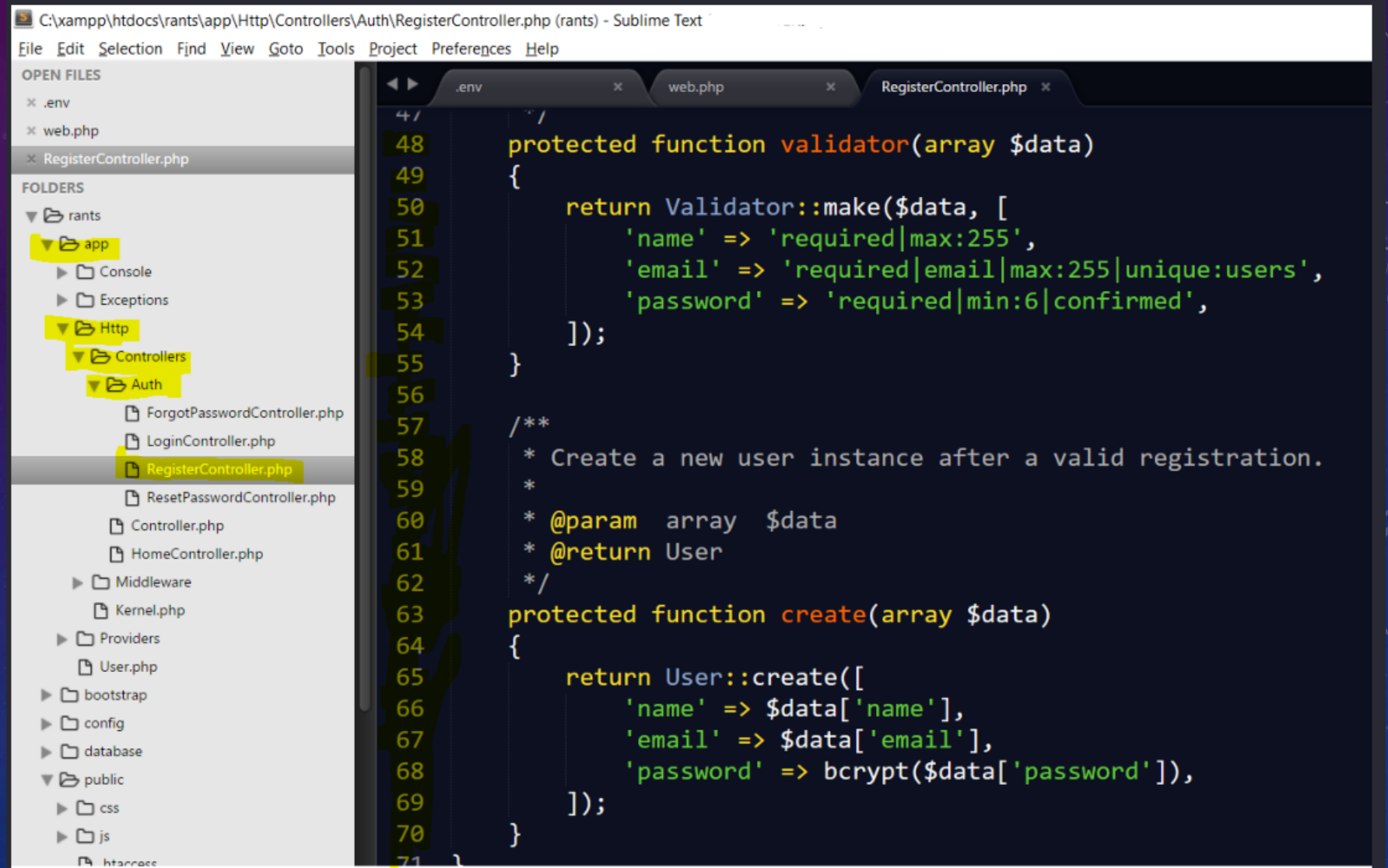
OOPS.. STILL AN ERROR



# FIXING REGISTER PROBLEMS

`C://xampp/htdocs/rants/app/Http/Controllers/Auth/RegisterController.php`

# CONVERT THIS CODE...



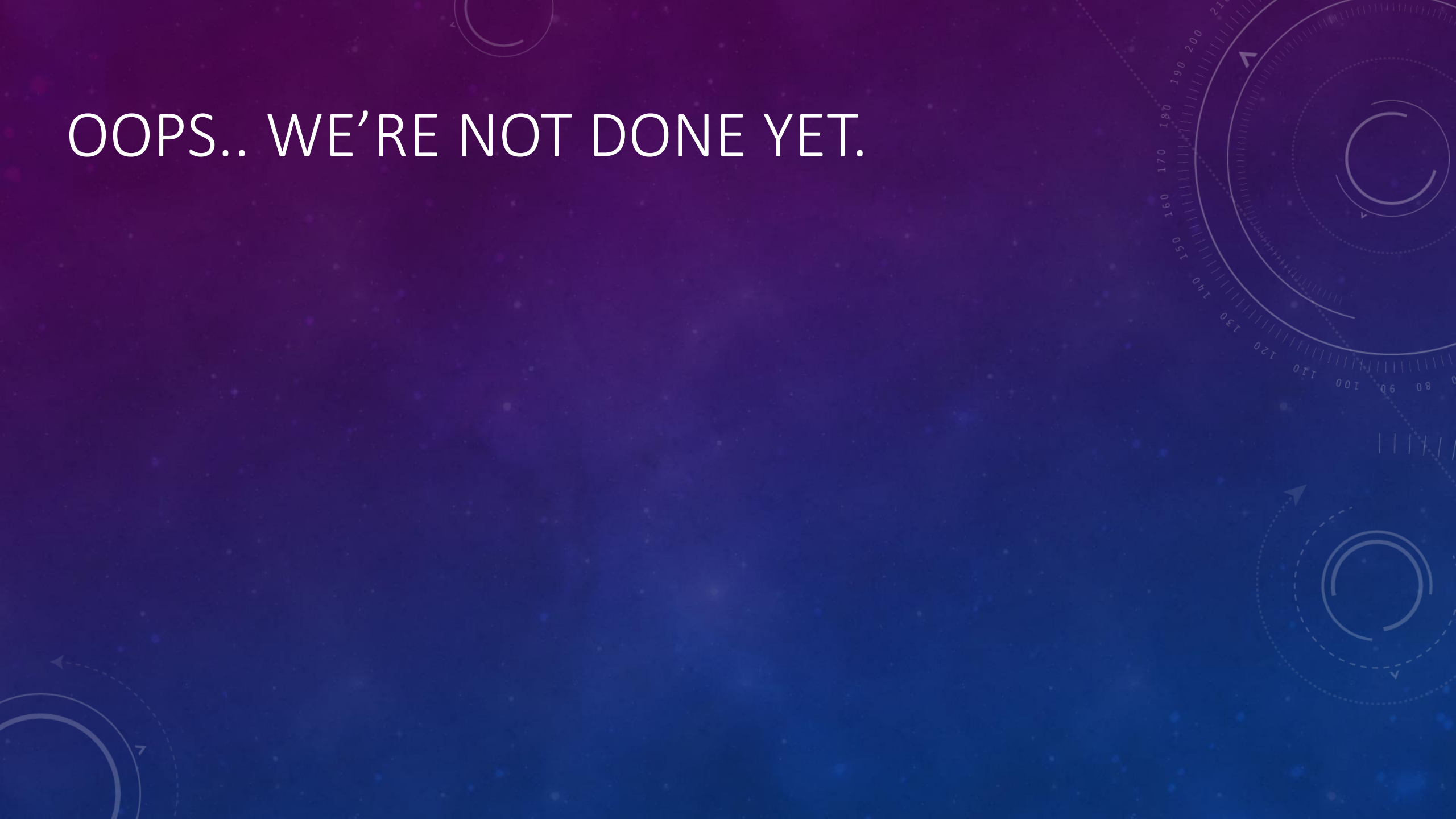
The screenshot shows a Sublime Text editor window with the title bar "C:\xampp\htdocs\rants\app\Http\Controllers\Auth\RegisterController.php (rants) - Sublime Text". The menu bar includes File, Edit, Selection, Find, View, Goto, Tools, Project, Preferences, and Help. On the left, the "OPEN FILES" panel lists .env, web.php, and RegisterController.php. The "FOLDERS" panel shows a tree structure with folders like rants, app, Console, Exceptions, Http, Controllers, Auth, and files like ForgotPasswordController.php, LoginController.php, RegisterController.php, ResetPasswordController.php, Controller.php, HomeController.php, Middleware, Kernel.php, Providers, User.php, bootstrap, config, database, public, css, js, and htaccess. The main editor area shows the code for RegisterController.php, with line numbers 47 to 71. The code includes a protected function validator and a protected function create.

```
47 /
48 protected function validator(array $data)
49 {
50     return Validator::make($data, [
51         'name' => 'required|max:255',
52         'email' => 'required|email|max:255|unique:users',
53         'password' => 'required|min:6|confirmed',
54     ]);
55 }
56
57 /**
58  * Create a new user instance after a valid registration.
59  *
60  * @param array $data
61  * @return User
62  */
63 protected function create(array $data)
64 {
65     return User::create([
66         'name' => $data['name'],
67         'email' => $data['email'],
68         'password' => bcrypt($data['password']),
69     ]);
70 }
71 }
```

# TO THIS CODE...

```
.env  web.php  RegisterController.php
47  /
48  protected function validator(array $data)
49  {
50      return Validator::make($data, [
51          'name' => 'required|max:255',
52          'username' => 'required|max:32|unique:users',
53          'email' => 'required|email|max:255|unique:users',
54          'password' => 'required|min:6|confirmed',
55      ]);
56  }
57
58  /**
59   * Create a new user instance after a valid registration.
60   *
61   * @param array $data
62   * @return User
63   */
64  protected function create(array $data)
65  {
66      return User::create([
67          'name' => $data['name'],
68          'username' => $data['username'],
69          'bdate' => $data['bdate'],
70          'email' => $data['email'],
71          'password' => bcrypt($data['password']),
72      ]);
73  }
74  }
```

OOPS.. WE'RE NOT DONE YET.



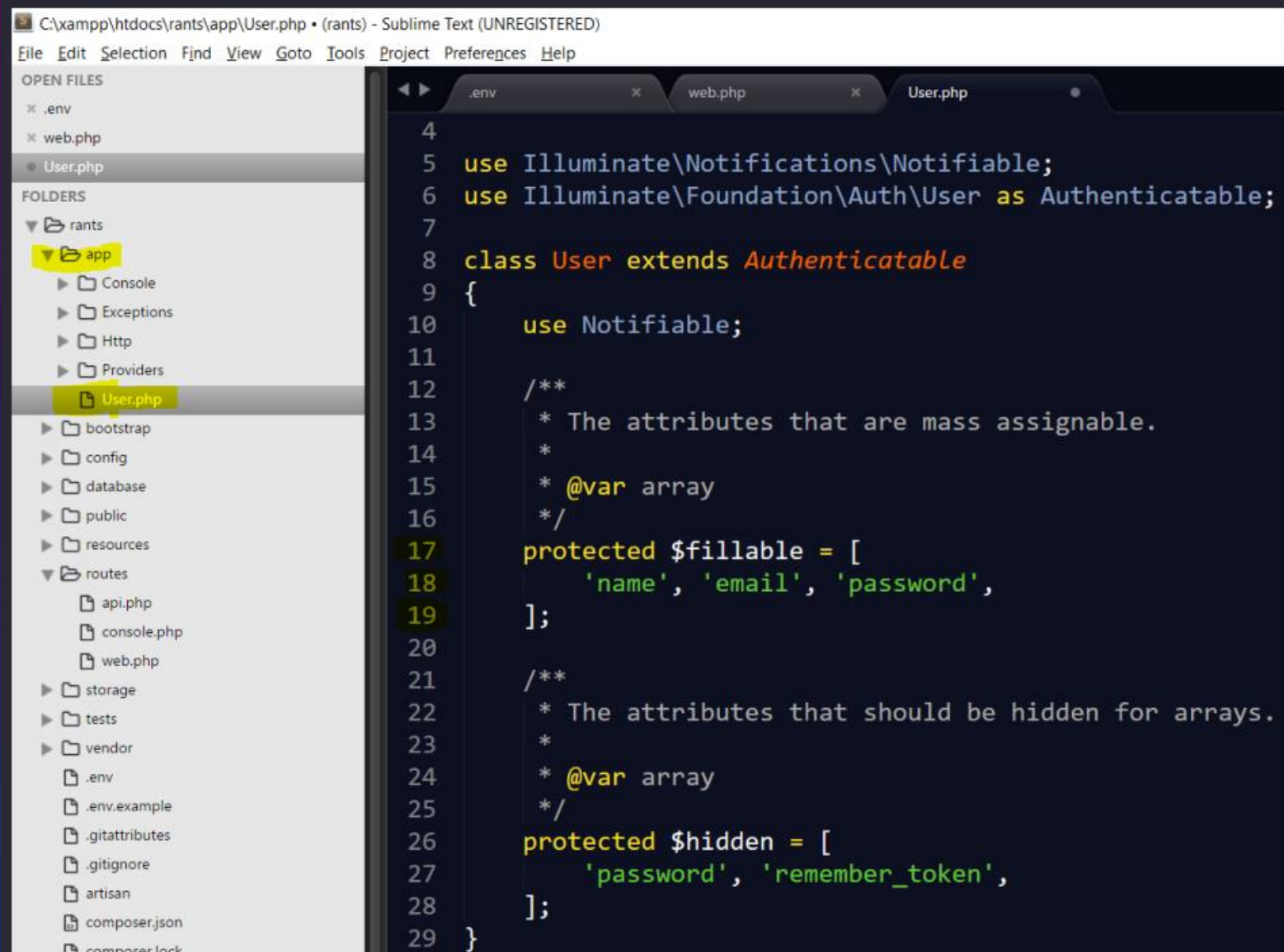


# EDIT THE USER MODEL!!

- Go to

`C://xampp/htdocs/rants/app/User.php`

# CHANGE THIS CODE...



```
C:\xampp\htdocs\rants\app\User.php • (rants) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

OPEN FILES
* .env
* web.php
• User.php

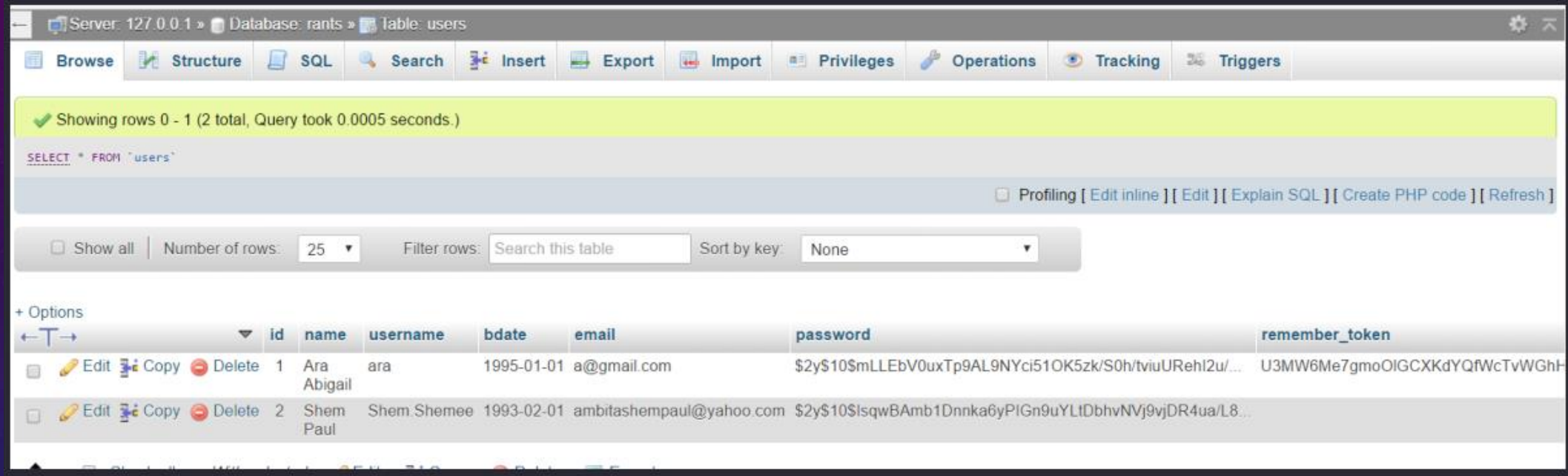
FOLDERS
▼ rants
  ▼ app
    ► Console
    ► Exceptions
    ► Http
    ► Providers
    User.php
  ► bootstrap
  ► config
  ► database
  ► public
  ► resources
  ▼ routes
    api.php
    console.php
    web.php
  ► storage
  ► tests
  ► vendor
    .env
    .env.example
    .gitattributes
    .gitignore
    artisan
    composer.json
    composer.lock

4
5 use Illuminate\Notifications\Notifiable;
6 use Illuminate\Foundation\Auth\User as Authenticatable;
7
8 class User extends Authenticatable
9 {
10     use Notifiable;
11
12     /**
13      * The attributes that are mass assignable.
14      *
15      * @var array
16      */
17     protected $fillable = [
18         'name', 'email', 'password',
19     ];
20
21     /**
22      * The attributes that should be hidden for arrays.
23      *
24      * @var array
25      */
26     protected $hidden = [
27         'password', 'remember_token',
28     ];
29 }
```

# TO THIS CODE...

```
4
5 use Illuminate\Notifications\Notifiable;
6 use Illuminate\Foundation\Auth\User as Authenticatable;
7
8 class User extends Authenticatable
9 {
10     use Notifiable;
11
12     /**
13      * The attributes that are mass assignable.
14      *
15      * @var array
16      */
17     protected $fillable = [
18         'name', 'email', 'password', 'username', 'bdate'
19     ];
20
21     /**
22      * The attributes that should be hidden for arrays.
23      *
24      * @var array
25      */
26     protected $hidden = [
27         'password', 'remember_token',
28     ];
29 }
```

# IF WE DOUBLE-CHECK OUR DATABASE..



The screenshot shows a database management interface with a toolbar at the top containing icons for Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, Tracking, and Triggers. Below the toolbar, a status bar indicates "Showing rows 0 - 1 (2 total, Query took 0.0005 seconds.)". The SQL query editor contains the text "SELECT \* FROM 'users'". To the right of the query editor are links for Profiling, Edit inline, Edit, Explain SQL, Create PHP code, and Refresh. Below these links are controls for "Show all", "Number of rows" (set to 25), "Filter rows" (with a search box), and "Sort by key" (set to None). The main area displays a table with the following data:

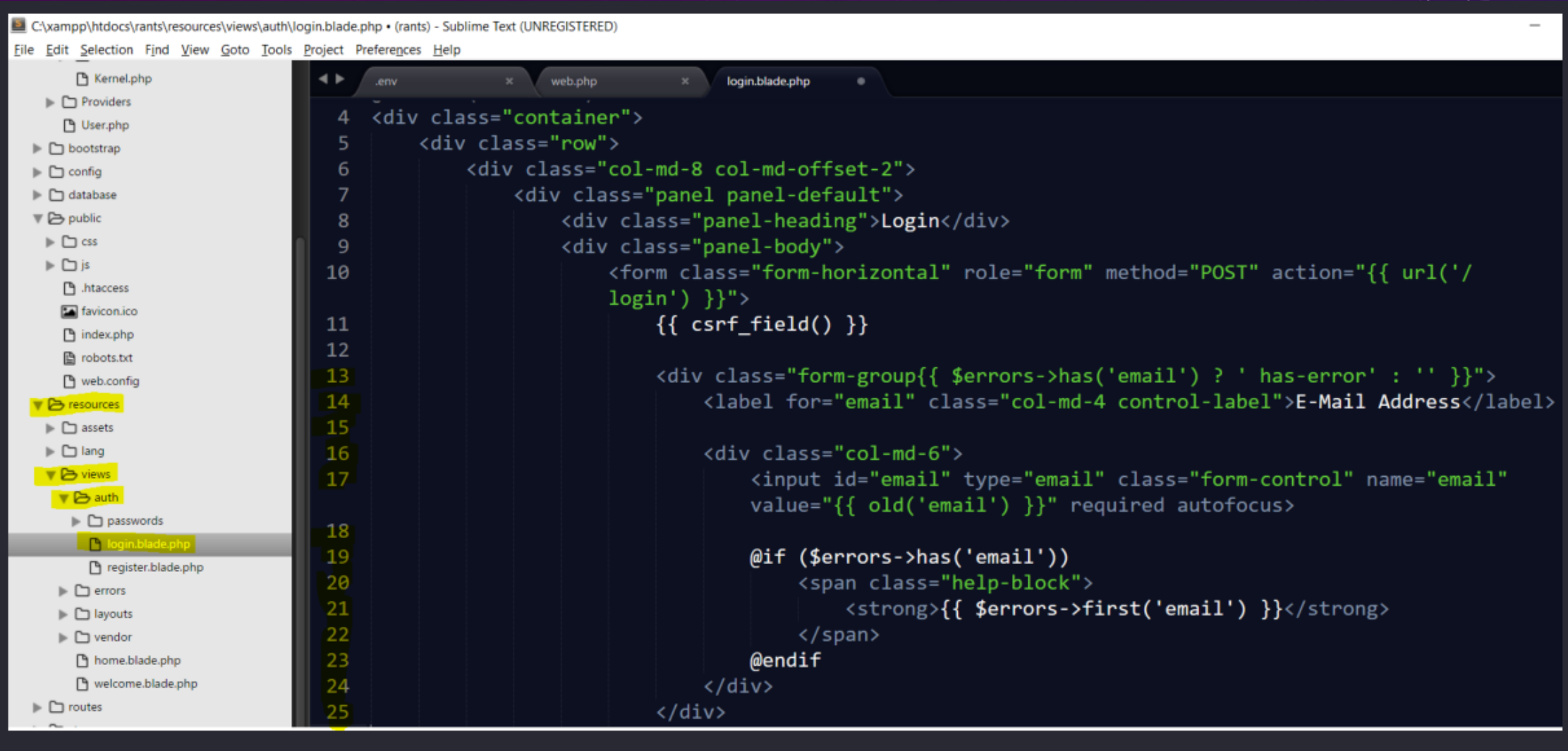
			id	name	username	bdate	email	password	remember_token
<input type="checkbox"/>	Edit	Copy	Delete	1	Ara Abigail	ara	1995-01-01	a@gmail.com	\$2y\$10\$mLLEbV0uxTp9AL9NYci51OK5zk/S0h/tviuURehI2u/... U3MW6Me7gmoOIGCXKdYQfWcTvWGhH
<input type="checkbox"/>	Edit	Copy	Delete	2	Shem Paul	Shem.Shemee	1993-02-01	ambitashempaul@yahoo.com	\$2y\$10\$lsqwBAmb1Dnnka6yPIGn9uYLtDbhvNVj9vjDR4ua/L8...



# MOVING ON...

- We will tweak our login page such that you will enter username instead of email address

# CONVERT THIS CODE



```
C:\xampp\htdocs\rants\resources\views\auth\login.blade.php • (rants) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

Kernel.php
Providers
User.php
bootstrap
config
database
public
css
js
.htaccess
favicon.ico
index.php
robots.txt
web.config
resources
assets
lang
views
auth
passwords
login.blade.php
register.blade.php
errors
layouts
vendor
home.blade.php
welcome.blade.php
routes

4 <div class="container">
5     <div class="row">
6         <div class="col-md-8 col-md-offset-2">
7             <div class="panel panel-default">
8                 <div class="panel-heading">Login</div>
9                 <div class="panel-body">
10                     <form class="form-horizontal" role="form" method="POST" action="{{ url('/
11                         login') }}">
12                         {{ csrf_field() }}
13
14                     <div class="form-group{{ $errors->has('email') ? ' has-error' : '' }}">
15                         <label for="email" class="col-md-4 control-label">E-Mail Address</label>
16
17                         <div class="col-md-6">
18                             <input id="email" type="email" class="form-control" name="email"
19                                 value="{{ old('email') }}" required autofocus>
20
21                             @if ($errors->has('email'))
22                                 <span class="help-block">
23                                     <strong>{{ $errors->first('email') }}</strong>
24                                 </span>
25                             @endif
26                         </div>
27                     </div>
28                 </div>
29             </div>
30         </div>
31     </div>
32 </div>
```

# TO THIS CODE

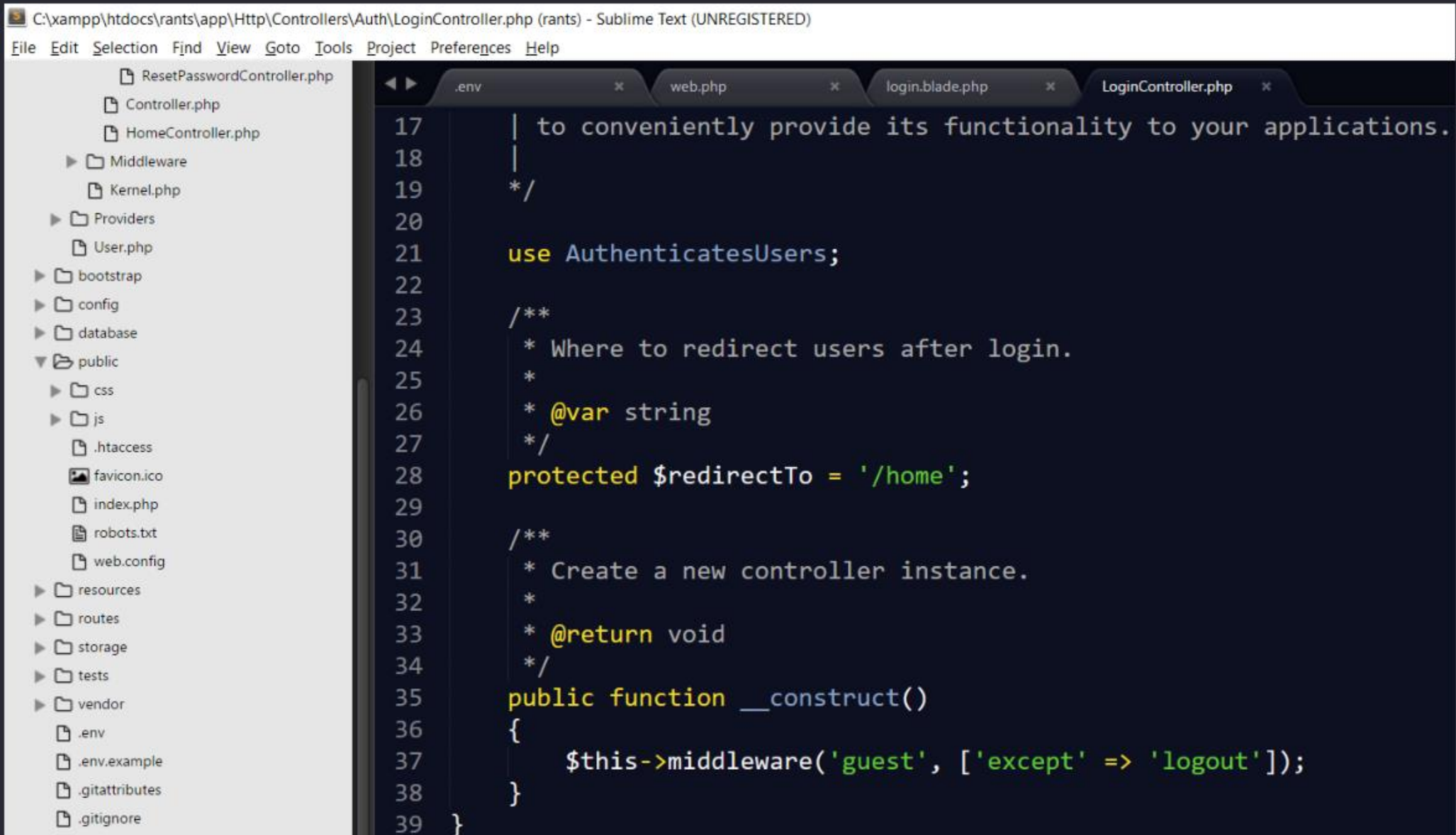
C:\xampp\htdocs\rants\resources\views\auth\login.blade.php (rants) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

- Kernel.php
- Providers
- User.php
- bootstrap
- config
- database
- public
  - css
  - js
  - .htaccess
  - favicon.ico
  - index.php
  - robots.txt
  - web.config
- resources
  - assets
  - lang
  - views
    - auth
      - passwords
      - login.blade.php
      - register.blade.php
- errors
- layouts
- vendor
  - home.blade.php
  - welcome.blade.php
- routes

```
4 <div class="container">
5     <div class="row">
6         <div class="col-md-8 col-md-offset-2">
7             <div class="panel panel-default">
8                 <div class="panel-heading">Login</div>
9                 <div class="panel-body">
10                     <form class="form-horizontal" role="form" method="POST" action="{{ url('/login') }}">
11                         {{ csrf_field() }}
12
13                     <div class="form-group{{ $errors->has('username') ? ' has-error' : '' }}">
14                         <label for="username" class="col-md-4 control-label">Username</label>
15
16                         <div class="col-md-6">
17                             <input id="username" type="text" class="form-control" name="username" value="{{ old('username') }}" required autofocus>
18
19                             @if ($errors->has('username'))
20                                 <span class="help-block">
21                                     <strong>{{ $errors->first('username') }}</strong>
22                                 </span>
23                             @endif
24                         </div>
25                     </div>
26                 </div>
27             </div>
28         </div>
29     </div>
30 </div>
```

# OPEN LOGINCONTROLLER.PHP



C:\xampp\htdocs\rants\app\Http\Controllers\Auth\LoginController.php (rants) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

ResetPasswordController.php  
Controller.php  
HomeController.php  
Middleware  
Kernel.php  
Providers  
User.php  
bootstrap  
config  
database  
public  
css  
js  
.htaccess  
favicon.ico  
index.php  
robots.txt  
web.config  
resources  
routes  
storage  
tests  
vendor  
.env  
.env.example  
.gitattributes  
.gitignore

```
17 | to conveniently provide its functionality to your applications.
18 |
19 */
20
21 use AuthenticatesUsers;
22
23 /**
24  * Where to redirect users after login.
25  *
26  * @var string
27  */
28 protected $redirectTo = '/home';
29
30 /**
31  * Create a new controller instance.
32  *
33  * @return void
34  */
35 public function __construct()
36 {
37     $this->middleware('guest', ['except' => 'logout']);
38 }
39 }
```



# GO TO AUTHENTICATESUSERS.PHP

C:\xampp\htdocs\rants\vendor\laravel\framework\src\Illuminate\Foundation\Auth\AuthenticatesUsers.php (rants) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

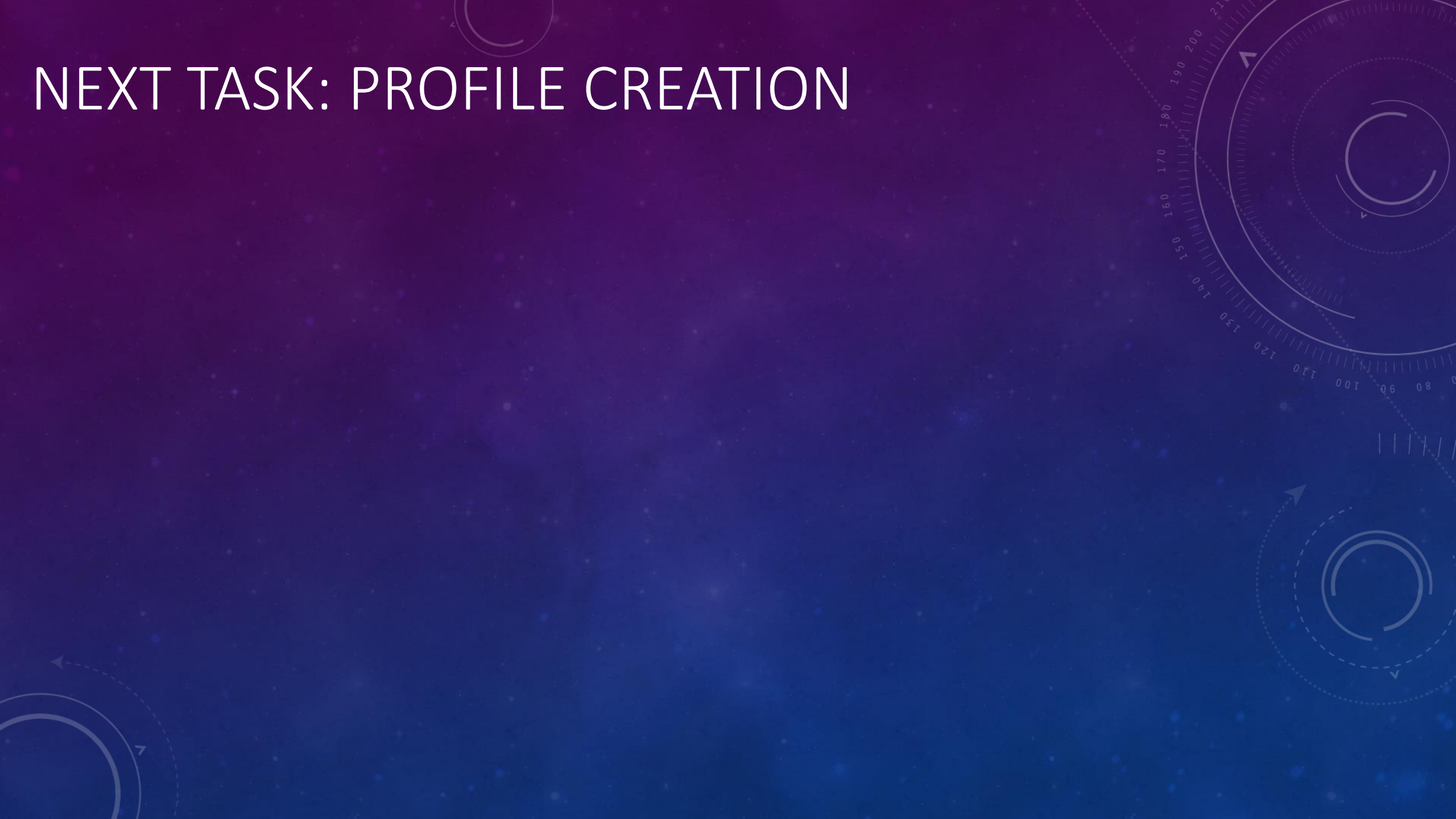
vendor

- bin
- classpreloader
- composer
- dnoegel
- doctrine
- fzaninotto
- hamcrest
- jakub-ondarka
- jeremeamia
- laravel
  - framework
    - src
      - Illuminate
        - Auth
        - Broadcasting
        - Bus
        - Cache
        - Config
        - Console
        - Container
        - Contracts
        - Cookie
        - Database
        - Encryption
        - Events
        - Filesystem
        - Foundation
          - Auth
            - Access

AuthenticatesUsers.php

```
119 /**
120  * Get the failed login response instance.
121  *
122  * @param \Illuminate\Http\Request $request
123  * @return \Illuminate\Http\RedirectResponse
124  */
125 protected function sendFailedLoginResponse(Request $request)
126 {
127     return redirect()->back()
128         ->withInput($request->only($this->username(), 'remember'))
129         ->withErrors([
130             $this->username() => Lang::get('auth.failed'),
131         ]);
132 }
133
134 /**
135  * Get the login username to be used by the controller.
136  *
137  * @return string
138  */
139 public function username()
140 {
141     return 'username';
142 }
143
144 /**
145  * Log the user out of the application.
```

# NEXT TASK: PROFILE CREATION



# CREATE PROFILECONTROLLER.PHP

Type in the command

`php artisan make:controller ProfileController`

Command Prompt

```
C:\xampp\htdocs\rants>php artisan make:controller ProfileController  
Controller created successfully.
```

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

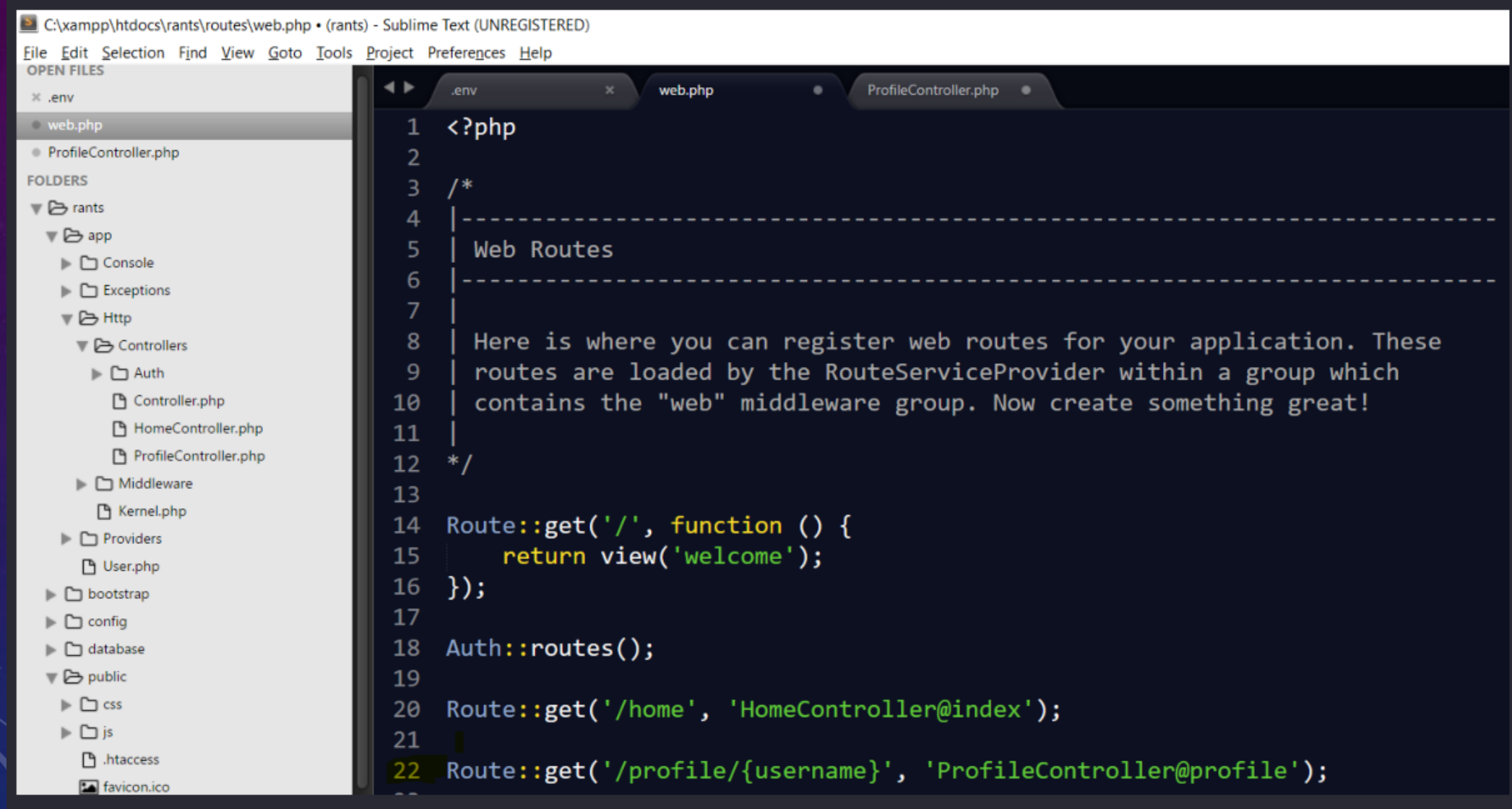
# CREATE PROFILECONTROLLER.PHP

If you check controllers,  
ProfileController.php is automatically created

Go to routes (web.php)



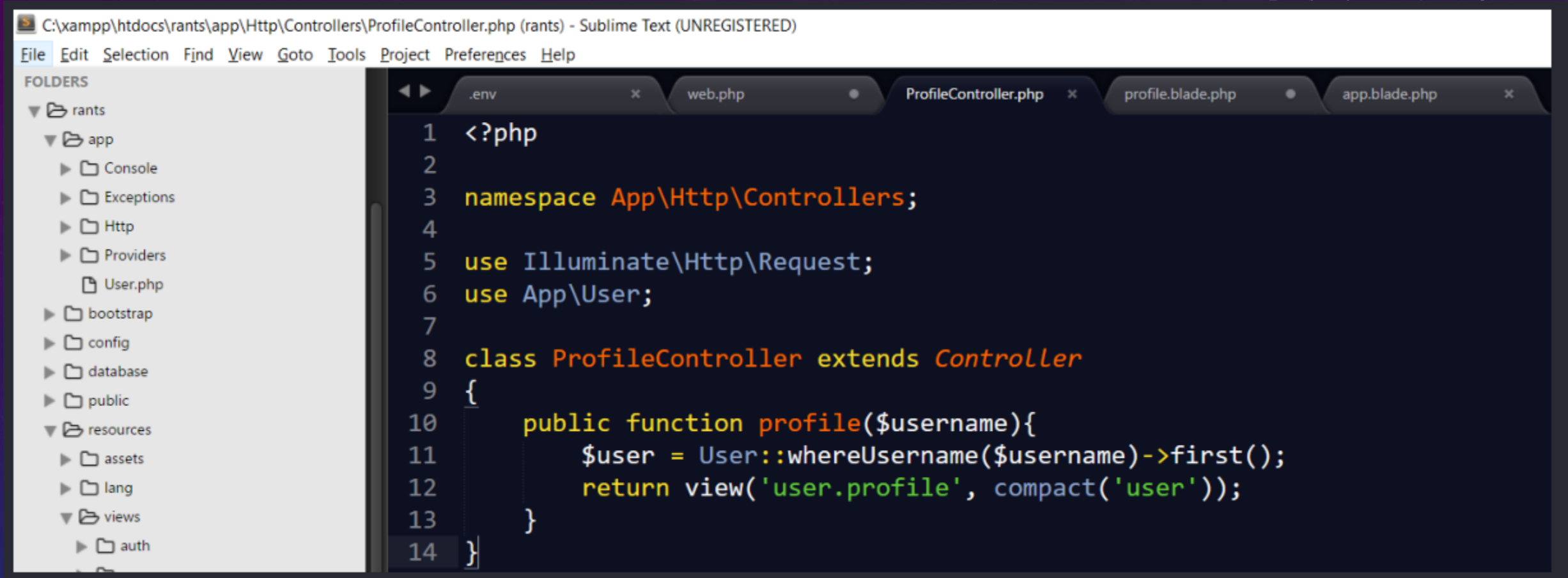
# ADD THIS CODE TO ROUTES



```
C:\xampp\htdocs\rants\routes\web.php • (rants) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
OPEN FILES
x .env
web.php
ProfileController.php
FOLDERS
rants
  app
    Console
    Exceptions
    Http
      Controllers
        Auth
          Controller.php
          HomeController.php
          ProfileController.php
        Middleware
          Kernel.php
        Providers
          User.php
    bootstrap
    config
    database
    public
      css
      js
      .htaccess
      favicon.ico

1 <?php
2
3 /*
4 |-----
5 | Web Routes
6 |-----
7 |
8 | Here is where you can register web routes for your application. These
9 | routes are loaded by the RouteServiceProvider within a group which
10 | contains the "web" middleware group. Now create something great!
11 |
12 */
13
14 Route::get('/', function () {
15     return view('welcome');
16 });
17
18 Auth::routes();
19
20 Route::get('/home', 'HomeController@index');
21
22 Route::get('/profile/{username}', 'ProfileController@profile');
```

# WE DEFINE FUNCTION PROFILE



C:\xampp\htdocs\rants\app\Http\Controllers\ProfileController.php (rants) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- ▼ rants
  - ▼ app
    - ▶ Console
    - ▶ Exceptions
    - ▶ Http
    - ▶ Providers
      - User.php
  - ▶ bootstrap
  - ▶ config
  - ▶ database
  - ▶ public
  - ▼ resources
    - ▶ assets
    - ▶ lang
    - ▼ views
      - ▶ auth

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\User;
7
8 class ProfileController extends Controller
9 {
10     public function profile($username){
11         $user = User::whereUsername($username)->first();
12         return view('user.profile', compact('user'));
13     }
14 }
```

# NOW, FOR THE VIEWS!

Putting the code to print out our profile in controllers  
will clutter our controllers

# SETTING UP PROFILE PAGE

Create a new folder user

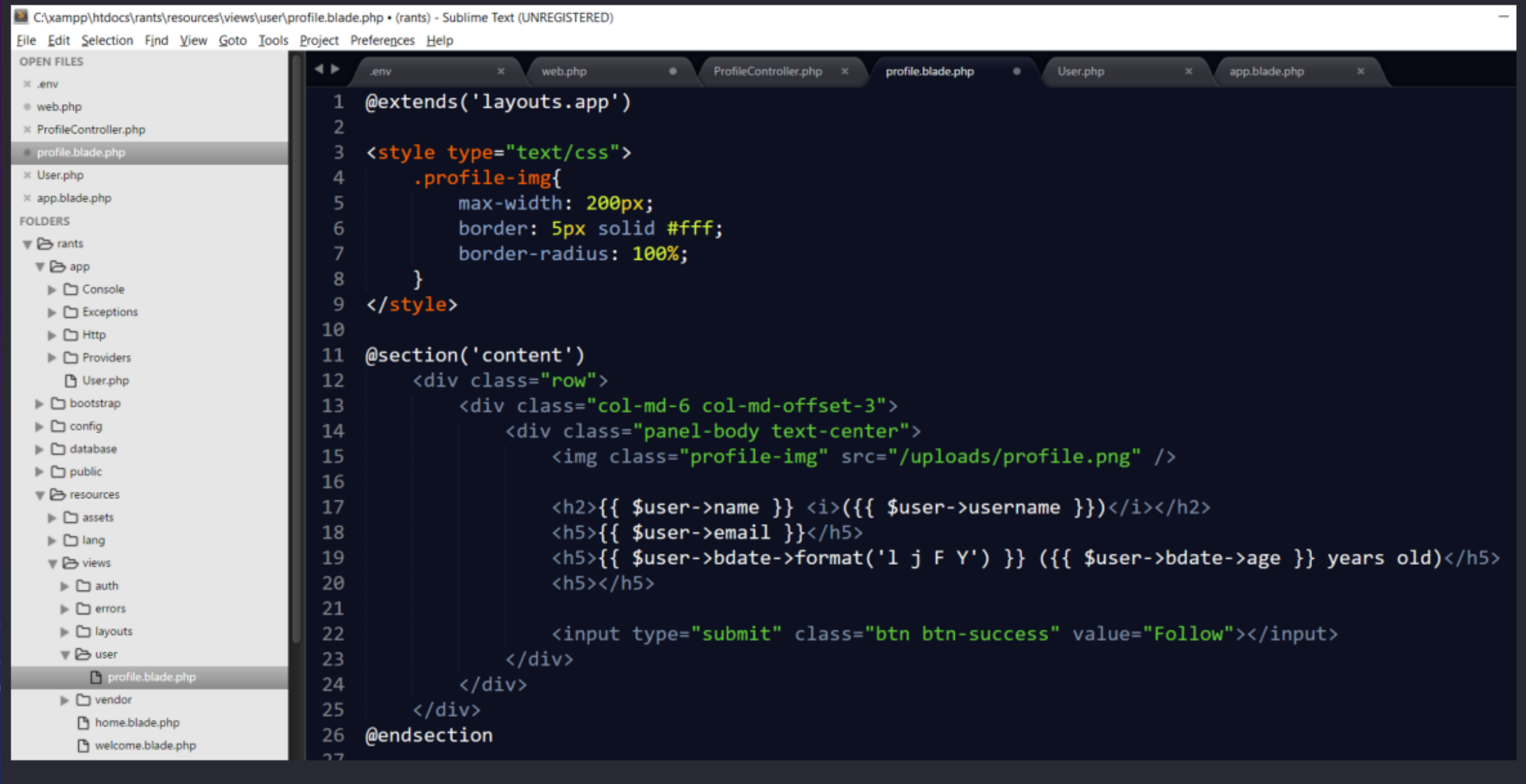
Create a new file `profile.blade.php`



# TO CREATE STORAGE FOR PROFILE PHOTOS

create new folder uploads in public folder

# ADD THIS CODE TO PROFILE.BLADE.PHP



```
C:\xampp\htdocs\rants\resources\views\user\profile.blade.php • (rants) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

OPEN FILES
x .env
x web.php
x ProfileController.php
x profile.blade.php
x User.php
x app.blade.php
FOLDERS
v rants
  v app
    Console
    Exceptions
    Http
    Providers
    User.php
  bootstrap
  config
  database
  public
  resources
    assets
    lang
    v views
      auth
      errors
      layouts
      v user
        profile.blade.php
        home.blade.php
        welcome.blade.php
  vendor

1 @extends('layouts.app')
2
3 <style type="text/css">
4     .profile-img{
5         max-width: 200px;
6         border: 5px solid #fff;
7         border-radius: 100%;
8     }
9 </style>
10
11 @section('content')
12     <div class="row">
13         <div class="col-md-6 col-md-offset-3">
14             <div class="panel-body text-center">
15                 
16
17                 <h2>{{ $user->name }} <i>({{ $user->username }})</i></h2>
18                 <h5>{{ $user->email }}</h5>
19                 <h5>{{ $user->bdate->format('1 j F Y') }} ({{ $user->bdate->age }} years old)</h5>
20                 <h5></h5>
21
22                 <input type="submit" class="btn btn-success" value="Follow"></input>
23             </div>
24         </div>
25     </div>
26 @endsection
27
```

# EXERCISE!!!

Create an about page

You can have your own design

If we type in

`localhost:8000/about`

It will redirect to an about page