When cchaching from same database

php artisan config:cache

If you want to modify a table in database then

Like your table name is posts

Then run php artisan make:migration add\_data\_to\_posts\_table

You can name it anything like edit\_data\_to\_posts\_table

In

*\*/*

**public** **function** up()

    {

        Schema::table('blogposts', **function** (**Blueprint** $table) {

*//*

         $table->string('content');

        });

    }

Here add the columns u need

Down

**public** **function** down()

    {

        Schema::table('blogposts', **function** (**Blueprint** $table) {

*//*

            $table->dropColumn(['description']);

        });

    }

Better not to do up and down at same time

To rename

Create a new migration

Php artisan make:migration change\_users\_table\_name

Then just use this

**public** **function** up()

    {

        Schema::rename('blog\_posts(existing table name)','blogposts\_renamed(new table\_name)');

Write opposite in case of down method

    }

*/\*\**

*\* Reverse the migrations.*

*\**

*\* @return void*

*\*/*

**public** **function** down()

    {

        Schema::rename('blogposts\_renamed', 'blog\_posts');

    }

**To create a factory**

php artisan make:factory CommentFactory --model=Comment

php artisan make:factory AuthorFactory --model=Author

php artisan make:factory ProfileFactory --model=Author

php artisan make:factory UserFactory --model=User

**php artisan make:factory BlogPostFactory --model=BlogPost**

to run faker with tinker

**To use the Factory App**

factory('App\Comment')->create(['blog\_post\_id' => 2]);

//For tests

./vendor/bin/phpunit

factory('App\BlogPost')->create();

factory('App\Author')->create();

**To make an authentication**

composer require laravel/ui --dev

php artisan ui vue --auth

then run npm install && npm run dev

yes

If you want to authenticate users with name or username you can login with username or without email in Laravel there is no username by default so to add username we need to create a seperate column

AuthenticateUsers.php

public function username(){

return 'name';

}

in LoginController

If we want users login with names

public function username(){

return 'name';

}

in login.blade.php

<div class="form-group row">

<label for="name" class="col-md-4 col-form-label text-md-right">{{ \_\_('Username') }}</label>

<div class="col-md-6">

<input id="name" type="text" class="form-control @error('name') is-invalid @enderror" name="name" value="{{ old('name') }}" required autocomplete="name" autofocus>

@error('name')

<span class="invalid-feedback" role="alert">

<strong>{{ $message }}</strong>

</span>

@enderror

</div>

</div>

Gurads are for sessisn and APIS are for Token

RedirectifAuthenticated.php

if (Auth::guard($guard)->check()) {

return redirect('/home');

}

We will be redirected if authenticated

Remember me?

protected function attemptLogin(Request $request)

{

return $this->guard()->attempt(

$this->credentials($request), $request->filled('remember')

);

}

**How to display the login for guest and non guest in menubar of layouts**

<div class="d-flex flex-column text-white flex-md-row align-items-center p-3 px-md-4 mb-3 bg-white shadow-sm menuclass">

<h5 class="my-0 mr-md-auto font-weight-normal"> <a class="navbar-brand" href="{{ url('/') }}">

{{ config('app.name', 'Laravel') }}

</a></h5>

<nav class="my-2 my-md-0 mr-md-3">

<a class="p-2 text-dark" href="{{ route('home') }}">Home</a>

<a class="p-2 text-dark" href="{{ route('contact') }}">Contact</a>

<a class="p-2 text-dark" href="{{ route('posts.index') }}">Blog Posts</a>

<a class="p-2 text-dark" href="{{ route('posts.create') }}">Add Blog Post</a>

@guest

@if (Route::has('register'))

<a class="p-2 text-dark" href="{{ route('register') }}">Register</a>

@endif

<a class="p-2 text-dark" href="{{ route('login') }}">Login</a>

@else

<a class="p-2 text-dark" href="{{ route('logout') }}"

onclick="event.preventDefault();document.getElementById('logout-form').submit();"

>Logout ({{ Auth::user()->name }})</a>

<form id="logout-form" action={{ route('logout') }} method="POST"

style="display: none;">

@csrf

</form>

@endguest

</nav>

</div>

To check id of current user dd(Auth::id());

To check authenticity of current user dd(Auth::check());

To check details of the user

dd(Auth::user());

To just get name of user

dd(Auth::user()->name);

To prevent any users other than ones registered from loggng in

public function \_\_construct()

{

$this->middleware('auth');

}

Specify the functions that are under or which guest can enter or do

public function \_\_construct()

{

$this->middleware('auth')

->only(['create','store','edit','update','destroy']);

}

**Integrity constraint violation: 1452 Cannot add or update a child row: a foreign key**

constraint fails (`laraveludemyfirst`.`#sql-1a94\_3c`, CONSTRAINT `blog\_posts\_user\_id\_foreign` FOREIGN KEY (`user\_id`) REFERENCES `users` (`id`))")?[39m

Solution to this problem

$table->unsignedBigInteger('user\_id')->nullable();

It will make or foreign key from 0 to NULL

To create seed

**php artisan db:seed**

**STR in Laravel 6**

**To avoid STR undefined**

**use Illuminate\Support\Str;**

**'remember\_token' => Str::random(10)**

**When we want to migrate the whole database keeping the users intact**

**php artisan migrate:fresh --seed**

**To create more users using Factory**

factory(App\User::class)->states('john-doe')->create();

factory(App\User::class, 20)->create();

password-secret

**Model relations between users**

**How to assign a User to create blog posts**

**and comments in Blog posts**

**auto generated Blog Posts and comments**

For generating random blog posts and class

$posts = factory(App\BlogPost::class, 50)->make()->each(function($post) use ($users) {

$post->user\_id = $users->random()->id;

$post->save();

});

$comments = factory(App\Comment::class, 150)->make()->each(function ($comment) use ($posts) {

$comment->blog\_post\_id = $posts->random()->id;

$comment->save();

});