**Model and Migrate Definition:**

**Users:**

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\DB;

use Illuminate\Support\Facades\Schema;

class CreateUsersTable extends Migration

{

    /\*\*

     \* Run the migrations.

     \*

     \* @return void

     \*/

    public function up()

    {

        Schema::create('users', function (Blueprint $table) {

            $table->id();

            $table->string('name');

            $table->string('email')->unique();

            $table->timestamp('email\_verified\_at')->nullable();

            $table->string('password');

            $table->integer('usertype');

            $table->rememberToken();

            $table->timestamps();

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*

     \* @return void

     \*/

    public function down()

    {

        Schema::dropIfExists('users');

    }

}

<?php

namespace App\Models;

use Illuminate\Contracts\Auth\MustVerifyEmail;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Foundation\Auth\User as Authenticatable;

use Illuminate\Notifications\Notifiable;

use Laravel\Sanctum\HasApiTokens;

class User extends Authenticatable

{

    use HasApiTokens, HasFactory, Notifiable;

    /\*\*

     \* The attributes that are mass assignable.

     \*

     \* @var string[]

     \*/

    protected $fillable = [

        'name',

        'email',

        'password',

    ];

    /\*\*

     \* The attributes that should be hidden for serialization.

     \*

     \* @var array

     \*/

    protected $hidden = [

        'password',

        'remember\_token',

    ];

    /\*\*

     \* The attributes that should be cast.

     \*

     \* @var array

     \*/

    protected $casts = [

        'email\_verified\_at' => 'datetime',

    ];

}

**Posts:**

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\DB;

use Illuminate\Support\Facades\Schema;

class CreatePostsTable extends Migration

{

    /\*\*

     \* Run the migrations.

     \*

     \* @return void

     \*/

    public function up()

    {

        Schema::create('posts', function (Blueprint $table) {

            $table->id();

            $table->string('title');

            $table->text('content');

            $table->Integer('userid');

            $table->timestamp('created\_at')->default(DB::raw('CURRENT\_TIMESTAMP'));

            $table->timestamp('update\_at')->nullable();

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*

     \* @return void

     \*/

    public function down()

    {

        Schema::dropIfExists('posts');

    }

}

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class Posts extends Model

{

    use HasFactory;

    public $timestamps = false;

    public function user()

    {

        return $this->hasOne(User::class);

    }

}

**PostViews:**

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\DB;

use Illuminate\Support\Facades\Schema;

class CreatePostsViewsTable extends Migration

{

    /\*\*

     \* Run the migrations.

     \*

     \* @return void

     \*/

    public function up()

    {

        Schema::create('posts\_views', function (Blueprint $table) {

            $table->id();

            $table->integer('id\_post')->unsigned();

            $table->integer('userid')->unsigned();

            $table->timestamp('created\_at')->default(DB::raw('CURRENT\_TIMESTAMP'));

            $table->timestamp('updated\_at')->nullable();

            //$table->foreign('userid')->references('id')->on('users');

            //$table->foreign('id\_post')->references('id')->on('posts');

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*

     \* @return void

     \*/

    public function down()

    {

        Schema::dropIfExists('posts\_views');

    }

}

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class PostsViews extends Model

{

    use HasFactory;

    public function user()

    {

        return $this->hasOne(User::class);

    }

    public function post()

    {

        return $this->hasOne(Posts::class);

    }

}

**Posts Comments:**

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\DB;

use Illuminate\Support\Facades\Schema;

class CreateCommentsPostsTable extends Migration

{

    /\*\*

     \* Run the migrations.

     \*

     \* @return void

     \*/

    public function up()

    {

        Schema::create('comments\_posts', function (Blueprint $table) {

            $table->id();

            $table->integer('id\_post')->unsigned();

            $table->text('descripcion');

            $table->integer('userid')->unsigned();

            $table->timestamp('created\_at')->default(DB::raw('CURRENT\_TIMESTAMP'));

            $table->timestamp('update\_at')->nullable();

            //$table->foreign('userid')->references('id')->on('users');

           // $table->foreign('id\_post')->references('id')->on('posts');

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*

     \* @return void

     \*/

    public function down()

    {

        Schema::dropIfExists('comments\_posts');

    }

}

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class CommentsPost extends Model

{

    use HasFactory;

    public $timestamps = false;

    public function user()

    {

        return $this->hasOne(User::class);

    }

    public function post()

    {

        return $this->hasOne(Posts::class);

    }

}

**Factories:**

**Posts:**

<?php

namespace Database\Factories;

use App\Models\Posts;

use App\Models\User;

use Illuminate\Database\Eloquent\Factories\Factory;

class PostsFactory extends Factory

{

    /\*\*

     \* The name of the factory's corresponding model.

     \*

     \* @var string

     \*/

    protected $model = Posts::class;

    /\*\*

     \* Define the model's default state.

     \*

     \* @return array

     \*/

    public function definition()

    {

        return [

            'title' => $this->faker->sentence(),

            'content' => $this->faker->paragraph(),

            'userid' =>$this->faker->randomElement(User::all())

        ];

    }

}

**PostsView:**

<?php

namespace Database\Factories;

use App\Models\Posts;

use App\Models\PostsViews;

use App\Models\User;

use Illuminate\Database\Eloquent\Factories\Factory;

class PostsViewsFactory extends Factory

{

    /\*\*

     \* The name of the factory's corresponding model.

     \*

     \* @var string

     \*/

    protected $model = PostsViews::class;

    /\*\*

     \* Define the model's default state.

     \*

     \* @return array

     \*/

    public function definition()

    {

        return [

            'id\_post' => $this->faker->randomElement(Posts::all()),

            'userid' => $this->faker->randomElement(User::all())

        ];

    }

}

**Posts Comments:**

<?php

namespace Database\Factories;

use App\Models\CommentsPost;

use App\Models\Posts;

use App\Models\User;

use Illuminate\Database\Eloquent\Factories\Factory;

class CommentsPostFactory extends Factory

{

    /\*\*

     \* The name of the factory's corresponding model.

     \*

     \* @var string

     \*/

    protected $model = CommentsPost::class;

    /\*\*

     \* Define the model's default state.

     \*

     \* @return array

     \*/

    public function definition()

    {

        return [

            'id\_post' => $this->faker->randomElement(Posts::all()),

            'descripcion' => $this->faker->paragraph(),

            'userid' => $this->faker->randomElement(User::all())

        ];

    }

}

**User:**

<?php

namespace Database\Factories;

use App\Models\User;

use Illuminate\Database\Eloquent\Factories\Factory;

use Illuminate\Support\Str;

class UserFactory extends Factory

{

    /\*\*

     \* The name of the factory's corresponding model.

     \*

     \* @var string

     \*/

    protected $model = User::class;

    /\*\*

     \* Define the model's default state.

     \*

     \* @return array

     \*/

    public function definition()

    {

        return [

            'name' => $this->faker->name(),

            'email' => $this->faker->unique()->safeEmail(),

            'email\_verified\_at' => now(),

            'password' => '$2y$10$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', // password

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

        ];

    }

    /\*\*

     \* Indicate that the model's email address should be unverified.

     \*

     \* @return \Illuminate\Database\Eloquent\Factories\Factory

     \*/

    public function unverified()

    {

        return $this->state(function (array $attributes) {

            return [

                'email\_verified\_at' => null,

            ];

        });

    }

}

**Seeding:**

**DatabaseSeeder:**

<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;

class DatabaseSeeder extends Seeder

{

    /\*\*

     \* Seed the application's database.

     \*

     \* @return void

     \*/

    public function run()

    {

        // \App\Models\User::factory(10)->create();

        $this->call(UserSeeder::class);

        $this->call(PostSeeder::class);

        $this->call(ViewesSeeder::class);

        $this->call(CommentsSeeder::class);

    }

}

**PostsSeeder:**

<?php

namespace Database\Seeders;

use App\Models\Posts;

use Illuminate\Database\Seeder;

use Illuminate\Support\Facades\DB;

class PostSeeder extends Seeder

{

    /\*\*

     \* Run the database seeds.

     \*

     \* @return void

     \*/

    public function run()

    {

        //

        Posts::factory(25)->create();

    }

}

**ViewsSeeder:**

<?php

namespace Database\Seeders;

use App\Models\PostsViews;

use Illuminate\Database\Seeder;

use Illuminate\Support\Facades\DB;

class ViewesSeeder extends Seeder

{

    /\*\*

     \* Run the database seeds.

     \*

     \* @return void

     \*/

    public function run()

    {

        //

        PostsViews::factory(150)->create();

    }

}

**CommentsSeeder:**

<?php

namespace Database\Seeders;

use App\Models\CommentsPost;

use Illuminate\Database\Seeder;

class CommentsSeeder extends Seeder

{

    /\*\*

     \* Run the database seeds.

     \*

     \* @return void

     \*/

    public function run()

    {

        //

        CommentsPost::factory(50)->create();

    }

}

**UserSeeder:**

<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;

use Illuminate\Support\Facades\DB;

class UserSeeder extends Seeder

{

    /\*\*

     \* Run the database seeds.

     \*

     \* @return void

     \*/

    public function run()

    {

        //

        DB::table('users')->insert(

            array(

                'name' => 'admin',

                'email' => 'admin@test.com',

                'password' =>'admin',

                'usertype' => 1

            )

        );

        DB::table('users')->insert(

            array(

                'name' => 'admin1',

                'email' => 'admin1@test.com',

                'password' =>'admin1',

                'usertype' => 1

            )

        );

        DB::table('users')->insert(

            array(

                'name' => 'nicole',

                'email' => 'nicole@test.com',

                'password' =>'1234',

                'usertype' => 2

            )

        );

    }

}