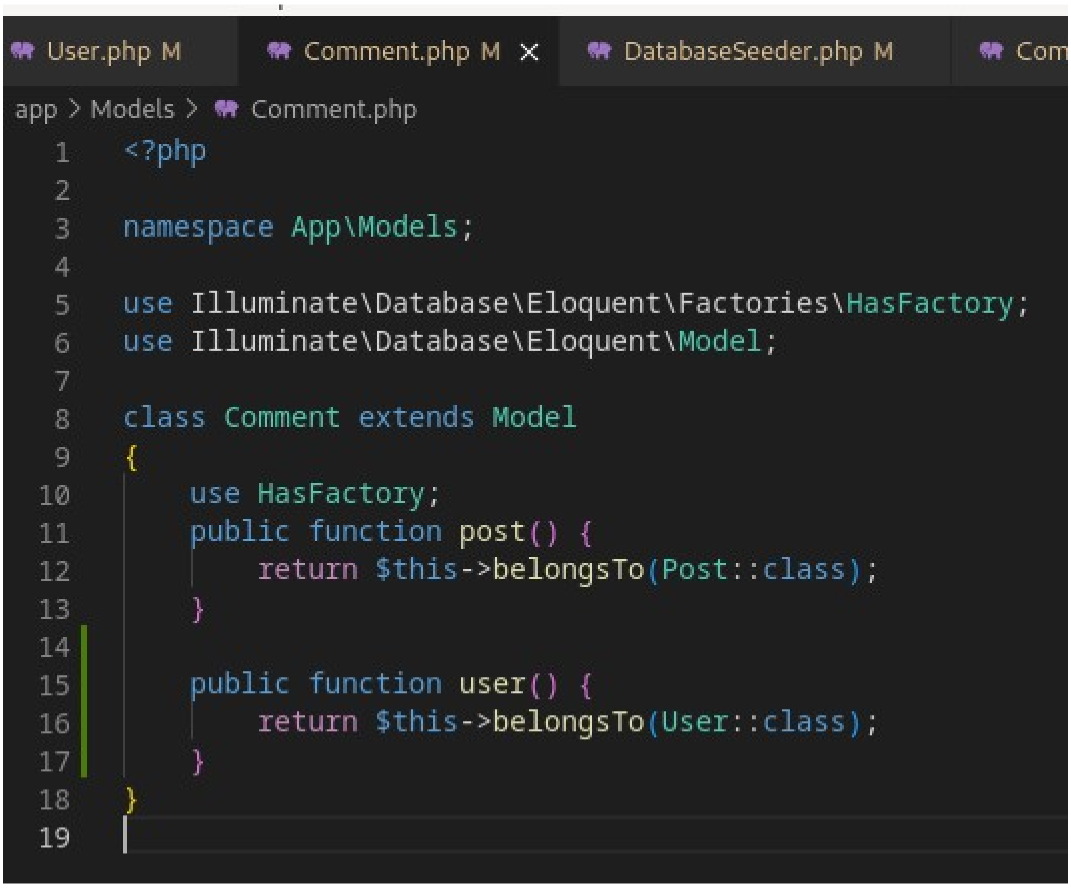
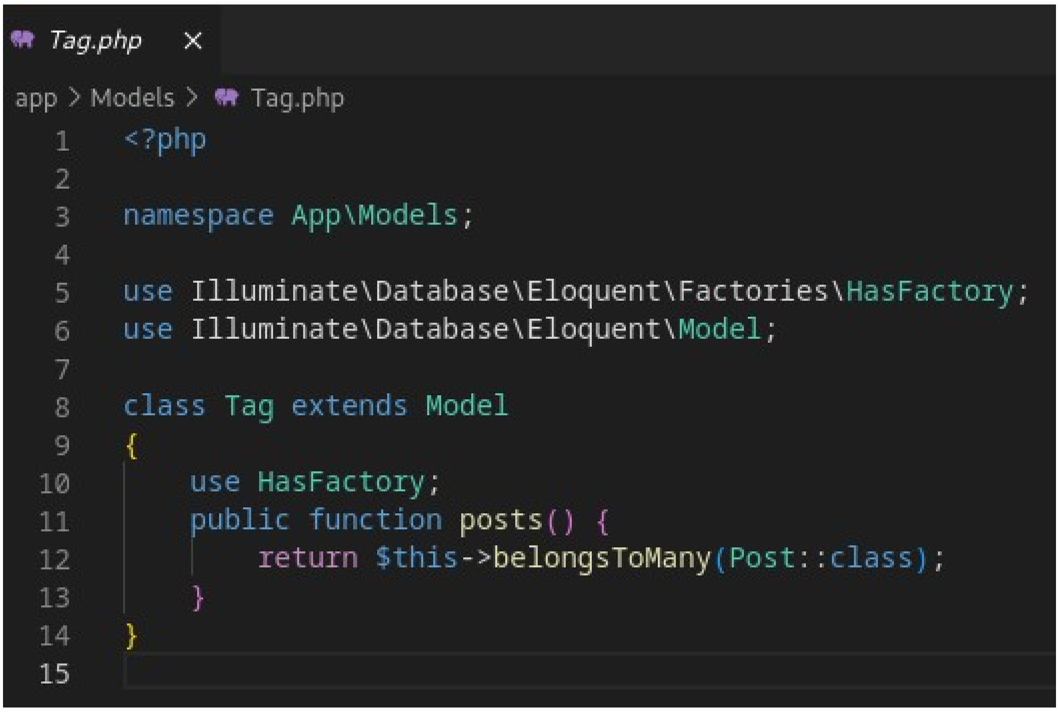
Model Definitions

Comment:



```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Comment extends Model
{
    use HasFactory;
    public function post() {
        return $this->belongsTo(Post::class);
    }

    public function user() {
        return $this->belongsTo(User::class);
    }
}
```

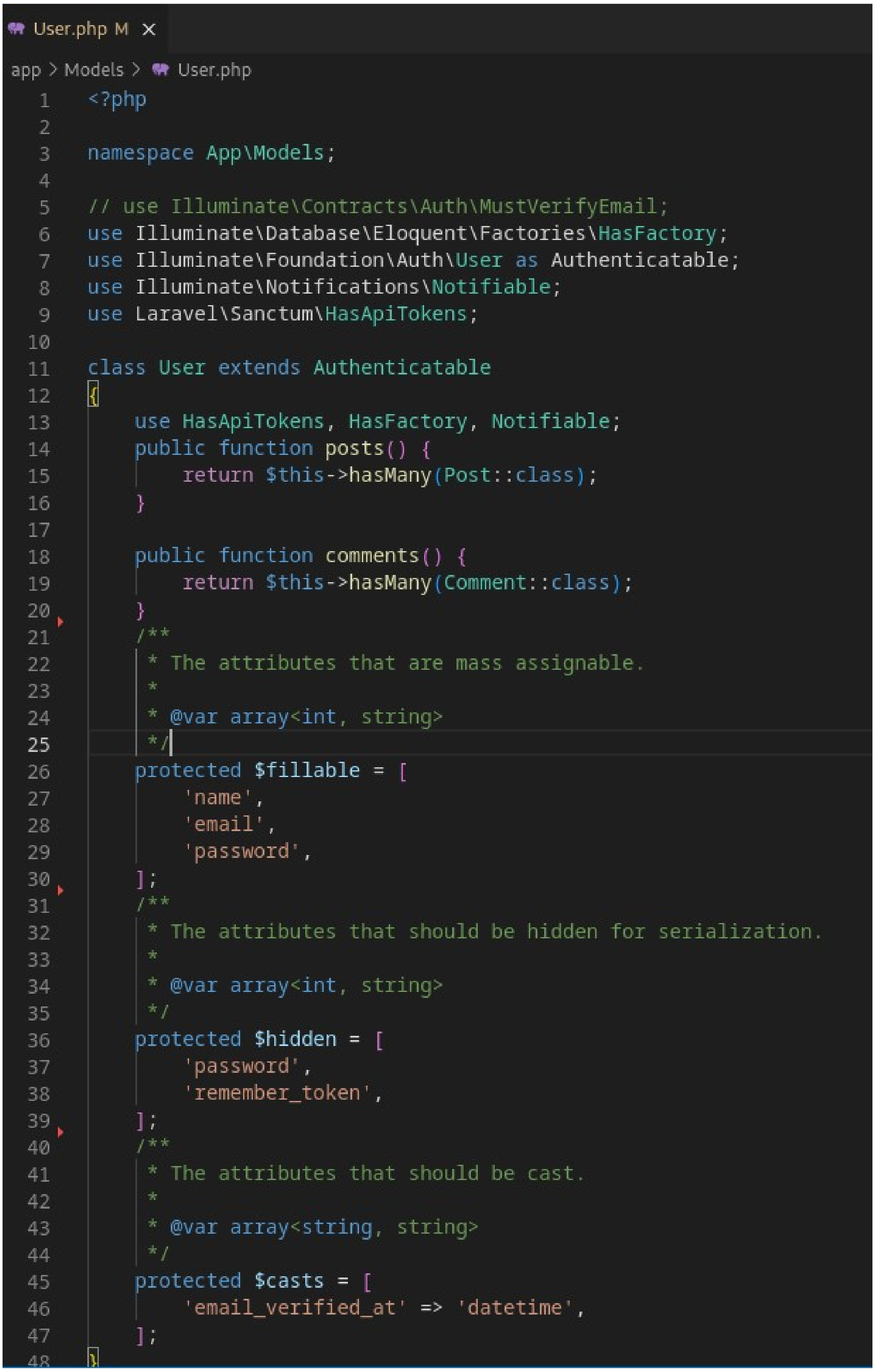
Post:



```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Tag extends Model
{
    use HasFactory;
    public function posts() {
        return $this->belongsToMany(Post::class);
    }
}
```

User:

Note: This is the default user model added by breeze, which I have modified to add the posts() and comments() methods.

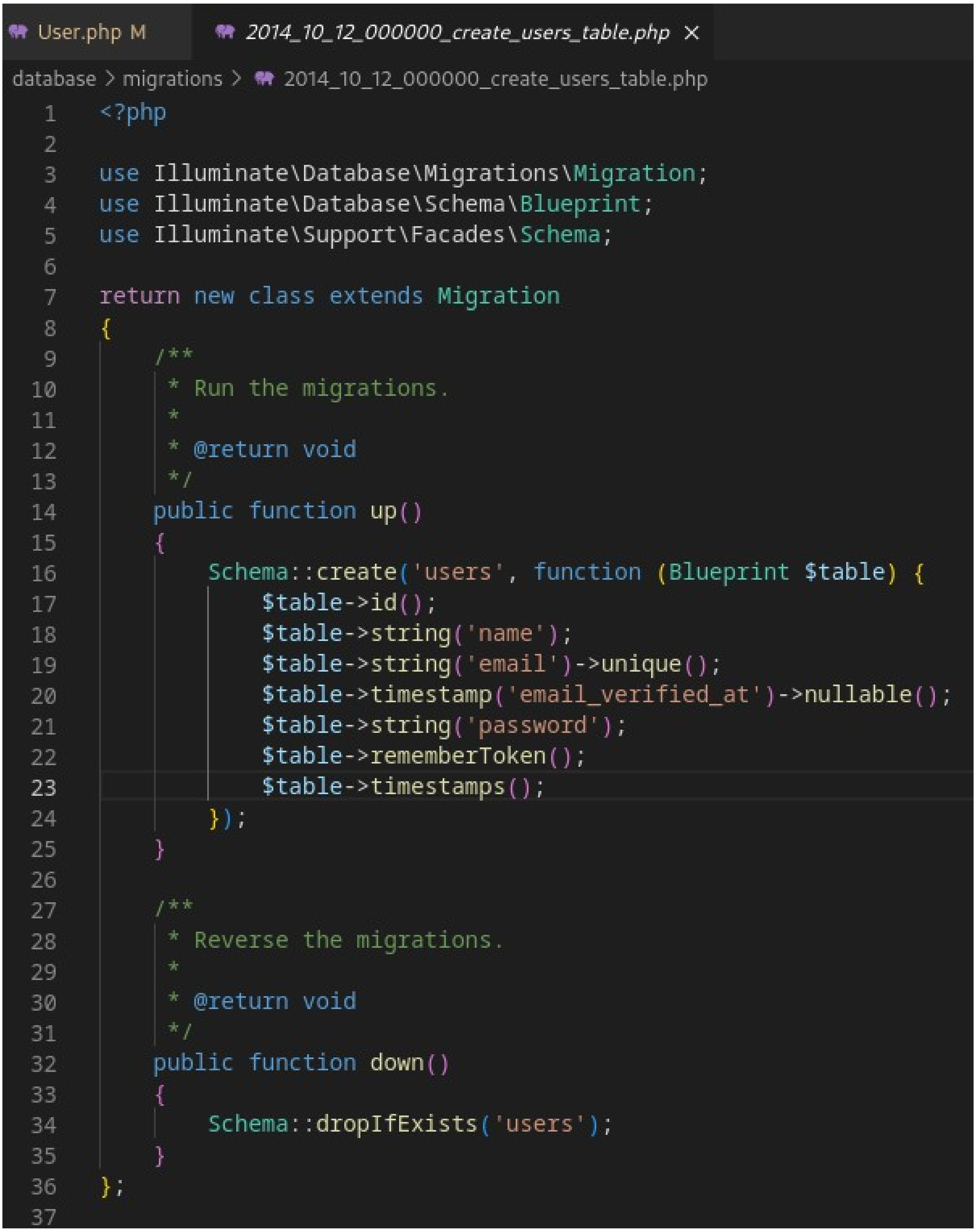


```php
<?php

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;
    public function posts() {
        return $this->hasMany(Post::class);
    }

    public function comments() {
        return $this->hasMany(Comment::class);
    }
    /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    protected $fillable = [
        'name',
        'email',
        'password',
    ];
    /**
     * The attributes that should be hidden for serialization.
     *
     * @var array<int, string>
     */
    protected $hidden = [
        'password',
        'remember_token',
    ];
    /**
     * The attributes that should be cast.
     *
     * @var array<string, string>
     */
    protected $casts = [
        'email_verified_at' => 'datetime',
    ];
}
```

Migrations:

User:

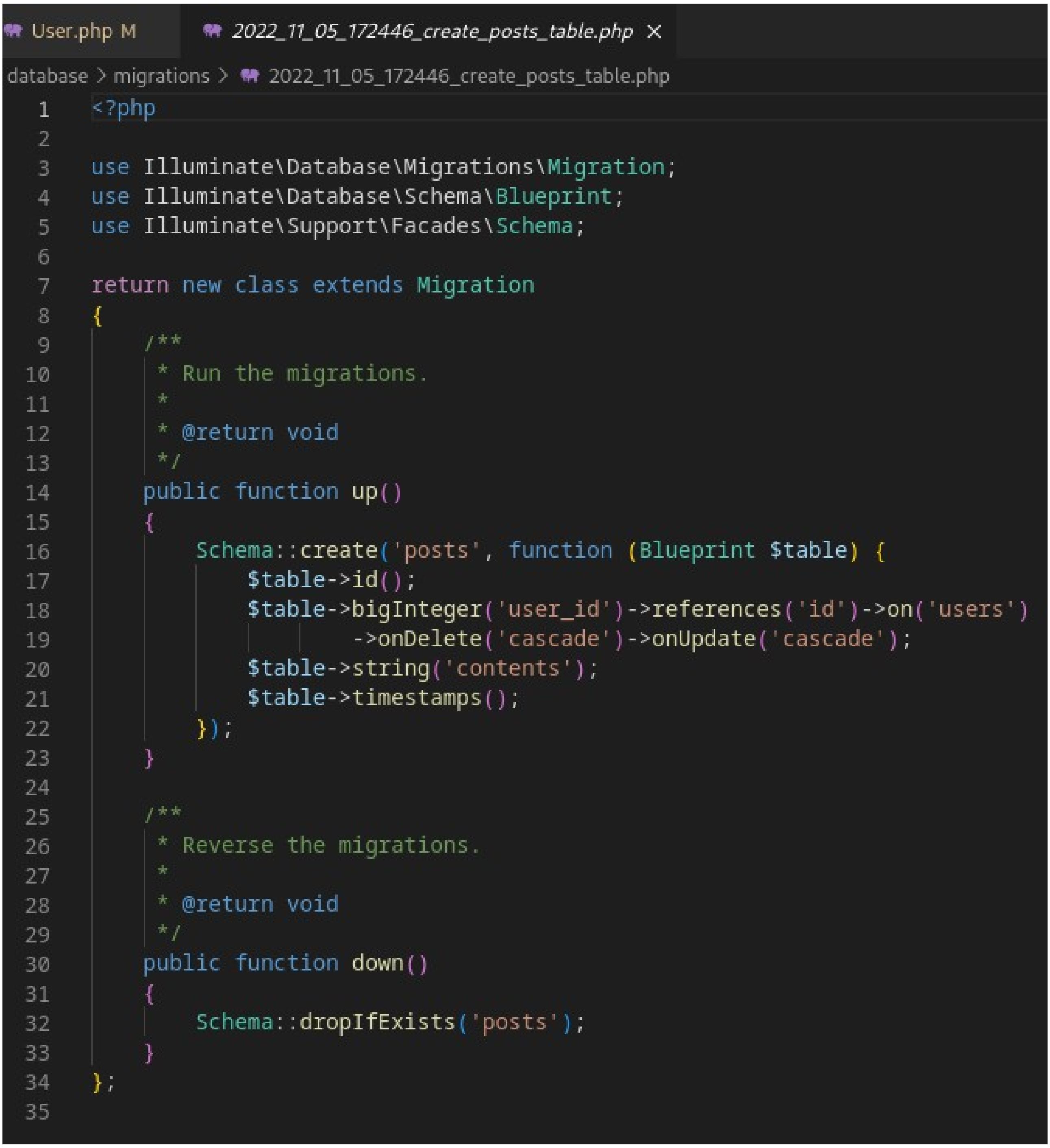
Note: This is the default users migration supplied by breeze



User.php M
2014_10_12_000000_create_users_table.php
database > migrations > 2014_10_12_000000_create_users_table.php

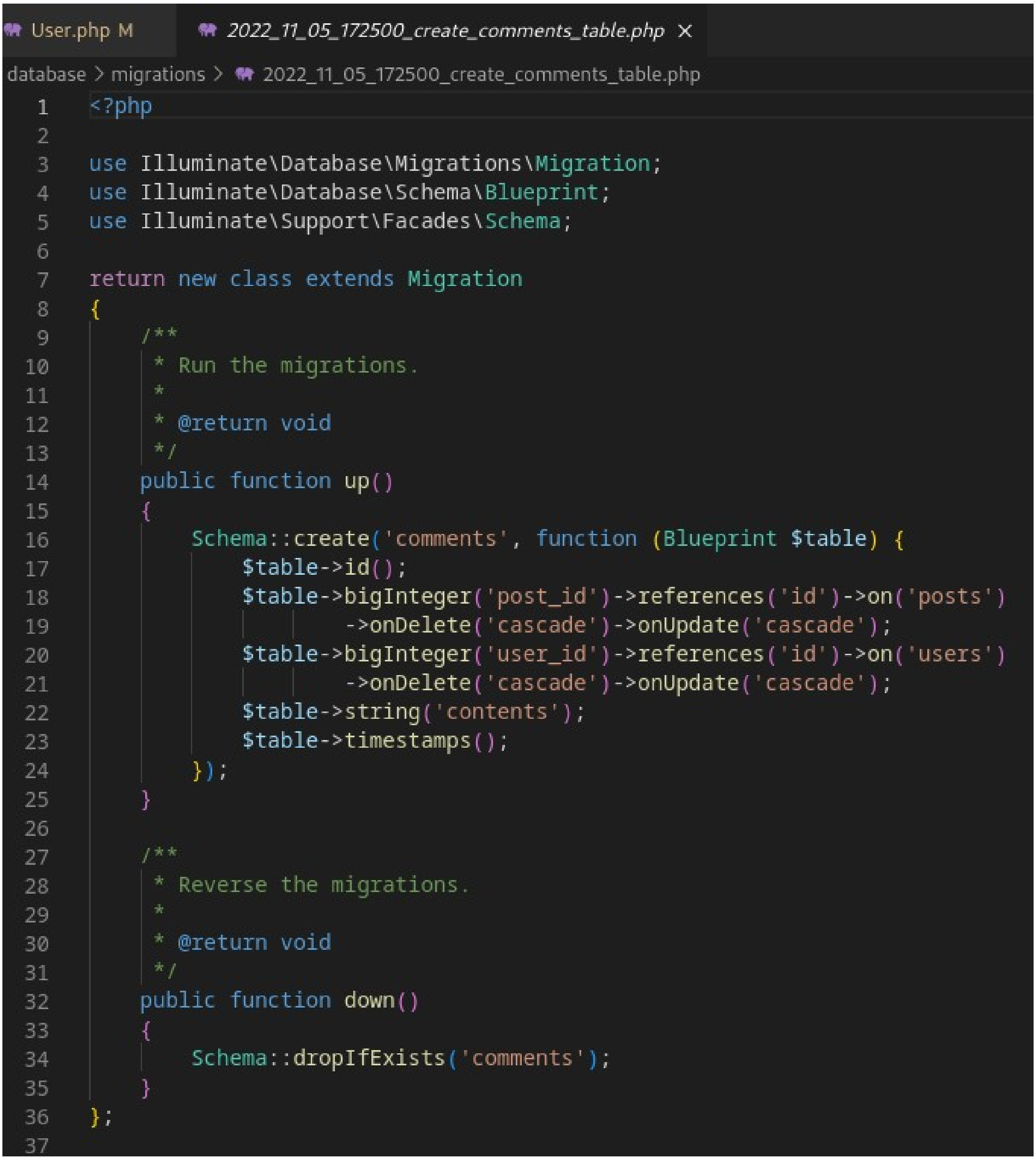
```php
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class 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->rememberToken();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('users');
    }
};

```

Posts:


User.php M
2022_11_05_172446_create_posts_table.php
database > migrations > 2022_11_05_172446_create_posts_table.php

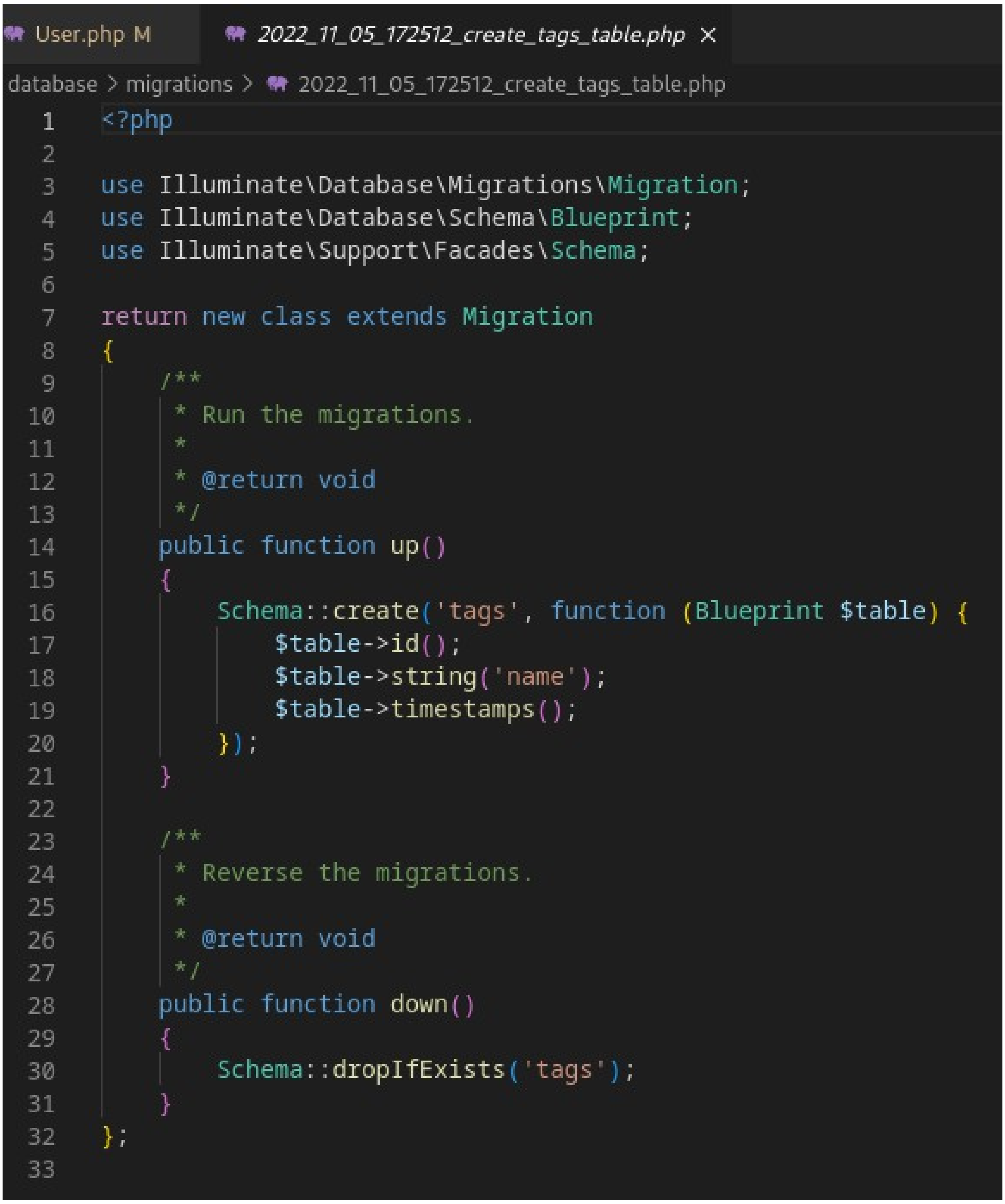
```php
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('posts', function (Blueprint $table) {
            $table->id();
            $table->bigInteger('user_id')->references('id')->on('users')
                    ->onDelete('cascade')->onUpdate('cascade');
            $table->string('contents');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('posts');
    }
};
```

Comments:


User.php M
2022_11_05_172500_create_comments_table.php ×
database > migrations > 2022_11_05_172500_create_comments_table.php

```php
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('comments', function (Blueprint $table) {
            $table->id();
            $table->bigInteger('post_id')->references('id')->on('posts')
                    ->onDelete('cascade')->onUpdate('cascade');
            $table->bigInteger('user_id')->references('id')->on('users')
                    ->onDelete('cascade')->onUpdate('cascade');
            $table->string('contents');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('comments');
    }
};

```

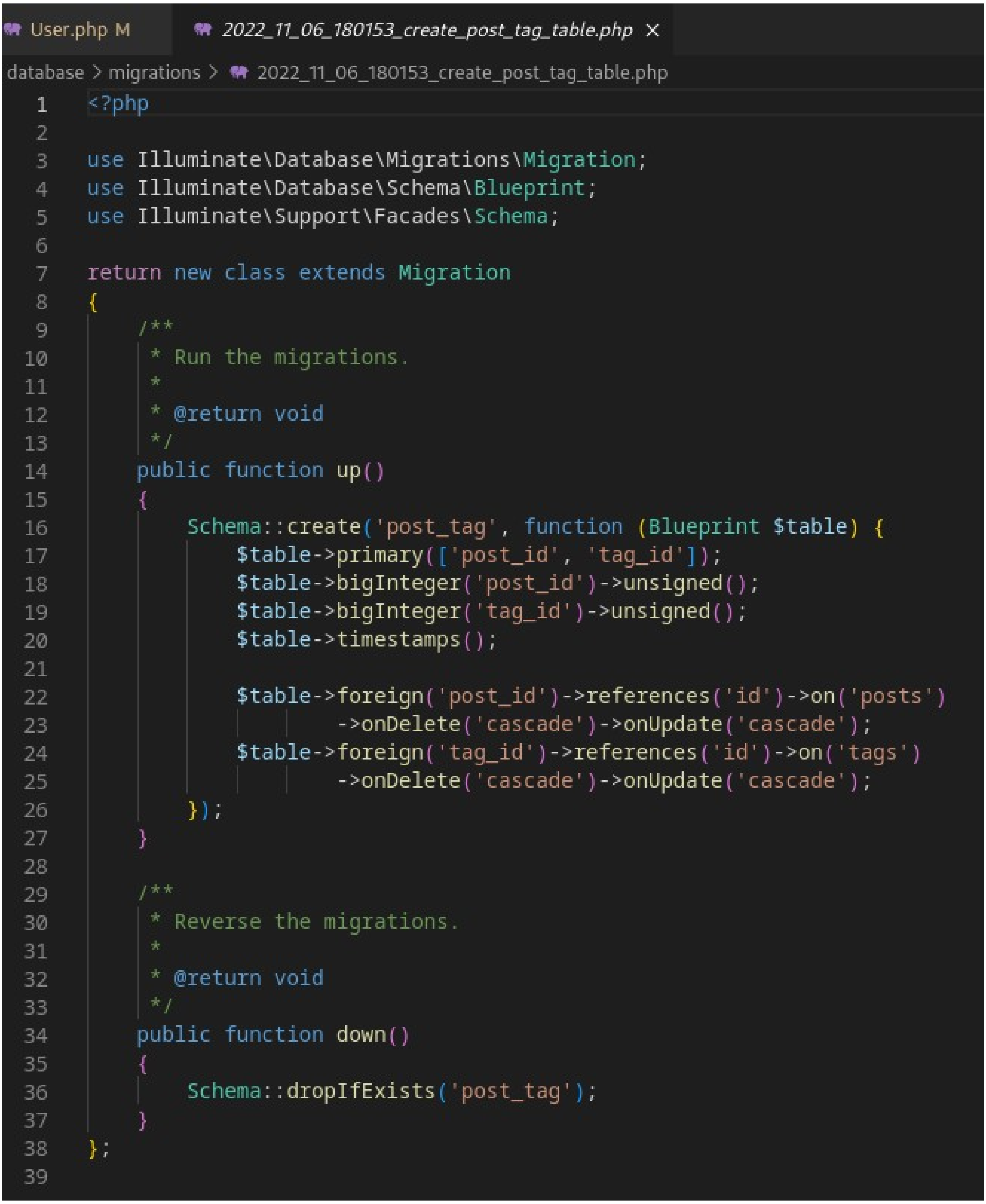
Tags:



User.php M
2022_11_05_172512_create_tags_table.php
database > migrations > 2022_11_05_172512_create_tags_table.php

```php
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('tags', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('tags');
    }
};
```

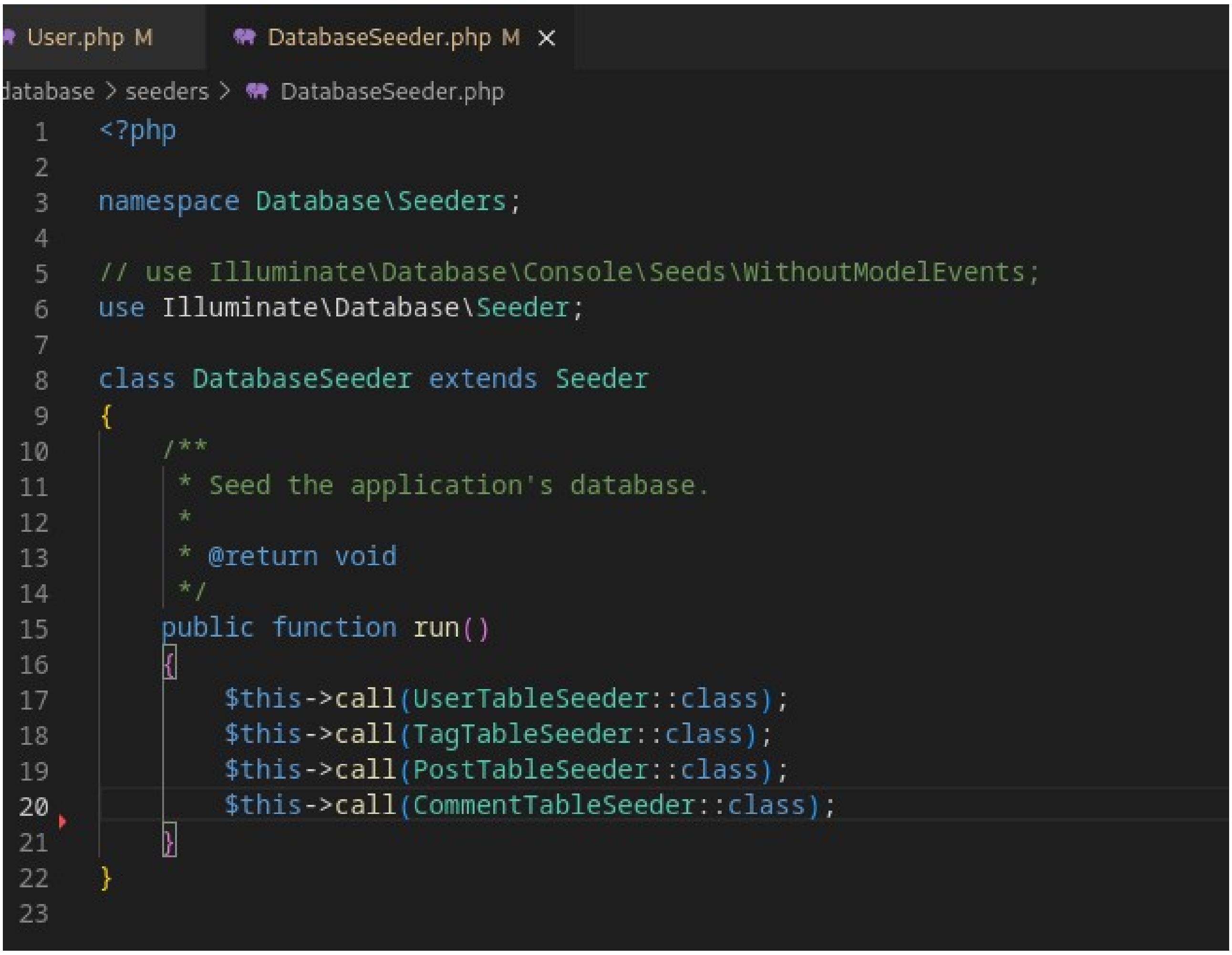
Post tag pivot table:


User.php M
2022_11_06_180153_create_post_tag_table.php
database > migrations > 2022_11_06_180153_create_post_tag_table.php

```php
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('post_tag', function (Blueprint $table) {
            $table->primary(['post_id', 'tag_id']);
            $table->bigInteger('post_id')->unsigned();
            $table->bigInteger('tag_id')->unsigned();
            $table->timestamps();

            $table->foreign('post_id')->references('id')->on('posts')
                ->onDelete('cascade')->onUpdate('cascade');
            $table->foreign('tag_id')->references('id')->on('tags')
                ->onDelete('cascade')->onUpdate('cascade');
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('post_tag');
    }
};

```

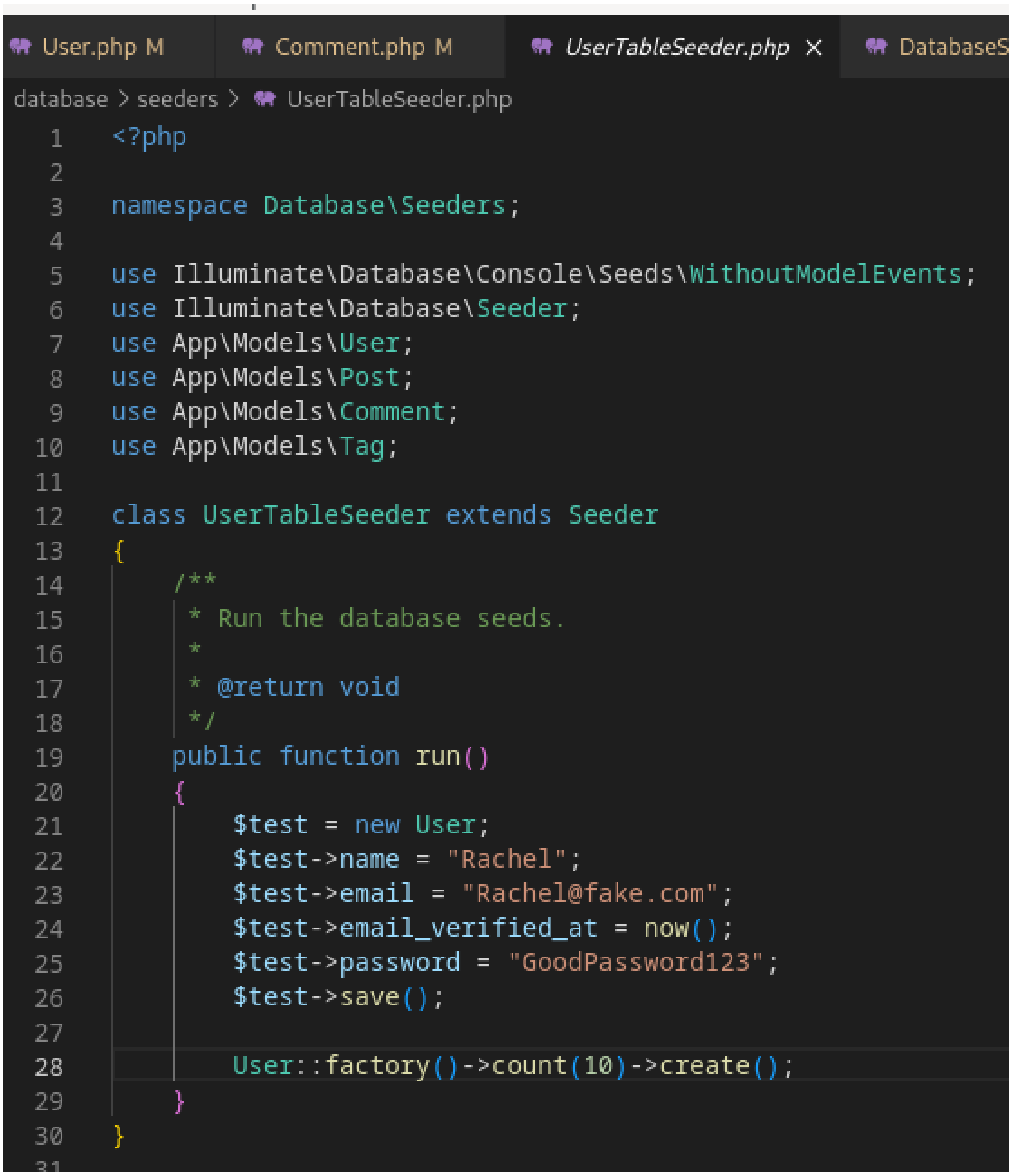
Seeders

Database seeder:



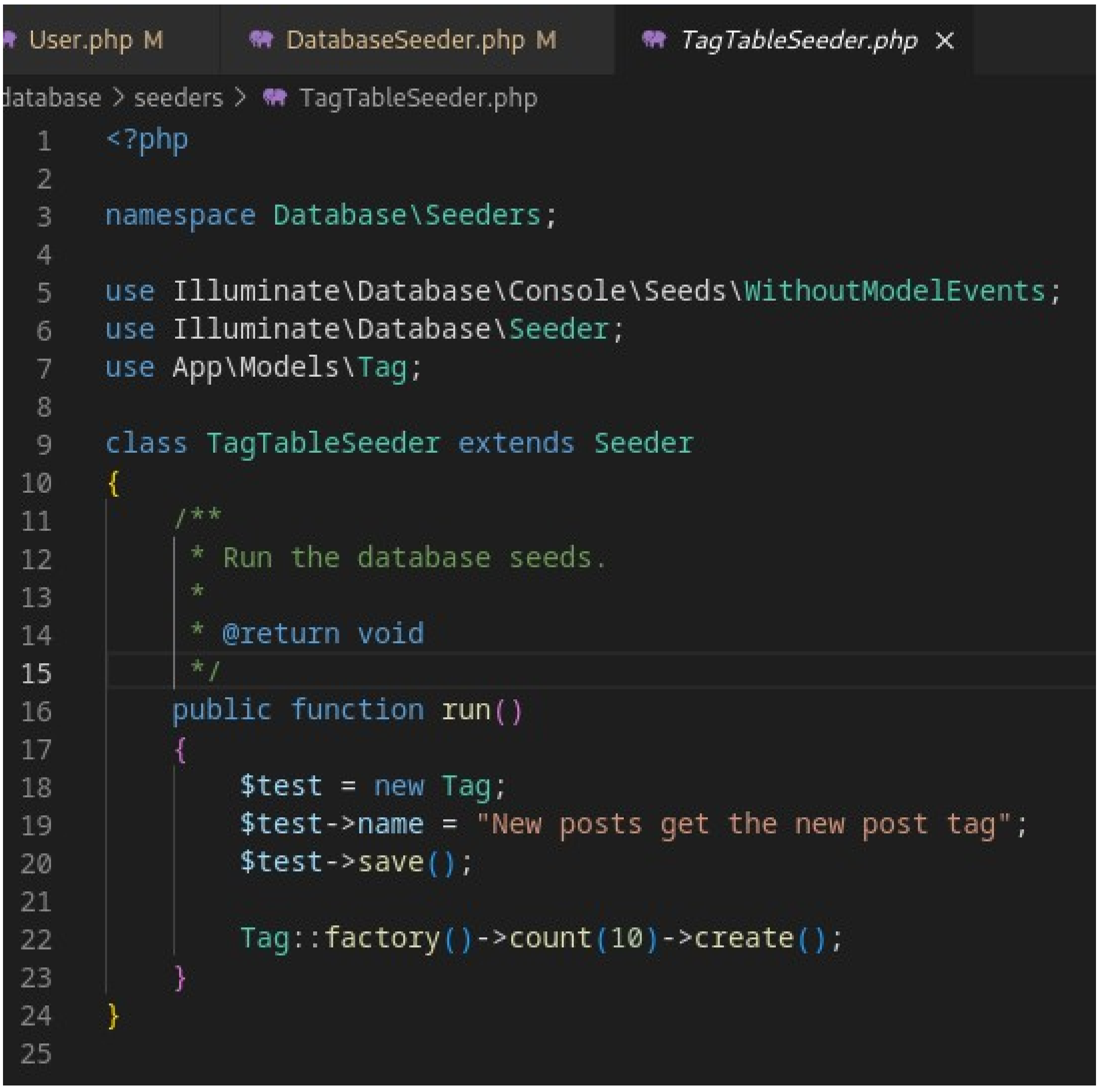
```php
<?php

namespace Database\Seeders;

// use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;

class DatabaseSeeder extends Seeder
{
    /**
     * Seed the application's database.
     *
     * @return void
     */
    public function run()
    {
        $this->call(UserTableSeeder::class);
        $this->call(TagTableSeeder::class);
        $this->call(PostTableSeeder::class);
        $this->call(CommentTableSeeder::class);
    }
}

```

User table seeder:



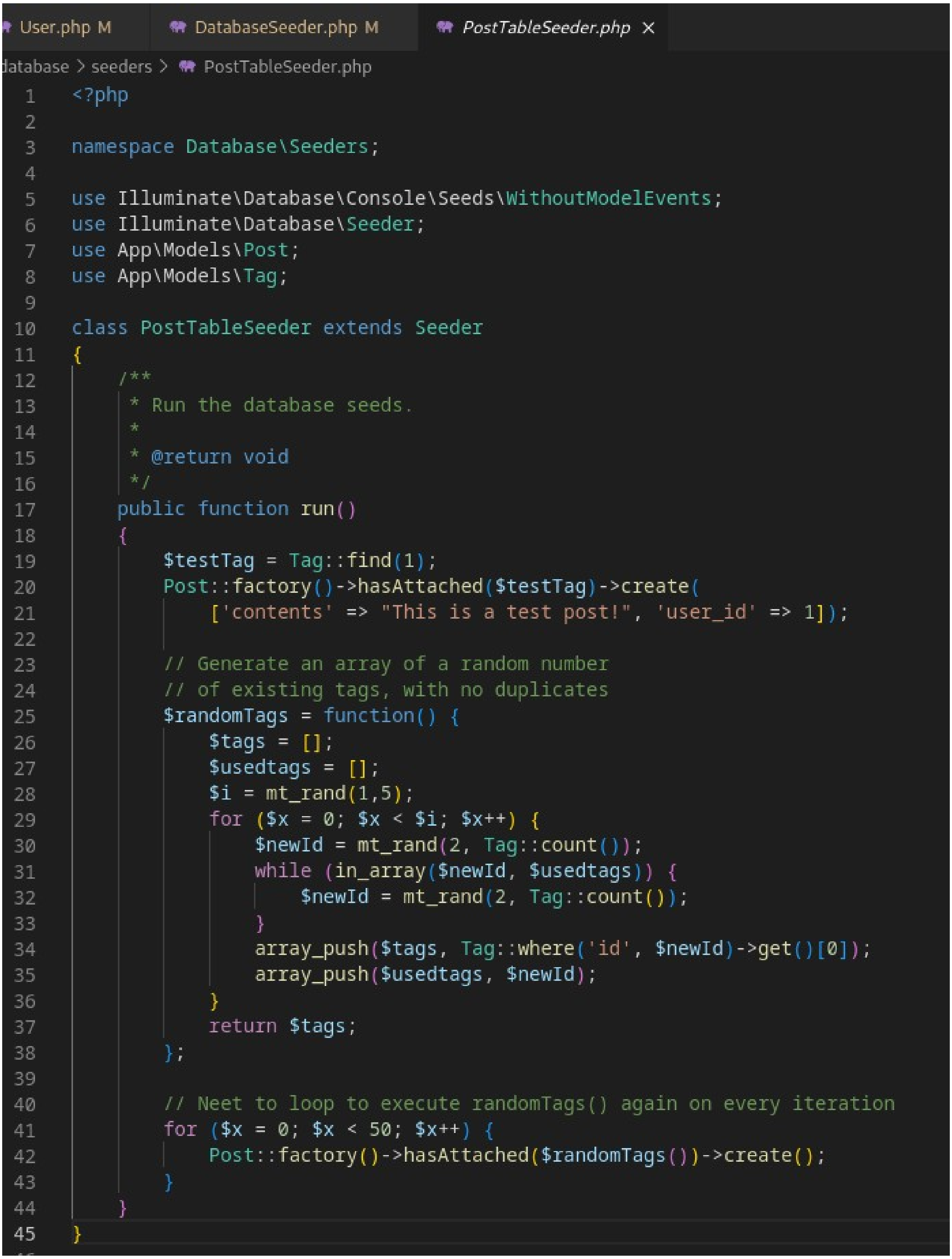
```php
<?php

namespace Database\Seeders;

use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;
use App\Models\User;
use App\Models\Post;
use App\Models\Comment;
use App\Models\Tag;

class UserTableSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        $test = new User;
        $test->name = "Rachel";
        $test->email = "Rachel@fake.com";
        $test->email_verified_at = now();
        $test->password = "GoodPassword123";
        $test->save();

        User::factory()->count(10)->create();
    }
}
```

Tag table seeder:



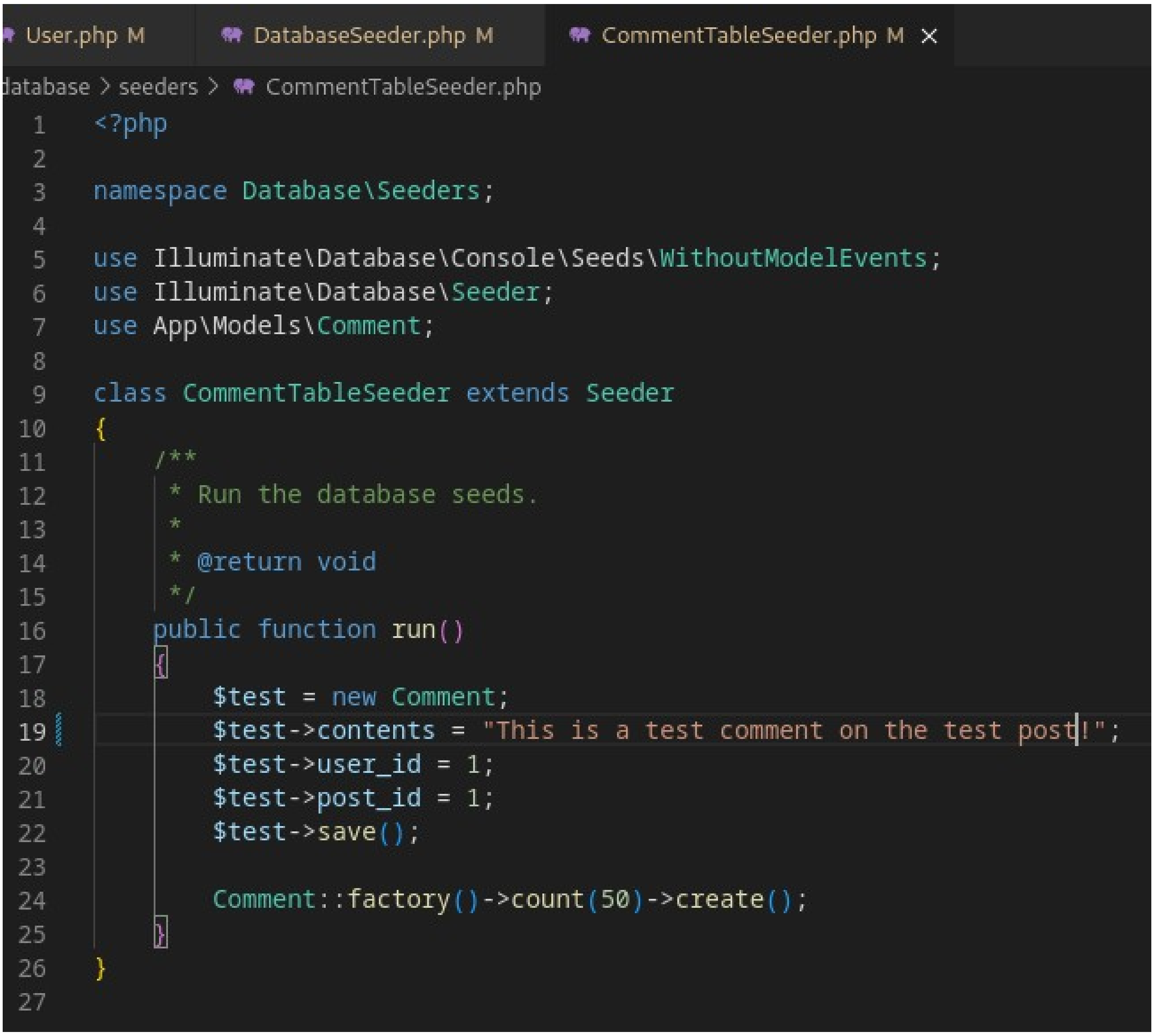
```php
<?php

namespace Database\Seeders;

use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;
use App\Models\Tag;

class TagTableSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        $test = new Tag;
        $test->name = "New posts get the new post tag";
        $test->save();

        Tag::factory()->count(10)->create();
    }
}
```

Post table seeder:



```php
<?php

namespace Database\Seeders;

use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;
use App\Models\Post;
use App\Models\Tag;

class PostTableSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        $testTag = Tag::find(1);
        Post::factory()->hasAttached($testTag)->create(
            ['contents' => "This is a test post!", 'user_id' => 1]);

        // Generate an array of a random number
        // of existing tags, with no duplicates
        $randomTags = function() {
            $tags = [];
            $usedtags = [];
            $i = mt_rand(1,5);
            for ($x = 0; $x < $i; $x++) {
                $newId = mt_rand(2, Tag::count());
                while (in_array($newId, $usedtags)) {
                    $newId = mt_rand(2, Tag::count());
                }
                array_push($tags, Tag::where('id', $newId)->get()[0]);
                array_push($usedtags, $newId);
            }
            return $tags;
        };

        // Neet to loop to execute randomTags() again on every iteration
        for ($x = 0; $x < 50; $x++) {
            Post::factory()->hasAttached($randomTags())->create();
        }
    }
}
```

Comment table seeder:

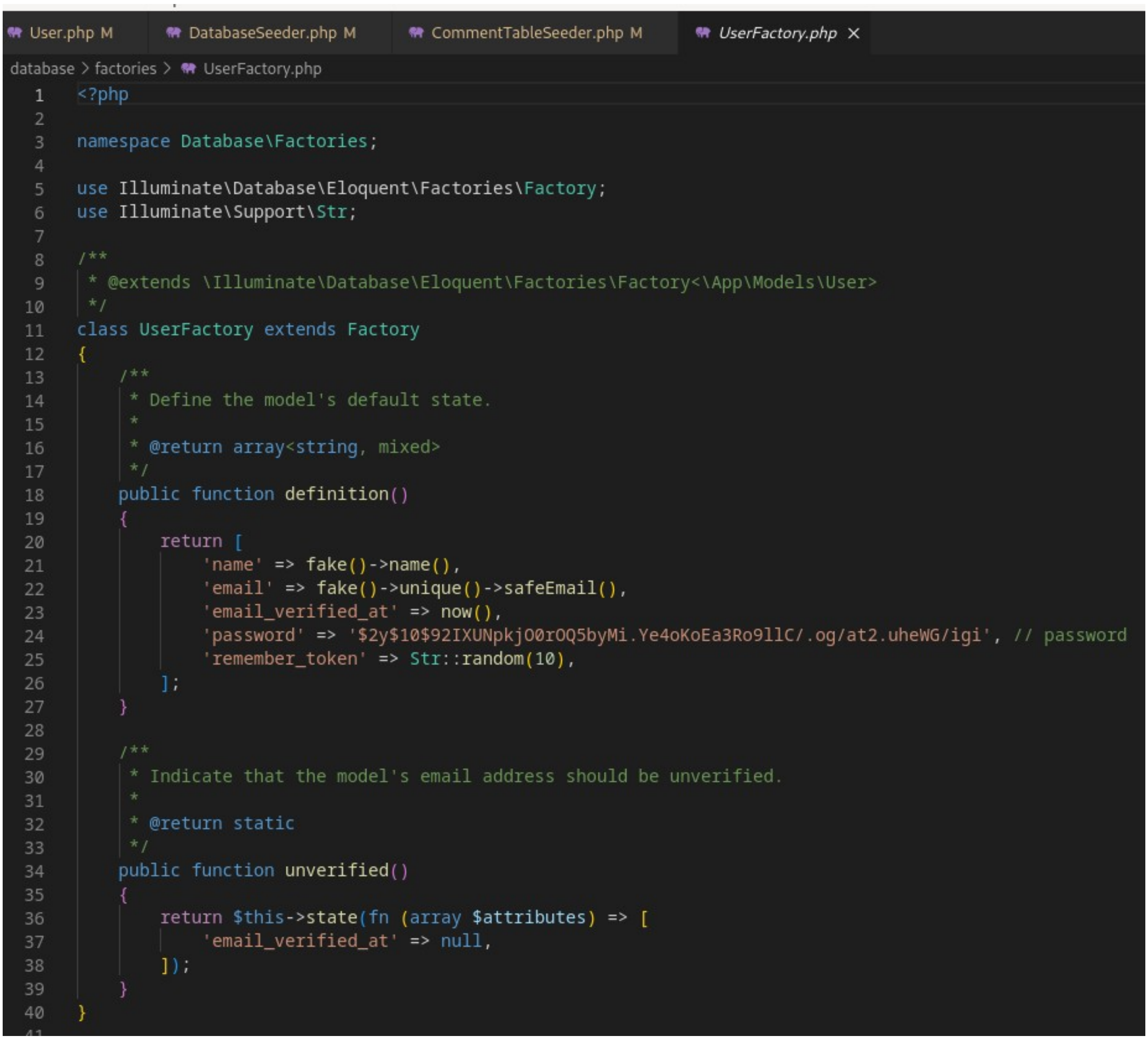

User.php M
DatabaseSeeder.php M
CommentTableSeeder.php M ×
database > seeders > CommentTableSeeder.php

```php
<?php

namespace Database\Seeders;

use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;
use App\Models\Comment;

class CommentTableSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        $test = new Comment;
        $test->contents = "This is a test comment on the test post!";
        $test->user_id = 1;
        $test->post_id = 1;
        $test->save();

        Comment::factory()->count(50)->create();
    }
}

```

Factories

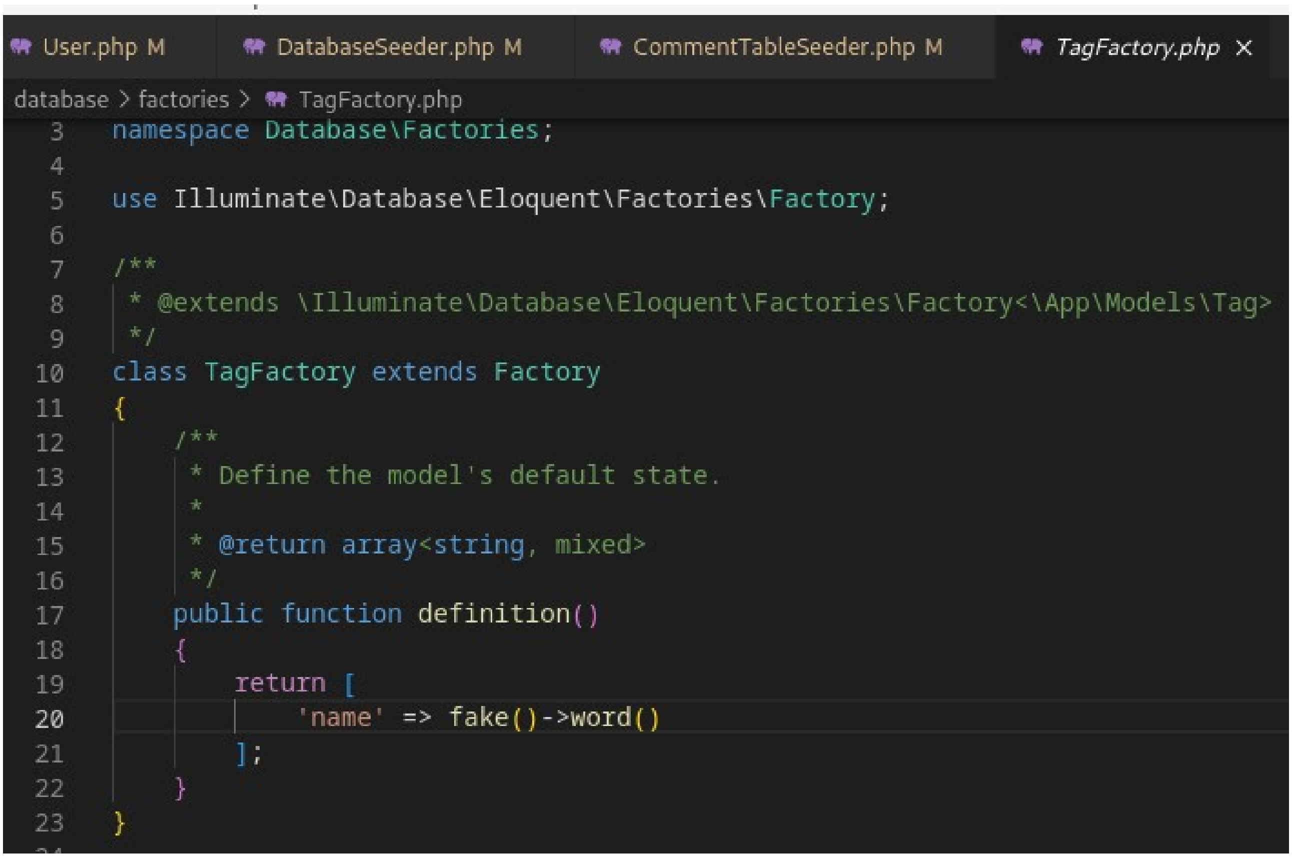
User factory:

Note: This is the default user factory provided. I have not modified it.



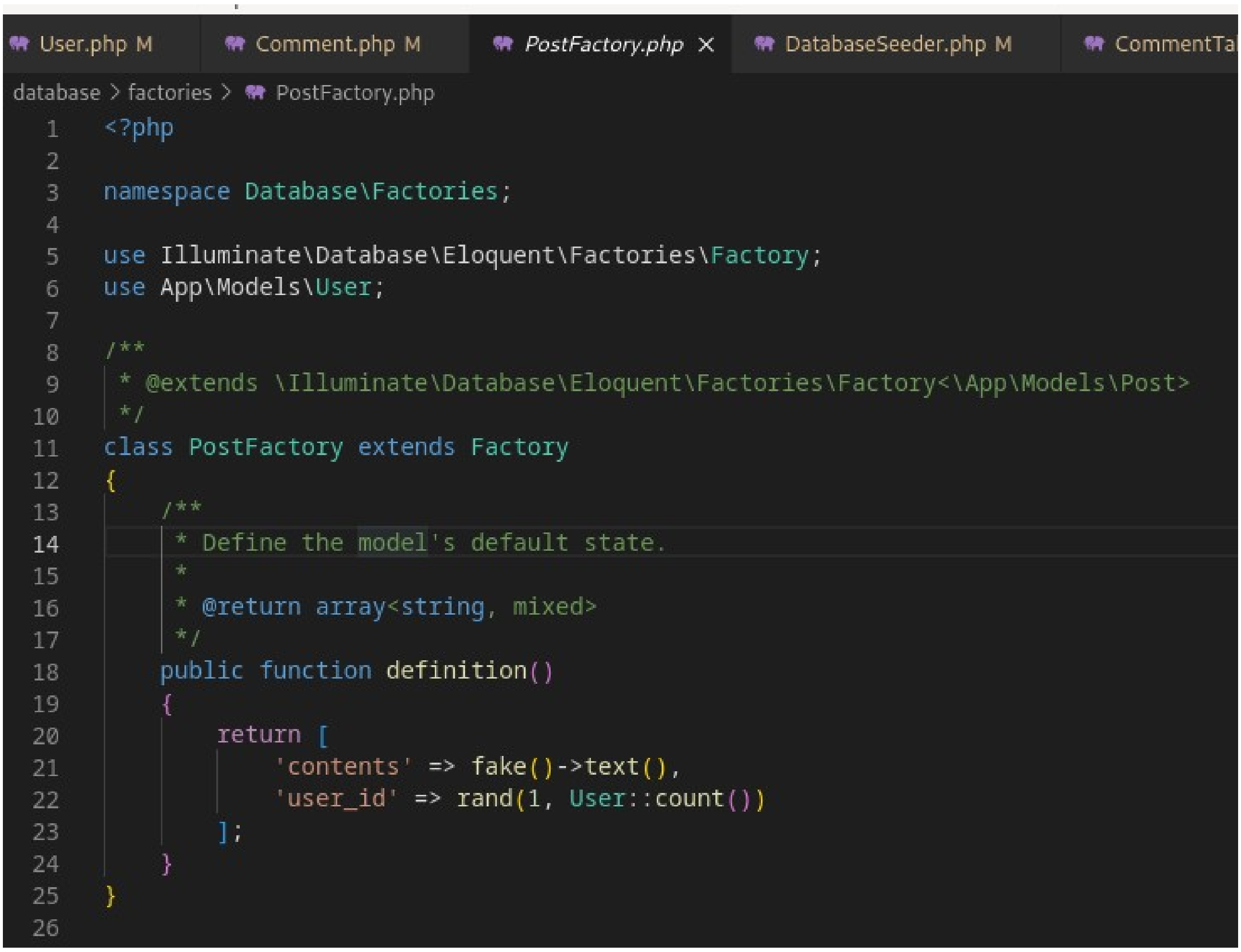
```php
<?php

namespace Database\Factories;

use Illuminate\Database\Eloquent\Factories\Factory;
use Illuminate\Support\Str;

/**
 * @extends \Illuminate\Database\Eloquent\Factories\Factory<\App\Models\User>
 */
class UserFactory extends Factory
{
    /**
     * Define the model's default state.
     *
     * @return array<string, mixed>
     */
    public function definition()
    {
        return [
            'name' => fake()->name(),
            'email' => fake()->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 static
     */
    public function unverified()
    {
        return $this->state(fn (array $attributes) => [
            'email_verified_at' => null,
        ]);
    }
}
```

Tag factory:



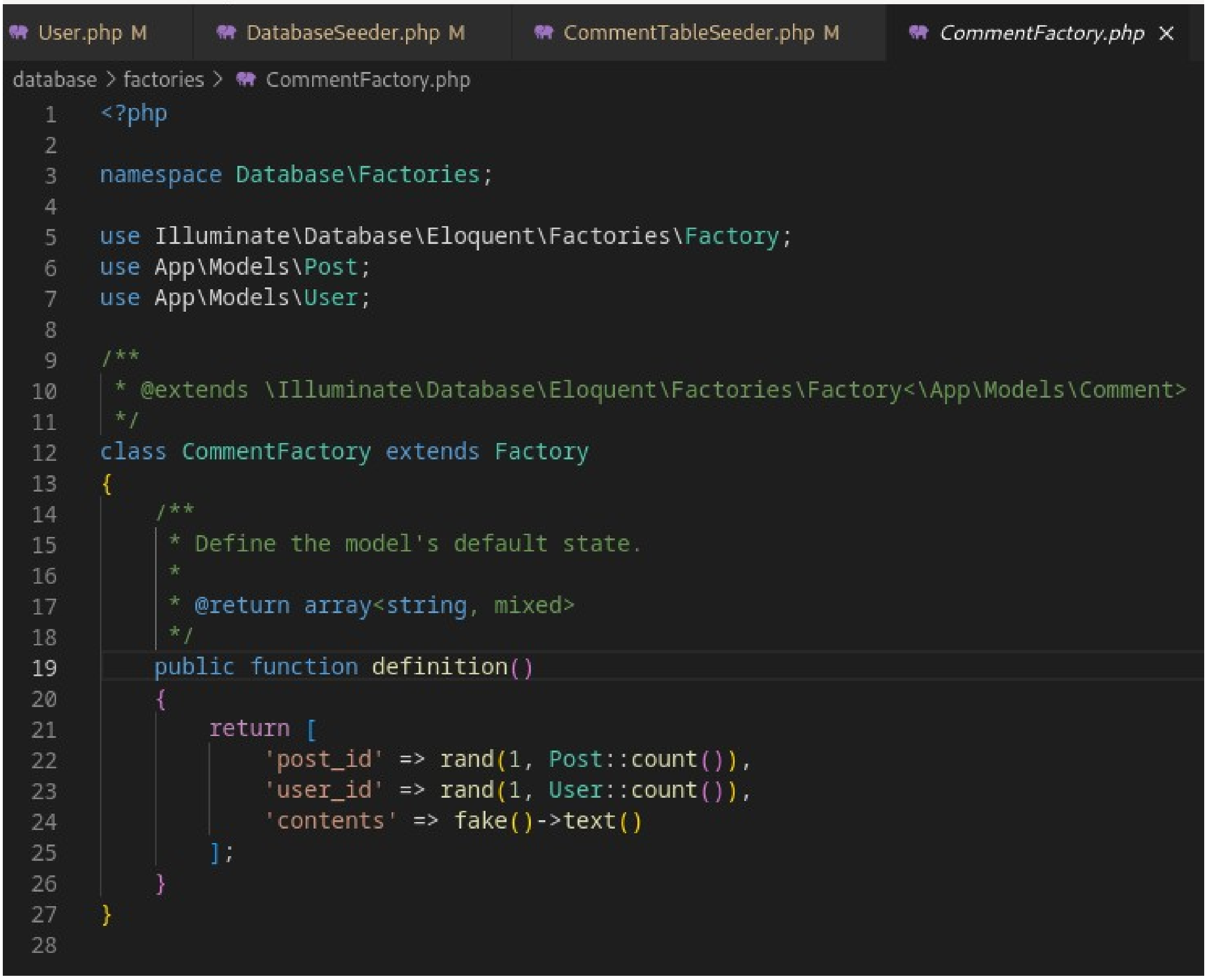
```php
namespace Database\Factories;

use Illuminate\Database\Eloquent\Factories\Factory;

/**
 * @extends \Illuminate\Database\Eloquent\Factories\Factory<\App\Models\Tag>
 */
class TagFactory extends Factory
{
    /**
     * Define the model's default state.
     *
     * @return array<string, mixed>
     */
    public function definition()
    {
        return [
            'name' => fake()->word()
        ];
    }
}
```

Post factory:



```php
<?php

namespace Database\Factories;

use Illuminate\Database\Eloquent\Factories\Factory;
use App\Models\User;

/**
 * @extends \Illuminate\Database\Eloquent\Factories\Factory<\App\Models\Post>
 */
class PostFactory extends Factory
{
    /**
     * Define the model's default state.
     *
     * @return array<string, mixed>
     */
    public function definition()
    {
        return [
            'contents' => fake()->text(),
            'user_id' => rand(1, User::count())
        ];
    }
}

```

Comment factory:



```php
<?php

namespace Database\Factories;

use Illuminate\Database\Eloquent\Factories\Factory;
use App\Models\Post;
use App\Models\User;

/**
 * @extends \Illuminate\Database\Eloquent\Factories\Factory<\App\Models\Comment>
 */
class CommentFactory extends Factory
{
    /**
     * Define the model's default state.
     *
     * @return array<string, mixed>
     */
    public function definition()
    {
        return [
            'post_id' => rand(1, Post::count()),
            'user_id' => rand(1, User::count()),
            'contents' => fake()->text()
        ];
    }
}

```