348 A1 submission PDF

Contents

Models

User:

```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;

class User extends Authenticatable
{
    /** @use HasFactory<\Database\Factories\UserFactory> */
    use HasFactory, Notifiable;

    /**
     * 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',
    ];

    /**
     * Get the attributes that should be cast.
     *
     * @return array<string, string>
     */
    protected function casts(): array
    {
        return [
            'email_verified_at' => 'datetime',
            'password' => 'hashed',
        ];
    }

    /**
     * Creation of relationships within users
     * uses all models which interact with user and in which way i.e. many,
belongs to.
     */

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

    }

    public function comments()
    {
        return $this->hasMany(Comments::class);

    }

    public function likes()
    {
        return $this->hasMany(Likes::class);

    }
}
```

Comments:

```php
<?php

namespace App\Models;

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

class Comments extends Model
{
    use HasFactory;

    /**
     * Creation of relationships within users
     * uses all models which interact with user and in which way i.e. many,
belongs to.
     */

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

    }

    public function post()
    {
        return $this->belongsTo(Posts::class);

    }
}
```

Posts:

```php
<?php

namespace App\Models;

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

class Posts extends Model
{
    use HasFactory;

    /**
     * Creation of relationships within users
     * uses all models which interact with user and in which way i.e. many,
belongs to.
     */

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

    }

    public function comments()
    {
        return $this->hasMany(Comments::class);

    }

    public function likes()
    {
        return $this->hasMany(Likes::class);

    }
}
```

Likes:

```php
<?php

namespace App\Models;

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

class Likes extends Model
{
    use HasFactory;

    /**
     * Creation of relationships within users
     * uses all models which interact with user and in which way i.e. many, 
belongs to.
     */

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

    }

    public function posts()
    {
        return $this->belongsTo(Posts::class);

    }
}
```

Migrations

Users table:

```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.
     */
    public function up(): void
    {
        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();
        });

        Schema::create('password_reset_tokens', function (Blueprint $table) {
            $table->string('email')->primary();
            $table->string('token');
            $table->timestamp('created_at')->nullable();
        });

        Schema::create('sessions', function (Blueprint $table) {
            $table->string('id')->primary();
            $table->foreignId('user_id')->nullable()->index();
            $table->string('ip_address', 45)->nullable();
            $table->text('user_agent')->nullable();
            $table->longText('payload');
            $table->integer('last_activity')->index();
        });
    }

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

Posts table:

```
<?php

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

return new class extends Migration
{
    /**
     * Run the migrations.
     */

     //develop a structuring Schema for user posts
    public function up(): void
    {
        Schema::create('posts', function (Blueprint $table) {
            $table->id();
            $table->timestamps();
            $table->string('postTitle');
            $table->string('postCaption');
            $table->bigInteger('user_id')->unsigned();
            $table->foreign('user_id')->references('id')->on('users')
            ->onDelete('cascade')->onUpdate('cascade');
        });
    }

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

Comments table:

```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.
     */

     //develop a structure schema for user comments
    public function up(): void
    {
        Schema::create('comments', function (Blueprint $table) {
            $table->id();
            $table->timestamps();
            $table->string('commentText');
            $table->bigInteger('user_id')->unsigned();
            $table->foreign('user_id')->references('id')->on('users')
            ->onDelete('cascade')->onUpdate('cascade');
            $table->bigInteger('posts_id')->unsigned();
            $table->foreign('posts_id')->references('id')->on('posts')
            ->onDelete('cascade')->onUpdate('cascade');
        });
    }

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

Likes table:

```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.
     */

     //develop a structure schema for user likes
    public function up(): void
    {
        Schema::create('likes', function (Blueprint $table) {
            $table->id();
            $table->timestamps();
            $table->bigInteger('user_id')->unsigned();
            $table->foreign('user_id')->references('id')->on('users')
            ->onDelete('cascade')->onUpdate('cascade');
            $table->bigInteger('posts_id')->unsigned();
            $table->foreign('posts_id')->references('id')->on('posts')
            ->onDelete('cascade')->onUpdate('cascade');
        });
    }

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

Seeding

Database Seeder:

```php
<?php

namespace Database\Seeders;

use App\Models\UserPosts;
use App\Models\User;
// use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;

class DatabaseSeeder extends Seeder
{
    /**
     * Seed the application's database.
     */
    public function run(): void
    {
        User::factory(10)->create();

        // User::factory()->create([
        //     'name' => 'Test User',
          //   'email' => 'test@example.com',
        //]);

      //call all necessary User related class seeders
       $this -> call(UserPostSeeder::class);
       $this -> call(UserLikesSeeder::class);
       $this -> call(UserCommentsSeeder::class);
    }
}
```

UserCommentsSeeder:

```php
<?php

namespace Database\Seeders;

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

class UserCommentsSeeder extends Seeder
{
    /**
     * Run the database seeder for UserComments using its parameters.
     */
    public function run(): void
    {
        $u = new Comments();
        $u->user_id = 1;
        $u->commentText = "First commment";
        $u->posts_id = 1;
        $u->save();
        $Comments = Comments::factory() -> count(20) -> create(); //produces a
random seeder of length 20 sample data
    }
}
```

UserLikesSeeder:

```php
<?php

namespace Database\Seeders;

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

class UserLikesSeeder extends Seeder
{
    /**
     * Run the database seeder for UserLikes using its parameters.
     */
    public function run(): void
    {
        $u = new Likes();
        $u->user_id = 1;
        $u->posts_id = 1;
        $u->save();
       $Likes = Likes::factory() -> count(20) -> create(); //produces a random
seeder of length 20 sample data
    }
}
```

UserPostSeeder:

```php
<?php

namespace Database\Seeders;

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

class UserPostSeeder extends Seeder
{
    /**
     * Run the database seeder for UserPosts using its parameters.
     */
    public function run(): void
    {
        $u = new Posts();

        $u->postTitle = "First post";
        $u->postCaption = "This is my first caption";
        $u->user_id = 3;
        $u->save();
        $Posts = Posts::factory() -> count(20) -> create(); //produces a
seeder of length 20 sample data
    }
}
```

Factories

User Factory:

```php
<?php

namespace Database\Factories;

use Illuminate\Database\Eloquent\Factories\Factory;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Str;
use App\Models\UserPosts;
/**
 * @extends \Illuminate\Database\Eloquent\Factories\Factory<\App\Models\User>
 */
class UserFactory extends Factory
{
    /**
     * The current password being used by the factory.
     */
    protected static ?string $password;

    /**
     * Define the model's default state.
     *
     * @return array<string, mixed>
     */
    public function definition(): array
    {
        return [
            'name' => fake()->name(), //assign a fake name for user
            'email' => fake()->unique()->safeEmail(), //assign a fake email
for user
            'email_verified_at' => now(), //verify
            'password' => static::$password ??= Hash::make('password'),
//password hash
            'remember_token' => Str::random(10), //password token
        ];
    }

    /**
     * Indicate that the model's email address should be unverified.
     */
    public function unverified(): static
    {
        return $this->state(fn (array $attributes) => [
            'email_verified_at' => null,
        ]);
    }
}
```

Comments Factory:

```
<?php

namespace Database\Factories;

use Illuminate\Database\Eloquent\Factories\Factory;
use App\Models\UserPosts;
/**
 * @extends
\Illuminate\Database\Eloquent\Factories\Factory<\App\Models\Comments>
 */
class CommentsFactory extends Factory
{
    /**
     * Define the model's default state.
     *
     * @return array<string, mixed>
     */
    public function definition(): array
    {
        return [
            'user_id' => fake() -> randomDigitNotNull(), //produce a not null,
random id to mimic user id
            'commentText' => fake() -> sentence(), // produce a randon
sentence to mimic comments
            'posts_id' => fake() -> randomDigitNotNull(), //produce a not null
digit to represent a post id
        ];
    }
}
```

Likes Factory:

```php
<?php

namespace Database\Factories;

use Illuminate\Database\Eloquent\Factories\Factory;
use App\Models\UserPosts;
/**
 * @extends \Illuminate\Database\Eloquent\Factories\Factory<\App\Models\Likes>
 */
class LikesFactory extends Factory
{
    /**
     * Define the model's default state.
     *
     * @return array<string, mixed>
     */
    public function definition(): array
    {
        return [
            'user_id' => fake() -> randomDigitNotNull(), //produce a random
not null user id
            'posts_id' => fake() -> randomDigitNotNull(), // produce a random
not null user post id
        ];
    }
}
```

Posts Factory:

```php
<?php

namespace Database\Factories;

use Illuminate\Database\Eloquent\Factories\Factory;
use App\Models\UserPosts;
/**
 * @extends \Illuminate\Database\Eloquent\Factories\Factory<\App\Models\Posts>
 */
class PostsFactory extends Factory
{
    /**
     * Define the model's default state.
     *
     * @return array<string, mixed>
     */
    public function definition(): array
    {
        return [
            'postTitle' => fake() -> sentence(), // produce a fake title
            'postCaption' => fake() -> sentence(), //produce a fake caption
            'user_id' => fake() -> randomDigitNotNull(), //create a fake non
null user id for the post
        ];
    }
}
```