Models

User.php

```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',
        ];
    }

    public function profile()
    {
        return $this->hasOne(Profile::class);
    }

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

    public function likedPosts()
    {
        return $this->belongsToMany(Post::class, 'post_user_like');
    }

    public function notifications()
    {
        return $this->hasMany(Notification::class);
    }
}
```

Profile.php

```
<?php

namespace App\Models;

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

class Profile extends Model
{
    use HasFactory;

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

Post.php

```
<?php

namespace App\Models;

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

class Post extends Model
{
    use HasFactory;

    public function users()
    {
        return $this->belongsToMany(User::class);
    }

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

    public function userLikes()
    {
        return $this->belongsToMany(User::class, 'post_user_like');
    }
}
```

Comment.php

```
<?php

namespace App\Models;

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

class Comment extends Model
{
    use HasFactory;

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

Notification.php

```
<?php

namespace App\Models;

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

class Notification extends Model
{
    use HasFactory;

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

Factories

UserFactory.php

```
<?php

namespace Database\Factories;

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

/**
 * @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(),
            'username' => $this->faker->unique()->userName,
            'email' => fake()->unique()->safeEmail(),
            'email_verified_at' => now(),
            'password' => static::$password ??= Hash::make('password'),
            'remember_token' => Str::random(10),
            'profile_image' => $this->faker->imageUrl(600,600),
            'premium_member' => $this->faker->boolean(),
        ];
    }

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

ProfileFactory.php

```
<?php

namespace Database\Factories;

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

/**
 * @extends
\Illuminate\Database\Eloquent\Factories\Factory<\App\Models\Profile>
 */
class ProfileFactory extends Factory
{
    /**
     * Define the model's default state.
     *
     * @return array<string, mixed>
     */
    public function definition(): array
    {
        return [
            'bio' => $this->faker->sentence(20),
            'date_of_birth' => $this->faker->dateTimeBetween('-25 years',
'-20 years'),
                'banner_image' => $this->faker->imageUrl(2000,1000),
        ];
    }
}
```

PostFactory.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\Post>
 */
class PostFactory extends Factory
{
    /**
     * Define the model's default state.
     *
     * @return array<string, mixed>
     */
    public function definition(): array
    {
        return [
            'user_id' => User::inRandomOrder()->value('id'),
            'title' => $this->faker->sentence(15),
            'post_image' => $this->faker->imageUrl(1500,1500),
            'caption' => $this->faker->sentence(20),
        ];
    }
}
```

CommentFactory.php

```
<?php

namespace Database\Factories;

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

/**
 * @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(): array
    {
        return [
            'user_id' => User::inRandomOrder()->value('id'),
            'post_id' => Post::inRandomOrder()->value('id'),
            'comment_text' => $this->faker->sentence(20),
        ];
    }
}
```

NotificationFactory.php

```
<?php

namespace Database\Factories;

use Illuminate\Database\Eloquent\Factories\Factory;

use App\Models\Notification;
use App\Models\User;

/**
 * @extends
\Illuminate\Database\Eloquent\Factories\Factory<\App\Models\Notification>
 */
class NotificationFactory extends Factory
{
    /**
     * Define the model's default state.
     *
     * @return array<string, mixed>
     */
    public function definition(): array
    {
        return [
            'user_id' => User::inRandomOrder()->value('id'),
            'type' => $this->faker->randomElement(['Post', 'Like']),
            'message' => $this->faker->sentence(20),
            'has_been_read' => $this->faker->boolean(),
        ];
    }
}
```

Migrations

0001_01_01_000000_create_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('username');
            $table->string('email')->unique();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->string('profile_image');
            $table->boolean('premium_member');
            $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');
    }
};
```

2024_11_03_100516_create_profiles_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('profiles', function (Blueprint $table) {
            $table->id();
            $table->unsignedBigInteger('user_id')->unique();
            $table->string('bio')->nullable();
            $table->dateTime('date_of_birth')->nullable();
            $table->string('banner_image')->nullable();

            $table->foreign('user_id')->references('id')->on('users')
                ->onDelete('cascade')->onUpdate('cascade');
            $table->timestamps();
        });
    }

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

2024_11_03_174829_create_posts_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('posts', function (Blueprint $table) {
            $table->id();
            $table->unsignedBigInteger('user_id');
            $table->string('title');
            $table->string('post_image');
            $table->string('caption');

            $table->foreign('user_id')->references('id')->on('users')
                ->onDelete('cascade')->onUpdate('cascade');
            $table->timestamps();
        });
    }

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

2024_11_03_181249_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.
     */
    public function up(): void
    {
        Schema::create('comments', function (Blueprint $table) {
            $table->id();
            $table->unsignedBigInteger('user_id');
            $table->unsignedBigInteger('post_id');
            $table->string('comment_text');
            $table->timestamps();

            $table->foreign('user_id')->references('id')->on('users')
                ->onDelete('cascade')->onUpdate('cascade');

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

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

2024_11_03_191639_create_post_user_like_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.
     */
    public function up(): void
    {
        Schema::create('post_user_like', function (Blueprint $table) {
            $table->id();
            $table->foreignId('user_id')->constrained('users')
                ->onDelete('cascade')->onUpdate('cascade');
            $table->foreignId('post_id')->constrained('posts')
```

```
                ->onDelete('cascade')->onUpdate('cascade');
            $table->timestamps();
        });
    }

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

2024_11_03_221926_create_notifications_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('notifications', function (Blueprint $table) {
            $table->id();
            $table->unsignedBigInteger('user_id');
            $table->string('type');
            $table->string('message');
            $table->string('has_been_read');

            $table->foreign('user_id')->references('id')->on('users')
                ->onDelete('cascade')->onUpdate('cascade');
            $table->timestamps();
        });
    }

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

Seeders

DatabaseSeeder.php

```
<?php
```

```php
namespace Database\Seeders;

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
    {
        $this->call(UserTableSeeder::class);
        $this->call(ProfileTableSeeder::class);
        $this->call(PostTableSeeder::class);
        $this->call(CommentTableSeeder::class);
        $this->call(LikeTableSeeder::class);
        $this->call(NotificationTableSeeder::class);
    }
}
```

UserTableSeeder.php

```php
<?php

namespace Database\Seeders;

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

class UserTableSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        User::factory()
        ->hasPosts(3)
        ->count(50)->create();
    }
}
```

ProfileTableSeeder.php

```php
<?php

namespace Database\Seeders;

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

class ProfileTableSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        $allUserIds = User::pluck('id');

        foreach ($allUserIds as $currentUserId) {
            Profile::factory()->create([
                'user_id' => $currentUserId,
            ]);
        }
    }
}
```

PostTableSeeder.php

```
<?php

namespace Database\Seeders;

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

class PostTableSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        Post::factory()->count(100)->create();
    }
}
```

CommentTableSeeder.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.
     */
    public function run(): void
    {
        Comment::factory()->count(25)->create();
    }
}
```

LikeTableSeeder.php

```php
<?php

namespace Database\Seeders;

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

use App\Models\User;
use App\Models\Post;

class LikeTableSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        $allUsers = User::all();
        $allPosts = Post::all();

        foreach ($allUsers as $user) {
            $listOfPosts = $allPosts->random(rand(1, 5));

            $user->likedPosts()->attach($listOfPosts->pluck('id')-
>toArray());
        }
    }
}
```

NotificationTableSeeder.php

```php
<?php

namespace Database\Seeders;

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

class NotificationTableSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        Notification::factory()->count(100)->create();
    }
}
```