Database relationship and seeding

1. Migrate:

1) Create_Cat_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.
     */     public function
up(): void
    {
        Schema::create('cats', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('country');
            $table->double('theLengthOfLife');
            $table->string('color');
            $table->foreignId('breed_id')->constrained('breeds')->onDele
te('cascade');
            $table->timestamps();
        });
    }

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

2) create_dog_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.
     */     public function
up(): void
    {
        Schema::create('dogs', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('country');
            $table->double('theLengthOfLife');
            $table->string('color');
            $table->foreignId('breed_id')->constrained('breeds')->onDele
te('cascade');
            $table->timestamps();
        });
    }

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

3) create_breeds_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.
     */   public function
up(): void
   {
        Schema::create('breeds', function (Blueprint $table) {
```

```
      $table->id();
      $table->string('name');
      $table->string('type');
      $table->string('country');
      $table->double('average_lifespan');
      $table->string('color')->nullable();
      $table->text('characteristics')->nullable();
      $table->timestamps();
    });
  }

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

4) create_PetSupplies_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.
   */     public function
up(): void
  {
    Schema::create('pet_supplies', function (Blueprint $table) {
      $table->id();
     $table->string('name');
    $table->string('type');
      $table->double('price');
  $table->text('description')->nullable();
     $table->timestamps();
   });
  }

  /**
```

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

5) create_cat_pet_supplies_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.
     */     public function
up(): void
    {
        Schema::create('cat_pet_supplies', function (Blueprint $table) {
            $table->id();
            $table->foreignId('cat_id')->constrained()->onDelete('cascad
e');
            $table->foreignId('pet_supply_id')->constrained()->onDelete(
'cascade');
            $table->timestamps();
        });
    }

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

6) create_dog_pet_supplies_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.
     */     public function
up(): void
    {
        Schema::create('dog_pet_supplies', function (Blueprint $table) {
            $table->id();
            $table->foreignId('dog_id')->constrained('dogs')->onDelete('
cascade');
            $table->foreignId('pet_supplies_id')->constrained('pet_suppl
ies')->onDelete('cascade');
            $table->timestamps();
        });
    }

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

7) create_comments_table:

```
<?php  use
Illuminate\Database\Migrations\Migration; use
Illuminate\Database\Schema\Blueprint; use
Illuminate\Support\Facades\Schema;
use Illuminate\Support\Facades\DB; return
new class extends Migration
{
    /**
     * Run the migrations.
     */     public function
up(): void
    {

        DB::statement('SET FOREIGN_KEY_CHECKS=0;');
        Schema::dropIfExists('comments');

        Schema::create('comments', function (Blueprint $table) {
            $table->id();
            $table->text('content');
            $table->foreignId('user_id')
                  ->constrained('users')
                  ->onDelete('cascade');
            $table->foreignId('post_id')
                  ->constrained('posts')
                  ->onDelete('cascade');
            $table->timestamps();
        });
        DB::statement('SET FOREIGN_KEY_CHECKS=1;');
    }

    /**
     * Reverse the migrations.
     */     public function
down(): void
    {
        DB::statement('SET FOREIGN_KEY_CHECKS=0;');

        Schema::dropIfExists('comments');

        DB::statement('SET FOREIGN_KEY_CHECKS=1;');

    }
};
```

8) create_post_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.
     */     public function
up(): void
```

```
    {
        Schema::create('posts', function (Blueprint $table) {
            $table->id();
            $table->string('title');
             $table->text('content');
            $table->foreignId('user_id')
                  ->constrained('users')
                  ->onDelete('cascade');
            $table->foreignId('cat_id')
                  ->nullable()
                  ->constrained('cats')
                  ->onDelete('set null');
            $table->foreignId('dog_id')
                  ->nullable()
                  ->constrained('dogs')
                  ->onDelete('set null');
            $table->timestamps();
        });
    }

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

9) create_user_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.
     */     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();
        });
    }

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

2.

models:

1) Cat:

```
<?php  namespace
App\Models;
 use
Illuminate\Database\Eloquent\Factories\HasFactory; use
Illuminate\Database\Eloquent\Model;
 class Cat extends
Model
{     protected $table = 'cats';     use HasFactory;
protected $fillable = ['name', 'country', 'theLengthOfLife',
'color', 'created_at', 'updated_at'];


public function petSupplies()
{     return $this->belongsToMany(PetSupplies::class,
'cat_pet_supplies');
} public function
breeds()
{     return $this->belongsTo(Breeds::class, 'breed_id');
}
}
```

2) Dog:

```
<?php  namespace
App\Models;
 use
Illuminate\Database\Eloquent\Factories\HasFactory; use
Illuminate\Database\Eloquent\Model;
 class Dog extends
Model
{     use
HasFactory;
 public function
petSupplies()
{
    return $this->belongsToMany(PetSupplies::class, 'dog_pet_supplies');
} public function
breeds()
{
    return $this->belongsTo(Breeds::class, 'breed_id');
}
}
```

3)Breeds:

```
<?php  namespace
App\Models;
 use
Illuminate\Database\Eloquent\Factories\HasFactory; use
Illuminate\Database\Eloquent\Model;
 class Breeds extends
Model
{     use HasFactory;
protected $table = 'breeds';
         public function
cats()
    {         return $this->hasMany(Cat::class, 'breed_id');
    }     public function
dogs()
    {         return $this->hasMany(Dog::class, 'breed_id');
    }
}
```

4) PetSupplies:

```
<?php  namespace
App\Models;
 use
Illuminate\Database\Eloquent\Factories\HasFactory; use
Illuminate\Database\Eloquent\Model;

class PetSupplies extends Model
{     use
HasFactory;
}
```

5) Comment:

```
<?php
namespace App\Models;
 use
Illuminate\Database\Eloquent\Factories\HasFactory; use
Illuminate\Database\Eloquent\Model;
 class Comment extends
Model
{     use HasFactory;     protected $fillable = ['content',
'user_id', 'post_id'];
     public function
post()
    {         return $this->belongsTo(Post::class);
    }      public
function user()
    {         return $this->belongsTo(User::class);
    }
}
```

6) Post:

```
<?php  namespace
App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory; use
Illuminate\Database\Eloquent\Model;
 class Post extends
Model
{
    protected $fillable = ['title', 'content', 'user_id', 'cat_id',
'dog_id'];      public
function comments()
    {         return $this->hasMany(Comment::class);
    }

    public function user()
    {         return $this->belongsTo(User::class);
    }      public
function cat()
    {
        return $this->belongsTo(Cat::class);
    }      public
function dog()
    {
        return $this->belongsTo(Dog::class);
    }

}
```

7) User:

```
<?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,
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

{         retur
n [
            'email_verified_at' => 'datetime',
            'password' => 'hashed',
        ];     }
public function posts()
{     return
$this->hasMany(Post::class);
}
public function comments()
{     return
$this->hasMany(Comment::class);
}
}
```

3. seeders:

1) BreedsTableSeeder

```
<?php  namespace
Database\Seeders; use
Illuminate\Database\Seeder;
use App\Models\Breeds; use
```

```
Illuminate\Support\Facades\
DB;
 class BreedsTableSeeder extends
Seeder
{
    /**
     * Run the database seeds.
     */     public function
run(): void
    {
         DB::statement('SET FOREIGN_KEY_CHECKS=0;');

         DB::table('dog_pet_supplies')->truncate();

         DB::table('dogs')->truncate();


         DB::table('breeds')->truncate();

         DB::statement('SET FOREIGN_KEY_CHECKS=1;');


        Breeds::insert([
            [
                'name' => 'Labrador Retriever',
                'type' => 'Dog',
                'country' => 'Canada',
                'average_lifespan' => 10.0,
                'color' => 'Black, Yellow, Chocolate',
                'characteristics' => 'Friendly, Outgoing',
                'created_at' => now(),
                'updated_at' => now(),
            ],
            [
                'name' => 'German Shepherd',
                'type' => 'Dog',
                'country' => 'Germany',
                'average_lifespan' => 11.0,
                'color' => 'Black and Tan, Sable',
                'characteristics' => 'Intelligent, Loyal, Versatile',
                'created_at' => now(),
                'updated_at' => now(),
           ],
            [
                'name' => 'Golden Retriever',
                'type' => 'Dog',
                'country' => 'Scotland',
                'average_lifespan' => 11.0,
                'color' => 'Golden, Cream',
                'characteristics' => 'Intelligent, Friendly, Devoted',
                'created_at' => now(),
                'updated_at' => now(),
            ],
            [
                'name' => 'Beagle',
                'type' => 'Dog',
                'country' => 'United Kingdom',
                'average_lifespan' => 12.0,
                'color' => 'Tri-color, Lemon, Red and White',
                'characteristics' => 'Curious, Merry, Friendly',
                'created_at' => now(),
                'updated_at' => now(),
            ],
            [
                'name' => 'Bulldog',
                'type' => 'Dog',
                'country' => 'England',
                'average_lifespan' => 9.0,
                'color' => 'Fawn, White, Brindle',
                'characteristics' => 'Docile, Willful, Friendly',
                'created_at' => now(),
                'updated_at' => now(),
            ],
            [
                'name' => 'Persian',
                'type' => 'Cat',
                'country' => 'Iran',
                'average_lifespan' => 14.0,
                'color' => 'White, Black, Blue, Cream',
                'characteristics' => 'Calm, Affectionate, Loyal',
                'created_at' => now(),
                'updated_at' => now(),
            ],
            [
                'name' => 'Scottish Fold',
                'type' => 'Cat',
                'country' => 'Scotland',
                'average_lifespan' => 13.5,
               'color' => 'Various colors and patterns, including white,
black, blue, tabby',
                'characteristics' => 'Friendly, Playful, Adaptable',
                'created_at' => now(),
                'updated_at' => now(),
            ],
            [
                'name' => 'Abyssinian',
                'type' => 'Cat',
                'country' => 'Egypt',
                'average_lifespan' => 13.5,
                'color' => 'Reddish, Blue, Honey, Sorrel',
                'characteristics' => 'Active, Curious, Playful',
                'created_at' => now(),
                'updated_at' => now(),
            ],
            [
                'name' => 'Bengal',
                'type' => 'Cat',
                'country' => 'United States',
                'average_lifespan' => 14.0,
                'color' => 'Golden or silver background with black or
brown spots',
                'characteristics' => 'Energetic, Intelligent, Playful',
                'created_at' => now(),
                'updated_at' => now(),
            ],
            [
                'name' => 'Burmese',
                'type' => 'Cat',
                'country' => 'Myanmar',
                'average_lifespan' => 14.0,
                'color' => 'Sable, Champagne, Blue, Apricot',
                'characteristics' => 'Social, Active, Intelligent',
                'created_at' => now(),
                'updated_at' => now(),
            ],
        ]);
        // Breeds::factory()->count(10)->create();
    }
}
```

2) CatTableSeeder

```
<?php  namespace
Database\Seeders;
```

```
 use Illuminate\Database\Seeder;
use App\Models\Cat;  use
Illuminate\Support\Facades\DB;
 class CatTableSeeder extends
Seeder
{
    /**
     * Run the database seeds.
     */     public function
run(): void
    {
        DB::statement('SET FOREIGN_KEY_CHECKS=0;');
DB::table('cats')->truncate();
        DB::statement('SET FOREIGN_KEY_CHECKS=1;');

        Cat::create([
            'name' => 'Persian',
            'country' => 'Iran',
            'theLengthOfLife' => 14.0,
            'color' => 'white, black, blue, cream',
            'breed_id' => 6,
            'created_at' => now(),
            'updated_at' => now(),
        ]);

        Cat::create([
            'name' => 'Burmese',
            'country' => 'Myanmar',
            'theLengthOfLife' => 14.0,
            'color' => 'sable, champagne, blue, apricot',
            'breed_id' =>10,
            'created_at' => now(),
            'updated_at' => now(),
        ]);

        Cat::create([
            'name' => 'Scottish Fold',
            'country' => 'Scotland',
           'theLengthOfLife' => 13.5,
            'color' => 'various colors and patterns, including white,
black, blue, tabby',
            'breed_id' =>7,
            'created_at' => now(),
            'updated_at' => now(),
        ]);
        Cat::create([
            'name' => 'Abyssinian',
            'country' => 'Egypt',
            'theLengthOfLife' => 13.5,
            'color' => 'reddish, blue, honey, sorrel',
            'breed_id' =>8,
            'created_at' => now(),
            'updated_at' => now(),
        ]);

        Cat::create([
            'name' => 'Bengal',
            'country' => 'United States',
            'theLengthOfLife' => 14.0,
            'color' => 'golden or silver background with black or brown
spots',
            'breed_id' =>9,
            'created_at' => now(),
            'updated_at' => now(),
        ]);
        Cat::factory()->count(10)->create();
    }
}
```

3) CommentSeeder

```
<?php
namespace Database\Seeders;
 use
Illuminate\Database\Console\Seeds\WithoutModelEvents; use
Illuminate\Database\Seeder;
 class CommentSeeder extends
Seeder
{
    /**
     * Run the database seeds.
     */     public function
run(): void
    {
        \App\Models\Comment::factory()->count(50)->create();
    }
}
```

4) DatabaseSeeder

```
<?php  namespace
Database\Seeders;
 use
App\Models\User;
// use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;
use Illuminate\Support\Facades\DB;
use App\Models\Cat;
use App\Models\Dog;
use App\Models\PetSupplies;
use App\Models\User;
use App\Models\Post;
use App\Models\Comment;

class DatabaseSeeder extends Seeder
{
    /**
     * Seed the application's database.
     */
    public function run(): void
    {
        DB::table('users')->truncate();
        DB::table('cat_pet_supplies')->truncate();
        DB::table('dog_pet_supplies')->truncate();

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

       $this->call(CatTableSeeder::class);
        $this->call(DogTableSeeder::class);
        $this->call(BreedsTableSeeder::class);
        $this->call(PetSuppliesTableSeeder::class);
        $this->call(PostSeeder::class);
        $this->call(CommentSeeder::class);
        $this->call(UsersTableSeeder::class);
        User::factory()->create([
            'name' => 'Test User',
            'email' => 'test@example.com',
        ]);

        $cats = Cat::all();
        $petSupplies = PetSupplies::all();
         foreach ($cats as $cat)
{

            $cat->petSupplies()->attach($petSupplies->random(rand(1,
3))->pluck('id')->toArray());
        }


        $dogs = Dog::all();
         foreach ($dogs as $dog)
{

            $dog->petSupplies()->attach($petSupplies->random(rand(1,
3))->pluck('id')->toArray());
        }
    }
}
```

5) DogTableSeeder

```
<?php  namespace
Database\Seeders;
 use Illuminate\Database\Seeder;
use App\Models\Dog;  use
Illuminate\Support\Facades\DB;
 class DogTableSeeder extends
Seeder
{
    /**
     * Run the database seeds.
     */      public function
run(): void
```

```
    {

        DB::statement('SET FOREIGN_KEY_CHECKS=0;');
DB::table('dogs')->truncate();
        DB::statement('SET FOREIGN_KEY_CHECKS=1;');


        Dog::create([
            'name' => 'Labrador Retriever',
            'country' => 'Canada',
            'theLengthOfLife' => 10.0,
            'color' => 'Black, Yellow, Chocolate',
            'breed_id' =>1,
            'created_at' => now(),
            'updated_at' => now(),
        ]);

        Dog::create([
            'name' => 'German Shepherd',
            'country' => 'Germany',
            'theLengthOfLife' => 11.0,
            'color' => 'Black and Tan, Sable',
            'breed_id' =>2,
            'created_at' => now(),
            'updated_at' => now(),
        ]);

        Dog::create([
            'name' => 'Golden Retriever',
            'country' => 'Scotland',
            'theLengthOfLife' => 11.0,
            'color' => 'Golden, Cream',
            'breed_id' =>3,
           'created_at' => now(),
         'updated_at' => now(),
       ]);

        Dog::create([
            'name' => 'Bulldog',
            'country' => 'England',
            'theLengthOfLife' => 9.0,
            'color' => 'Fawn, White, Brindle',
            'breed_id' =>5,
            'created_at' => now(),
            'updated_at' => now(),
        ]);

        Dog::create([
            'name' => 'Beagle',
            'country' => 'United Kingdom',
            'theLengthOfLife' => 12.0,
            'color' => 'Tri-color, Lemon, Red and White',
            'breed_id' =>4,
            'created_at' => now(),
            'updated_at' => now(),
        ]);
        Dog::factory()->count(10)->create();
    }
}
```

6) PetSuppliesTableSeeder

```
<?php
namespace Database\Seeders;
 use Illuminate\Database\Seeder;
use App\Models\PetSupplies; use
Illuminate\Support\Facades\DB;

class PetSuppliesTableSeeder extends Seeder
{
    /**
* Run the database seeds.
  *
* @return void
  */
  public function run()
    {

        DB::statement('SET FOREIGN_KEY_CHECKS=0;');
        DB::table('dog_pet_supplies')->truncate();
        DB::table('pet_supplies')->truncate();
        DB::statement('SET FOREIGN_KEY_CHECKS=1;');



        PetSupplies::create([
            'name' => 'Premium Cat Food',
            'type' => 'Food',
            'price' => 20.2,
            'description' => 'Nutritious cat food with high protein
content.',         ]);

        PetSupplies::create([
            'name' => 'Catnip Mouse Toy',
            'type' => 'Toy',
            'price' => 9.99,
            'description' => 'Soft plush mouse filled with catnip for
playful cats.',
        ]);

        PetSupplies::create([
            'name' => 'Flea and Tick Prevention',
            'type' => 'Care',
            'price' => 24.99,
            'description' => 'Effective treatment for fleas and ticks.',
        ]);

        PetSupplies::create([
            'name' => 'Training Pads',
            'type' => 'Training',
            'price' => 19.06,
            'description' => 'Absorbent training pads for housebreaking
puppies.',
        ]);
        PetSupplies::create([
            'name' => 'Cat Brush',
            'type' => 'Grooming',
 'price' => 14.43,
           'description' => 'Effective brush for removing loose fur and
reducing shedding.',
        ]);
        PetSupplies::factory()->count(10)->create();
    }
}
```

7) PostSeeder

```
<?php  namespace
Database\Seeders;
```

```
 use
Illuminate\Database\Console\Seeds\WithoutModelEvents; use
Illuminate\Database\Seeder;
 class PostSeeder extends
Seeder
{
    /**
     * Run the database seeds.
     */     public function
run(): void
    {
        \App\Models\Post::factory()->count(10)->create();
    }
}
```

8) UserTableSeeder

```
<?php  namespace
Database\Seeders;
 use
Illuminate\Database\Console\Seeds\WithoutModelEvents; use
Illuminate\Database\Seeder;
use App\Models\User;
 class UsersTableSeeder extends
Seeder
{
    /**
     * Run the database seeds.
     */     public function
run(): void
    {
        User::factory()->count(10)->create();
    }
}
```

4. factory:

1) BreedsFactory

```
<?php  namespace
Database\Factories;
 use
Illuminate\Database\Eloquent\Factories\Factory; use
App\Models\Breeds;
/**
* @extends
\Illuminate\Database\Eloquent\Factories\Factory<\App\Models\Breeds>
 */
class BreedsFactory extends Factory
{
    /**
* Define the model's default state.
     *
* @return array<string, mixed>
     */
    public function definition(): array

{         return
[
            'name' => $this->faker->unique()->word(),

            'type' => $this->faker->randomElement(['Dog', 'Cat']),
                     'country' => $this->faker->country(),
       'average_lifespan' => $this->faker->randomFloat(1, 5, 20),

            'color' => $this->faker->word() . ', ' .
$this->faker->word() . ', ' . $this->faker->word(),
            'characteristics' => $this->faker->sentence(),
            'created_at' => now(),
            'updated_at' => now(),
        ];
    }
}
```

2) CatFactory

```
<?php  namespace
Database\Factories;
 use App\Models\Cat; use
Illuminate\Database\Eloquent\Factories\Factory; use
App\Models\Breeds; class CatFactory extends Factory
```

```
{     protected $model =
Cat::class;

    public function definition()

{         retur
n [
            'name' => $this->faker->word,
            'country' => $this->faker->country,
            'theLengthOfLife' => $this->faker->numberBetween(10, 15),
            'color' => $this->faker->word,
            'breed_id' => Breeds::inRandomOrder()->first()->id,
        ];
    }
}
```

3) CommentFactory

```
<?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

{         retur
n [
            'content' => $this->faker->sentence,
'user_id' => User::factory(),
            'post_id' => Post::factory(),
        ];
    }
}
```

4) DogFactory

```
<?php  namespace
Database\Factories;
 use App\Models\Dog;
use Illuminate\Database\Eloquent\Factories\Factory;
use App\Models\Breeds;
 class DogFactory extends
Factory {     protected $model =
Dog::class;
     public function
definition()

{         retur
n [
            'name' => $this->faker->name,
            'country' => $this->faker->country,
            'theLengthOfLife' => $this->faker->numberBetween(8, 15),
            'color' => $this->faker->colorName,
            'breed_id' => Breeds::inRandomOrder()->first()->id,
            'created_at' => now(),
            'updated_at' => now(),
        ];
    }
}
```

5) PetSuppliesFactory

```
<?php  namespace
Database\Factories;
 Use
Illuminate\Database\Eloquent\Factories\Factory;
use App\Models\PetSupplies;
/**
* @extends
\Illuminate\Database\Eloquent\Factories\Factory<\App\Models\PetSupplies
```

```
>  */ class PetSuppliesFactory extends
Factory
{
    /**
* Define the model's default state.
     *
* @return array<string, mixed>
     */
    public function definition(): array

{         return
[         'name
' =>
$this->faker->u
nique()->word()
. ' ' .
$this->faker->word(),
            'type' => $this->faker->randomElement(['Food', 'Toy',
'Care', 'Training', 'Grooming']),
            'price' => $this->faker->randomFloat(2, 5, 50),

            'description' => $this->faker->sentence(),
             'created_at' => now(),
            'updated_at' => now(),
        ];
    }
}
```

6) PostFactory

```
<?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

{          retur
n [
            'title' => $this->faker->sentence,
            'content' => $this->faker->paragraph(5),
            'user_id' => User::factory(),
        ];
    }
}
```

7) UserFactory

```
<?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
{         retur
n [
            'name' => fake()->name(),
            'email' => fake()->unique()->safeEmail(),
            'email_verified_at' => now(),
            'password' => static::$password ??= Hash::make('password'),
            'remember_token' => Str::random(10),
        ];
    }

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