CSC348 – Code Submission

All code submitted is to demonstrate the use of exclusively Models, Migrations, Seeders and Factories.

1. **Model Classes**
   1. User class
   2. Post class
   3. Comment class
2. **Migrations**
   1. Users Table Migration
   2. Posts Table Migration
   3. Comments Table Migration
3. **Seeds**
   1. Database Seeder
   2. User Seeder
   3. Post Seeder
   4. Comment Seeder
4. **Factories**
   1. User Factory
   2. Post Factory
   3. Comment Factory

**1 Model Classes**

**1.1 User Class**

<?php

namespace App;

use Illuminate\Contracts\Auth\MustVerifyEmail;

use Illuminate\Foundation\Auth\User as Authenticatable;

use Illuminate\Notifications\Notifiable;

class User extends Authenticatable

{

    use Notifiable;

    /\*\*

     \* The attributes that are mass assignable.

     \* @var array

     \*/

    protected $fillable = [

        'name', 'email', 'password',

    ];

    /\*\*

     \* The attributes that should be hidden for arrays.

     \* @var array

     \*/

    protected $hidden = [

        'password', 'remember\_token',

    ];

    /\*\*

     \* The attributes that should be cast to native types.

     \* @var array

     \*/

    protected $casts = [

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

    ];

    /\*\*

     \* Return the many Comments that belong to a user

     \*/

    public function comments()

    {

        return $this->hasMany('App\Comment');

    }

    /\*\*

     \* Return the many Posts that belong to a user

     \*/

    public function posts()

    {

        return $this->hasMany('App\Post');

    }

}

**1.2 Post Class**

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Post extends Model

{

    /\*\*

     \* Returns the creator of the post

     \*/

    public function user()

    {

        return $this->belongsTo('App\User');

    }

    /\*\*

     \* Returns the many Comments on the post

     \*/

    public function comments()

    {

        return $this->hasMany('App\Comment');

    }

}

**1.3 Comment Class**

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Comment extends Model

{

    /\*\*

     \* Returns the creator of the comment

     \*/

    public function user()

    {

        return $this->belongsTo('App\User');

    }

    /\*\*

     \* Returns the post the comment belongs to

     \*/

    public function post()

    {

        return $this->belongsTo('App\Post');

    }

}

**2 Migrations**

**2.1 Users Table Migration**

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreateUsersTable extends Migration

{

    /\*\*

     \* Run the migrations.

     \*

     \* @return void

     \*/

    public function up()

    {

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

            $table->bigIncrements('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');

    }

}

**2.2 Posts Table Migration**

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreatePostsTable extends Migration

{

    /\*\*

     \* Run the migrations.

     \*

     \* @return void

     \*/

    public function up()

    {

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

        {

            $table->bigIncrements('id');

            $table->string('content');

            $table->timestamps();

            $table->bigInteger('user\_id')->unsigned();

        });

        /\*\*

         \* Configuring Relationships

         \*/

        Schema::table('posts', function($table)

        {

            $table->foreign('user\_id')->references('id')

                ->on('users')->onDelete('cascade')->onDelete('cascade');

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*

     \* @return void

     \*/

    public function down()

    {

        Schema::dropIfExists('posts');

    }

}

**2.3 Comments Table Migration**

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreateCommentsTable extends Migration

{

    /\*\*

     \* Run the migrations.

     \* @return void

     \*/

    public function up()

    {

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

        {

            $table->bigIncrements('id');

            $table->string('content');

            $table->timestamps();

            $table->bigInteger('user\_id')->unsigned();

            $table->bigInteger('post\_id')->unsigned();

        });

        /\*\*

         \* Configuring Relationships

         \*/

        Schema::table('comments', function($table)

        {

            $table->foreign('post\_id')->references('id')

                ->on('posts')->onDelete('cascade')->onDelete('cascade');

            $table->foreign('user\_id')->references('id')

                ->on('users')->onDelete('cascade')->onDelete('cascade');

        });

    }

    /\*\*

     \* Reverse the migrations.

     \* @return void

     \*/

    public function down()

    {

        Schema::dropIfExists('comments');

    }

}

**3 Seeds**

**3.1 Database Seeder**

<?php

use Illuminate\Database\Seeder;

class DatabaseSeeder extends Seeder

{

    /\*\*

     \* Seed the application's database.

     \*

     \* @return void

     \*/

    public function run()

    {

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

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

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

    }

}

**3.2 User Seeder**

<?php

use Illuminate\Database\Seeder;

class UsersTableSeeder extends Seeder

{

    /\*\*

     \* Run the database seeds.

     \*

     \* @return void

     \*/

    public function run()

    {

        factory(App\User::class, 5)->create();

    }

}

**3.3 Post Seeder**

<?php

use Illuminate\Database\Seeder;

class PostsTableSeeder extends Seeder

{

    /\*\*

     \* Run the database seeds.

     \*

     \* @return void

     \*/

    public function run()

    {

        factory(App\Post::class, 25)->create();

    }

}

**3.3 Comments Seeder**

<?php

use Illuminate\Database\Seeder;

class CommentsTableSeeder extends Seeder

{

    /\*\*

     \* Run the database seeds.

     \*

     \* @return void

     \*/

    public function run()

    {

        factory(App\Comment::class, 150)->create();

    }

}

**4 Factories**

**4.1 User Factory**

<?php

/\*\* @var \Illuminate\Database\Eloquent\Factory $factory \*/

use App\User;

use Faker\Generator as Faker;

use Illuminate\Support\Str;

$factory->define(User::class, function (Faker $faker) {

    return [

        'name' => $faker->name,

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

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

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

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

    ];

});

**4.2 Post Factory**

<?php

/\*\* @var \Illuminate\Database\Eloquent\Factory $factory \*/

use App\Post;

use Faker\Generator as Faker;

$factory->define(Post::class, function (Faker $faker) {

    return [

        'content' => $faker->realText($maxNbChars = 100,),

        'user\_id' => $faker->randomElement(App\User::get()),

    ];

});

**4.3 Comment Factory**

<?php

/\*\* @var \Illuminate\Database\Eloquent\Factory $factory \*/

use App\Comment;

use Faker\Generator as Faker;

$factory->define(Comment::class, function (Faker $faker) {

    return [

        'content' => $faker->realText($maxNbChars = 25),

        'post\_id' => $faker->randomElement(App\Post::get()),

        'user\_id' => $faker->randomElement(App\User::get()),

    ];

});