Tutorial 02 to do in class – Remember to upload the repo link to Teams

A. Configuring our database

- Execute XAMPP, *start* the Apache Module, *start* the MySQL module, and click the *MySQL Admin* button (of the MySQL module). It takes us to the phpMyAdmin application (see Fig. 10-2).

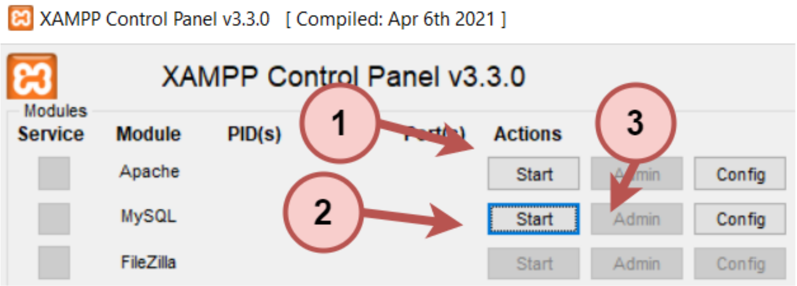
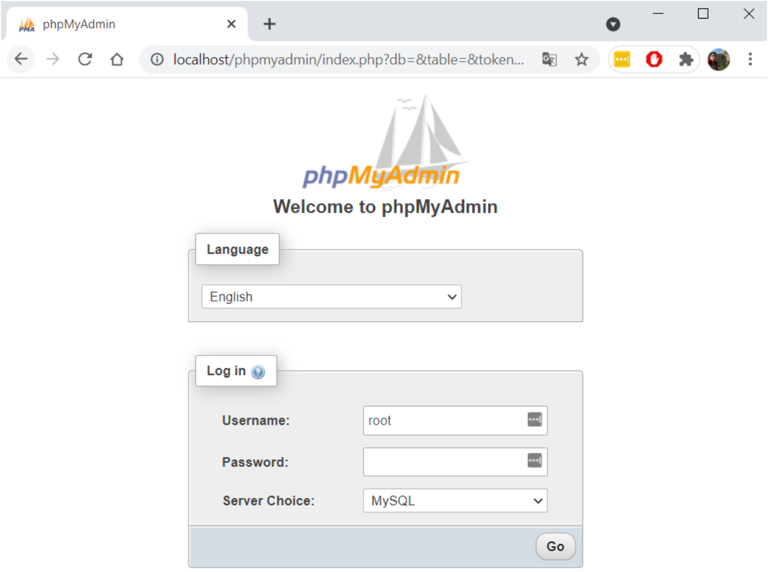


Figure 10-2. Starting MySQL module in XAMPP.

- **Note:** If you are using WAMP or another similar application, the phpMyAdmin application can be commonly accessed through the next route: http://localhost/phpmyadmin/.

- In the phpMyAdmin application, enter your username and password. Default values are "root" (for the username) and an empty password (see Fig. 10-3).



localhost/phpmyadmin/index.php?db=&table=&token...

Language: English

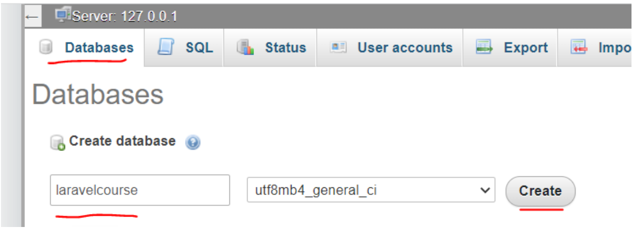
Username: root

Password:

Server Choice: MySQL

Figure 10-3. XAMPP phpMyAdmin application.

- Once logged in to phpMyAdmin, click the *Databases* tab. Enter the database name *laravelcourse*, and click *Create* button (see Fig. 10-4).



laravelcourse

utf8mb4_general_ci

Figure 10-4. Database creation.

B. A basic product model

Product migration

- Let’s create our product migration. In the Terminal, go to the project directory, and execute the following:

Execute in Terminal

```
php artisan make:migration create_products_table
```

- The previous command creates a products table migration file inside the *database/migrations* folder. Each migration file name contains a timestamp that allows Laravel to determine the order of the migrations. In our case, it created a file named *2024_08_14_153916_create_products_table.php*.

- Now, open the previously generated file. Delete all the existing code and fill it with the following code.

Replace Entire Code

```
<?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('products', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->integer('price');
            $table->timestamps();
        });
    }

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

Modifying the .env file

- To execute the migrations, we need to modify the *.env* file (located at the project root folder). In the *.env* file, make the following changes in **bold**. You need to set the *DB_DATABASE*, *DB_USERNAME*, and *DB_PASSWORD* (**and remove the corresponding "#" at the beginning of those variables**). If you have a different database name, username, or password, make the corresponding changes.

Modify Bold Code

```
...
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=laravelcourse
DB_USERNAME=root
DB_PASSWORD=
...
```

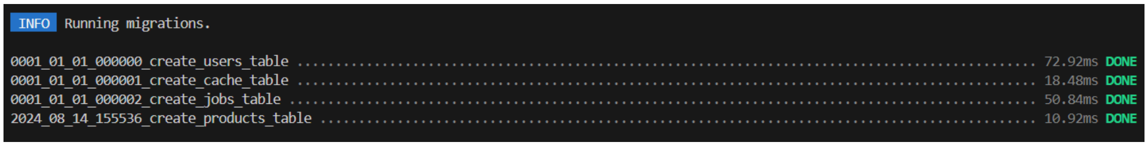
Executing the migrations

- To run the migrations, in the Terminal, go to the project directory, and execute the following:

Execute in Terminal

```
php artisan migrate
```

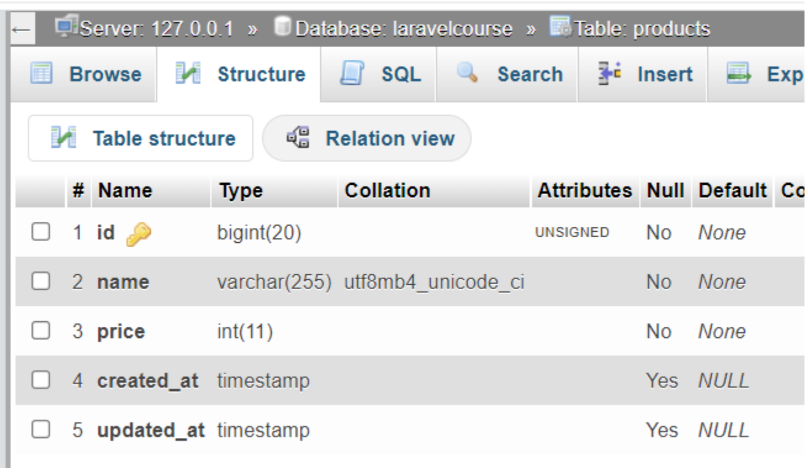
- You should see a result as presented in Fig. 11-1.



```
 INFO  Running migrations.

0001_01_01_000000_create_users_table .................................................................... 72.92ms DONE
0001_01_01_000001_create_cache_table .................................................................... 18.48ms DONE
0001_01_01_000002_create_jobs_table ..................................................................... 50.84ms DONE
2024_08_14_155536_create_products_table ................................................................. 10.92ms DONE
```

Figure 11-1. Execution of Laravel migrations.

- You should see the new products table in phpMyAdmin in Fig. 11-2.


Server: 127.0.0.1 » Database: laravelcourse » Table: products
Browse
Structure
SQL
Search
Insert
Exp
Table structure
Relation view

	#	Name	Type	Collation	Attributes	Null	Default	Co
□	1	id	bigint(20)		UNSIGNED	No	None	
□	2	name	varchar(255)	utf8mb4_unicode_ci		No	None	
□	3	price	int(11)			No	None	
□	4	created_at	timestamp			Yes	NULL	
□	5	updated_at	timestamp			Yes	NULL	

Figure 11-2. Products table.

Model

- In the Terminal, go to the project directory, and execute the following:

Execute in Terminal

```
php artisan make:model Product
```

- You will see the *Product.php* file inside the *app/Models* folder. In *app/Models/Product.php*, make the following changes in **bold**.

Modify Bold Code

```
<?php

namespace App\Models;

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

class Product extends Model
{
    use HasFactory;

    /**
     * PRODUCT ATTRIBUTES
     * $this->attributes['id'] - int - contains the product primary key (id)
     * $this->attributes['name'] - string - contains the product name
     * $this->attributes['price'] - int - contains the product price
    */

    protected $fillable = ['name','price'];

    public function getId(): int
    {
        return $this->attributes['id'];
    }

    public function setId($id) : void
    {
        $this->attributes['id'] = $id;
    }

    public function getName(): string
    {
        return $this->attributes['name'];
    }

    public function setName($name) : void
    {
        $this->attributes['name'] = $name;
    }

    public function getPrice(): int
    {
        return $this->attributes['price'];
    }

    public function setPrice($price) : void
    {
        $this->attributes['price'] = $price;
    }
}
```

Factories

- Go to *database/factories/* and create a file *ProductFactory.php*, with the following content:

Add Entire Code

```
<?php

namespace Database\Factories;

use Illuminate\Database\Eloquent\Factories\Factory;

class ProductFactory extends Factory
{
    public function definition(): array
    {
        return [
            'name' => $this->faker->company,
            'price' => $this->faker->numberBetween($min = 200, $max = 9000),
        ];
    }
}
```

DatabaseSeeder

- Go to *database/seeders/DatabaseSeeder.php*, make the following changes in **bold**.

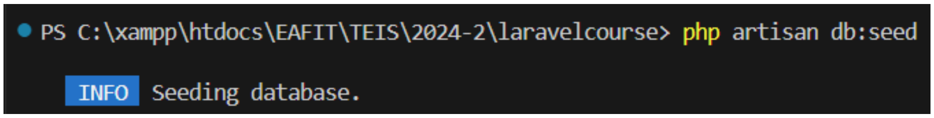
Modify Bold Code

```
<?php

namespace Database\Seeders;

use App\Models\Product;
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
    {
        // \App\Models\User::factory(10)->create();
        Product::factory(8)->create();
        ~~User::factory()->create([~~
            ~~'name' => 'Test User',~~
            ~~'email' => 'test@example.com',~~
        ~~]);~~
    }
}
```

- In the Terminal, go to the project directory, and execute the following:

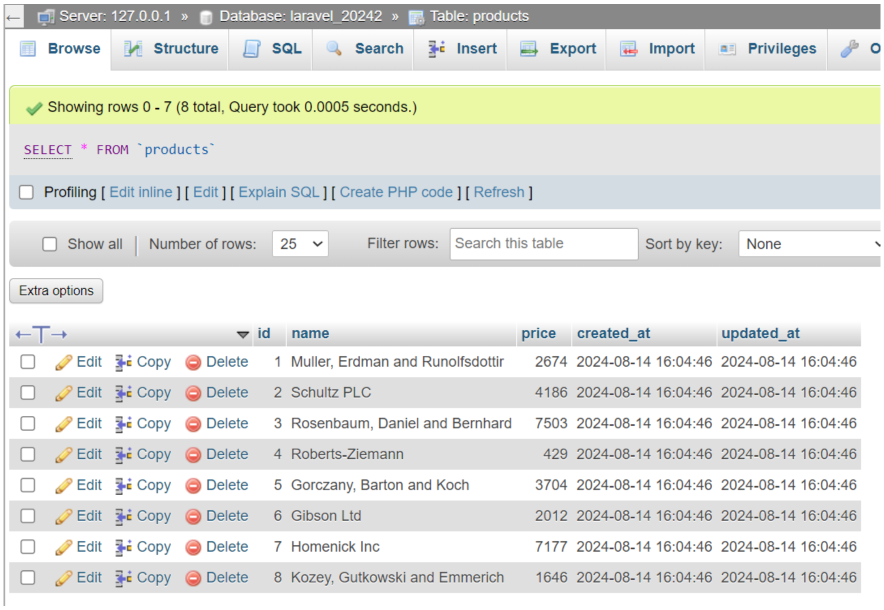
Execute in Terminal

```
php artisan db:seed
```



```
PS C:\xampp\htdocs\EAFIT\TEIS\2024-2\laravelcourse> php artisan db:seed

   INFO  Seeding database.
```

- You should see eight new products added into the database.



Server: 127.0.0.1 » Database: laravel_20242 » Table: products
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
O

✓ Showing rows 0 - 7 (8 total, Query took 0.0005 seconds.)

```
SELECT * FROM `products`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options

←T→				id	name	price	created_at	updated_at
□	Edit	Copy	Delete	1	Muller, Erdman and Runolfsdottir	2674	2024-08-14 16:04:46	2024-08-14 16:04:46
□	Edit	Copy	Delete	2	Schultz PLC	4186	2024-08-14 16:04:46	2024-08-14 16:04:46
□	Edit	Copy	Delete	3	Rosenbaum, Daniel and Bernhard	7503	2024-08-14 16:04:46	2024-08-14 16:04:46
□	Edit	Copy	Delete	4	Roberts-Ziemann	429	2024-08-14 16:04:46	2024-08-14 16:04:46
□	Edit	Copy	Delete	5	Gorczany, Barton and Koch	3704	2024-08-14 16:04:46	2024-08-14 16:04:46
□	Edit	Copy	Delete	6	Gibson Ltd	2012	2024-08-14 16:04:46	2024-08-14 16:04:46
□	Edit	Copy	Delete	7	Homenick Inc	7177	2024-08-14 16:04:46	2024-08-14 16:04:46
□	Edit	Copy	Delete	8	Kozey, Gutkowski and Emmerich	1646	2024-08-14 16:04:46	2024-08-14 16:04:46

C. A basic listing products from database

Controller

- In *app/Http/Controllers/ProductController.php*, make the following changes in **bold**.

Modify Bold Code

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\View\View;
use App\Models\Product;

class ProductController extends Controller
{
    ~~public static $products = [~~
        ~~["id"=>"1", "name"=>"TV", "description"=>"Best TV"],~~
        ~~["id"=>"2", "name"=>"iPhone", "description"=>"Best iPhone"],~~
        ~~["id"=>"3", "name"=>"Chromecast", "description"=>"Best Chromecast"],~~
        ~~["id"=>"4", "name"=>"Glasses", "description"=>"Best Glasses"]~~
    ~~];~~

    public function index(): View
    {
        $viewData = [];
        $viewData["title"] = "Products - Online Store";
        $viewData["subtitle"] =  "List of products";
        $viewData["products"] = Product::all();
        return view('product.index')->with("viewData", $viewData);
    }

    public function show(string $id): View
    {
        $viewData = [];
        $product = Product::findOrFail($id);
        $viewData["title"] = $product["name"]." - Online Store";
        $viewData["subtitle"] =  $product["name"]." - Product information";
        $viewData["product"] = $product;
        return view('product.show')->with("viewData", $viewData);
    }

    ...
```

View

- In *resources/views/product/show.blade.php*, make the following changes in **bold**.

Modify Bold Code

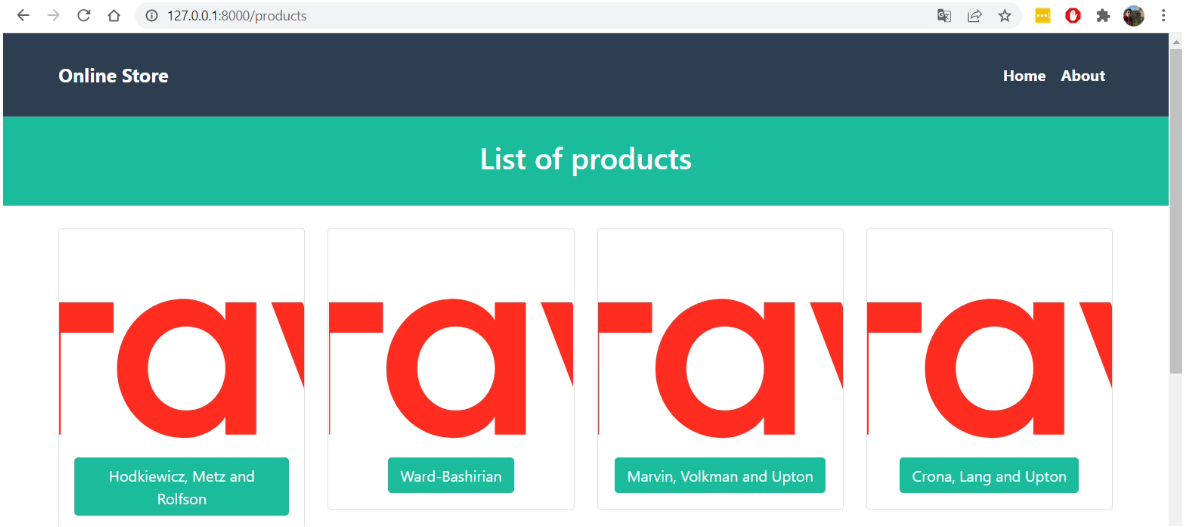
```
@extends('layouts.app')
@section('title', $viewData["title"])
@section('subtitle', $viewData["subtitle"])
@section('content')
<div class="card mb-3">
  <div class="row g-0">
    <div class="col-md-4">
      <img src="https://laravel.com/img/logotype.min.svg" class="img-fluid rounded-start">
    </div>
    <div class="col-md-8">
      <div class="card-body">
        <h5 class="card-title">
          {{ $viewData["product"]["name"] }}
        </h5>
~~        <p class="card-text">{{ $viewData["product"]["description"] }}</p>~~
        **<p class="card-text">{{ $viewData["product"]["price"] }}</p>**
      </div>
    </div>
  </div>
</div>
@endsection
```

Execution

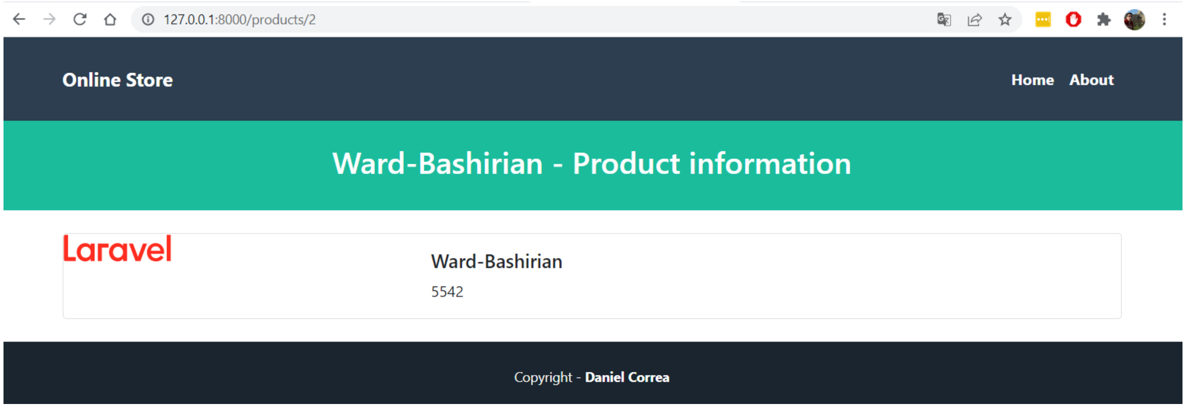
Execute in Terminal

```
php artisan serve
```

Go to the ("/products") route and you should see the application running with information retrieved from the database.



List of products



Ward-Bashirian - Product information

Ward-Bashirian

5542

Copyright - **Daniel Correa**

D. A basic save product

Controller

- In *app/http/Controllers/ProductController.php*, make the following changes in **bold**.

Modify Bold Code

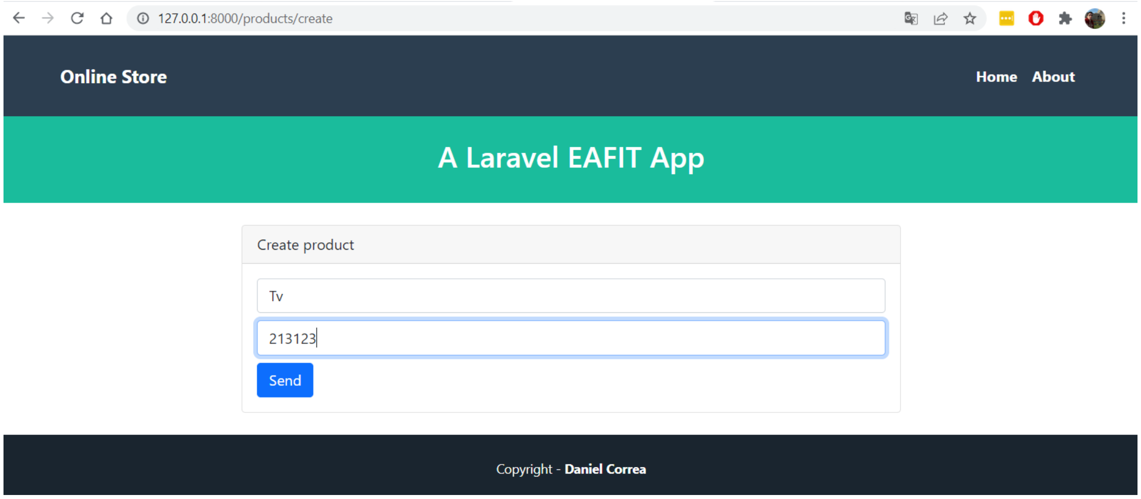
```
...

    public function save(Request $request): \Illuminate\Http\RedirectResponse
    {
        $request->validate([
            "name" => "required",
            "price" => "required"
        ]);
~~        dd($request->all());~~

~~        //here will go the code to call the model and save it to the database~~
        Product::create($request->only(["name","price"]));

        return back();
    }
}
```

Execution

Execute in Terminal

```
php artisan serve
```

Go to the ("/products/create") route and insert a new product. It should be added into the database.



Create product

Tv

213123

E. A basic comment model and comment relationship

Comment migration

- Let’s create our comment migration. In the Terminal, go to the project directory, and execute the following:

Execute in Terminal

```
php artisan make:migration create_comments_table
```

- The previous command creates a products table migration file inside the *database/migrations* folder. Each migration file name contains a timestamp that allows Laravel to determine the order of the migrations. In our case, it created a file named *2023_02_21_203622_create_comments_table.php*.

- Now, open the previously generated file. Delete all the existing code and fill it with the following code.

Replace Entire Code

```
<?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->text('description');
            $table->unsignedBigInteger('product_id');
            $table->foreign('product_id')->references('id')->on('products');
            $table->timestamps();
        });
    }

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

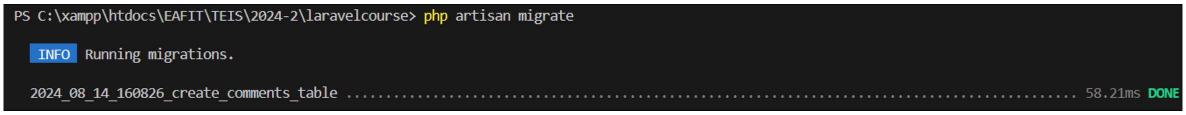
Executing the migrations

- To run the migrations, in the Terminal, go to the project directory, and execute the following:

Execute in Terminal

```
php artisan migrate
```

- You should see a result as presented in next figure.



```
PS C:\xampp\htdocs\EAFIT\TEIS\2024-2\laravelcourse> php artisan migrate

   INFO  Running migrations.

   2024_08_14_160826_create_comments_table .............................................................................. 58.21ms DONE
```

Model

- In the Terminal, go to the project directory, and execute the following:

Execute in Terminal

```
php artisan make:model Comment
```

- You will see the *Comment.php* file inside the *app/Models* folder. In *app/Models/Comment.php*, make the following changes in **bold**.

Modify Bold Code

```
<?php

namespace App\Models;

```

```
use App\Models\Product;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\BelongsTo;

class Comment extends Model
{
    use HasFactory;

    /**
     * PRODUCT ATTRIBUTES
     * $this->attributes['id'] - int - contains the product primary key (id)
     * $this->attributes['description'] - string - contains the comment description
     * $this->product - Product - contains the associated Product
    */

    protected $fillable = ['description', 'product_id'];

    public function getId(): int
    {
        return $this->attributes['id'];
    }

    public function setId(int $id): void
    {
        $this->attributes['id'] = $id;
    }

    public function getDescription(): string
    {
        return $this->attributes['description'];
    }

    public function setDescription(string $desc): void
    {
        $this->attributes['description'] = $desc;
    }
    public function getProductId(): int
    {
        return $this->attributes['product_id'];
    }

    public function setProductId(int $pId): void
    {
        $this->attributes['product_id'] = $pId;
    }

    public function product(): BelongsTo
    {
        return $this->belongsTo(Product::class);
    }

    public function getProduct(): Product
    {
        return $this->product;
    }

    public function setProduct($product): void
    {
        $this->product = $product;
    }
}
```

Product Model

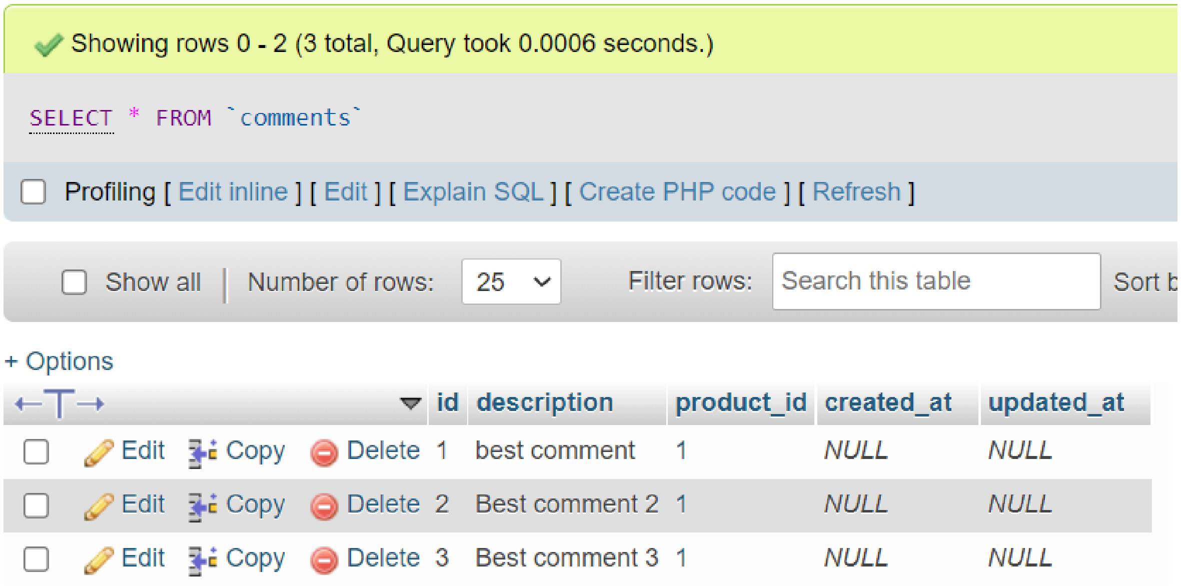
- In *app/Models/Product.php*, make the following changes in **bold**.

Modify Bold Code

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use App\Models\Comment;
use Illuminate\Database\Eloquent\Relations\HasMany;
use Illuminate\Database\Eloquent\Collection;

class Product extends Model
{
    use HasFactory;

    /**
     * PRODUCT ATTRIBUTES
     * $this->attributes['id'] - int - contains the product primary key (id)
     * $this->attributes['name'] - string - contains the product name
     * $this->attributes['price'] - int - contains the product price
     * $this->comments - Comment[] - contains the associated comments
    */

    ...

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

    public function getComments(): Collection
    {
        return $this->comments;
    }

    public function setComments(Collection $comments): void
    {
        $this->comments = $comments;
    }
}
```

Comments rows

- Open phpMyAdmin, go the *laravelcourse* database, go the *comments* table, click the *insert* tabs, and manually create 3 comments for the product number 1 (with id = 1).


Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all
Filter rows:
Search this table
Sort b
+ Options
Edit
Copy
Delete

✓ Showing rows 0 - 2 (3 total, Query took 0.0006 seconds.)

```
SELECT * FROM `comments`
```

Number of rows: 25

id	description	product_id	created_at	updated_at
1	best comment	1	*NULL*	*NULL*
2	Best comment 2	1	*NULL*	*NULL*
3	Best comment 3	1	*NULL*	*NULL*

Product show view

- Go to *resources/views/product/show.blade.php* and make the following changes in **bold**.

Modify Bold Code

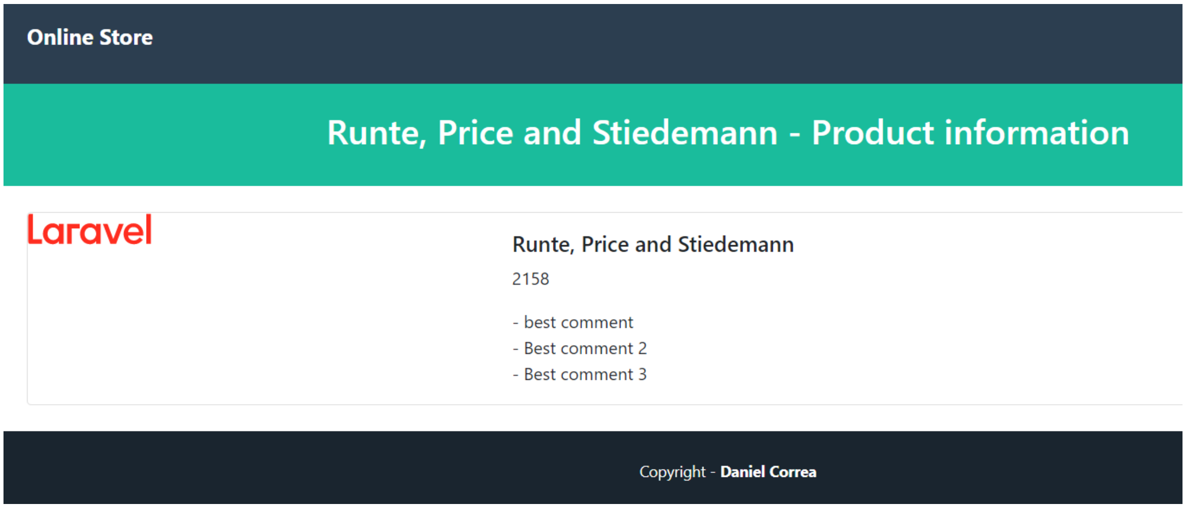
```
@extends('layouts.app')
@section('title', $viewData["title"])
@section('subtitle', $viewData["subtitle"])
@section('content')
<div class="card mb-3">
  <div class="row g-0">
    <div class="col-md-4">
      <img src="https://laravel.com/img/logotype.min.svg" class="img-fluid rounded-start">
    </div>
    <div class="col-md-8">
      <div class="card-body">
        <h5 class="card-title">
          {{ $viewData["product"]["name"] }}
        </h5>
    <p class="card-text">{{ $viewData["product"]["price"] }}</p>

    @foreach($viewData["product"]->comments as $comment)
     - {{ $comment->getDescription() }}<br />
    @endforeach

   </div>
  </div>
 </div>
</div>
@endsection
```

Execution

Execute in Terminal

```
php artisan serve
```

Go to the ("/products/1") route. You will see the product with id 1 with its comments.


Online Store
Runte, Price and Stiedemann - Product information
Laravel
Runte, Price and Stiedemann
2158
- best comment
- Best comment 2
- Best comment 3
Copyright - Daniel Correa

Congratulations, you have created an app which uses Laravel Models (including relationships), Fakers, Migrations, Seeds, and many more.

NOTE: remember to run Laravel Pint to clean the code.