

# Component-2: CRUD

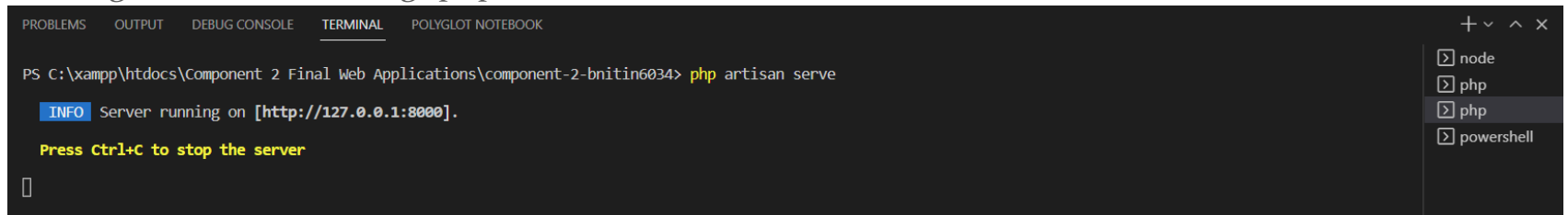
## ADVANCED WEB ENGINEERING

Nitin Utsav Bartaula | 77261073 | 31<sup>st</sup> December, 2022

Prerequisites to doing CRUD and Login:

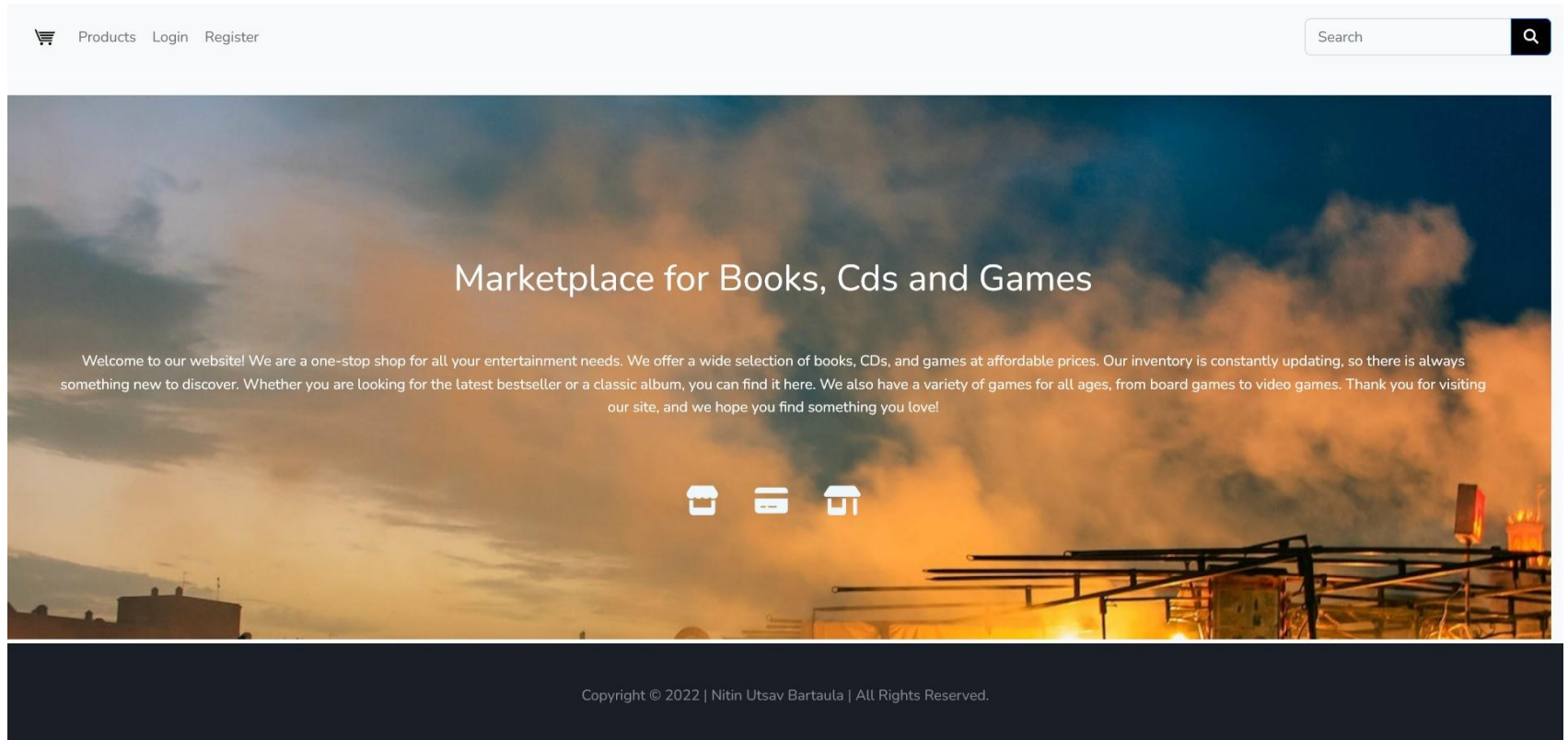
Installed: Bootstrap, Auth Scaffolding, npm install and npm run dev

### 1. Starting Laravel serve using -php artisan serve



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL POLYGLOT NOTEBOOK
PS C:\xampp\htdocs\Component 2 Final Web Applications\component-2-bnitin6034> php artisan serve
[INFO] Server running on [http://127.0.0.1:8000].
Press Ctrl+C to stop the server
[]
```

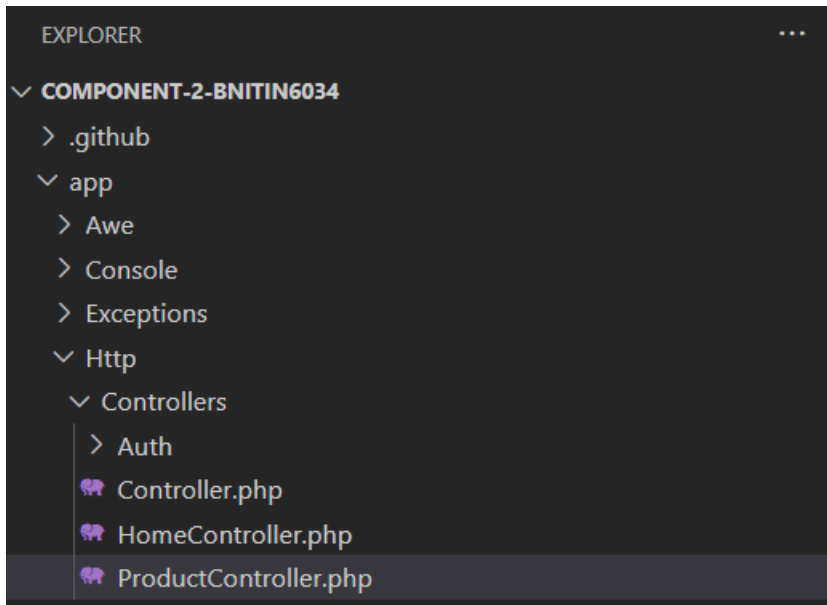
## Custom Made Homepage:



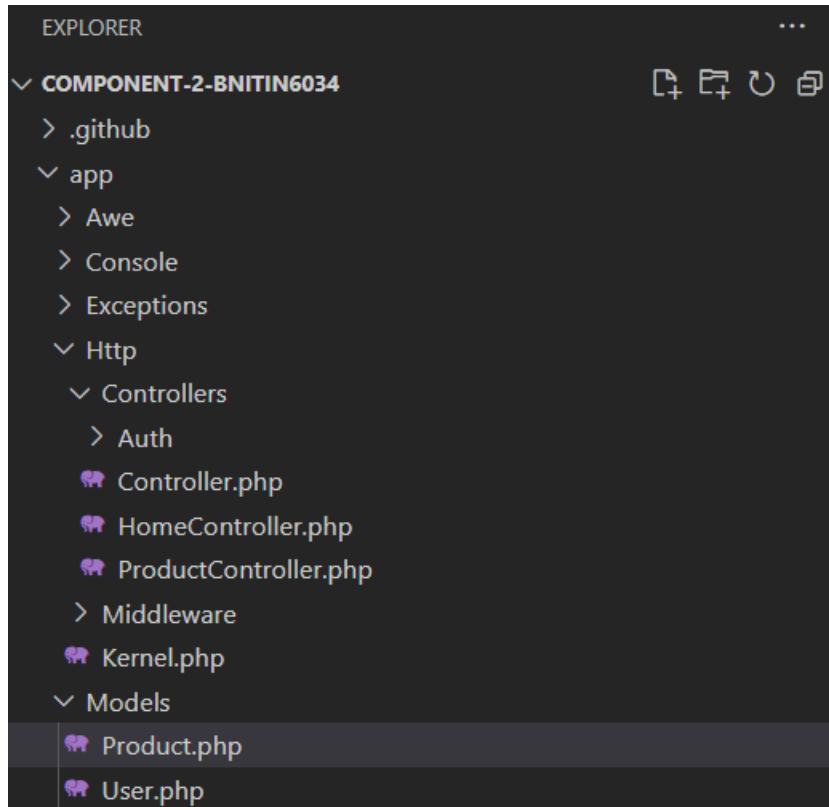
## 2. Creating a Resource Controller and Model using -php artisan make:controller ProductController -m Product

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL POLYGLOT NOTEBOOK  
PS C:\xampp\htdocs\Component 2 Final Web Applications\component-2-bniti6034> php artisan make:controller ProductController -m Product
```

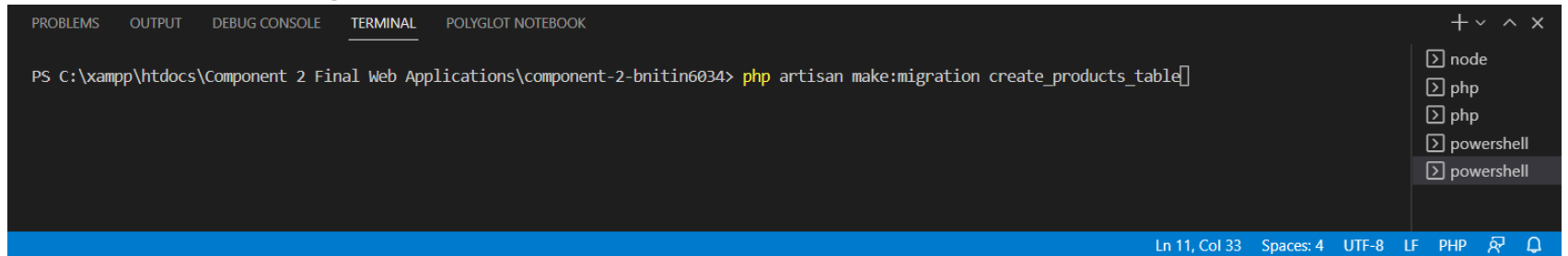
ProductController:



Model:



3. Creating a 'products' table using:  
-php artisan make:migration create\_products\_table



The screenshot shows a Visual Studio Code interface with a terminal window open. The terminal title bar includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is active), and POLYGLOT NOTEBOOK. The terminal content shows a PowerShell prompt at the path C:\xampp\htdocs\Component 2 Final Web Applications\component-2-bniti6034, followed by the command `php artisan make:migration create_products_table`. A context menu is open on the right side of the terminal, listing file types: node, php, powershell, and powershell (highlighted). The status bar at the bottom indicates the current line and column as Ln 11, Col 33, with other details like Spaces: 4, UTF-8, LF, and PHP.

```
PS C:\xampp\htdocs\Component 2 Final Web Applications\component-2-bniti6034> php artisan make:migration create_products_table
```

## Adding attributes to the database table

The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays a project structure for 'COMPONENT-2-BNITIN6034'. The file '2022\_12\_29\_175541\_create\_products\_table.php' is selected under the 'migrations' directory. The main editor displays the code for this migration file, which extends the 'Migration' class and defines a 'up()' method. The 'up()' method uses the 'Schema::create()' method to create a 'products' table with columns: 'id' (autoincrement), 'title' (string), 'firstname' (string, nullable), 'surname' (string), 'price' (float), 'pages' (float), 'image' (string), 'category' (string), and 'timestamps()'. The bottom panel shows the 'TERMINAL' tab with the command 'php artisan serve' and the output 'Server running on [http://127.0.0.1:8000]'. A list of tasks on the right includes 'node', 'php', and 'powershell'.

```
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('products', function (Blueprint $table) {
17             $table->id()->autoincrement();
18             $table->string('title');
19             $table->string('firstname')->nullable;
20             $table->string('surname');
21             $table->float('price');
22             $table->float('pages');
23             $table->string('image');
24             $table->string('category');
25             $table->timestamps();
26         });
27     }
28 }
```

PS C:\xampp\htdocs\Component 2 Final Web Applications\component-2-bniti6034> php artisan serve

INFO Server running on [http://127.0.0.1:8000].

Press Ctrl+C to stop the server

Running command: -php artisan migrate, to update mysql database

The screenshot displays the phpMyAdmin web interface. On the left, a sidebar shows the database structure with 'mnm' expanded to show tables like 'failed\_jobs', 'migrations', 'password\_resets', 'personal\_access\_tokens', 'products', and 'users'. The main panel shows the 'products' table selected. A green message bar at the top of the main panel states: 'MySQL returned an empty result set (i.e. zero rows). (Query took 0.0002 seconds.)'. Below this, the SQL query 'SELECT \* FROM `products`' is shown. A toolbar with options like 'Profiling', 'Edit inline', 'Edit', 'Explain SQL', 'Create PHP code', and 'Refresh' is visible. Below the query, a table header is shown: 'Id title firstname surname price pages image category created\_at updated\_at'. A 'Query results operations' dropdown menu is open, showing a 'Create view' option.

Server: 127.0.0.1 » Database: mnm » Table: products

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0002 seconds.)

SELECT \* FROM `products`

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Id	title	firstname	surname	price	pages	image	category	created_at	updated_at
----	-------	-----------	---------	-------	-------	-------	----------	------------	------------

Query results operations

Create view



#### 4. Using Factory and Seeder class to populate database with 15 'Faker' Data

```
public function definition()
{
    $gender = $this->faker->randomElement(['male', 'female']);
    return [
        'firstname' => $this->faker->firstName($gender),
        'surname' => $this->faker->lastName,
        // 'price' => $this->faker->numberBetween($min = 9.99, $max = 99.99),
        'price' => $this->faker->randomFloat(1,20,30,40),
        'category' => $this->faker->randomElement(['book', 'cd', 'game']),
    ];
}

public function makeBook()
{
    return $this->state(function (array $attributes) {
        return [
            'title' => $this->faker->randomElement(['Oliver Twist', 'Origin of Species', 'A Week with Gandhi', 'A Woman's Life', 'Aadhe Adhure', 'Adventures of Sherlock Holmes', 'Adventures of Tom Sawyer', 'Ain - i - Akbari', 'Akbarnama', 'Alice in Wonderland']),
            'image' => $this->faker->randomElement(['book1.jpg', 'book2.jpg', 'book3.jpg']),
            'pages' => $this->faker->numberBetween($min = 100, $max = 500),
            'category' => 'book',
        ];
    });
}

public function makeCd()
{
    return $this->state(function (array $attributes) {
        return [
            'title' => $this->faker->randomElement(['Led Zeppelin IV', 'Bad', 'Bat Out of Hell', 'Jagged Little Pill', 'Dirty Dancing', 'Falling into You', 'Metallica', 'Let's Talk About Love', 'Legend: The Best of Bob Marley & The Wailers', 'Appetite for Destruction']),
        ];
    });
}
```

```
1  <?php
2
3  namespace Database\Seeders;
4
5  use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6  use Illuminate\Database\Seeder;
7  use App\Models\Product;
8
9  class ProductSeeder extends Seeder
10 {
11     /**
12      * Run the database seeds.
13      *
14      * @return void
15      */
16     public function run()
17     {
18         Product::factory()->makeBook()->count(5)->create();
19         Product::factory()->makeCd()->count(5)->create();
20         Product::factory()->makeGame()->count(5)->create();
21     }
22 }
23
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  POLYGLOT NOTEBOOK: VARIABLES

PS C:\xampp\htdocs\Component 2 Final Web Applications\component-2-bniti6034> php artisan db:seed ProductSeeder
```

## 5. Index blade page and its corresponding Resource Controller function to display items

```
EXPLORER  ...  index.blade.php  ProductController.php

COMPONENT-2-BNITI6034
├── lang
├── node_modules
├── public
├── readme_images
├── resources
│   ├── css
│   ├── js
│   ├── sass
│   └── views
│       ├── auth
│       ├── layouts
│       ├── app.blade.php
│       └── products
│           ├── create.blade.php
│           ├── edit.blade.php
│           └── index.blade.php
│       ├── home.blade.php
│       └── welcome.blade.php
├── routes
│   ├── api.php
│   ├── channels.php
│   ├── console.php
│   ├── web.php
│   ├── storage
│   ├── tests
│   ├── vendor
│   ├── .editorconfig
│   ├── .env
│   ├── .env.example
│   ├── .gitattributes
│   └── .gitignore
├── OUTLINE
└── TIMELINE

resources > views > products > index.blade.php
1  @extends('layouts.app')
2
3  @section('content')
4  <link rel="stylesheet" href="{{ asset('css/global.css') }}">
5  <link rel="stylesheet" href="{{ asset('css/style.css') }}">
6  <script src="https://kit.fontawesome.com/f6dd6c55d1.js" crossorigin="anonymous"></script>
7
8  <div class="container">
9      <div class="col text-center">
10         <h1>All Products<i class="fa-solid fa-basket-shopping ms-3"></i></h1>
11         <hr>
12     </div>
13     <div class="col text-end mb-2">
14         <a href="{{ route('products.create') }}" class="btn btn-secondary" id="button-addnew">Add New</a>
15     </div>
16     <div class="row">
17         @foreach ($products as $product)
18             <div class="col-4 mb-2">
19                 <div class="card h-100 position-relative">
20                     
21                     <div class="card-body">
22                         <h5 class="card-title text-center">{{ $product -> title }}</h5>
23                         <p class="card-text">{{ $product -> firstname }}</p>
24                         <p class="card-text">{{ $product -> surname }}</p>
25                         <p class="card-text"><b> $ </b> {{ $product -> price }}</p>

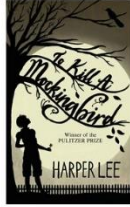
```

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\Product;
6  use Illuminate\Http\Request;
7
8  class ProductController extends Controller
9  {
10     /**
11      * Display a listing of the resource.
12      *
13      * @return \Illuminate\Http\Response
14      */
15     public function index()
16     {
17         // $products = Product::all();
18         $products = Product::latest()->get();
19         return view('products.index', compact('products'));
20     }
21 }
```



## All Products

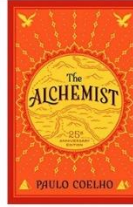
Add New



Alice in Wonderland

Jessy  
McDermott  
\$ 23.1  
Page 426

Select



Origin of Species

Bernhard  
Kub  
\$ 21.6  
Page 241

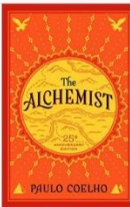
Select



Origin of Species

Jeffery  
Prosacco  
\$ 22.6  
Page 294

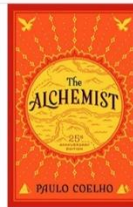
Select



Ain - i - Akbari

Florida  
Berge  
\$ 25  
Page 224

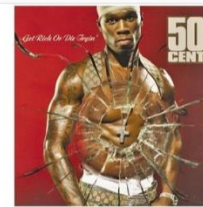
Select



A Woman's Life

Gracie  
Streich  
\$ 22.3  
Page 483

Select



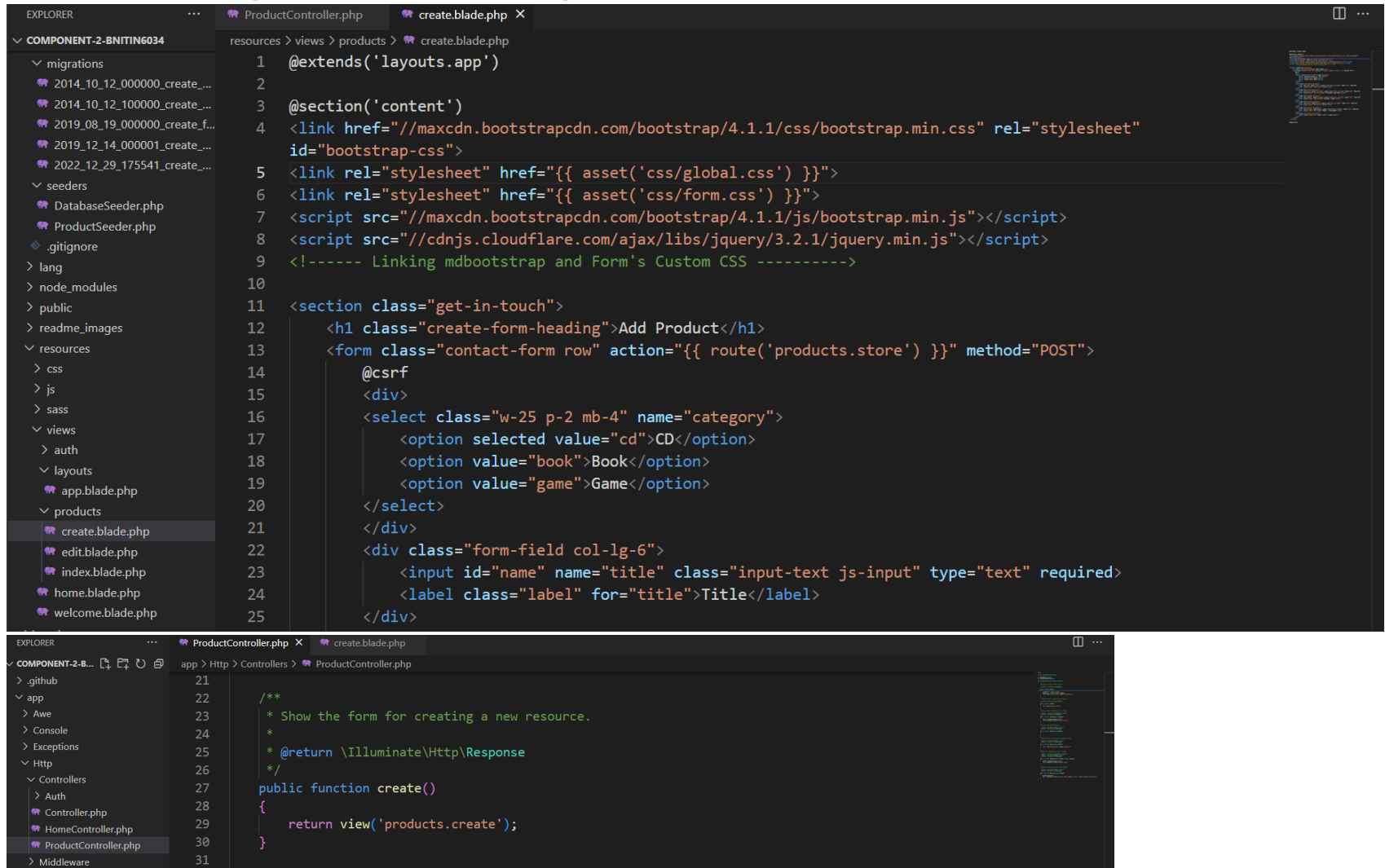
Dirty Dancing

Doris  
Grant  
\$ 26.2  
Playlength 41 min

Select



## 6. Add Item Form page and its corresponding Resource Controller function



The image displays two screenshots of a code editor, likely Visual Studio Code, showing the implementation of a product creation form and its corresponding controller.

**Top Screenshot: Blade Template (create.blade.php)**

```
1 @extends('layouts.app')
2
3 @section('content')
4 <link href="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css" rel="stylesheet"
  id="bootstrap-css">
5 <link rel="stylesheet" href="{{ asset('css/global.css') }}">
6 <link rel="stylesheet" href="{{ asset('css/form.css') }}">
7 <script src="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"></script>
8 <script src="//cdn.jsdelivr.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
9 <!-- Linking mdbbootstrap and Form's Custom CSS -->
10
11 <section class="get-in-touch">
12   <h1 class="create-form-heading">Add Product</h1>
13   <form class="contact-form row" action="{{ route('products.store') }}" method="POST">
14     @csrf
15     <div>
16       <select class="w-25 p-2 mb-4" name="category">
17         <option selected value="cd">CD</option>
18         <option value="book">Book</option>
19         <option value="game">Game</option>
20       </select>
21     </div>
22     <div class="form-field col-lg-6">
23       <input id="name" name="title" class="input-text js-input" type="text" required>
24       <label class="label" for="title">Title</label>
25     </div>
26   </form>
27 </section>
```

**Bottom Screenshot: PHP Controller (ProductController.php)**

```
21
22 /**
23  * Show the form for creating a new resource.
24  *
25  * @return \Illuminate\Http\Response
26  */
27 public function create()
28 {
29     return view('products.create');
30 }
31
```



## ADD PRODUCT

CD ▾

Title

Firstname (optional)

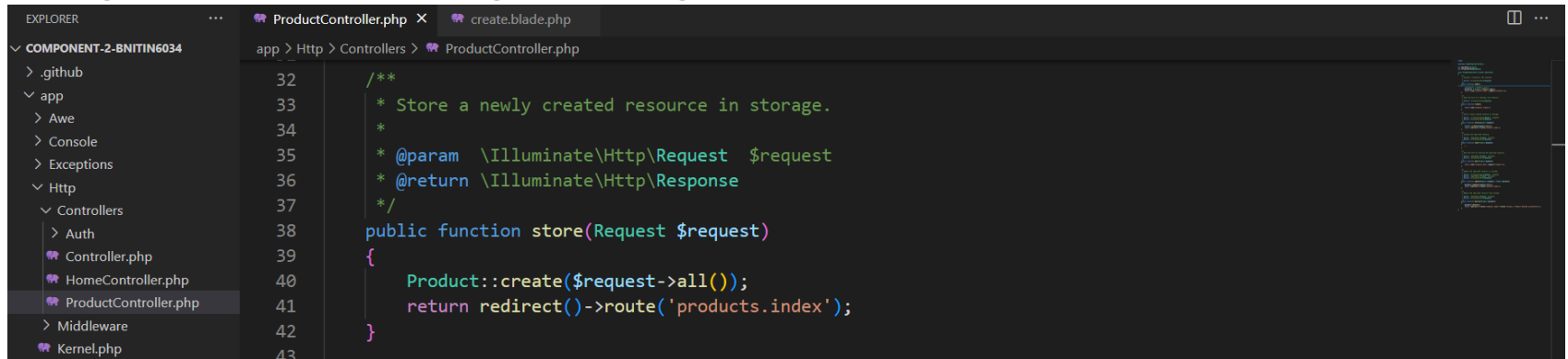
Surname / Band

Price

Pages / Playlength

SUBMIT

## 7. Adding data to the db table using form. Using 'Store' function of Resource Controller to do so



The screenshot shows a code editor with the following structure:

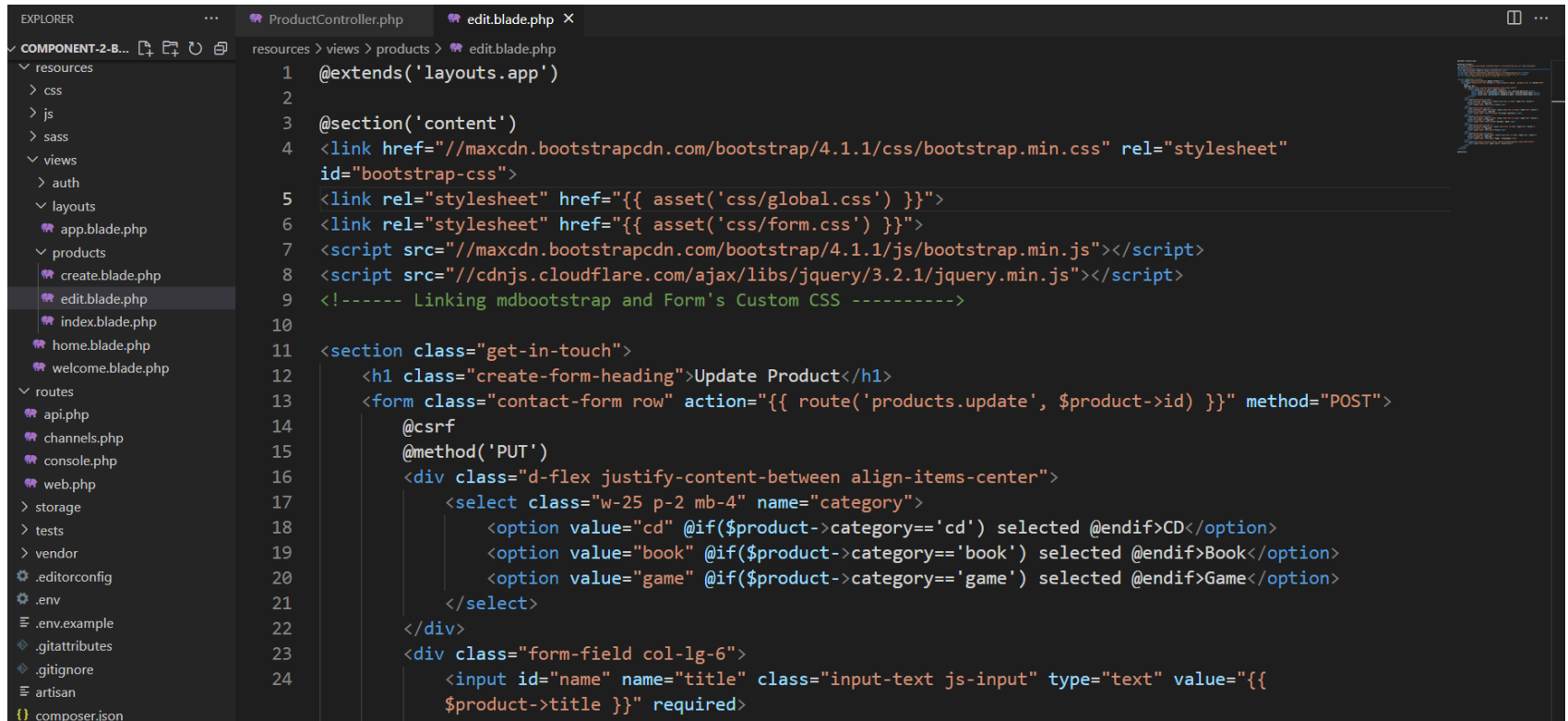
- EXPLORER** (Left Panel):
  - COMPONENT-2-BNITIN6034
    - .github
    - app
      - Awe
      - Console
      - Exceptions
      - Http
        - Controllers
          - Auth
          - Controller.php
          - HomeController.php
          - ProductController.php** (Selected)
          - Middleware
          - Kernel.php

- ProductController.php** (Main Editor):

```
32 /**
33  * Store a newly created resource in storage.
34  *
35  * @param \Illuminate\Http\Request $request
36  * @return \Illuminate\Http\Response
37  */
38 public function store(Request $request)
39 {
40     Product::create($request->all());
41     return redirect()->route('products.index');
42 }
43
```



## 8. Modify table data using Form. Using edit.blade.php and its corresponding controller function 'edit' and 'update'




```
1 @extends('layouts.app')
2
3 @section('content')
4 <link href="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css" rel="stylesheet"
  id="bootstrap-css">
5 <link rel="stylesheet" href="{{ asset('css/global.css') }}">
6 <link rel="stylesheet" href="{{ asset('css/form.css') }}">
7 <script src="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"></script>
8 <script src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
9 <!-- Linking mdbootstrap and Form's Custom CSS -->
10
11 <section class="get-in-touch">
12   <h1 class="create-form-heading">Update Product</h1>
13   <form class="contact-form row" action="{{ route('products.update', $product->id) }}" method="POST">
14     @csrf
15     @method('PUT')
16     <div class="d-flex justify-content-between align-items-center">
17       <select class="w-25 p-2 mb-4" name="category">
18         <option value="cd" @if($product->category=='cd') selected @endif>CD</option>
19         <option value="book" @if($product->category=='book') selected @endif>Book</option>
20         <option value="game" @if($product->category=='game') selected @endif>Game</option>
21       </select>
22     </div>
23     <div class="form-field col-lg-6">
24       <input id="name" name="title" class="input-text js-input" type="text" value="{{
        $product->title }}" required>
```


```
COMPONENT-2-B...  app > Http > Controllers > ProductController.php


> .github
> app
> Awe
> Console
> Exceptions
> Http
  > Controllers
    > Auth
    ProductController.php
    HomeController.php
    ProductController.php
  > Middleware
    Kernel.php
  > Models
    Product.php
    User.php
  > Providers
> bootstrap
> config
> database
  > factories
    ProductFactory.php
    UserFactory.php
  > migrations
    2014_10_12_000000_create_...
    2014_10_12_100000_create_...
    2019_08_19_000000_create_f...
    2019_12_14_000001_create_...
    2022_12_29_175541_create_...

53 }
54
55 /**
56  * Show the form for editing the specified resource.
57  *
58  * @param \App\Models\Product $product
59  * @return \Illuminate\Http\Response
60  */
61 public function edit(Product $product)
62 {
63     return view('products.edit',compact('product'));
64 }
65
66 /**
67  * Update the specified resource in storage.
68  *
69  * @param \Illuminate\Http\Request $request
70  * @param \App\Models\Product $product
71  * @return \Illuminate\Http\Response
72  */
73 public function update(Request $request, Product $product)
74 {
75     $product->update($request->all());
76     return redirect()->route('products.index');
77 }
78
```

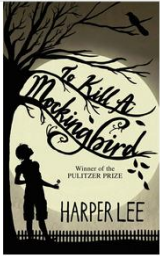
To change the title of this card

 Products Login Register

Search 

All Products 

Add New



Alice in Wonderland

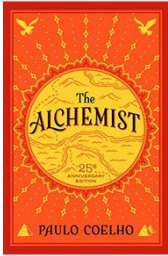
Jessy

McDermott

\$ 23.1

Page 426

Select



Origin of Species

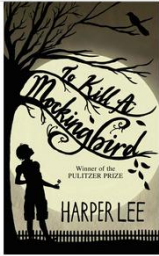
Bernhard

Kub

\$ 21.6

Page 241

Select



Origin of Species

Jeffery

Prosacco

\$ 22.6

Page 294

Select

PAGE 18

## Changing title 'Alice in Wonderland' to 'Alice in Borderland' and pressing submit

[logo](#) [Products](#) [Login](#) [Register](#)



# UPDATE PRODUCT

Book ▼

Title

Alice in Borderland

Firstname (optional)

Jessy

Surname / Band

McDermott

Price


23.1


Pages / Playlength

426

SUBMIT

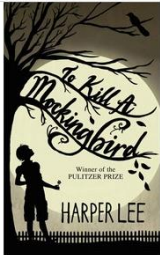
Title Updated Successfully

 Products Login Register

Search 

## All Products

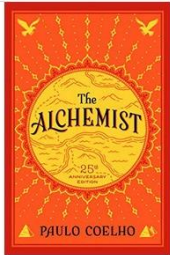
Add New



Alice in Borderland

Jessy  
McDermott  
\$ 23.1  
Page 426

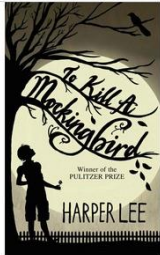
Select



Origin of Species

Bernhard  
Kub  
\$ 21.6  
Page 241

Select



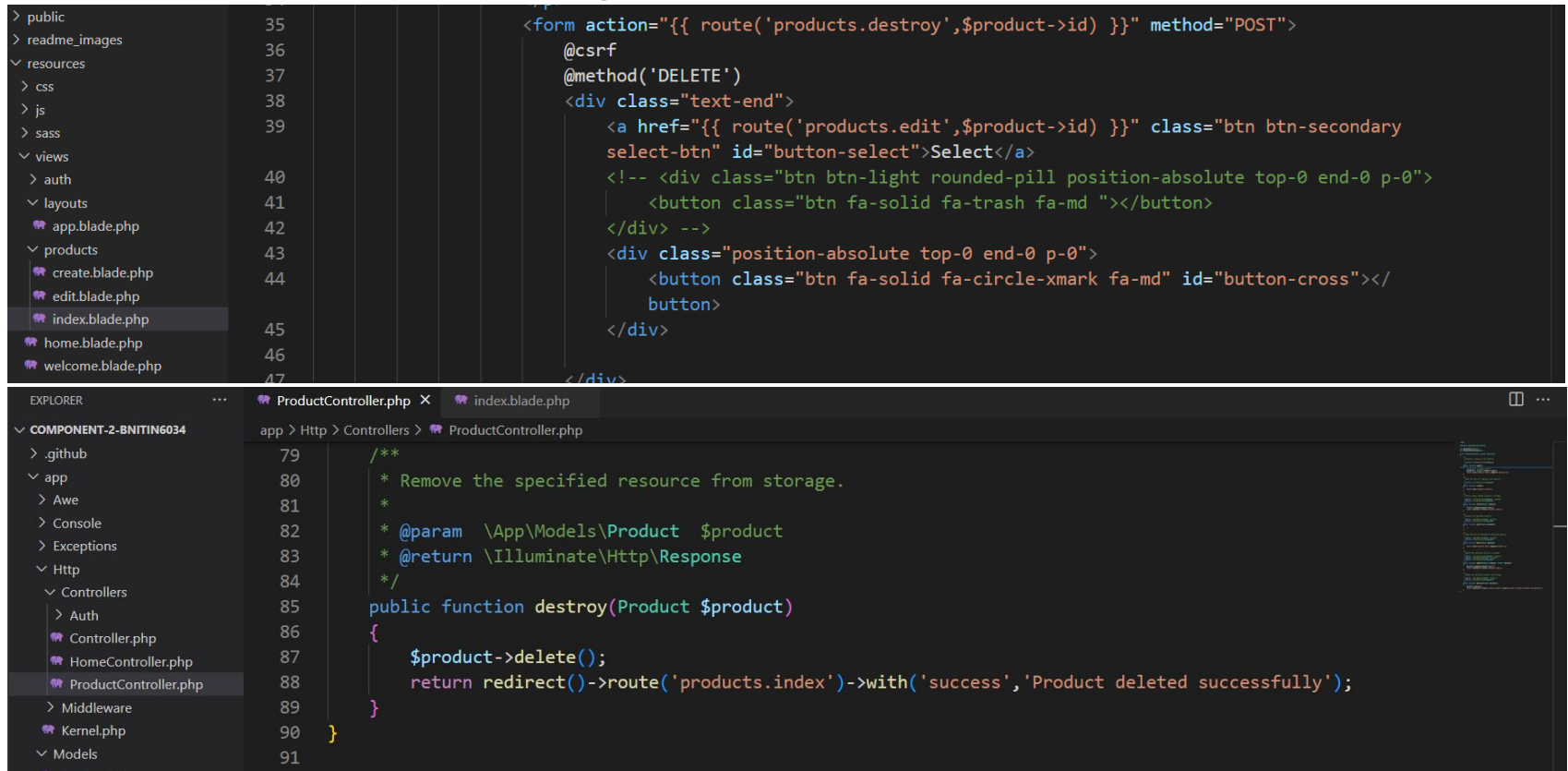
Origin of Species

Jeffery  
Prosacco  
\$ 22.6  
Page 294

Select

PAGE 20

9. Lastly, for our Delete operation, creating a form to use destroy function





The screenshot displays two files in a code editor. The top file, `index.blade.php`, contains a form for deleting a product. The bottom file, `ProductController.php`, contains the `destroy` method.


```
<form action="{{ route('products.destroy',$product->id) }}" method="POST">
    @csrf
    @method('DELETE')
    <div class="text-end">
        <a href="{{ route('products.edit',$product->id) }}" class="btn btn-secondary
        select-btn" id="button-select">Select</a>
        <!-- <div class="btn btn-light rounded-pill position-absolute top-0 end-0 p-0">
            <button class="btn fa-solid fa-trash fa-md "></button>
        </div> -->
        <div class="position-absolute top-0 end-0 p-0">
            <button class="btn fa-solid fa-circle-xmark fa-md" id="button-cross"></
            button>
        </div>
    </div>
</form>
```

```
/**
 * Remove the specified resource from storage.
 *
 * @param \App\Models\Product $product
 * @return \Illuminate\Http\Response
 */
public function destroy(Product $product)
{
    $product->delete();
    return redirect()->route('products.index')->with('success','Product deleted successfully');
}
```


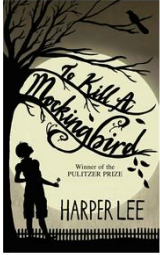
Deleting this book titled 'Alice in Borderland' using the top right cross icon

 Products Login Register

Search 

All Products 


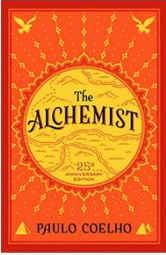
Add New



Alice in Wonderland

Jessy  
McDermott  
\$ 23.1  
Page 426


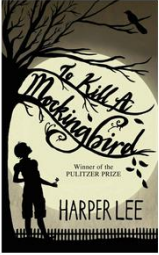
Select



Origin of Species

Bernhard  
Kub  
\$ 21.6  
Page 241

Select





Origin of Species


Jeffery  
Prosacco  
\$ 22.6  
Page 294

Select

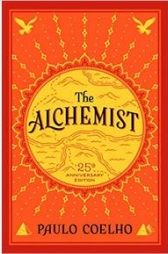
And just like that, there is no more any record of a book called ‘Alice in Borderland’

 Products Login Register

Search 

All Products 

Add New



Origin of Species

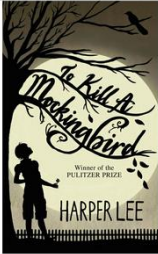
Bernhard

Kub

\$ 21.6

Page 241

Select



Origin of Species

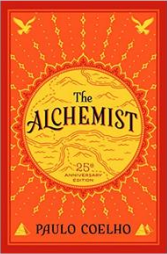
Jeffery

Prosacco

\$ 22.6

Page 294

Select



Ain - i - Akbari


Florida


Berge

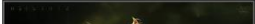
\$ 25

Page 224

Select







PAGE 23



END