DOCUMENTATION Component-3

Name: Sushant Chapagain

University ID: 77227166

Group: A

INCLUDES IN APPLICATION:

- VIEWS (layout homepage admin dashboard etc.)
- Routes
- Laravel Breeze With Admin Middleware
- Models (User Category and Product)
- Controller (ProductContoller and DashboardController)
- Pagination
- Mail chimp API
- CRUD
- Number of Products and User
- MySQL database

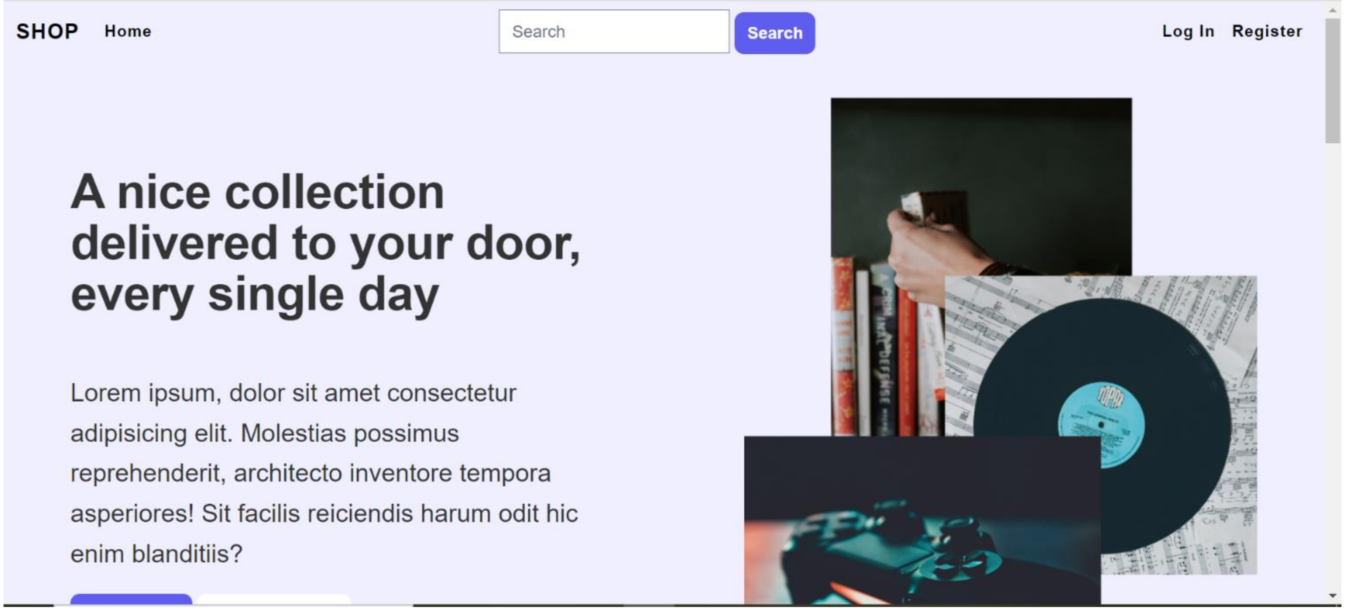
GITHUB REPO LINK:

❖ *https://github.com/tbc-bsc-l6/component-2-sushantchapagain1*

YOUTUBE DEMO LINK:

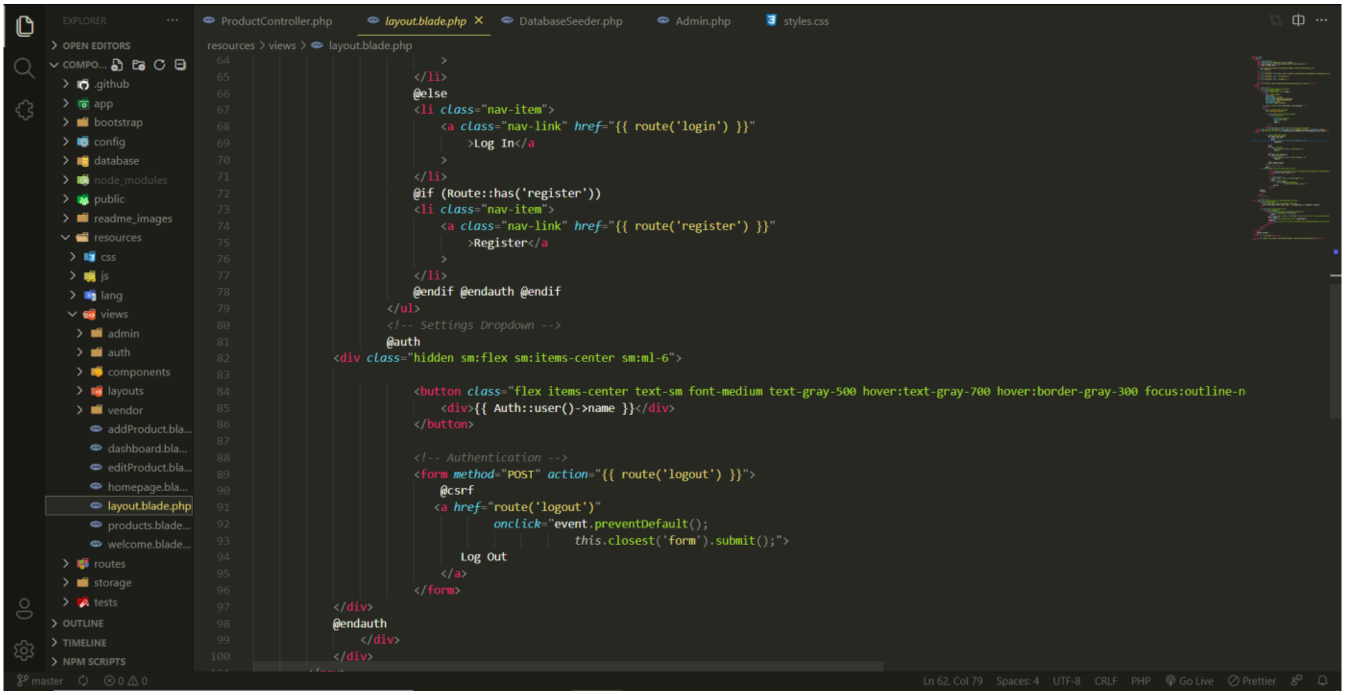
❖ https://www.youtube.com/watch?v=w6QBgoU5ZKk&t

VIEWS:

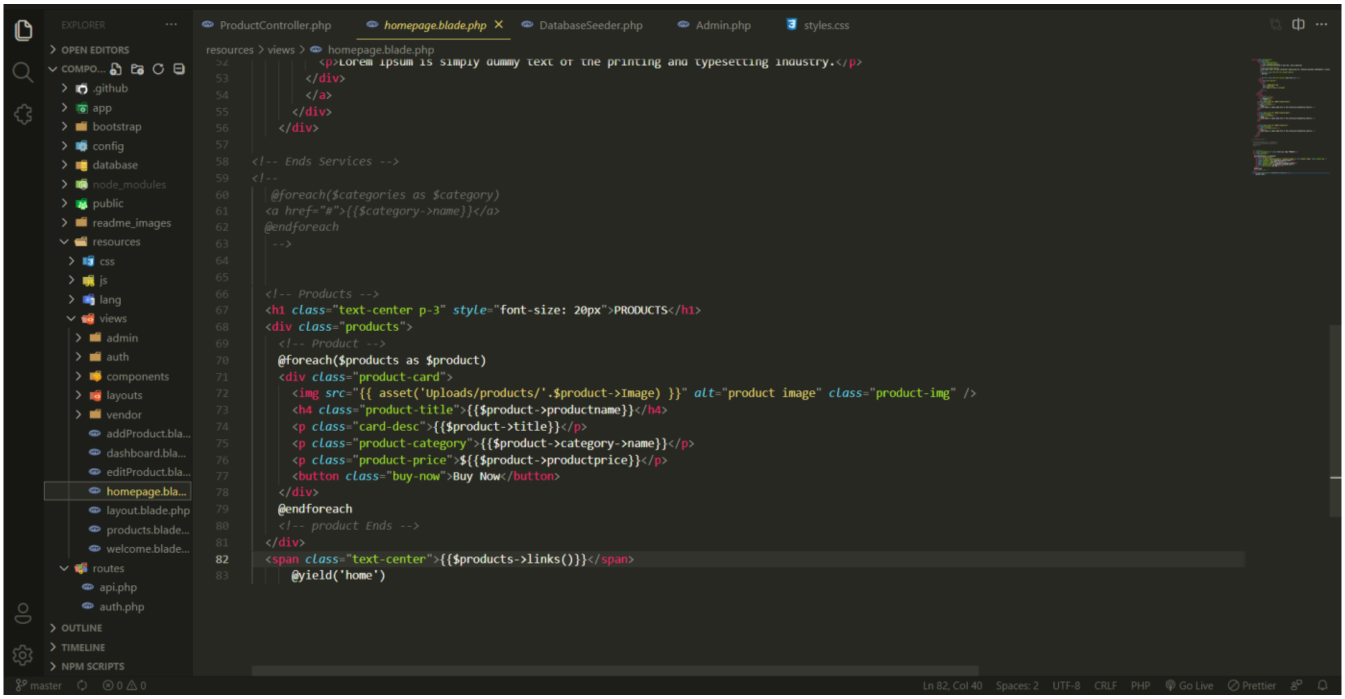


A nice collection delivered to your door, every single day

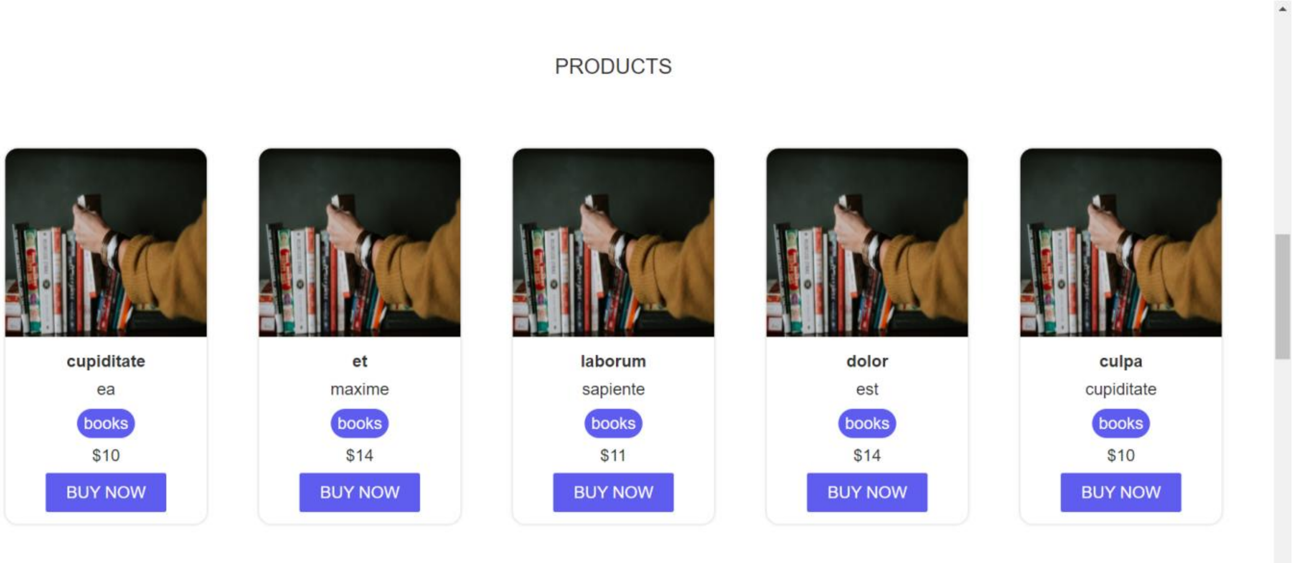
Lorem ipsum, dolor sit amet consectetur adipisicing elit. Molestias possimus reprehenderit, architecto inventore tempora asperiores! Sit facilis reiciendis harum odit hic enim blanditiis?



```
                    >
                </li>
                @else
                <li class="nav-item">
                    <a class="nav-link" href="{{ route('login') }}"
                        >Log In</a
                    >
                </li>
                @if (Route::has('register'))
                <li class="nav-item">
                    <a class="nav-link" href="{{ route('register') }}"
                        >Register</a
                    >
                </li>
                @endif @endauth @endif
            </ul>
            <!-- Settings Dropdown -->
            @auth
<div class="hidden sm:flex sm:items-center sm:ml-6">

                <button class="flex items-center text-sm font-medium text-gray-500 hover:text-gray-700 hover:border-gray-300 focus:outline-n
                    <div>{{ Auth::user()->name }}</div>
                </button>

                <!-- Authentication -->
                <form method="POST" action="{{ route('logout') }}">
                    @csrf
                   <a href="route('logout')"
                            onclick="event.preventDefault();
                                        this.closest('form').submit();">
                      Log Out
                    </a>
                </form>
</div>
@endauth
    </div>
</div>
```



```
          <p>Lorem ipsum is simply dummy text of the printing and typesetting industry.</p>
        </div>
        </a>
      </div>
    </div>

<!-- Ends Services -->
<!--
   @foreach($categories as $category)
  <a href="#">{{$category->name}}</a>
  @endforeach
   -->


  <!-- Products -->
  <h1 class="text-center p-3" style="font-size: 20px">PRODUCTS</h1>
  <div class="products">
    <!-- Product -->
    @foreach($products as $product)
    <div class="product-card">
      <img src="{{ asset('Uploads/products/'.$product->Image) }}" alt="product image" class="product-img" />
      <h4 class="product-title">{{$product->productname}}</h4>
      <p class="card-desc">{{$product->title}}</p>
      <p class="product-category">{{$product->category->name}}</p>
      <p class="product-price">${{$product->productprice}}</p>
      <button class="buy-now">Buy Now</button>
    </div>
    @endforeach
    <!-- product Ends -->
  </div>
  <span class="text-center">{{$products->links()}}</span>
      @yield('home')
```



PRODUCTS

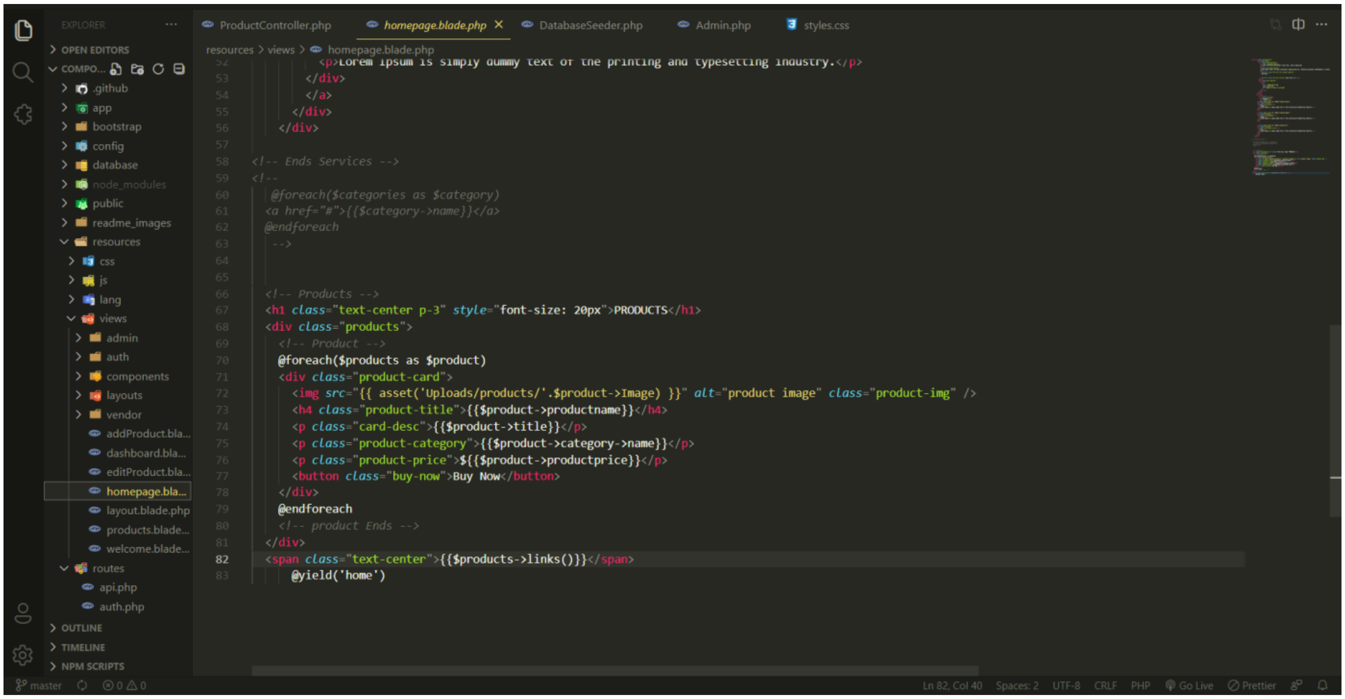
cupiditate
ea
books
$10
BUY NOW

et
maxime
books
$14
BUY NOW

laborum
sapiente
books
$11
BUY NOW

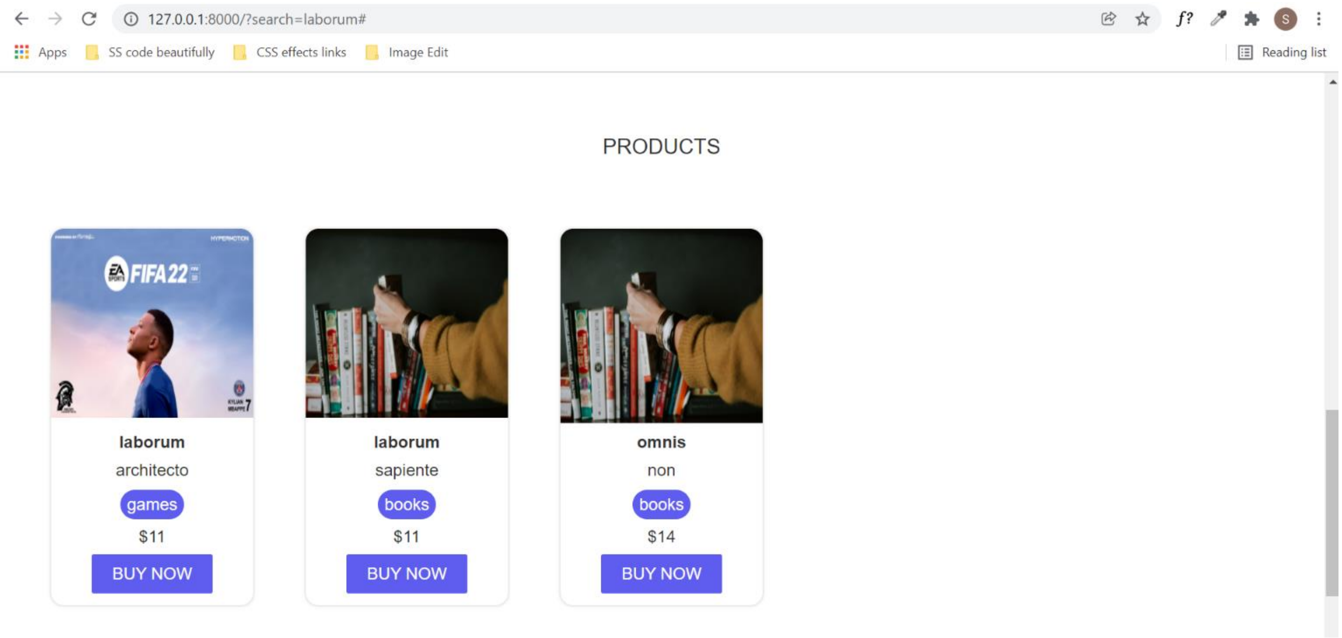
dolor
est
books
$14
BUY NOW

culpa
cupiditate
books
$10
BUY NOW



```
          <p>Lorem ipsum is simply dummy text of the printing and typesetting industry.</p>
        </div>
        </a>
      </div>
    </div>

<!-- Ends Services -->
<!--
   @foreach($categories as $category)
  <a href="#">{{$category->name}}</a>
  @endforeach
   -->


  <!-- Products -->
  <h1 class="text-center p-3" style="font-size: 20px">PRODUCTS</h1>
  <div class="products">
    <!-- Product -->
    @foreach($products as $product)
    <div class="product-card">
      <img src="{{ asset('Uploads/products/'.$product->Image) }}" alt="product image" class="product-img" />
      <h4 class="product-title">{{$product->productname}}</h4>
      <p class="card-desc">{{$product->title}}</p>
      <p class="product-category">{{$product->category->name}}</p>
      <p class="product-price">${{$product->productprice}}</p>
      <button class="buy-now">Buy Now</button>
    </div>
    @endforeach
    <!-- product Ends -->
  </div>
  <span class="text-center">{{$products->links()}}</span>
      @yield('home')
```

After sending from the route it goes to Product Controller and return the product and category variable to this view.



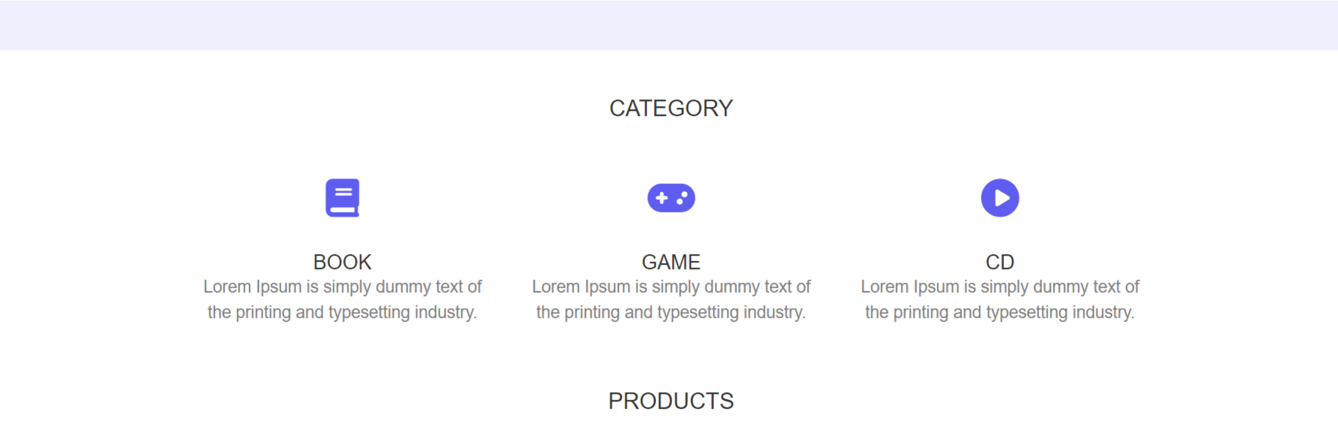
PRODUCTS

laborum
architecto
games
$11
BUY NOW

laborum
sapiente
books
$11
BUY NOW

omnis
non
books
$14
BUY NOW

After searching a product with laborum we can see that in URL also.



CATEGORY

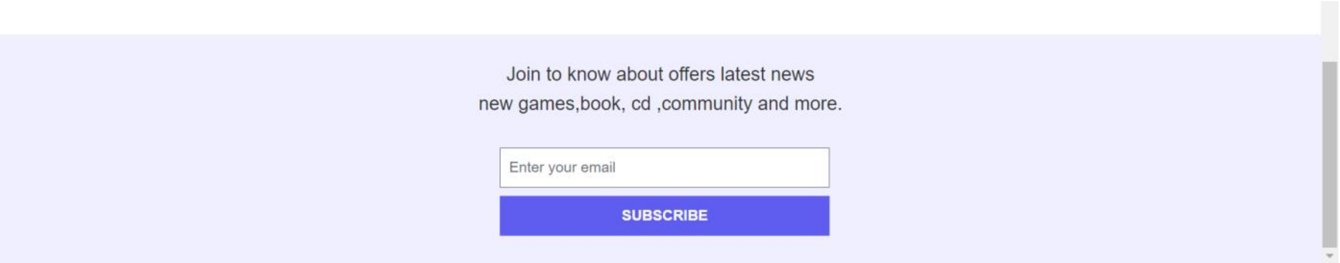
BOOK
Lorem Ipsum is simply dummy text of the printing and typesetting industry.

GAME
Lorem Ipsum is simply dummy text of the printing and typesetting industry.

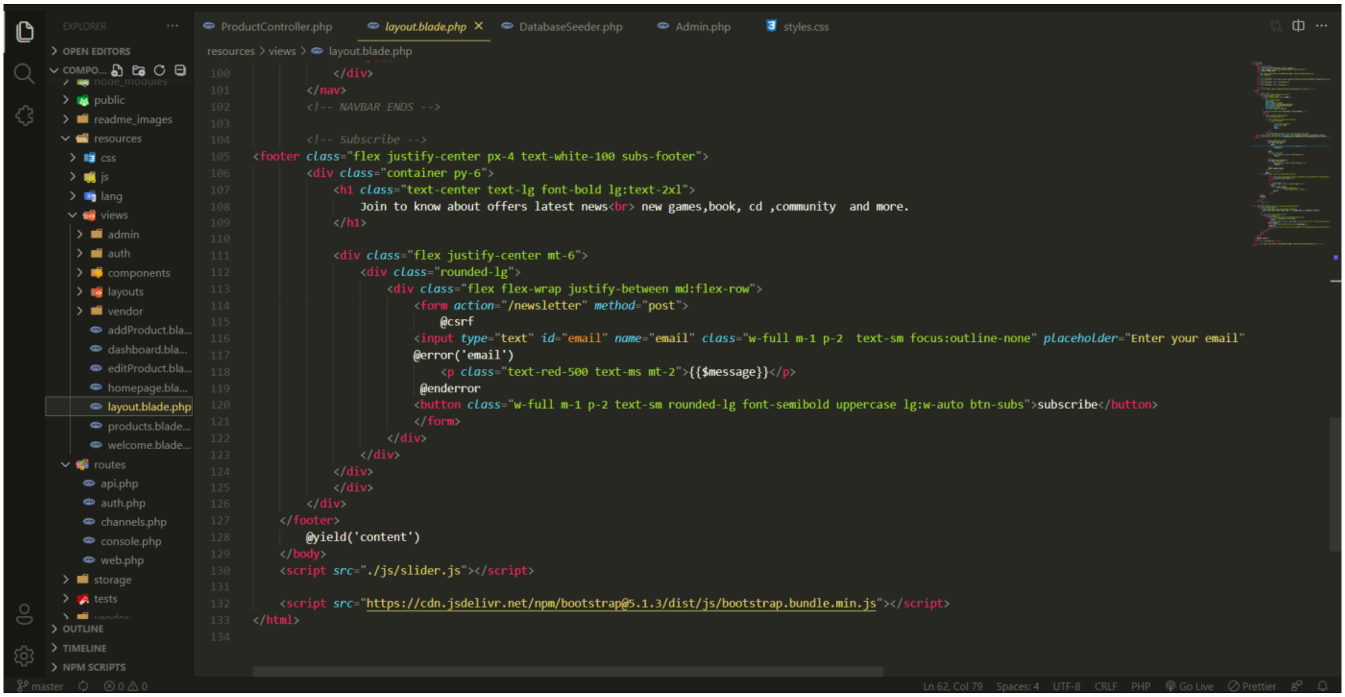
CD
Lorem Ipsum is simply dummy text of the printing and typesetting industry.

PRODUCTS

Category coming from database table.

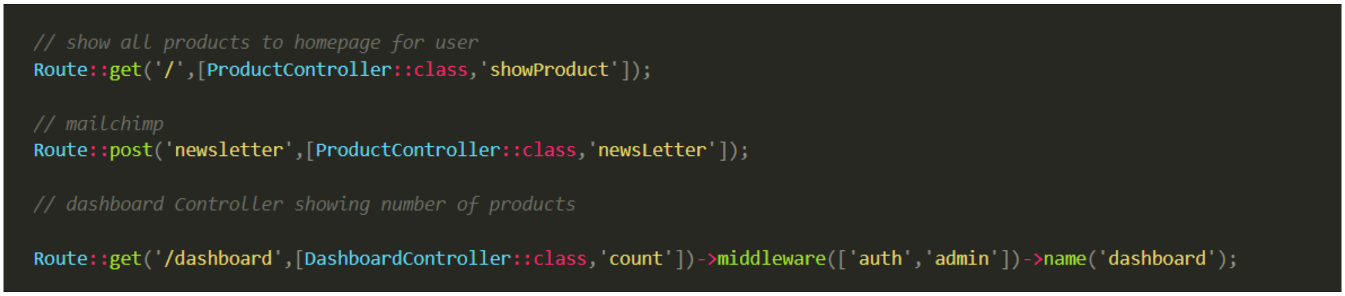


Join to know about offers latest news
new games,book, cd ,community and more.


ProductController.php
layout.blade.php
DatabaseSeeder.php
Admin.php
styles.css
resources > views > layout.blade.php
Ln 62, Col 79 Spaces: 4 UTF-8 CRLF PHP Go Live Prettier

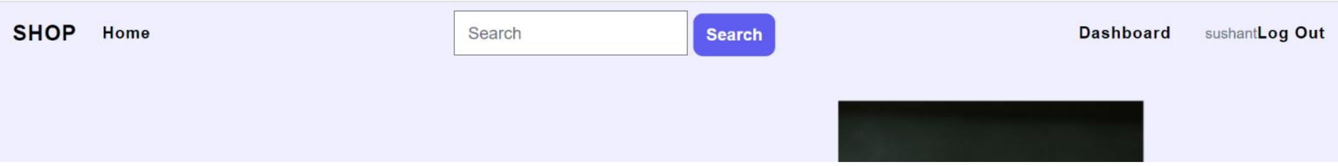
```
            </div>
        </nav>
        <!-- NAVBAR ENDS -->

        <!-- Subscribe -->
<footer class="flex justify-center px-4 text-white-100 subs-footer">
        <div class="container py-6">
            <h1 class="text-center text-lg font-bold lg:text-2xl">
                Join to know about offers latest news<br> new games,book, cd ,community  and more.
            </h1>

            <div class="flex justify-center mt-6">
                <div class="rounded-lg">
                    <div class="flex flex-wrap justify-between md:flex-row">
                        <form action="/newsletter" method="post">
                            @csrf
                        <input type="text" id="email" name="email" class="w-full m-1 p-2  text-sm focus:outline-none" placeholder="Enter your email"
                        @error('email')
                            <p class="text-red-500 text-ms mt-2">{{$message}}</p>
                         @enderror
                        <button class="w-full m-1 p-2 text-sm rounded-lg font-semibold uppercase lg:w-auto btn-subs">subscribe</button>
                        </form>
                    </div>
                </div>
            </div>
            </div>
        </div>
    </footer>
        @yield('content')
    </body>
    <script src="./js/slider.js"></script>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</html>
```

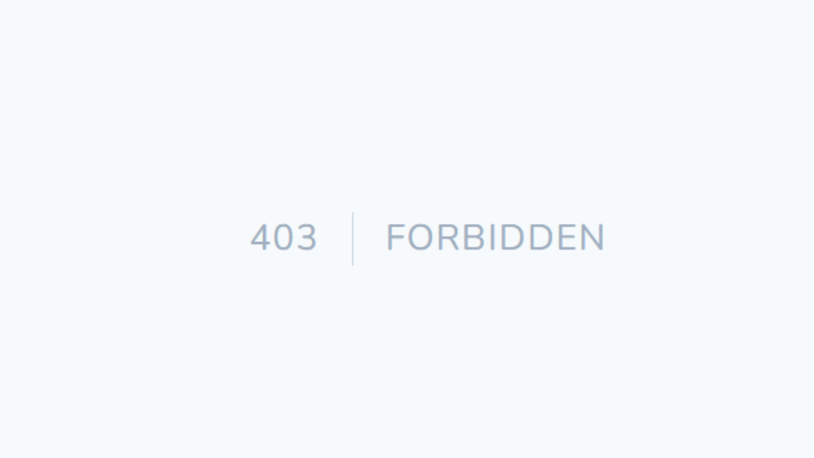
Code of newsletter view.



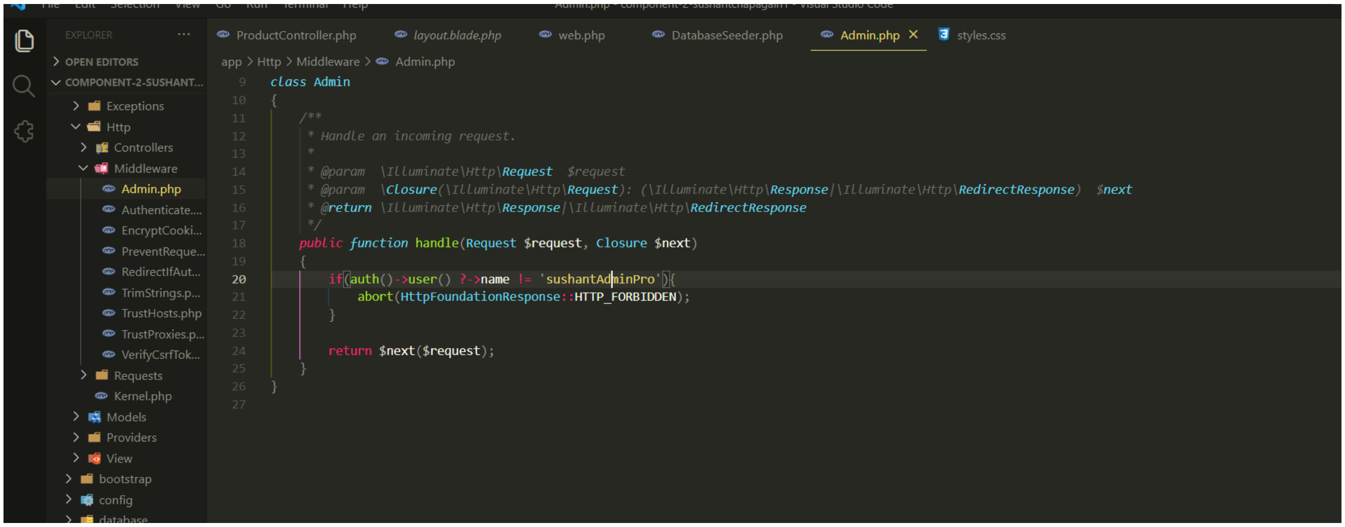
```
// show all products to homepage for user
Route::get('/',[ProductController::class,'showProduct']);

// mailchimp
Route::post('newsletter',[ProductController::class,'newsLetter']);

// dashboard Controller showing number of products

Route::get('/dashboard',[DashboardController::class,'count'])->middleware(['auth','admin'])->name('dashboard');
```

Routes for Showing Products in homepage.





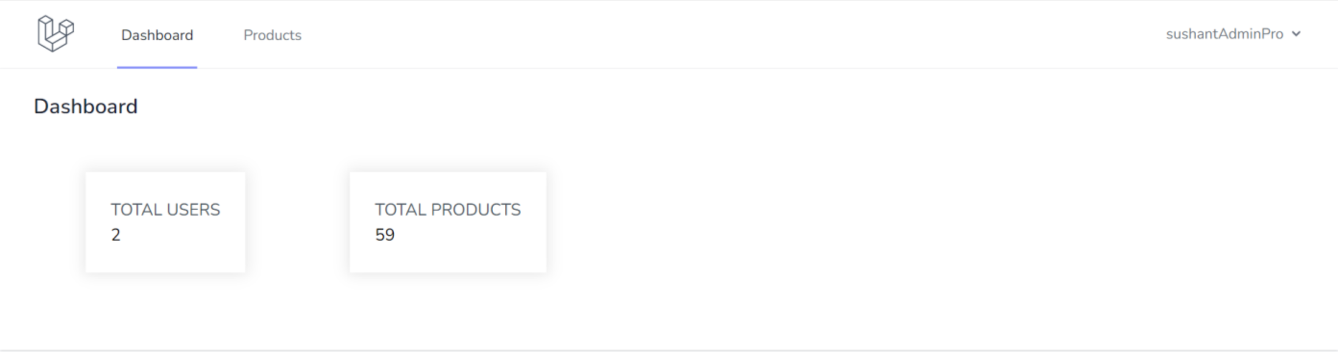
403 | FORBIDDEN



```php
class Admin
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure(\Illuminate\Http\Request): (\Illuminate\Http\Response|\Illuminate\Http\RedirectResponse)  $next
     * @return \Illuminate\Http\Response|\Illuminate\Http\RedirectResponse
     */
    public function handle(Request $request, Closure $next)
    {
        if(auth()->user() ?->name != 'sushantAdminPro'){
            abort(HttpFoundationResponse::HTTP_FORBIDDEN);
        }

        return $next($request);
    }
}
```

In Laravel breeze by default everyone is logged in in admin so we have to create middleware to prevent it.

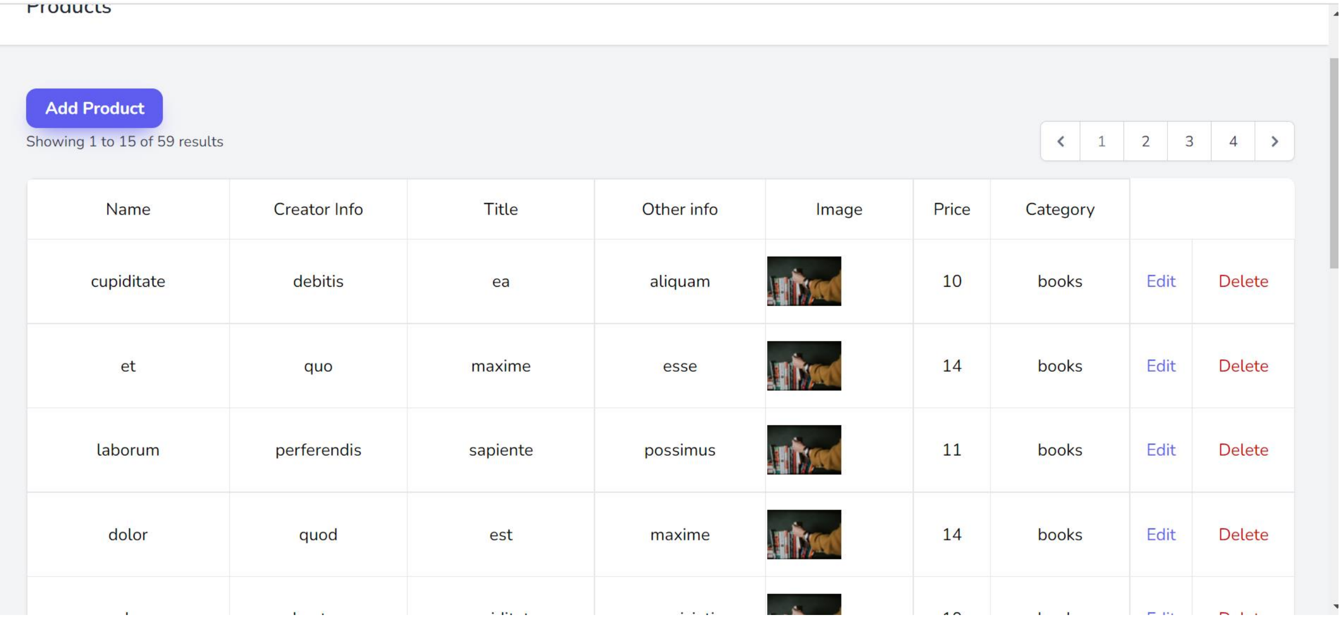
Admin View



Dashboard

TOTAL USERS
2

TOTAL PRODUCTS
59

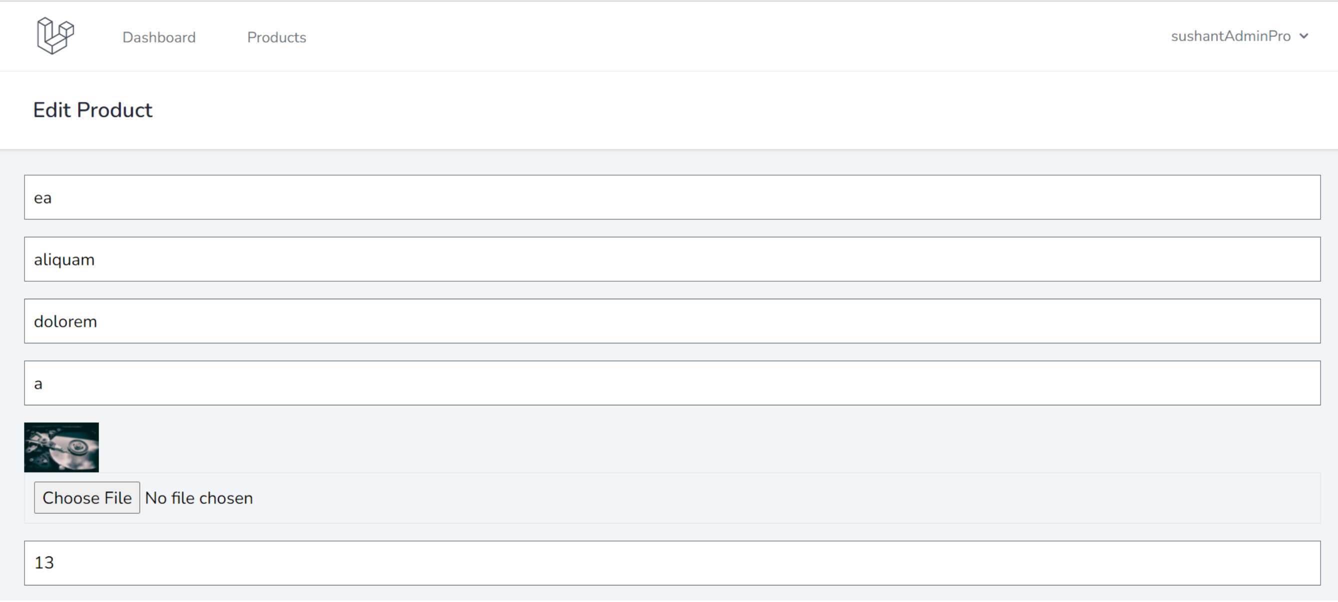


Products

Add Product

Showing 1 to 15 of 59 results

Name	Creator Info	Title	Other info	Image	Price	Category		
cupiditate	debitis	ea	aliquam		10	books	Edit	Delete
et	quo	maxime	esse		14	books	Edit	Delete
laborum	perferendis	sapiente	possimus		11	books	Edit	Delete
dolor	quod	est	maxime		14	books	Edit	Delete



Edit Product

ea

aliquam

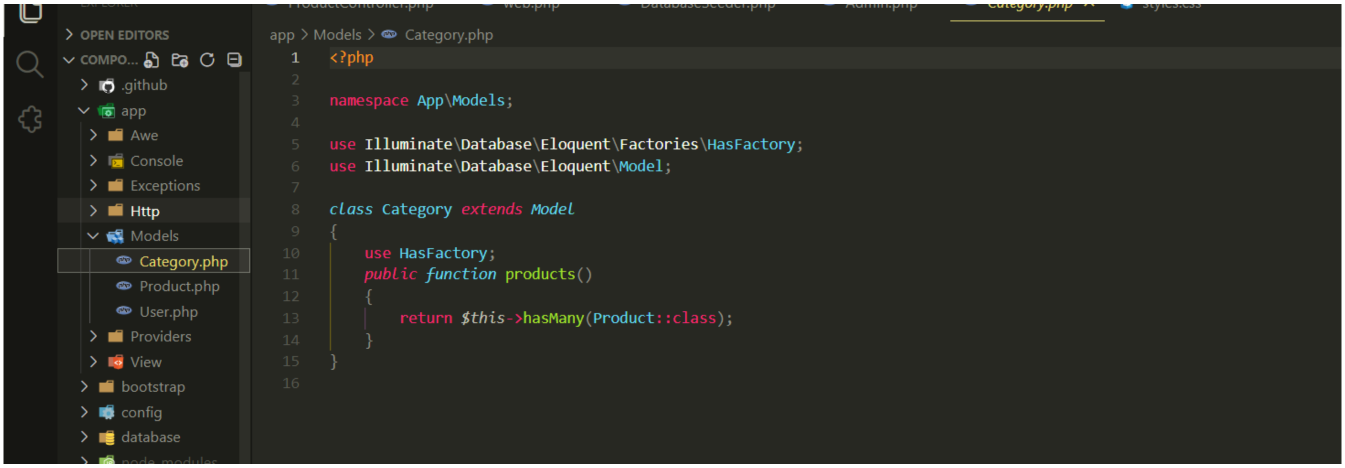
dolorem

a

13

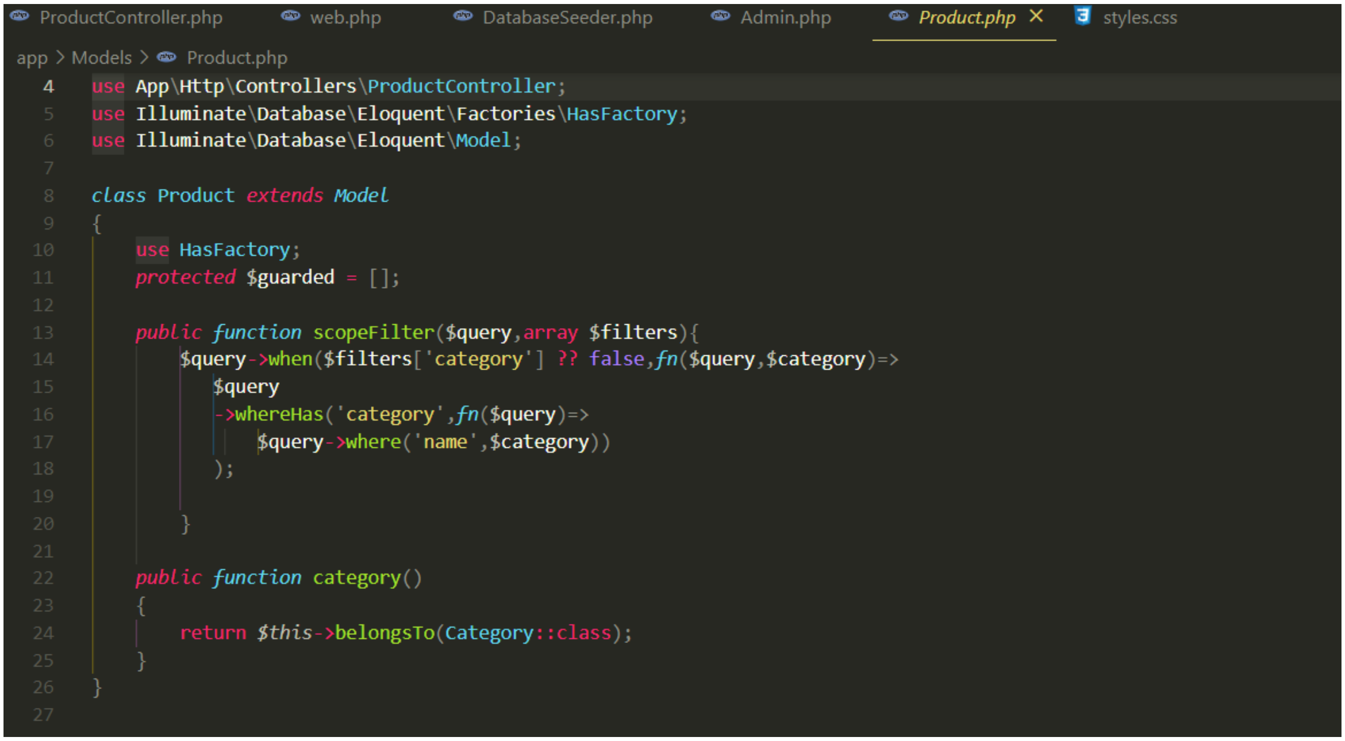
Edit with all old data.

Model



```php
<?php

namespace App\Models;

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

class Category extends Model
{
    use HasFactory;
    public function products()
    {
        return $this->hasMany(Product::class);
    }
}
```

Category model with relation to product model

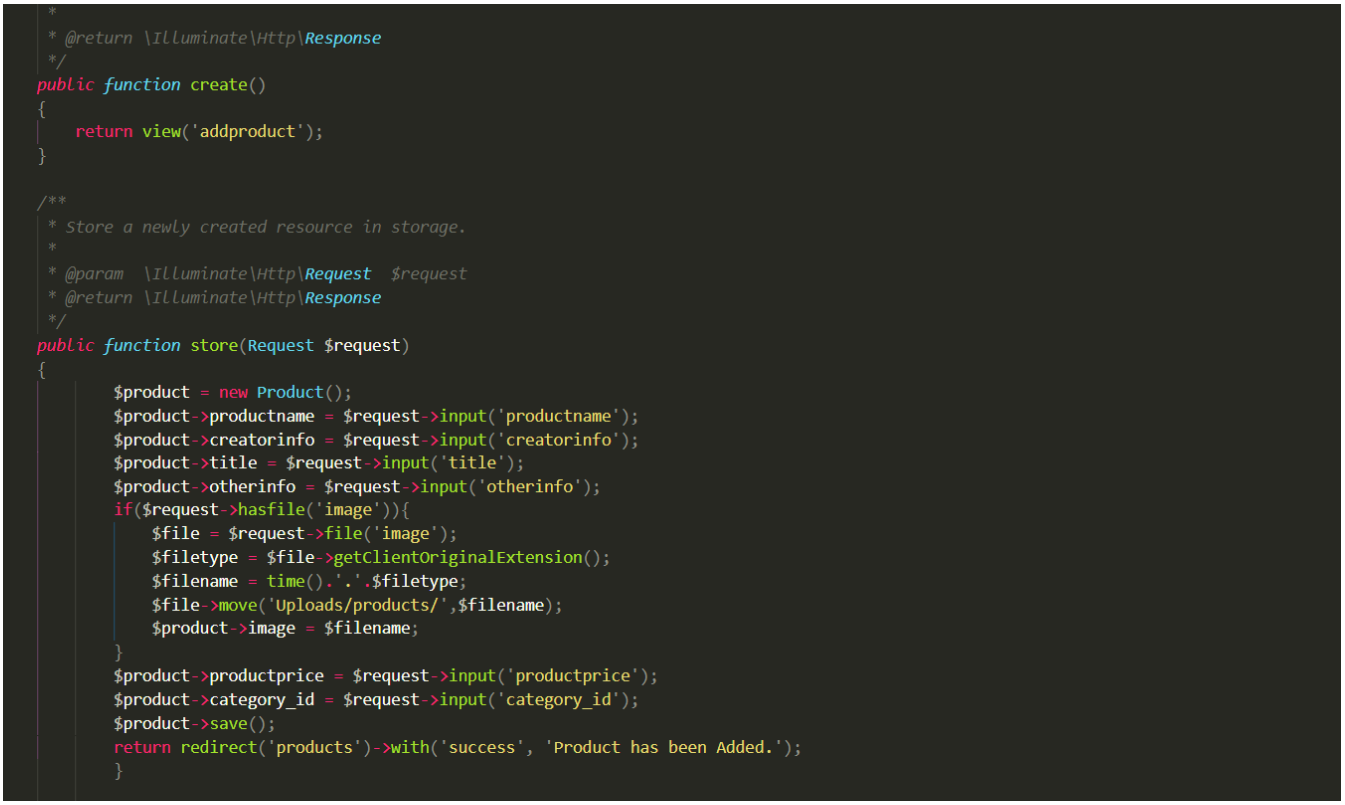


```php
use App\Http\Controllers\ProductController;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Product extends Model
{
    use HasFactory;
    protected $guarded = [];

    public function scopeFilter($query,array $filters){
        $query->when($filters['category'] ?? false,fn($query,$category)=>
            $query
            ->whereHas('category',fn($query)=>
                $query->where('name',$category))
            );

        }

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

Product Model with Category filtering

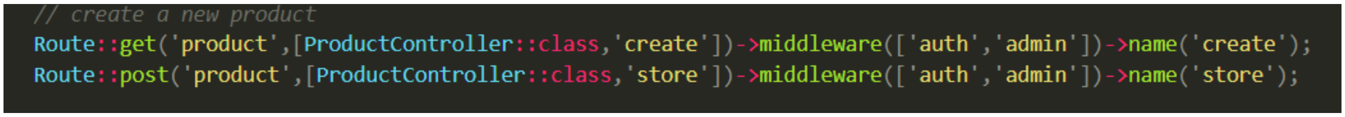
Controllers

Create and store methods

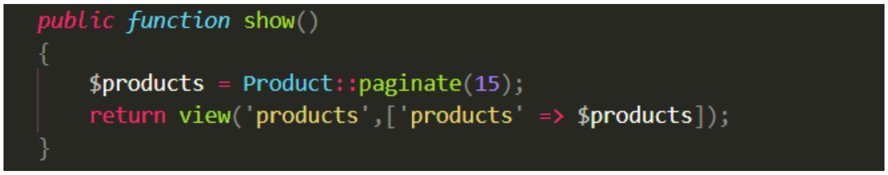


```
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        return view('addproduct');
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
            $product = new Product();
            $product->productname = $request->input('productname');
            $product->creatorinfo = $request->input('creatorinfo');
            $product->title = $request->input('title');
            $product->otherinfo = $request->input('otherinfo');
            if($request->hasfile('image')){
                $file = $request->file('image');
                $filetype = $file->getClientOriginalExtension();
                $filename = time().'.'.$filetype;
                $file->move('Uploads/products/',$filename);
                $product->image = $filename;
            }
            $product->productprice = $request->input('productprice');
            $product->category_id = $request->input('category_id');
            $product->save();
            return redirect('products')->with('success', 'Product has been Added.');
            }
```

This method takes data from add product view file and store the data with image.

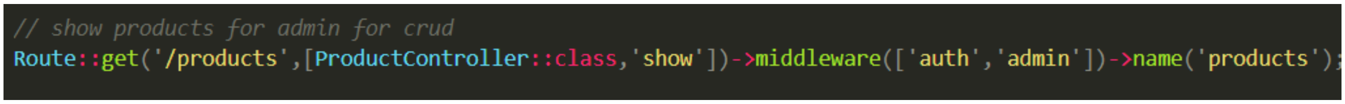


```
// create a new product
Route::get('product',[ProductController::class,'create'])->middleware(['auth','admin'])->name('create');
Route::post('product',[ProductController::class,'store'])->middleware(['auth','admin'])->name('store');
```



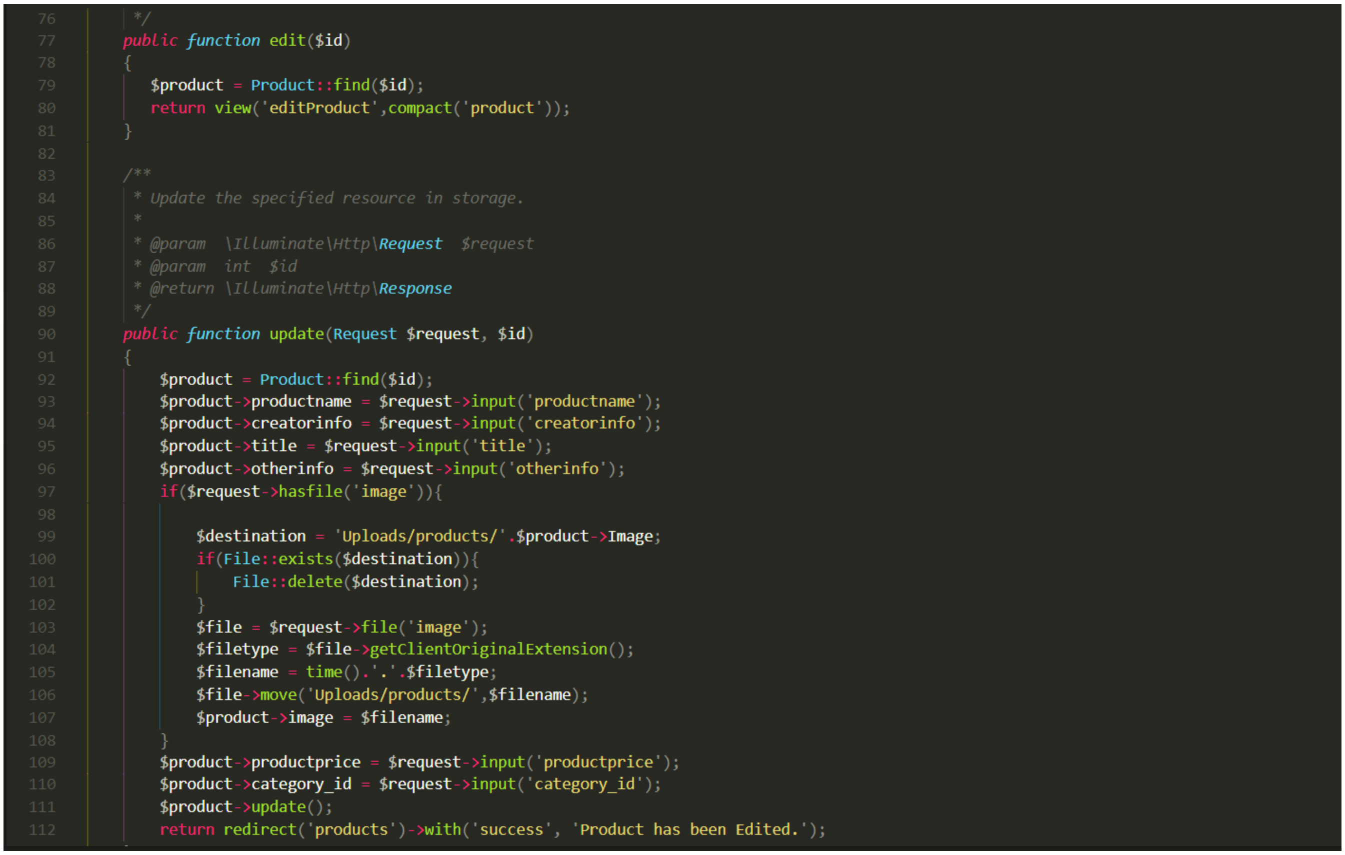
```
public function show()
{
    $products = Product::paginate(15);
    return view('products',['products' => $products]);
}
```

Show Method

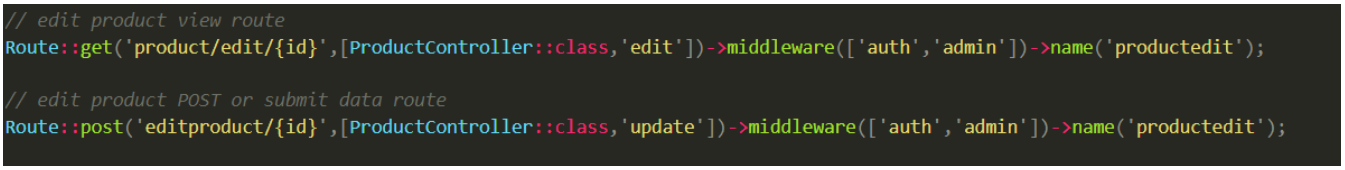


```php
// show products for admin for crud
Route::get('/products',[ProductController::class,'show'])->middleware(['auth','admin'])->name('products');
```

Edit and Update method

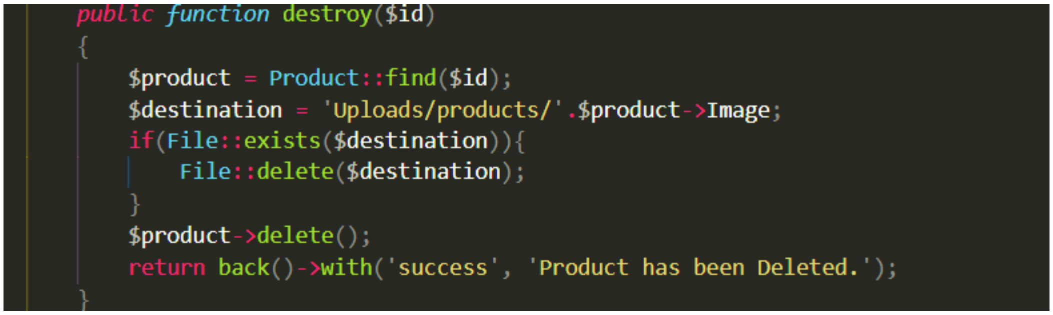


```php
     */
    public function edit($id)
    {
       $product = Product::find($id);
       return view('editProduct',compact('product'));
    }

    /**
     * Update the specified resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request, $id)
    {
        $product = Product::find($id);
        $product->productname = $request->input('productname');
        $product->creatorinfo = $request->input('creatorinfo');
        $product->title = $request->input('title');
        $product->otherinfo = $request->input('otherinfo');
        if($request->hasfile('image')){

            $destination = 'Uploads/products/'.$product->Image;
            if(File::exists($destination)){
                File::delete($destination);
            }
            $file = $request->file('image');
            $filetype = $file->getClientOriginalExtension();
            $filename = time().'.'.$filetype;
            $file->move('Uploads/products/',$filename);
            $product->image = $filename;
        }
        $product->productprice = $request->input('productprice');
        $product->category_id = $request->input('category_id');
        $product->update();
        return redirect('products')->with('success', 'Product has been Edited.');
```

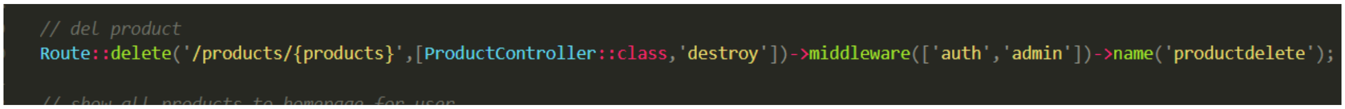


```php
// edit product view route
Route::get('product/edit/{id}',[ProductController::class,'edit'])->middleware(['auth','admin'])->name('productedit');

// edit product POST or submit data route
Route::post('editproduct/{id}',[ProductController::class,'update'])->middleware(['auth','admin'])->name('productedit');
```

Destroy / Delete Method



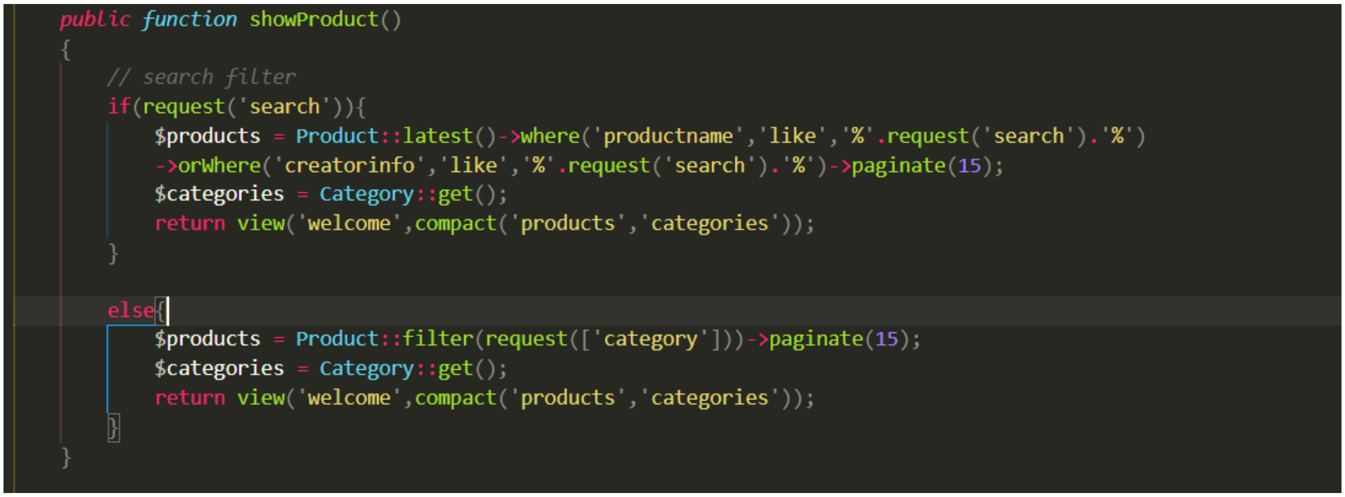
```php
    public function destroy($id)
    {
        $product = Product::find($id);
        $destination = 'Uploads/products/'.$product->Image;
        if(File::exists($destination)){
            File::delete($destination);
        }
        $product->delete();
        return back()->with('success', 'Product has been Deleted.');
    }
```



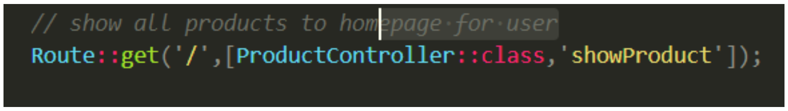
```
// del product
Route::delete('/products/{products}',[ProductController::class,'destroy'])->middleware(['auth','admin'])->name('productdelete');

// show all products to homepage for user
```

Route for deleting by getting id.

Show product method with filtering

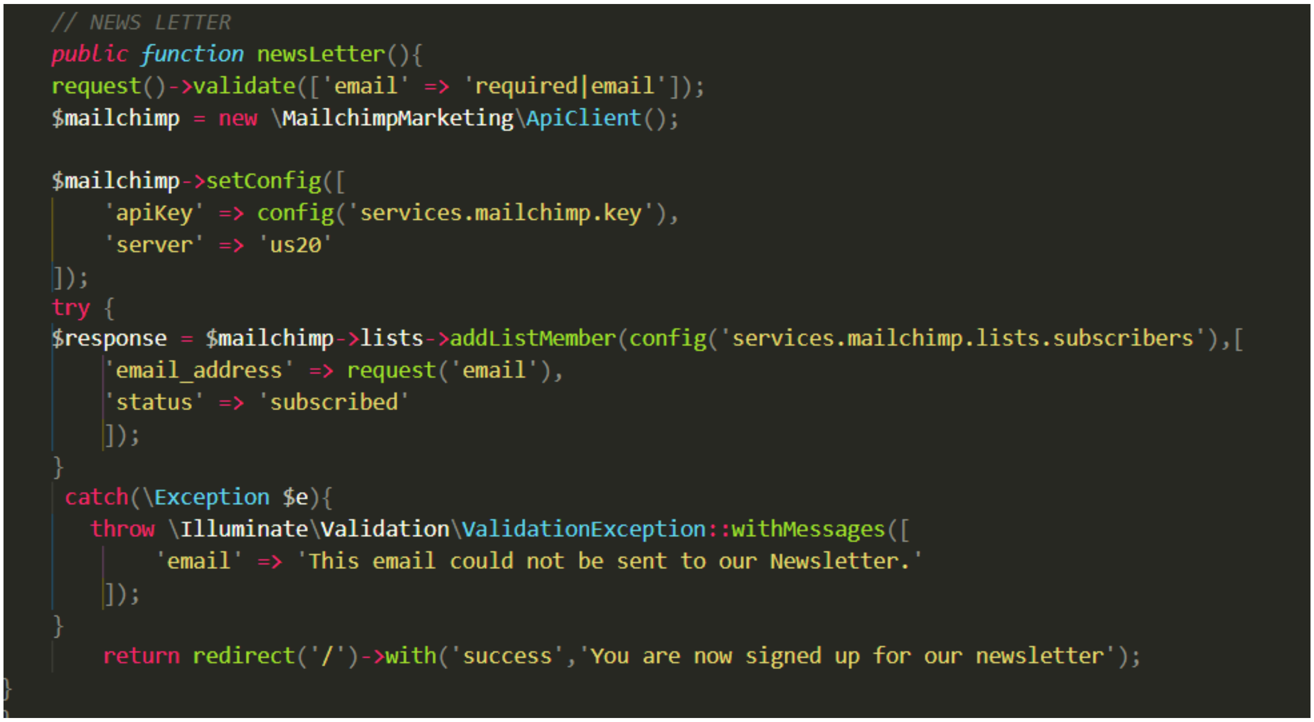


```
public function showProduct()
{
    // search filter
    if(request('search')){
        $products = Product::latest()->where('productname','like','%'.request('search').'%')
        ->orWhere('creatorinfo','like','%'.request('search').'%')->paginate(15);
        $categories = Category::get();
        return view('welcome',compact('products','categories'));
    }

    else{
        $products = Product::filter(request(['category']))->paginate(15);
        $categories = Category::get();
        return view('welcome',compact('products','categories'));
    }
}
```

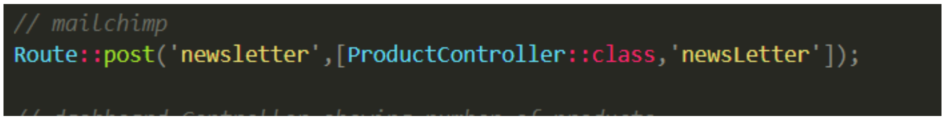


```
// show all products to homepage for user
Route::get('/',[ProductController::class,'showProduct']);
```

Newsletter method for mail chimp API

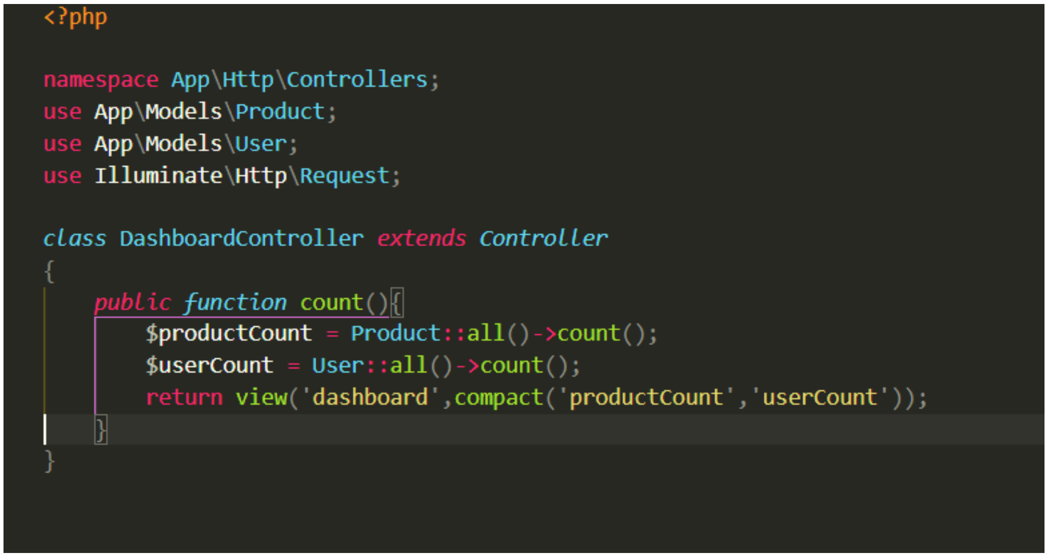


```
    // NEWS LETTER
    public function newsLetter(){
    request()->validate(['email' => 'required|email']);
    $mailchimp = new \MailchimpMarketing\ApiClient();

    $mailchimp->setConfig([
        'apiKey' => config('services.mailchimp.key'),
        'server' => 'us20'
    ]);
    try {
    $response = $mailchimp->lists->addListMember(config('services.mailchimp.lists.subscribers'),[
        'email_address' => request('email'),
        'status' => 'subscribed'
        ]);
    }
     catch(\Exception $e){
       throw \Illuminate\Validation\ValidationException::withMessages([
           'email' => 'This email could not be sent to our Newsletter.'
        ]);
    }
        return redirect('/')->with('success','You are now signed up for our newsletter');
}
```



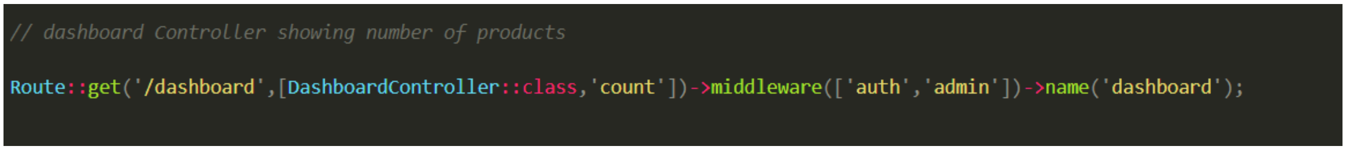
```
// mailchimp
Route::post('newsletter',[ProductController::class,'newsLetter']);
```

Dashboard Controller



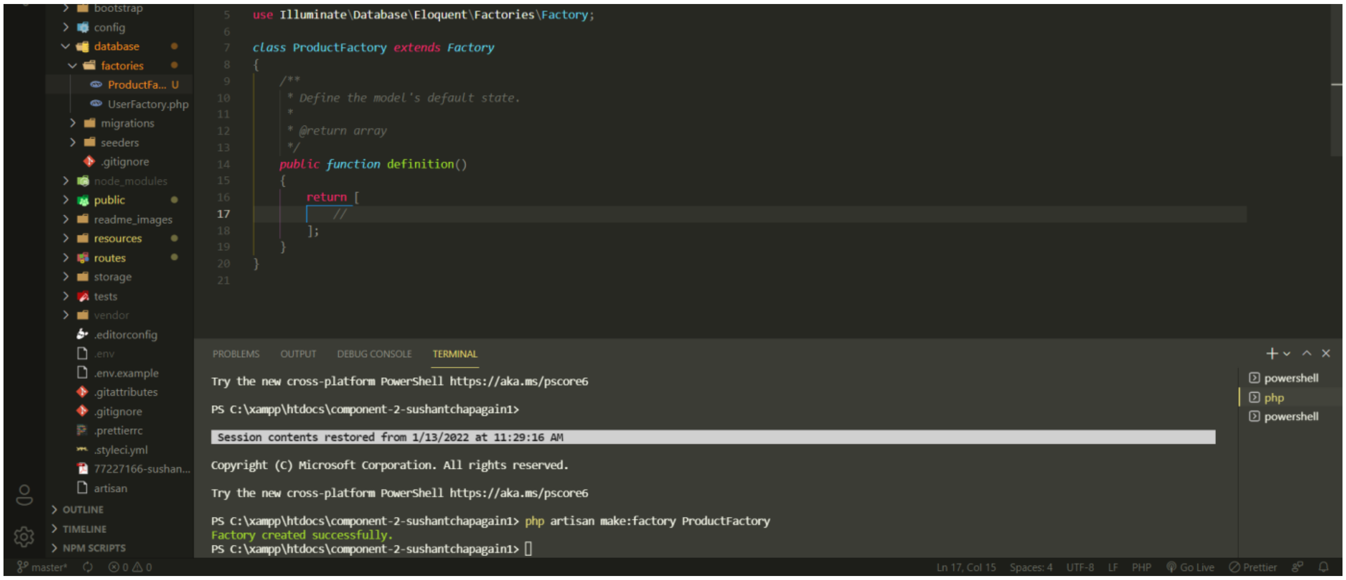
```
<?php

namespace App\Http\Controllers;
use App\Models\Product;
use App\Models\User;
use Illuminate\Http\Request;

class DashboardController extends Controller
{
    public function count(){
        $productCount = Product::all()->count();
        $userCount = User::all()->count();
        return view('dashboard',compact('productCount','userCount'));
    }
}
```

Count total number of products and users form database



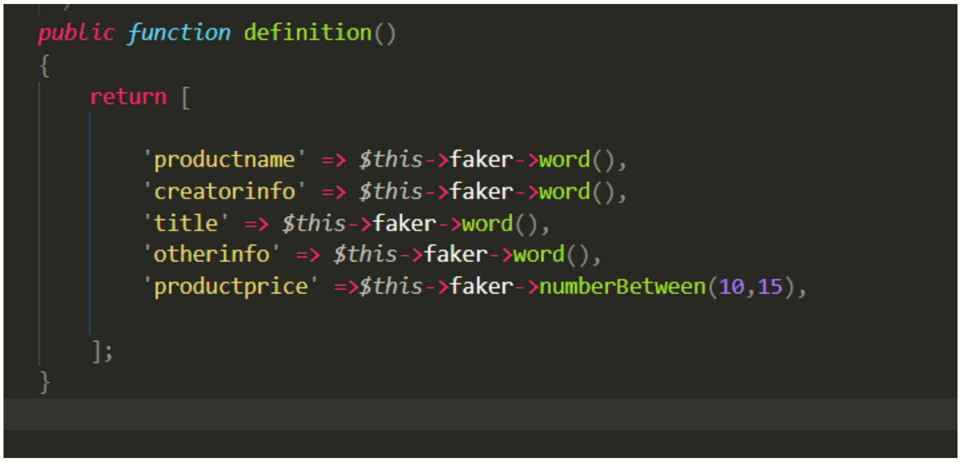
```php
// dashboard Controller showing number of products

Route::get('/dashboard',[DashboardController::class,'count'])->middleware(['auth','admin'])->name('dashboard');
```

Creating Factory



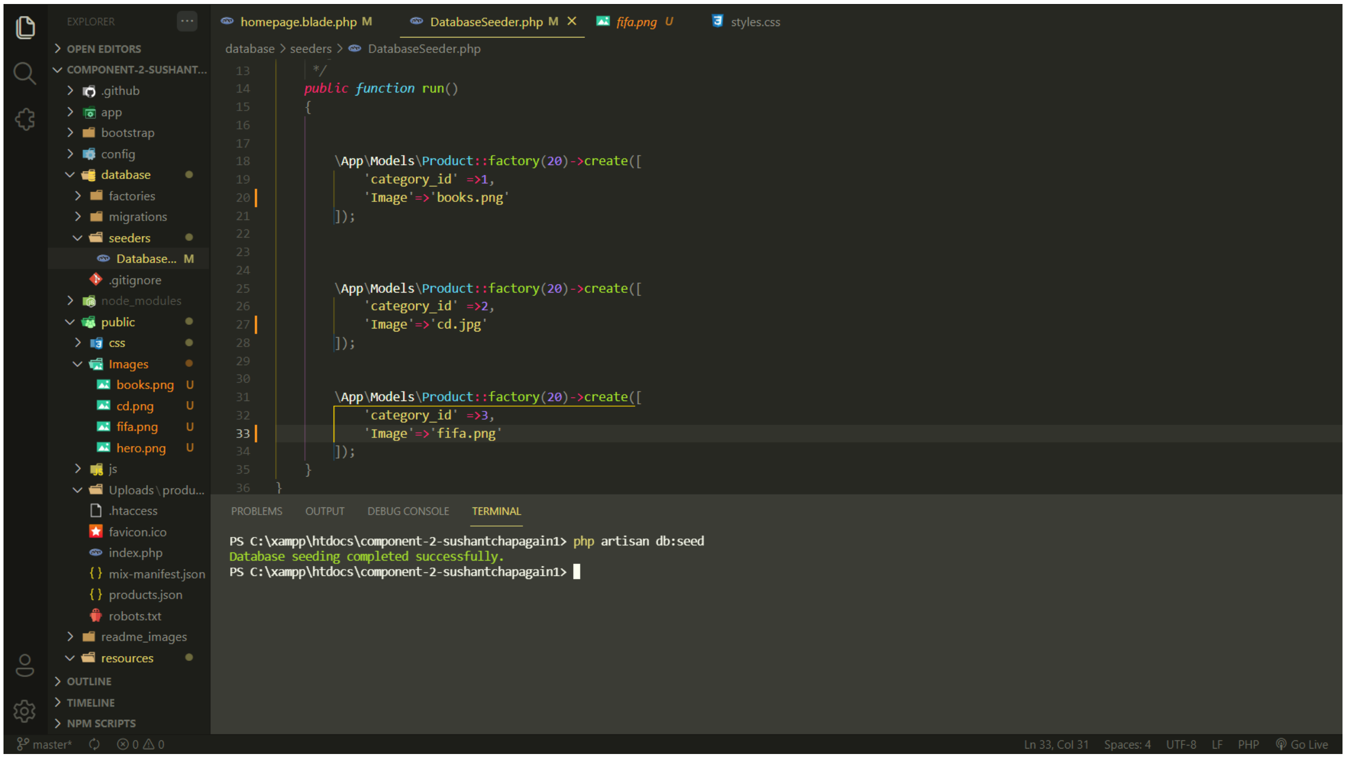
```php
use Illuminate\Database\Eloquent\Factories\Factory;

class ProductFactory extends Factory
{
    /**
     * Define the model's default state.
     *
     * @return array
     */
    public function definition()
    {
        return [
            //
        ];
    }
}
```

```
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\xampp\htdocs\component-2-sushantchapagain1>

 Session contents restored from 1/13/2022 at 11:29:16 AM

Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\xampp\htdocs\component-2-sushantchapagain1> php artisan make:factory ProductFactory
Factory created successfully.
PS C:\xampp\htdocs\component-2-sushantchapagain1> 
```



```php
public function definition()
{
    return [

        'productname' => $this->faker->word(),
        'creatorinfo' => $this->faker->word(),
        'title' => $this->faker->word(),
        'otherinfo' => $this->faker->word(),
        'productprice' =>$this->faker->numberBetween(10,15),

    ];
}
```

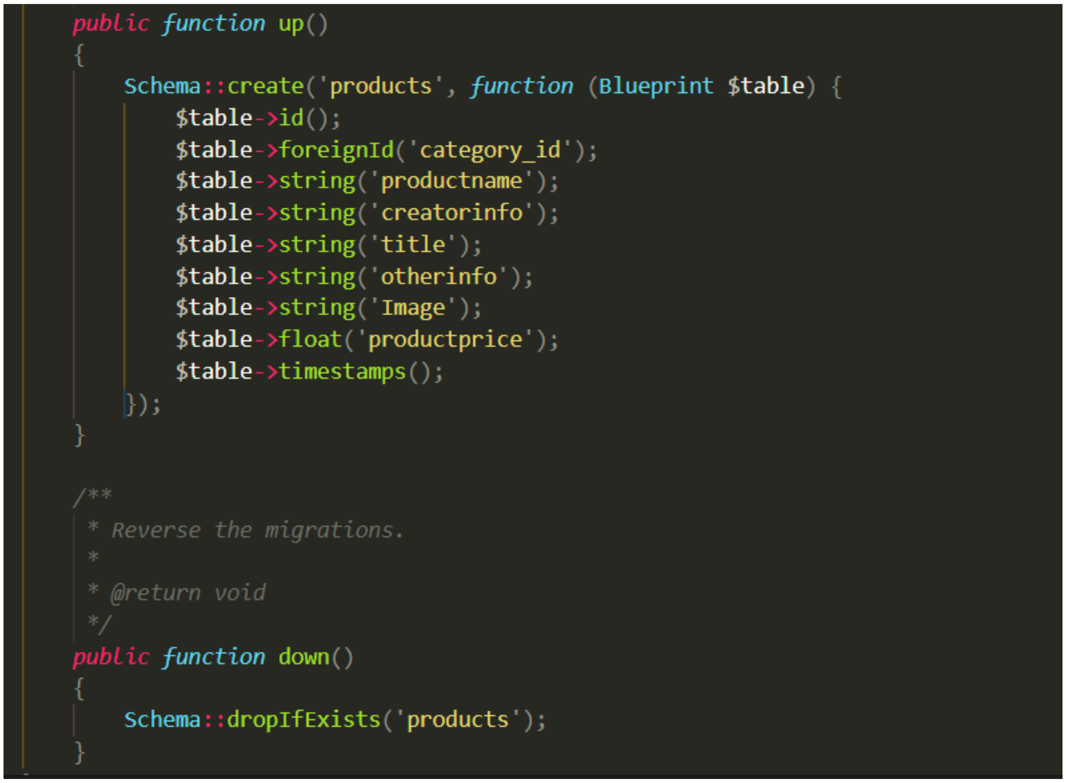
Seeder



```php
     */
    public function run()
    {


        \App\Models\Product::factory(20)->create([
            'category_id' =>1,
            'Image'=>'books.png'
        ]);



        \App\Models\Product::factory(20)->create([
            'category_id' =>2,
            'Image'=>'cd.jpg'
        ]);


        \App\Models\Product::factory(20)->create([
            'category_id' =>3,
            'Image'=>'fifa.png'
        ]);
    }
}
```

```
PS C:\xampp\htdocs\component-2-sushantchapagain1> php artisan db:seed
Database seeding completed successfully.
PS C:\xampp\htdocs\component-2-sushantchapagain1>
```

DATABASE AND MIGARTION FILES



```
public function up()
{
    Schema::create('products', function (Blueprint $table) {
        $table->id();
        $table->foreignId('category_id');
        $table->string('productname');
        $table->string('creatorinfo');
        $table->string('title');
        $table->string('otherinfo');
        $table->string('Image');
        $table->float('productprice');
        $table->timestamps();
    });
}

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

Creates products able, similarly there is migration for category and user table.

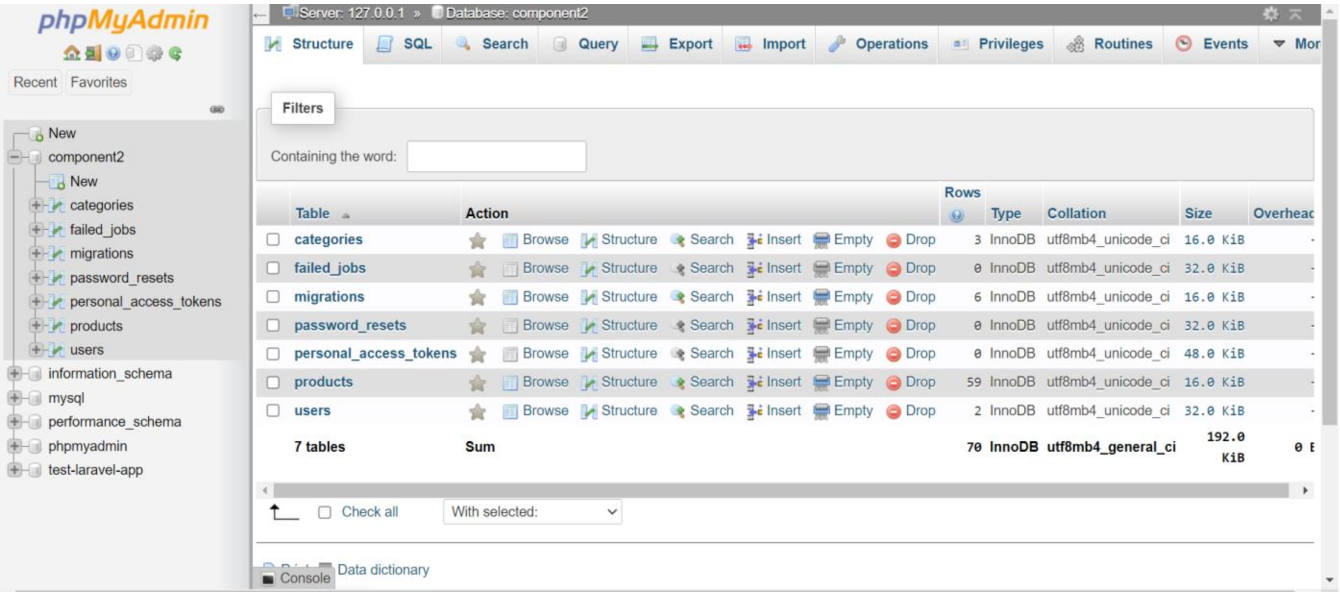
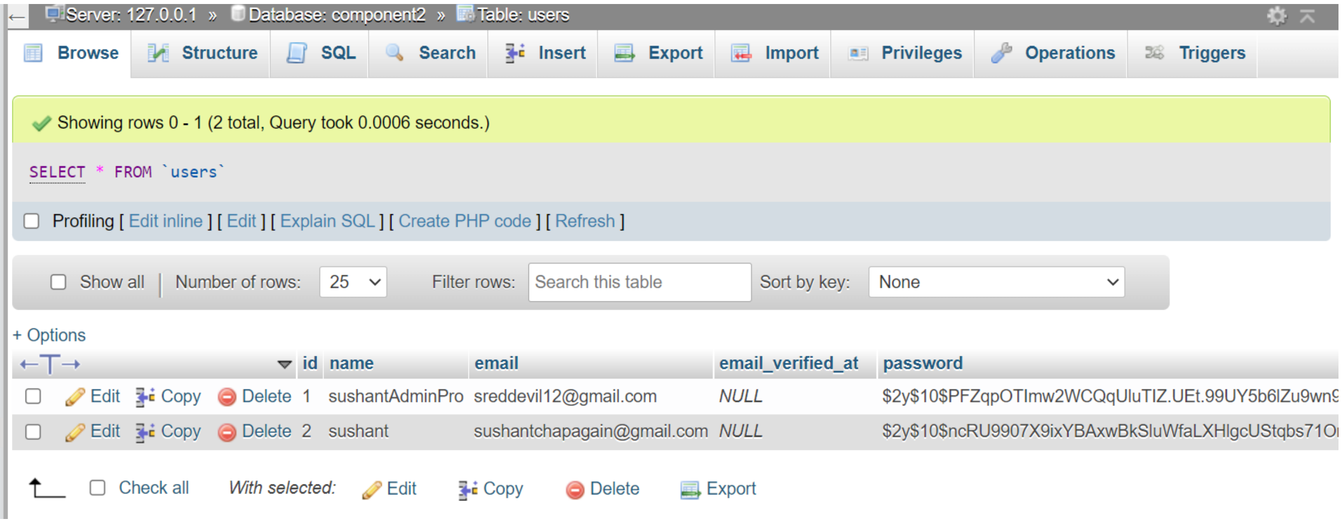


Table	Action							Rows	Type	Collation	Size	Overheac
categories	☆	Browse	Structure	Search	Insert	Empty	Drop	3	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
failed_jobs	☆	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
migrations	☆	Browse	Structure	Search	Insert	Empty	Drop	6	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
password_resets	☆	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
personal_access_tokens	☆	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_unicode_ci	48.0 KiB	-
products	☆	Browse	Structure	Search	Insert	Empty	Drop	59	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
users	☆	Browse	Structure	Search	Insert	Empty	Drop	2	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
7 tables	Sum							70	InnoDB	utf8mb4_general_ci	192.0 KiB	0 E



Server: 127.0.0.1 » Database: component2 » Table: users
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Triggers

✓ Showing rows 0 - 1 (2 total, Query took 0.0006 seconds.)

```
SELECT * FROM `users`
```

□ 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

				id	name	email	email_verified_at	password
□	Edit	Copy	Delete	1	sushantAdminPro	sreddevil12@gmail.com	*NULL*	$2y$10$PFZqpOTImw2WCQqUluTIZ.UEt.99UY5b6lZu9wn
□	Edit	Copy	Delete	2	sushant	sushantchapagain@gmail.com	*NULL*	$2y$10$ncRU9907X9ixYBAxwBkSluWfaLXHlgcUStqbs71O