



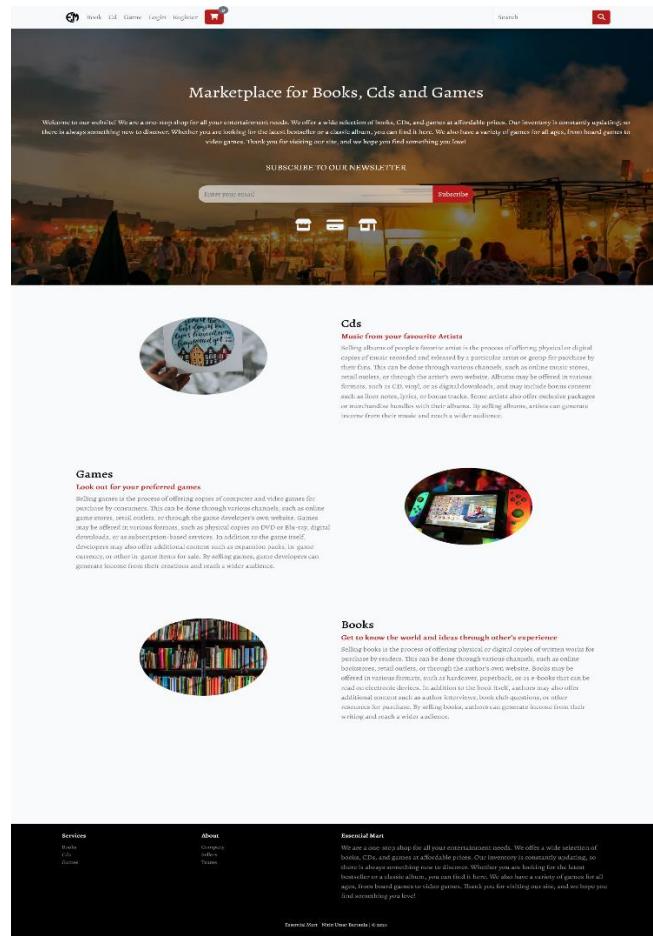
Laravel: Final Web Application

ADVANCED WEB ENGINEERING

Nitin Utsav Bartaula | 77261073 | 19th January 2023

Essential Mart: Homepage

Homepage image link: <https://imgur.com/a/oXdyFoz>



Book Page

The screenshot shows a grid of book covers on a website. At the top left is the Essential Mart logo. The top right features a search bar with a magnifying glass icon. Below the header, there's a section titled "Book Products". The main content area displays eight book covers in two rows of four. Each book cover includes a "SALE" badge, the title, author, and a brief description. Below each cover are its average rating (from 1 to 5 stars) and price.

Book Cover	Title	Author	Description	Avg Rating	Price
	fuga	MIRCEA CARTARESCU	TRANSLATED BY SEAN PITTIER	★★★★★	\$ 28.9
	qui	MIRCEA CARTARESCU	TRANSLATED BY SEAN PITTIER	★★★★★	\$ 27.3
	ut	MIRCEA CARTARESCU	A HISTORY OF NINETEENTH CENTURY MUSIC	★★★★★	\$ 28.2
	illo	MIRCEA CARTARESCU	A HISTORY OF NINETEENTH CENTURY MUSIC	★★★★★	\$ 24.5
	THE BELOVED VISION	MIRCEA CARTARESCU	A HISTORY OF NINETEENTH CENTURY MUSIC	★★★★★	\$ 25.5
	AN IMMENSE WORLD	ED YONG	PULITZER PRIZE-winning author of <i>I Contain Multitudes</i>	a	\$ 26.4
	HAVEN	EAVAN BOLAND	S#1 NEW YORK TIMES BESTSELLING AUTHOR OF <i>ROOM</i>	ab	\$ 23.5
	THE BELOVED VISION	MIRCEA CARTARESCU	A HISTORY OF NINETEENTH CENTURY MUSIC	et	\$ 26.3

Below the book grid is a navigation bar with page numbers: < 1 2 3 4 5 6 7 >. The background of the page is white.

The footer is divided into three columns. The first column, "Services", lists "Books", "Cd's", and "Games". The second column, "About", lists "Company", "Sellers", and "Teams". The third column, "Essential Mart", contains a paragraph about the store's mission to provide a one-stop shop for entertainment needs. At the bottom of the footer, there's a small copyright notice: "Essential Mart | Nitin Utsav Barraula | © 2023".

- Book Factory & Seeder class

```

/*
 * @extends \Illuminate\Database\Eloquent\Factories\Factory<\App\Models\Book>
 */
class BookFactory extends Factory
{
    /**
     * Define the model's default state.
     *
     * @return array<string, mixed>
     */
    public function definition()
    {
        $gender = $this->faker->randomElement(['male', 'female']);
        return [
            'title' => $this->faker->unique()->word,
            'name' => $this->faker->firstName($gender),
            'author' => $this->faker->name($gender),
            'description' => $this->faker->paragraph(),
            'price' => $this->faker->randomFloat(1, 20, 30, 40),
            'image' => $this->faker->randomElement(['book1.jpg', 'book2.jpg', 'book3.jpg', 'book4.jpg', 'book5.jpg', 'book6.jpg']),
            'pages' => $this->faker->numberBetween($min = 100, $max = 500),
        ];
    }
}

class BookSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        Book::factory()->count(50)->create();
    }
}

```

- Book Migration command: `php artisan make:migration create_books_table`

```
database > migrations > 2023_01_07_102613_create_books_table.php
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
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('books', function (Blueprint $table) {
17             $table->id()->autoincrement();
18             $table->string('title');
19             $table->string('name')->nullable();
20             $table->string('author');
21             $table->float('price');
22             $table->text('description');
23             $table->float('pages');
24             $table->string('image');
25             $table->string('category')->default('book');
26             $table->timestamps();
27         });
28     }
}
```

Cd Page

The screenshot shows a website interface for 'Cd Products'. At the top, there is a navigation bar with links for Book, Cd, Game, Login, Register, and a search bar with a magnifying glass icon. Below the navigation, the page title 'Cd Products' is displayed. The main content area features a grid of CD covers. The first row contains four items: 'libero' (rating 4.5 stars, price \$22.7), 'omnis' (rating 4.5 stars, price \$21), 'delectus' (rating 4.5 stars, price \$22.5), and 'eligendi' (rating 4.5 stars, price \$28.8). The second row contains four items: 'IMAGINE DRAGONS ORIGINS' (rating 4.5 stars, price \$20.9), 'IMAGINE DRAGONS EVOLVE' (rating 4.5 stars, price \$20.6), 'LEMONADE' (rating 4.5 stars, price \$23.6), and 'eum' (rating 4.5 stars, price \$23.9). Below the grid is a navigation bar with page numbers 1 through 7.

Product	Rating	Price
libero	★★★★★	\$22.7
omnis	★★★★★	\$21
delectus	★★★★★	\$22.5
eligendi	★★★★★	\$28.8
IMAGINE DRAGONS ORIGINS	★★★★★	\$20.9
IMAGINE DRAGONS EVOLVE	★★★★★	\$20.6
LEMONADE	★★★★★	\$23.6
eum	★★★★★	\$23.9

The footer section of the website includes several links: 'Services' (Books, Cds, Games), 'About' (Company, Sellers, Teams), and 'Essential Mart'. The 'Essential Mart' section contains a paragraph about the store's mission to provide a one-stop shop for entertainment needs at affordable prices. At the bottom, there is a copyright notice: 'Essential Mart | Nitin Utsav Barauta | © 2023'.

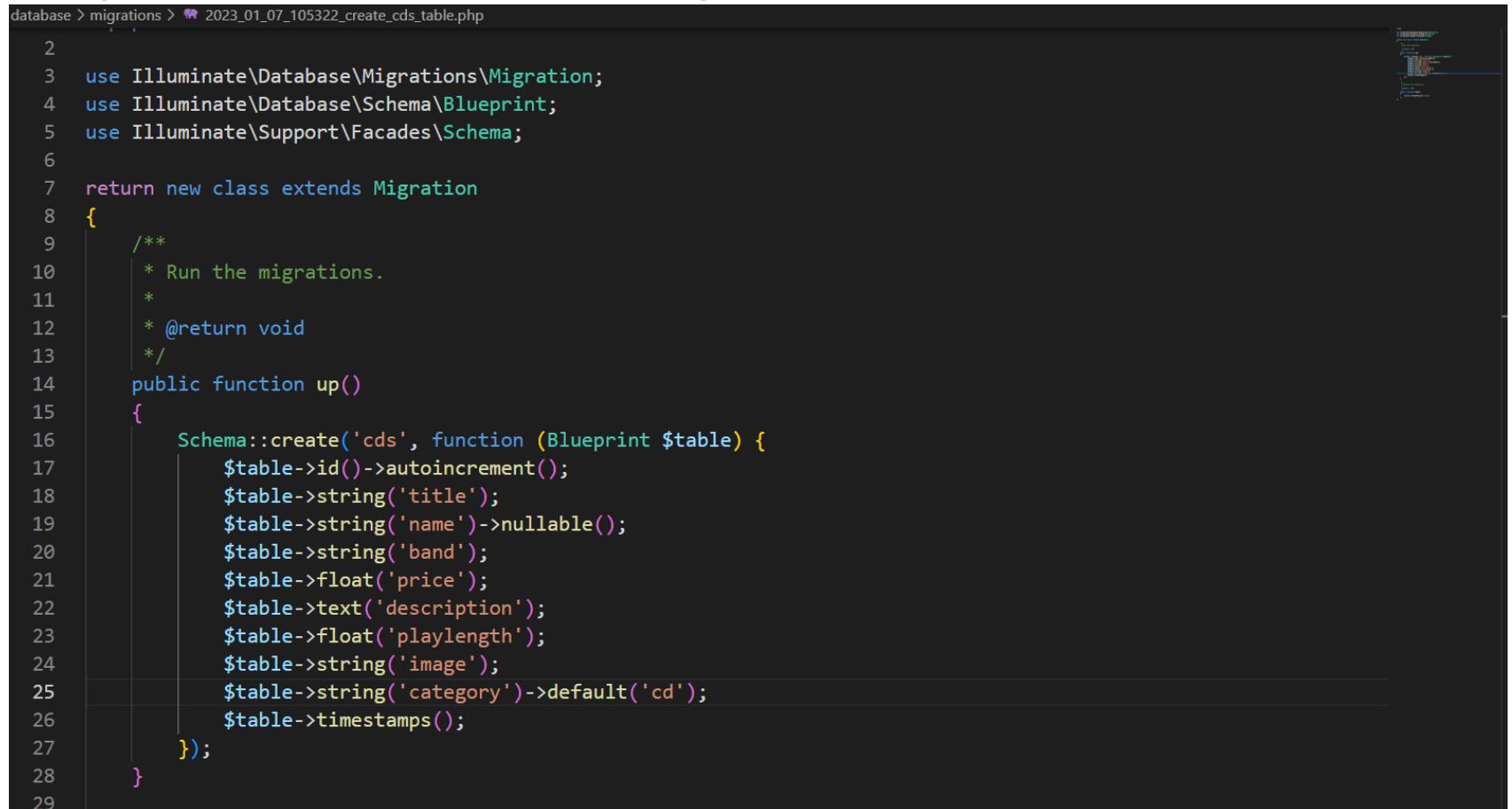
Services
Books
Cds
Games

About
Company
Sellers
Teams

Essential Mart
We are a one-stop shop for all your entertainment needs. We offer a wide selection of books, CDs, and games at affordable prices. Our inventory is constantly updating, so there is always something new to discover. Whether you are looking for the latest bestseller or a classic album, you can find it here. We also have a variety of games for all ages, from board games to video games. Thank you for visiting our site, and we hope you find something you love!

Essential Mart | Nitin Utsav Barauta | © 2023

- Cd Migration command: `php artisan make:migration create_cds_table`



The screenshot shows a code editor window with a dark theme. The file path at the top is "database > migrations > 2023_01_07_105322_create_cds_table.php". The code is a PHP class definition for a database migration:

```

1
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
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('cds', function (Blueprint $table) {
17             $table->id()->autoincrement();
18             $table->string('title');
19             $table->string('name')->nullable();
20             $table->string('band');
21             $table->float('price');
22             $table->text('description');
23             $table->float('playlength');
24             $table->string('image');
25             $table->string('category')->default('cd');
26             $table->timestamps();
27         });
28     }
29 }
```

- Cd Factory & Seeder class

```
/**
 * @extends \Illuminate\Database\Eloquent\Factories\Factory<\App\Models\cd>
 */
class CdFactory extends Factory
{
    /**
     * Define the model's default state.
     *
     * @return array<string, mixed>
     */
    public function definition()
    {
        $gender = $this->faker->randomElement(['male', 'female']);
        return [
            'title' => $this->faker->unique()->word,
            'name' => $this->faker->name($gender),
            'band' => $this->faker->randomElement(["The Beatles", "Led Zeppelin", "The Rolling Stones", "Pink Floyd", "Queen", "U2", "Metallica", "AC/DC", "The Eagles", "Nirvana"]),
            'description' =>$this->faker->paragraph(),
            'price'=>$this->faker->randomFloat(1,20,30,40),
            'image' => $this->faker->randomElement(['cd1.jpg', 'cd2.jpg', 'cd3.jpg', 'cd4.jpg', 'cd5.jpg','cd6.jpg']),
            'playlength' => $this->faker->numberBetween($min = 2, $max = 59),
        ];
    }
}

class CdSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        Cd::factory()->count(50)->create();
    }
}
```

Game Page

The screenshot shows a website interface for a game store. At the top, there is a navigation bar with links for Book, Cd, Game, Login, and Register, along with a search bar and a user icon. Below the navigation bar, the page title "Game Products" is displayed. The main content area features a grid of game covers, each with a price and a rating. The games shown include God of War, Red Dead Redemption II, Apex Legends, Watch Dogs 2, and several other titles. A navigation bar at the bottom allows users to browse through more pages.

Game Title	Rating	Price
GOD OF WAR	★★★★★	\$ 27.4
RED DEAD REDEMPTION II	★★★★★	\$ 25.1
APEX LEGENDS	★★★★★	\$ 25.9
GOD OF WAR	★★★★★	\$ 26.4
RED DEAD REDEMPTION II	★★★★★	\$ 20.5
THE WITCHER 3: WILD HUNT	★★★★★	\$ 23.4
APEX LEGENDS	★★★★★	\$ 29.1
WATCH DOGS 2	★★★★★	\$ 25.6

The screenshot shows a dark-themed footer section with various links and a message. On the left, there are links for Services (Books, CDs, Games), About (Company, Seller, Team), and Essential Mart. The central part of the footer contains a message about the store's offerings. At the bottom, there is a copyright notice: "Essential Mart | Nitin Ussav Barautala | © 2023".

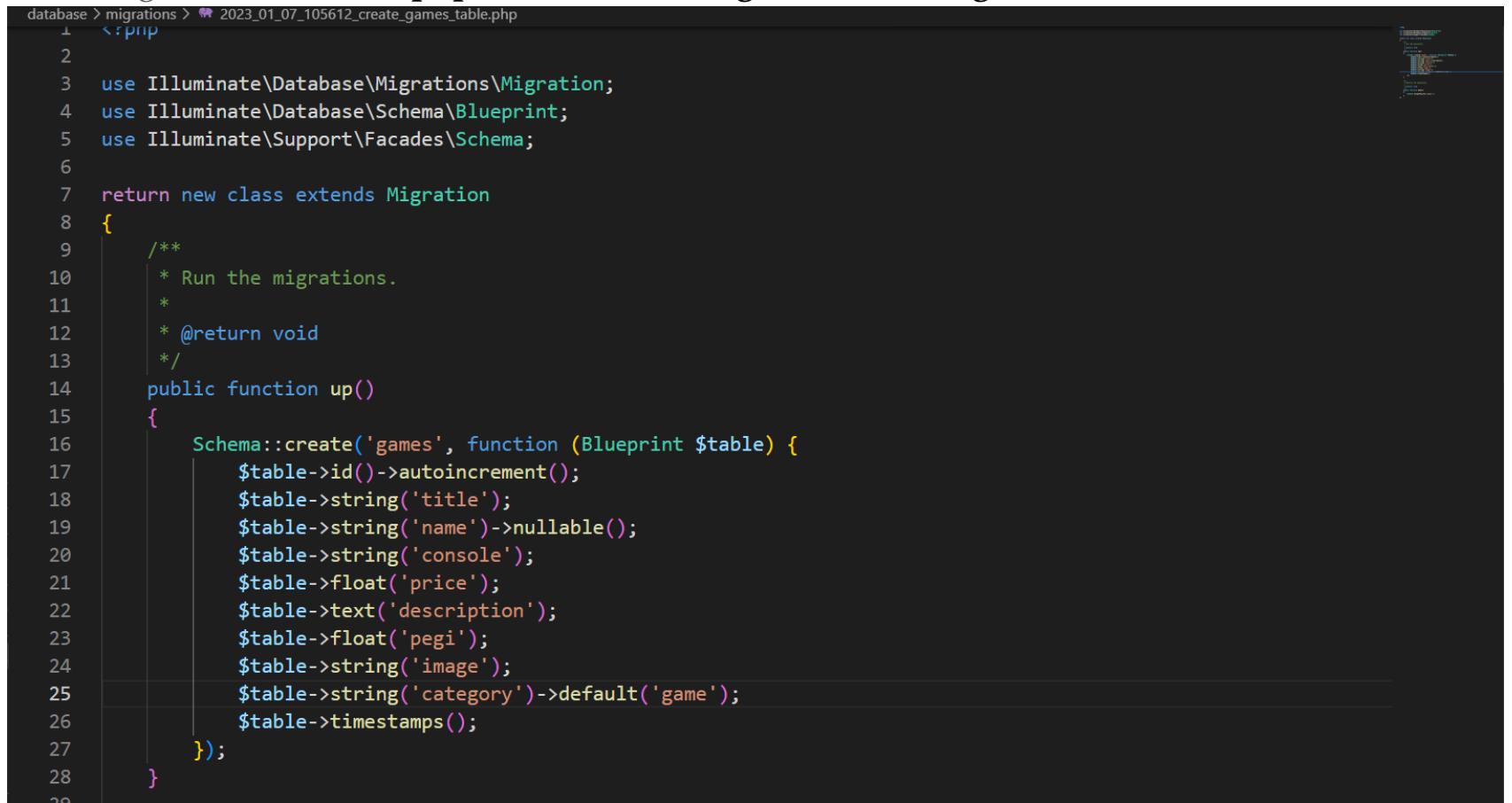
Services
Books
CDs
Games

About
Company
Seller
Team

Essential Mart
We are a one-stop shop for all your entertainment needs. We offer a wide selection of books, CDs, and games at affordable prices. Our inventory is constantly updating, so there is always something new to discover. Whether you are looking for the latest bestseller or a classic album, you can find it here. We also have a variety of games for all ages, from board games to video games. Thank you for visiting our site, and we hope you find something you love!

Essential Mart | Nitin Ussav Barautala | © 2023

- Game Migration command: **php artisan make:migration create_games_table**



```

database > migrations > 2023_01_07_105612_create_games_table.php
  ↗ 2023_01_07_105612_create_games_table.php
1  <:PHP
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
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('games', function (Blueprint $table) {
17              $table->id()->autoincrement();
18              $table->string('title');
19              $table->string('name')->nullable();
20              $table->string('console');
21              $table->float('price');
22              $table->text('description');
23              $table->float('pegi');
24              $table->string('image');
25              $table->string('category')->default('game');
26              $table->timestamps();
27          });
28      }
29  }

```

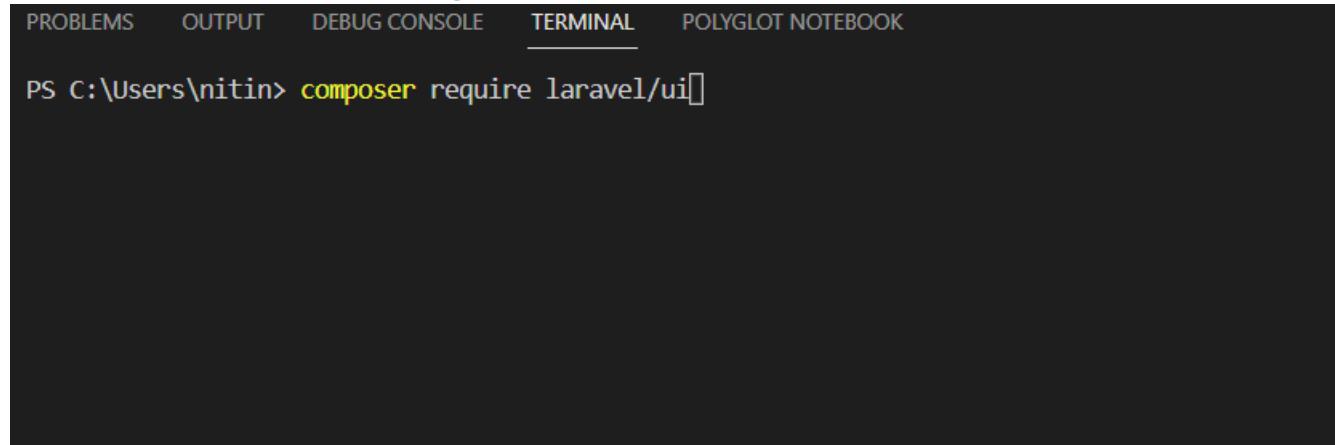
- Game Factory & Seeder class

```
/*
 * @extends \Illuminate\Database\Eloquent\Factories\Factory<\App\Models\Game>
 */
class GameFactory extends Factory
{
    /**
     * Define the model's default state.
     *
     * @return array<string, mixed>
     */
    public function definition()
    {
        $gender = $this->faker->randomElement(['male', 'female']);
        return [
            'title' => $this->faker->unique()->word,
            'name' => $this->faker->name($gender),
            'console' => $this->faker->randomElement(["PlayStation 5", "Xbox Series X|S", "Nintendo Switch",
                "PlayStation 4", "Xbox One", "Nintendo 3DS", "PlayStation Vita", "Wii U", "PlayStation 3", "Xbox
                360"]),
            'description'=>$this->faker->paragraph(),
            'price'=>$this->faker->randomFloat(1,20,30,40),
            'image' => $this->faker->randomElement(['game1.jpg', 'game2.jpg', 'game3.jpg', 'game4.jpg', 'game5.
                jpg', 'game6.jpg']),
            'pegi' => $this->faker->randomElement(['3','7','12','16','18']),
        ];
    }
}

class GameSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        Game::factory()->count(50)->create();
    }
}
```

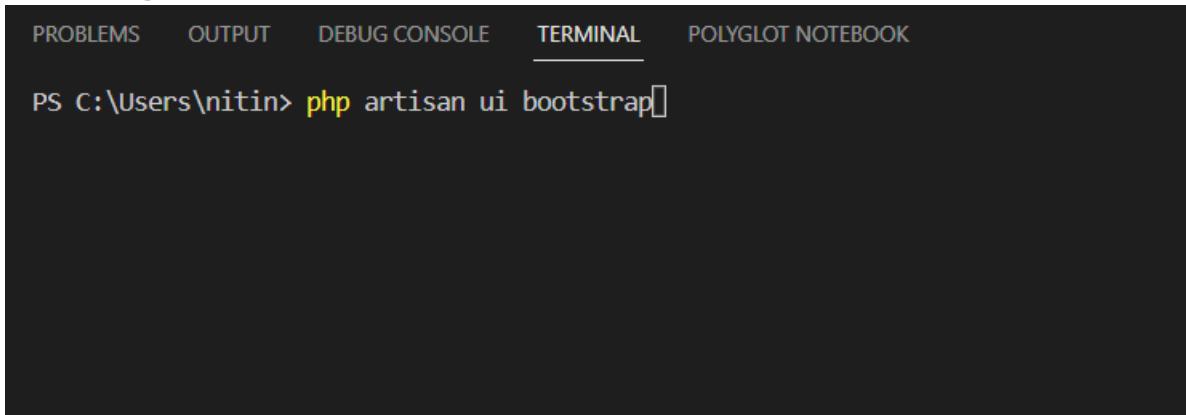
For Login and Register using Laravel UI/ auth Scaffolding package

- Laravel UI composer package installation



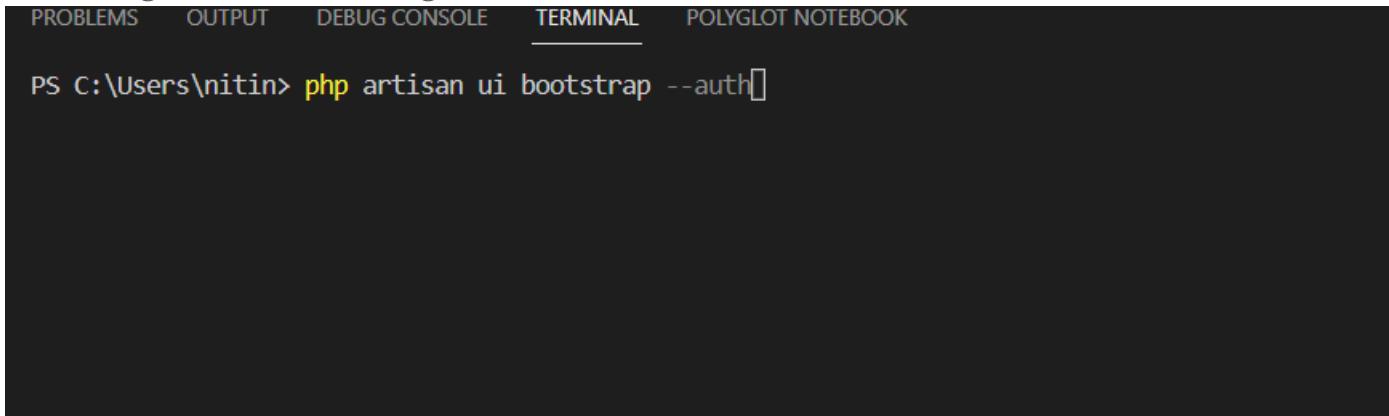
A screenshot of a terminal window with a dark background. At the top, there are tabs labeled 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'POLYGLOT NOTEBOOK'. The 'TERMINAL' tab is underlined. Below the tabs, the terminal prompt 'PS C:\Users\nitin>' is visible, followed by the command 'composer require laravel/ui'. The command is partially typed, with 'composer' in yellow and 'require laravel/ui' in white.

- Installing bootstrap



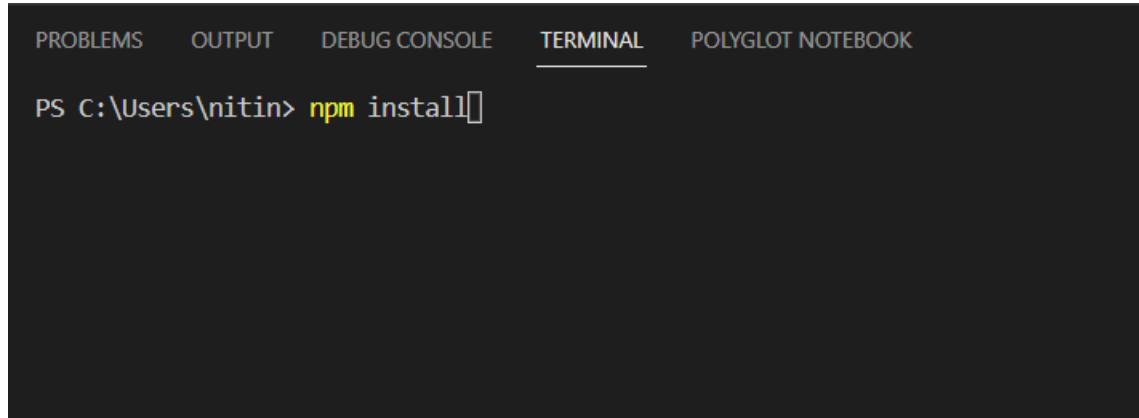
A screenshot of a terminal window with a dark background. At the top, there are tabs labeled PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and POLYGLOT NOTEBOOK. Below the tabs, the terminal prompt is "PS C:\Users\nitin>". The user has typed the command "php artisan ui bootstrap" into the terminal. The text is white, and the command is partially visible as "php artisan ui bootstrap[]".

- Installing Auth Scaffolding



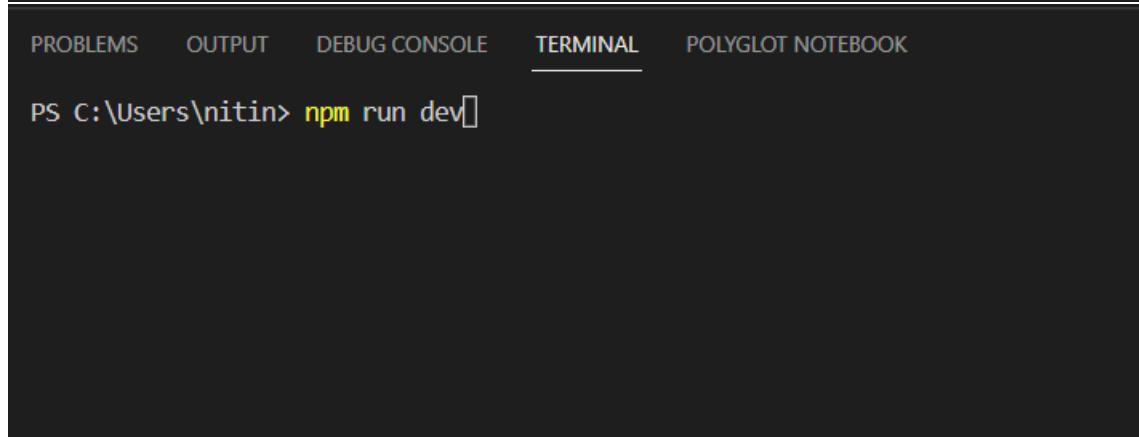
A screenshot of a terminal window with a dark background. At the top, there are tabs labeled PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and POLYGLOT NOTEBOOK. Below the tabs, the terminal prompt is "PS C:\Users\nitin>". The user has typed the command "php artisan ui bootstrap --auth" into the terminal. The text is white, and the command is partially visible as "php artisan ui bootstrap --auth[]".

- Npm install



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL POLYGLOT NOTEBOOK

PS C:\Users\nitin> **npm install**



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL POLYGLOT NOTEBOOK

PS C:\Users\nitin> **npm run dev**

Login and Register custom page



The screenshot shows a custom login page for a website named "Essential Mart". At the top left is a logo consisting of a stylized "EM" monogram. To its right are navigation links: Book, Cd, Game, Login, Register, and a shopping cart icon with a small circular badge showing the number "0". On the far right is a search bar with a magnifying glass icon.

Login

Welcome to Essential Mart! Please enter your credentials.

Email address

Password

Remember Me

[Forgot Your Password?](#)

LOGIN



Book Cd Game Login Register



Search



Register

New to Essential Mart? Register Here.

Your Name

Your Email Address

Your Password

Confirm Password

REGISTER

- User's custom dashboard page

The screenshot shows a user's custom dashboard page. At the top, there is a navigation bar with a logo (EM), menu items (Book, Cd, Game, BookAdmin), a red shopping cart icon with a notification dot, and a search bar. Below the navigation bar, the title "Your Dashboard" is displayed. On the left, there is a dark sidebar featuring a user profile picture, the text "Hi! BookAdmin", and "Welcome to Essential Mart". To the right of the sidebar, there is a card titled "General Information" containing fields for Email (book@gmail.com) and Role (bookAdmin). Below this, there is a section titled "Projects" and another titled "Created at" with the date 2023-01-19 06:27:58.

Book Admin Crud Functionality

- Add Book option at the top right corner of the body section – Only visible to Book Admin

The screenshot shows a web-based application interface for managing books. At the top, there is a navigation bar with icons for Book, Cd, Game, BookAdmin, a shopping cart with a notification badge, and a search bar with a magnifying glass icon. Below the navigation bar, the page title is "Book Products". On the right side of the title, there is a red button labeled "Add Book". The main content area displays a grid of book covers. The first two books in the grid are identical, showing the cover of "SILENT END" by Mircea Cartarescu, Translated by Sean Cotter. Both covers have a "SALE" badge. The next two books in the grid are identical, showing the cover of "THE BELOVED VISION: A HISTORY OF NINETEENTH CENTURY MUSIC" by ut, Translated by illo. These also have "SALE" badges. Below each book cover, there is a title, author/translator, rating (5 stars), and price. The titles and authors for the first two books are "fuga" and "qui" respectively. The titles and authors for the last two books are "ut" and "illo" respectively. The prices are \$28.9, \$27.3, \$28.2, and \$24.5. At the bottom of the page, there are partial views of four more book covers: "THE HIDDEN REALMS AROUND US" by AN, "LIVIVIA DONOGHUE", and "THE".

Title	Author	Rating	Price
SILENT END	MIRCEA CARTARESCU TRANSLATED BY SEAN COTTER	★★★★★	\$ 28.9
SILENT END	MIRCEA CARTARESCU TRANSLATED BY SEAN COTTER	★★★★★	\$ 27.3
THE BELOVED VISION	ut A HISTORY OF NINETEENTH CENTURY MUSIC	★★★★★	\$ 28.2
THE BELOVED VISION	illo A HISTORY OF NINETEENTH CENTURY MUSIC	★★★★★	\$ 24.5
THE HIDDEN REALMS AROUND US	AN	★★★★★	\$ 27.3
LIVIVIA DONOGHUE	LIVIVIA DONOGHUE	★★★★★	\$ 28.2
THE		★★★★★	\$ 24.5

- Book Admin's 'Add Book' form

The screenshot shows a web application interface for adding books. At the top, there is a navigation bar with icons for Book, Cd, Game, and BookAdmin, along with a search bar and a red search button. Below the navigation bar, the title "Add Books" is centered above a form. The form consists of several input fields: "Title", "Name (optional)", "Author", "Price", "Description", and "Pages". There is also a file upload field labeled "Browse..." with the message "No file selected.". A large red "Submit" button is located at the bottom of the form.

Add Books

Title

Name (optional)

Author

Price

Description

Pages

Browse... No file selected.

Submit

- Book Admin's 'Edit Page' button and Form

The screenshot shows a web-based book administration system. At the top, there is a navigation bar with a logo, a search bar, and a red 'Search' button. Below the navigation is a header with a user icon and the text 'BookAdmin'. A shopping cart icon with a '0' notification is also present. The main content area displays a book cover for 'SOL' by Mircea Cartarescu, translated by Sean Cotter. To the right of the book image, there is a 'Back' button, a trash can icon, and a 'BOOK FUGA \$ 28.9' section. Below this, a placeholder text block reads: 'Commodi dolor veritatis eum minima. Cumque architecto ut iste. Quasi nihil ut et iusto quod.' Underneath, it says 'TOTAL PAGES: 237' and features a red 'ADD TO CART' button.



Book Cd Game BookAdmin ▾



Search



Edit Books



fuga

Alexis. ab

Damien Schumm V

28.9

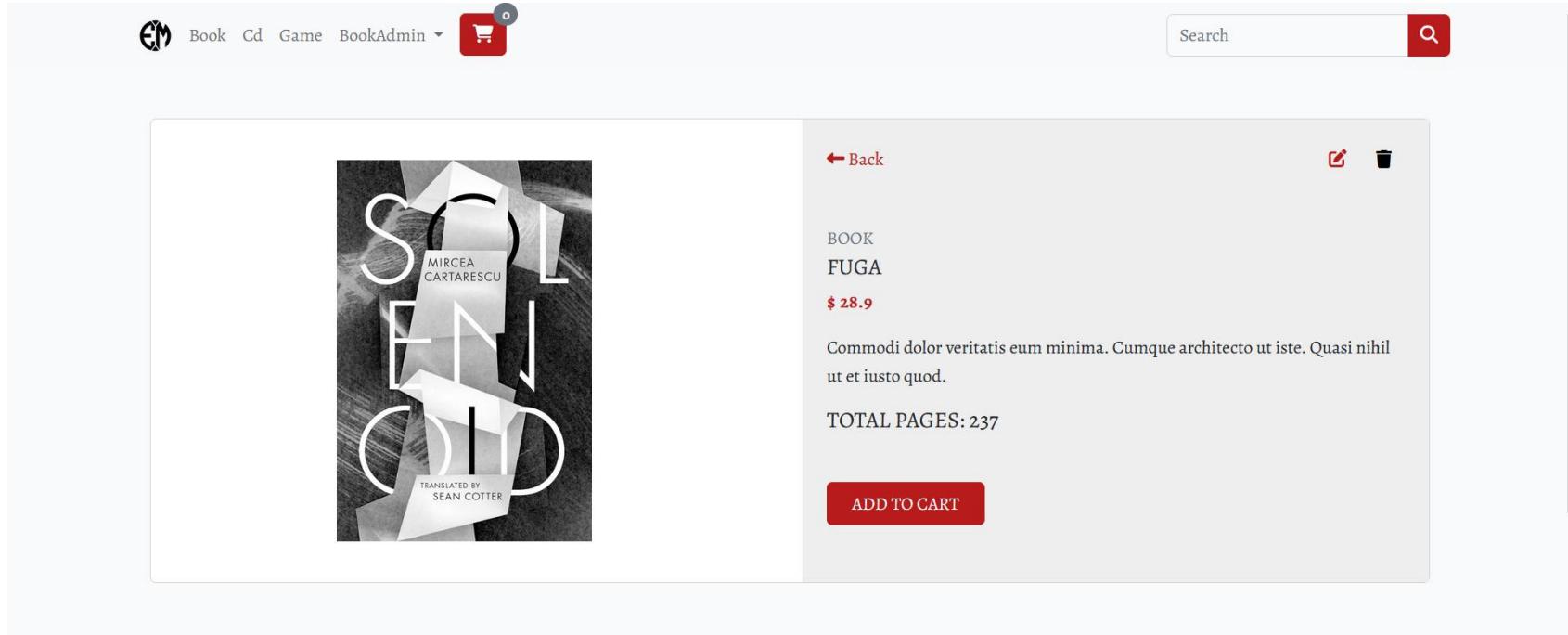
Commodi dolor veritatis eum minima. Cumque architecto
ut iste. Quasi nihil ut et iusto quod.

237

Browse... No file selected.

Update

- Book Admin's 'Delete' button and functionality



The screenshot shows a web-based book administration system. At the top, there is a navigation bar with a logo, a search bar, and a red 'BookAdmin' dropdown menu. A red shopping cart icon with a notification dot is also present. The main content area displays a book cover for 'SOL EN FUGA' by Mircea Cartarescu, translated by Sean Cotter. To the right of the book image, there is a 'Back' button, a 'Delete' button (with a trash icon), and a 'Edit' button. Below the book image, the title 'BOOK FUGA' is listed, followed by the price '\$ 28.9'. A placeholder text 'Commodi dolor veritatis eum minima. Cumque architecto ut iste. Quasi nihil ut et iusto quod.' is shown. The total number of pages is listed as 'TOTAL PAGES: 237'. At the bottom right is a red 'ADD TO CART' button.



Book Cd Game BookAdmin

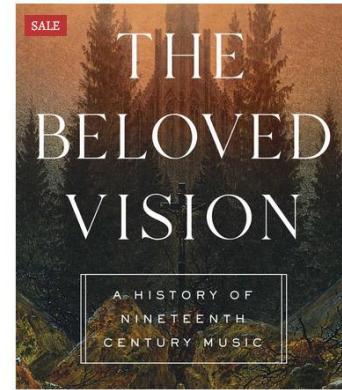
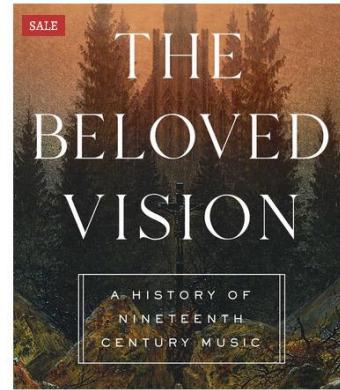
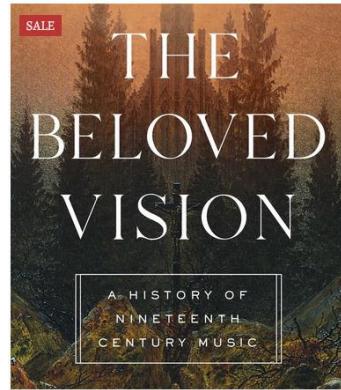


Search



Book Products

Add Book



qui
★★★★★

\$ 27.3

ut
★★★★★

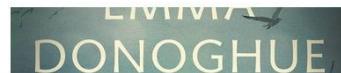
\$ 28.2

illo
★★★★★

\$ 24.5

voluptatem
★★★★★

\$ 25.5



Book deleted successfully X

NOTE: Same form and design is implemented for CD and Game admin's CRUD so, to avoid redundancy, I won't be showcasing BookAdmin and CDAdmin's Crud functionality. However, I will show case them in the demo video.

Form Validation

The screenshot shows a web application interface for managing books. At the top, there is a navigation bar with icons for Book, Cd, Game, and BookAdmin, along with a search bar. Below the navigation bar, the title "Add Books" is displayed. The main content area contains a form for adding a book, which includes fields for Title, Name (optional), Author, Price, Description, and Pages. Each field has a red circular validation icon to its right. Below each field, there is a red error message indicating the validation rule that failed. The "Description" field is highlighted with a gray background, and its error message states "The description field is required." The "Pages" field also has an error message stating "The pages must be a number." The "Image" field, represented by a file input, has an error message stating "The image field is required." A "Submit" button is located at the bottom of the form.

Add Books

Title ①
Title must only contain alphabets, whitespace & hyphens

Name (optional) ①
Name must only contain alphabets, whitespace & dots

Author ①
Author must only contain alphabets, whitespace & dots

Price ①
The price must be a number.

Description ①
The description field is required.

Pages ①
The pages must be a number.

Browse... No file selected. ①
The image field is required.

Submit

- Controller Code

```
app > Http > Controllers > BookController.php
32     return view('books.create');
33 }
34
35 /**
36 * Store a newly created resource in storage.
37 *
38 * @param \Illuminate\Http\Request $request
39 * @return \Illuminate\Http\Response
40 */
41 public function store(Request $request)
42 {
43     //dd($request);
44     $validateMsg = $request -> validate([
45         'title' => 'required|max:50|regex:/^[\pL\s\-\]+$/u',
46         'name' => 'nullable|min:3|regex:/^[\pA-Z\p{Pd}]+$/',
47         'author' => 'required|regex:/^[\pA-Z\p{Pd}]+$/',
48         'price' => 'required|numeric',
49         'description' => 'required|min:25',
50         'pages' => 'required|numeric',
51         'image' => 'required',
52     ], [
53         'title.regex' => 'Title must only contain alphabets, whitespace & hyphens',
54         'name.regex' => 'Name must only contain alphabets, whitespace & dots',
55         'author.regex' => 'Author must only contain alphabets, whitespace & dots',
56         'image.image' => 'Selected file must be an image.'
57     ]);
58     Book::create($request->all());
59     return redirect()->route('books.index')->with('success', 'Book Added Sucessfully!');
60 }
```

- Edit Form Validation

The screenshot shows a web application interface for managing books. At the top, there is a navigation bar with icons for Book, Cd, Game, and BookAdmin, along with a search bar and a shopping cart icon. The main content area is titled "Edit Books" and displays a book cover for "Sonia" by Kristina Kihl I. Below the cover, there are several input fields with validation messages:

- Title:** Sonia (Required)
- Author:** Kristina Kihl I.* (Author must only contain alphabets, whitespace & dots)
- Pages:** 27.3
- Description:** Iste et mollitia aut voluptatem. Laborum ducimus ipsam quis et voluptas est eos. Aut nostrum exercitationem occaecati consequatur delectus ut.
- Pages (again):** 485b (The pages must be a number)
- File Selection:** Browse... No file selected.

At the bottom of the form is a red "Update" button.

- Update form validation code

```
app > Http > Controllers > BookController.php
84     /**
85      * Update the specified resource in storage.
86      *
87      * @param \Illuminate\Http\Request $request
88      * @param \App\Models\Book $book
89      * @return \Illuminate\Http\Response
90      */
91     public function update(Request $request, Book $book)
92     {
93         if($request->image==null){
94             $request['image']=$book->image;
95         }
96         $validateMsg = $request -> validate([
97             'title' => 'required|max:50|regex:/^[\pL\s\-\]+$/u',
98             'name' => 'nullable|regex:/^[\pL\s\.\-]+$/|min:3',
99             'author' => 'required|regex:/^[\pL\s\.\-]+$/',
100            'price' => 'required|numeric',
101            'description' => 'required|min:25',
102            'pages' => 'required|numeric',
103        ], [
104            'title.regex' => 'Title must only contain alphabets, whitespace & hyphens',
105            'name.regex' => 'Name must only contain alphabets, whitespace & dots',
106            'author.regex' => 'Author must only contain alphabets, whitespace & dots',
107        ]);
108
109         $request->except('image');
110         $book->update($request->all());
111         return redirect()->route('books.show',$book->id)->with('success','Book Data Updated Sucessfully!');
112     }
```

Search Functionality

- Showing All data by default

The screenshot shows a search results page with the following interface elements:

- Header:** Includes a logo (EM), navigation links (Book, Cd, Game, BookAdmin), a shopping cart icon with a notification dot, and a search bar containing the text "io".
- Filter/Sort:** Buttons for "Sort By" and "Sort".
- Category:** A horizontal menu with links: All, Cd, Book, Games.
- Search Results:** Three main book covers displayed horizontally:
 - AN IMMENSE WORLD** by Ed Yong. Cover features a monkey's face and a butterfly. Rating: ★★★★☆, Price: \$ 28.2.
 - THE BELOVED VISION**. Cover features a forest scene. Rating: ★★★★☆, Price: \$ 28.8.
 - BRIDGE TO THE SUN**. Cover features three soldiers holding an American flag. Rating: ★★★★☆, Price: \$ 29.4.
- Bottom Row:** Two additional book covers displayed horizontally:
 - LEMONADE**. Repeated cover of the first book.
 - IMAGINE DRAGONS ORIGINS**. Cover featuring a rainbow and a cityscape. Rating: ★★★★★, Price: \$ 29.4.

- Selecting ‘Book’ category of search

The screenshot shows a user interface for a library or bookstore. At the top, there is a navigation bar with a logo, a search bar containing the text "io", and a magnifying glass icon. Below the navigation bar, there are buttons for "Sort By" and "Sort". To the right, there are category links: All, Cd, Book (which is underlined in red), and Games.

The main content area displays three book covers:

- An Immense World by Ed Yong**: The cover features a close-up of a lemur's face and a butterfly, with the subtitle "THE HIDDEN REALMS AROUND US" and "PULITZER PRIZE-winning author of I CONTAIN MULTITUDES".
- The Beloved Vision: A History of Nineteenth-Century Music**: The cover shows a landscape with trees and a small building, with the subtitle "A HISTORY OF NINETEENTH CENTURY MUSIC".
- Bridge to the Sun: The Secret Role of the Japanese Americans Who Fought in the Pacific in World War II**: The cover features three Japanese American soldiers standing in a field, holding rifles and a flag, with the subtitle "The Secret Role of the Japanese Americans Who Fought in the Pacific in World War II".

Below each book cover, there is some descriptive text and a price:

- For "An Immense World":
ratione
★★★★★
\$ 28.2
- For "The Beloved Vision":
odio
★★★★★
\$ 28.8
- For "Bridge to the Sun":
distinctio
★★★★★
\$ 29.4

- Selecting ‘Game’ category

EM Book Cd Game BookAdmin ▾ 0

io Sort

Sort By Sort

All Cd Book Games

atque
★★★★★
\$ 25.1

maxime
★★★★★
\$ 26.4

magnam
★★★★★
\$ 25.6

non
★★★★★
\$ 27

Sorting Functionality

- Sort by 'Price' Low-High

The screenshot shows a user interface for managing inventory or products. At the top, there is a navigation bar with a logo, a search bar, and a shopping cart icon. Below the navigation, there are buttons for 'Sort By' and 'Sort'. A horizontal menu bar includes 'All', 'Cd', 'Book', and 'Games', with 'Games' currently selected. The main content area displays a grid of four items:

Image	Name	Rating	Price
	voluptatem	★★★★★	\$ 21.7
	exercitationem	★★★★★	\$ 23.9
	consequatur	★★★★★	\$ 25.8
	maiores	★★★★★	\$ 26.5

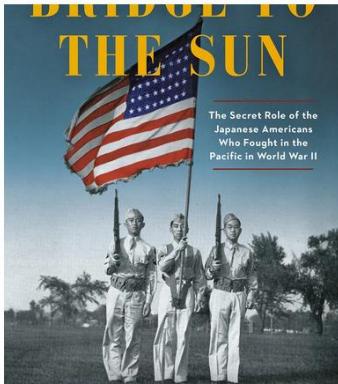
- Sort by ‘Price’ High-Low

EM Book Cd Game BookAdmin ▾ 0

Search 🔍

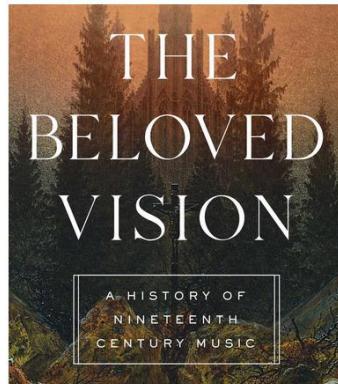
Sort By ▼ Sort

All Cd Book Games



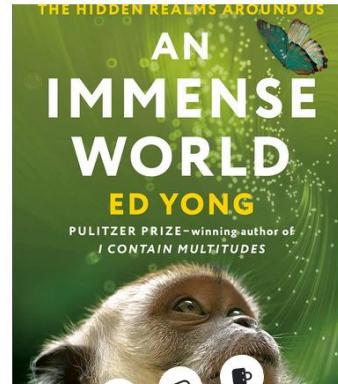
Bridge to THE SUN
The Secret Role of the Japanese Americans Who Fought in the Pacific in World War II

distinctio
★★★★★
\$ 29.4



THE BELOVED VISION
A HISTORY OF NINETEENTH CENTURY MUSIC

odio
★★★★★
\$ 28.8



AN IMMENSE WORLD
ED YONG
PULITZER PRIZE-winning author of *I Contain Multitudes*

ratione
★★★★★
\$ 28.2

- Sort by 'Alphabet' A-Z

Screenshot of a game store interface showing sorted results:

Header: Book Admin, Search bar, Sort button (red), Title A-Z dropdown.

Navigation: All, Cd, Book, Games (Games is selected).

Sort by: Title A-Z.

Results:

Image	Name	Rating	Price
	WATCH DOGS 2	★★★★★	\$ 21
	WATCH DOGS 2	★★★★★	\$ 26.3
	APEX LEGENDS	★★★★★	\$ 22.1
	RED DEAD REDEMPTION II	★★★★★	\$ 25.1

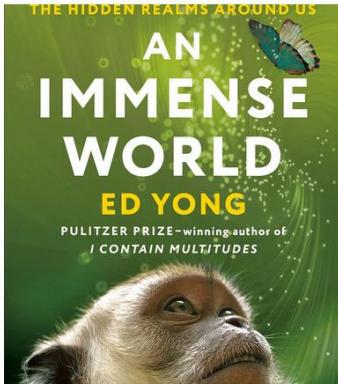
- Sort by 'Alphabet' Z-A

EM Book Cd Game BookAdmin ▾ 0

Search 🔍

Title Z- Sort

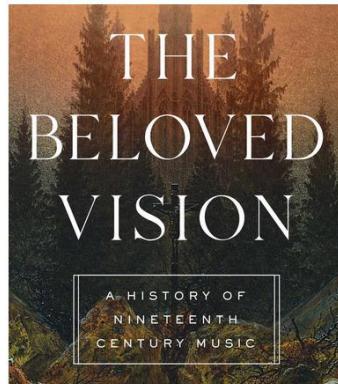
All Cd Book Games



**AN
IMMENSE
WORLD**
ED YONG

PULITZER PRIZE-winning author of
I CONTAIN MULTITUDES

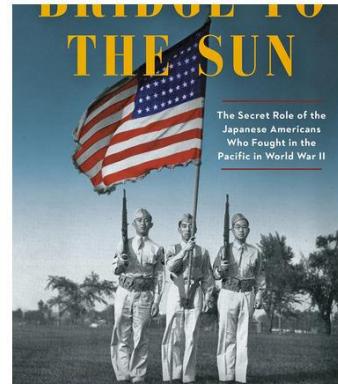
ratione
★★★★★
\$ 28.2



**THE
BELOVED
VISION**

A HISTORY OF
NINETEENTH
CENTURY MUSIC

odio
★★★★★
\$ 28.8



**BRIDGE TO
THE SUN**

The Secret Role of the
Japanese Americans
Who Fought in the
Pacific in World War II

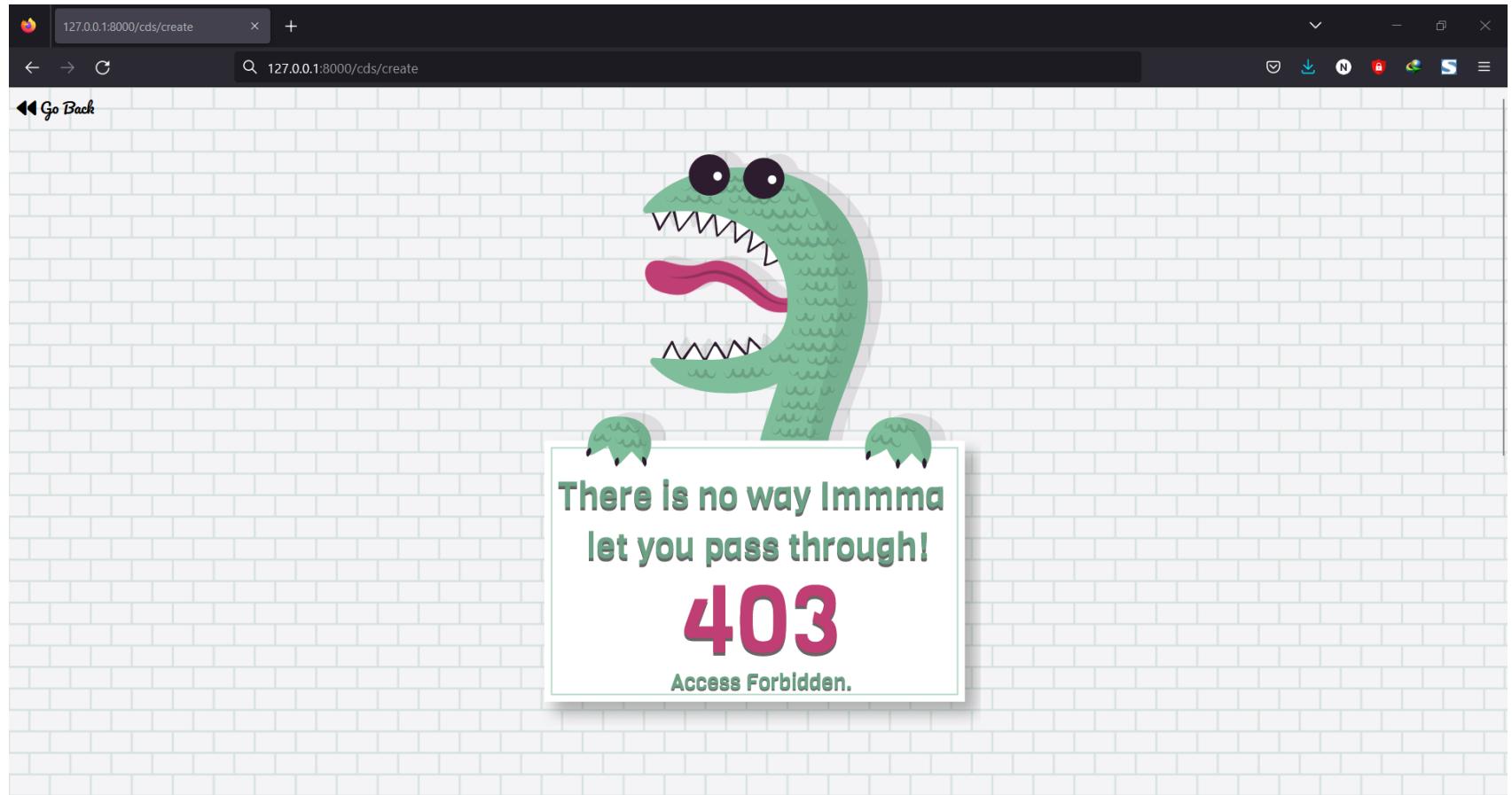
distinctio
★★★★★
\$ 29.4

- Searching and Sorting Model Code

```
app > Models > Book.php
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class Book extends Model
9  {
10     use HasFactory;
11     protected $fillable = ['title', 'name', 'author', 'price', 'description', 'pages', 'image'];
12     protected $guarded = ['id'];
13
14     public function scopeFilter($query, array $filters)
15     {
16         if($filters['search'] ?? false)
17         {
18             $query->where('title','like','%'.$filters['search'].'%')
19                 ->orWhere('name','like','%'.$filters['search'].'%')
20                 ->orWhere('author','like','%'.$filters['search'].'%')
21                 ->orWhere('category','like','%'.$filters['search'].'%');
22
23             if($filters['sort'] ?? false) {
24                 $sort = explode("-", $filters['sort']);
25                 $query->orderBy($sort[0],$sort[1]);
26             }
27         }
28         else {
29             return $query->whereRaw('1 = 0');
30         }
31     }
32 }
```

Middleware and Laravel Gate

- If BookAdmin tries to access CdAdmin's crud form my manually typing the URL, 403 page will be shown



- Middleware Code

```
<?php

namespace App\Http\Controllers;

use App\Models\Book;
use Illuminate\Http\Request;

class BookController extends Controller
{
    public function __construct()
    {
        $this->middleware('can:isbookAdmin')->except(['index', 'show']);
    }
}
```

- Book Admin Gate Defined Code

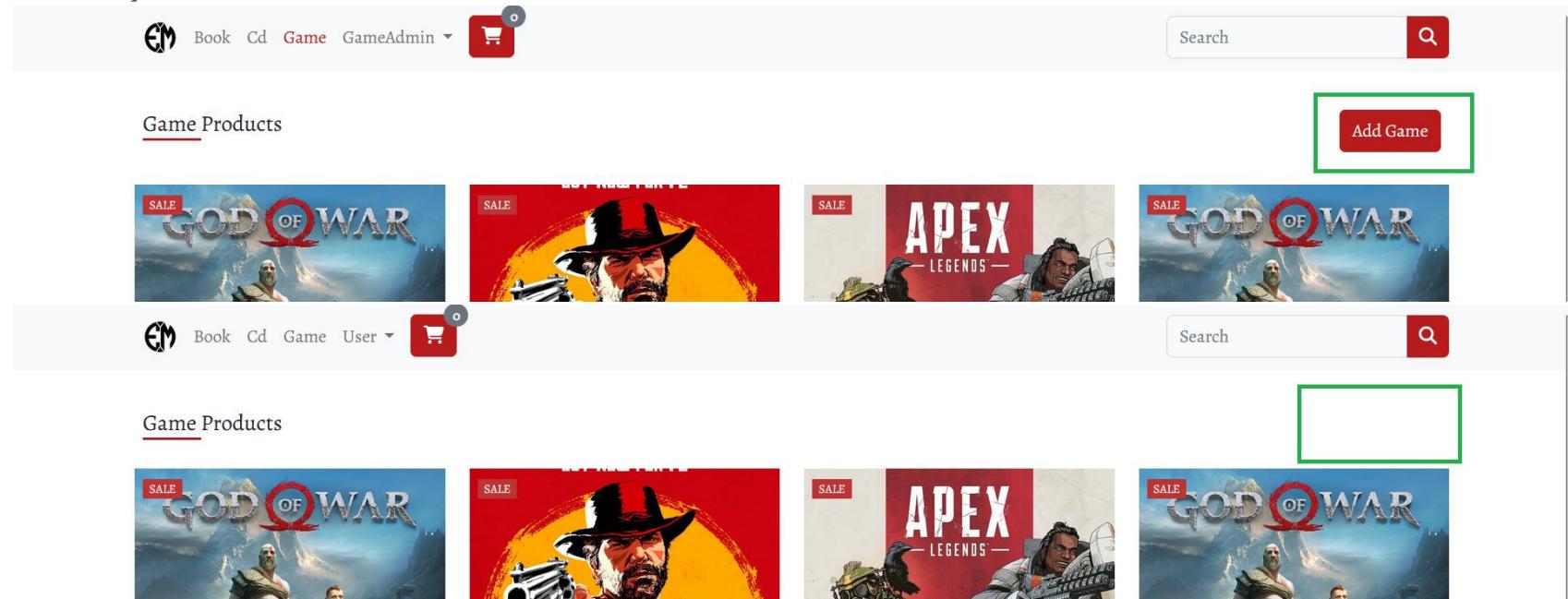


```

app > Providers > AuthServiceProvider.php
24     public function boot()
25     {
26         $this->registerPolicies();
27
28         /* define an admin user role */
29         Gate::before(function ($user) {
30             if ($user->role == 'admin') {
31                 return true;
32             }
33         });
34
35         /* define a book admin user role */
36         Gate::define('isbookAdmin', function($user) {
37
38             if ($user->role == 'bookAdmin') {
39                 return true;
40             }
41
42         });
43
44         /* define a cd admin user role */
45         Gate::define('iscdAdmin', function($user) {
46
47             if ($user->role == 'cdAdmin') {
48                 return true;
49             }
50
51         });
52

```

- Same 403-Access Denied page, in the case of CdAdmin and GameAdmin if they try to access another admin's CRUD page
- Similarly, normal User, Guest or other Admins won't be shown the CRUD button



EM Book Cd Game GameAdmin ▾ 

Search 



◀ Back

GAME
QUI
\$ 27.4

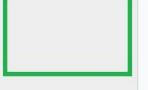
EM Book Cd Game User ▾ 

Search 



◀ Back

GAME
QUI
\$ 27.4



- However, there is a SuperAdmin who has access to all other individual admin's crud functionality

Book Products

Add Book

Search

SOL ENI
MIRCEA CARTARESCU

THE BELOVED VISION

THE BELOVED VISION

THE BELOVED VISION

BOOK QUI \$ 27.3

Iste et mollitia aut voluptatem. Laborum ducimus ipsam quis et voluptas est

 Book Cd Game Nitin ▾ 

Search 

Cd Products



 Book Cd Game Nitin ▾ 

Search 



 Back

CD
LIBERO
\$ 22.7

 Book Cd Game Nitin ▾ 

Search 

Game Products







Book Cd Game Nitin ▾



Search



← Back



GAME

QUI

\$ 27.4

Reprehenderit dolores blanditiis quia. Ut magni non dolores natus ratione.
Quia error nesciunt neque odit neque.

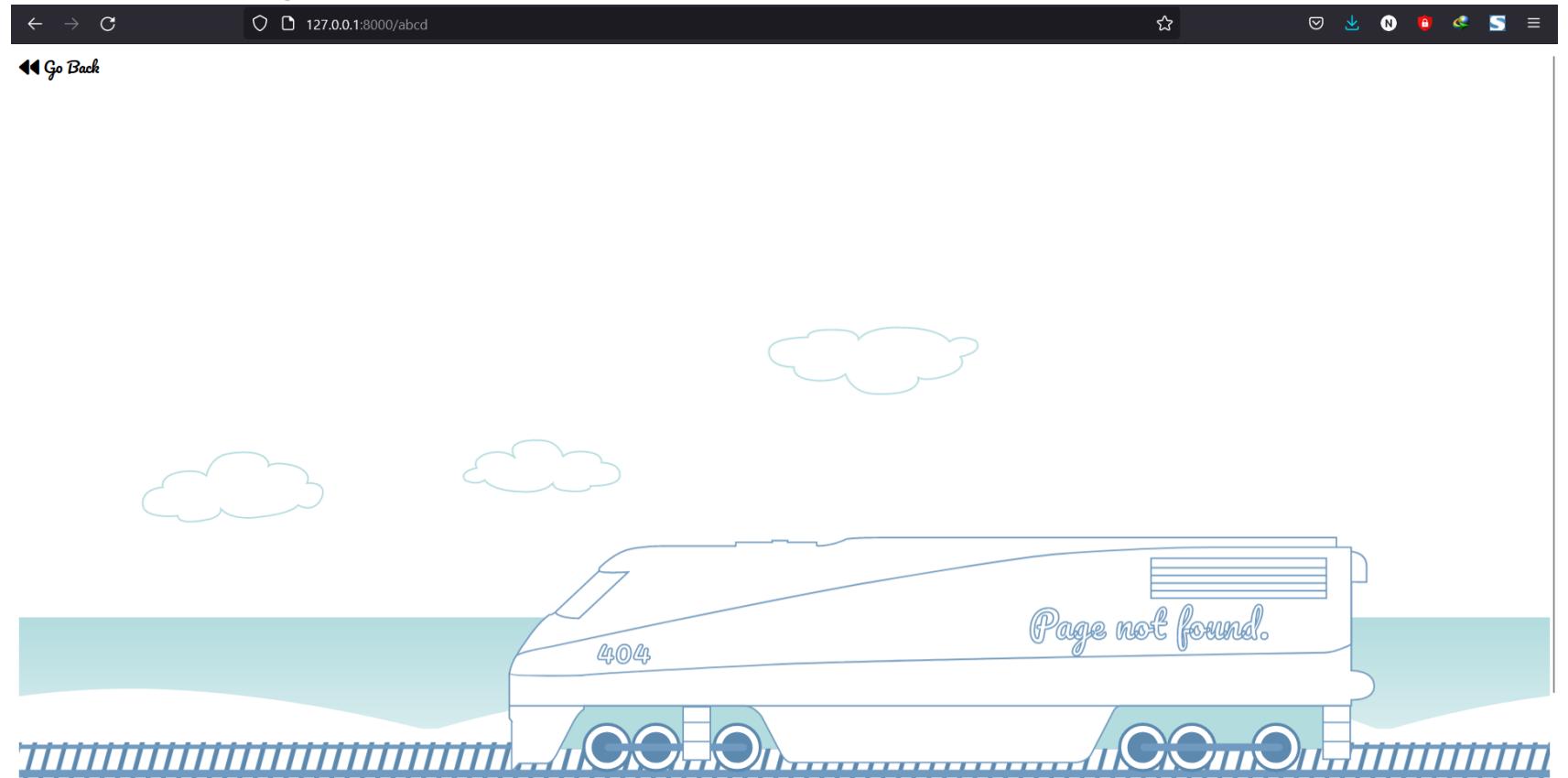
- Using Laravel Gate::before for Super Admin, i.e., admin

```
app > Providers > AuthServiceProvider.php
  ...
11     * The model to policy mappings for the application.
12     *
13     * @var array<class-string, class-string>
14     */
15 protected $policies = [
16     // 'App\Models\Model' => 'App\Policies\ModelPolicy',
17 ];
18
19 /**
20 * Register any authentication / authorization services.
21 *
22 * @return void
23 */
24 public function boot()
25 {
26     $this->registerPolicies();
27
28         /* define an admin user role */
29         Gate::before(function ($user) {
30             if ($user->role == 'admin') {
31                 return true;
32             }
33         });
34 }
```

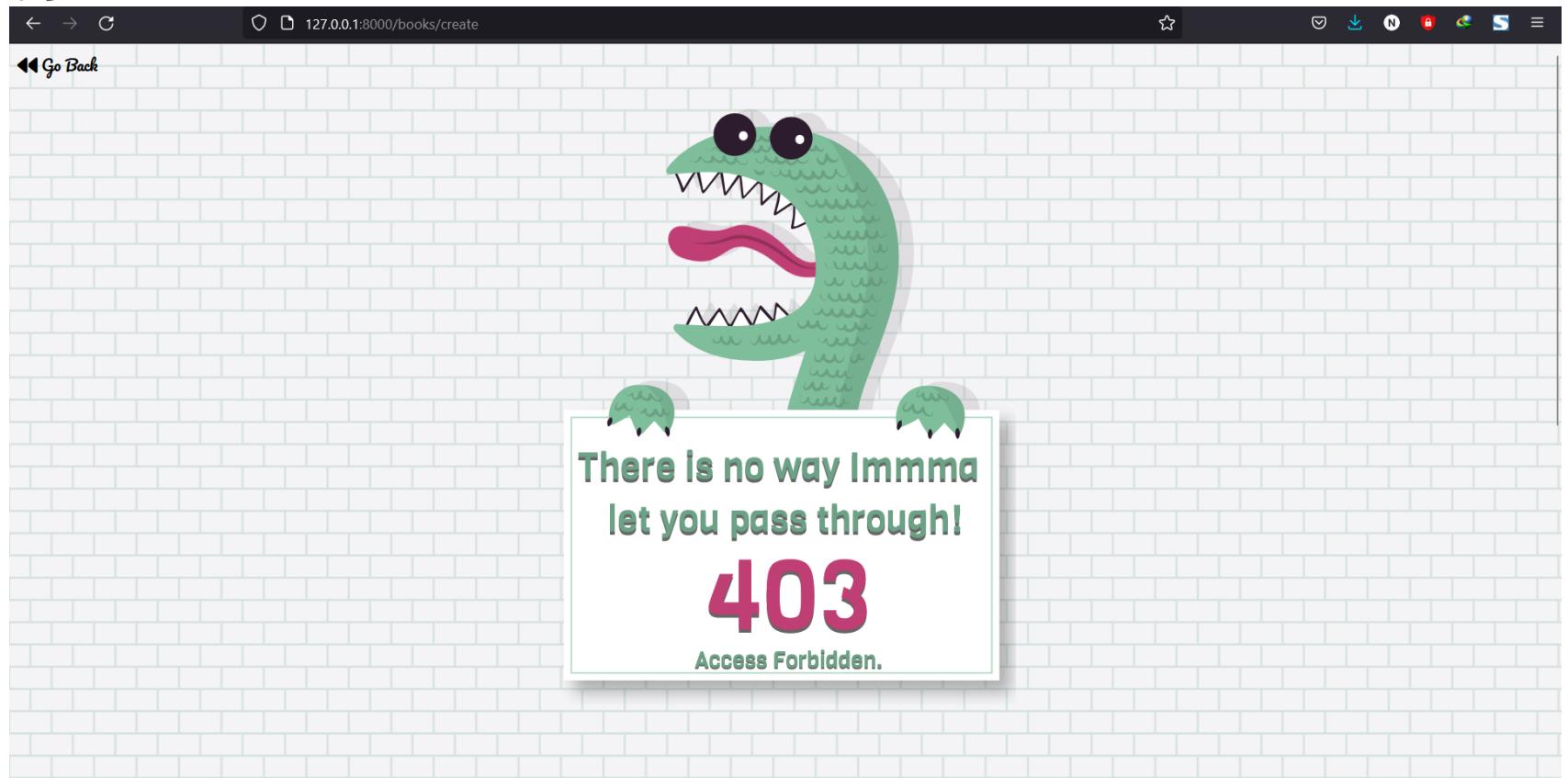
- Thus, in total my website has 5 users: Guest, Logged in User, Book Admin, Cd Admin, Game Admin and Super Admin.

Custom 404 Not Found and 403 Forbidden Page

- 404 Not Found Page



- 403 Forbidden



Pagination for Book, Cd and Game pages



- Pagination code

```
public function index()
{
    $books = Book::latest()->paginate(8);
    return view('books.index', compact('books'));
}

<div class="d-flex justify-content-center">
    {{ $books->links()  }}
</div>
```

- Pagination custom color using scss

```
// Pagination
$pagination-active-bg: #B71C1C;
$pagination-active-border-color: #fff;
$pagination-color: #000;
$pagination-hover-color: #fff;
$pagination-hover-bg: #B71C1C;
$pagination-focus-color: #000;
$pagination-focus-box-shadow: none;
```

Forget Password using SMTP

The screenshot displays a two-step process for forgot password on a website.

Step 1: Enter Email Address

The top part shows the initial "Forgot Password?" form. It includes a logo, navigation links (Book, Cd, Game, Login, Register), a shopping cart icon with a notification dot, and a search bar. The main form has a header "Reset Password" and a sub-header "Forgot Password?". It contains an "Email Address" input field containing "nitinutsav80@gmail.com" and a red "Send Password Reset Link" button.

Step 2: Confirmation Message

The bottom part shows the confirmation message after the link was sent. The same header and navigation links are present. The main form now displays a green success message: "We have emailed your password reset link!". The "Email Address" input field is empty, and the "Send Password Reset Link" button is visible below it.



Example <awetest8848@gmail.com>
to me ▾

3:30 PM (0 minutes ago)



Hello!

You are receiving this email because we received a password reset request for your account.

[Reset Password](#)

This password reset link will expire in 60 minutes.

If you did not request a password reset, no further action is required.

Regards,
Laravel

If you're having trouble clicking the "Reset Password" button, copy and paste the URL below into your web browser: <http://127.0.0.1:8000/password/reset/7d4aad6d795cf698a12c3b528021b6d4f5c4f95686e5b77c5664a641c62e33b3?email=nitinutsav80%40gmail.com>

© 2023 Laravel. All rights reserved.

Reply

Forward

The screenshot shows a user interface for a website. At the top left is a logo consisting of a stylized 'EM'. To its right are navigation links: Book, Cd, Game, Login, and Register. Next to the 'Login' link is a red shopping cart icon with a small white circle containing the number '0' above it. On the far right is a search bar with the placeholder 'Search' and a magnifying glass icon.

The main content area features a 'Reset Password' form. It has three input fields: 'Email Address' containing 'nitinutsav80@gmail.com', 'Password' containing six dots ('••••••'), and 'Confirm Password' also containing six dots ('••••••'). Below these fields is a red button labeled 'Reset Password'.

- After changing the password, user is logged in and redirected to the dashboard page

Add to Cart using darryldecode package

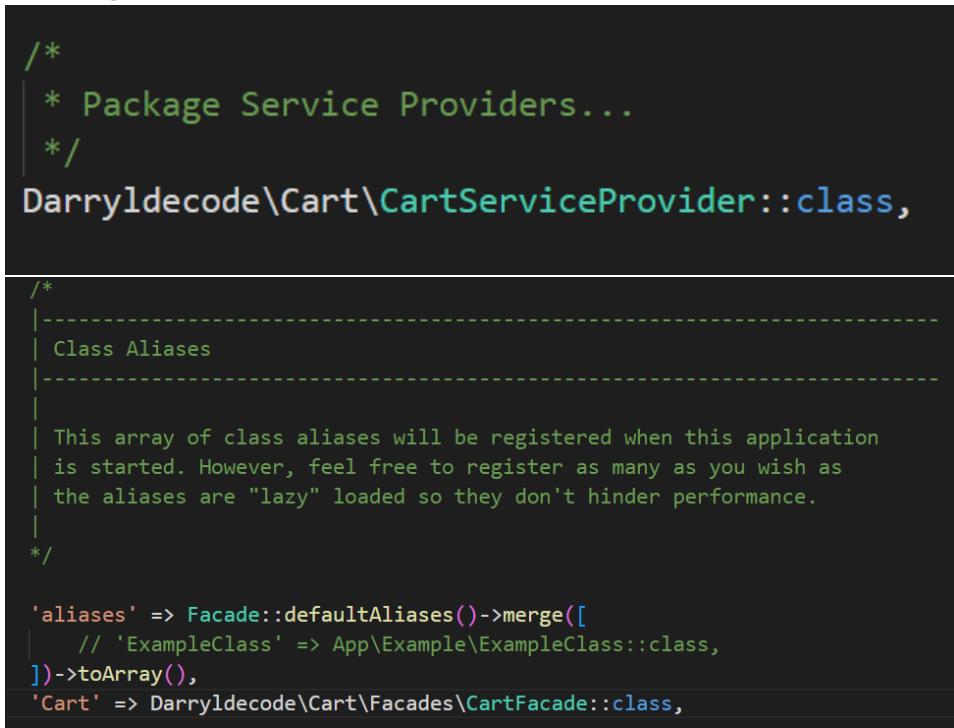
- Installing necessary composer package for add-to-cart



```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    POLYGLOT NOTEBOOK

PS C:\Users\nitin> composer require "darryldecode/cart"
```

- Adding these lines of code to Service Provider Array inside config/app.php



```
/*
| * Package Service Providers...
|
| */
Darryldecode\Cart\CartServiceProvider::class,
```



```
/*
| -----
| Class Aliases
| -----
|
| This array of class aliases will be registered when this application
| is started. However, feel free to register as many as you wish as
| the aliases are "lazy" loaded so they don't hinder performance.
|
| */
'aliases' => Facade::defaultAliases()->merge([
    // 'ExampleClass' => App\Example\ExampleClass::class,
])->toArray(),
'Cart' => Darryldecode\Cart\Facades\CartFacade::class,
```

- Adding item to cart from ‘index’ page

The screenshot shows a web-based bookstore interface. At the top, there is a navigation bar with a logo, a search bar, and a user icon indicating two notifications. Below the navigation bar, a section titled "Book Products" is displayed. A grid of book covers is shown, with four books in the main row and two partial book covers below them.

Books in the Main Row:

- Left Book:** Title "SEN", Author "MIRCEA CARTARESCU", "SALE" badge. Below the cover: "qui" with 5 stars, "\$ 27.3".
- Middle Book:** Title "THE BELOVED VISION", Subtitle "A HISTORY OF NINETEENTH CENTURY MUSIC", "SALE" badge. Below the cover: "ut" with 5 stars, "\$ 28.2".
- Right Book:** Title "THE BELOVED VISION", Subtitle "A HISTORY OF NINETEENTH CENTURY MUSIC", "SALE" badge. Below the cover: "illo" with 5 stars, "\$ 24.5".
- Far Right Book:** Title "THE BELOVED VISION", Subtitle "A HISTORY OF NINETEENTH CENTURY MUSIC", "SALE" badge. Below the cover: "voluptatem" with 5 stars, "\$ 25.5".

Books in the Row Below:

- Left Book:** Title "THE HIDDEN REALMS AROUND US", Author "AN", "SALE" badge.
- Middle Book:** Title "DONOGHUE", Author "LIVIVIVI", "SALE" badge.
- Right Book:** Title "THE", Subtitle "A HISTORY OF NINETEENTH CENTURY MUSIC", "SALE" badge.

A green callout box in the bottom right corner of the screen displays the message "Item Added to Cart Successfully" with a close button.

- Adding item to cart from 'show' page

EM Book Cd Game User 7 

Search 



← Back

CD
LIBERO
\$ 22.7

Ipsam ut voluptates et quam molestiae. Voluptatem molestias dolor illum et. Molestiae facere maiores molestias. Reprehenderit qui culpa magnam et. Ut ratione quisquam sed tempora.

PLAYLENGTH: 22 MIN

ADD TO CART

Services About Essential Mart

Item Added to Cart Successfully 

- Updating item's quantity

The screenshot shows the shopping cart page of the Essential Mart website. At the top, there is a navigation bar with links for Book, Cd, Game, User, and a search bar with a magnifying glass icon. A red shopping cart icon with a '6' notification is also present.

The main content area is titled "Your Cart". Below it, a link "Continue Shopping" is visible. The total grand total is listed as "\$ 163.8". It is mentioned that there are 6 items in the cart. A "Remove All" button is located on the right side.

A single item is displayed in the cart:

	qui	<input type="button" value="6"/> <input type="button" value="Update"/>	\$ 27.3	
--	-----	--	---------	--

At the bottom of the page is a large red button labeled "Proceed to Checkout".

The screenshot shows the footer of the Essential Mart website. It includes links for Services (Books, Cds, Games), About (Company, Sellers, Team), and a section for Essential Mart. The footer also features a green success message box.

Services

- Books
- Cds
- Games

About

- Company
- Sellers
- Team

Essential Mart

We are a one-stop shop for all your entertainment needs: books, CDs, and games at affordable prices. Our inventory is constantly updated, so you can always find what you're looking for.

Item's Quantity Updated Successfully

- Deleting items from cart

Your Cart

← Continue Shopping

Grand Total: \$ 213.9
You have 8 items in your cart

Remove All

Item	Quantity	Price	Action
qui	6	\$ 27.3	<input type="button" value="Remove"/>
libero	1	\$ 22.7	<input type="button" value="Remove"/>
qui	1	\$ 27.4	<input type="button" value="Remove"/>

Proceed to Checkout



Book Cd Game User



Search



Your Cart

[← Continue Shopping](#)

Grand Total: \$ 186.5

You have 7 items in your cart

[Remove All](#)



qui

6

[Update](#)

\$ 27.3



libero

1

[Update](#)

\$ 22.7



[Proceed to Checkout](#)

Item Removed from Cart Successfully

[Services](#)

[About](#)

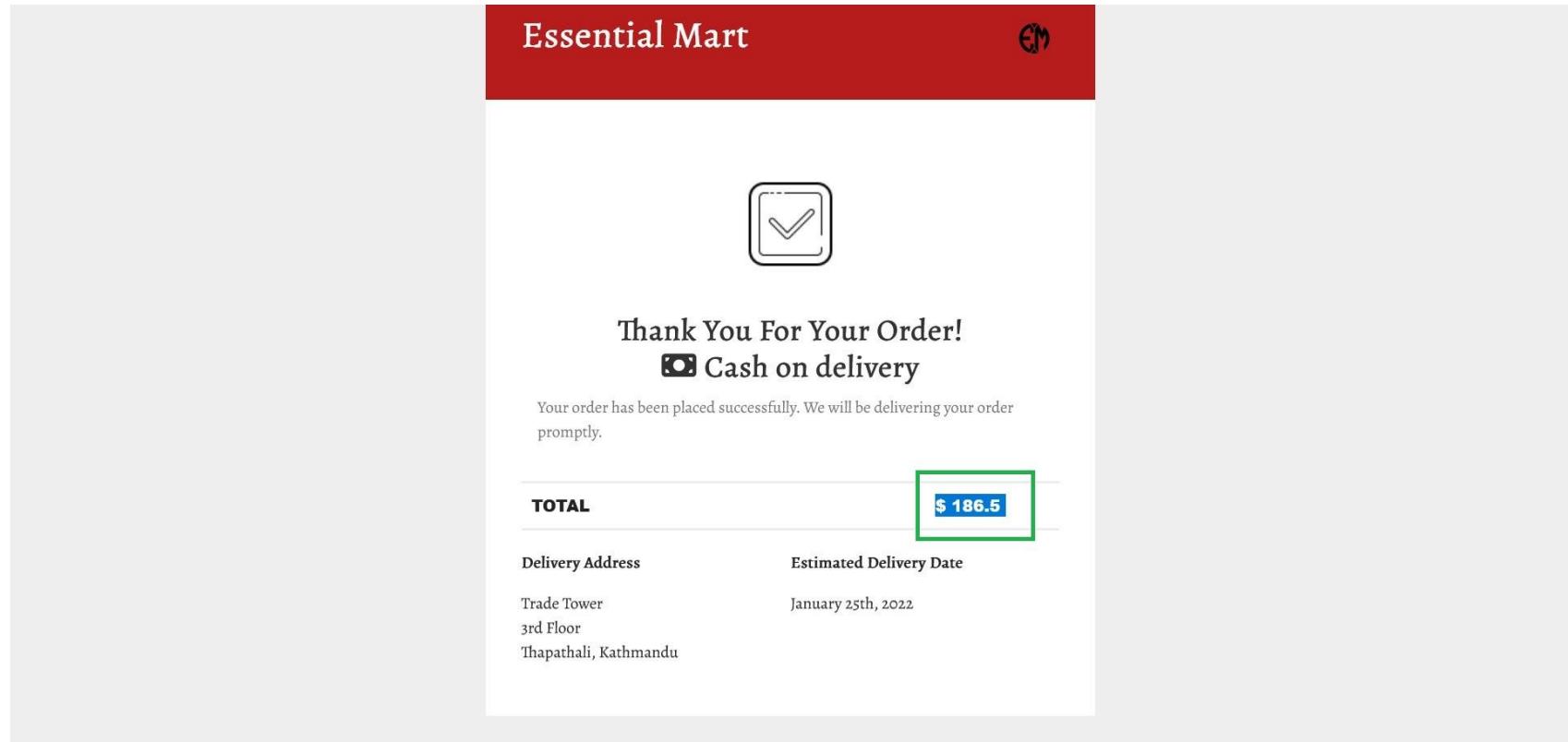
[Essential Mart](#)

- Proceed to checkout functionality

The screenshot shows a shopping cart page from an e-commerce website. At the top, there is a navigation bar with a logo, menu items (Book, Cd, Game, User), a search bar, and a red shopping cart icon with a '7' notification. Below the header, the page title 'Your Cart' is displayed. A 'Continue Shopping' link is available. The cart summary shows a 'Grand Total: \$ 186.5' (with a green box around the '\$ 186.5') and a note that there are 7 items in the cart. A 'Remove All' button is located on the right. Two items are listed in the cart:

Item	Quantity	Update	Price	Action
qui	6	Update	\$ 27.3	Remove
libero	1	Update	\$ 22.7	Remove

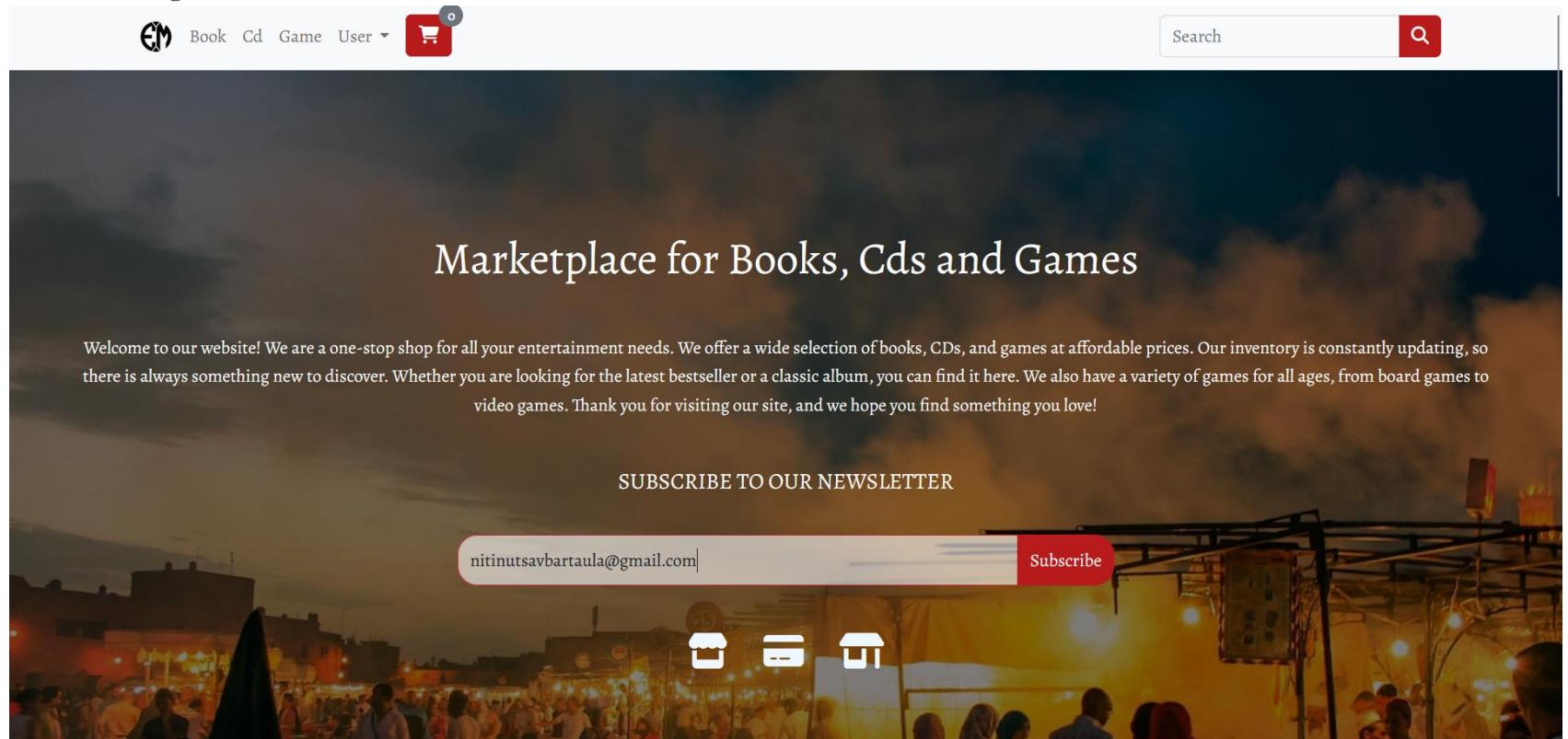
A large black button at the bottom center contains the text 'Proceed to Checkout'. A green arrow points upwards towards this button.



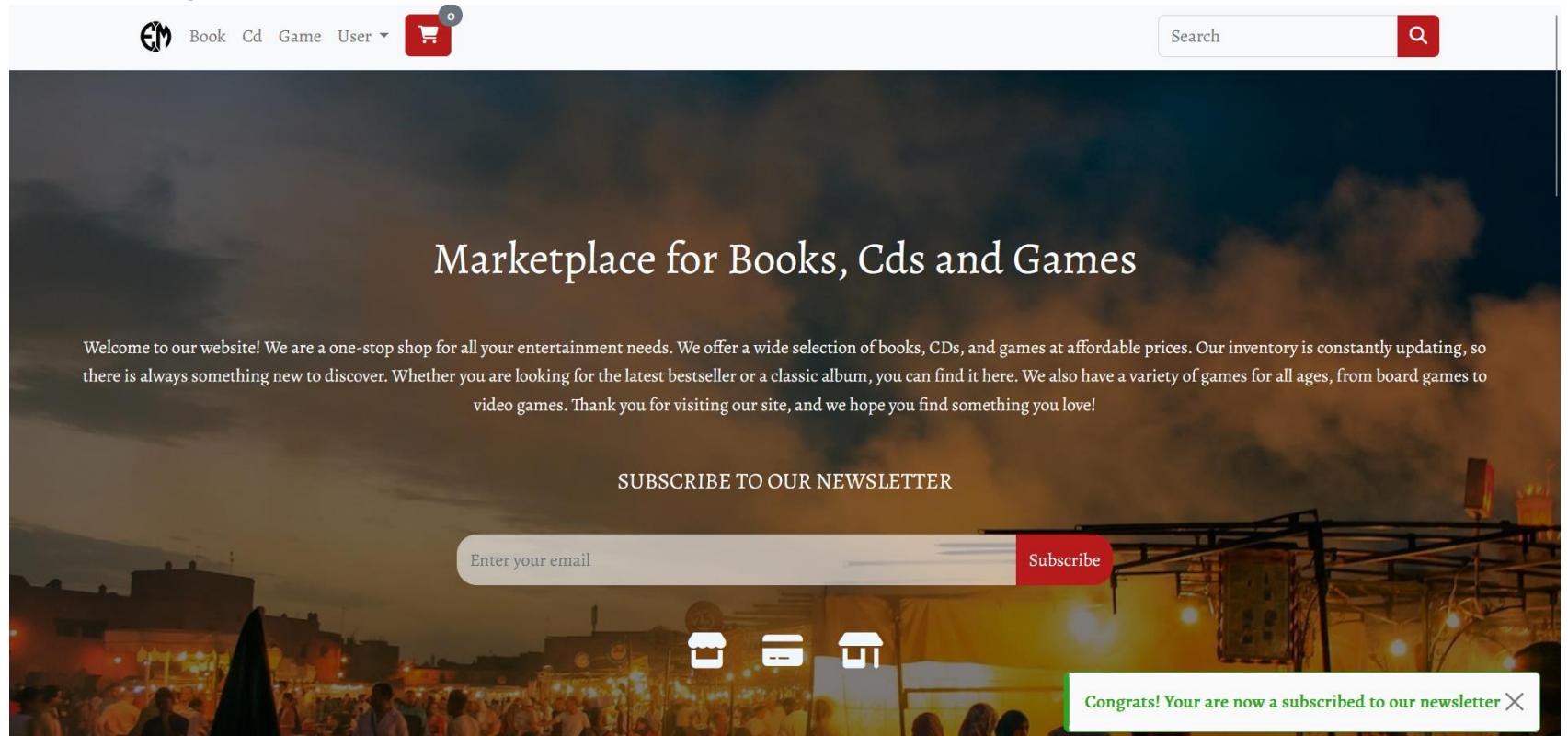
- After 'checkout', the user is redirected to this page, showing their grand total. The cart then gets cleared.

Mailchimp API Integration

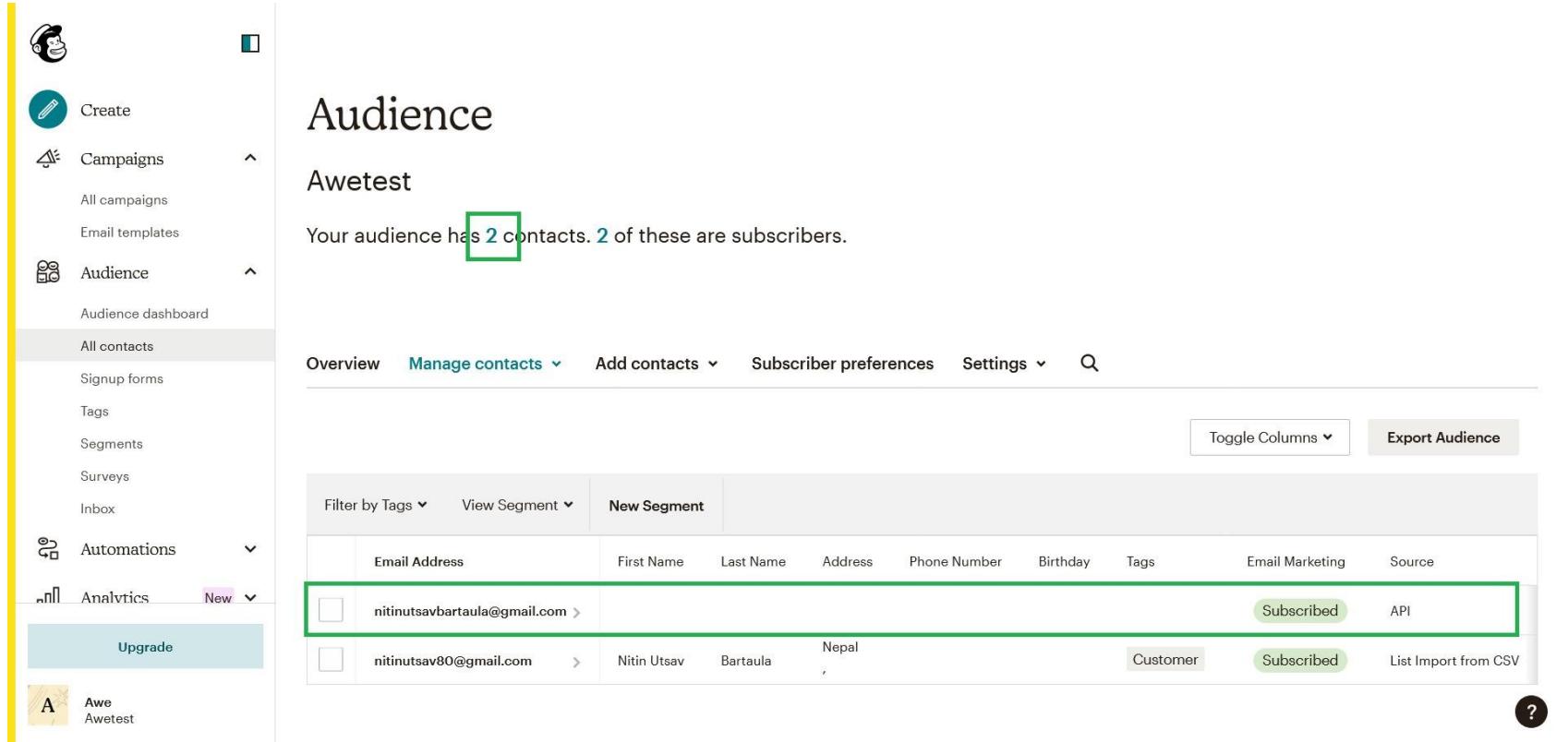
- Subscribing to newsletter



- Alert message after subscribed successfully



- Contact added to MailChimp Audience



The screenshot shows the MailChimp Audience interface. On the left, there's a sidebar with various navigation options: Create, Campaigns (with sub-options All campaigns and Email templates), Audience (with sub-options Audience dashboard, All contacts, Signup forms, Tags, Segments, Surveys, and Inbox), Automations, Analytics (with a New button), and Upgrade. Below the sidebar is a user profile icon labeled "Awe Awetest". The main area is titled "Audience" and has a sub-section titled "Awetest". It displays the message "Your audience has 2 contacts. 2 of these are subscribers." Below this, there's a table with columns: Email Address, First Name, Last Name, Address, Phone Number, Birthday, Tags, Email Marketing, and Source. Two contacts are listed:

Email Address	First Name	Last Name	Address	Phone Number	Birthday	Tags	Email Marketing	Source
<input type="checkbox"/> nitinutsavbartaula@gmail.com >							Subscribed	API
<input type="checkbox"/> nitinutsav80@gmail.com >	Nitin Utsav	Bartaula	Nepal				Customer	Subscribed

- First, setting up an API key from MailChimp and using it on .env file

The screenshot shows the Mailchimp interface for managing API keys. On the left, there's a sidebar with links like Audience dashboard, All contacts, Signup forms, Tags, Segments, Surveys, and Inbox. Below that is a dropdown menu for Automations, and at the bottom, a yellow bar with an 'Upgrade' button. The main area is titled 'Your API keys' and contains a table with one row. The table columns are 'Created', 'User', 'Label', 'API key', and 'Status'. The data in the table is:

Created	User	Label	API key	Status
Jan 18, 2023 1:11 am	Awe Test (owner)	none set	609fa1ee1457a643b9e76c2854acecd7	<input checked="" type="checkbox"/> X

Below the table is a 'Create A Key' button. To the right of the table is a help icon (a question mark inside a circle) and a 'X' icon.

Below this section is another titled 'Your Mobile SDK Client keys'.

At the bottom of the screenshot is a terminal window showing a .env file with the following content:

```

> vluca
> voku
> webmozart
> autoload.php
> editorconfig
67
68 MAILCHIMP_KEY=609fa1ee1457a643b9e76c2854acecd7

```

- Configuring MailChimp as third-party services on config/services.php

```
config > services.php
22     ],
23
24     'postmark' => [
25         'token' => env('POSTMARK_TOKEN'),
26     ],
27
28     'ses' => [
29         'key' => env('AWS_ACCESS_KEY_ID'),
30         'secret' => env('AWS_SECRET_ACCESS_KEY'),
31         'region' => env('AWS_DEFAULT_REGION', 'us-east-1'),
32     ],
33
34     'mailchimp' => [
35         'key' => env('MAILCHIMP_KEY')
36     ]
37
38 ];
39
```

- MailChimp route function

```
//Route for our MailChimp
Route::get('newsletter', function() {
    request()->validate(['email' => 'required|email']);

    $mailchimp = new \MailchimpMarketing\ApiClient();

    $mailchimp->setConfig([
        'apiKey' => config('services.mailchimp.key'),
        'server' => 'us21'
    ]);

    $response = $mailchimp->lists->addListMember('1401b68cce',[ 
        'email_address' => request('email'),
        'status' => 'subscribed'
    ]);

    return redirect('/')->with('success','Congrats! You are now a subscribed to our newsletter');

});
```

PayPal Payment Package Integration

Lastly, I tried using PayPal payment integration for my cart checkout. However, I faced some errors which I unfortunately could not solve. However, I have documented my payment integration steps nonetheless

- Installing necessary composer package for PayPal

The screenshot shows a terminal window within a development environment. The terminal tab is active, displaying the command:

```
PS C:\Users\nitin\OneDrive\Documents\laravel front end\paypal documentation> composer require srmklive/paypal
```

Output from the command:

```
Info from https://repo.packagist.org: #StandWithUkraine
Using version ^3.0 for srmklive/paypal
./composer.json has been updated
Running composer update srmklive/paypal
Loading composer repositories with package information
Updating dependencies
Lock file operations: 1 install, 0 updates, 0 removals
- Locking srmklive/paypal (3.0.16)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
```

Below the terminal, another terminal window is visible, showing the results of the command:

```
PS C:\Users\nitin\OneDrive\Documents\laravel front end\paypal documentation> composer require srmklive/paypal
```

Output from the second terminal:

```
PS C:\Users\nitin\OneDrive\Documents\laravel front end\paypal documentation> composer require srmklive/paypal
nunomaduro/termwind ..... DONE
spatie/laravel-ignition ..... DONE
srmklive/paypal ..... DONE
```

Both terminals show the status "Ln 70, Col 70 Spaces: 4 UTF-8 LF Properties" at the bottom.

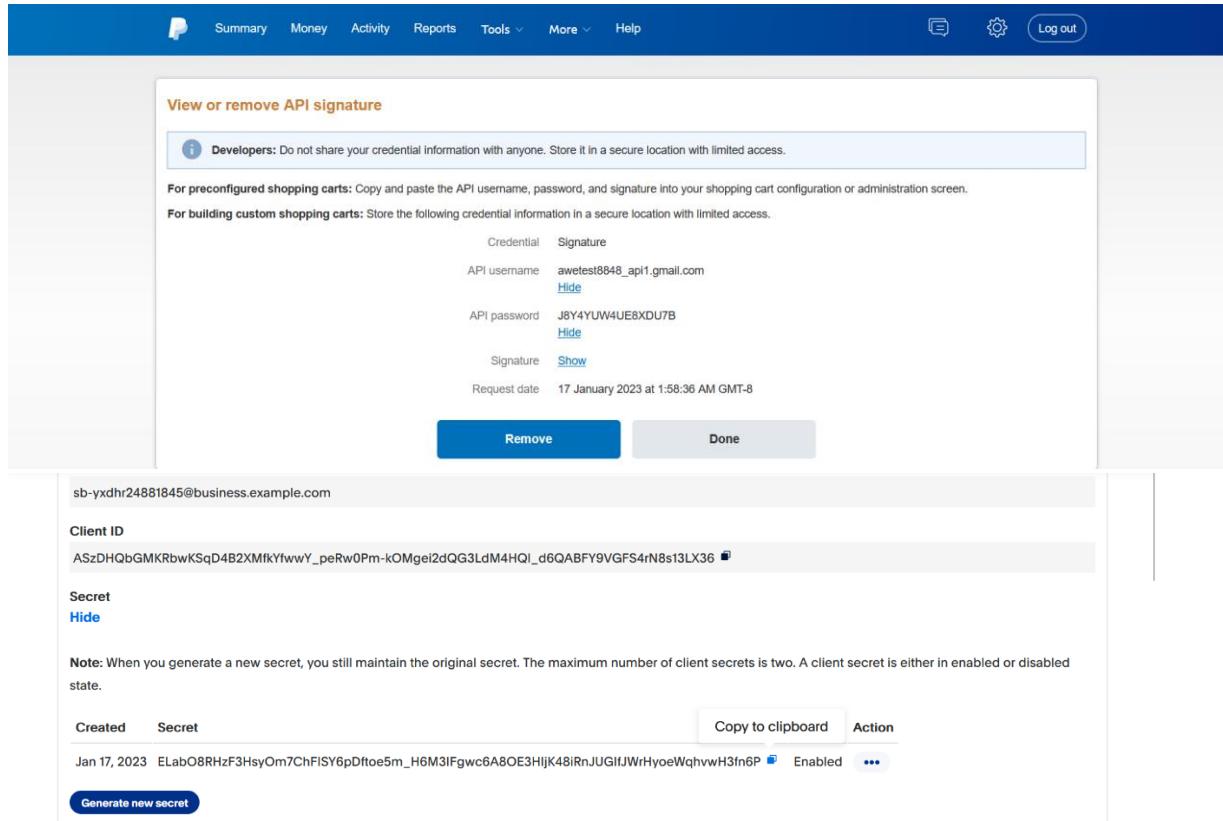
- Registering PayPal services into config/app.php

```

Go Run Terminal Help app.php - paypal documentation - Visual Studio Code
.env app.php X
config > app.php
175     Illuminate\Auth\Passwords\PasswordResetServiceProvider::class,
180     Illuminate\Session\SessionServiceProvider::class,
181     Illuminate\Translation\TranslationServiceProvider::class,
182     Illuminate\Validation\ValidationServiceProvider::class,
183     Illuminate\View\ViewServiceProvider::class,
184
185     /*
186     * Package Service Providers...
187     */
188     Darryldecode\Cart\CartServiceProvider::class,
189     Srmklive\PayPal\Providers\PayPalServiceProvider::class,
190
191     /*
.config > app.php
206     |
207     | This array of class aliases will be registered when this application
208     | is started. However, feel free to register as many as you wish as
209     | the aliases are "lazy" loaded so they don't hinder performance.
210     |
211     */
212
213     'aliases' => Facade::defaultAliases()->merge([
214         // 'ExampleClass' => App\Example\ExampleClass::class,
215     ])->toArray(),
216     'Cart' => Darryldecode\Cart\Facades\CartFacade::class,
217     'PayPal' => Srmklive\PayPal\Facades\PayPal::class,
218
219 ];
220

```

- Injecting PayPal details inside .env



The screenshot shows the PayPal Developers API signature view page. At the top, there's a navigation bar with links for Summary, Money, Activity, Reports, Tools, More, Help, and Log out. Below the navigation is a section titled "View or remove API signature". It contains a note for developers to store their credentials securely. It lists the following information:

Credential	Signature
API username	awetest8848_api1@gmail.com Hide
API password	J8Y4YUW4UE8XDU7B Hide
Signature	Show
Request date	17 January 2023 at 1:58:36 AM GMT-8

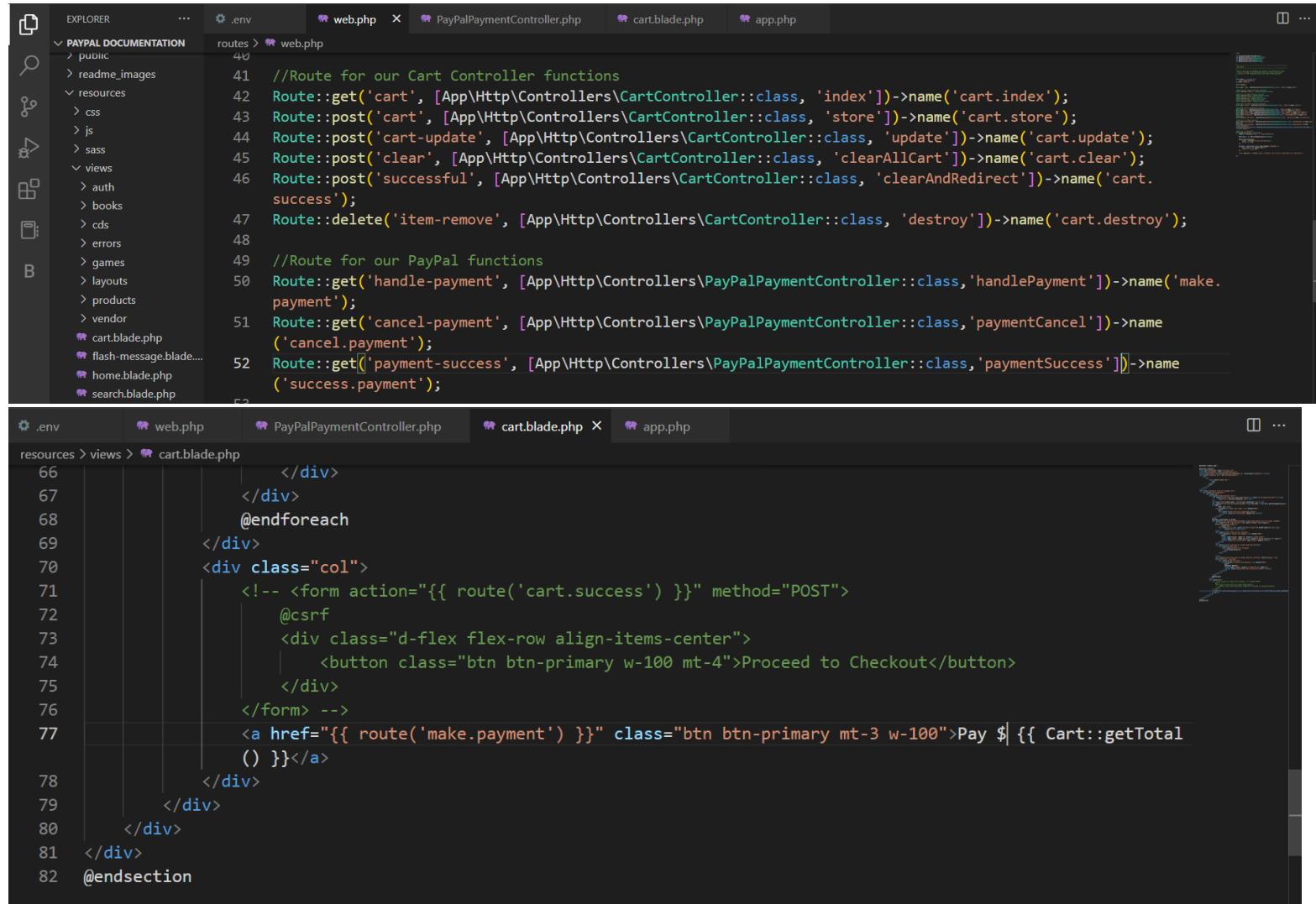
At the bottom of this section are "Remove" and "Done" buttons. Below this, the "Client ID" and "Secret" are displayed. The Client ID is a long string of characters, and the Secret is also a long string. A note states: "Note: When you generate a new secret, you still maintain the original secret. The maximum number of client secrets is two. A client secret is either in enabled or disabled state." Below the secret, there's a table for managing secrets, showing one entry with columns for Created, Secret, Copy to clipboard, Action, and three dots. A "Generate new secret" button is also present.

```

69
70 PAYPAL_MODE=sandbox
71 PAYPAL_SANDBOX_API_USERNAME=awetest8848_api1@gmail.com
72 PAYPAL_SANDBOX_API_PASSWORD=J8Y4YUW4UE8XDU7B
73 PAYPAL_SANDBOX_API_SECRET=ELabO8RHzF3HsyOm7ChFISY6pDftoe5m_H6M3IFgwc6A8OE3HIjK48iRnJUGIfJWrHyoewqhvwh3fn6P
74 PAYPAL_CURRENCY=USD
75 PAYPAL_SANDBOX_API_CERTIFICATE=
76

```

- Route and Blade code



The image shows a code editor interface with two tabs open:

- routes/web.php**: This tab displays Laravel route definitions. The routes include various methods for managing a shopping cart (index, store, update, clear, destroy) and handling PayPal payments (handle-payment, paymentCancel, paymentSuccess). The code uses the `Route::get`, `Route::post`, and `Route::delete` helpers to map URLs to controller actions.
- resources/views/cart.blade.php**: This tab displays Blade template code. It contains logic for displaying items in a cart, a form for proceeding to checkout, and a button to pay the total amount. The template uses `@foreach` loops, `<form>` tags, and `` links.

- When clicked on Pay button, faced with an error

Your Cart

← Continue Shopping

Grand Total: \$ 27.3
You have 1 items in your cart

 qui \$ 27.3 

STACK CONTEXT DEBUG SHARE DOCS ⚙

Error PHP 8.0.24 9.45.1

Class "Srmklive\PayPal\Services\ExpressCheckout" not found

C:\Users\nitin\OneDrive\Documents\laravel front end\paypal documentation\app\Http\Controllers\PayPalPaymentController.php :25

```
10     $product['items'] = [
11         [
12             'name' => 'Nike Joyride 2',
13             'price' => 112,
14             'desc' => 'Running shoes for Men',
15             'qty' => 2
16         ]
17     ];
18
19     $product['invoice_id'] = 1;
20     $product['invoice_description'] = "Order #{$product['invoice_id']} Bill";
21     $product['return_url'] = route('success.payment');
22     $product['cancel_url'] = route('cancel.payment');
23     $product['total'] = 224;
24
25     $paypalModule = new ExpressCheckout;
26
27     $res = $paypalModule->setExpressCheckout($product);
28     $res = $paypalModule->setExpressCheckout($product, true);
```

As a result, instead of integrating PayPal on checkout, I implemented a simple proceed to checkout functionality where after the checkout button is clicked, user is redirected to a new page that displays ‘Your order has been placed’ and shows their grand total amount. After which user can go back to the homepage of the site and the cart gets cleared (just like I have shown above/ right above MailChimp section).

The web app is fully responsive as well, which I will be showcasing on the demo. Thank You.

END