



# LARAVEL 11+ DOCUMENT

BY ZAYNOUN JAMAL

ZAYNOUN

MARCH 19, 2025

ZAYNOUN JAMAL  
COMPUTER SCIENCE

[GitHub](#)

[LinkedIn](#)

0 | Page

## CREATING LARAVEL PROJECT FROM A TO Z

Author: Zaynoun Jamal

MAJOR: COMPUTER SCIENCE

### LARAVEL 11 BY ZAYNOUN JAMAL

#### 1. LARAVEL PROJECT CREATION

- Run in the CMD (Command Prompt): laravel new project-name
- Run in the CMD (Command Prompt): cd project-name
- Open the IDE(VSCODE) open the .env file and set the following lines to:

```
- DB_CONNECTION=mysql (your database type)  
DB_HOST=127.0.0.1  
DB_PORT=3306  
DB_DATABASE=royal_cedar_scents_database(your database name)  
DB_USERNAME=root  
DB_PASSWORD=
```

- Open the terminal in IDE (VSCODE for example):
  - run the following command: composer require laravel/Jetstream
  - run also: php artisan Jetstream:intall livewire
- Open the GIT bash: Run the following commands
  - npm install
  - npm run dev
- Open the IDE again:
  - Make Changes in the database/migration/user-table
  - php artisan migrate
- Now your ready to go...

#### Lets Somehow recreate the Royal-Cedar-Scents Project

#### 2. ROYAL CEDAR SCENTS

##### 2.1 FROM A TO Z

### Registration

- Open the resources folder: resources/auth/register.blade.php

- Add inputs as follow:

```
- <div class="mt-4">
    <x-label for="address" value="{{ __('Address') }}" />
    <x-input id="address" class="block mt-1 w-full"
type="text" name="address" :value="old('address')" required
autocomplete="username" />
</div>
<div class="mt-4">
    <x-label for="phone" value="{{ __('Phone') }}" />
    <x-input id="phone" class="block mt-1 w-full"
type="number" name="phone" :value="old('phone')" required
autocomplete="username" />
</div>
```

- Open the app/models/User.php file, add the added inputs to the fillable(in order) in our case:

```
- protected $fillable = [
    'name',
    'email',
    'address',
    'phone',
    'password',
];
```

- Open the app/actions/fortify/CreateNewUser.php file,add the added fields :

```
- return User::create([
    'name' => $input['name'],
    'email' => $input['email'],
    'phone' => $input['phone'],
    'address' => $input['address'],
    'password' => Hash::make($input['password']),
]);
```

## Redirection in case many user-type

- Open the config/fortify.php file: (in case many user types)
  - change the line : 'home' => '/dashboard' to 'home' => '/redirect'
- Go to the Route/web.php file: define redirect route:
  - First: use App\Http\Controllers\HomeController;
  - Second define the route: route::get('/redirect',[HomeController::class,'redirect']);
- I now an error will appear don't panic:
  - Run in the terminal: php artisan make:controller HomeController

- Open the app\http\Controller\HomeController.php:

```
- public function redirect()
{
    $usertype = Auth::user()->usertype; /*taking the usertype from the
database*/
    if ($usertype == "admin") {
        return view('admin.home'); // we will return to this later
    } else {
        return view('home.userpage'); // we will return to this later
    }
}
```

- Whenever you want to use User (Model), write at the head of the file

```
-use App\Models\User;
```

## Logout in Laravel

- Build-in logout option in Laravel:  
-go to the dashboard.blade.php : (directly in the views folder)  
Take only:
- ```
<x-app-layout>
</x-app-layout>
```
- To remove the background color from the edited dashboard.blade.php :  
-go to the app.blade.php : (in the layouts folder)  
Remove the classes for line 24/or 23
  - To remove the Logo and Navigation Links from dashboard.blade.php :  
-go the navigation.blade.php remove the code for the <!—Navigation Links → and  
←!—Logo →
  - Go to the welcome.blade.php:  
-on line 26 after @Auth, remove /dashboard replace it with /redirect
  - Keep in mind any addition of css and javascript files will be added in the public folder.

## Changing in the web.php and Creating index function

- Go to web.php file (as we know in the Routes folder):

- remove the Route::get('/....);
- write:  
Route::get('/',[HomeController::class,'index']);//the index function is in the HomeController.php
- In the HomeController:  
-public function index(){  
    return view('home.userpage');//userpage.blade.php in the home folder  
}
- An error will occur when finishing the last step but don't panic:  
-in the resources folder create another folder home.  
-add in it userpage.blade.php file (create it)

## Creating userpage.blade.php

- Now we will suppose we have an html template (user side) or html codes that we have on our desktops: in this template we suppose we have header, slider section, why shop with us section, new arrival section, product section, subscribe section, footer section, and copyright section.(You can as many sections as you want)
- In this case the html code is so big, and is not clear and readable (also if you have written the code by yourself with comments it wont be clear), to let everything clear we take each section I mentioned before in the last point,(also if others do the same) , in we put them in separate blade.php files.
- We will have header.blade.php, slider.blade.php, slider.blade.php, why.blade.php,new.blade.php,new.blade.php,product.blade.php,subscribe.blade.php,footer.blade.php,copy.blade.php...
- In this case we separated the codes to let the code more understandable to us, and easy to be changed
- In the home/userpage.blade.php (that we created)  
-to include this section we use @include().  
-the css,js folders is in the home folder which is in the css,js folders as we mentioned before  
-this is the userpage.blade.php

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no" />
    <meta name="keywords" content="" />
    <meta name="description" content="" />
```

```

<meta name="author" content="" />
<link rel="shortcut icon" href="images/favicon.png" type="">
<title>Royal Cedar Scents</title>
<link rel="stylesheet" type="text/css" href="home/css/bootstrap.css">
/>
<link href="home/css/font-awesome.min.css" rel="stylesheet" />
<link href="home/css/style.css" rel="stylesheet" />
<link href="home/css/responsive.css" rel="stylesheet" />
</head>

<body>
<div class="hero_area">
    @include('home.header')// we included the header.blade.php
    @include('home.slider')// we included the slider.blade.php
</div>
@include('home.why') // we included the why.blade.php
@include('home.product') // we included the product.blade.php
@include('home.subscribe') // we included the subscribe.blade.php
@include('home.footer') // we included the footer.blade.php
@include('home.copy') // we included the copy.blade.php
<script src="home/js/jquery-3.4.1.min.js"></script>
<script src="home/js/popper.min.js"></script>
<script src="home/js/bootstrap.js"></script>
<script src="home/js/custom.js"></script>
<script>
    document.addEventListener("DOMContentLoaded", function(event) {
        var scrollpos = localStorage.getItem('scrollpos');
        if (scrollpos) window.scrollTo(0, scrollpos);
    });

    window.onbeforeunload = function() {
        localStorage.setItem('scrollpos', window.scrollY);
    };
</script>
</body>

</html>

```

## Build ins in Laravel

- Note in Laravel we can use bootstrap classes for buttons and other such as for login register , " btn btn-success" "btn btn-primary"

- Build in routes in Laravel:

-route('login') , route('register')

- example:

<a href="{{ route('login') }}>Login </a> // when clicking it will go to

login.blade.php build in blade.php file

<a href="{{ route('register') }}>Register</a> // when clicking it will go to

register.blade.php build in blade.php file

## Handling Login/Register and Logout

- Now to handle to show the login and register to show only in case not logged-in, and to show log out if logged in:

```
@if (Route::has('login'))
    @auth // in case logged in
        //show the log-out button
        <a href="{{ route('logout') }}>Logout</a>
    @else //if not logged-in show the login and register button
        <a href="{{ route('login') }}>Login </a>
        <a href="{{ route('register') }}>Register</a>
    @endauth
@endif
```

- You can integrate the logout button in the html (blade.php in Laravel) use the dashboard.blade.php that we edited <x-app-layout></x-app-layout>  
But there is css problems, to fix it go to the navigation.blade.php remove the classes from the first line, third line, and fourth line.

```
@if (Route::has('login'))
    @auth // in case logged in
        //show the log-out button
        <x-app-layout></x-app-layout>
    @else //if not logged-in show the login and register button
        <a href="{{ route('login') }}>Login </a>
        <a href="{{ route('register') }}>Register</a>
    @endauth
@endif
```

- Third way for logout in laravel, in a form of a method="post" and action="{{ route('logout') }}" with @csrf inside the form // <form>@csrf </form>

## Build in profile in Laravel

- To show build in profile in Laravel

-<a href="{{ route('profile.show') }}> Show Profile</a>

## Including admin side blades.php

- Integrating/Including admin template/side in Laravel: we do the same steps that we did for userpage.blade.php
- php artisan make:controller AdminController (in the terminal)
- make a folder named admin in the resources , and make blade.php file for each section as we did for the user side.(also for the css and javascript if you want, the common ones)
- In the admin folder: [home.blade.php](#)

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
    <title>Admin</title>
    @include('admin.css')
  </head>
  <body>
    <div class="container-scroller">
      <!-- partial:partials/_sidebar.html -->
      @include('admin.sidebar')
      <!-- partial -->
      @include('admin.header')
      <!-- partial -->
      @include('admin.body')
      <!-- container-scroller -->
      <!-- plugins:js -->
      @include('admin.scripts')
    </body>
  </html>
```

- In the admin/sidebar.blade.php file:  
We define a route/url view\_category, when clicking on it (Category) will take us to another page

```
• <li class="nav-item menu-items">
•   <a class="nav-link" href="{{ url('view_category') }}">
•     <span class="menu-icon">
•       <i class="mdi mdi-playlist-play"></i>
•     </span>
•     <span class="menu-title">Category</span>
•   </a>
```

- </li>
- Define the route in the web.php:  
Route::get('/view\_category',[AdminController::class,'view\_category']);
- In the AdminController:

```
• Use App\Models\Category;
• public function view_category()
• {
•     $data = Category::all(); //get all Categories from Category Model
•     return view('admin.category', compact('data'));
• }
```

- Two errors will show on the screen let fix it one by one:
  - php artisan make:model Category -m//run this in the terminal this will make model with the migration table
  - make changes in the category migration table in the database folder/migration/category

```
public function up(): void
{
    Schema::create('categories', function (Blueprint $table) {
        $table->id();
        $table->string('name');// add this
        $table->timestamps();
    });
}
```

- Run the command migrate in the terminal to migrate the table in the database:
  - php artisan migrate
  - first error is fixed
- Now the second error, create category.blade.php in the admin folder in resources, and the second error is fixed
- In the category.blade.php (admin side)
  - showing all categories in the blade.php
  - why @forelse[]?

Better from @foreach() in handling in case no \$data,no categories available in the database table categories,@empty handles in case \$data is empty in @forelse[].

```
-<h2>All Categories</h2>

<table>
<thead>
<tr>
<th>Category</th>
<th>Action</th>
</tr>
</thead>
```

```

<tbody>
@forelse($data as $category)//$data that we send it with the view in the
AdminController
<tr>

<td>{{ $category->name }}</td>
</tr>

@empty
<tr>
<td colspan="2">No categories found.</td>
</tr>
@endforelse
</tbody>

```

- Go now to sidebar.blade.php add this:

- <li class="nav-item menu-items">
 <a class="nav-link" href="{{ url('view\_add\_category') }}>
 <span class="menu-icon">
 <i class="mdi mdi-playlist-play"></i>
 </span>
 <span class="menu-title">Add Category</span>
 </a>
 </li>

- Handle the route in the web.php:

```
route::get('/view_add_category',[AdminController::class,'view_add_category']);
```

- In the AdminController:

- public function view\_add\_category()
 {
 return view('admin.view\_add\_category');
 }

- Create the view\_add\_category.blade.php in the admin (resources):
 -create a form in it, with method post for adding category

- <h2>Add Category</h2>
 <form action="{{url('add\_category')}}" method="POST">
 @csrf // if removed error will occur
 <label for="category">Category Name:</label>
 <input type="text" id="category" style="color: gold;" name="category"
 placeholder="Enter category name" required>
 <input type="submit" value="Add Category" name="submit" class="gold-
 button" />
 </form>

- Lets handle the route in the web.php for adding a category:
- `route::post('/add_category',[AdminController::class,'add_category']);`
- In the AdminController:
- ```
public function add_category(Request $request)
{
    $category = new Category;// creating new instance
    $category->name = $request->category;// taking the input from request
    $category->save();// saving it in database
    return redirect()->back()->with('message', 'Category Added
Successfully');//message send with session
}
```
- How to show the message that is sent in blade.php:
- ```
@if(session()->has('message'))
<div class="alert alert-success" role="alert">
<button type="button" class="close" data-dismiss="alert" aria-
hidden="true">X</button>
{{ session('message') }}
</div>
```
- Deleting a Category (two accepted methods delete/get, recommended get)
  - in the category.blade.php add this in to the table we created:  
We send the category id which is unique with the route.  
You can handle the onclick. Im not taking into consid

```
<td>
<a href="{{ url('delete_category', $category->id) }}" class="gold-button
delete-btn"> Delete </a></td>
```

- In the web.php:
- `route::get('/delete_category/{id}',[AdminController::class,'delete_category']);// the sent id is binded here {id}`
- In the AdminController:
- ```
public function delete_category($id)//binded id as a parameter
{
    Category::find($id)->delete();// find the category by id then delete it
    return redirect()->back()->with('message', 'Category Deleted
Successfully');// message sent in session
}
```
- Now lets create the Product and its database migration:
  - php artisan make:model Product -m
- In the product migration: (now we definitely its location)

```
- public function up(): void
{
```

```

Schema::create('products', function (Blueprint $table) {
    $table->id();
    $table->string('name')->nullable(); //add this
    $table->string('volume')->nullable(); //add this
    $table->string(column: 'description')->nullable(); //add this
    $table->double(column: 'price')->nullable(); //add this
    $table->integer('discount_price')->nullable(); //add this
    $table->integer(column: 'stock')->nullable(); //add this
    $table->string(column: 'product_category')->nullable(); //add this
    $table->string('image_url')->nullable(); //add this
    $table->timestamps();
});
}

```

- php artisan migrate // the tables migrate
- In case a you need to change in a migrated table you cant unless you write this command in terminal:(data will be lost in the table/s)
   
php artisan migrate:rollback
   
now change what you want in the product table and re-migrate
- in the sidebar.blade.php (admin side)

```

-<li class="nav-item menu-items">
    <a class="nav-link" href="{{ url('/view_add_product') }}">
        <span class="menu-icon">
            <i class="mdi mdi-playlist-play"></i>
        </span>
        <span class="menu-title">Add Products</span>
    </a>
-</li>

```

- In the web.php define the route:

```

$route::get('/view_add_product',[AdminController::class,'view_add_product']);
);

```

- In the AdminController

```

-public function view_add_product()
{
    $categories = Category::all();
    return view('admin.addproduct', compact('categories'));
}

```

- Create addproduct.blade.php in the admin folder in the resources folder as usual.
- In addproduct.blade.php:
   
Create form to add product with method post.

```
<!-- Add Product Form -->
<div class="container">
<h1>Add New Product</h1>
<form action="{{ url('/add_product') }}" method="POST"
enctype="multipart/form-data">
@csrf
<div class="form-group">
<label for="product_name">Product Name</label>
<input type="text" id="product_name" name="product_name" required>
</div>
<div class="form-group">
<label for="description">Description</label>
<textarea id="description" name="description" rows="4"
required></textarea>
</div>
<div class="form-group">
<label for="category">Category</label>
<select id="category" name="category" required>
<option value="">Select Category</option>
@foreach ($categories as $category )
<option value="{{ $category->name }}>{{ $category->name }}</option>
@endforeach
</select>
</div>
<div class="form-group">
<label for="price">Price</label>
<input type="number" id="price" name="price" required>
</div>
<div class="form-group">
<label for="discount_price">Discount Price</label>
<input type="number" id="discount_price" name="discount_price">
</div>
<div class="form-group">
<label for="volume">Volume (ml)</label>
<input type="number" id="volume" name="volume" required>
</div>
<div class="form-group">
<label for="stock">Stock</label>
<input type="number" id="stock" name="stock" required>
</div>
<div class="form-group">
<label for="product_image">Product Image</label>
<input type="file" id="product_image" name="product_image"
accept="image/*" required>
</div>
```

```
<input type="submit" value="Add Product" class="btn btn-primary button"
href="" />
</form>
</div>
```

- In the web.php:

```
-route::post('/add_product',[AdminController::class,'add_product']);
```

- In the AdminController:

```
-public function add_product(Request $request)
{//Taking the data from the user request from the input name
$product = new Product;
$product->name = $request->product_name;
$product->price = $request->price;
$product->volume = $request->volume;
$product->discount_price = $request->discount_price;
$product->description = $request->description;
$product->product_category = $request->category;
$product->stock = $request->stock;
$image = $request->product_image;
//handle the image always like this
$imagename = time() . '.' . $image->getClientOriginalExtension();
$request->product_image->move('product', $imagename);//create folder
product in public folder, all product images will be stored there
$product->image_url = $imagename; //store it in the database
$product->save();// save product in the database table
return redirect()->back()->with('message', 'Product Added Successfully');
}
```

- As I mentioned before create product folder in the public folder to store product images
- In the sidebar.blade.php (admin side) add this: Showing products in a view

```
-<li class="nav-item menu-items">
<a class="nav-link" href="{{ url('/view_products') }}">
<span class="menu-icon">
<i class="mdi mdi-playlist-play"></i>
</span>
<span class="menu-title">Show Products</span>
</a>
</li>
```

- In the web.php:

```
-route::get('/view_products',[AdminController::class,'view_product'])-
>name('admin.products');
```

- In the AdminController:

```
use App\Models\Product;
public function view_product()

{
    $data = Product::all();
    return view('admin.showproduct', compact('data'));
}
```

- In showproduct.blade.php (after creation in the admin folder)

```
<h1>Manage Products</h1>

<table>
<thead>
<tr>
<th>Product Name</th>
<th>Category</th>
<th>Stock</th>
<th>Volume</th>
<th>Price</th>
<th>Discount Price</th>
<th>Description</th>
<th>Image</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
@forelse($data as $product)// data is the sent parameter from controller
<tr>
<td>{{ $product->name }}</td>
<td>{{ $product->product_category }}</td>
<td>{{ $product->stock }}</td>
<td>{{ $product->volume }} ml</td>
<td>{{ $product->price }}</td>
@if($product->discount_price != null)
<td>{{ $product->discount_price }}</td>
@else
<td>N/A</td>
@endif
<td>{{ $product->description }}</td>
//we put the product folder and the url of the image in the database
<td></td>// the image will appear with no problems
<td class="action-btns">
```

```

<a href="{{ url('view_edit_product', $product->id) }}" class="btn-edit">Edit</a> // will explain later
</td>

<td class="action-btns">
// handled the onclick with javascript
<a href="javascript:void(0); " class="btn-delete"
onclick="confirmDelete('{{ $product->id }})">Delete</a>
</td>
</tr>
@empty
<tr>
<td colspan="8">No products found.</td>
</tr>
@endforelse
</tbody>
</table>

```

- Delete Product in Admin-Side:

```

<a href="{{url('/delete_product', $product->id)}}" class="btn-delete">Delete</a>

```

- In the web.php: using get method for delete recommended better than delete

```

$route::get('/delete_product/{id}', [AdminController::class, 'delete_product']);

```

- In the AdminController: handling the deletion in the controller

```

public function delete_product($id)
{
Product::find($id)->delete();
return redirect()->back()->with('message', 'Product Deleted Successfully');
}

```

- Even though Our topic its not about css: but sometimes an error or css doesn't work on a view it may be you want to add in html head
- Now for editing product: we send with the unique product id
[Edit]({{ url('view_edit_product', $product->id) }})
- In the web.php: to get the view (blade.php)
- `$route::get('/view_edit_product/{id}', [AdminController::class, 'view_edit_product']);`
- In the AdminController:

- `public function view_edit_product($id)`
- `{`
- `$product = Product::find($id); //find the product by the sent id`
- `$categories = Category::all();`
- `return view('admin.editproduct', compact('product', 'categories'));`
- `}`
- In the editproduct.blade.php:  
-we fetch the product sent by the controller.

```

• <h1>Edit Product</h1>
• // because we are updating the value not adding we send id with the
route
• <form action="{{ url('/update_product', $product->id) }}" method="POST"
enctype="multipart/form-data"> // enctype for images
• @csrf // don't remove!
• <div class="form-group">
• <label>Product Name</label>
• <input type="text" name="name" value="{{ $product->name }}" required>
• </div>
• <div class="form-group">
• <label>Category</label>
• <select name="category" required>
• <option value="{{ $product->product_category }}" selected>{{ $product-
>product_category }}</option>
• </option>
• @foreach ($categories as $category)
• <option value="{{ $category->name }}>{{ $category->name }}</option>
• @endforeach
• </select>
• </div>
• <div class="form-group">
• <label>Stock</label>
• <input type="number" name="stock" value="{{ $product->stock }}" required>
• </div>
• <div class="form-group">
• <label>Volume (ml)</label>
• <input type="number" name="volume" value="{{ $product->volume }}" required>
• </div>
• <div class="form-group">
• <label>Price</label>
• <input type="number" name="price" value="{{ $product->price }}" required>
• </div>
• <div class="form-group">

```

- <label>Discount Price</label>
- <input type="number" name="discount\_price" value="{{ \$product->discount\_price }}">
- </div>
- <div class="form-group">
- <label>Description</label>
- <textarea name="description" required>{{ \$product->description }}</textarea>
- </div>
- <div class="form-group">
- <label for="current-image">Current Image</label>
- 
- </div>
- <div class="form-group">
- <label>Change Product Image</label>
- <input type="file" name="image">
- </div>
- <button type="submit">Update Product</button>
- </form>
- To handle the product edit/update  
-In the web.php:

```
route::post('/update_product/{id}', [AdminController::class, 'update_product']);
```

- In the AdminController:

```
- public function update_product(Request $request, $id)

{
$product = Product::find($id);
$product->name = $request->name;
$product->price = $request->price;
$product->volume = $request->volume;
$product->discount_price = $request->discount_price;
$product->description = $request->description;
$product->product_category = $request->category;
$product->stock = $request->stock;
$image = $request->image;
//here we check if we added the image and handle to change it because its
//not required input or field, if added we change the image_url if not
//added keep the old image
if ($image) {
$imagename = time() . '.' . $image->getClientOriginalExtension();
$request->image->move('product', $imagename);
```

```

$product->image_url = $imagename;
}
$product->save();
return redirect()->route('admin.products')->with('message', 'Product
Updated Successfully');
}

```

- Example for redirecting to specific route:
  - in old tasks, we are always redirecting back not to specific route
  - to redirect to specific route:

```

- return redirect()->route('admin.products')->with('message', 'Product
Updated Successfully');

```

-name of the route in the web.php should be admin.products

```

- route::get('/view_products',[AdminController::class,'view_product'])-
>name('admin.products');

```

## User Side blades.php

- Now to show products in the User side :
  - Go to the home/product.blade.php (in resources)
  - Make a unique/single card for the product (with your own style and css)
  - Go to the HomeController to send all products to the redirect function and also for index function // index function its when the user is not logged in no redirection
- In Home Controller:redirect() function

```

public function redirect()
{
    $usertype = Auth::user()->usertype;// take user type from database
    //if usertype is admin redirect to home.blade.php (admin side)
    if ($usertype == "admin") {
        return view('admin.home');
    } else {
        //if usertype is user or other undefined type
        //get all products in pagination of 3, like each time 3 products
        $products = Product::paginate(3);
        return view('home.userpage', compact('products'));
    }
}

```

- In HomeController index() function:

```

• public function index()
{
    //get all products in pagination of 3, like each time 3 products
    $products = Product::paginate(3);

    return view('home.userpage', compact('products'));
}

```

- We are compacting (sending) products to the userpage.blade.php (user side), which includes the product.blade.php, then the products is also compacted to product.blade.php
- In product.blade.php: after making a unique card for product -make @foreach() or @forelse(), we know the difference.

```

-<section class="product_section layout_padding">

<div class="container">
<div class="heading_container heading_center">
<h2>
Our <span>products</span>
</h2>
</div>
<div class="row">
//the compacted products using foreach
@foreach ($products as $product)
<div class="col-sm-6 col-md-4 col-lg-4">
<div class="box">
<div class="option_container">
<div class="options">
//the add to cart will be explained soon in the document
<form method="POST" action="{{ url('add_product_to_cart', $product->id) }}>
@csrf
<button type="submit" class="option1">
Add to Cart
</button>
</form>
//this will be explained also soon
<a href="{{ url('product_details', $product->id) }}" class="option2">
Product Details
</a>
</div>
</div>
<div class="img-box">
//showing the image of a product

```

```


</div>
<div class="detail-box">
//showing the name of a product
<h5 class="product-name">{{ $product->name }}</h5>
//showing the volume of a product
<h6 class="product-volume">{{ $product->volume }}ml</h6>
//checking if product's discount_price is not null
@if ($product->discount_price != null)
<div class="price-container">
<span class="label">From</span>
<div class="old-price-container">
//showing the old price which is product's price
<span class="old-price">{{ $product->price }}$</span>
</div>
<span class="label">To</span>
//showing the product discount_price as the new price
<span class="discount-price">{{ $product->discount_price }}$</span>
</div>
//if product discount price is null
@else
<div class="price-container">
<span class="label">Only For:</span>
//showing the product price only
<span class="regular-price" style="color:gold;">{{ $product->price }}$</span>
</div>
@endif
</div>
</div>
</div>
@endforeach
//using pagination bootstrap because the products is paginated in the
//controller you can style the pagination with css or bootstrap
<!-- Pagination -->
<div class="pagination-container">
{!! $products->withQueryString()->links('pagination::bootstrap-5') !!}
</div>

</div>
<div class="btn-box">
<a href="">
View All products
</a>

```

```
</div>
</div>
</section>
```

- To See Product Details:

```
-// we send the unique Id to the route
<a href="{{ url('product_details', $product->id) }}" class="option2">
Product Details
</a>
```

- In the web.php:
  - now the binding logic is clarified to us

```
route::get('/product_details/{id}', [HomeController::class, 'view_product_details']);
```

- In the HomeController:

```
• public function view_product_details($id)
• {
•   //find the product by the send id then send it to the view
•   $product = Product::find($id);
•   return view('home.productdetails', compact('product'));
• }
```

- Create the productdetails.blade.php in the home folder.
- In productdetails.blade.php:

```
• <!DOCTYPE html>
• <html>
•
•   <head>
•     <title>Perfume Product Page</title>
•     <base href="/public">
•     <link rel="stylesheet" href="productdetails/perfume-productpage.css"
type="text/css" />
•     <script src="https://kit.fontawesome.com/a193cceeae.js"
crossorigin="anonymous"></script>
•     <link rel="preconnect" href="https://fonts.googleapis.com" />
•     <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />
•     <link
      href="https://fonts.googleapis.com/css2?family=Gloock&display=swap"
      rel="stylesheet" />
•     <meta name="viewport" content="width=device-width, initial-scale=1" />
•     <link rel="shortcut icon" href="images/favicon.png" type="">
•     <!-- bootstrap core css -->
•     <link rel="stylesheet" type="text/css" href="home/css/bootstrap.css" />
```

```

• <!-- font awesome style -->
• <link href="home/css/font-awesome.min.css" rel="stylesheet" />
• <!-- Custom styles for this template -->
• <link href="home/css/style.css" rel="stylesheet" />
• <!-- responsive style -->
• <link href="home/css/responsive.css" rel="stylesheet" />
• </head>
•
• <body>
• <div class="hero_area">
• @include('home.header')
• <!-- Main Content -->
• <main>
• <section style="margin-top: 0; margin-bottom: 190px;">
• <div id="bod">
• <div id="div1">
• <!-- Use an <img> tag to display the product image -->
• //showing the product image (as explained before)
• name }}""
• style="width: 100%; height: 100%;">
• </div>
• <div id="div2">
• <p id="perfume">PERFUME</p>
• //showing the product name
• <h1>{{ $product->name }}</h1>
• //showing the product description
• <p>{{ $product->description }}</p>
•
• <!-- Added category line -->
• <hr style="border: 0.5px solid gold; margin: 10px 0;">
• //showing product category
• <p id="category">Category: <span>{{ $product->product_category
• }}</span></p>
• //showing the product volume
• <p id="volume">Volume: <span>{{ $product->volume }} ml</span></p>
•
• <!-- Added stock info -->
• //showing product stock
• <p id="stock">Stock: <span>{{ $product->stock }} left</span></p>
• //checking if product discount_price is not equal to null
• @if ($product->discount_price != null)
• <!-- Adjusted price alignment -->
• <div id="price">
• //show product discount price

```

```

• <h1 style="display: inline-block; margin-right: 10px;">${{ $product->discount_price }}</h1>
• //showing product old price which is price
• <p
• style="display: inline-block; text-decoration: line-through; color: red; font-weight: bold;">
• ${{ $product->price }}</p>
• </div>
• //if product discount_price is equal to null
• @else
• <div id="price">
• //showing product price
• <h1 style="display: inline-block; margin-right: 10px;">${{ $product->price }}</h1>
• </div>
• @endif
• //will be explained later
• <form method="post" action="{{ url('add_product_details_to_cart', $product->id) }}>
• @csrf
•
• <!-- Quantity Selector -->
• <div id="quantity-container">
• <button type="button" onclick="changeQuantity(-1)">-</button>
• <input type="text" id="quantity" name="quantity" value="1" readonly />
• <button type="button" onclick="changeQuantity(1)">+</button>
• </div>
•
• <button type="submit" name="submit">
• <i id="btn" class="fa fa-shopping-cart" aria-hidden="true"></i> Add to Cart
• </button>
• </form>
• </div>
• </div>
• </section>
• </main>
•
• <!-- Footer -->
• @include('home.footer')
• @include('home.copy')
•
• <!-- jQery -->
• <script src="home/js/jquery-3.4.1.min.js"></script>

```

```

• <!-- popper js -->
• <script src="home/js/popper.min.js"></script>
• <!-- bootstrap js -->
• <script src="home/js/bootstrap.js"></script>
• <!-- custom js -->
• <script src="home/js/custom.js"></script>
• <script>
• function changeQuantity(amount) {
•   let quantityInput = document.getElementById("quantity");
•   let currentValue = parseInt(quantityInput.value);
•   if (currentValue + amount >= 1) {
•     quantityInput.value = currentValue + amount;
•   }
• }
• </script>
• </body>
•
•
• </html>
•

```

- Go to the header.blade.php (user side)

- Give for the home and logo the following href

```

- <li class="nav-item active">
- <a class="nav-link" href="{{ url('/') }}>Home <span class="sr-
only">(current)</span></a>
- </li>
- For the logo:
- <a class="navbar-brand" href="{{ url('/') }}></a>

```

- Now for Cart Section/Page:

- Make in the header.blade.php a button my cart with href="{{url('view\_cart')}}"

```

- <li class="nav-item">
- <a class="nav-link" href="{{ url('view_cart') }}>Cart</a>
- </li>

```

- Create the cart.blade.php in the home folder

- The web.php:

```

-route::get('/view_cart',[HomeController::class,'view_cart'])-
>name('cart');

```

- In the HomeController:

```

- public function view_cart()

{
//if user is logged in
if (Auth::id()) {
$id = Auth::user()->id;
$user = Auth::user();
//will be explained later
$carts = Cart::where('user_id', '=', $id)->get();
return view('home.cart', compact('carts', 'user'));
}
//if user is not logged navigate or redirect to login page
else {
return redirect()->route('login')->with('message', 'Please login to view
cart');
}
}

```

- Add Product to Cart:

-First method:

- <form method="POST" action="{{ url('add\_product\_to\_cart', \$product->id)
}}>
- @csrf
- <button type="submit" class="option1">
- Add to Cart
- </button>

-Second method:

- <a href="{{url('add\_product\_to\_cart', \$product->id)}}>Add to Cart</a>
- In the web.php:

```

-Route::post('/add_product_to_cart/{id}', [HomeController::class,
'add_product_to_cart']);

```

- Now in HomeController:

-If not logged in -> navigate to login page when clicking on add to cart button

```

• Public function add_to_cart($id){
• If(Auth::id()){
• //cart logic will be written soon
• }else{
• return redirect('login');
• }
• }

```

- Now Create Cart Model and migration table:

-php artisan make:model Cart -m

- In Cart migration table:

```
-public function up(): void
{
    Schema::create('carts', function (Blueprint $table) {
        $table->id();
        $table->string('name')->nullable(); //add this
        $table->string('email')->nullable(); //add this
        $table->string('product_category')->nullable(); //add this
        $table->string('phone')->nullable(); //add this
        $table->string('address')->nullable(); //add this
        $table->string('product_title')->nullable(); //add this
        $table->integer('product_quantity')->nullable(); //add this
        $table->double('product_price')->nullable(); //add this
        $table->string('image')->nullable(); //add this
        $table->string('product_id')->nullable(); //add this
        $table->string('user_id')->nullable(); //add this
        $table->timestamps();
    });
}
```

- php artisan migrate
- In the HomeController:

```
- public function add_product_to_cart(Request $request, $id)
{
//if logged in
if (Auth::check()) {
$user = Auth::user();
$product = Product::find($id);
//if product not found in database
if (!$product) {
return redirect()->back()->with('error', 'Product not found.');
}

// Check if product exists in cart using id
$product_exist_in_cart = Cart::where('product_id', $id)
->where('user_id', $user->id)
->first(); // Use first() instead of get('id')->first()
//if yes
if ($product_exist_in_cart) {
//  Correctly retrieve the cart item and update the quantity
// and stock greater or equal than 1
if ($product->stock >= 1) {
//add quantity in cart
$product_exist_in_cart->product_quantity += 1;
}}
```

```

// decrement the stock by 1
$product->stock = $product->stock - 1;
$product->save();
}
//if stock is less than 1 which means 0
else {
return redirect()->back()->with('error', 'Not enough quantity in stock');
}
//checking if discount is not null
if ($product->discount != null) {
//the price logic
$product_exist_in_cart->product_price = $product->discount *
$product_exist_in_cart->product_quantity;
} else {
// the price logic if no discount price
$product_exist_in_cart->product_price = $product->price *
$product_exist_in_cart->product_quantity;
}
// save in database
$product_exist_in_cart->save();

return redirect()->back()->with('success', 'Product quantity increased in
cart.');
}
//if new item in cart, product not found in cart
else {
//  Create a new cart entry
$cart = new Cart;
//save the details
$cart->user_id = $user->id;
$cart->product_category = $product->product_category;
$cart->product_id = $product->id;
$cart->product_quantity = 1;
$cart->product_title = $product->name;
$cart->image = $product->image_url;
$cart->name = $user->name;
$cart->email = $user->email;
$cart->phone = $user->phone;
$cart->address = $user->address;

//  Save discounted price if available
$cart->product_price = $product->discount_price ?? $product->price;

$cart->save();
}

```

```

    return redirect()->back()->with('success', 'Product added to cart.');
}
}
//if not logged in
else {
return redirect()->route('login')->with('message', 'Please login to add
product to cart.');
}
}
}

```

- The previous logic if a single add to cart quantity ( quantity =1)
- Now if quantity is w.r.t the request:

In the productdetails.blade.php:

- <form method="post" action="{{ url('add\_product\_details\_to\_cart',
\$product->id) }}>
- @csrf
- 
- <!-- Quantity Selector -->
- <div id="quantity-container">
- <button type="button" onclick="changeQuantity(-1)">-</button>
- <input type="text" id="quantity" name="quantity" value="1" readonly />
- <button type="button" onclick="changeQuantity(1)">+</button>
- </div>
- 
- <button type="submit" name="submit">
- <i id="btn" class="fa fa-shopping-cart" aria-hidden="true"></i> Add to
Cart
- </button>
- </form>
- In the web.php:
- route::post('/add\_product\_details\_to\_cart/{id}', [
HomeController::class, 'add\_product\_details\_to\_cart']);
- In the HomeController:

```

- public function add_product_details_to_cart(Request $request, $id)
{
if (Auth::check()) { //  Use Auth::check() instead of Auth::id()
$user = Auth::user();
$product = Product::find($id); // find product by id
//if product not found in database
if (!$product) {
return redirect()->back()->with('error', 'Product not found.');
}
// Check if product exists in cart

```

```

$product_exist_in_cart = Cart::where('product_id', $id)
->where('user_id', $user->id)
->first(); // Use first() instead of get('id')->first()
// if product exist in cart
if ($product_exist_in_cart) {
    // if stock greater than quantity requested by user
    if ($product->stock >= $request->quantity) {
        //add product requested quantity to product quantity in cart
        $product_exist_in_cart->product_quantity += $request->quantity;
        // decrement the stock by the requested quantity
        $product->stock = $product->stock - $request->quantity;
        // save to database the product
        $product->save();
    }
    //if requested quantity greater than stock
    else {
        return redirect()->back()->with('error', 'Not enough quantity in stock');
    }
    //if product discount price not equal to null
    if ($product->discount != null) {
        // price logic
        $product_exist_in_cart->product_price = $product->discount *
        $product_exist_in_cart->product_quantity;
    }
    //if product discount price equal to null
    else {
        // price logic
        $product_exist_in_cart->product_price = $product->price *
        $product_exist_in_cart->product_quantity;
    }
    // save to database
    $product_exist_in_cart->save();

    return redirect()->back()->with('success', 'Product quantity increased in
cart.');
}
//if not found in Cart
else {
    $cart = new Cart;
    $cart->user_id = $user->id;
    $cart->product_id = $product->id;
    $cart->product_quantity = $request->quantity;
    $cart->product_title = $product->name;
    $cart->product_category = $product->product_category;
    $cart->image = $product->image_url;
}

```

```

$cart->name = $user->name;
$cart->email = $user->email;
$cart->phone = $user->phone;
$cart->address = $user->address;
//if product discount price not equal to null
if ($product->discount_price != null) {
    $cart->product_price = $product->discount_price * $request->quantity;
}
//if product discount price equal to null
else {
    $cart->product_price = $product->price * $request->quantity;
}
// save cart to database
$cart->save();
return redirect()->back()->with('success', 'Product added to cart.');
}
}
// if not logged in
else {
    return redirect()->route('login')->with('message', 'Please login to add
product to cart.');
}
}

```

- Build the Cart.blade.php the (html and css)
- Web.php:

```

-route::get('/view_cart',[HomeController::class,'view_cart'])-
>name('cart');

```

- In the HomeController:

```

public function view_cart()
{
    if (Auth::id()) {
        // take the id for the currently logged user
        $id = Auth::user()->id;
        $user = Auth::user();
        //get the cart for the specific user
        $carts = Cart::where('user_id', '=', $id)->get();
        return view('home.cart', compact('carts', 'user'));
    } else {
        return redirect()->route('login')->with('message', 'Please login to view
cart');
    }
}

```

```
}
```

- In the cart.blade.php:

```
-<!DOCTYPE html>

<html lang="en">

<head>
<base href="/public">
<!-- Basic -->
<meta charset="utf-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<!-- Mobile Metas -->
<meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no" />
<!-- Site Metas -->
<meta name="keywords" content="" />
<meta name="description" content="" />
<meta name="author" content="" />
<link rel="shortcut icon" href="images/favicon.png" type="" />
<title>Royal Cedar Scents</title>
<!-- bootstrap core css -->
<link rel="stylesheet" type="text/css" href="home/css/bootstrap.css" />
<meta name="csrf-token" content="{{ csrf_token() }}">
<!-- font awesome style -->
<link href="home/css/font-awesome.min.css" rel="stylesheet" />
<link href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min
.css" rel="stylesheet" />
<link href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet" />

<!-- Custom styles for this template -->
<link href="home/css/style.css" rel="stylesheet" />
<link href="home/cart/style.css" rel="stylesheet" />
<!-- responsive style -->
<link href="home/css/responsive.css" rel="stylesheet" />
<!-- Add CSS for Modal -->
<style>
/* Modal Styling */
.modal {
display: none;
position: fixed;
z-index: 1000;
```

```
left: 0;
top: 0;
width: 100%;
height: 100%;
background-color: rgba(0, 0, 0, 0.7);
justify-content: center;
align-items: center;
}

.modal-content {
background: black;
color: gold;
padding: 20px;
border-radius: 10px;
text-align: center;
width: 90%;
max-width: 400px;
}

.modal-buttons {
margin-top: 20px;
}

.confirm-btn {
background: gold;
color: black;
padding: 10px 15px;
border: none;
cursor: pointer;
border-radius: 5px;
}

.cancel-btn {
background: black;
color: gold;
padding: 10px 15px;
border: 1px solid gold;
cursor: pointer;
border-radius: 5px;
margin-left: 10px;
}

.confirm-btn:hover,
.cancel-btn:hover {
opacity: 0.8;
```

```

}
</style>
</head>

<body>
<div class="hero_area">
@include('home.header')
<div class="card">
<div class="row">
<div class="col-md-8 cart">
<div class="title">
<div class="row">
<div class="col">
// taking from the compacted user the name
<h4><b>{{ strtoupper($user->name) }}'s Shopping Cart</b></h4>
</div>
<div class="col align-self-center text-right text-muted">{{ $carts->count() }} items</div>
</div>
</div>
// declaring totalprice variable
<?php $totalprice = 0; ?>
@foreach ($carts as $cart)
<div class="test row border-top border-bottom">
<div class="row main align-items-center">
//showing the images
<div class="col-2"></div>
<div class="col">
// cart id
<input hidden class="cart-id" value="{{ $cart->id }}" />
// cart product title
<div class="row text-muted">{{ $cart->product_title }}</div>
//cart product category
<div class="row">{{ $cart->product_category }}</div>
</div>
<div class="col">
<div class="quantity-controls">
<a href="#" class="decrease changeQuantity">-</a>
//cart product quantity
<a class="border quantity-value">{{ $cart->product_quantity }}</a>
<a href="#" class="increase changeQuantity">+</a>
</div>
</div>
//cart each product price

```

```

<div class="col">$ {{\$cart->product_price}}
// delete cart product
<a href="#" class="close delete-btn"
data-delete-url="{{ url('delete_product_cart', \$cart->id) }}">&#10005;</a>
</div>
</div>
</div>
//calculating total price
<?php \$totalprice = \$totalprice + \$cart->product_price?>
@endforeach
<div class="back-to-shop"><a href="{{ url('/') }}">&leftarrow;</a><span
class="text-muted">Back
to shop</span></div>
</div>
<div class="col-md-4 summary">
<div>
<h5><b>Summary</b></h5>
</div>
<hr>
<div class="row">
// number of carts unique items
<div class="col" style="padding-left:0;">ITEMS {{ \$carts->count() }}</div>
//showing the totalprice
<div class="col text-right">$ <?php echo \$totalprice; ?></div>
</div>
<form>
<p>SHIPPING</p>
<select>
<option class="text-muted">Standard-Delivery- \$4.00</option>
</select>
<!--<p>GIVE CODE</p>
<input id="code" placeholder="Enter your code">-->
</form>
<div class="row" style="border-top: 1px solid rgba(0,0,0,.1); padding: 2vh
0;">
<div class="col">TOTAL PRICE</div>
//showing the total price with delivery
<div class="col text-right">$ <?php echo \$totalprice + 4; ?></div>
</div>
//will be explained soon
<form method="post" action="{{url('checkout')}}">
@csrf
<button class="btn">CHECKOUT</button>
</form>
</div>

```

```
</div>

</div>
<!-- Custom Delete Confirmation Modal --&gt;
&lt;div id="deleteModal" class="modal"&gt;
&lt;div class="modal-content"&gt;
&lt;h4>Delete Product</h4>
<p>Are you sure you want to delete this product?</p>
<div class="modal-buttons">
<button id="confirmDelete" class="confirm-btn">Yes, Delete</button>
<button id="cancelDelete" class="cancel-btn">Cancel</button>
</div>
</div>
</div>
@include('home.footer')
@include('home.copy')
<!-- jquery -->
<script src="home/js/jquery-3.4.1.min.js"></script>
<!-- popper js -->
<script src="home/js/popper.min.js"></script>
<!-- bootstrap js -->
<script src="home/js/bootstrap.js"></script>
<!-- custom js -->
<script src="home/js/custom.js"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.bundle.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<script>
document.addEventListener('DOMContentLoaded', function() {
// Select all quantity controls
const quantityControls = document.querySelectorAll('.quantity-controls');

quantityControls.forEach(control => {
const decreaseButton = control.querySelector('.decrease');
const increaseButton = control.querySelector('.increase');
const quantityValue = control.querySelector('.quantity-value');

// Set initial quantity
let quantity = parseInt(quantityValue.textContent);

// Decrease quantity
decreaseButton.addEventListener('click', function(e) {
```

```
e.preventDefault(); // Prevent the default link behavior
if (quantity > 1) {
  quantity--;
  quantityValue.textContent = quantity;
  updateTotalPrice(); // Update the total price if needed
}
});

// Increase quantity
increaseButton.addEventListener('click', function(e) {
e.preventDefault(); // Prevent the default link behavior
quantity++;
quantityValue.textContent = quantity;
updateTotalPrice(); // Update the total price if needed
});
});

// Function to update the total price (if needed)
function updateTotalPrice() {
// Add logic to update the total price based on the quantity
console.log('Update total price logic here');
}
});

</script>
<!-- Add JavaScript for Modal -->
<script>
document.addEventListener('DOMContentLoaded', function() {
const deleteButtons = document.querySelectorAll('.delete-btn');
const modal = document.getElementById('deleteModal');
const confirmDelete = document.getElementById('confirmDelete');
const cancelDelete = document.getElementById('cancelDelete');
let deleteUrl = '';

deleteButtons.forEach(button => {
button.addEventListener('click', function(event) {
event.preventDefault();
deleteUrl = this.getAttribute('data-delete-url');
modal.style.display = 'flex';
});
});

confirmDelete.addEventListener('click', function() {
window.location.href = deleteUrl;
});
});
```

```
cancelDelete.addEventListener('click', function() {
  modal.style.display = 'none';
});

window.addEventListener('click', function(event) {
  if (event.target === modal) {
    modal.style.display = 'none';
  }
});
});

</script>
//ajax.js
<script>
$(document).ready(function() {
  $(".changeQuantity").click(function(e) {
    e.preventDefault();

    var $row = $(this).closest('.test');
    var id = $row.find('.cart-id').val();
    var quantity = $row.find('.quantity-value').text().trim();

    $.ajax({
      url: '{{ url('update_cart') }}',
      type: 'POST',
      data: {
        id: id,
        quantity: quantity,
        _token: '{{ csrf_token() }}'
      },
      success: function(response) {
        if (response.success) {
          location.reload(); // Reload the page to reflect changes
        } else {
          alert("Failed to update cart.");
        }
      },
      error: function(xhr) {
        console.log(xhr.responseText);
        alert("An error occurred. Please try again.");
      }
    });
  });
});

</script>
</body>
```

```
</html>
```

- Delete product from Cart:  
-<a href="{{url('delete\_product\_cart',\$cart->id)}}>Delete</a>
- In the web.php:
- `route::get('/delete_product_cart/{id}',  
[HomeController::class, 'delete_product_cart']);`
- In the HomeController:

```
-public function delete_product_cart($id)

{
$cart = Cart::find($id);
//if cart found
if ($cart) {
//find the product in the cart
$product = Product::find($cart->product_id);
// restock the quantity
$product->stock += $cart->product_quantity;
// save to database
$product->save();
}
//delete
$cart->delete();
return redirect()->back()->with('success', 'Product deleted from cart.');
}
```

- Javascript frameworks recommended for more advanced functionality with javascript
- Now for the Order:  
-php artisan make:model Order -m
- In case, route problems or view:  
-run this commands:  
-php artisan route:clear  
-php artisan cache:clear  
-php artisan config:clear  
-php artisan serve
- In the migration of the Order:

```
-public function up(): void
{
Schema::create('orders', function (Blueprint $table) {
$table->id();
// Customer Information
$table->string('customer_name')->nullable();
```

```



```

- php artisan migrate
- make a checkout button in the cart.blade.php:

```

-<form method="post" action="{{url('checkout')}}">
@csrf
<button class="btn">CHECKOUT</button>
</form>

```

- In the web.php:

```

-Route::post('/checkout', [HomeController::class, 'checkout'])-
>name('checkout');

```

- In the HomeController:

```
- public function checkout()
{
    //If logged in
    if (Auth::id()) {
        //take the id of the user
        $id = Auth::user()->id;
        $user = Auth::user();
        //find the cart of the user
        $carts = Cart::where('user_id', '=', $id)->get();
        // Initialize the total order price
        $total = 0;

        // Loop through cart items to calculate the total
        //Sum the total of the cart
        foreach ($carts as $cart) {
            $total += $cart->product_quantity * $cart->product_price;
        }
        //don't let user to checkout if total = 0
        if ($total == 0) {
            return redirect()->route('cart')->with('message', 'No items in cart');
        }
        // redirect to checkout, we should create the checkout
        return view('home.checkout', compact('carts', 'user'));
    }
    //if not logged in redirect to login page
    else {
        return redirect()->route('login')->with('message', 'Please login to
        checkout');
    }
}
```

- Create the checkout.blade.php in the home folder:

```
-<!DOCTYPE html>
<html lang="en">

<head>
<base href="/public">
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Checkout</title>
<link rel="stylesheet" href="home/checkout/style.css" />
</head>
```

```
<body>
<header>
<h3>Checkout</h3>
</header>

<main>
<section class="checkout-form">
//we should define this route in the web.php
<form action="{{ route('place_order') }}" method="POST">
@csrf // don't forget the @csrf
<h6>Contact Information</h6>
<div class="form-control">
<label for="checkout-email">E-mail</label>
<div>
<span class="fa fa-envelope"></span>
//put the user's email as a default value
<input type="email" id="checkout-email" name="checkout-email"
placeholder="Enter your email..." value="{{ $user->email }}" required>
</div>
</div>
<div class="form-control">
<label for="checkout-phone">Phone</label>
<div>
<span class="fa fa-phone"></span>
//put the user's phone as a value
<input type="tel" name="checkout-phone" id="checkout-phone"
placeholder="Enter your phone..." value="{{ $user->phone }}" required>
</div>
</div>

<br>
<h6>Shipping Address</h6>
<div class="form-control">
<label for="checkout-name">Full Name</label>
<div>
<span class="fa fa-user-circle"></span>
//put the user's name as a default value
<input type="text" id="checkout-name" name="checkout-name"
placeholder="Enter your name..."
value="{{ $user->name }}" required>
</div>
</div>
<div class="form-control">
<label for="checkout-address">Address in Details</label>
```

```
<div>
<span class="fa fa-home"></span>
//put the user's address as a default value
<input type="text" name="checkout_address" id="checkout-address"
placeholder="Your address..." value="{{ $user->address }}" required>
</div>
</div>
<div class="form-control">
<label for="checkout-city">City</label>
<div>
<span class="fa fa-building"></span>
<input type="text" name="checkout_city" id="checkout-city"
placeholder="Your city..." required>
</div>
</div>
<!--<div class="form-group">
<div class="form-control">
<label for="checkout-country">Country</label>
<div>
<span class="fa fa-globe"></span>
<input type="text" name="checkout-country" id="checkout-country"
placeholder="Your country..." list="country-list" required>
<datalist id="country-list">
<option value="Lebanon"></option>
<option value="Syria"></option>
<option value="Iraq"></option>
</datalist>
</div>
</div>
</div>-->

<br>
//optional if billing address different from shipping address
<h6>Different Billing Address</h6>
<div class="form-control-billing">
<input type="checkbox" id="billing-checkbox"><label for="billing-
checkbox">Billing address different from shipping address?</label>
</div>

<div id="billing-section" style="display: none;">
<div class="form-control">
<label for="billing-address">Billing Address</label>
<div>
<span class="fa fa-home"></span>
```

```
<input type="text" name="billing-address" id="billing-address"
placeholder="Your billing address...">
</div>
</div>
<div class="form-control">
<label for="billing-city">City</label>
<div>
<span class="fa fa-building"></span>
<input type="text" name="billing_city" id="billing-city"
placeholder="Billing city...">
</div>
</div>
</div>

<br>
//different payment methods as radio buttons
<h6>Payment Method</h6>
<div class="form-control">
<label>Choose Payment Method</label>
<div class="payment-options">
<label><input type="radio" name="payment_method" value="credit_card"
checked> Credit Card</label>
<label><input type="radio" name="payment_method" value="paypal">
PayPal</label>
<label><input type="radio" name="payment_method" value="stripe">
Stripe</label>
<label><input type="radio" name="payment_method" value="cash_on_delivery">
Cash on Delivery</label>
</div>
</div>

<div class="form-control-btn">
<button type="submit">Place Order</button>
</div>
</form>
</section>

<section class="checkout-details">
<div class="checkout-details-inner">
<div class="checkout-lists">
<h1>Order Details</h1>
</div>
<div class="checkout-shipping">
<h6>Shipping</h6>
<p>$4</p>
```

```

</div>
<?php $total = 0; ?>
<div class="checkout-total">
<h6>Total</h6>
<p>
//calculating the total
@foreach ($carts as $cart)
<?php $total += $cart->product_price * $cart->product_quantity; ?>
@endforeach
//displaying the total
$<?php echo $total + 4; ?>
</p>
</div>
</div>
</section>
</main>
<script>
document.getElementById('billing-checkbox').addEventListener('change',
function() {
document.getElementById('billing-section').style.display = this.checked ?
'block' : 'none';
});
</script>
</body>

</html>

```

- In the web.php:  
-route name place.order

```

-Route::post('/place_order', [HomeController::class, 'place_order'])
>name('place_order')

```

- In the HomeController:

```

- public function place_order(Request $request)
{
// Get the authenticated user
$id = Auth::user()->id;
$user = Auth::user();

// Get all items in the user's cart
$carts = Cart::where('user_id', '=', $id)->get();

// Initialize the total order price

```

```

$total = 0;

// Loop through cart items to calculate the total
foreach ($carts as $cart) {
    $total += $cart->product_quantity * $cart->product_price;
}
if ($total == 0) {
    return redirect()->route('cart')->with('message', 'No items in cart');
}

// Get the payment method from the request
$payment_method = $request->payment_method; // Default is credit card

// Billing address and billing city (Optional: check if different from
// shipping)
$billing_address = $request->input('billing_address');
$billing_city = $request->billing_city;
$shipping_city = $request->checkout_city;
$shipping_address = $request->input('checkout_address');

// Check if the payment method is credit card, paypal, or stripe
if (in_array($payment_method, ['credit_card', 'paypal'])) {
    // If the payment method is credit card, paypal, or stripe, redirect to a
    // specific route without saving the data
    // For example, to a payment gateway route
    return redirect()->back();
}
// handling stripe
if ($payment_method == "stripe") {
    // sending data into stripe.blade.php
    return view('home.stripe', compact(
        'total',
        'carts',
        'billing_address',
        'billing_city',
        'shipping_address',
        'shipping_city',
        'user'
    ));
}

// If the payment method is cash on delivery, we save the order
if ($payment_method == "cash_on_delivery") {
    // Create a new order
    foreach ($carts as $cart) {

```

```

$order = new Order;

// Assign customer information from authenticated user
$order->user_id = $user->id;
$order->customer_name = $user->name;
$order->email = $user->email;
$order->customer_phone = $user->phone;
$order->shipping_address = $shipping_address; // Assuming the user has a
shipping address
$order->shipping_city = $request->checkout_city; // Assuming the user has
a city field

// Use the billing address and billing city if provided, otherwise use the
shipping address and city
$order->billing_address = $billing_address ?: $shipping_address; //
Default to shipping address
$order->billing_city = $billing_city ?: $request->checkout_city; //
Default to shipping city

// Loop through cart items and save product details in the order

$order->product_name = $cart->product_title;
$order->quantity = $cart->product_quantity;
$order->price = $cart->product_price;
$order->image_url = $cart->image;
$order->product_id = $cart->product_id;

// Set payment method
$order->payment_method = $payment_method;
$order->total_order_price = $total + 4; // Total calculated from the cart

// Set payment status and order status
$order->payment_status = 'pending';
$order->order_status = 'pending';

// Save the order
$order->save();
}

// Clear the user's cart after placing the order
Cart::where('user_id', '=', $id)->delete();

// Return a success response for cash on delivery
return redirect()->route('cart')->with('message', 'Successfully Added
Order');

```

```
}
```

- Now for Order History:
- add in the header.blade.php in user-side:

```
<li class="nav-item">
<a class="nav-link" href="{{ url('order_history') }}">Orders</a>
</li>
```

- In the web.php:

```
-Route::get('/order_history', [HomeController::class, 'order_history'])->
name('order_history');
```

- In the HomeController:

```
-public function order_history()
{
//if user is logged in
if (Auth::id()) {
$user = Auth::user();
//taking the orders for specific user
$orders = Order::where('user_id', $user->id)->get();
//sending data
return view('home.order', compact('orders'));
}
//if user is not logged in
else {
return redirect()->route('login')->with('message', 'Please login to view
order history.');
}
}
```

- Create the order.blade.php and integrate the sent data orders
- Orders.blade.php:

- 
- <body>
- <div class="hero\_area">
- @include('home.header')
- <div class="container order-history-container">
- <h2 class="text-center" style="color: #d4af37; margin-top: 20px;">Order
History</h2>
- 
- <!-- Filter and Search Form -->
- //we will define it in the web.php
- <form action="{{ url('search\_filter') }}" method="GET" class="filter-
section">

```

• //for the filtering
• <select name="time_filter">
•   <option value="all" {{ request('time_filter') == 'all' ? 'selected' : '' }}>All Time</option>
•   <option value="newtoold" {{ request('time_filter') == 'newtoold' ? 'selected' : '' }}>Newer To Older</option>
•   <option value="oldtonew" {{ request('time_filter') == 'oldtonew' ? 'selected' : '' }}>Old To Newer</option>
•   <option value="last7" {{ request('time_filter') == 'last7' ? 'selected' : '' }}>Last 7 Days</option>
•   <option value="last30" {{ request('time_filter') == 'last30' ? 'selected' : '' }}>Last 30 Days</option>
•   <option value="last90" {{ request('time_filter') == 'last90' ? 'selected' : '' }}>Last 90 Days</option>
• </select>
• //for the search
• <input type="text" name="search" placeholder="Search by Order ID or Item..." value="{{ request('search') }}" />
• <button type="submit">Apply</button>
• </form>
•
• <!-- Order History Table -->
• <table class="order-history-table">
•   <thead>
•     <tr>
•       <th>Order ID</th>
•       <th>Image</th>
•       <th>Date</th>
•       <th>Items Name</th>
•       <th>Quantity</th>
•       <th>Total</th>
•       <th>Status</th>
•     </tr>
•   </thead>
•   <tbody>
•     //displaying the data
•     @foreach ($orders as $order)
•       <tr>
•         <td data-label="Order ID">#{{ $order->id }}</td>
•         <td data-label="Image">item_name }}"/></td>
•         <td data-label="Date">{{ $order->created_at->format('Y-m-d') }}</td>
•         <td data-label="Items Name">{{ $order->product_name }}</td>
•         <td data-label="Quantity">{{ $order->quantity }}</td>

```

```

• <td data-label="Total">${{ number_format($order->total_order_price, 2)
  }}</td>
• <td data-label="Status" class="status {{ $order->status == 'cancelled'
  ? 'cancelled' : '' }}">
• {{ ucfirst($order->order_status) }}</td>
• </tr>
• @endforeach
• </tbody>
• </table>
• </div>
•
• @include('home.footer')
• @include('home.copy')
• <!-- jQery -->
• <script src="home/js/jquery-3.4.1.min.js"></script>
• <!-- popper js -->
• <script src="home/js/popper.min.js"></script>
• <!-- bootstrap js -->
• <script src="home/js/bootstrap.js"></script>
• <!-- custom js -->
• <script src="home/js/custom.js"></script>
• </body>
•

```

- In the web.php:

```
-Route::get('/search_filter', action: [HomeController::class,
'search_filter_order'])->name('order.history');
```

- In the HomeController:

```

• public function search_filter_order(Request $request)
• {
•   $query = Order::query();
•
•   // Filter by time
•   if ($request->has('time_filter')) {
•     $timeFilter = $request->input('time_filter');//take the requested
•     filter
•     $now = now();// return the current date and time
•
•     switch ($timeFilter) {
•       case 'newtoold':
•         $query->orderBy('created_at', 'desc');
•         break;

```

```

• case 'oldtonew':
•     $query->orderBy('created_at', 'asc');
•     break;
• //return the last 7 days
• case 'last7':
•     $query->where('created_at', '>=', $now->subDays(7));
•     break;
• //return the last 30 days
• case 'last30':
•     $query->where('created_at', '>=', $now->subDays(30));
•     break;
• //return the last 90 days
• case 'last90':
•     $query->where('created_at', '>=', $now->subDays(90));
•     break;
• // 'all' is the default, so no filtering needed
• }
• }

• // Search by Order ID or Item Name
• if ($request->has('search')) {
•     $searchTerm = $request->input('search');
•     $query->where(function ($q) use ($searchTerm) {
•         $q->where('id', 'like', "%$searchTerm%");
•         //product_name is column in the order database table
•         ->orWhere('product_name', 'like', "%$searchTerm%");
•         //order_status is column in the order database table
•         ->orWhere('order_status', 'like', "%$searchTerm%");
•     });
• }
• 

• // Get the filtered orders or searched order
• $orders = $query->get();
• 

• return view('home.order', compact('orders'));
• }
```

- There are many ways to search data in Laravel we will discuss them later but one of them is the way I mentioned before.

## Integrating Card Payment (Stripe) in Laravel

- Integrating Card Payment (Stripe) in Laravel:
  - General steps, we will dive in them:
  - 1-composer require stripe/stripe-php (Run it in the terminal of the IDE)
  - 2>Create stripe account on [stripe.com](https://stripe.com)
  - 3- changes in the .env file
  - 4-Make a view for payment
  - 5-Write code for payment in the Controller
- After creating stripe account:
  - go to [stripe.com](https://stripe.com) and get the Publishable Key and Secret Key
- Go to the .env file and add at the bottom:

```
-  

STRIPE_KEY=pk_test_51QpE7fQU0yQg39NN0f2x066UQS2cpqFQ7APsi2rVxpR0caTzRneD30  

1FjirBiM (Publishable Key)  

  

STRIPE_SECRET=sk_test_51QpE7fQU0yQg39NNZrxnPNaETEtP0EQcyHlgWGkg1QeyVzX8mvZ  

B(Secret key)
```

- Make a route in the web.php: for posting

```
-Route::post('stripe/{total}/{shipping_city}', [HomeController::class,  

'stripePost'])->name('stripe.post');
```

- How to go to the view? You have two ways, I mentioned it one of them before in the place\_order function in the HomeController ,the second way is making route::get as usual to get view.
- In the HomeController:
  - stripePost function that we took it also from the Laravel11Stripe documentation

```
-<?php  

  

namespace App\Http\Controllers;  

  

use Illuminate\Http\Request;  

use Stripe;  

use Illuminate\View\View;  

use Illuminate\Http\RedirectResponse;  

  

class StripePaymentController extends Controller  

{  

    /**  

     * success response method.  

     */
```

```

    * @return \Illuminate\Http\Response
    */ // not needed in our code!!!!!!!!!!!!!!!
public function stripe(): View
{
    return view('stripe');
}

/**
 * success response method.
 *
 * @return \Illuminate\Http\Response
 */
public function stripePost(Request $request): RedirectResponse
{
    Stripe\Stripe::setApiKey(env('STRIPE_SECRET'));

    Stripe\Charge::create([
        "amount" => 10 * 100,
        "currency" => "usd",
        "source" => $request->stripeToken,
        "description" => "Test payment from itsolutionstuff.com."
    ]);

    return back()
        ->with('success', 'Payment successful!');
}

```

- Update it to: (only the stripePost function)

```

- public function stripePost(Request $request, $total, $shipping_city): RedirectResponse
{
    Stripe\Stripe::setApiKey(env('STRIPE_SECRET'));

    Stripe\Charge::create([
        //the total we sent In the view (handled in cents)
        "amount" => $total * 100,
        "currency" => "usd",
        "source" => $request->stripeToken,
        "description" => "Thanks for payment"
    ]);
    $id = Auth::user()->id;
    $user = Auth::user();
    // Get all items in the user's cart
}

```

```
$carts = Cart::where('user_id', '=', $id)->get();
foreach ($carts as $cart) {
$order = new Order;

// Assign customer information from authenticated user
$order->user_id = $user->id;
$order->customer_name = $user->name;
$order->email = $user->email;
$order->customer_phone = $user->phone;
$order->shipping_address = $user->address; // Assuming the user has a
shipping address
$order->shipping_city = $shipping_city; // Assuming the user has a city
field

// Use the billing address and billing city if provided, otherwise use the
shipping address and city
$order->billing_address = "N/A"; // Default to shipping address
$order->billing_city = "N/A"; // Default to shipping city

// Loop through cart items and save product details in the order

$order->product_name = $cart->product_title;
$order->quantity = $cart->product_quantity;
$order->price = $cart->product_price;
$order->image_url = $cart->image;
$order->product_id = $cart->product_id;

// Set payment method
$order->payment_method = "stripe";
$order->total_order_price = $total; // Total calculated from the cart

// Set payment status and order status
$order->payment_status = 'PAID';
$order->order_status = 'pending';

// Save the order
$order->save();
}

// Clear the user's cart after placing the order
Cart::where('user_id', '=', $id)->delete();
return redirect()->route('cart');
}
```

## Admin Side

- Now to show Orders in Admin Panel:

-add in the sidebar.blade.php:(admin side)

```
<li class="nav-item menu-items">
<a class="nav-link" href="{{ url('view_orders') }}>
<span class="menu-icon">
<i class="mdi mdi-playlist-play"></i>
</span>
<span class="menu-title">View Orders</span>
</a>
</li>
```

- In the web.php:

```
-Route::get('/view_orders', [AdminController::class, 'view_orders']);
```

- In the AdminController:

```
-use App\Models\Order;

public function view_orders()
{
    $data = Order::all();
    return view('admin.orders', compact('data'));
}
```

- Create orders.blade.php in the admin side:

```
-<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
<title>Corona Admin</title>
@include('admin.css')
<style>
.card {
background-color: black;
color: gold;
```

```
border-radius: 10px;
padding: 20px;
}

.table {
color: gold;
border-color: gold;
width: 100%;
}

.table th,
.table td {
border: 1px solid gold;
text-align: center;
padding: 10px;
}

.table th {
background-color: gold;
font-weight: bold;
}

.table img {
width: 50px;
height: 50px;
border-radius: 5px;
object-fit: cover;
}

.btn {
color: black;
font-weight: bold;
padding: 8px 12px;
border-radius: 5px;
}

.btn-warning {
background-color: gold;
border: none;
}

.btn-info {
background-color: #d4af37;
border: none;
}
```

```
.btn-success {  
background-color: #b8860b;  
border: none;  
}  
  
.modal-content {  
background-color: black;  
color: gold;  
}  
  
.modal-header,  
.modal-footer {  
border-color: gold;  
}  
  
.modal-dialog {  
display: flex;  
align-items: center;  
justify-content: center;  
min-height: 100vh;  
}  
  
.search-filter-container {  
display: flex;  
justify-content: center;  
align-items: center;  
gap: 10px;  
padding-bottom: 30px;  
flex-wrap: wrap;  
}  
  
.search-filter-container input[type="text"],  
.search-filter-container select {  
width: 250px;  
padding: 10px;  
border: 2px solid gold;  
border-radius: 5px;  
background-color: black;  
color: gold;  
font-size: 16px;  
outline: none;  
transition: all 0.3s ease-in-out;  
}
```

```
.search-filter-container input[type="text"]::placeholder {  
color: gold;  
opacity: 0.7;  
}  
  
.search-filter-container input[type="text"]:focus,  
.search-filter-container select:focus {  
box-shadow: 0px 0px 10px gold;  
}  
  
.search-filter-container select {  
cursor: pointer;  
appearance: none;  
background-image: linear-gradient(45deg, transparent 50%, gold 50%),  
linear-gradient(135deg, gold 50%, transparent 50%);  
background-position: calc(100% - 15px) calc(1em + 2px), calc(100% - 10px)  
calc(1em + 2px);  
background-size: 5px 5px, 5px 5px;  
background-repeat: no-repeat;  
padding-right: 30px;  
}  
  
.search-filter-container .btn {  
padding: 10px 15px;  
background-color: gold;  
color: black;  
border: none;  
border-radius: 5px;  
font-weight: bold;  
cursor: pointer;  
transition: 0.3s ease-in-out;  
}  
  
.search-filter-container .btn:hover {  
background-color: darkgoldenrod;  
}  
</style>  
</head>  
  
<body>  
<div class="container-scroller">  
@include('admin.sidebar')  
@include('admin.header')  
<div class="main-panel">  
<div class="content-wrapper">
```

```
<div class="container-fluid">
<div class="card">
<div class="card-header text-center">
<h3>Orders Management</h3>
</div>
<div class="search-filter-container">
// we now how to handle filter
<form action="{{ route('search.orders') }}" method="get">
@csrf
<input type="text" name="search" placeholder="Search for something...">
<select name="filter">
<option value="">Filter by...</option>
<option value="oldtonew">Oldest to Newest</option>
<option value="newtoold">Newest to Oldest</option>
<option value="pending">Pending Orders</option>
<option value="delivered">Delivered Orders</option>
<option value="paid">Paid Orders</option>
<option value="unpaid">Unpaid Orders</option>
</select>
<input type="submit" value="Search" class="btn">
</form>
</div>
<div class="card-body">
<div class="table-responsive">
<table class="table table-bordered">
<thead>
<tr>
<th>ID</th>
<th>Customer Name</th>
<th>Email</th>
<th>Phone</th>
<th>Shipping Address|Shipping City</th>
<th>Billing Address|Billing City</th>
<th>Product Name</th>
<th>Image</th>
<th>Quantity</th>
<th>Price</th>
<th>Payment Method</th>
<th>Order Status</th>
<th>Payment Status</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
//using forelse to handle empty case, displaying all data
```

```

@forelse($data as $order)
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| {{ $order->id }} | {{ $order->customer_name }} | {{ $order->email }} | {{ $order->customer_phone }} | {{ $order->shipping_address }}, {{ $order->shipping_city }} | {{ $order->billing_address }}, {{ $order->billing_city }} | {{ $order->product_name }} | @if ($order->image_url) Product Image @else No Image @endif | {{ $order->quantity }} | {{{ number_format($order->price, 2) }}} | {{ ucfirst(str_replace('_', ' ', $order->payment_method)) }} | {{ ucfirst($order->order_status) }} | {{ ucfirst($order->payment_status) }} | // we will handle it soon Send Email // we will handle it soon Print PDF // we know how to delete with get method!! Delete Order //button for updating order_status in case pending @if ($order->order_status == 'pending') <button class="btn btn-info btn-sm" data-toggle="modal" data-target="#orderModal{{ $order->id }}>Update Order</button> @else <p>Delivered!</p> @endif //button for updating payment_status in case pending @if ($order->payment_status == 'pending') <button class="btn btn-success btn-sm" data-toggle="modal" data-target="#paymentModal{{ $order->id }}>Update Payment</button> @endif |

```

```
<p>Paid!</p>
@endif
</td>
</tr>
<!-- Order Status Modal -->
<div class="modal fade" id="orderModal{{ $order->id }}" tabindex="-1"
role="dialog">
<div class="modal-dialog" role="document">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title">Confirm Order Status Update</h5>
<button type="button" class="close"
data-dismiss="modal">&times;</button>
</div>
<div class="modal-body">
<p>Are you sure you want to update the order status?</p>
</div>
<div class="modal-footer">
<a href="{{ url('update_order_status', $order->id) }}"
class="btn btn-success">Yes</a>
<button type="button" class="btn btn-danger"
data-dismiss="modal">No</button>
</div>
</div>
</div>
</div>

<!-- Payment Status Modal -->
<div class="modal fade" id="paymentModal{{ $order->id }}" tabindex="-1"
role="dialog">
<div class="modal-dialog" role="document">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title">Confirm Payment Status Update</h5>
<button type="button" class="close"
data-dismiss="modal">&times;</button>
</div>
<div class="modal-body">
<p>Are you sure you want to update the payment status?</p>
</div>
<div class="modal-footer">
<a href="{{ url('update_payment_status', $order->id) }}"
class="btn btn-success">Yes</a>
<button type="button" class="btn btn-danger"
data-dismiss="modal">No</button>
```

```

</div>
</div>
</div>
</div>
@empty
<tr>
<td colspan="14" class="text-center">No orders found.</td>
</tr>
@endforelse
</tbody>
</table>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
@include('admin/scripts')
</body>

</html>

```

- Now for Updating Delivery status:

- In the web.php:
 

```

Route::get('/update_order_status/{id}', [AdminController::class,
    'update_order_status']);
      
```
- In the AdminController:

```

public function update_order_status($id)
{
$order = Order::find($id);
$order->order_status = 'Delivered';
$order->save();
return redirect()->back()->with('message', 'Order Status Updated
Successfully');
}

```

- Now for Updating Payment Status:

- In the web.php:

```

Route::get('/update_payment_status/{id}', [AdminController::class,
    'update_payment_status']);
      
```

- In the AdminController:

```

public function update_payment_status($id)

```

```
{
$order = Order::find($id);
$order->payment_status = 'Paid';
$order->save();
return redirect()->back()->with('message', 'Payment Status Updated
Successfully');
}
```

- Now for Downloading or Printing PDF File: One step only
  - Composer require barryvdh/laravel-dompdf
- In the orders.blade.php (admin side):

```
-<a href="{{ url('print_pdf', $order->id) }}"
class="btn btn-warning btn-sm">Print PDF</a>
```

- In the AdminController:

```
->
use Barryvdh\DomPDF\Facade\Pdf as PDF;
public function print_pdf($id)

{
// find the order by id
$order = Order::find($id);
// handling the download with pdf.blade.php
$pdf = PDF::loadView('admin.pdf', compact('order'));
// download in pdf named order_details.pdf
return $pdf->download('order_details.pdf');
}
```

- In the pdf.blade.php:
  - For example:

```
<body>
<div class="invoice-box">
<h2>Invoice</h2>

<!-- Customer Information -->
<p class="section-title">Customer Details:</p>
<p><strong>Name:</strong> {{ $order->customer_name ?? 'N/A' }}</p>
<p><strong>Email:</strong> {{ $order->email ?? 'N/A' }}</p>
<p><strong>Phone:</strong> {{ $order->customer_phone ?? 'N/A' }}</p>

<!-- Shipping Information -->
<p class="section-title">Shipping Address:</p>
<p>{{ $order->shipping_address ?? 'N/A' }}, {{ $order->shipping_city ??
'N/A' }}</p>
```

```

<!-- Billing Information -->


Billing Address:</p>
<p>{{ $order->billing_address ?? 'N/A' }}, {{ $order->billing_city ?? 'N/A' }}</p>

<!-- Order Information -->


Order Details:</p>
<p><strong>Order ID:</strong> {{ $order->id }}</p>
<p><strong>Order Date:</strong> {{ $order->order_date }}</p>
<p><strong>Payment Method:</strong> {{ ucfirst(str_replace('_', ' ', $order->payment_method)) }}</p>
<p><strong>Payment Status:</strong> {{ ucfirst($order->payment_status) }}</p>
<p><strong>Order Status:</strong> {{ ucfirst($order->order_status) }}</p>
<p class="section-title">Product Details:</p>
<!-- Product Details -->


| Image                                                                                                                | Product                     | Quantity                | Price                                    | Total                                                        |
|----------------------------------------------------------------------------------------------------------------------|-----------------------------|-------------------------|------------------------------------------|--------------------------------------------------------------|
| @if(\$order->image_url)  @else No Image @endif | {{ \$order->product_name }} | {{ \$order->quantity }} | \${{ number_format(\$order->price, 2) }} | \${{ number_format(\$order->quantity * \$order->price, 2) }} |
| Grand Total                                                                                                          |                             |                         |                                          | \${{ number_format(\$order->total_order_price, 2) }}         |


```

```
</table>
</div>
</body>
```

## Email-Verification in LARAVEL

- Now for Email-Verification in LARAVEL:  
 -Go to the .env file:  
 Update or/and add:

```
MAIL_MAILER=smtp
MAIL_SCHEME=null
MAIL_HOST=smtp.gmail.com
MAIL_PORT=587
MAIL_ENCRYPTION=tls /or ssl
MAIL_USERNAME=thee.z.agencyy@gmail.com (your email)
MAIL_PASSWORD=velvdhiewh8w89ihswks (mail password)
MAIL_FROM_ADDRESS="thee.z.agencyy@gmail.com"
MAIL_FROM_NAME="Royal Cedar Scents Mail"
```

-To get the MAIL\_PASSWORD:

Go to the gmail.com > manage google account > security option > 2 step verification must be on > search for app passwords > generate app password > add it to MAIL\_PASSWORD, remove spaces.

- Go to user.php:
- make it implement MustVerifyEmail
- add, use Illuminate\Contracts\Auth\MustVerifyEmail;

```
• <?php
•
• namespace App\Models;
•
• // use Illuminate\Contracts\Auth\MustVerifyEmail;
• use Illuminate\Database\Eloquent\Factories\HasFactory;
• use Illuminate\Foundation\Auth\User as Authenticatable;
• use Illuminate\Notifications\Notifiable;
• use Illuminate\Contracts\Auth\MustVerifyEmail;
• use Laravel\Fortify\TwoFactorAuthenticatable;
• use Laravel\Jetstream\HasProfilePhoto;
• use Laravel\Sanctum\HasApiTokens;
• // make it implement MustVerifyEmail
• class User extends Authenticatable implements MustVerifyEmail
• {
```

```
• use HasApiTokens;  
•  
• /** @use HasFactory<\Database\Factories\UserFactory> */  
• use HasFactory;  
• use HasProfilePhoto;  
• use Notifiable;  
• use TwoFactorAuthenticatable;  
•  
• /**  
• * The attributes that are mass assignable.  
• *  
• * @var array<int, string>  
• */  
• protected $fillable = [  
•     'name',  
•     'email',  
•     'address',  
•     'phone',  
•     'password',  
• ];  
•  
• /**  
• * The attributes that should be hidden for serialization.  
• *  
• * @var array<int, string>  
• */  
• protected $hidden = [  
•     'password',  
•     'remember_token',  
•     'two_factor_recovery_codes',  
•     'two_factor_secret',  
• ];  
•  
• /**  
• * The accessors to append to the model's array form.  
• *  
• * @var array<int, string>  
• */  
• protected $appends = [  
•     'profile_photo_url',  
• ];  
•  
• /**  
• * Get the attributes that should be cast.  
• *
```

- \* @return array<string, string>
- \*/
- protected function casts(): array
- {
- return [
- 'email\_verified\_at' => 'datetime',
- 'password' => 'hashed',
- ];
- }
- // Define the relationship with the Subscription model
- public function subscription()
- {
- return \$this->hasOne(Subscribe::class);
- }
- }
-

- Go to Fortify.php:

- -just uncomment Features::emailVerification(),
- 'features' => [
- Features::registration(),
- Features::resetPasswords(),
- Features::emailVerification(),
- Features::updateProfileInformation(),
- Features::updatePasswords(),
- Features::twoFactorAuthentication([
- 'confirm' => true,
- 'confirmPassword' => true,
- // 'window' => 0,
- ]),
- ],

- Go and update to the web.php and update the redirect route:

```
- route::get('redirect',[HomeController::class,'redirect'])-
>middleware('auth','verified');
```

- If you want to change the (resend verification link view) go to resources/auth/verify-email.blade.php
- If you want to change the email itself change the email.blade.php

## Sending Email-Notification in LARAVEL

- Sending Email-Notification in LARAVEL:  
-in the orders.blade.php (admin side):

```
<a href="{{ url('send_email', $order->id) }}>
```

```
class="btn btn-danger btn-sm">Send Email</a>
```

-In the web.php:

```
Route::get('/send_email/{id}', [AdminController::class, 'send_email']);
```

-In the AdminController:

```
public function send_email($id)
{
    $order = Order::find($id);
    return view('admin.email_order', compact('order'));
}
```

-Create the email\_order.blade.php (admin side)

For example:

```
<body>
<div class="container-scroller">
<!-- partial:partials/_sidebar.html -->
@include('admin.sidebar')
<!-- partial -->
<div class="container-fluid page-body-wrapper">
<!-- partial:partials/_navbar.html -->
@include('admin.header')
<!-- partial -->
<div class="main-panel">
<div class="content-wrapper">
@if (session()->has('message'))
<div class="alert alert-success">
<button type="button" class="close" data-dismiss="alert">x</button>
{{ session()->get('message') }}
</div>
@endif
<h1 class="email-form-title">Send Email to {{$order->email}}</h1>
// we will handle it soon
<form class="email-form" method="POST"
action="{{url('send_user_email',$order->id)}}">
@csrf
<div class="email-form-group">
<label>Email Greeting :</label>
<input type="text" name="greeting">
</div>
<div class="email-form-group">
<label>Email FirstLine :</label>
<input type="text" name="firstline">
```

```

</div>
<div class="email-form-group">
<label>Email Body :</label>
<input type="text" name="body">
</div>
<div class="email-form-group">
<label>Email Button name :</label>
<input type="text" name="button">
</div>
<div class="email-form-group">
<label>Email Url :</label>
<input type="text" name="url">
</div>
<div class="email-form-group">
<label>Email LastLine :</label>
<input type="text" name="lastline">
</div>
<div class="email-form-group">
<input type="submit" value="Send Email" class="email-form-submit">
</div>
</form>

</div>
</div>

<!-- main-panel ends -->
</div>
<!-- page-body-wrapper ends -->
</div>
<!-- container-scroller -->
<!-- plugins:js -->
@include('admin.scripts')
<!-- End custom js for this page -->
</body>

```

- Run in the terminal:  
 -php artisan notification:table  
 -php artisan migrate  
 -php artisan make:notification MyFirstNotification
- Go to notifications/MyFirstNotification.php  
 -add private \$details  
 -and modify the code to be like this:

```
<?php
```

```
namespace App\Notifications;

use Illuminate\Bus\Queueable;
use Illuminate\Contracts\Queue\ShouldQueue;
use Illuminate\Notifications\Messages\MailMessage;
use Illuminate\Notifications\Notification;
use App\Notifications\details;

class MyFirstNotification extends Notification
{
    use Queueable;
    // add $details
    private $details;

    /**
     * Create a new notification instance.
     */
    // add to the constructor the $details
    public function __construct($details)
    {
        // set the value
        $this->details = $details;
    }

    /**
     * Get the notification's delivery channels.
     *
     * @return array<int, string>
     */
    public function via(object $notifiable): array
    {
        return ['mail'];
    }

    /**
     * Get the mail representation of the notification.
     */
    public function toMail(object $notifiable): MailMessage
    {
        return (new MailMessage)
            ->greeting($this->details['greeting']) // add greeting
            ->line($this->details['firstline']) // add firstline
            ->line($this->details['body'])//add body
            // add button and its url
            ->action($this->details['button'], $this->details['url'])
    }
}
```

```

        // add the lastline
        ->line($this->details['lastline']);
        // you can add more and more
    }

    /**
     * Get the array representation of the notification.
     *
     * @return array<string, mixed>
     */
    public function toArray(object $notifiable): array
    {
        return [
            //
        ];
    }
}

```

- In the email\_order.blade.php:

```

-<form class="email-form" method="POST"
action="{{url('send_user_email',$order->id)}}>
@csrf
<div class="email-form-group">
<label>Email Greeting :</label>
<input type="text" name="greeting">
</div>
<div class="email-form-group">
<label>Email FirstLine :</label>
<input type="text" name="firstline">
</div>
<div class="email-form-group">
<label>Email Body :</label>
<input type="text" name="body">
</div>
<div class="email-form-group">
<label>Email Button name :</label>
<input type="text" name="button">
</div>
<div class="email-form-group">
<label>Email Url :</label>
<input type="text" name="url">
</div>
<div class="email-form-group">
<label>Email LastLine :</label>

```

```
<input type="text" name="lastline">
</div>
<div class="email-form-group">
<input type="submit" value="Send Email" class="email-form-submit">
</div>
</form>
```

- In the web.php:

```
-Route::post('/send_user_email/{id}', [AdminController::class,
'send_user_email']);
```

- In the AdminController:

```
-public function send_user_email(Request $request, $id)
{
    // find the Order by id
    $order = Order::find($id);
    // add the details data
    $details = [
        // same names as mention in MyFirstNotification.php
        // its case sensitive
        // taking data from the request
        'greeting' => $request->greeting,
        'firstline' => $request->firstline,
        'body' => $request->body,
        'button' => $request->button,
        'url' => $request->url,
        'lastline' => $request->lastline,
    ];
    //notify (be sure in the order there is column named email to let
//the magic work
    $order->notify(new MyFirstNotification($details));
    return redirect()->back()->with('message', 'Email Sent
Successfully');
}
```

- In the Order.php:

```
-class Order extends Model
{
    // add both
    use HasFactory;
    use Notifiable;
}
```

- If you want to customize the email css/or whatever:  
-php artisan vendor:publish --tag=Laravel-mail

## Second way for making Search In Laravel (the easy way)

- Now the second way for making Search In Laravel (the easy way)
  - make a form with method get and @csrf and action="{{url('search-order')}}
  - make in the web.php:

```
Route::get('/search-orders', [AdminController::class, 'searchOrders'])->name('search.orders');
```

-In the AdminController:

```
public function searchOrders(Request $request)
{
    $query = Order::query();
    $now = now();
    // Search Functionality
    if ($request->has('search') && !empty($request->search)) {
        $searchTerm = $request->search;
        $query->where(function ($q) use ($searchTerm) {
            // columnname in database, Like, the searchTerm
            $q->where('customer_name', 'LIKE', "%$searchTerm%")
                ->orWhere('email', 'LIKE', "%$searchTerm%")
                ->orWhere('product_name', 'LIKE', "%$searchTerm%")
                ->orWhere('order_status', 'LIKE', "%$searchTerm%")
                ->orWhere('created_at', 'LIKE', "%$searchTerm%")
                ->orWhere('id', 'LIKE', "%$searchTerm%")
                ->orWhere('payment_status', 'LIKE', "%$searchTerm%")
                ->orWhere('total_order_price', 'LIKE', "%$searchTerm%")
                ->orWhere('payment_method', 'LIKE', "%$searchTerm%")
                ->orWhere('shipping_address', 'LIKE', "%$searchTerm%")
                ->orWhere('shipping_city', 'LIKE', "%$searchTerm%")
                ->orWhere('billing_address', 'LIKE', "%$searchTerm%")
                ->orWhere('billing_city', 'LIKE', "%$searchTerm%")
                ->orWhere('quantity', 'LIKE', "%$searchTerm%")
                ->orWhere('price', 'LIKE', "%$searchTerm%")
                ->orWhere('billing_city', 'LIKE', "%$searchTerm%");
        });
    }

    // Filter Functionality
    if ($request->has('filter') && !empty($request->filter)) {
        if ($request->filter == 'pending') {

```

```

$query->where('order_status', 'pending');
} elseif ($request->filter == 'delivered') {
$query->where('order_status', 'delivered');
} elseif ($request->filter == 'paid') {
$query->where('payment_status', 'paid');
} elseif ($request->filter == 'unpaid') {
$query->where('payment_status', 'pending');
} elseif ($request->filter == 'oldtonew') {
$query->orderBy('created_at', 'asc');
} elseif ($request->filter == 'newtoold') {
$query->orderBy('created_at', 'desc');
}
}

$data = $query->latest()->get(); // Fetch filtered orders
return view('admin.orders', compact('data'));
}
//The simple way for only the search
public function searchdata(Request $request)
{
//the input field name is search name="search"
$searchTerm = $request->search;
$orders=Order::where('customer_name','LIKE','%$searchTerm%')
->orWhere('email', 'LIKE', "%$searchTerm%")
->orWhere('product_name', 'LIKE', "%$searchTerm%")
->orWhere('order_status', 'LIKE', "%$searchTerm%")
->orWhere('created_at', 'LIKE', "%$searchTerm%")
->orWhere('id', 'LIKE', "%$searchTerm%")
->orWhere('payment_status', 'LIKE', "%$searchTerm%")
->orWhere('total_order_price', 'LIKE', "%$searchTerm%")
->orWhere('payment_method', 'LIKE', "%$searchTerm%")
->orWhere('shipping_address', 'LIKE', "%$searchTerm%")
->orWhere('shipping_city', 'LIKE', "%$searchTerm%")
->orWhere('billing_address', 'LIKE', "%$searchTerm%")
->orWhere('billing_city', 'LIKE', "%$searchTerm%")
->orWhere('quantity', 'LIKE', "%$searchTerm%")
->orWhere('price', 'LIKE', "%$searchTerm%")
->orWhere('billing_city', 'LIKE', "%$searchTerm%");

return view('admin.orders',compact('orders'));
}

```

- Display admin Dashboard Data in Laravel:  
-go to the sidebar.blade.php (admin side):

```
<li class="nav-item menu-items">
// it becomes the admin home page
<a class="nav-link" href="{{url('/redirect')}}">
<span class="menu-icon">
<i class="mdi mdi-speedometer"></i>
</span>
<span class="menu-title">Dashboard</span>
</a>
</li>
```

-To send data to admin home page:

Go to the redirect function in the HomeController:

Update it to:

```
public function redirect()
{
    $usertype = Auth::user()->usertype;
    if ($usertype == "admin") {
        $total_product = Product::all()->count();
        $total_order = Order::all()->count();
        $total_user = User::all()->count();
        $order = Order::all();
        $total_revenue = 0;
        foreach ($order as $o) {
            $total_revenue += $o->price;
        }
        $total_order_pending = Order::where('order_status', '=', 'pending')->count();
        $total_order_complete = Order::where('order_status', '=', 'delivered')->count();
        $total_order_not_paid = Order::where('payment_status', '=', 'pending')->count();
        $total_order_paid = Order::where('payment_status', '=', 'paid')->count();
        //sending the data to the view
        return view('admin.home', compact(
            'total_product',
            'total_order',
            'total_user',
            'total_order_pending',
            'total_order_complete',
            'total_order_paid',
            'total_order_not_paid',
            'total_revenue'
        ));
    } else {
        $products = Product::paginate(3);
    }
}
```

```

$user = Auth::user();
return view('home.userpage', compact('products'));
}
}

```

## Sweet-Alerts in Laravel

- Adding Sweet-Alerts in Laravel:
  - composer require realrashid/sweetalert
  - php artisan sweetalert:publish
  - Add this to the view where you want to show the message:  
-@include('sweetalert::alert')
  - Go to the HomeController or/and AdminController

```

use RealRashid\SweetAlert\Facades\Alert;
public function add_user(Request $request){
$user = new User;
$user->name = $request->name;
$user->email = $request->email;
$user->phone = $request->phone;
$user->address = $request->address;
$user->password = bcrypt($request->password);
$user->email_verified_at=now();
$user->usertype = $request->usertype;
$user->save();
// first type
Alert::success('User Added Success','We have added user to the list of
users');
//second type
Alert::warning('User not Added Success','We haven't added user to the list
of users');
//third type
Alert::info('any info you want',' of users');
return redirect()->route('view_users');
}

```