

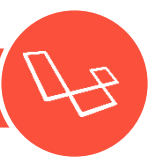


FileUpload DB

Laravel Notes

File Upload in Laravel Functions

- The **getClientOriginalName()** method used to get the actual name of the file when it was uploaded.
- The **getFilename()** method used to get temporary file name given to our file on temporary location.
- The **getRealPath()** method used to get current location of uploaded file.
- The **getClientSize()** method used to get the size of the uploaded file in bytes.
- The **getClientMimeType()** method used to get the mime type of uploaded file.
- The **guessClientExtension()** method used to get extension of uploaded file.
- The **move()** method used to move uploaded file to target location. First parameter is the target or destination directory second parameter is name of the file.

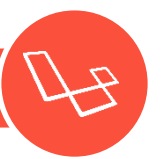


| File Upload in Laravel DB



Steps

- Create New Project
- DB Connection

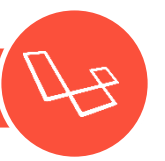


File Config

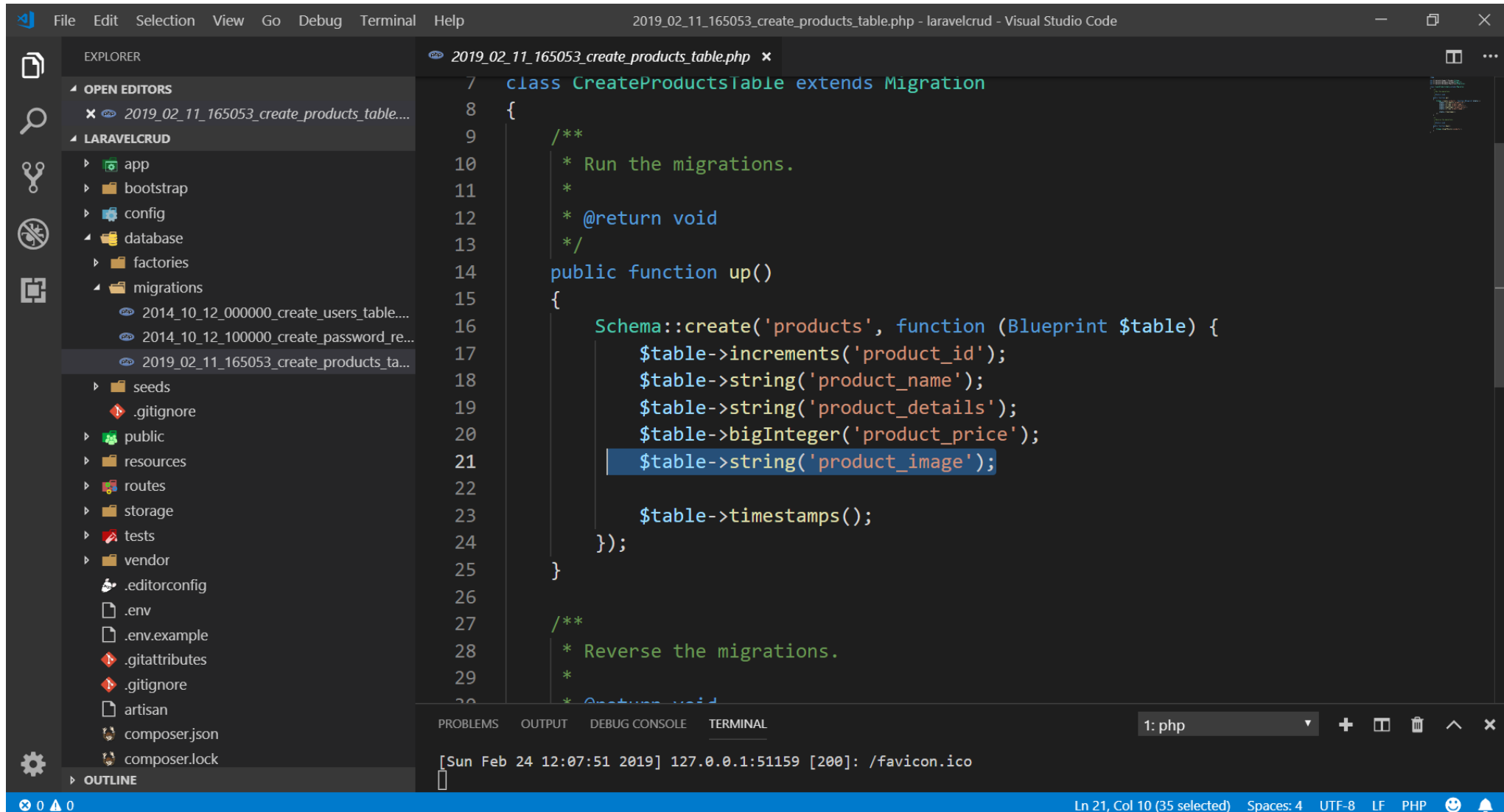
```
filesystems.php - laravelcrud - Visual Studio Code

FILED SYSTEMS
38 | // Have been setup for each driver as an example of the required options.
39 |
40 | Supported Drivers: "local", "ftp", "sftp", "s3", "rackspace"
41 |
42 | */
43 |
44 | 'disks' => [
45 |
46 |     'local' => [
47 |         'driver' => 'local',
48 |         'root' => storage_path('app'),
49 |     ],
50 |
51 |     'public' => [
52 |         'driver' => 'local',
53 |         'root' => storage_path('app/public'),
54 |         'url' => env('APP_URL').'/storage',
55 |         'visibility' => 'public',
56 |     ],
57 |
58 |     's3' => [
59 |         'driver' => 's3',
60 |         'key' => env('AWS_ACCESS_KEY_ID'),
61 |         'secret' => env('AWS_SECRET_ACCESS_KEY'),
62 |     ],
63 | ],
```

[Thu Feb 21 23:47:30 2019] 127.0.0.1:55811 [200]: /favicon.ico



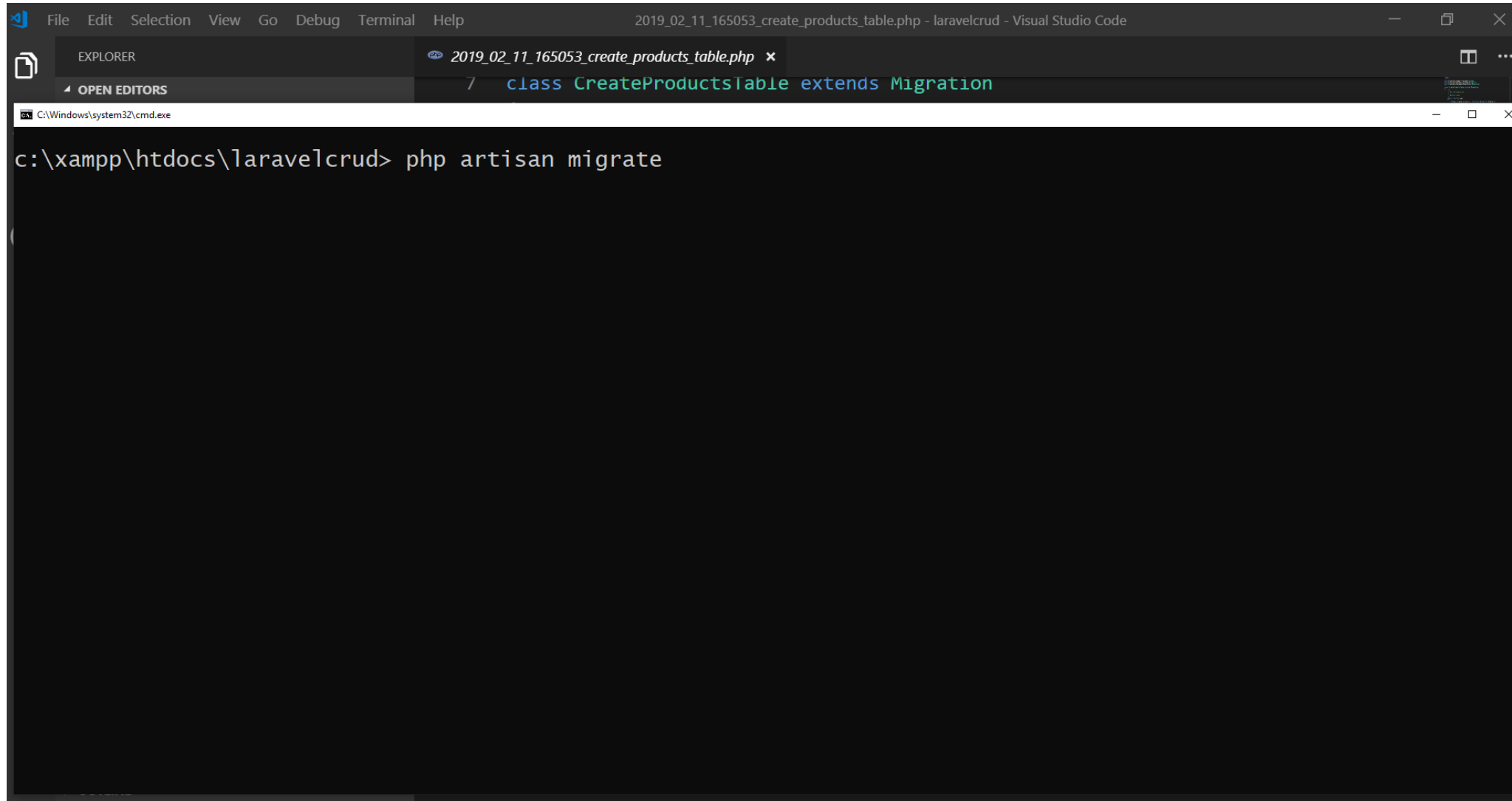
Db Migration File Add Product Image Field



```
2019_02_11_165053_create_products_table.php
7  class CreateProductsTable extends Migration
8  {
9      /**
10       * Run the migrations.
11       *
12       * @return void
13       */
14     public function up()
15     {
16         Schema::create('products', function (Blueprint $table) {
17             $table->increments('product_id');
18             $table->string('product_name');
19             $table->string('product_details');
20             $table->bigInteger('product_price');
21             $table->string('product_image');
22
23             $table->timestamps();
24         });
25     }
26
27     /**
28      * Reverse the migrations.
29      *
30      * @return void
31      */
32     public function down()
33     {
34         Schema::dropIfExists('products');
35     }
36 }
```



Migrate Database



The image shows a Visual Studio Code editor window with a file named `2019_02_11_165053_create_products_table.php` open. The file content shows a PHP class `CreateProductsTable` extending the `Migration` class. Below the editor, a terminal window is open with the command `php artisan migrate` entered at the prompt `c:\xampp\htdocs\laravelcrud>`.

```
2019_02_11_165053_create_products_table.php - laravelcrud - Visual Studio Code
```

EXPLORER

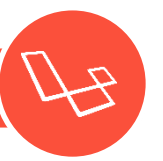
2019_02_11_165053_create_products_table.php x

OPEN EDITORS

/ class CreateProductsTable extends Migration

C:\Windows\system32\cmd.exe

```
c:\xampp\htdocs\laravelcrud> php artisan migrate
```

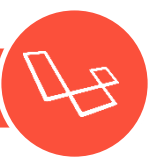


Database is Ready

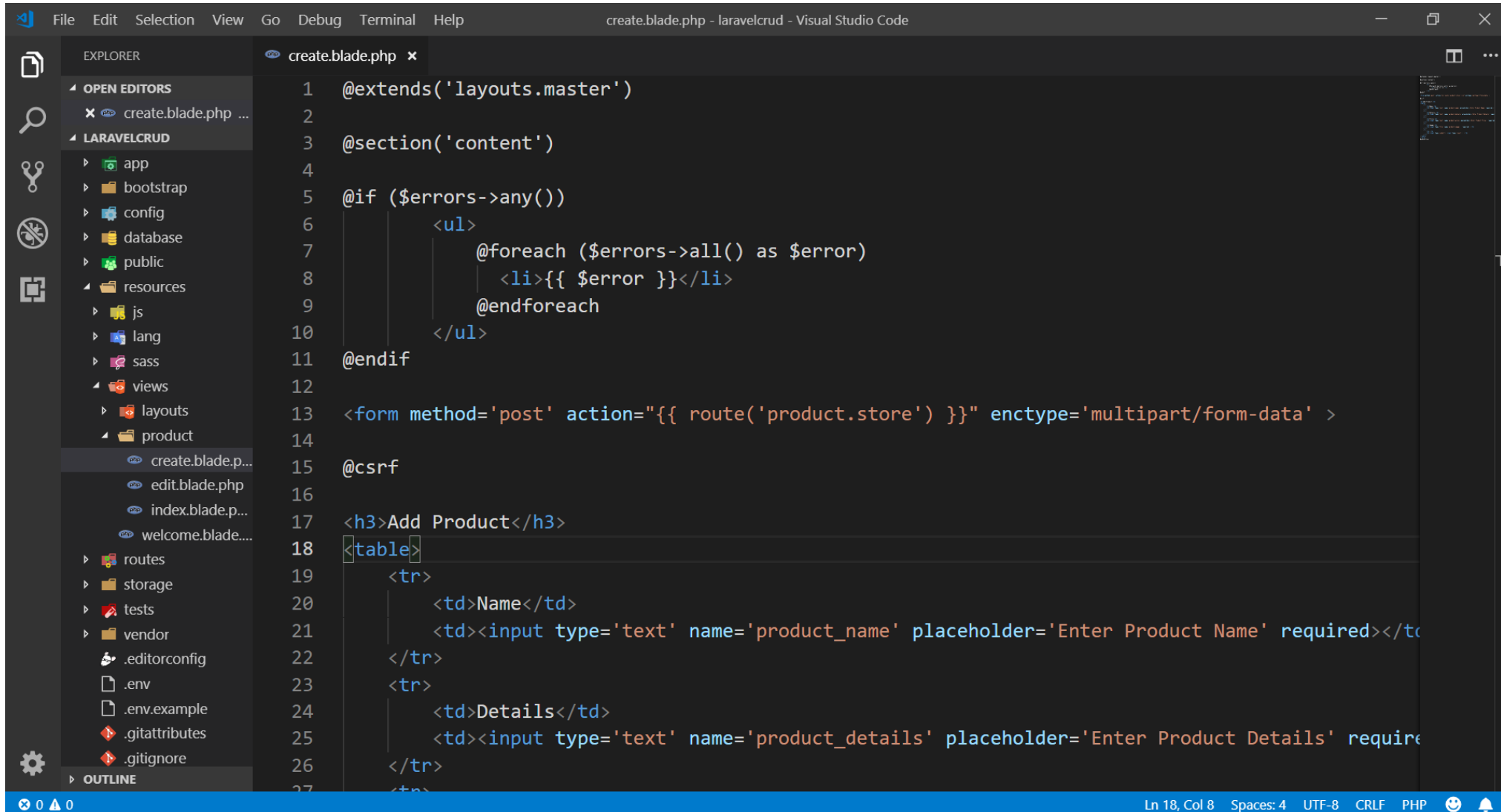
The screenshot shows the phpMyAdmin interface. The left sidebar displays a tree view of databases and tables, with 'laravelcrud' selected. The main panel shows the 'Table structure' view for the 'products' table. The table has 7 columns: product_id (int(10), UNSIGNED, AUTO_INCREMENT), product_name (varchar(191), utf8mb4_unicode_ci), product_details (varchar(191), utf8mb4_unicode_ci), product_price (bigint(20)), product_image (varchar(191), utf8mb4_unicode_ci), created_at (timestamp), and updated_at (timestamp). Below the table structure, there are options to 'Check all', 'With selected', 'Browse', 'Change', 'Drop', 'Primary', 'Unique', 'Index', and 'Add to central columns'. At the bottom, there is an 'Indexes' section showing a PRIMARY index on the 'product_id' column.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	product_id	int(10)		UNSIGNED	No	None		AUTO_INCREMENT	
2	product_name	varchar(191)	utf8mb4_unicode_ci		No	None			
3	product_details	varchar(191)	utf8mb4_unicode_ci		No	None			
4	product_price	bigint(20)			No	None			
5	product_image	varchar(191)	utf8mb4_unicode_ci		No	None			
6	created_at	timestamp			Yes	NULL			
7	updated_at	timestamp			Yes	NULL			

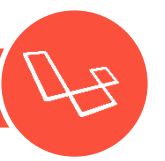
Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit	PRIMARY	BTREE	Yes	No	product_id	2	A	No	



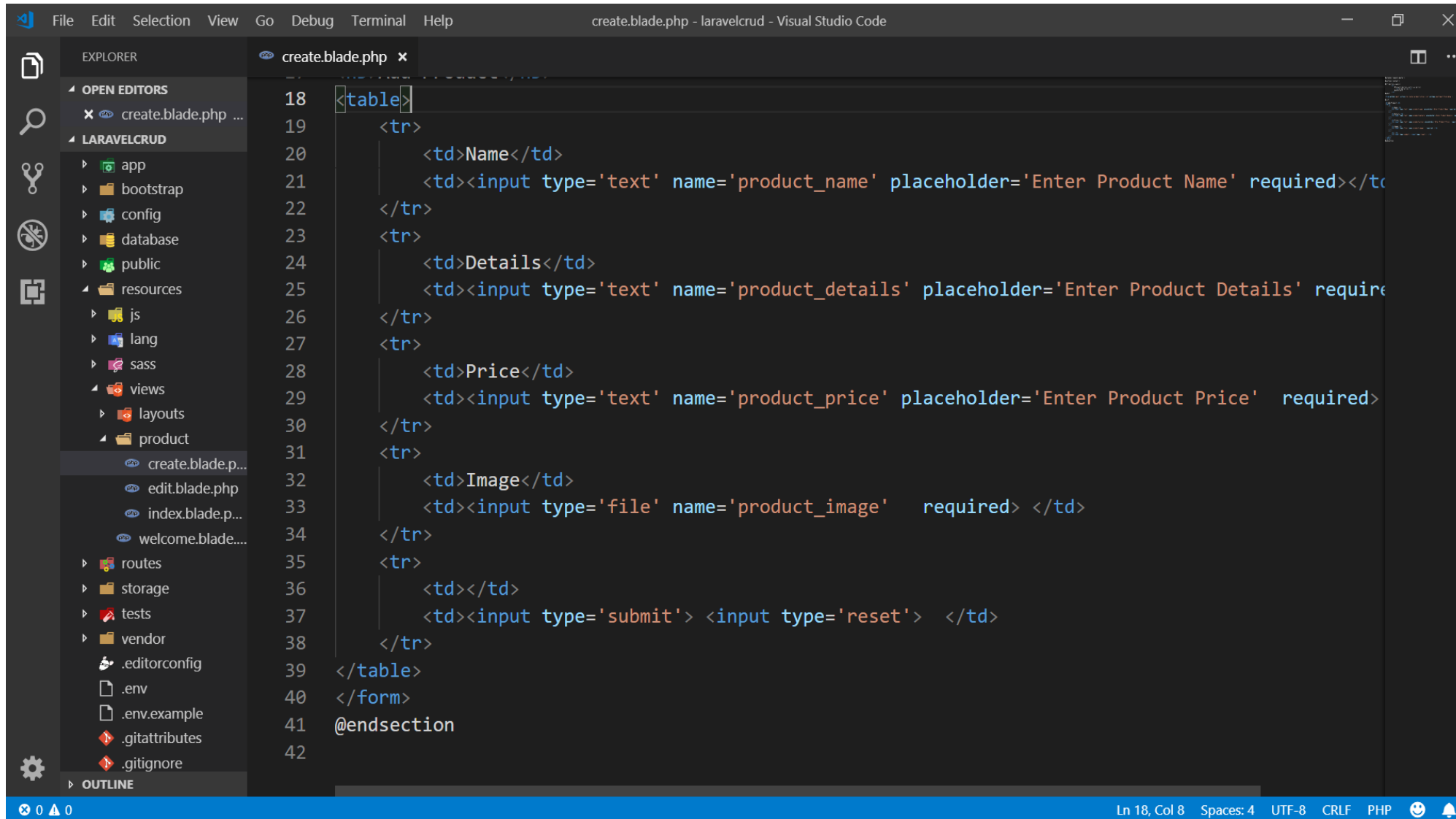
Create New Blade for Add Record



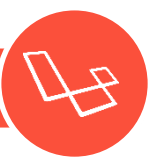
```
1 @extends('layouts.master')
2
3 @section('content')
4
5 @if ($errors->any())
6     <ul>
7         @foreach ($errors->all() as $error)
8             <li>{{ $error }}</li>
9         @endforeach
10    </ul>
11 @endif
12
13 <form method='post' action="{{ route('product.store') }}" enctype='multipart/form-data' >
14
15 @csrf
16
17 <h3>Add Product</h3>
18 <table>
19     <tr>
20         <td>Name</td>
21         <td><input type='text' name='product_name' placeholder='Enter Product Name' required></td>
22     </tr>
23     <tr>
24         <td>Details</td>
25         <td><input type='text' name='product_details' placeholder='Enter Product Details' required>
26     </tr>
27 </table>
```



Conti..



```
18 <table>
19   <tr>
20     <td>Name</td>
21     <td><input type='text' name='product_name' placeholder='Enter Product Name' required></td>
22   </tr>
23   <tr>
24     <td>Details</td>
25     <td><input type='text' name='product_details' placeholder='Enter Product Details' required>
26   </tr>
27   <tr>
28     <td>Price</td>
29     <td><input type='text' name='product_price' placeholder='Enter Product Price' required>
30   </tr>
31   <tr>
32     <td>Image</td>
33     <td><input type='file' name='product_image' required> </td>
34   </tr>
35   <tr>
36     <td></td>
37     <td><input type='submit'> <input type='reset'> </td>
38   </tr>
39 </table>
40 </form>
41 @endsection
42
```



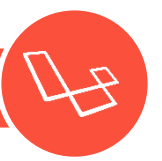


Laravel CURD

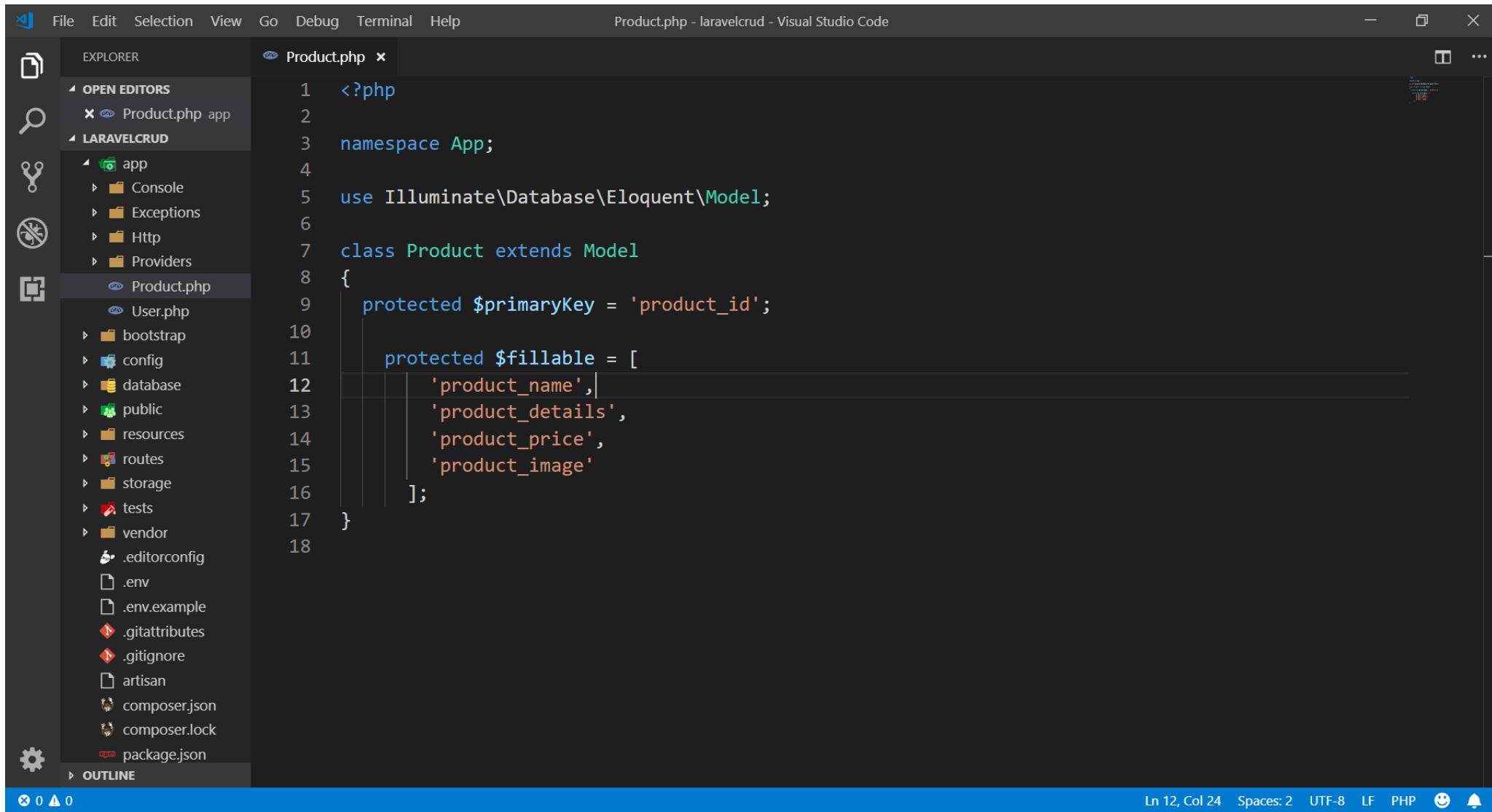
[Add](#) | [View](#)

Add Product

Name	<input type="text" value="Enter Product Name"/>
Details	<input type="text" value="Enter Product Details"/>
Price	<input type="text" value="Enter Product Price"/>
Image	<input type="button" value="Choose File"/> No file chosen
	<input type="button" value="Submit"/> <input type="button" value="Reset"/>



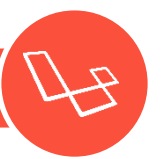
Define Model



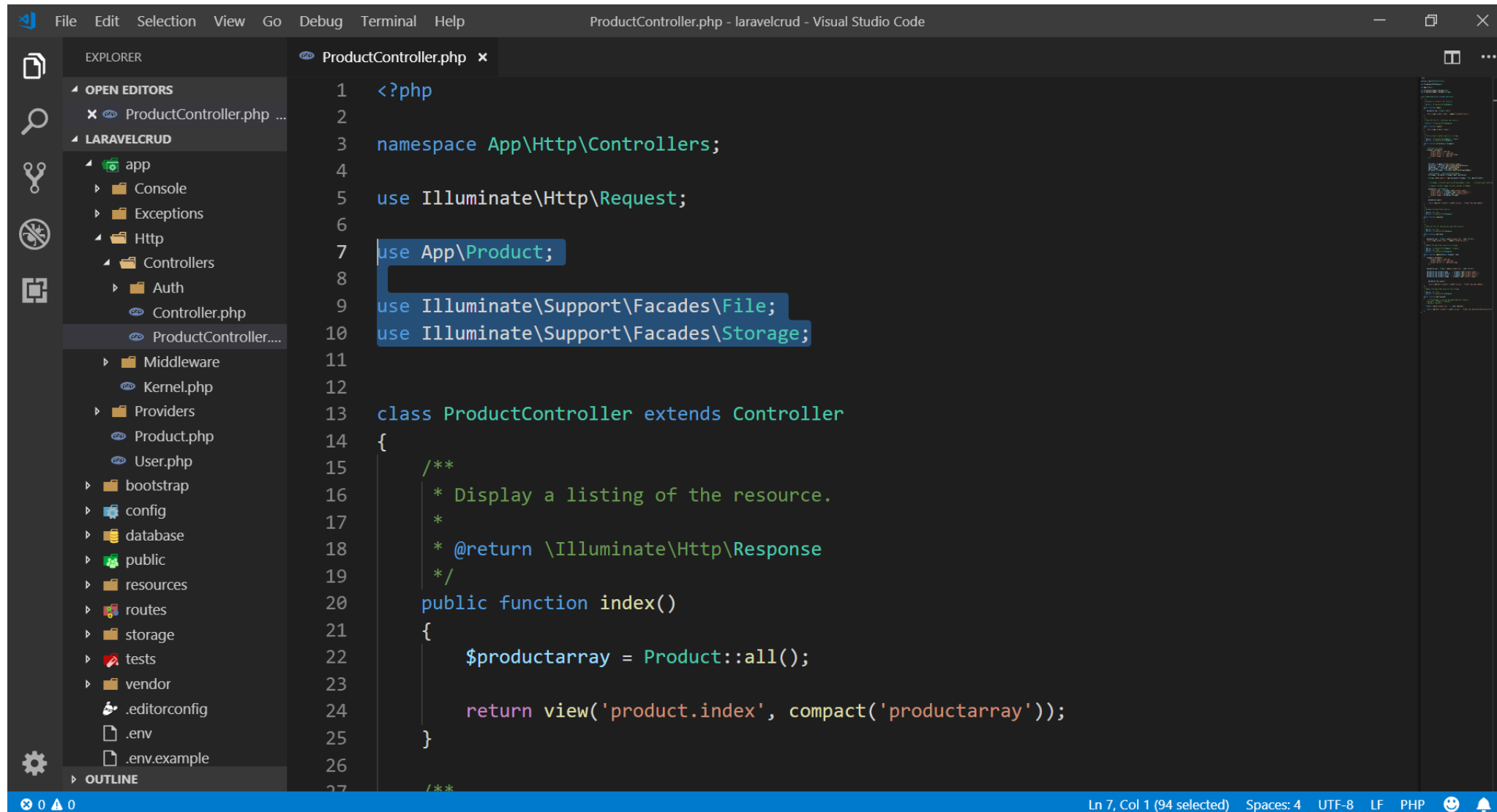
The screenshot shows the Visual Studio Code editor with a file explorer on the left and a code editor on the right. The file explorer shows the project structure, including the 'app' directory and the 'Product.php' file. The code editor displays the following PHP code:

```
1 <?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class Product extends Model
8 {
9     protected $primaryKey = 'product_id';
10
11     protected $fillable = [
12         'product_name',
13         'product_details',
14         'product_price',
15         'product_image'
16     ];
17 }
18
```

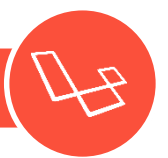
The status bar at the bottom indicates the current position is Line 12, Column 24, with 2 spaces, UTF-8 encoding, LF line endings, and the PHP language mode.



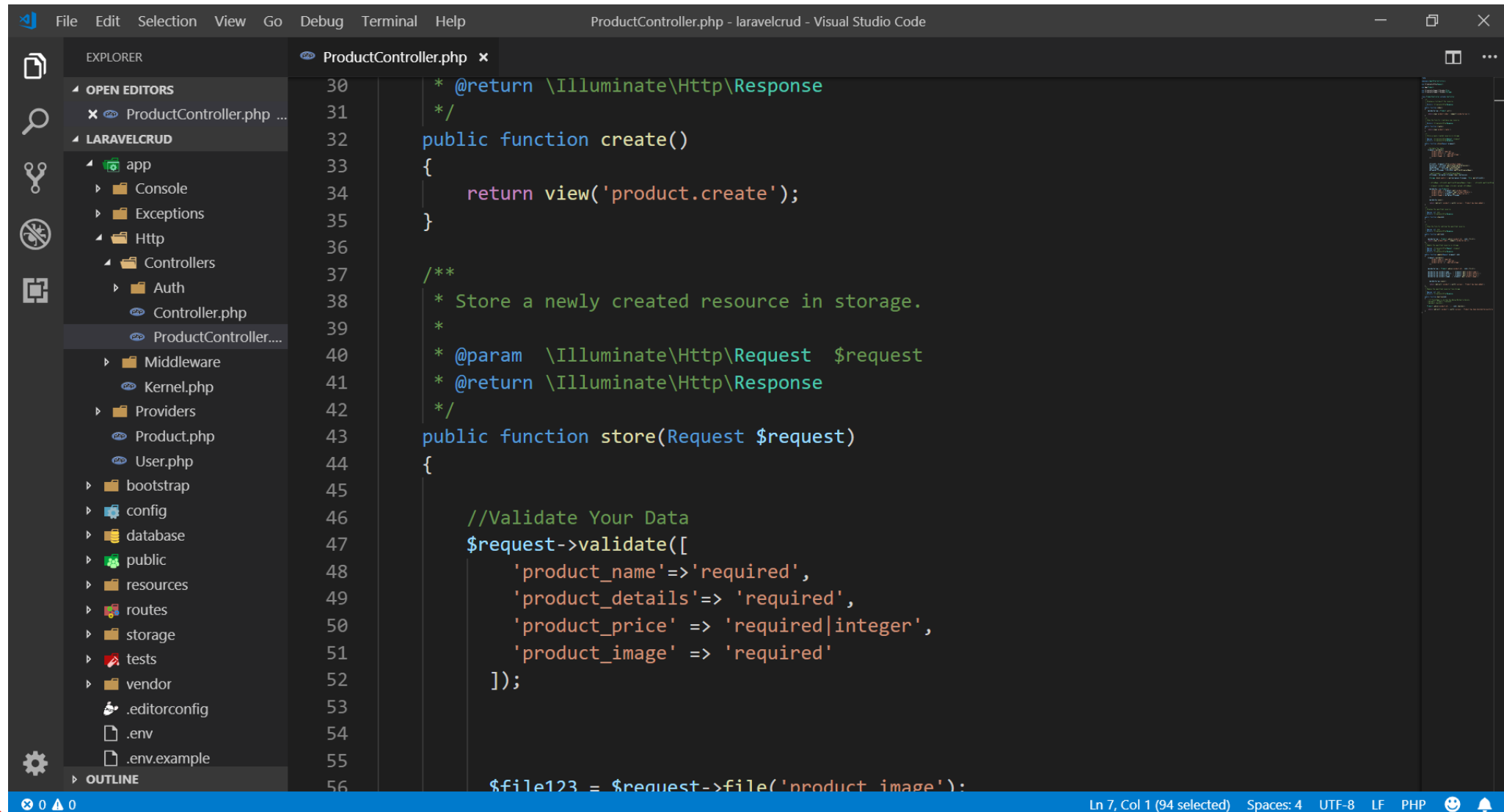
Add File and Storage Namespace



```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 use App\Product;
8
9 use Illuminate\Support\Facades\File;
10 use Illuminate\Support\Facades\Storage;
11
12
13 class ProductController extends Controller
14 {
15     /**
16      * Display a listing of the resource.
17      *
18      * @return \Illuminate\Http\Response
19      */
20     public function index()
21     {
22         $productarray = Product::all();
23
24         return view('product.index', compact('productarray'));
25     }
26
27     /**
```



Add Record Process



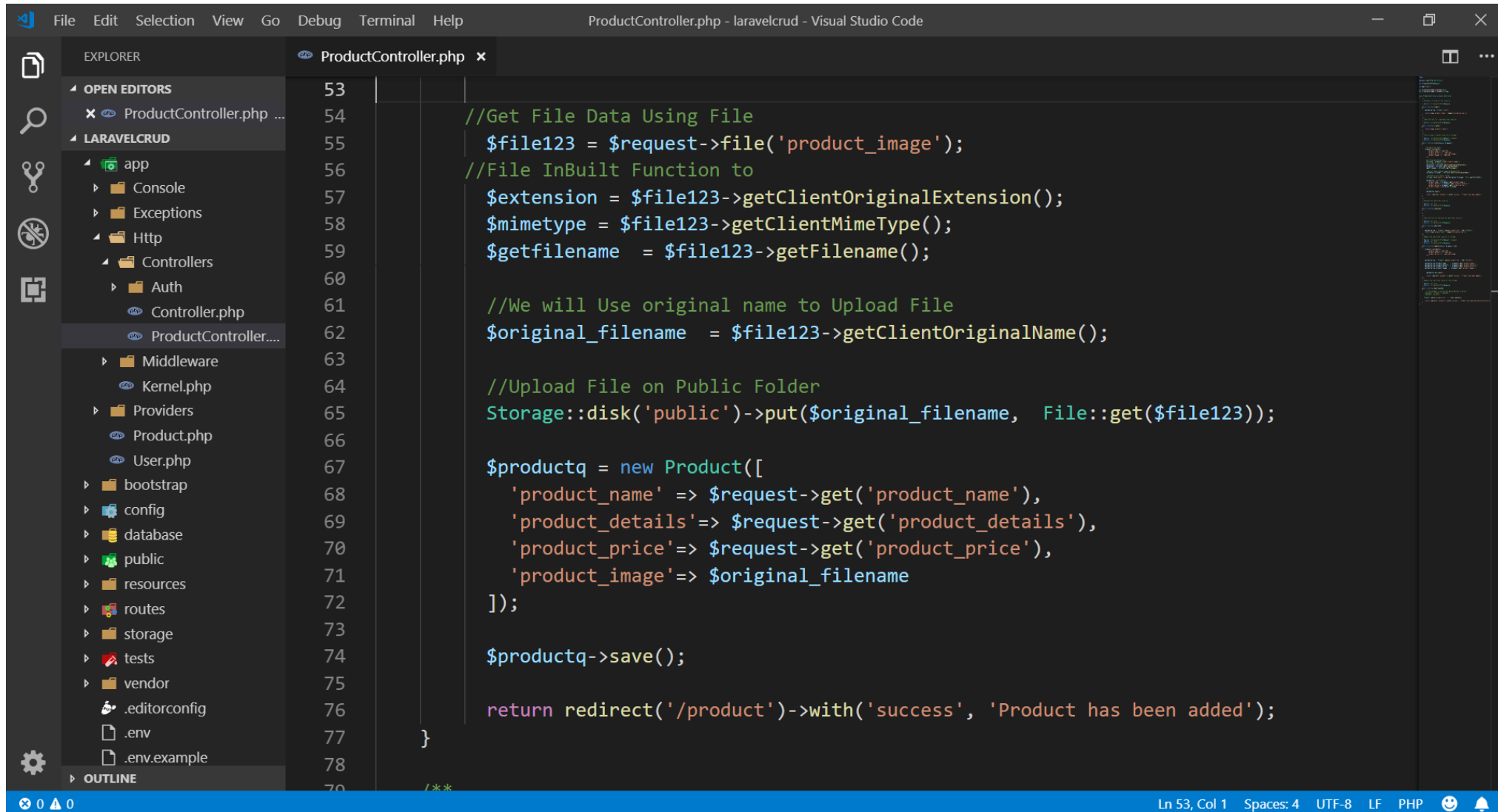
```
ProductController.php - laravelcrud - Visual Studio Code

EXPLORER
├── OPEN EDITORS
│   └── ProductController.php ...
├── LARAVELCRUD
│   ├── app
│   │   ├── Console
│   │   ├── Exceptions
│   │   └── Http
│   │       ├── Controllers
│   │       │   ├── Auth
│   │       │   ├── Controller.php
│   │       │   └── ProductController...
│   │       ├── Middleware
│   │       │   ├── Kernel.php
│   │       ├── Providers
│   │       │   ├── Product.php
│   │       │   └── User.php
│   │       ├── bootstrap
│   │       ├── config
│   │       ├── database
│   │       ├── public
│   │       ├── resources
│   │       ├── routes
│   │       ├── storage
│   │       ├── tests
│   │       └── vendor
│   │           ├── .editorconfig
│   │           ├── .env
│   │           └── .env.example
│   └── OUTLINE
└── ProductController.php x

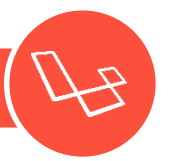
30 * @return \Illuminate\Http\Response
31 */
32 public function create()
33 {
34     return view('product.create');
35 }
36
37 /**
38  * Store a newly created resource in storage.
39  *
40  * @param \Illuminate\Http\Request $request
41  * @return \Illuminate\Http\Response
42  */
43 public function store(Request $request)
44 {
45
46     //Validate Your Data
47     $request->validate([
48         'product_name'=>'required',
49         'product_details'=>'required',
50         'product_price' => 'required|integer',
51         'product_image' => 'required'
52     ]);
53
54     $file123 = $request->file('product_image');
```



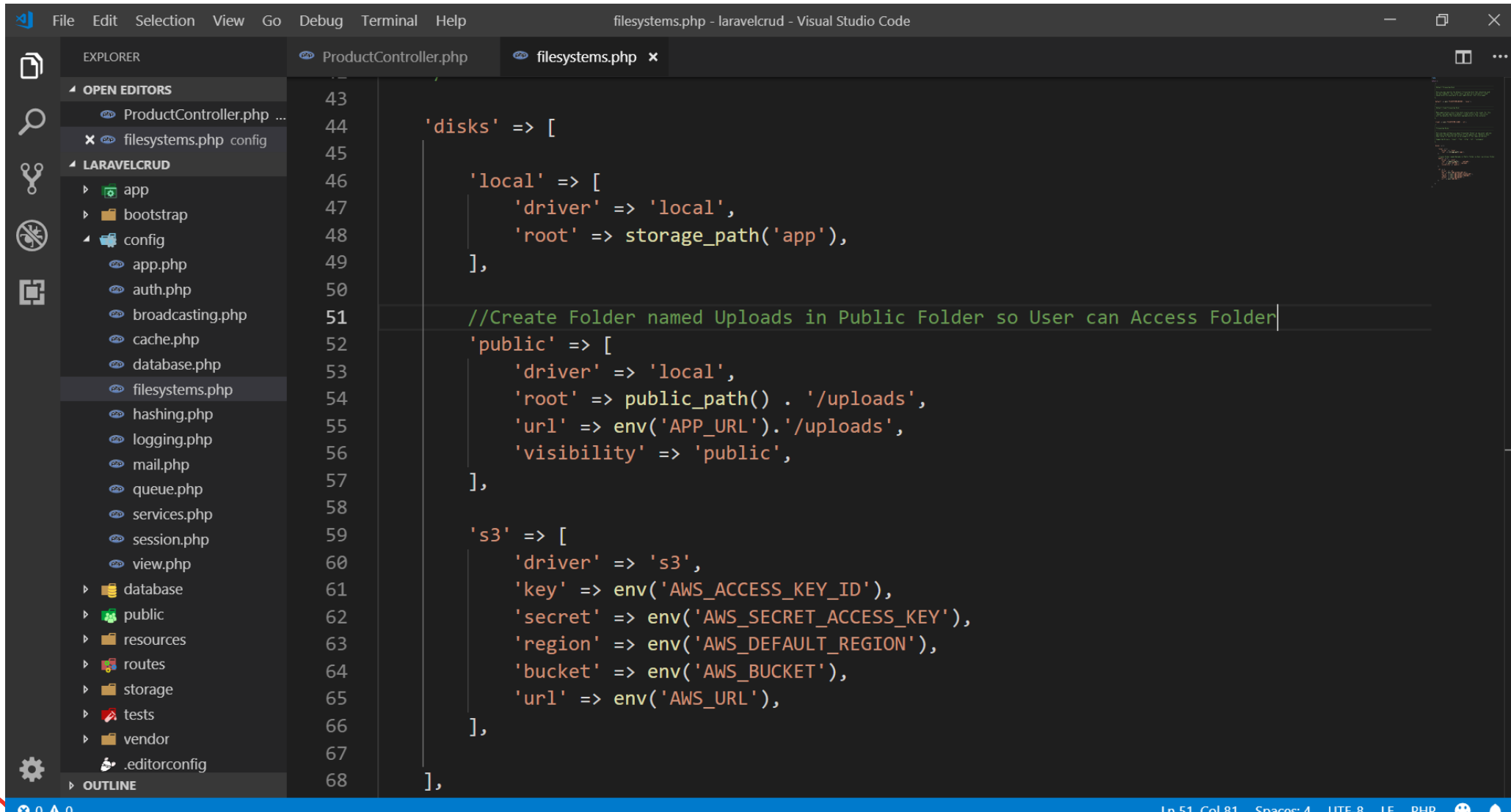
Store File in Folder



```
ProductController.php
53
54 //Get File Data Using File
55 $file123 = $request->file('product_image');
56 //File InBuilt Function to
57 $extension = $file123->getClientOriginalExtension();
58 $mimetype = $file123->getClientMimeType();
59 $getfilename = $file123->getFilename();
60
61 //We will Use original name to Upload File
62 $original_filename = $file123->getClientOriginalName();
63
64 //Upload File on Public Folder
65 Storage::disk('public')->put($original_filename, File::get($file123));
66
67 $productq = new Product([
68     'product_name' => $request->get('product_name'),
69     'product_details'=> $request->get('product_details'),
70     'product_price'=> $request->get('product_price'),
71     'product_image'=> $original_filename
72 ]);
73
74 $productq->save();
75
76 return redirect('/product')->with('success', 'Product has been added');
77
78 }
```

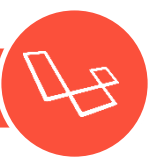


FileSystem Config.



```
43
44
45
46
47
48
49
50
51 //Create Folder named Uploads in Public Folder so User can Access Folder
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
```

```
'disks' => [
    'local' => [
        'driver' => 'local',
        'root' => storage_path('app'),
    ],
    'public' => [
        'driver' => 'local',
        'root' => public_path() . '/uploads',
        'url' => env('APP_URL') . '/uploads',
        'visibility' => 'public',
    ],
    's3' => [
        'driver' => 's3',
        'key' => env('AWS_ACCESS_KEY_ID'),
        'secret' => env('AWS_SECRET_ACCESS_KEY'),
        'region' => env('AWS_DEFAULT_REGION'),
        'bucket' => env('AWS_BUCKET'),
        'url' => env('AWS_URL'),
    ],
],
```



Test Code

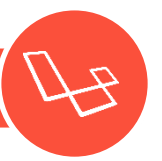


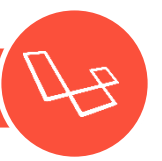
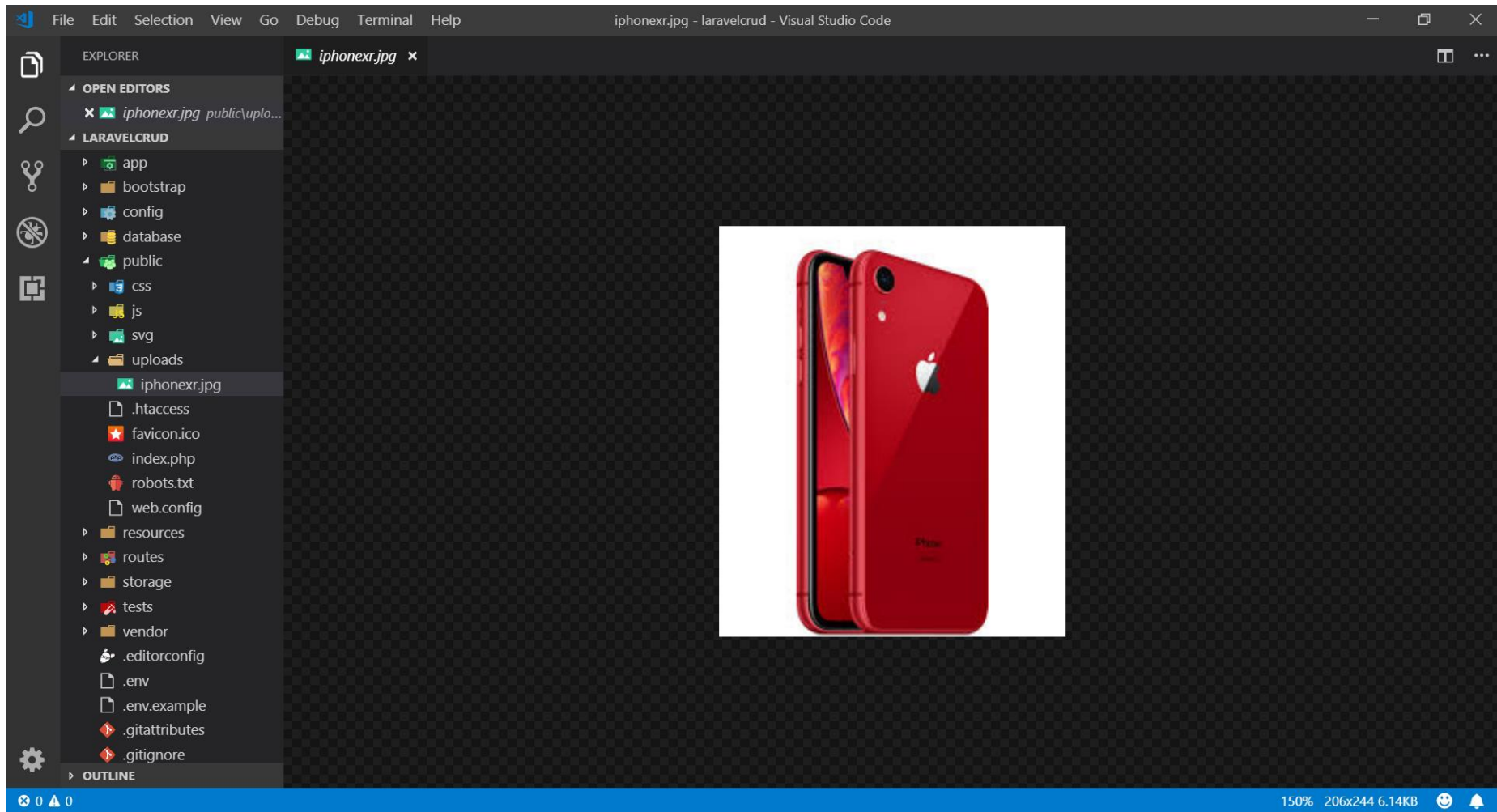
Laravel CURD

[Add](#) | [View](#)

Add Product

Name	<input type="text" value="iPhone"/>
Details	<input type="text" value="iPhone XR"/>
Price	<input type="text" value="85000"/>
Image	<input type="button" value="Choose File"/> <input type="text" value="iphonexr.jpg"/>
	<input type="button" value="Submit"/> <input type="button" value="Reset"/>





localhost / 127.0.0.1 / laravelcrud x 127.0.0.1:8000/product

localhost/phpmyadmin/sql.php?db=laravelcrud&table=products&pos=0

phpMyAdmin

Recent Favorites

- New
- 111
- devicefinder
- information_schema
- laravelcrud
 - New
 - migrations
 - password_resets
 - products
 - users
- mysql
- performance_schema
- phpmyadmin
- pmit
- pmts_db
- sample
- test
- testlaravel
- tours

Server: 127.0.0.1 » Database: laravelcrud » Table: products

Browse Structure SQL Search Insert Export Import Privileges More

0 - 0 (1 total, Query took 0.0015 seconds.)

Products`

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

Number of rows: 25 Filter rows: Search this table

	product_id	product_name	product_details	product_price	product_image	created_at	updated_at
Copy Delete	1	iPhone	iPhone XR	85000	iphonexr.jpg	2019-02-24 06:52:06	2019-02-24 06:52:06

all With selected: Edit Copy Delete Export

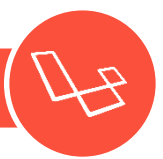
Number of rows: 25 Filter rows: Search this table

Operations

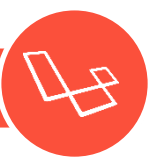
to clipboard Export Display chart Create view

This SQL query

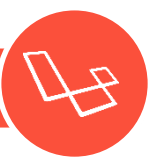
☒ Console ☐ Let every user access this bookmark

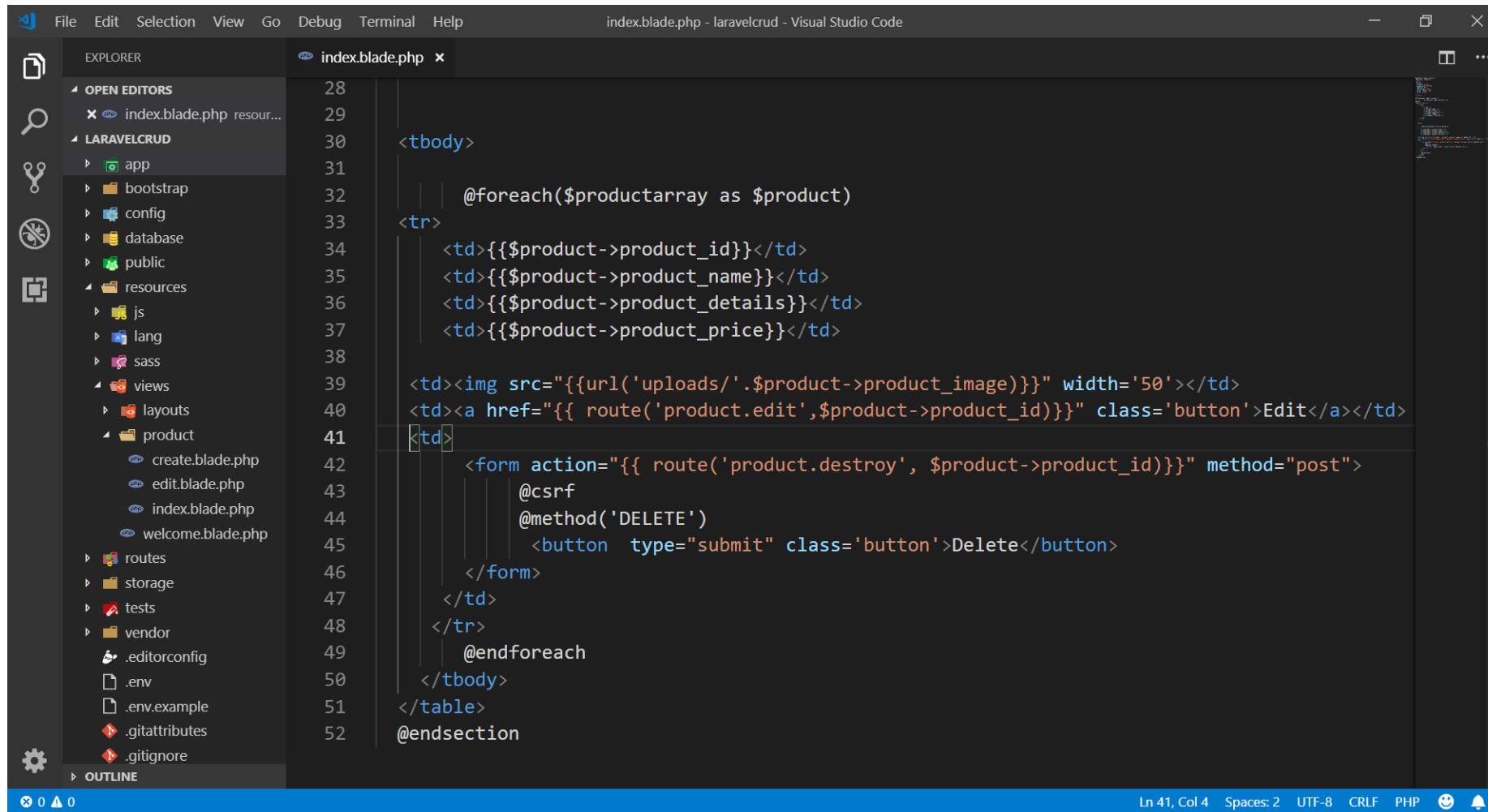


```
ProductController.php x
9  use Illuminate\Support\Facades\File;
10 use Illuminate\Support\Facades\Storage;
11
12
13 class ProductController extends Controller
14 {
15     /**
16      * Display a listing of the resource.
17      *
18      * @return \Illuminate\Http\Response
19      */
20     public function index()
21     {
22         $productarray = Product::all();
23
24         return view('product.index', compact('productarray'));
25     }
26
27     /**
28      * Show the form for creating a new resource.
29      *
30      * @return \Illuminate\Http\Response
31      */
32     public function create()
33     {
34         return view('product.create');
35     }
36 }
```

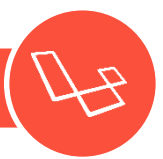


```
1 @extends('layouts.master')
2 @section('content')
3 <style>
4 .button {
5     display: block;
6     background: #f0f0f0;
7     padding: 5px;
8     text-align: center;
9     border-radius: 5px;
10    color: black;
11 }
12 </style>
13
14 @if(session()->get('success'))
15     {{ session()->get('success') }}
16 @endif
17 <table border='1'>
18     <thead>
19         <tr>
20             <th>ID</th>
21             <th>Product Name</th>
22             <th>Product Price</th>
23             <th>Product Quantity</th>
24             <th>Product Image</th>
25             <th colspan="2">Action</th>
26         </tr>
27     </thead>
```





```
28
29
30 <tbody>
31
32     @foreach($productarray as $product)
33     <tr>
34         <td>{{ $product->product_id }}</td>
35         <td>{{ $product->product_name }}</td>
36         <td>{{ $product->product_details }}</td>
37         <td>{{ $product->product_price }}</td>
38
39         <td></td>
40         <td><a href="{{ route('product.edit', $product->product_id) }}" class='button'>Edit</a></td>
41         <td>
42             <form action="{{ route('product.destroy', $product->product_id) }}" method="post">
43                 @csrf
44                 @method('DELETE')
45                 <button type="submit" class='button'>Delete</button>
46             </form>
47         </td>
48     </tr>
49     @endforeach
50 </tbody>
51 </table>
52 @endsection
```




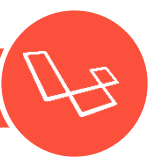
Display Code



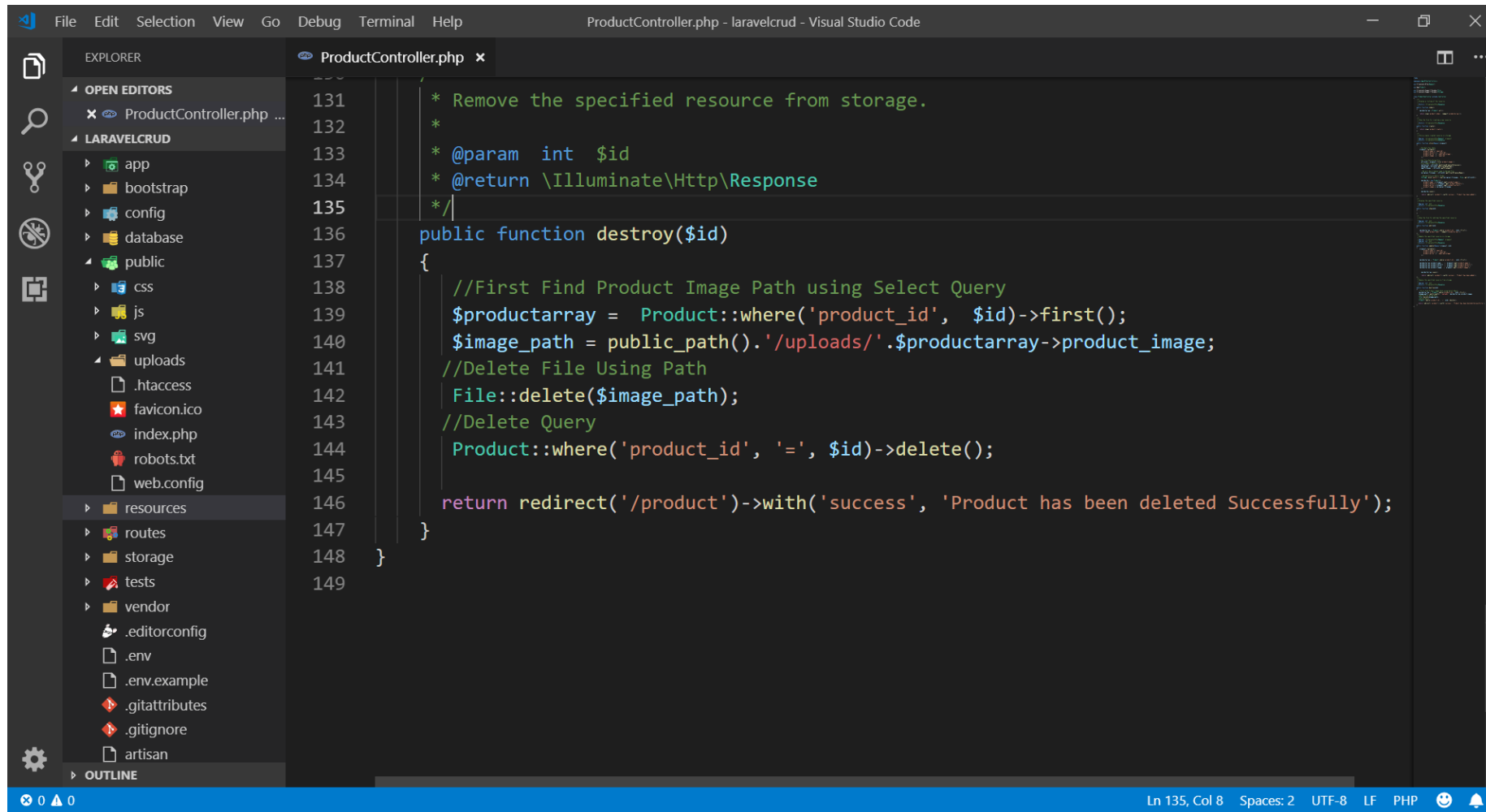
Laravel CURD

[Add](#) | [View](#) Product has been added

ID	Product Name	Product Price	Product Quantity	Product Image	Action	
1	iPhone	iPhone XR	85000		Edit	<button>Delete</button>

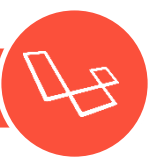


Delete Physical File



The screenshot shows the Visual Studio Code editor with the file `ProductController.php` open. The left sidebar displays the Explorer view with the project structure of a Laravel application. The main editor area shows the `destroy` method, which is responsible for deleting a product and its associated image file from storage.

```
131  * Remove the specified resource from storage.
132  *
133  * @param int $id
134  * @return \Illuminate\Http\Response
135  */
136  public function destroy($id)
137  {
138      //First Find Product Image Path using Select Query
139      $productarray = Product::where('product_id', $id)->first();
140      $image_path = public_path().'/uploads/'.$productarray->product_image;
141      //Delete File Using Path
142      File::delete($image_path);
143      //Delete Query
144      Product::where('product_id', '=', $id)->delete();
145
146      return redirect('/product')->with('success', 'Product has been deleted Successfully');
147  }
148
149
```



Get Exclusive Video Tutorials



www.apptutorials.com

<https://www.youtube.com/user/Akashtips>





Get More Details

www.akashsir.com



If You Liked It !

Rating Us Now



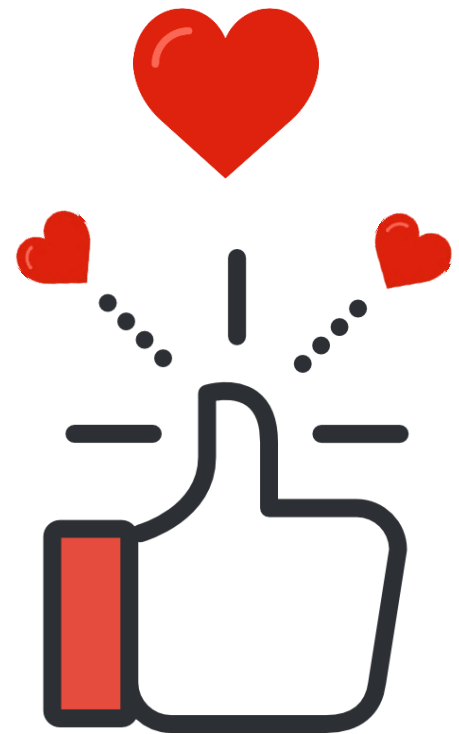
Just Dial

https://www.justdial.com/Ahmedabad/Akash-Technolabs-Navrangpura-Bus-Stop-Navrangpura/079PXX79-XX79-170615221520-S5C4_BZDET



Sulekha

<https://www.sulekha.com/akash-technolabs-navrangpura-ahmedabad-contact-address/ahmedabad>



Connect With Me



Akash Padhiyar
#AkashSir

www.akashsir.com

www.akashtechlabs.com

www.akashpadhiyar.com

www.apptutorials.com

Social Info



Akash.padhiyar



Akashpadhiyar



Akash_padhiyar



+91 99786-21654



#Akashpadhiyar

#apptutorials