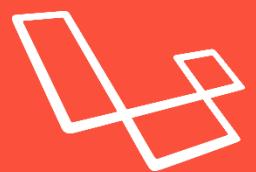


# Laravel CRUD

# Laravel Notes

# Laravel CRUD



# Open CMD

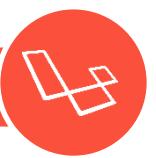
```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.17763.253]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Akash>
```



Akash Technolabs

[www.akashsir.com](http://www.akashsir.com)



# Navigate to Xampp htdocs

C:\Windows\system32\cmd.exe

Microsoft Windows [Version 10.0.17763.253]  
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Akash>cd c:/xampp/htdocs



Akash Technolabs

[www.akashsir.com](http://www.akashsir.com)



# Download Laravel and Create New Project

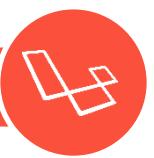
```
c:\Windows\system32\cmd.exe - composer create-project --prefer-dist laravel/laravel laravelcrud
```

```
c:\xampp\htdocs>composer create-project --prefer-dist laravel/laravel laravelcrud
```



Akash Technolabs

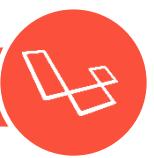
[www.akashsir.com](http://www.akashsir.com)



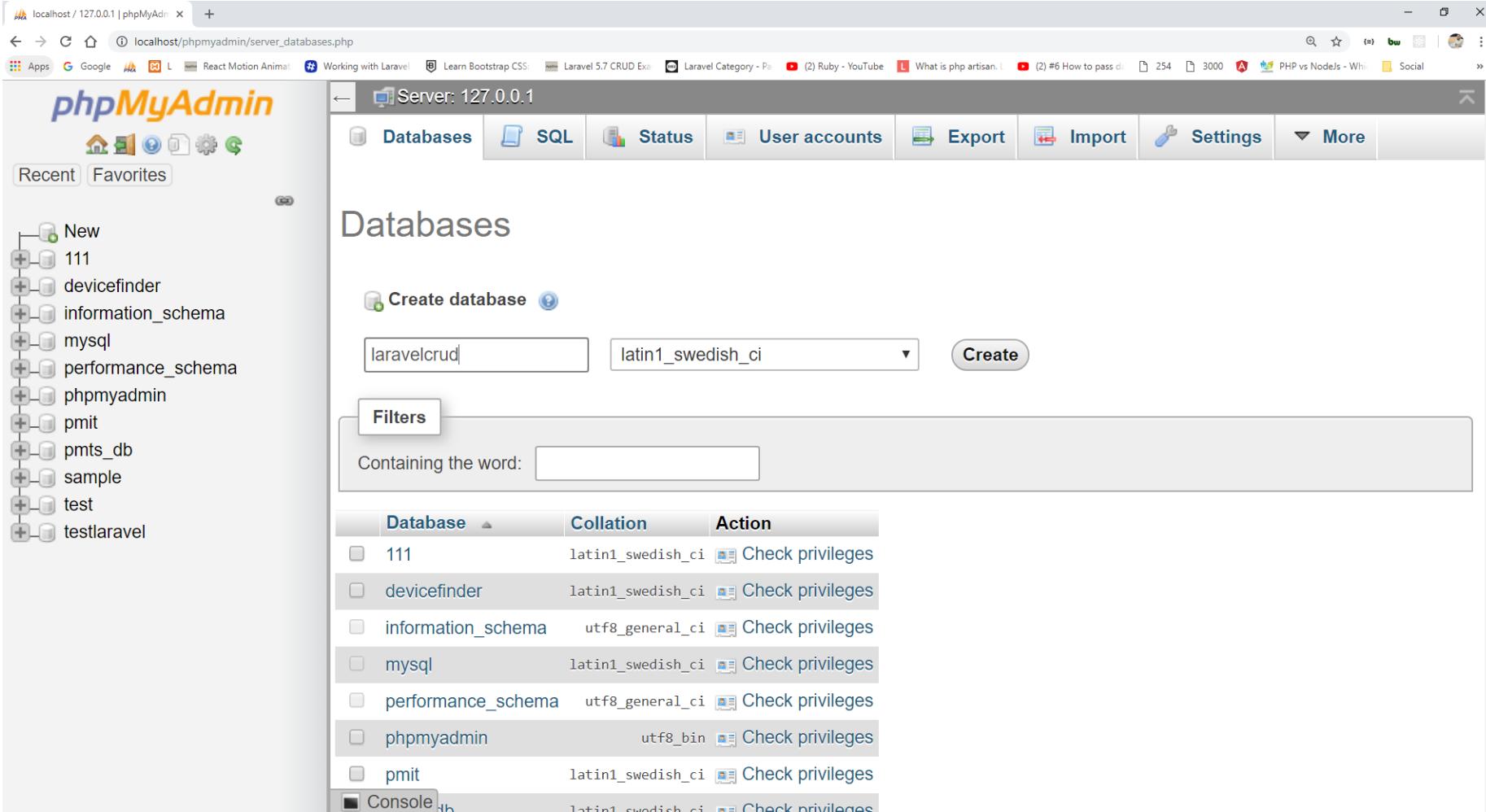
# Wait for 10 to 20 Minutes

```
C:\Windows\system32\cmd.exe
psy/psysh suggests installing ext-pdo-sqlite (The doc command requires SQLite to work.)
psy/psysh suggests installing hoa/console (A pure PHP readline implementation. You'll want this
tall doesn't already support readline or libedit.)
filp/whoops suggests installing whoops/soap (Formats errors as SOAP responses)
sebastian/global-state suggests installing ext-uopz (*)
sebastian/environment suggests installing ext-posix (*)
phpunit/php-code-coverage suggests installing ext-xdebug (^2.6.0)
phpunit/phpunit suggests installing phpunit/php-invoker (^2.0)
phpunit/phpunit suggests installing ext-soap (*)
phpunit/phpunit suggests installing ext-xdebug (*)
Writing lock file
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
Discovered Package: beyondcode/laravel-dump-server
Discovered Package: fideloper/proxy
Discovered Package: laravel/nexmo-notification-channel
Discovered Package: laravel/slack-notification-channel
Discovered Package: laravel/tinker
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/collision
Package manifest generated successfully.
> @php artisan key:generate --ansi
Application key set successfully.

c:\xampp\htdocs>
c:\xampp\htdocs>
```

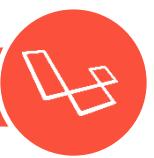


# Create New Database in Phpmyadmin

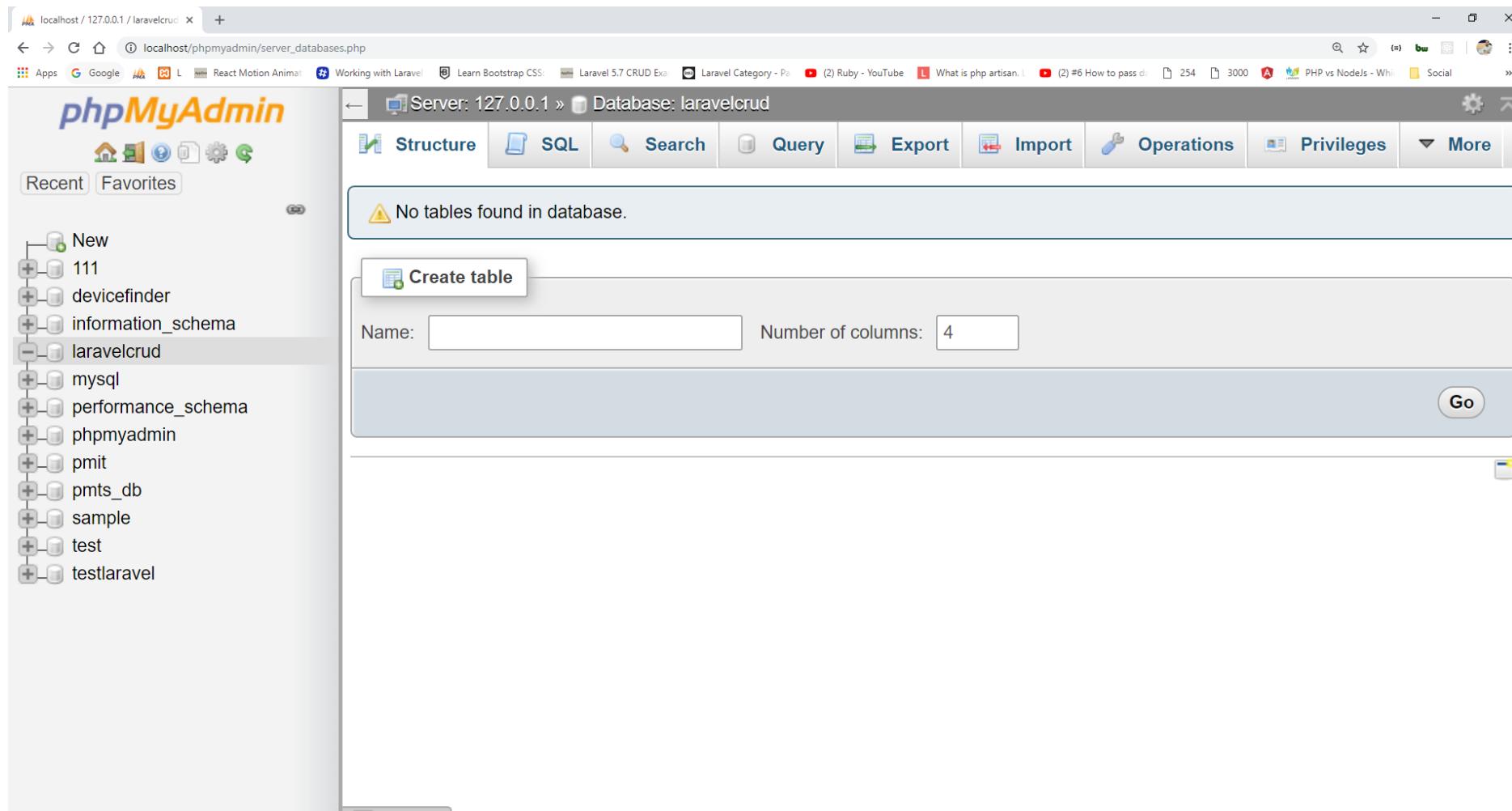


The screenshot shows the phpMyAdmin interface on a web browser. The title bar says "localhost / 127.0.0.1 | phpMyAdmin". The main menu has tabs for Databases, SQL, Status, User accounts, Export, Import, Settings, and More. On the left, there's a sidebar with "Recent" and "Favorites" buttons, and a tree view of databases including "New", "111", "devicefinder", "information\_schema", "mysql", "performance\_schema", "phpmyadmin", "pmit", "pmts\_db", "sample", "test", and "testlaravel". The main content area is titled "Databases" and shows a "Create database" form with "laravelcrud" in the input field and "latin1\_swedish\_ci" in the dropdown for collation. Below it is a "Filters" section with a search box. A table lists existing databases with their collations and checkboxes for "Check privileges":

Database	Collation	Action
111	latin1_swedish_ci	<input type="checkbox"/> Check privileges
devicefinder	latin1_swedish_ci	<input type="checkbox"/> Check privileges
information_schema	utf8_general_ci	<input type="checkbox"/> Check privileges
mysql	latin1_swedish_ci	<input type="checkbox"/> Check privileges
performance_schema	utf8_general_ci	<input type="checkbox"/> Check privileges
phpmyadmin	utf8_bin	<input type="checkbox"/> Check privileges
pmit	latin1_swedish_ci	<input type="checkbox"/> Check privileges
Console	latin1_swedish_ci	<input checked="" type="checkbox"/> Check privileges

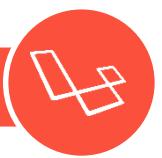


# Database Created

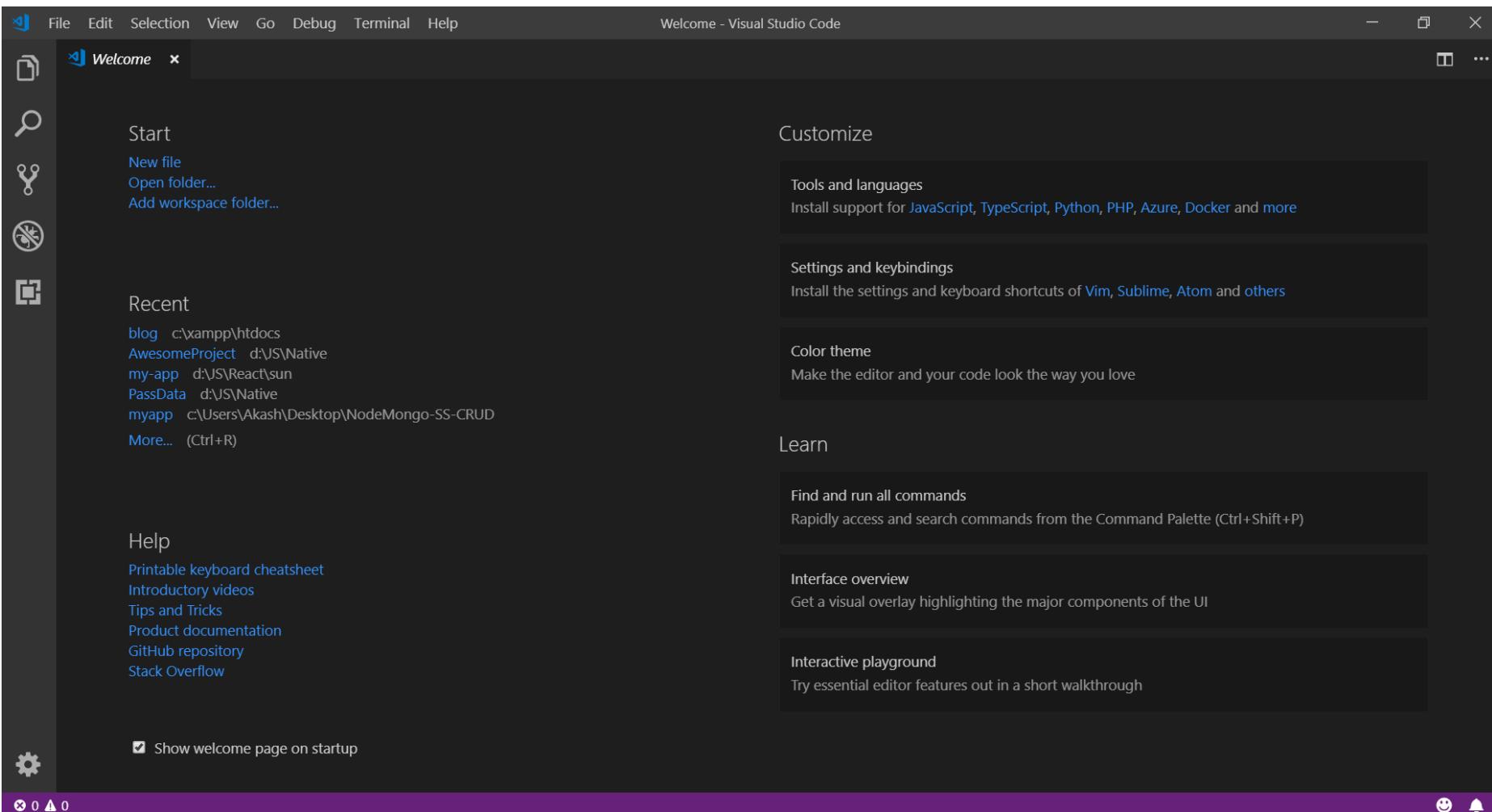


Akash Technolabs

[www.akashsir.com](http://www.akashsir.com)

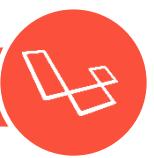


# Open VS Code Editor

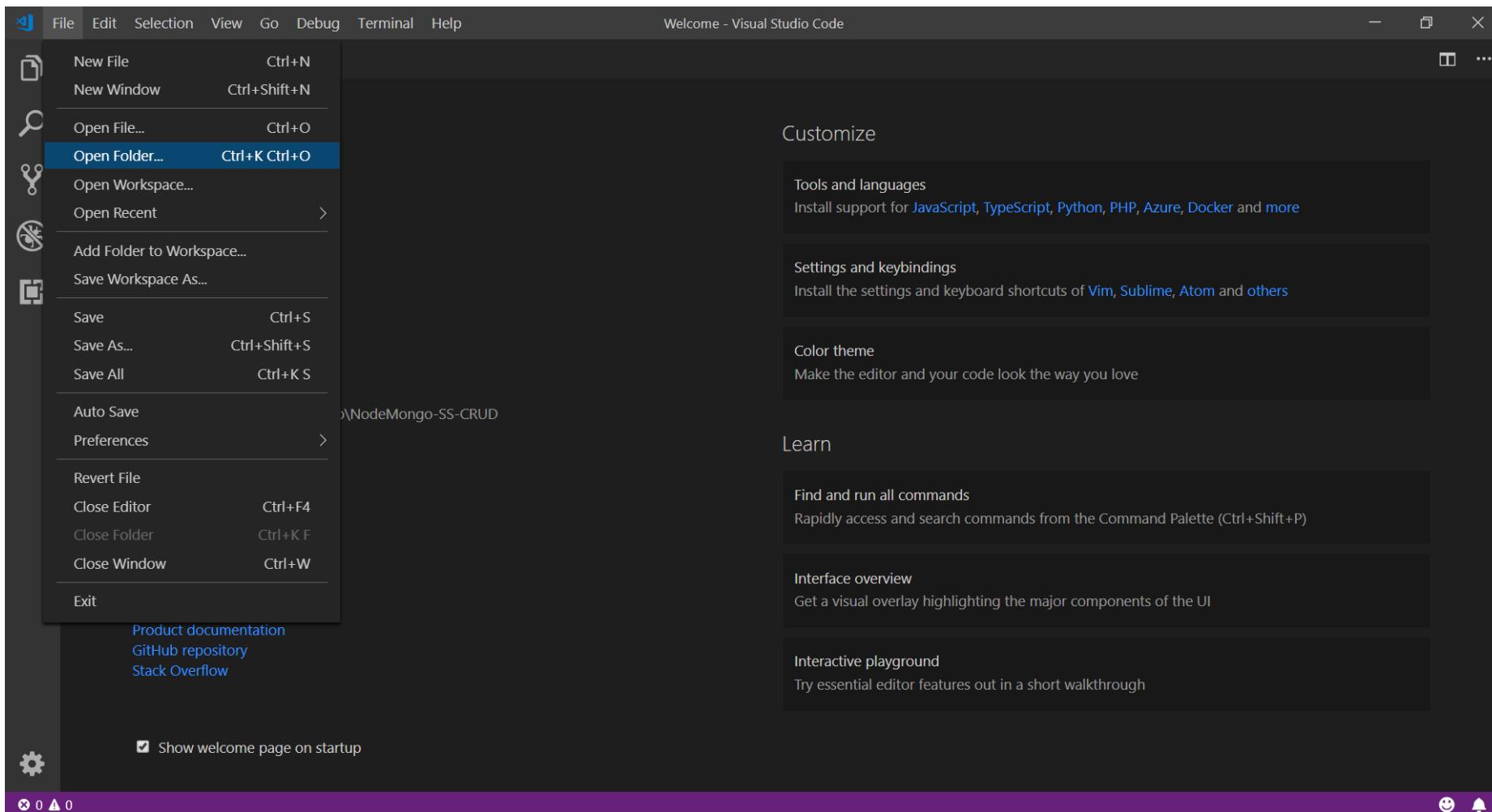


Akash Technolabs

[www.akashsir.com](http://www.akashsir.com)

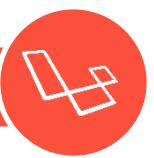


# Open Project Folder in VS Code Editor

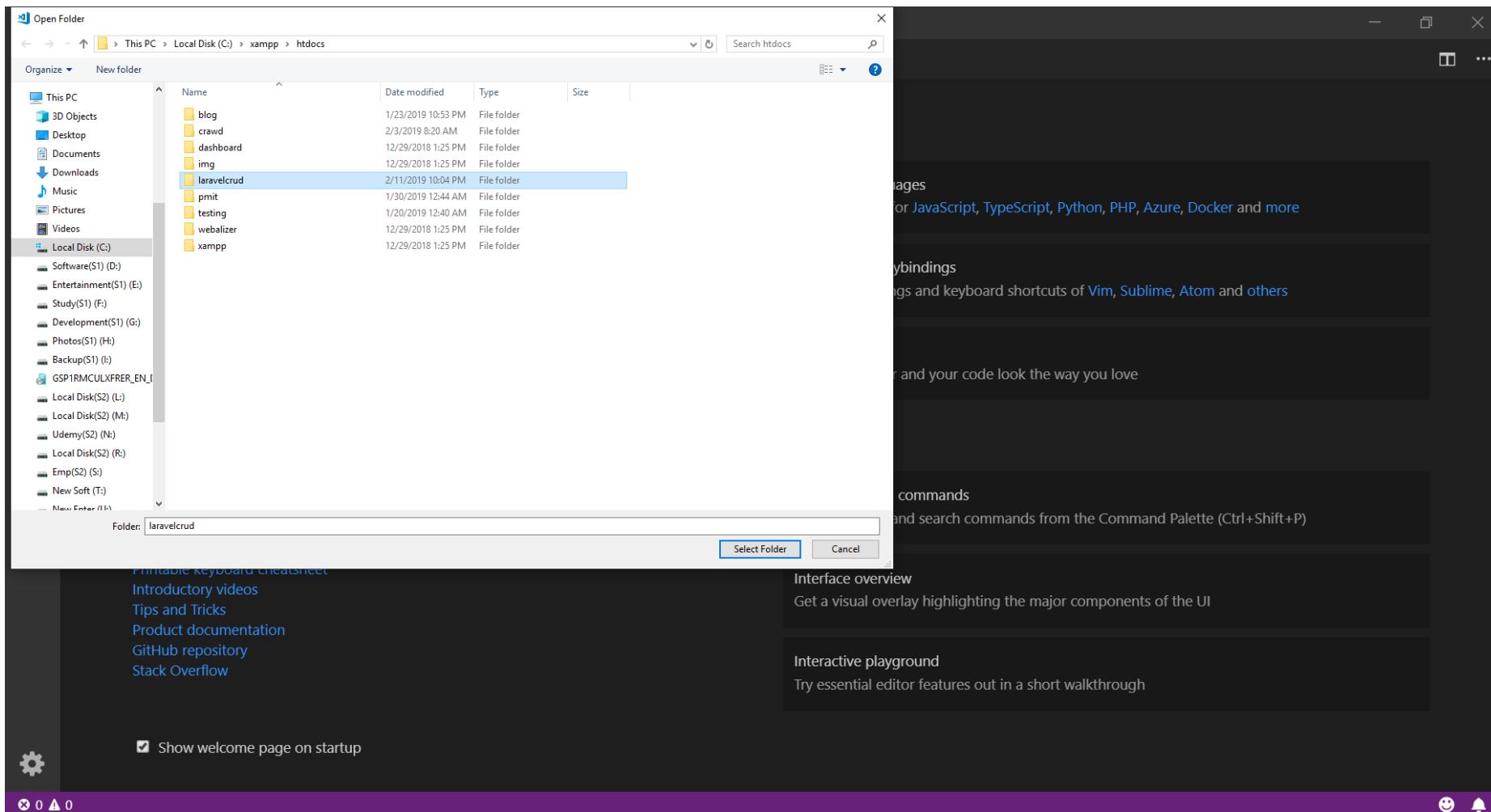


Akash Technolabs

[www.akashsir.com](http://www.akashsir.com)

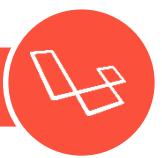


# Select the Folder in VS Code

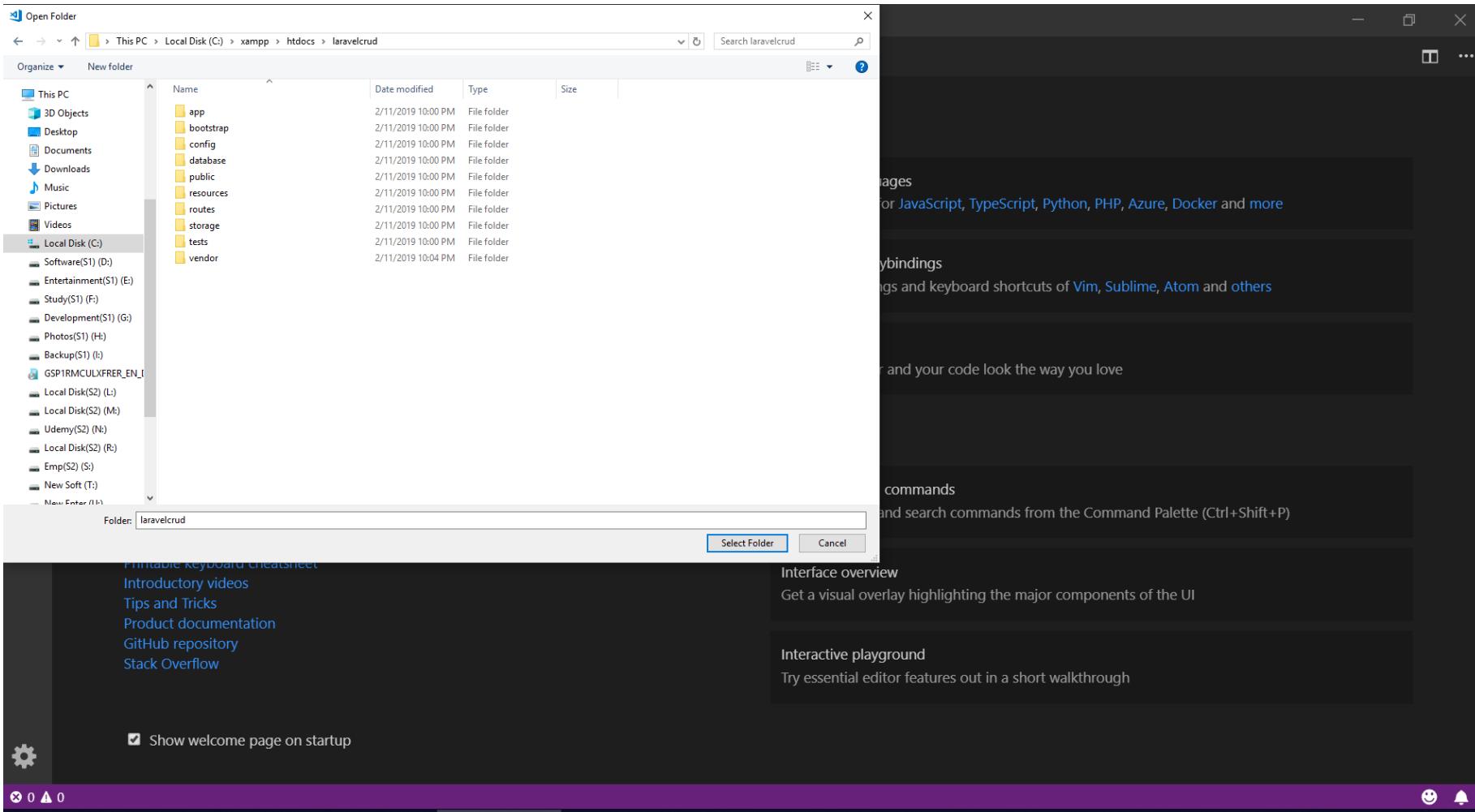


Akash Technolabs

[www.akashsir.com](http://www.akashsir.com)

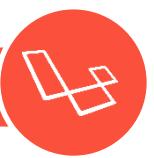


# Confirm

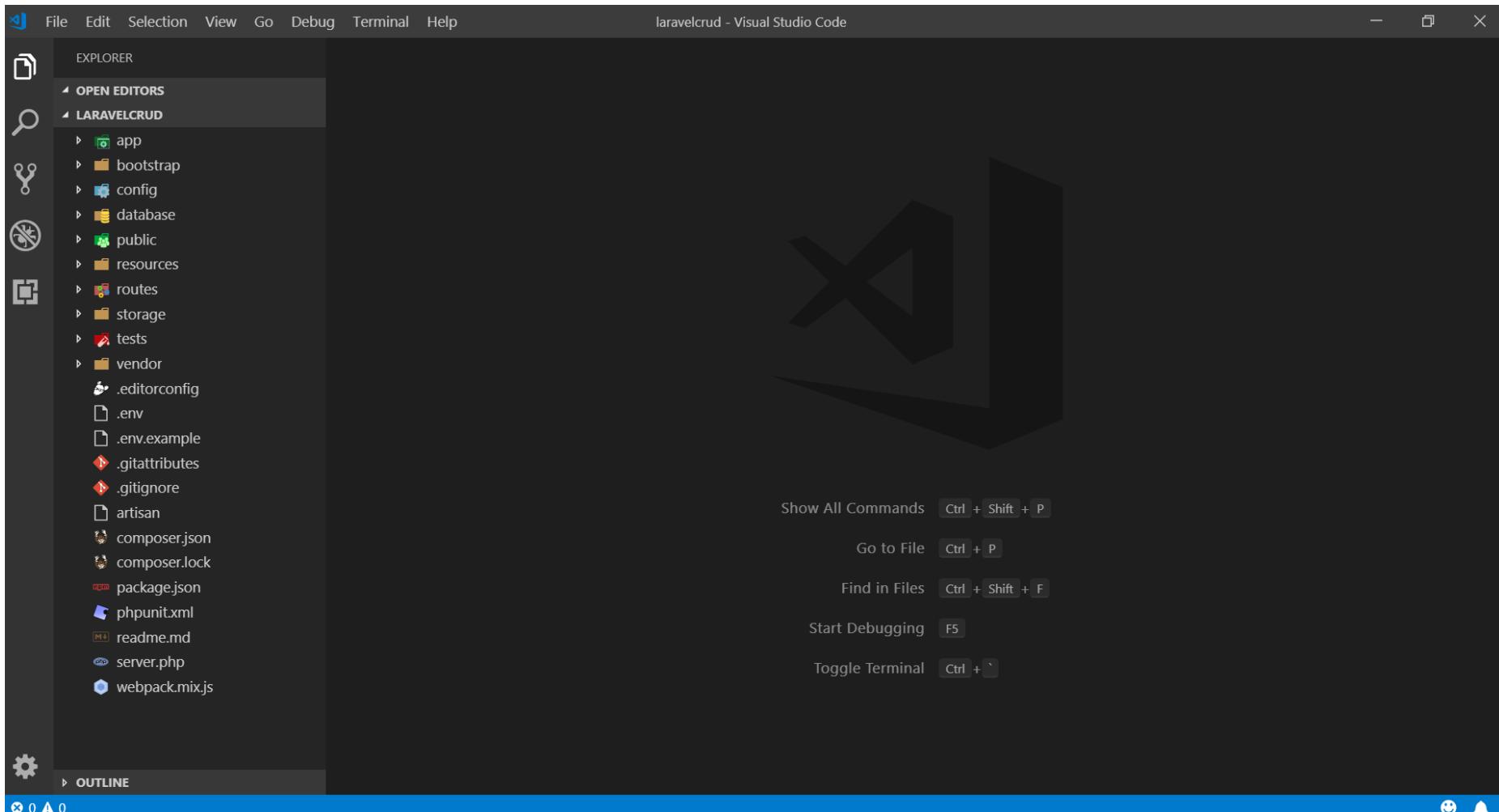


Akash Technolabs

[www.akashsir.com](http://www.akashsir.com)

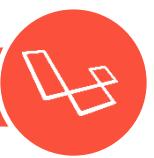


# Laravel Project

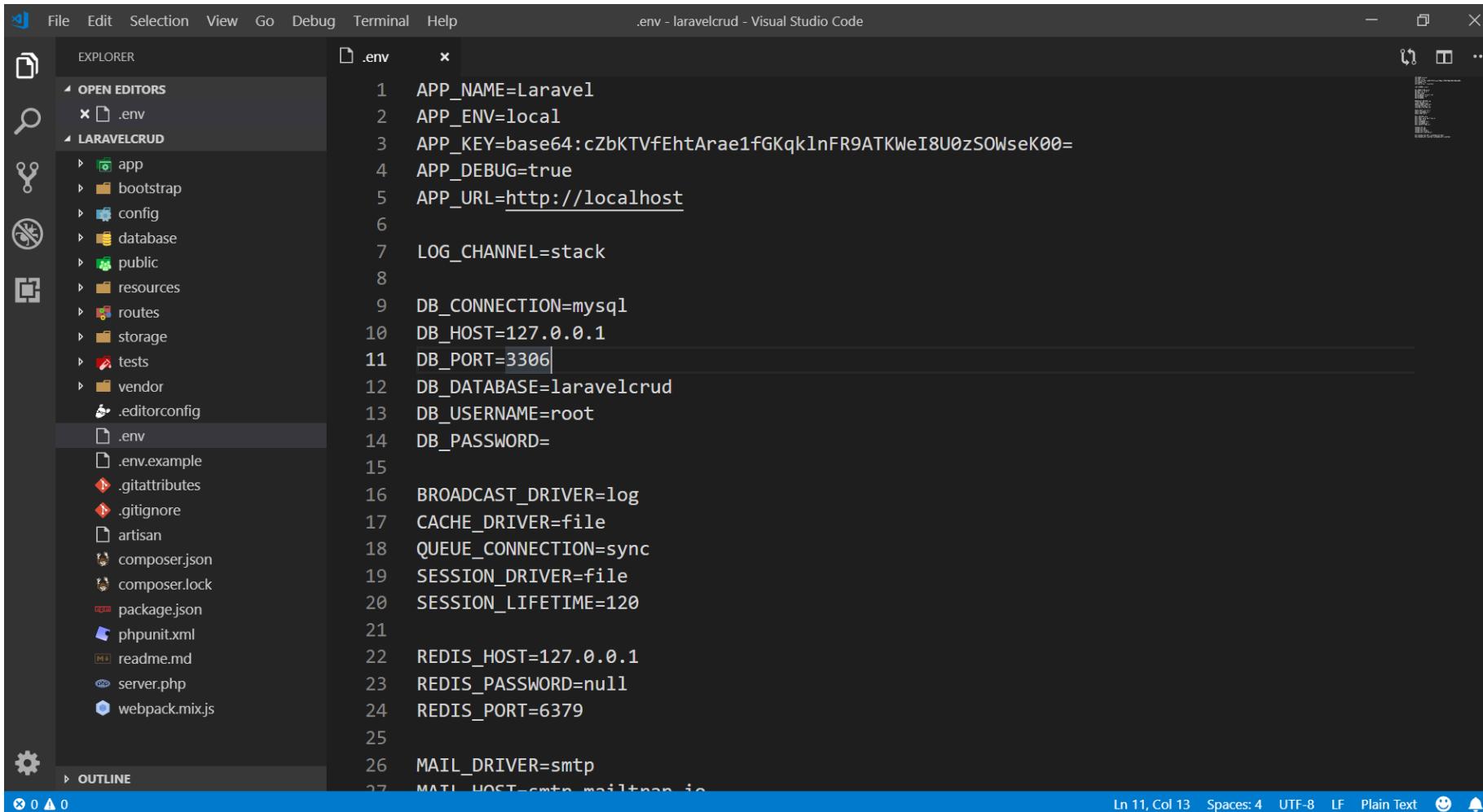


Akash Technolabs

[www.akashsir.com](http://www.akashsir.com)

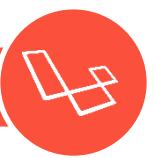


# Database Connection in .ENV File

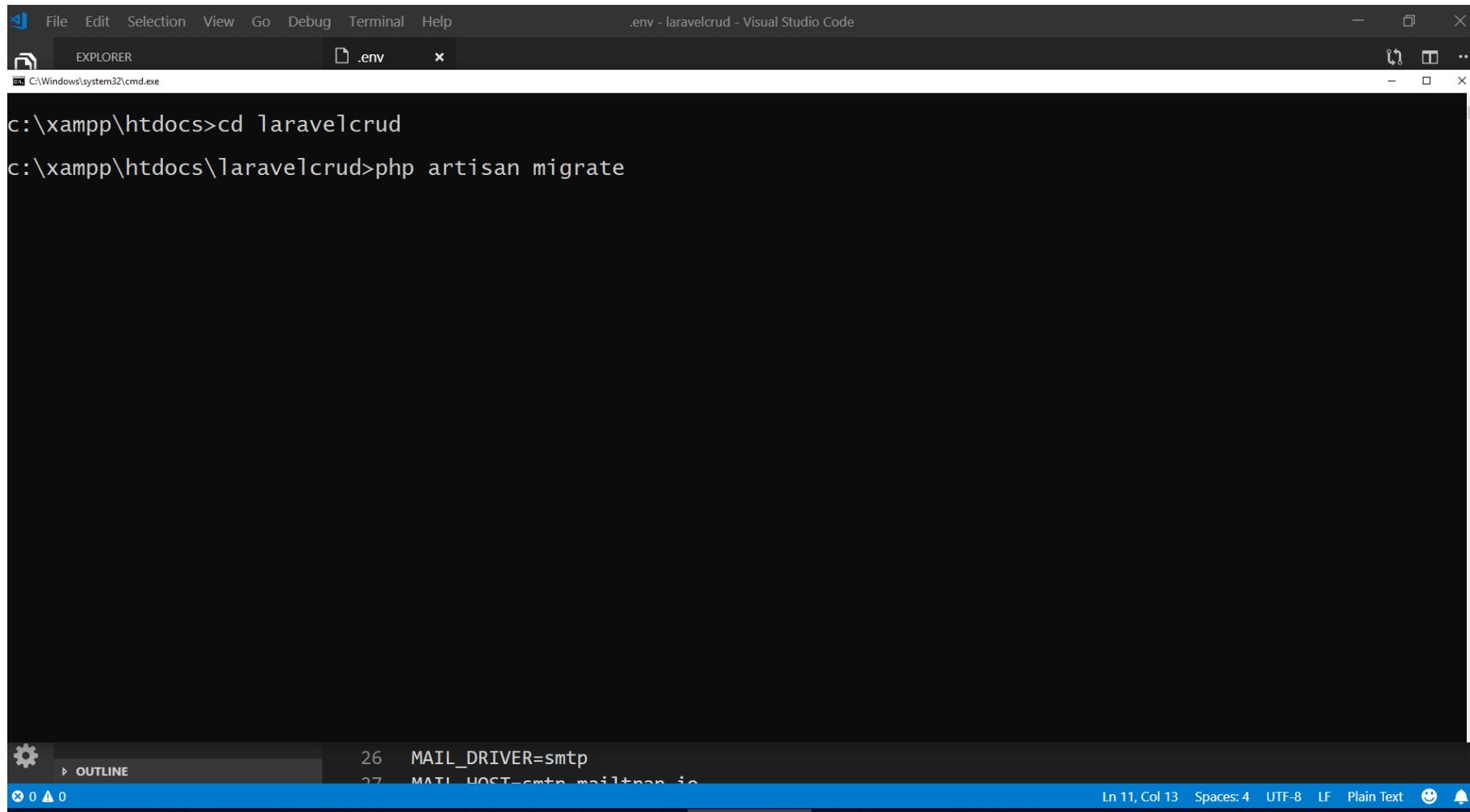


The screenshot shows the Visual Studio Code interface with the '.env' file open. The file contains environment variables for a Laravel application. The variables include APP\_NAME, APP\_ENV, APP\_KEY, APP\_DEBUG, APP\_URL, LOG\_CHANNEL, DB\_CONNECTION, DB\_HOST, DB\_PORT, DB\_DATABASE, DB\_USERNAME, DB\_PASSWORD, BROADCAST\_DRIVER, CACHE\_DRIVER, QUEUE\_CONNECTION, SESSION\_DRIVER, SESSION\_LIFETIME, REDIS\_HOST, REDIS\_PASSWORD, REDIS\_PORT, MAIL\_DRIVER, and MAIL\_HOST. The code editor has syntax highlighting and line numbers. The status bar at the bottom shows file statistics and encoding information.

```
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:cZbKTVfEhtArae1fGKqklnFR9ATKWeI8U0zSOWseK00=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8
9 DB_CONNECTION=mysql
10 DB_HOST=127.0.0.1
11 DB_PORT=3306
12 DB_DATABASE=laravelcrud
13 DB_USERNAME=root
14 DB_PASSWORD=
15
16 BROADCAST_DRIVER=log
17 CACHE_DRIVER=file
18 QUEUE_CONNECTION=sync
19 SESSION_DRIVER=file
20 SESSION_LIFETIME=120
21
22 REDIS_HOST=127.0.0.1
23 REDIS_PASSWORD=null
24 REDIS_PORT=6379
25
26 MAIL_DRIVER=smtp
27 MAIL_HOST=smtp.mailtrap.io
```



# Migrate Database Using Migrate Command



A screenshot of a Visual Studio Code interface. The title bar says ".env - laravelcrud - Visual Studio Code". The main area is a terminal window with the following text:

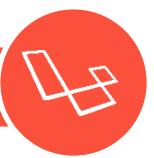
```
c:\xampp\htdocs>cd laravelcrud
c:\xampp\htdocs\laravelcrud>php artisan migrate
```

The bottom status bar shows file numbers 26 and 27, and configuration settings: MAIL\_DRIVER=smtp, MAIL\_HOST=smtp.mailtrap.io. The status bar also indicates the current line (Ln 11), column (Col 13), spaces (Spaces: 4), encoding (UTF-8 LF), and text type (Plain Text). There are also icons for a gear, outline, and other developer tools.

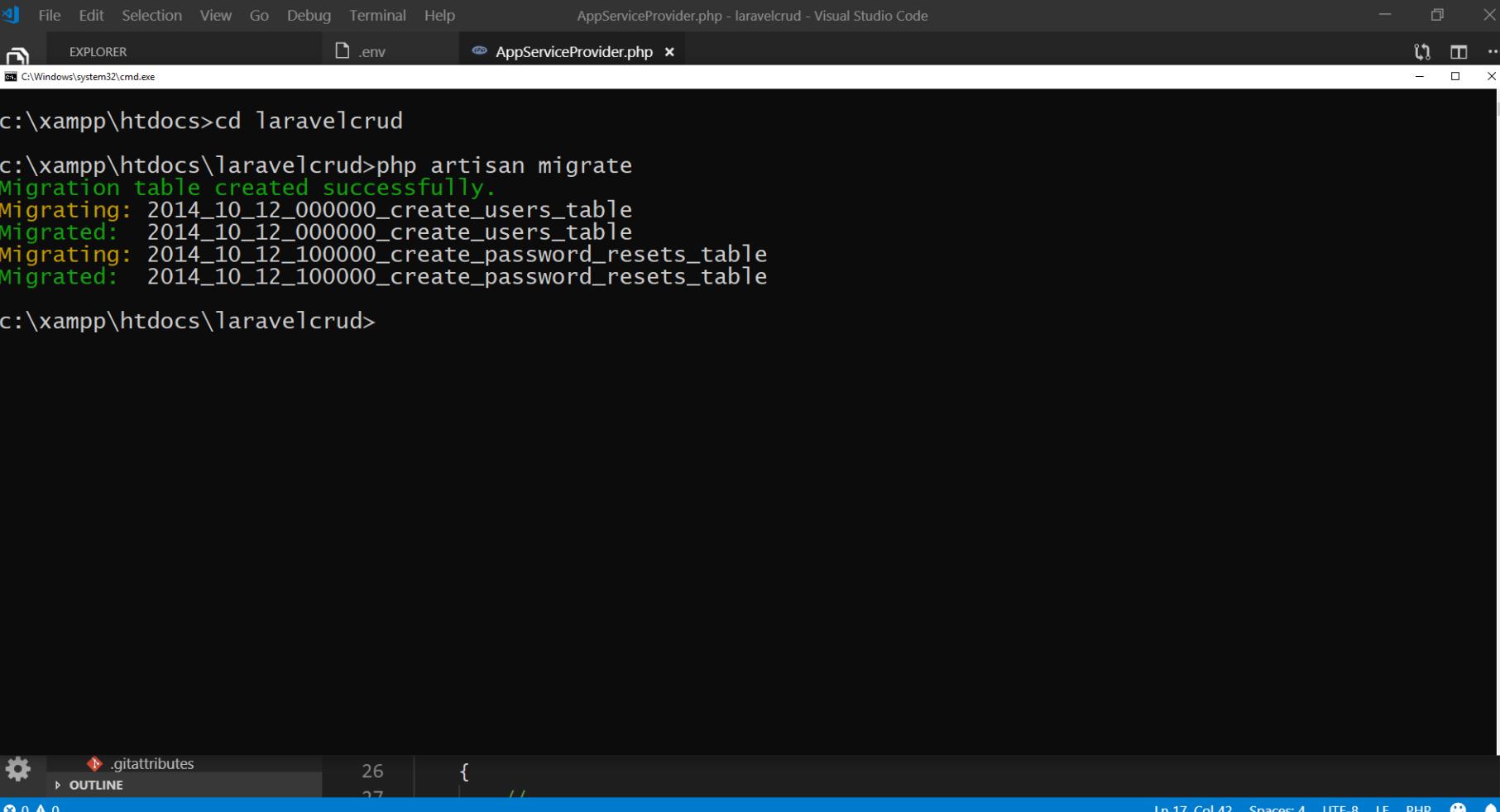


Akash Technolabs

[www.akashsir.com](http://www.akashsir.com)

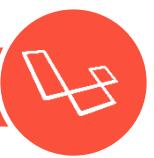


# Migration Completed



A screenshot of Visual Studio Code showing a terminal window. The terminal is running a command prompt (cmd.exe) and executing Laravel artisan commands. The output shows the migration process:

```
c:\xampp\htdocs>cd laravelcrud
c:\xampp\htdocs\laravelcrud>php artisan migrate
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table
c:\xampp\htdocs\laravelcrud>
```

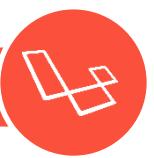


# Tables Created in Database

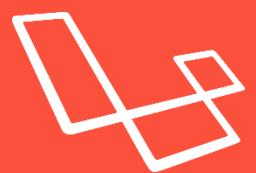
The screenshot shows the phpMyAdmin interface for the 'laravelcrud' database. The left sidebar lists various databases, and the main area displays the 'Structure' tab for the selected database. The table list shows the following details:

Table	Action	Rows	Type	Collation	Size
migrations	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_unicode_ci	16 KiB
password_resets	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16 KiB
users	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16 KiB
3 tables	Sum	2	InnoDB	latin1_swedish_ci	48 KiB

At the bottom, a 'Create table' form is open, showing fields for 'Name:' (with a value of '4') and 'Number of columns:'. A 'Go' button is located to the right of the column count input.



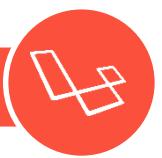
# Model



# Create New Model

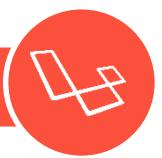
Create Module and -m will create Migration file for Product Table

```
C:\Windows\system32\cmd.exe
c:\xampp\htdocs\laravelcrud>php artisan make:model Product -m
```

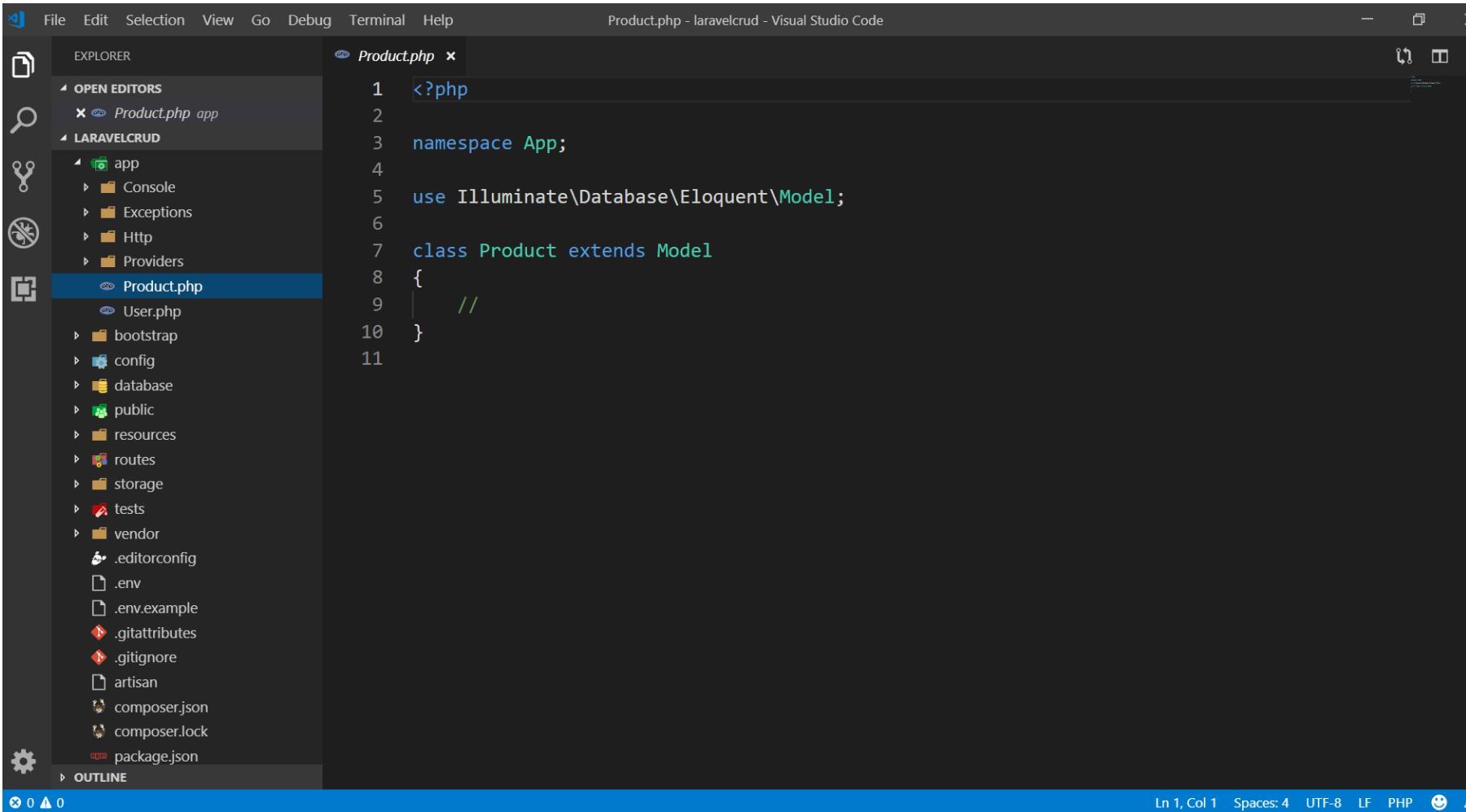


# Product Table Created

```
cl C:\Windows\system32\cmd.exe
c:\xampp\htdocs\laravelcrud>php artisan make:model Product -m
Model created successfully.
Created Migration: 2019_02_11_165053_create_products_table
c:\xampp\htdocs\laravelcrud>
```



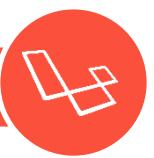
# Model File Created in App Folder



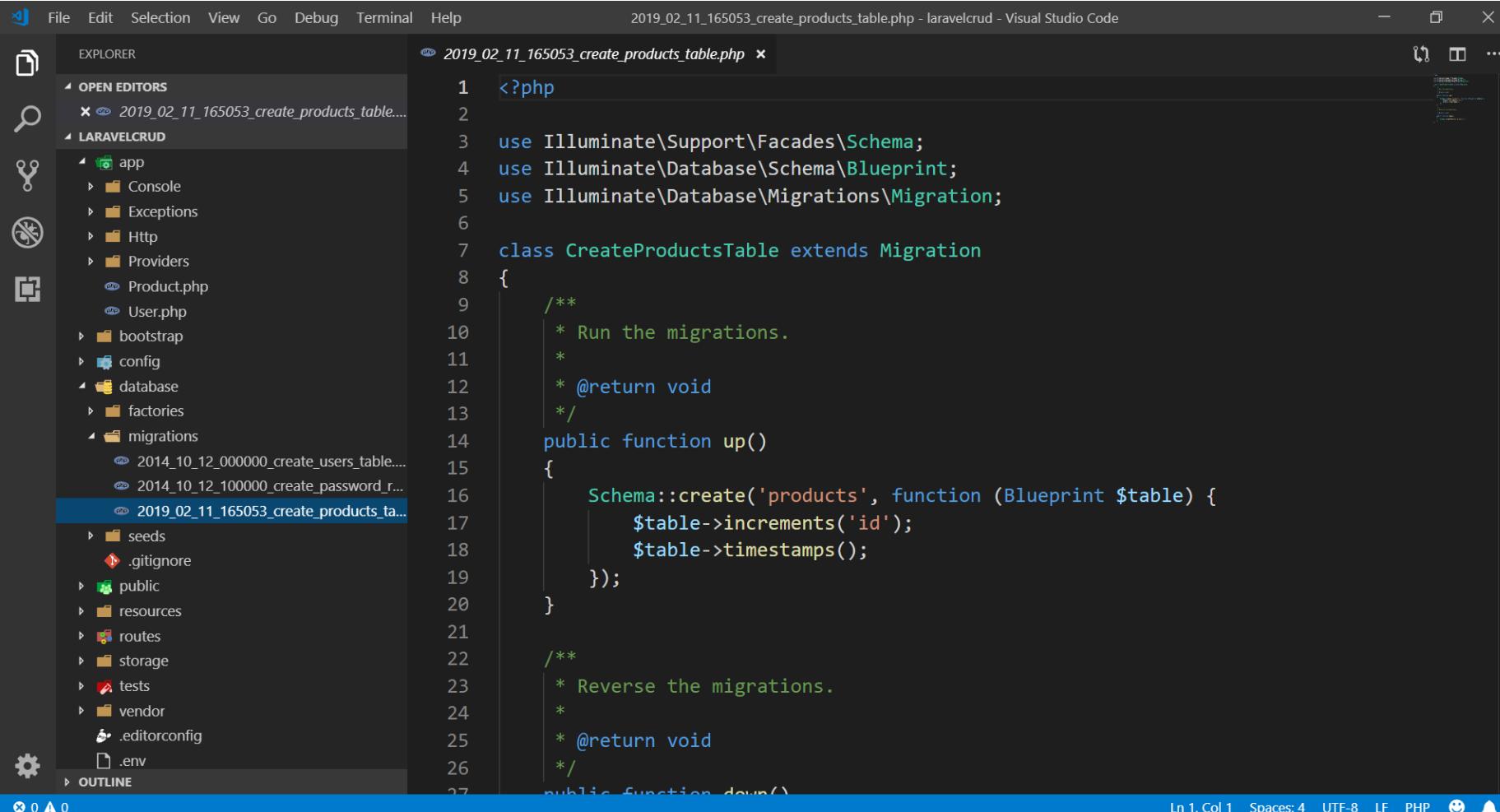
The screenshot shows a Visual Studio Code interface with a dark theme. The left sidebar displays the project structure under 'LARAVELCRUD'. The 'Product.php' file is selected in the 'OPEN EDITORS' list. The code editor shows the following PHP code:

```
1 <?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class Product extends Model
8 {
9     //
10 }
```

The status bar at the bottom indicates the file is 'Product.php - laravelcrud - Visual Studio Code' with 'Ln 1, Col 1' and encoding 'UTF-8 LF PHP'.



# Migration File Created in Folder



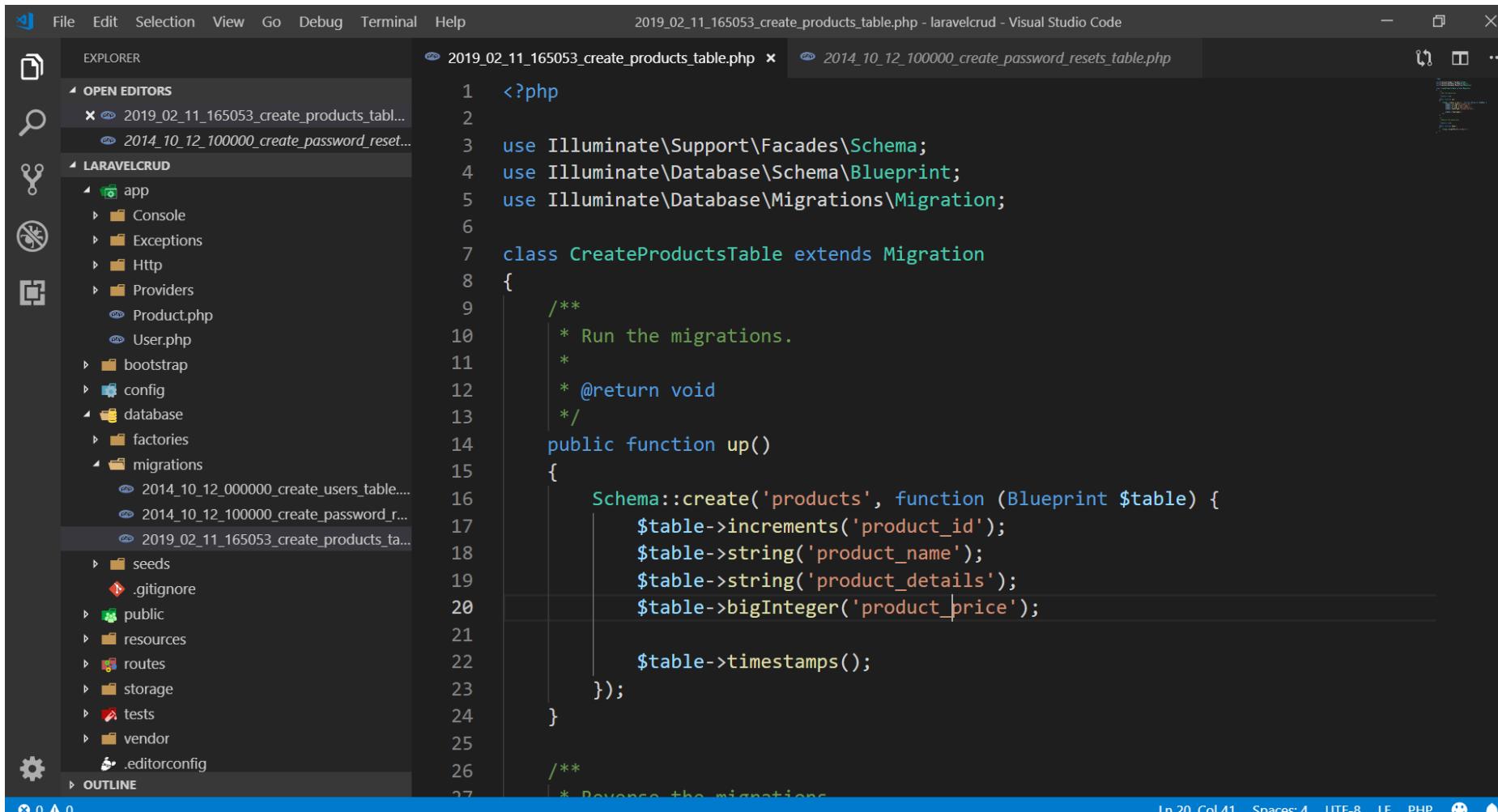
The screenshot shows a Visual Studio Code interface with a dark theme. The left sidebar is the Explorer view, showing the project structure of a Laravel application named 'LARAVELCRUD'. The 'migrations' folder contains several migration files, including '2014\_10\_12\_000000\_create\_users\_table.php', '2014\_10\_12\_100000\_create\_password\_r...', and '2019\_02\_11\_165053\_create\_products\_ta...'. The file '2019\_02\_11\_165053\_create\_products\_table.php' is currently open in the main editor area. The code is a PHP migration class:

```
<?php  
use Illuminate\Support\Facades\Schema;  
use Illuminate\Database\Schema\Blueprint;  
use Illuminate\Database\Migrations\Migration;  
  
class CreateProductsTable extends Migration  
{  
    /**  
     * Run the migrations.  
     *  
     * @return void  
     */  
    public function up()  
    {  
        Schema::create('products', function (Blueprint $table) {  
            $table->increments('id');  
            $table->timestamps();  
        });  
    }  
  
    /**  
     * Reverse the migrations.  
     *  
     * @return void  
     */  
    public function down()  
    {  
    }  
}
```

The status bar at the bottom indicates the file is at line 1, column 1, with 4 spaces, using LF line endings, and the language is set to PHP.

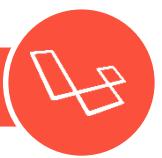


# Specify Column Name for Product Table

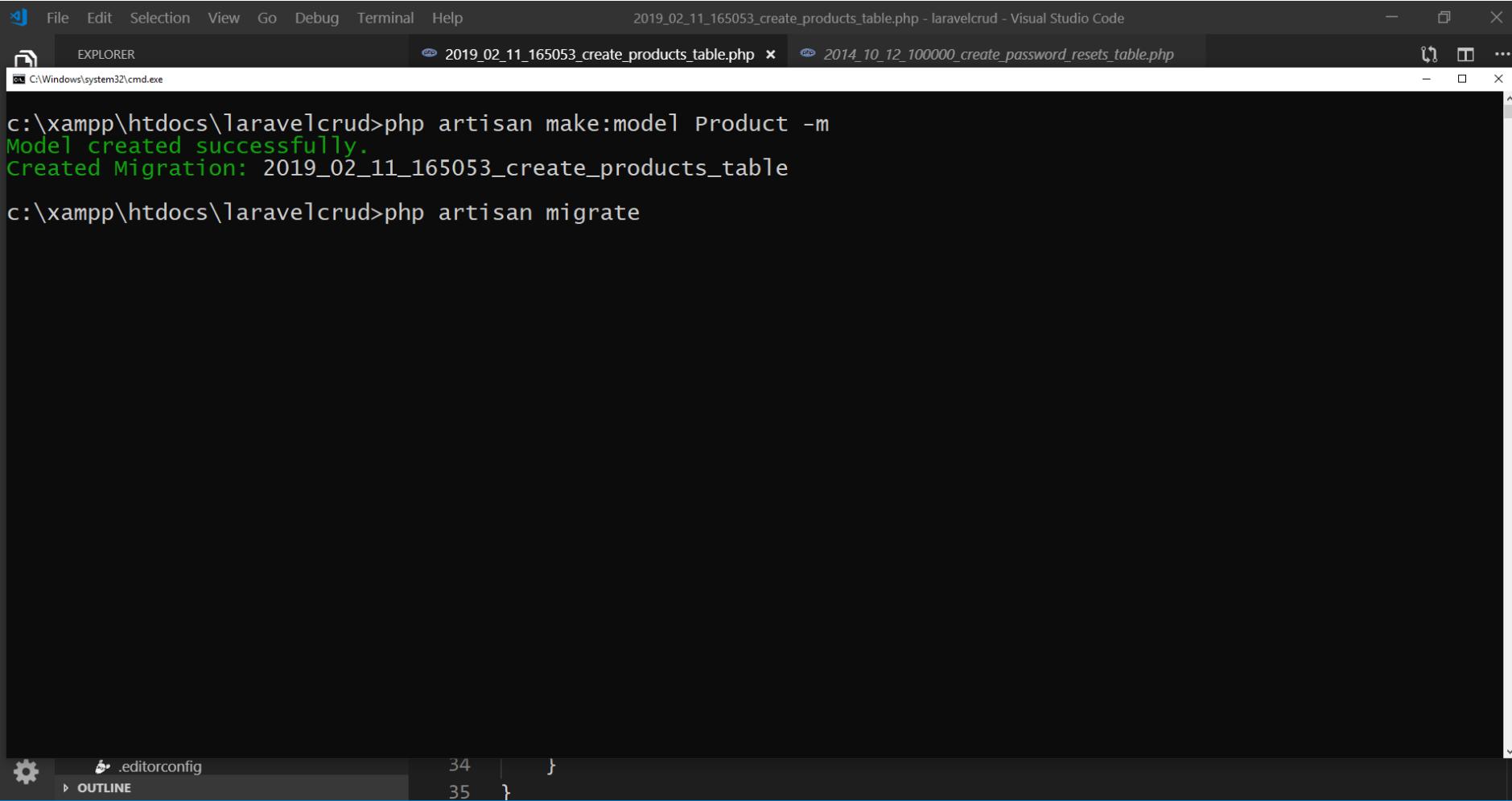


The screenshot shows a Visual Studio Code interface with a dark theme. The left sidebar contains an Explorer view with a tree structure of project files under 'LARAVELCRUD'. The main editor area displays a PHP migration file named '2019\_02\_11\_165053\_create\_products\_table.php'. The code defines a 'CreateProductsTable' class that extends 'Migration'. The 'up()' method uses the Schema facade to create a 'products' table with columns for 'product\_id' (increments), 'product\_name' (string), 'product\_details' (string), and 'product\_price' (bigInteger). The 'timestamps()' method is also called. The 'down()' method is present but empty.

```
<?php  
use Illuminate\Support\Facades\Schema;  
use Illuminate\Database\Schema\Blueprint;  
use Illuminate\Database\Migrations\Migration;  
  
class CreateProductsTable extends Migration  
{  
    /**  
     * Run the migrations.  
     *  
     * @return void  
     */  
    public function up()  
    {  
        Schema::create('products', function (Blueprint $table) {  
            $table->increments('product_id');  
            $table->string('product_name');  
            $table->string('product_details');  
            $table->bigInteger('product_price');  
            $table->timestamps();  
        });  
    }  
    /**  
     * Reverse the migrations.  
     */  
}
```

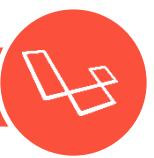


# Now Migrate Product Table

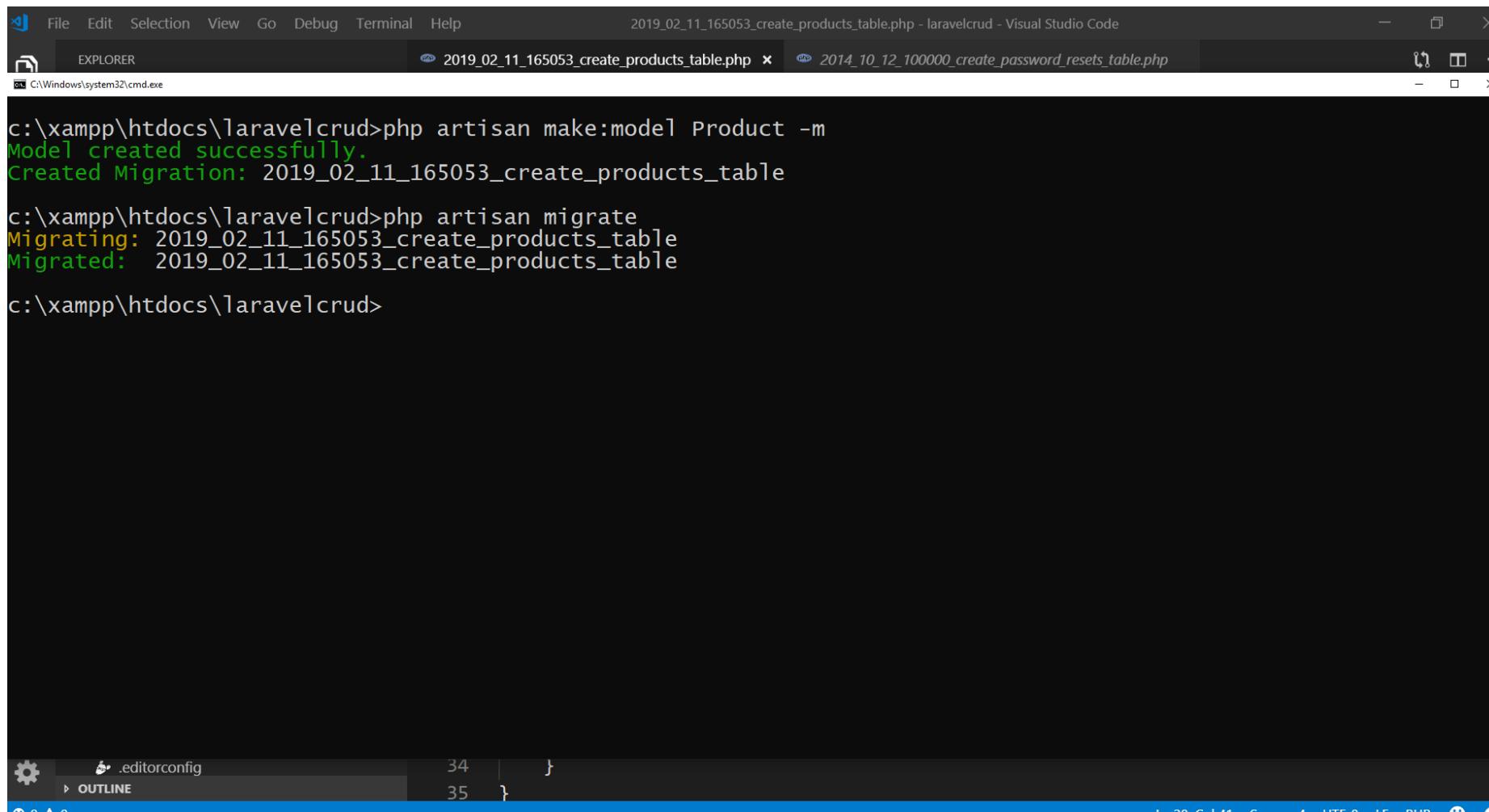


A screenshot of the Visual Studio Code interface showing a terminal window. The terminal shows the following command-line session:

```
c:\xampp\htdocs\laravelcrud>php artisan make:model Product -m
Model created successfully.
Created Migration: 2019_02_11_165053_create_products_table
c:\xampp\htdocs\laravelcrud>php artisan migrate
```



# Product Table Migration



A screenshot of Visual Studio Code showing a terminal window. The terminal shows the following command-line session:

```
c:\xampp\htdocs\laravelcrud>php artisan make:model Product -m
Model created successfully.
Created Migration: 2019_02_11_165053_create_products_table

c:\xampp\htdocs\laravelcrud>php artisan migrate
Migrating: 2019_02_11_165053_create_products_table
Migrated: 2019_02_11_165053_create_products_table

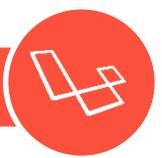
c:\xampp\htdocs\laravelcrud>
```

The terminal interface includes tabs for '2019\_02\_11\_165053\_create\_products\_table.php' and '2014\_10\_12\_100000\_create\_password\_resets\_table.php'. The status bar at the bottom shows file counts (139, 141), encoding (UTF-8), line numbers (34, 35), and other standard developer tools information.

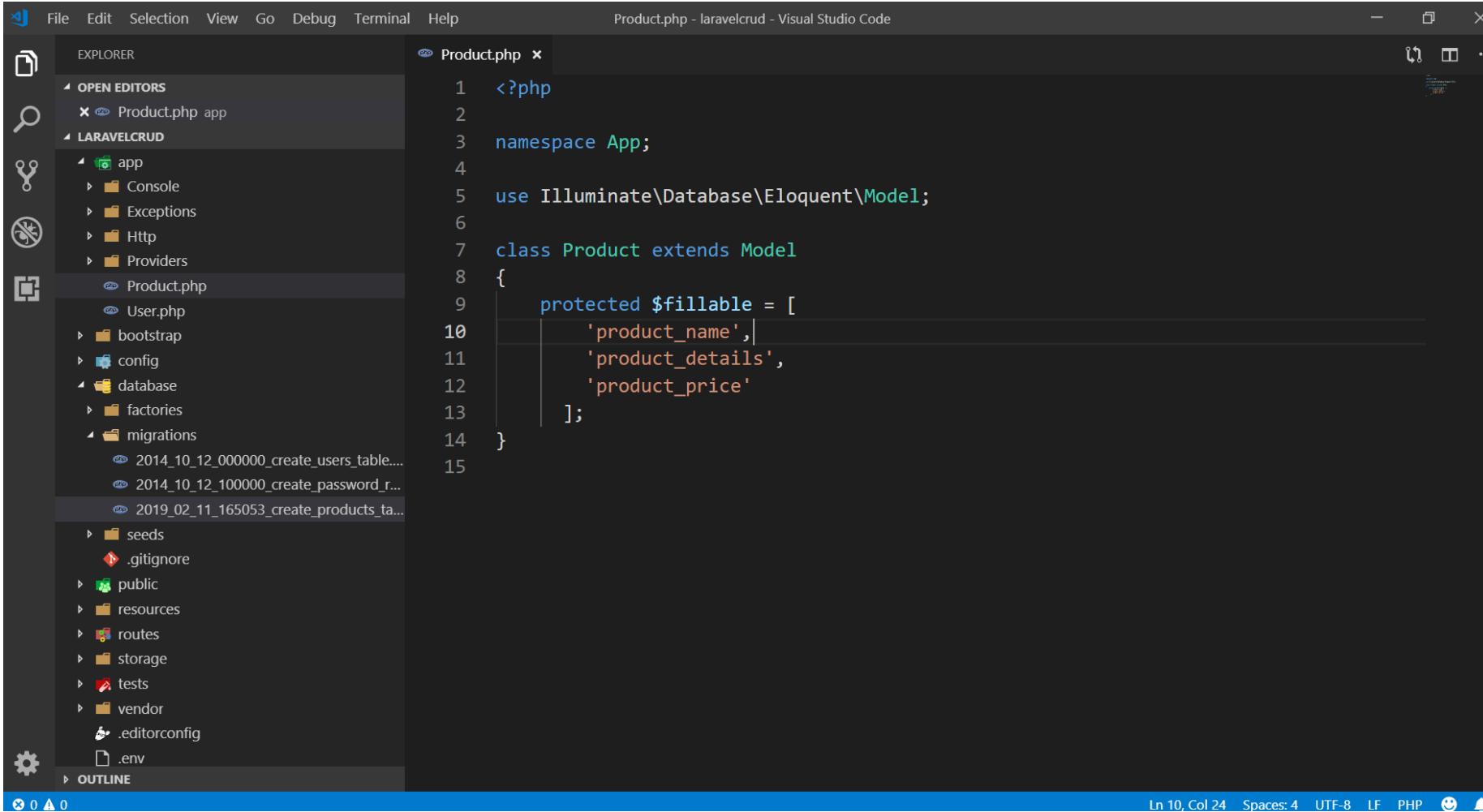


Akash Technolabs

[www.akashsir.com](http://www.akashsir.com)



# Product Model Specify your Field Name



The screenshot shows a Visual Studio Code interface with the following details:

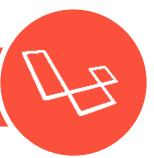
- File Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** Product.php - laravelcrud - Visual Studio Code.
- Explorer:** Shows the project structure under "LARAVELCRUD".
  - app
    - Console
    - Exceptions
    - Http
    - Providers
    - Product.php (selected)
    - User.php
  - bootstrap
  - config
  - database
    - factories
    - migrations
      - 2014\_10\_12\_000000\_create\_users\_table...
      - 2014\_10\_12\_100000\_create\_password\_r...
      - 2019\_02\_11\_165053\_create\_products\_ta...
    - seeds
  - .gitignore
  - public
  - resources
  - routes
  - storage
  - tests
  - vendor
    - .editorconfig
    - .env
- OUTLINE

- Product.php Content:**

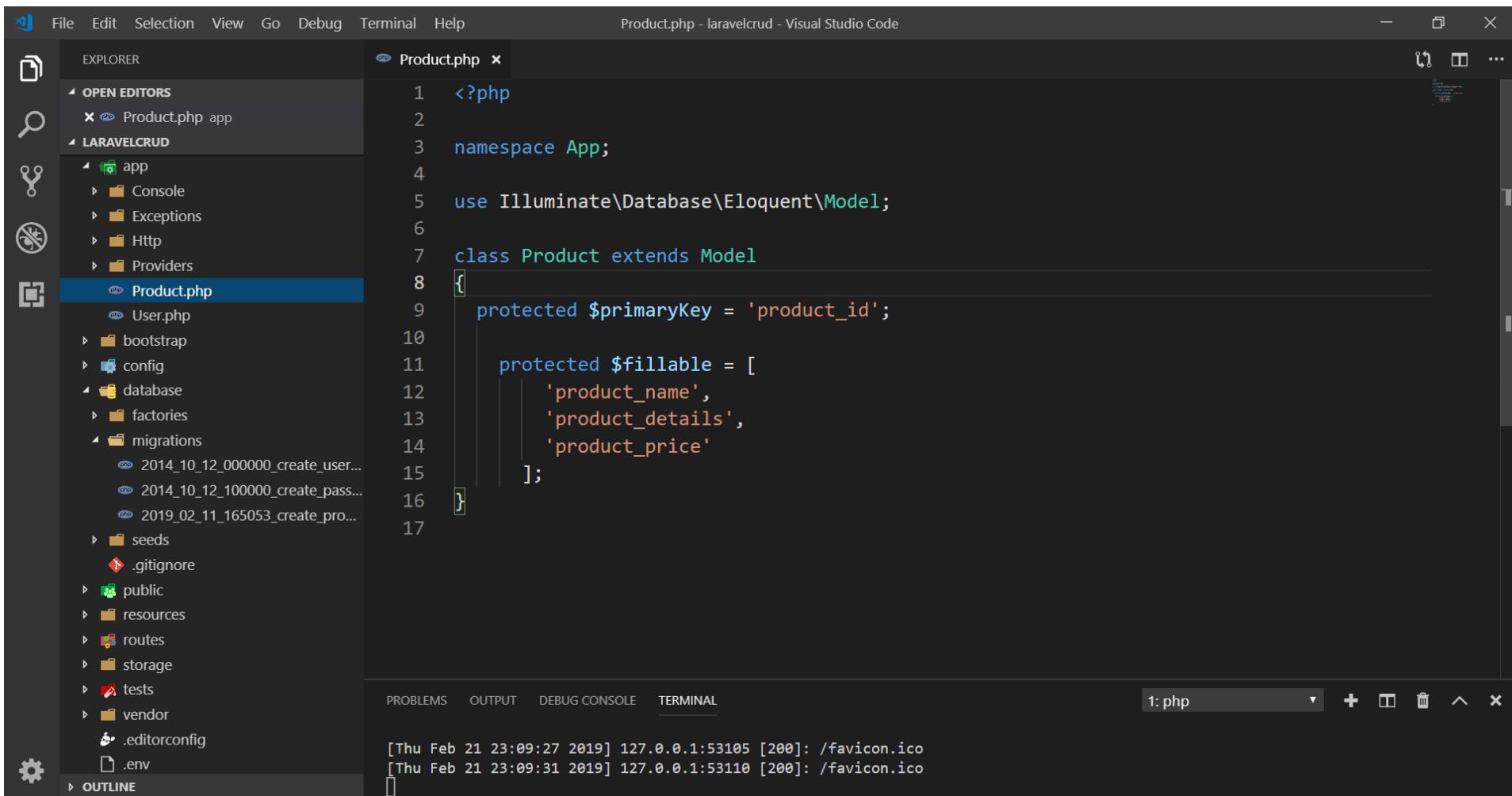
```
1 <?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class Product extends Model
8 {
9     protected $fillable = [
10         'product_name',
11         'product_details',
12         'product_price'
13     ];
14 }
15
```
- Status Bar:** Ln 10, Col 24, Spaces: 4, UTF-8, LF, PHP, 😊, ⚡.


Akash Technolabs

[www.akashsir.com](http://www.akashsir.com)



# Specify Primary Key

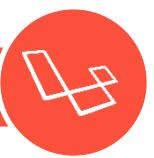


The screenshot shows a Visual Studio Code interface with the following details:

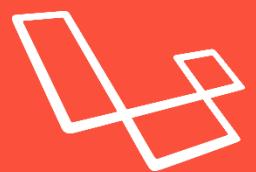
- File Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** Product.php - laravelcrud - Visual Studio Code.
- Explorer:** Shows the project structure under "LARAVELCRUD". The "Product.php" file is selected and highlighted in blue.
- Editor:** Displays the PHP code for the Product model:

```
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    ];
16 }
17
```
- Terminal:** Shows two log entries from a previous run:

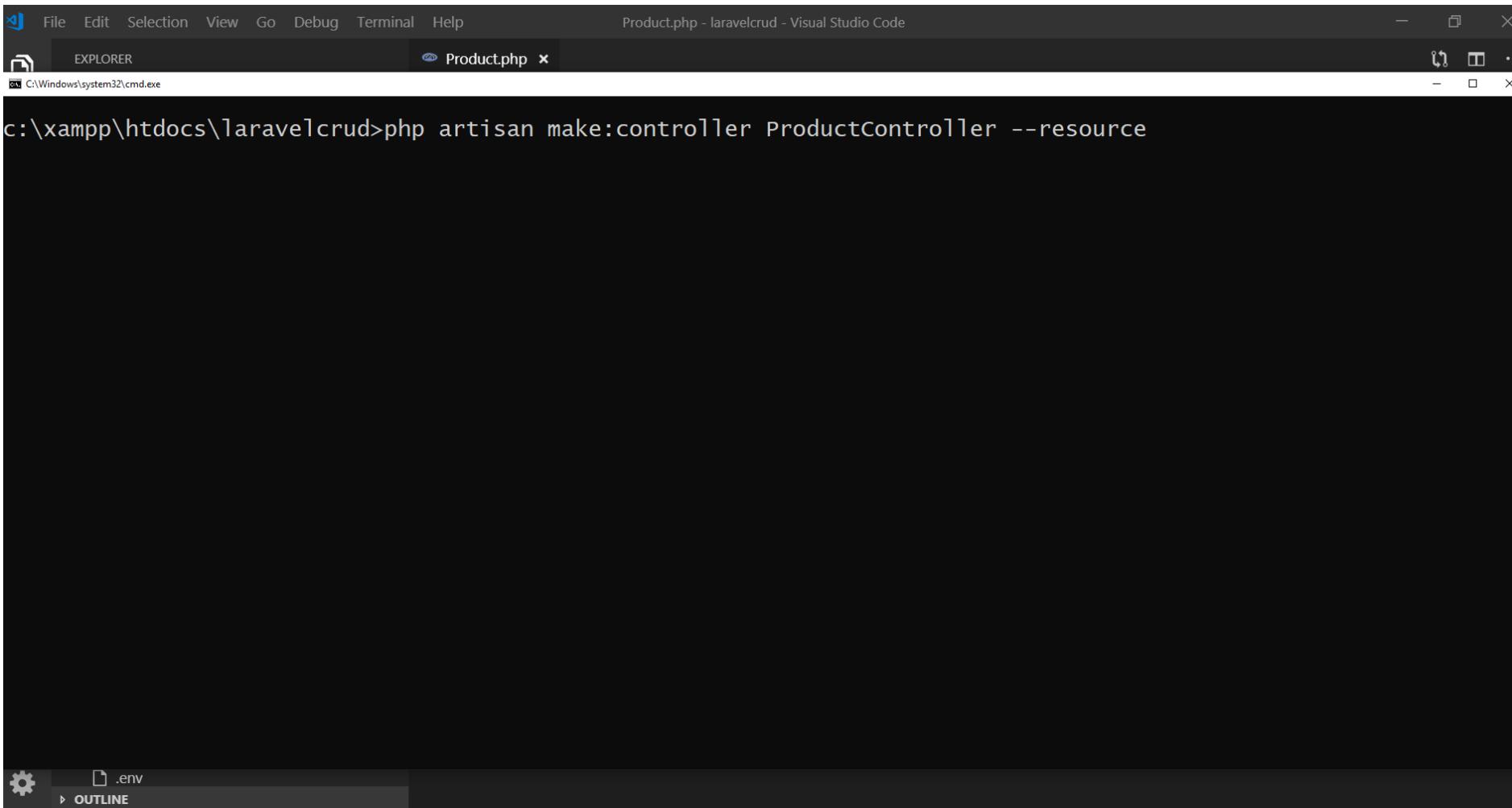
```
[Thu Feb 21 23:09:27 2019] 127.0.0.1:53105 [200]: /favicon.ico
[Thu Feb 21 23:09:31 2019] 127.0.0.1:53110 [200]: /favicon.ico
```



# Controller



# Create Resource Controller for Product

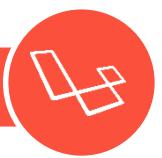


A screenshot of Visual Studio Code showing a terminal window. The title bar says "Product.php - laravelcrud - Visual Studio Code". The terminal tab is active, showing the command: "c:\xampp\htdocs\laravelcrud>php artisan make:controller ProductController --resource". The rest of the terminal window is blank black space.

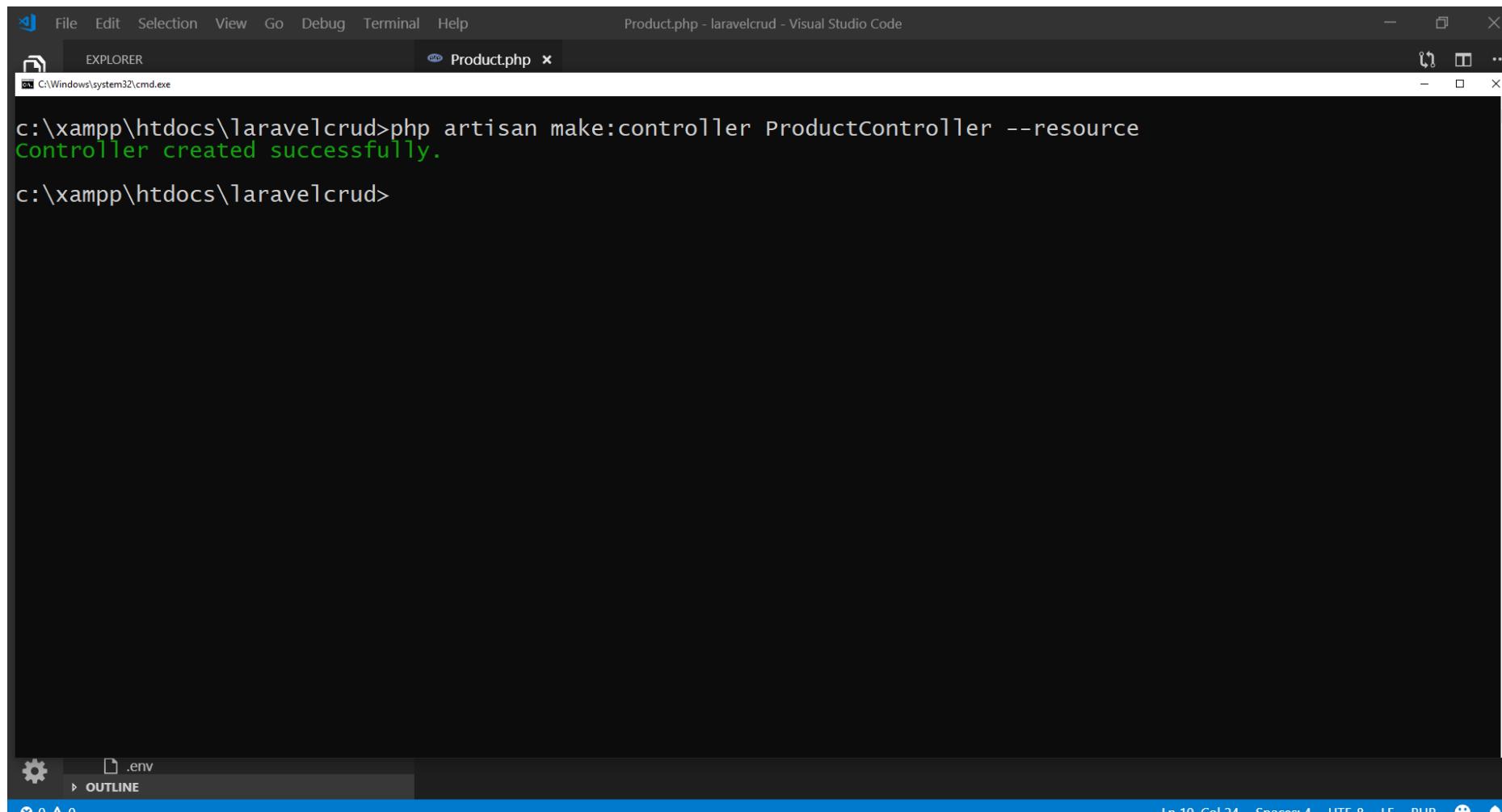


Akash Technolabs

[www.akashsir.com](http://www.akashsir.com)



# Controller Created



A screenshot of a Visual Studio Code interface. The title bar says "Product.php - laravelcrud - Visual Studio Code". The main area is a terminal window showing the following command and its output:

```
c:\xampp\htdocs\laravelcrud>php artisan make:controller ProductController --resource
Controller created successfully.

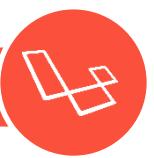
c:\xampp\htdocs\laravelcrud>
```

The terminal window has a dark background. Below the terminal, the VS Code interface shows the Explorer sidebar with ".env" and "OUTLINE" items, and the bottom status bar displays "In 10 Col 24 Spaces: 4 UFT-8 LF DUR".



Akash Technolabs

[www.akashsir.com](http://www.akashsir.com)

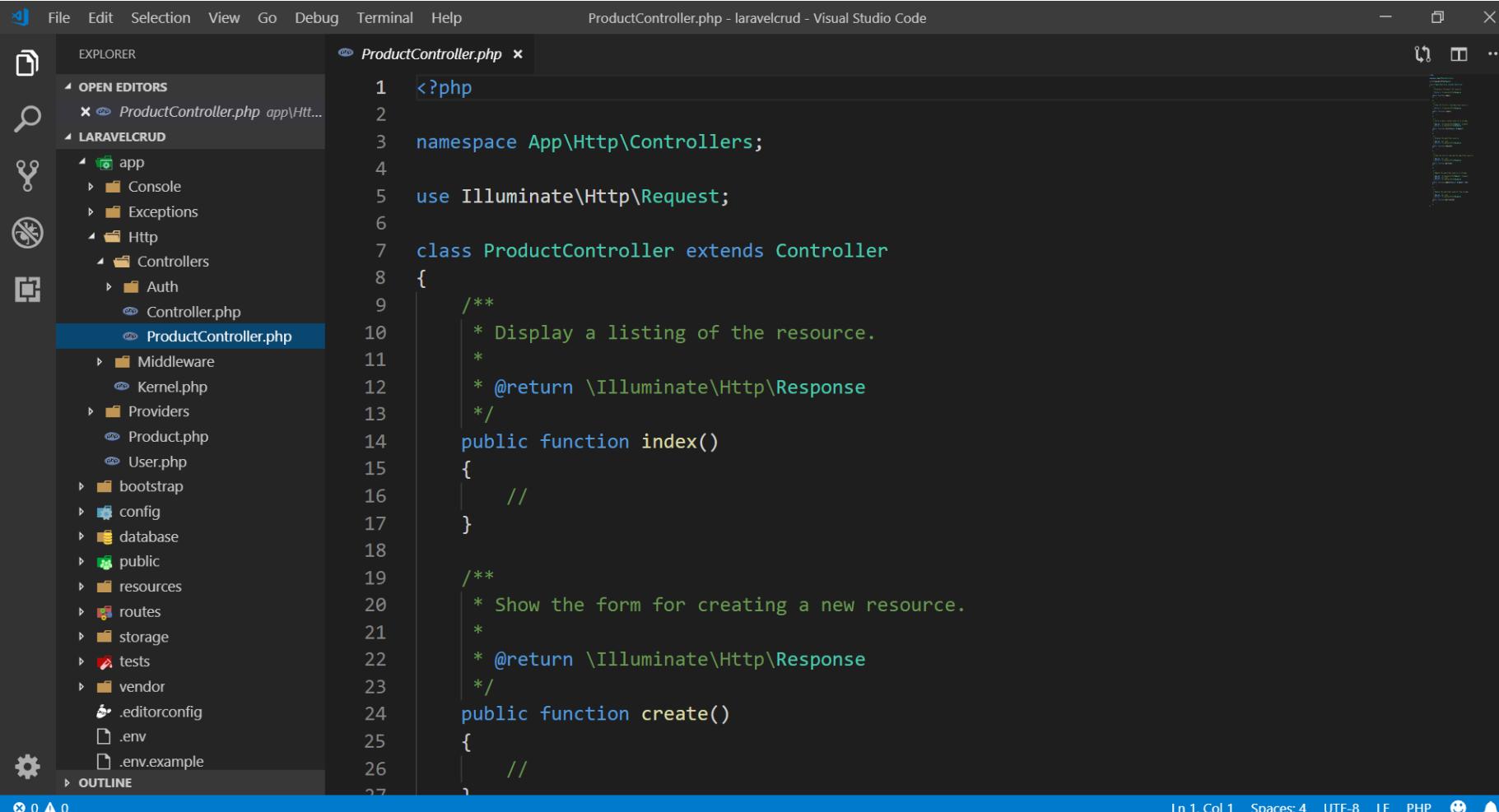


# Resource Controller

- It will create method
  - Index(Get)
    - Display Record
    - Controllername.index
  - Create (Get)
    - Display Form Design
    - Controllername.create
  - Store (Post)
    - Process form
    - Controllername.store
  - Show (Get)
    - Display Single Record
    - Controllername.show
  - Edit (Get)
    - Fetch Single Record
    - Controllername.edit
  - Update (Put | Patch)
    - Update Records
  - Destroy (Delete)
    - Delete Record
    - Controller.destroy



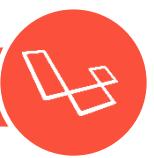
# Res Controller Created Index() Create()



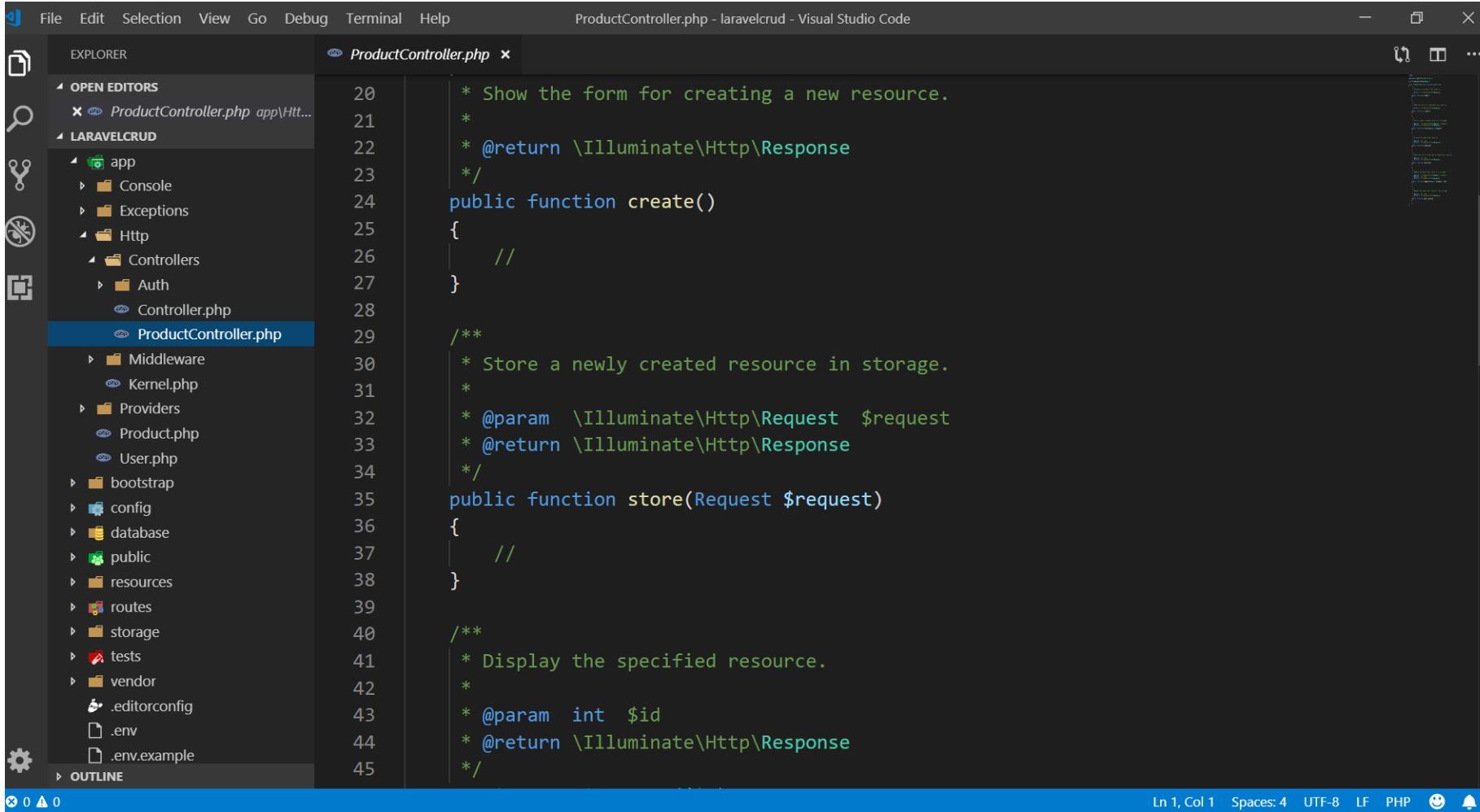
The screenshot shows a Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** ProductController.php - laravelcrud - Visual Studio Code.
- Editor Area:** Contains the PHP code for the ProductController.php file.
- Explorer Panel:** Shows the project structure under "LARAVELCRUD".
- Status Bar:** Ln 1, Col 1, Spaces: 4, UTF-8, LF, PHP, smiley face, bell icon.

```
<?php  
namespace App\Http\Controllers;  
use Illuminate\Http\Request;  
  
class ProductController extends Controller  
{  
    /**  
     * Display a listing of the resource.  
     *  
     * @return \Illuminate\Http\Response  
     */  
    public function index()  
    {  
        //  
    }  
  
    /**  
     * Show the form for creating a new resource.  
     *  
     * @return \Illuminate\Http\Response  
     */  
    public function create()  
    {  
        //  
    }  
}
```



# Res Controller Created Create() Store()

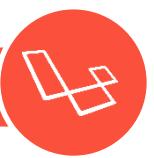


The screenshot shows a Visual Studio Code interface with the following details:

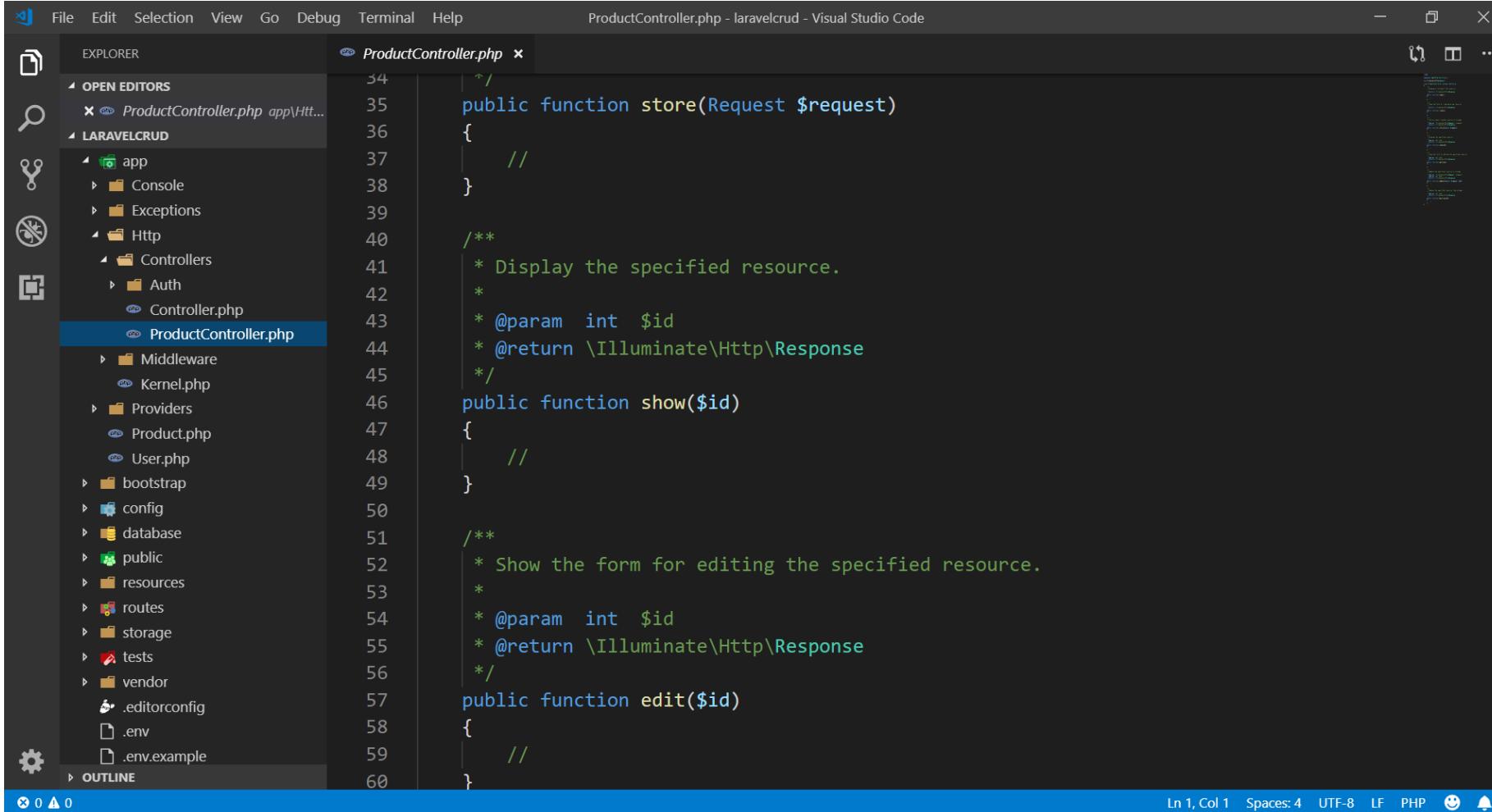
- File Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** ProductController.php - laravelcrud - Visual Studio Code.
- Explorer View:** Shows the project structure under "LARAVELCRUD". The "ProductController.php" file is selected and highlighted in blue.
- Code Editor:** Displays the PHP code for the ProductController.php file. The code includes methods for creating and storing resources, along with their respective documentation comments.
- Status Bar:** ShowsLn 1, Col 1, Spaces: 4, UTF-8, LF, PHP, and icons for file status and notifications.

```
ProductController.php

20 * Show the form for creating a new resource.
21 *
22 * @return \Illuminate\Http\Response
23 */
24 public function create()
25 {
26     //
27 }
28
29 /**
30 * Store a newly created resource in storage.
31 *
32 * @param \Illuminate\Http\Request $request
33 * @return \Illuminate\Http\Response
34 */
35 public function store(Request $request)
36 {
37     //
38 }
39
40 /**
41 * Display the specified resource.
42 *
43 * @param int $id
44 * @return \Illuminate\Http\Response
45 */
```



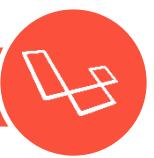
# Res Controller Created Store() Show() Edit()



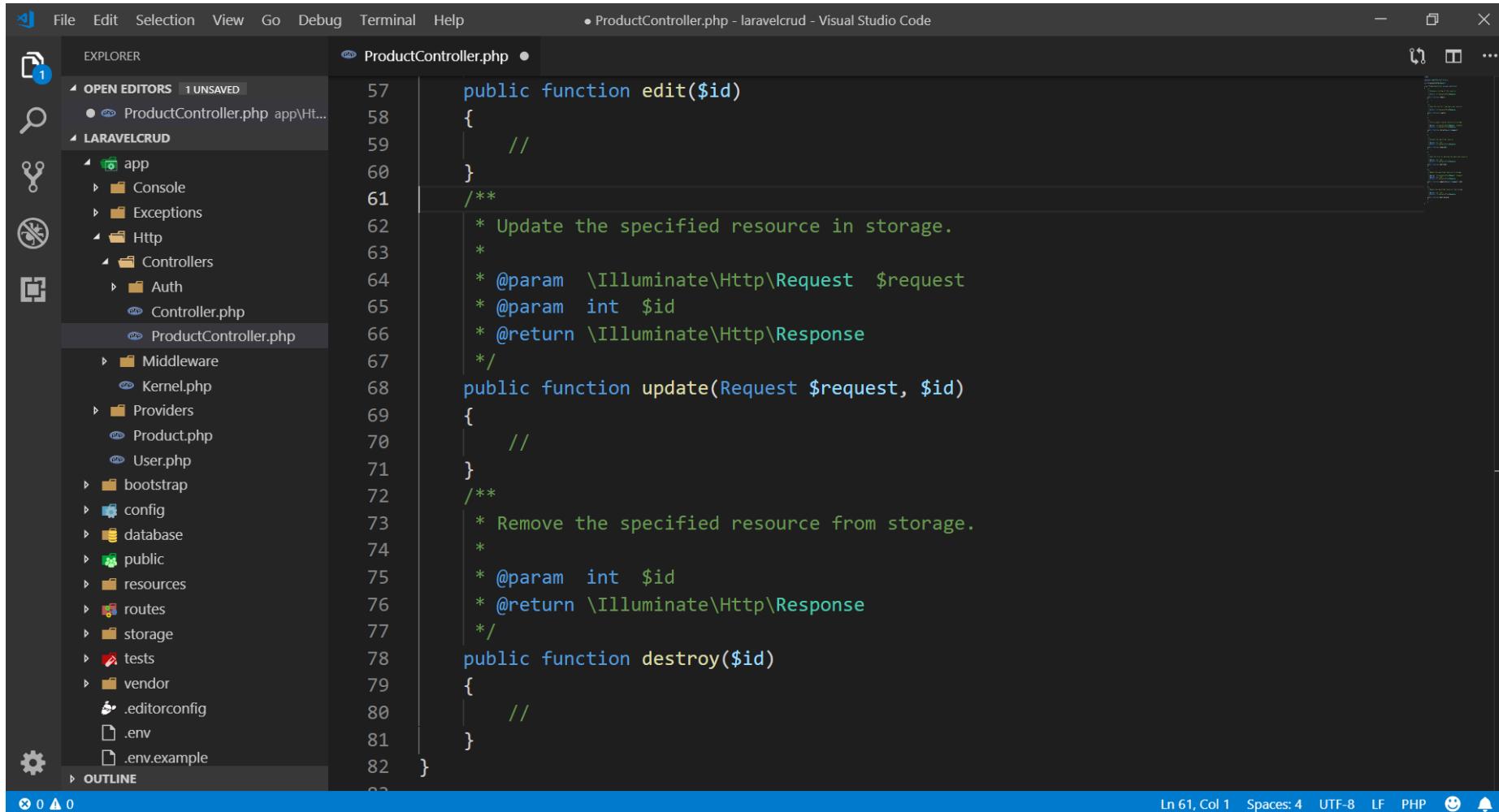
The screenshot shows a Visual Studio Code interface with the following details:

- Title Bar:** File, Edit, Selection, View, Go, Debug, Terminal, Help
- Active File:** ProductController.php - laravelcrud - Visual Studio Code
- Explorer:** Shows the project structure under LARAVELCRUD, including app, Http, and other Laravel components.
- Code Editor:** Displays the PHP code for ProductController.php, which includes three methods: store(), show(\$id), and edit(\$id). The code uses annotations like `* @param int $id` and `* @return \Illuminate\Http\Response`.
- Bottom Status Bar:** Shows Ln 1, Col 1, Spaces: 4, UTF-8, LF, PHP, and icons for file status and notifications.

```
ProductController.php
1 /**
2  * Store a newly created resource in storage.
3  *
4  * @param Request $request
5  * @return Response
6 */
7 public function store(Request $request)
8 {
9     //
10
11    /**
12     * Display the specified resource.
13     *
14     * @param int $id
15     * @return Response
16     */
17    public function show($id)
18    {
19        //
20
21        /**
22         * Show the form for editing the specified resource.
23         *
24         * @param int $id
25         * @return Response
26         */
27        public function edit($id)
28        {
29            //
30        }
31    }
32}
```



# Res Controller Created Update() Destroy()

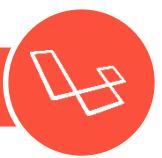


The screenshot shows a Visual Studio Code interface with the following details:

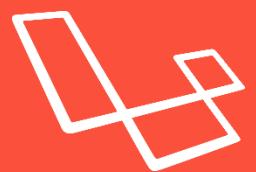
- File Menu:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** • ProductController.php - laravelcrud - Visual Studio Code
- Explorer View:** Shows the project structure under "LARAVELCRUD". The "ProductController.php" file is selected.
- Code Editor:** Displays the PHP code for the ProductController:

```
public function edit($id)
{
    //
}
/** 
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param int $id
 * @return \Illuminate\Http\Response
*/
public function update(Request $request, $id)
{
    //
}
/** 
 * Remove the specified resource from storage.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
*/
public function destroy($id)
{
    //
}
```

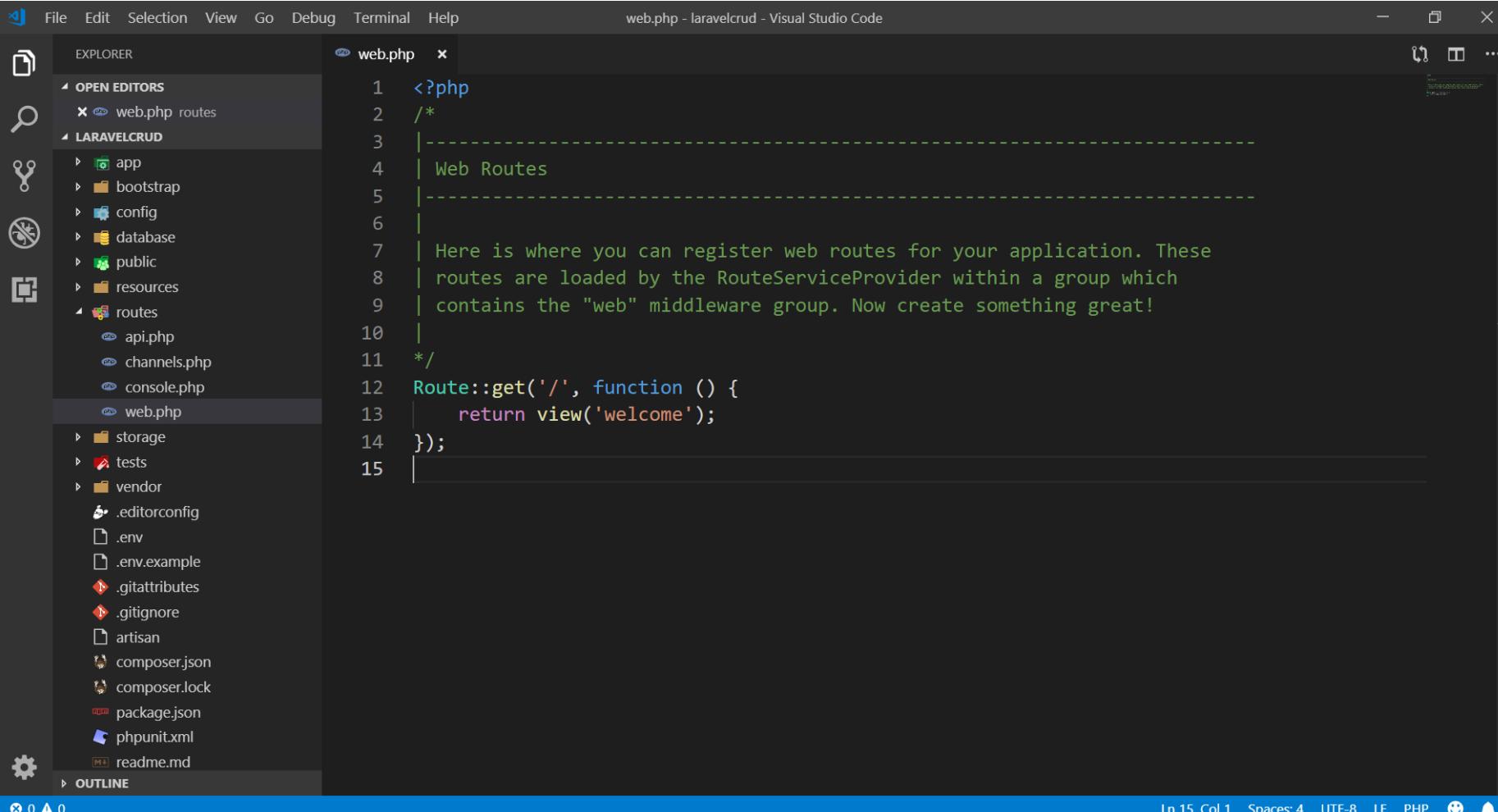
- Status Bar:** Ln 61, Col 1 | Spaces: 4 | UTF-8 | LF | PHP | Icons (0)



# Routes



# Goto Routes->web.php



The screenshot shows a Visual Studio Code interface with a dark theme. The left sidebar is the Explorer view, showing the project structure of a Laravel application named 'LARAVELCRUD'. The 'routes' folder contains files like api.php, channels.php, console.php, and web.php, with web.php currently selected. The main editor area displays the contents of web.php:

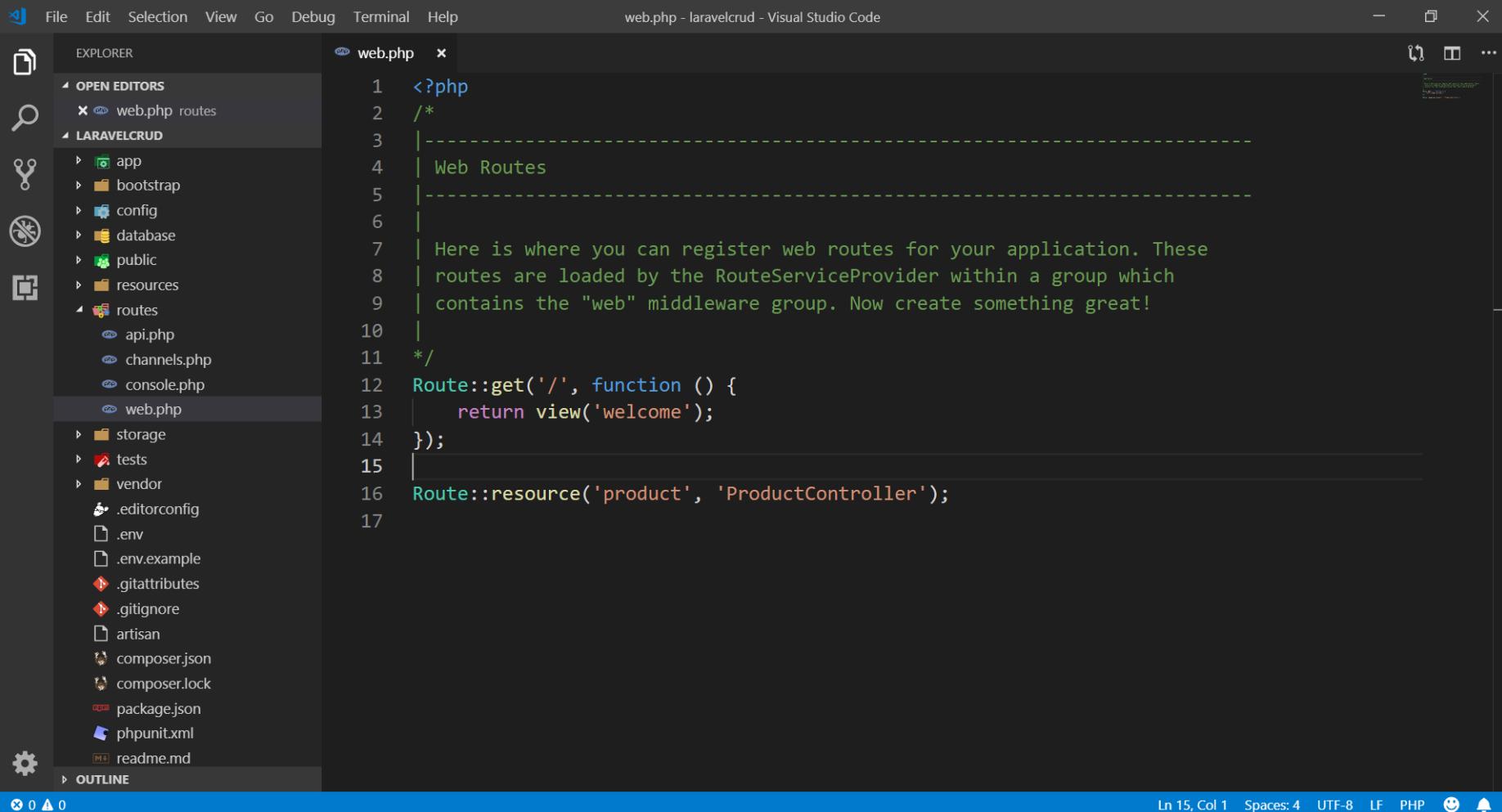
```
1 <?php
2 /*
3 | -----
4 | Web Routes
5 | -----
6 |
7 | Here is where you can register web routes for your application. These
8 | routes are loaded by the RouteServiceProvider within a group which
9 | contains the "web" middleware group. Now create something great!
10 |
11 */
12 Route::get('/', function () {
13     return view('welcome');
14 });
15 |
```

The status bar at the bottom shows the file is at line 15, column 1, with 4 spaces, using UTF-8 encoding, and the PHP language mode.



# Specify Route for Res Product Controller

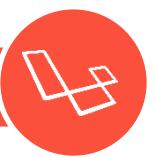
First is URL Second is Controller name



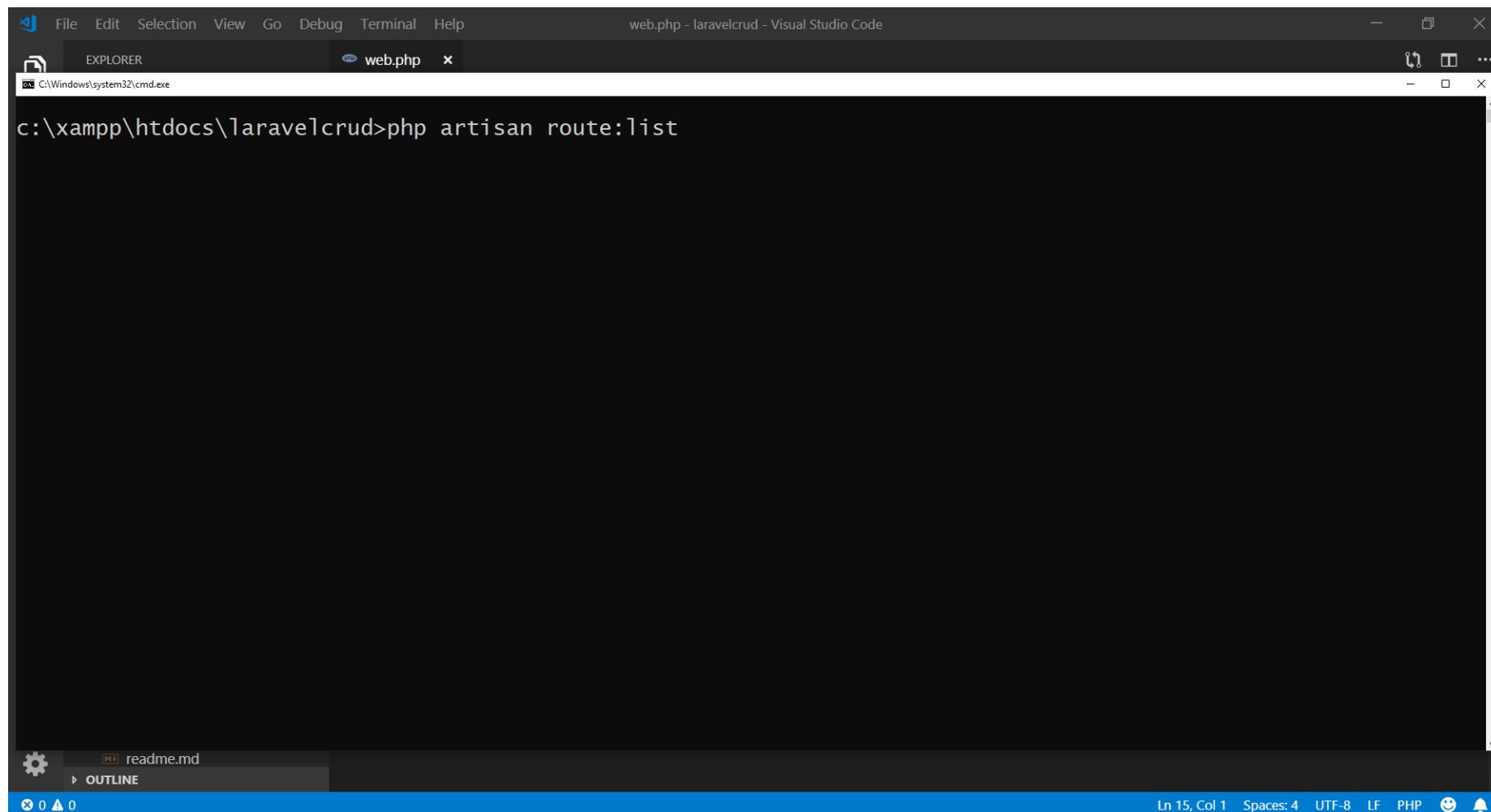
The screenshot shows the Visual Studio Code interface with the file `web.php` open in the editor. The code defines two routes: a simple get route for the root path '/' and a resource route for the 'product' controller. The code is as follows:

```
<?php
/*
|--------------------------------------------------------------------------
| Web Routes
|--------------------------------------------------------------------------
|
| Here is where you can register web routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| contains the "web" middleware group. Now create something great!
|
*/
Route::get('/', function () {
    return view('welcome');
});

Route::resource('product', 'ProductController');
```



# View Route Listing

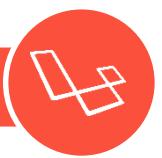


A screenshot of the Visual Studio Code interface. The title bar reads "web.php - laravelcrud - Visual Studio Code". The main area is a terminal window showing the command "c:\xampp\htdocs\laravelcrud>php artisan route:list". The bottom status bar shows "Ln 15, Col 1" and other file-related information.

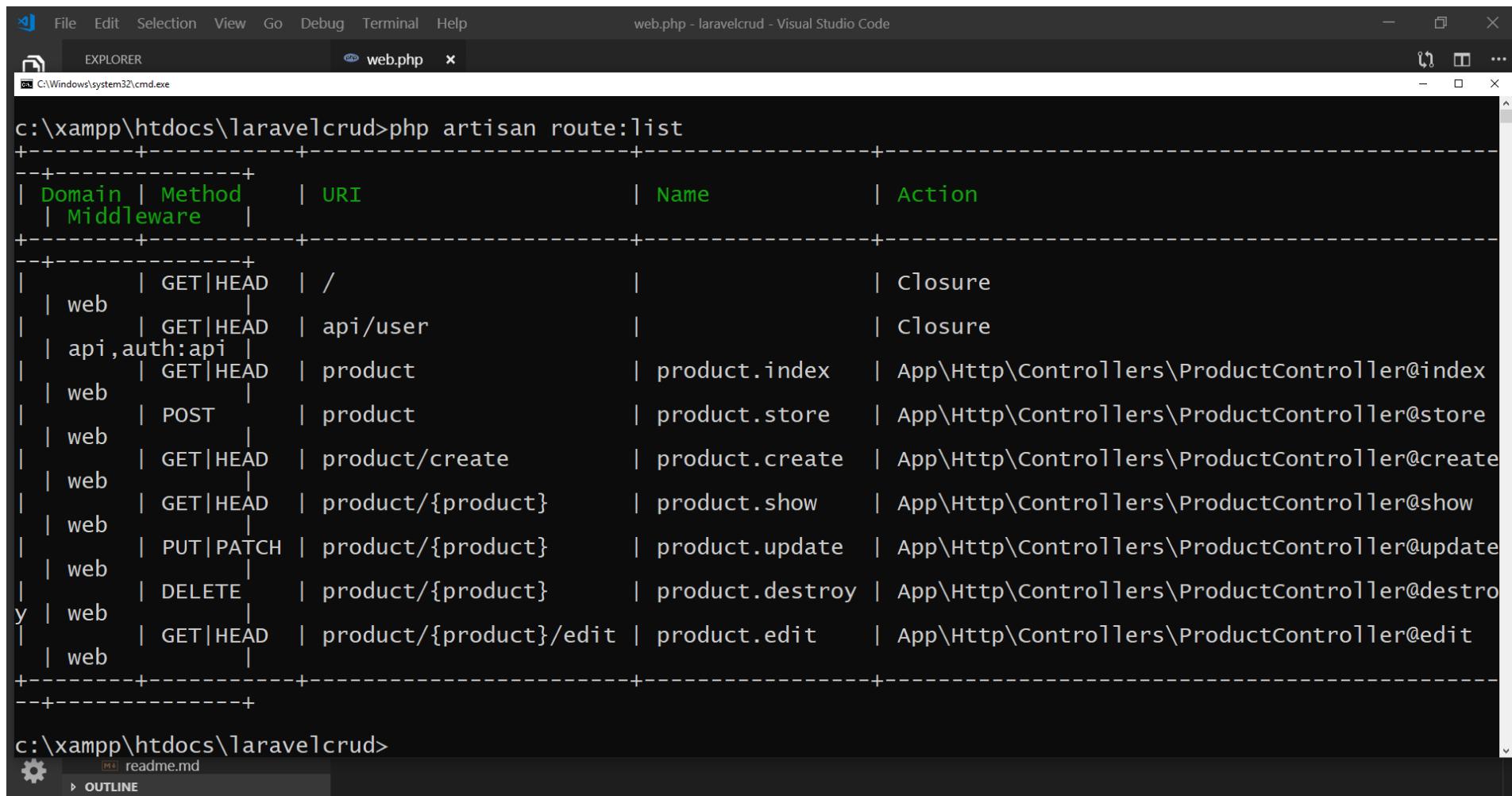


Akash Technolabs

[www.akashsir.com](http://www.akashsir.com)

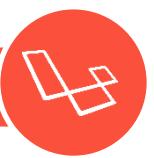


# All Routes are Ready to Use

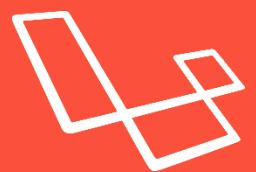


A screenshot of Visual Studio Code showing a terminal window titled "web.php - laravelcrud - Visual Studio Code". The terminal displays the output of the command "php artisan route:list". The output shows a table of routes with columns: Domain, Method, URI, Name, and Action. The routes listed include various methods (GET, POST, PUT, PATCH, DELETE) for different URIs such as '/', '/api/user', '/product', and '/product/{product}'. The "Name" column lists names like "closure", "product.index", "product.store", etc., and the "Action" column lists controller methods like "App\Http\Controllers\ProductController@index".

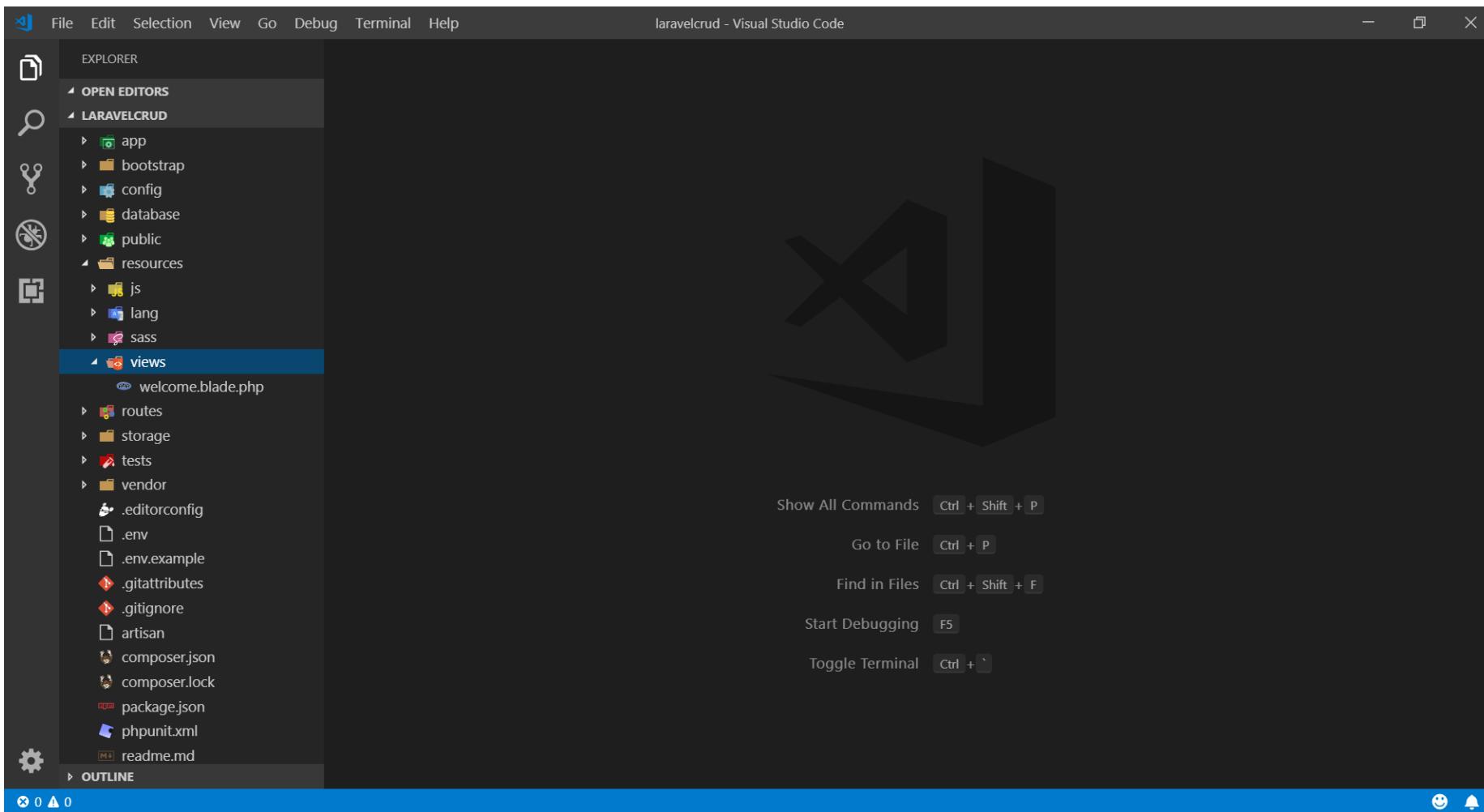
Domain	Method	URI	Name	Action
Middleware				
web	GET HEAD	/		closure
web	GET HEAD	api/user		closure
api,auth:api	GET HEAD			
web	GET HEAD	product	product.index	App\Http\Controllers\ProductController@index
web	POST	product	product.store	App\Http\Controllers\ProductController@store
web	GET HEAD	product/create	product.create	App\Http\Controllers\ProductController@create
web	GET HEAD	product/{product}	product.show	App\Http\Controllers\ProductController@show
web	PUT PATCH	product/{product}	product.update	App\Http\Controllers\ProductController@update
web	DELETE	product/{product}	product.destroy	App\Http\Controllers\ProductController@destroy
web	GET HEAD	product/{product}/edit	product.edit	App\Http\Controllers\ProductController@edit



# Views

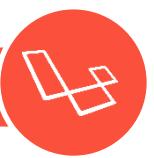


# Go to Views



Akash Technolabs

[www.akashsir.com](http://www.akashsir.com)



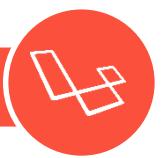
# Create Master.blade.php in Layouts Folder

The screenshot shows a Visual Studio Code interface with the following details:

- File Path:** master.blade.php - laravelcrud - Visual Studio Code
- Explorer:** Shows the project structure under "LARAVELCRUD". The "views/layouts" folder contains "master.blade.php" and "welcome.blade.php". Other folders like "bootstrap", "config", "database", "public", "resources", "routes", "storage", "tests", and "vendor" are also listed.
- Editor:** The code editor displays the following Blade template:

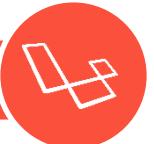
```
<html>
<body>
<center><h1> Laravel CURD</h1></center>
<hr/>
<a href='{{ route('product.create') }}'>Add</a> | 
<a href='{{ route('product.index') }}'>View</a>
@yield('content')
</body>
</html>
```

- Terminal:** Shows the command "php artisan serve" being run in the terminal, with the output indicating the Laravel development server is started at "http://127.0.0.1:8000".
- Status Bar:** Shows the file is 1: php, with line 9, column 18, spaces: 4, encoding: UTF-8, and file type: PHP.

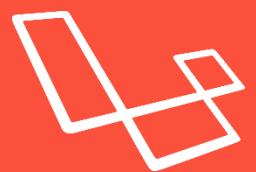


- Create new Folder in View named product

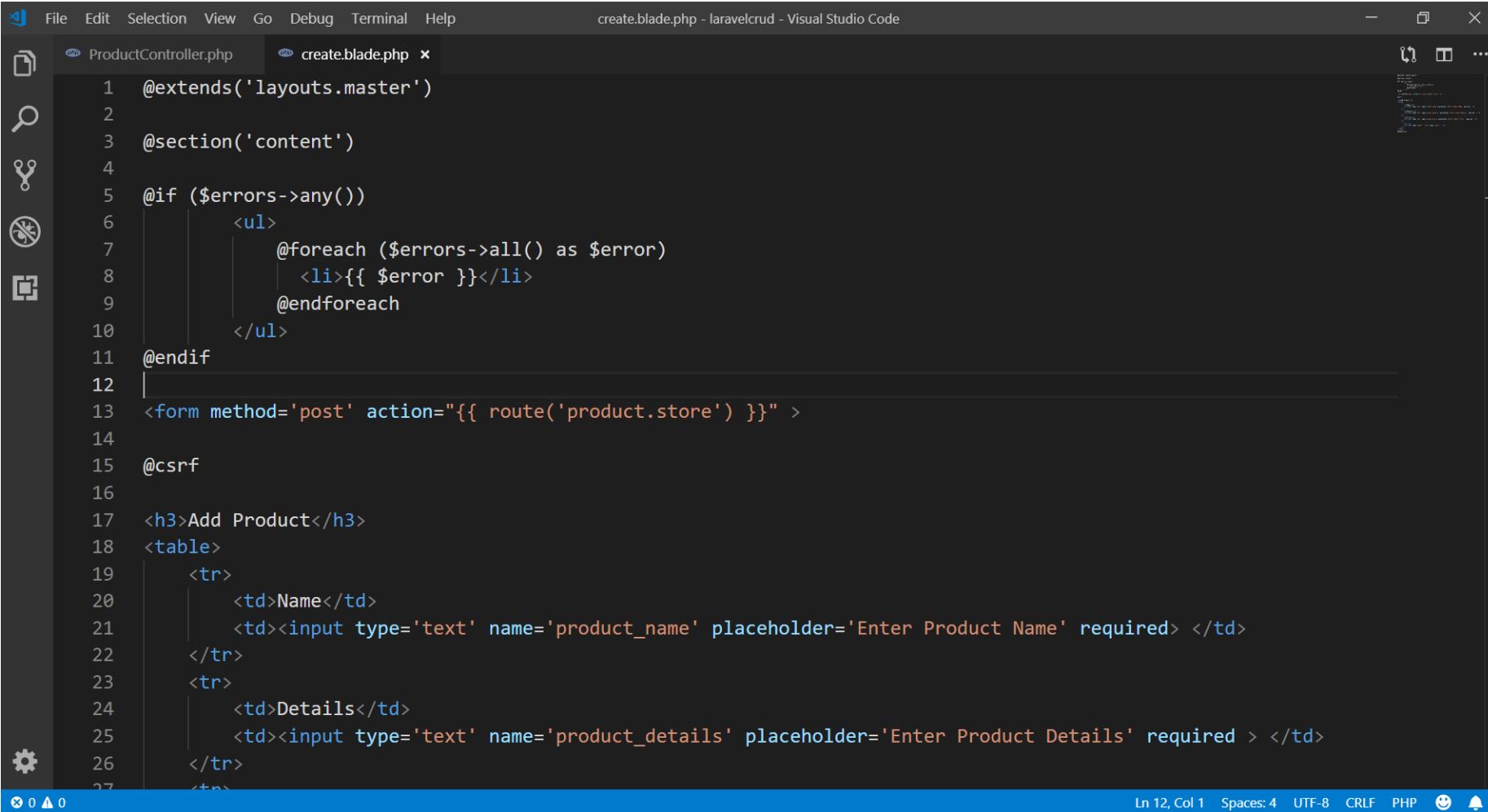
- Product
  - Index.blade.php
  - Create.blade.php
  - Edit.blade.php



# Insert Record



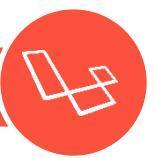
# Create New Folder Product in Views and Add create.blade.php file



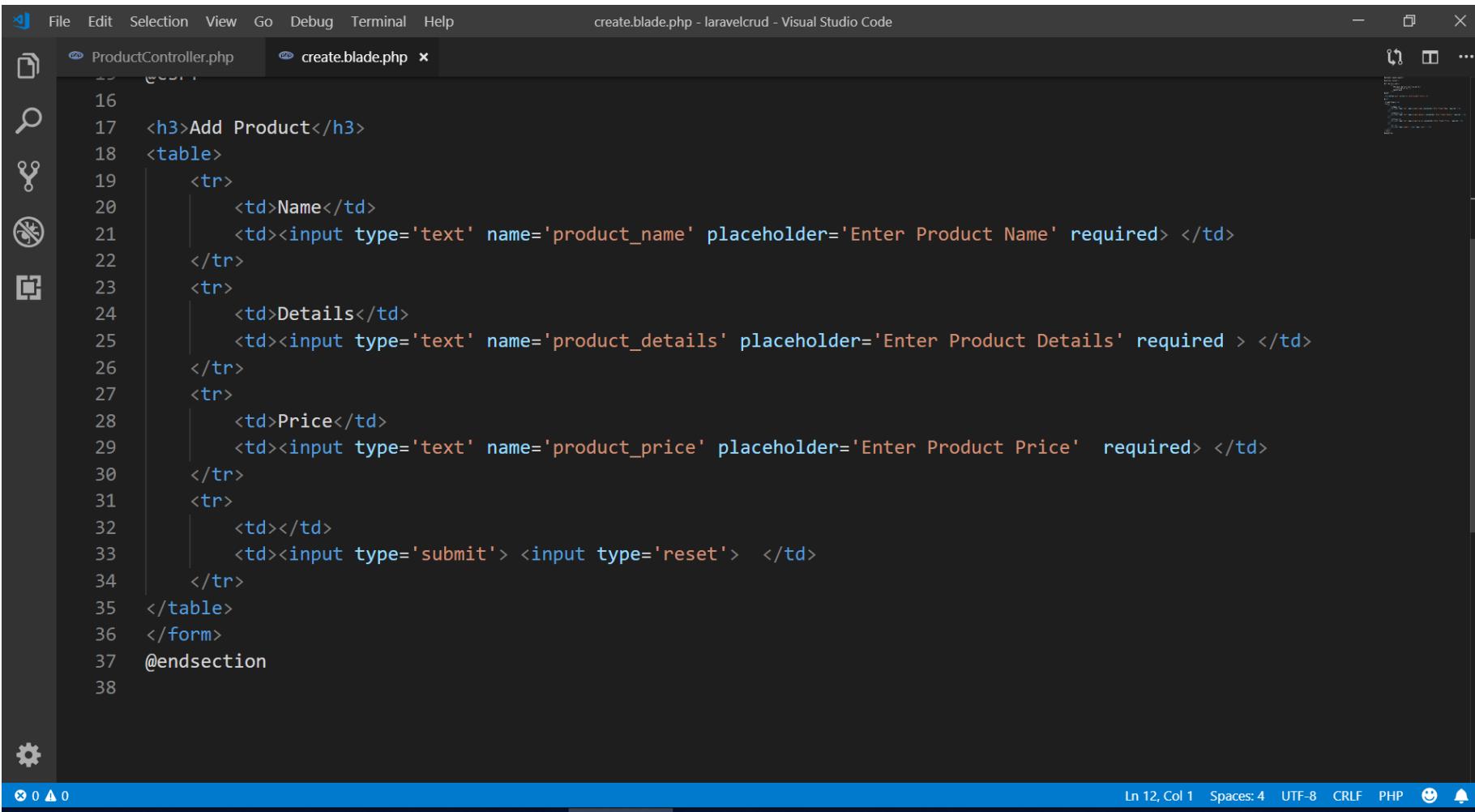
The screenshot shows a Visual Studio Code interface with a dark theme. The left sidebar has icons for File, Search, Find, and Settings. The top menu bar includes File, Edit, Selection, View, Go, Debug, Terminal, and Help. The title bar says "create.blade.php - laravelcrud - Visual Studio Code". The main editor area contains the following code:

```
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') }}" >
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 > </td>
26     </tr>
27     <tr>
```

The status bar at the bottom shows "Ln 12, Col 1" and "Spaces: 4" and other file details.



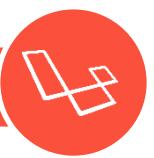
# Add Form HTML Controls



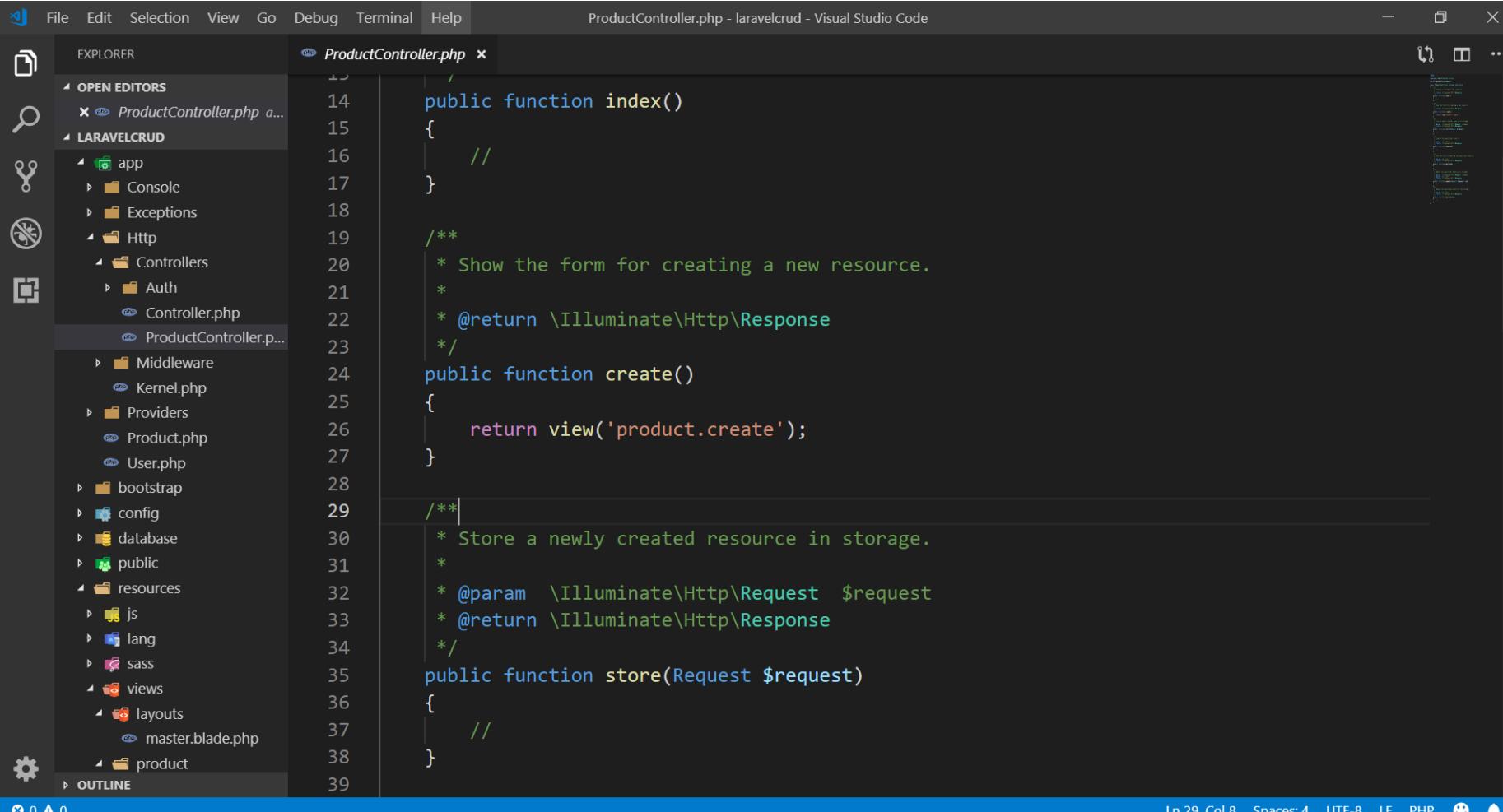
A screenshot of Visual Studio Code showing the code for a Laravel blade template named `create.blade.php`. The code defines a form for adding a product, featuring a table with four rows: Name, Details, Price, and a row containing submit and reset buttons. Each input field includes attributes for name, placeholder, and required validation.

```
File Edit Selection View Go Debug Terminal Help
ProductController.php create.blade.php - laravelcrud - Visual Studio Code
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 > </td>
26   </tr>
27   <tr>
28     <td>Price</td>
29     <td><input type='text' name='product_price' placeholder='Enter Product Price' required> </td>
30   </tr>
31   <tr>
32     <td></td>
33     <td><input type='submit'> <input type='reset'> </td>
34   </tr>
35 </table>
36 </form>
37 @endsection
38
```

Ln 12, Col 1 Spaces: 4 UTF-8 CRLF PHP ⚡ 🎵



# Product Controller Return FormView in Create()

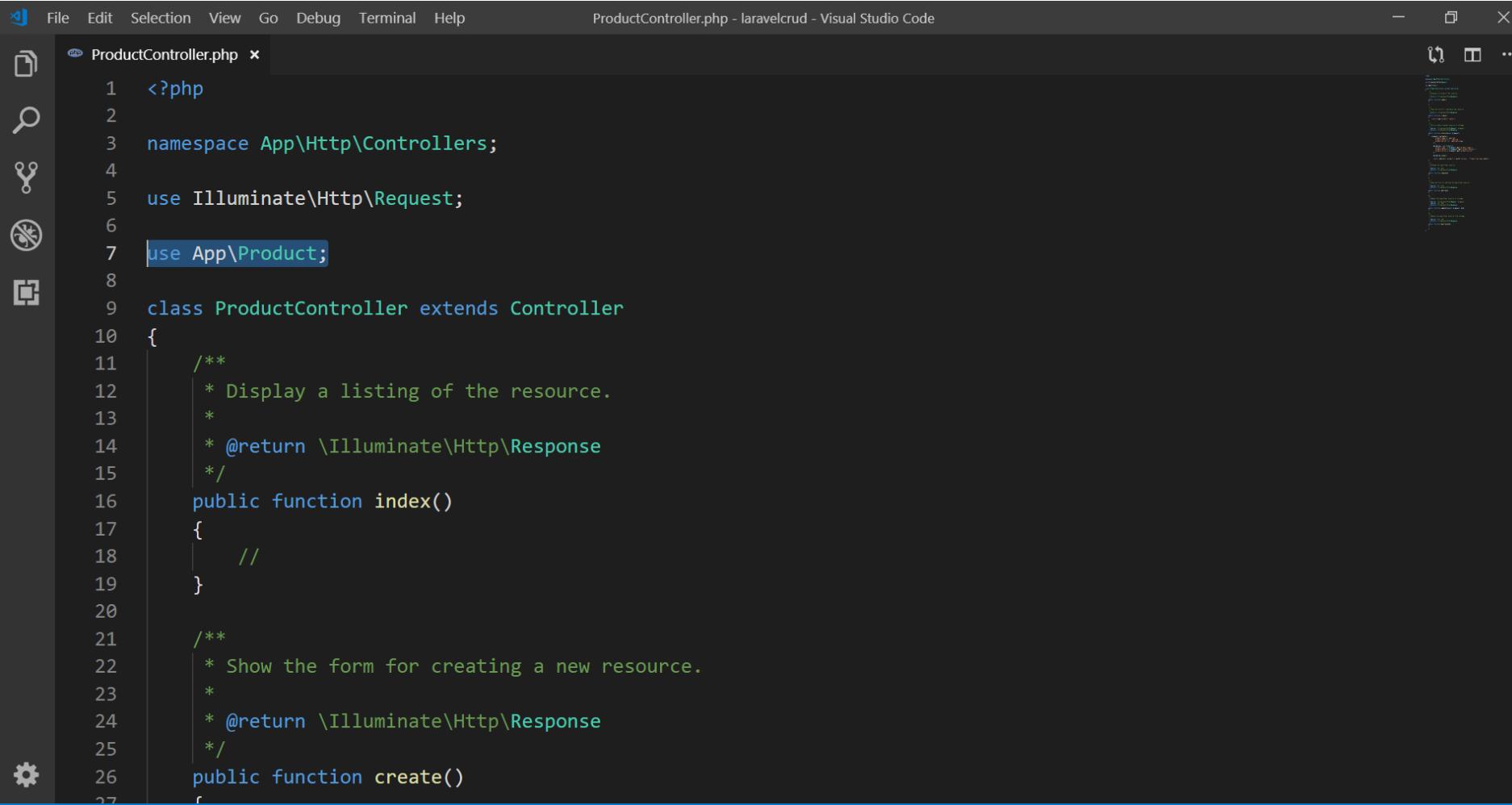


The screenshot shows a Visual Studio Code interface with the following details:

- File Path:** ProductController.php - laravelcrud - Visual Studio Code
- Code Content:** The code is a Laravel controller named ProductController.php. It contains three methods: index(), create(), and store().
  - index():** Shows the form for creating a new resource.
  - create():** Returns the view for product.create.
  - store(Request \$request):** Stores a newly created resource in storage.
- Explorer View:** Shows the project structure under LARAVELCRUD, including app, Http, Controllers, Providers, and various configuration and resource files like master.blade.php and product.blade.php.
- Status Bar:** Shows the current line (Ln 29), column (Col 8), and other settings like Spaces: 4, UTF-8, LF, PHP, and icons for file status and notifications.

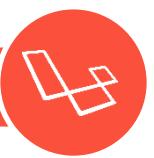


# Call Product Model in ProductController

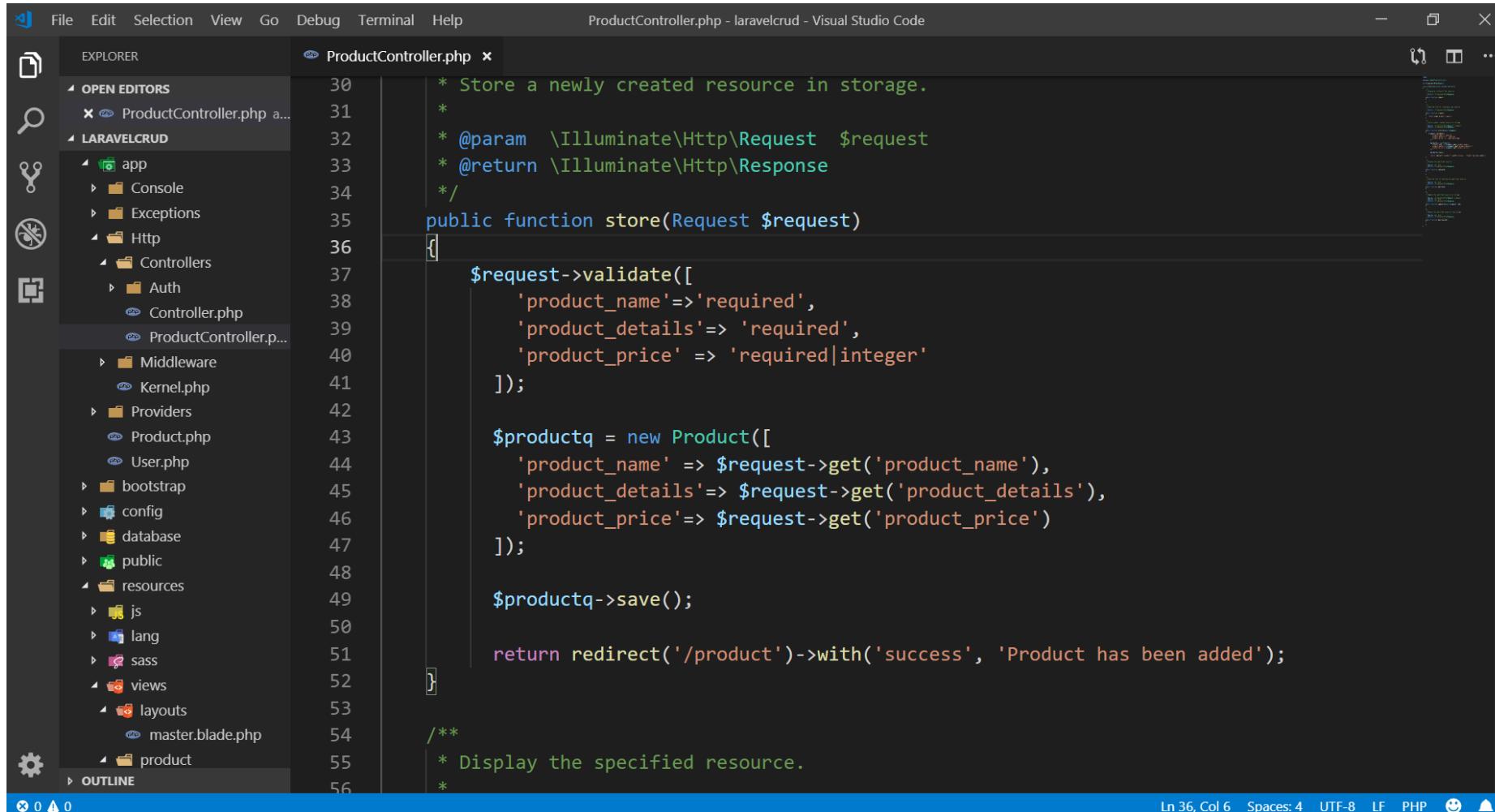


A screenshot of Visual Studio Code showing the file `ProductController.php`. The code defines a controller class `ProductController` that extends `Controller`. It contains two methods: `index()` and `create()`. The `index()` method is annotated with `@return Illuminate\Http\Response`. The `create()` method is also annotated with `@return Illuminate\Http\Response`. The code uses the `App\Product` model, which is imported via `use App\Product;`.

```
<?php  
namespace App\Http\Controllers;  
use Illuminate\Http\Request;  
use App\Product;  
  
class ProductController extends Controller  
{  
    /**  
     * Display a listing of the resource.  
     *  
     * @return Illuminate\Http\Response  
     */  
    public function index()  
    {  
        //  
    }  
  
    /**  
     * Show the form for creating a new resource.  
     *  
     * @return Illuminate\Http\Response  
     */  
    public function create()  
    {  
        //  
    }  
}
```



# Process Post Method Data in Store()



The screenshot shows a Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** ProductController.php - laravelcrud - Visual Studio Code.
- Explorer:** Shows the project structure under "OPEN EDITORS" and "LARAVELCRUD".
- Code Editor:** Displays the PHP code for the `store` method in `ProductController.php`. The code handles validation, creates a new `Product` instance, saves it, and then redirects with a success message.
- Status Bar:** Shows the current file is `ProductController.php`, line 36, column 6, with 4 spaces, using UTF-8 encoding, and includes icons for file status and notifications.

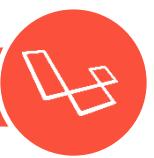
```
* Store a newly created resource in storage.
*
* @param \Illuminate\Http\Request $request
* @return \Illuminate\Http\Response
*/
public function store(Request $request)
{
    $request->validate([
        'product_name'=>'required',
        'product_details'=> 'required',
        'product_price' => 'required|integer'
    ]);

    $productq = new Product([
        'product_name' => $request->get('product_name'),
        'product_details'=> $request->get('product_details'),
        'product_price'=> $request->get('product_price')
    ]);

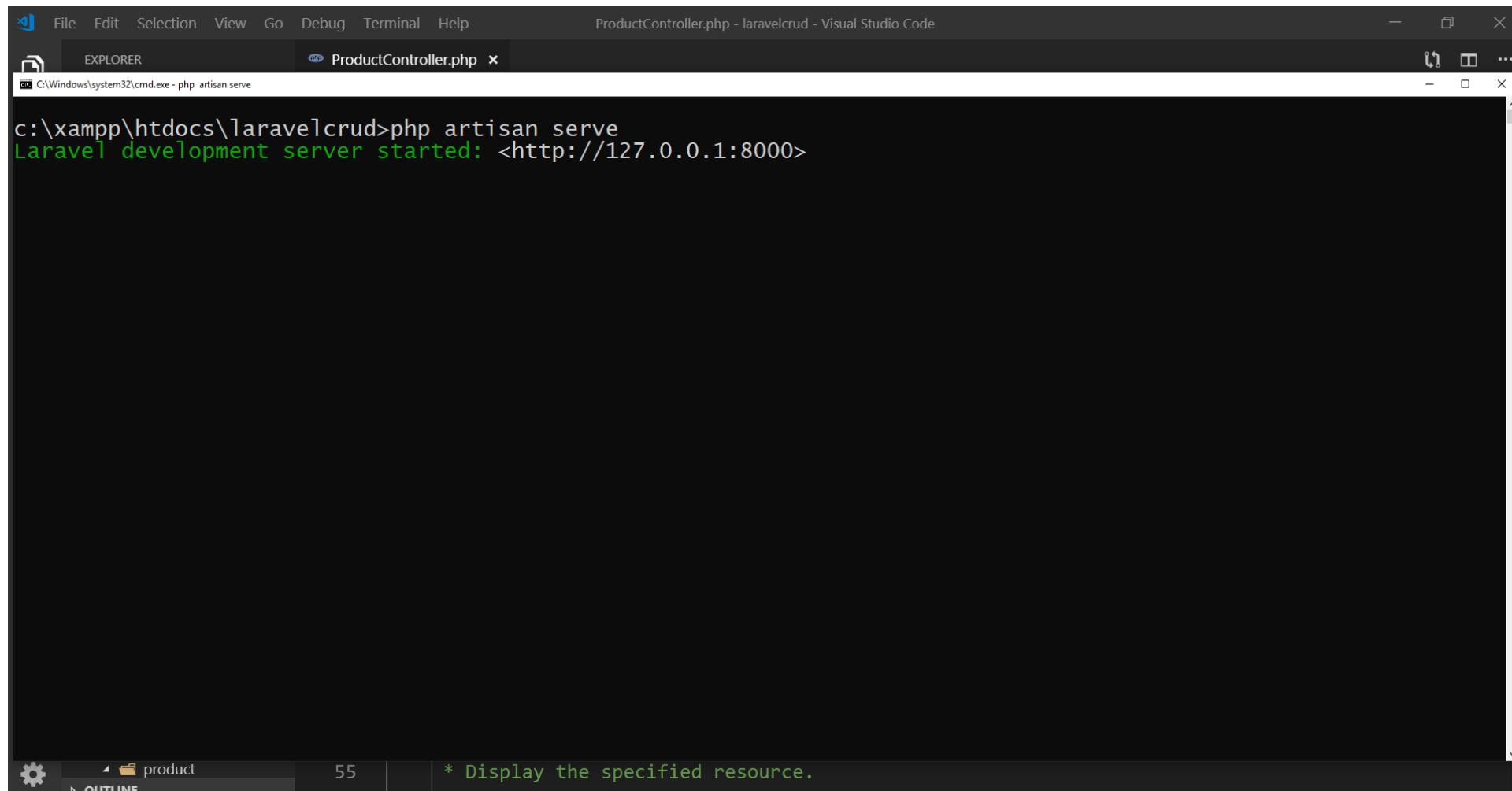
    $productq->save();

    return redirect('/product')->with('success', 'Product has been added');
}

/**
 * Display the specified resource.
*/
```



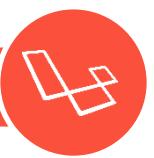
# Run the Project Using Artisan Serve



A screenshot of a Visual Studio Code interface. The title bar says "ProductController.php - laravelcrud - Visual Studio Code". The main area is a terminal window with the following text:

```
c:\xampp\htdocs\laravelcrud>php artisan serve
Laravel development server started: <http://127.0.0.1:8000>
```

The terminal window has a dark background and white text. At the bottom of the screen, there is a status bar with icons and text: "product" (with a gear icon), "OUTLINE" (with a folder icon), "55", and "\* Display the specified resource.".



# Product/Create run Form Design

127.0.0.1:8000/product/create

Laravel CURD

Add | View

**Add Product**

Name

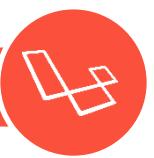
Details

Price



Akash Technolabs

www.akashsir.com



# Add Some Data

The screenshot shows a web browser window with the URL `127.0.0.1:8000/product/create`. The page title is "Laravel CURD". Below the title, there are two links: "Add" and "View". The main content is titled "Add Product". It contains three input fields: "Name" with value "iPhone XR", "Details" with value "iphone xr", and "Price" with value "85000". At the bottom are "Submit" and "Reset" buttons.

127.0.0.1:8000/product/create

Laravel CURD

Add | View

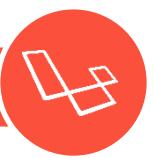
Add Product

Name iPhone XR

Details iphone xr

Price 85000

Submit Reset



# Data Added in Database

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

Server: 127.0.0.1 » Database: laravelcrud » Table: products

Browse Structure SQL Search Insert Export Import Privileges More

Showing rows 0 - 0 (1 total, Query took 0.0008 seconds.)

SELECT \* FROM `products`

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

Show all Number of rows: 25 Filter rows: Search this table

product_id	product_name	product_details	product_price	created_at	updated_at
1	iPhone XR	iPhone XR	85000	2019-02-11 17:46:07	2019-02-11 17:46:07

+ Options

Check all With selected: Edit Copy Delete Export

Show all Number of rows: 25 Filter rows: Search this table

Query results operations

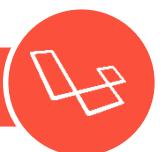
Print Copy to clipboard Export Display chart Create view

Bookmark this SQL query

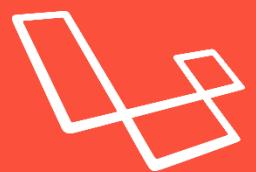
The screenshot shows the phpMyAdmin interface for the 'laravelcrud' database. The 'products' table is selected. A single row is displayed with the following data:

product_id	product_name	product_details	product_price	created_at	updated_at
1	iPhone XR	iPhone XR	85000	2019-02-11 17:46:07	2019-02-11 17:46:07

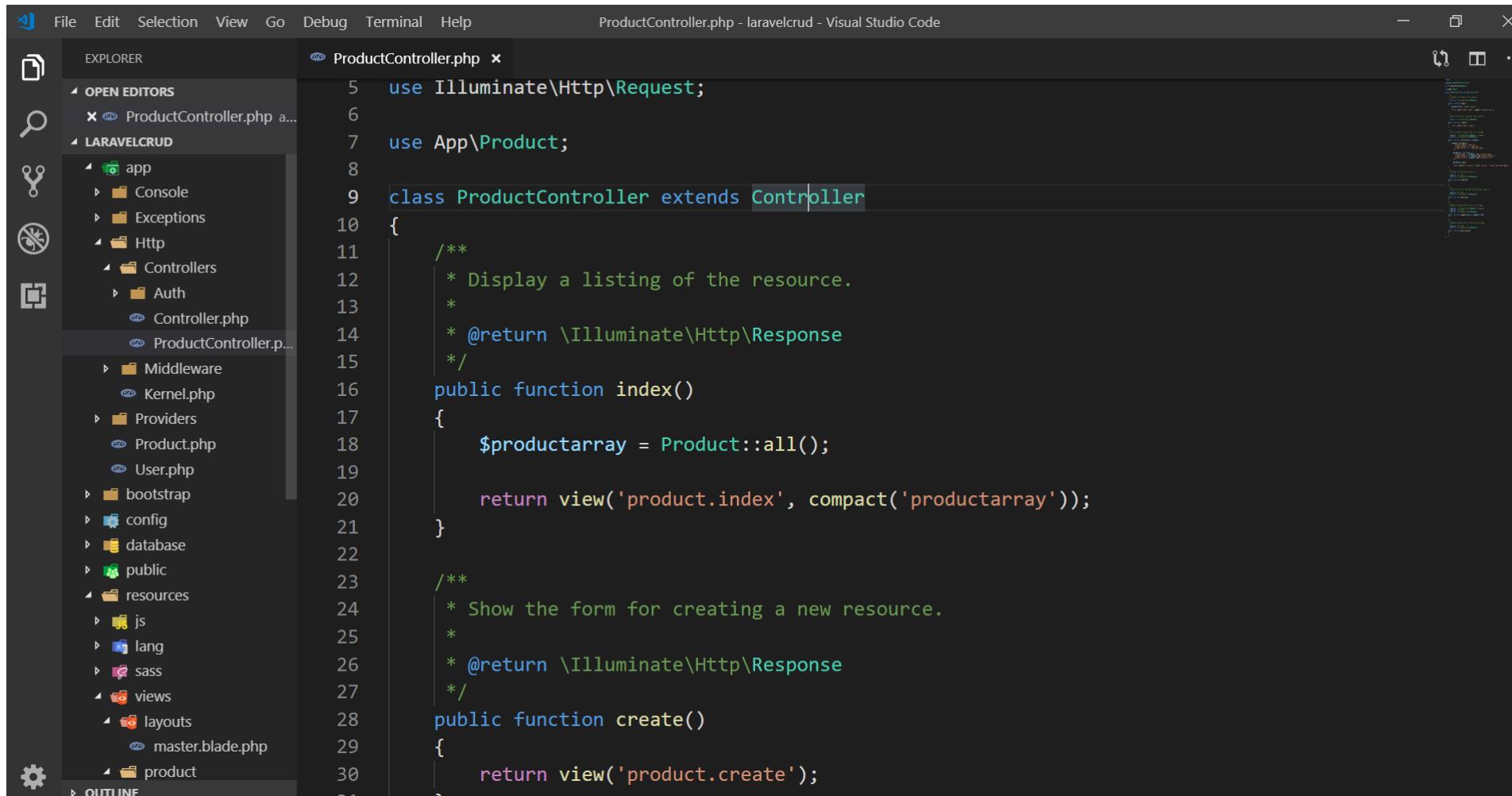
Below the table, there are various navigation and operation buttons like 'Edit', 'Copy', 'Delete', 'Export', etc.



# Display Record



# Fetch Data From Table and Store in and Array



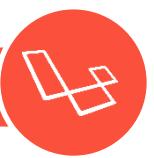
A screenshot of Visual Studio Code showing the ProductController.php file. The code fetches all products from the database and stores them in an array.

```
use Illuminate\Http\Request;
use App\Product;

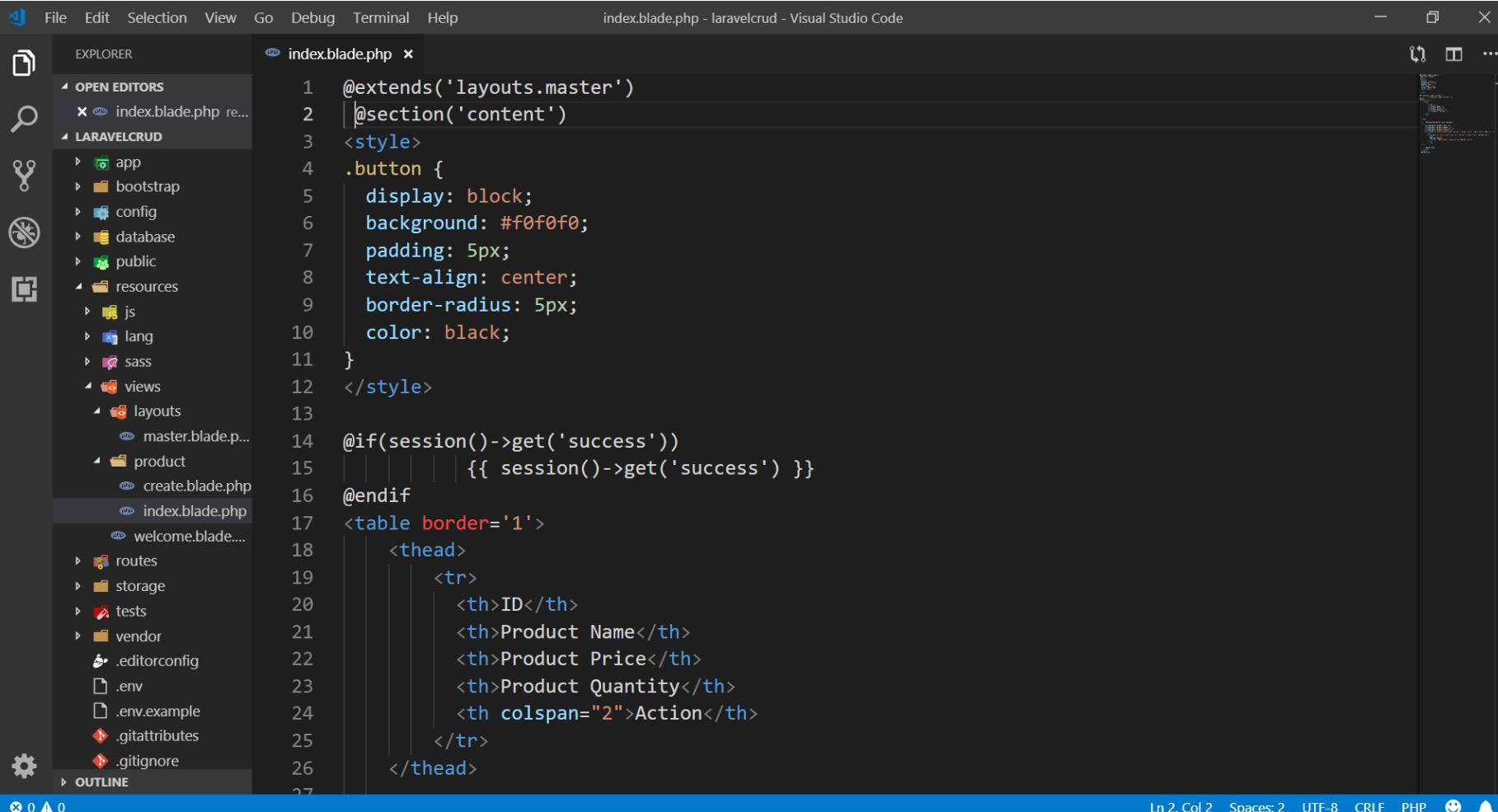
class ProductController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $productarray = Product::all();

        return view('product.index', compact('productarray'));
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        return view('product.create');
    }
}
```



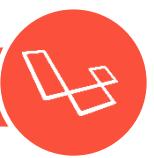
# Create index.blade.php in Product Folder



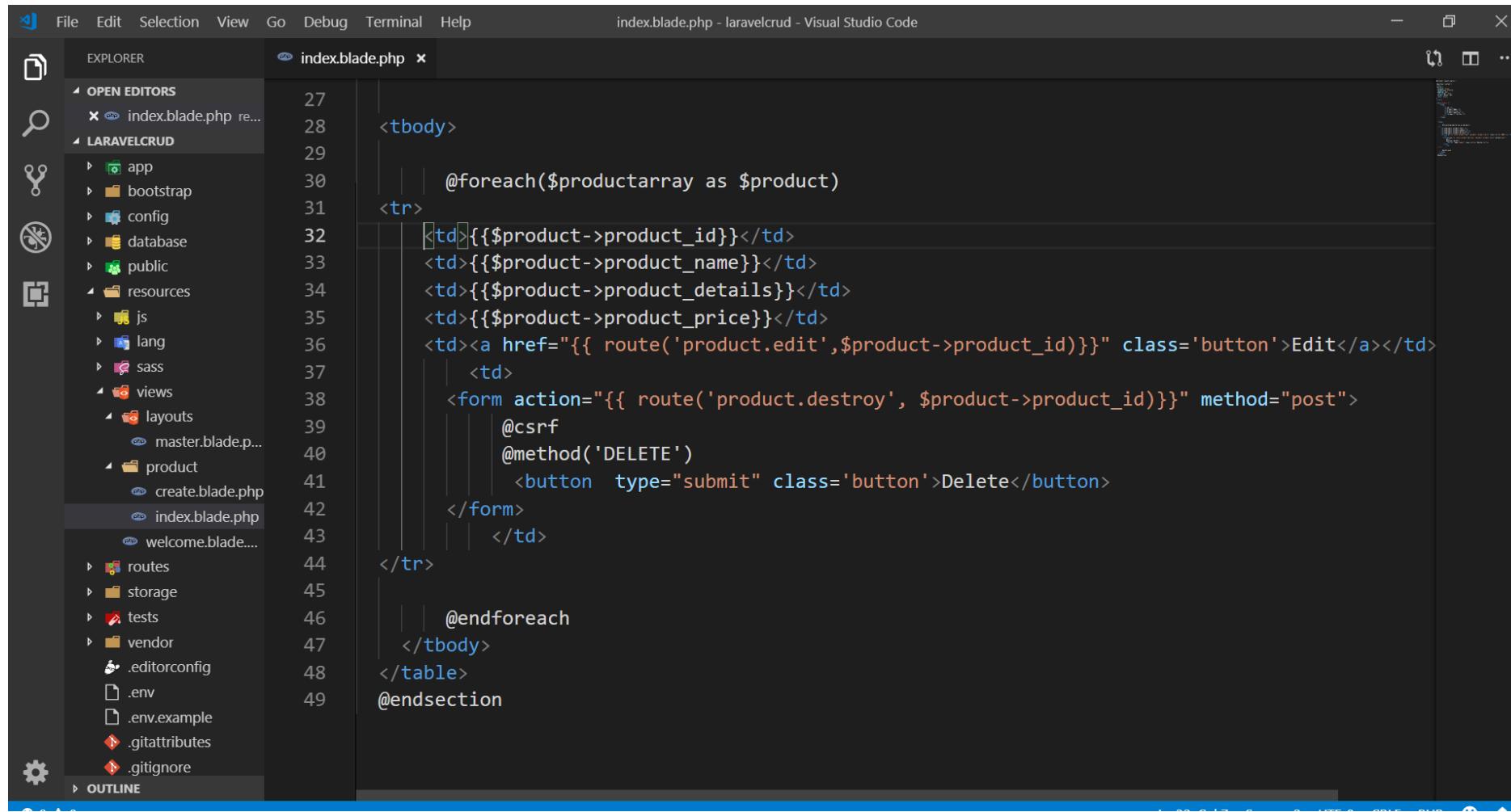
The screenshot shows the Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** index.blade.php - laravelcrud - Visual Studio Code.
- Explorer Panel:** Shows the project structure under 'LARAVELCRUD'. The 'index.blade.php' file is currently selected.
- Editor Panel:** Displays the code for 'index.blade.php'. The code includes Blade templating syntax like @extends, @section, @style, @if, @endif, and HTML elements like <table>, <thead>, and <th>.
- Bottom Status Bar:** Shows 'Ln 2, Col 2' and other file metadata.

```
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 colspan="2">Action</th>
25        </tr>
26    </thead>
```

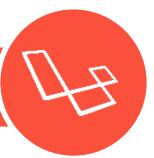


# Fetch Data and print in HTML Table

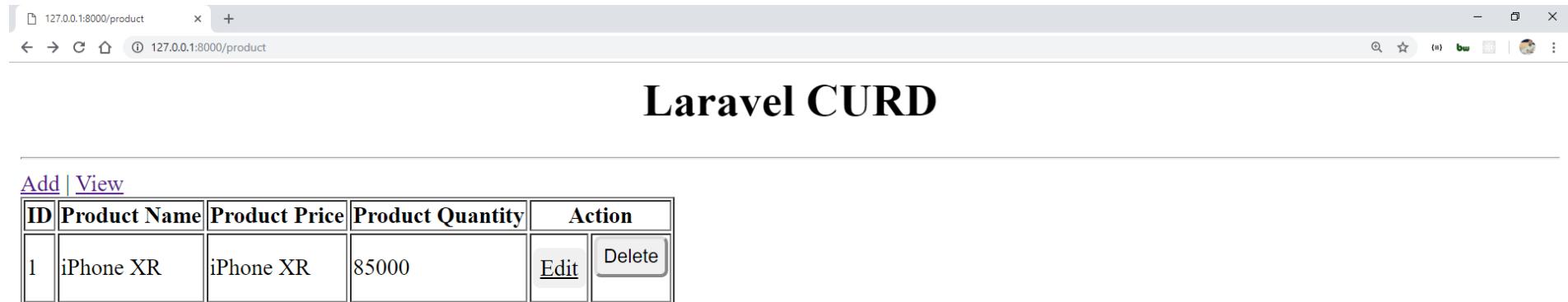


The screenshot shows a Visual Studio Code interface with a dark theme. The left sidebar contains the Explorer, showing a file tree for a Laravel project named 'LARAVELCRUD'. The 'index.blade.php' file is open in the main editor area. The code displays a table structure using Laravel's Blade templating syntax. It iterates over an array of products using a foreach loop. Each row ('tr') contains four data cells ('td'): product ID, product name, product details, and product price. The fourth column also contains links for 'Edit' and 'Delete' actions, generated by Laravel's route helper. The 'Delete' button uses the DELETE HTTP method.

```
<tbody>
    @foreach($productarray as $product)
        <tr>
            <td>{{$product->product_id}}</td>
            <td>{{$product->product_name}}</td>
            <td>{{$product->product_details}}</td>
            <td>{{$product->product_price}}</td>
            <td><a href="{{ route('product.edit', $product->product_id) }}" class='button'>Edit</a></td>
            <td>
                <form action="{{ route('product.destroy', $product->product_id) }}" method="post">
                    @csrf
                    @method('DELETE')
                    <button type="submit" class='button'>Delete</button>
                </form>
            </td>
        </tr>
    @endforeach
</tbody>
</table>
@endsection
```



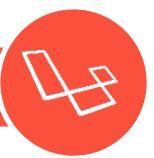
# Display Record From Table



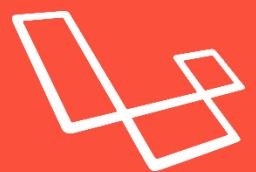
A screenshot of a web browser displaying a Laravel CURD application. The title bar shows the URL `127.0.0.1:8000/product`. The main content area has a heading **Laravel CURD**. Below it is a table with the following data:

ID	Product Name	Product Price	Product Quantity	Action
1	iPhone XR	iPhone XR	85000	<a href="#">Edit</a> <a href="#">Delete</a>

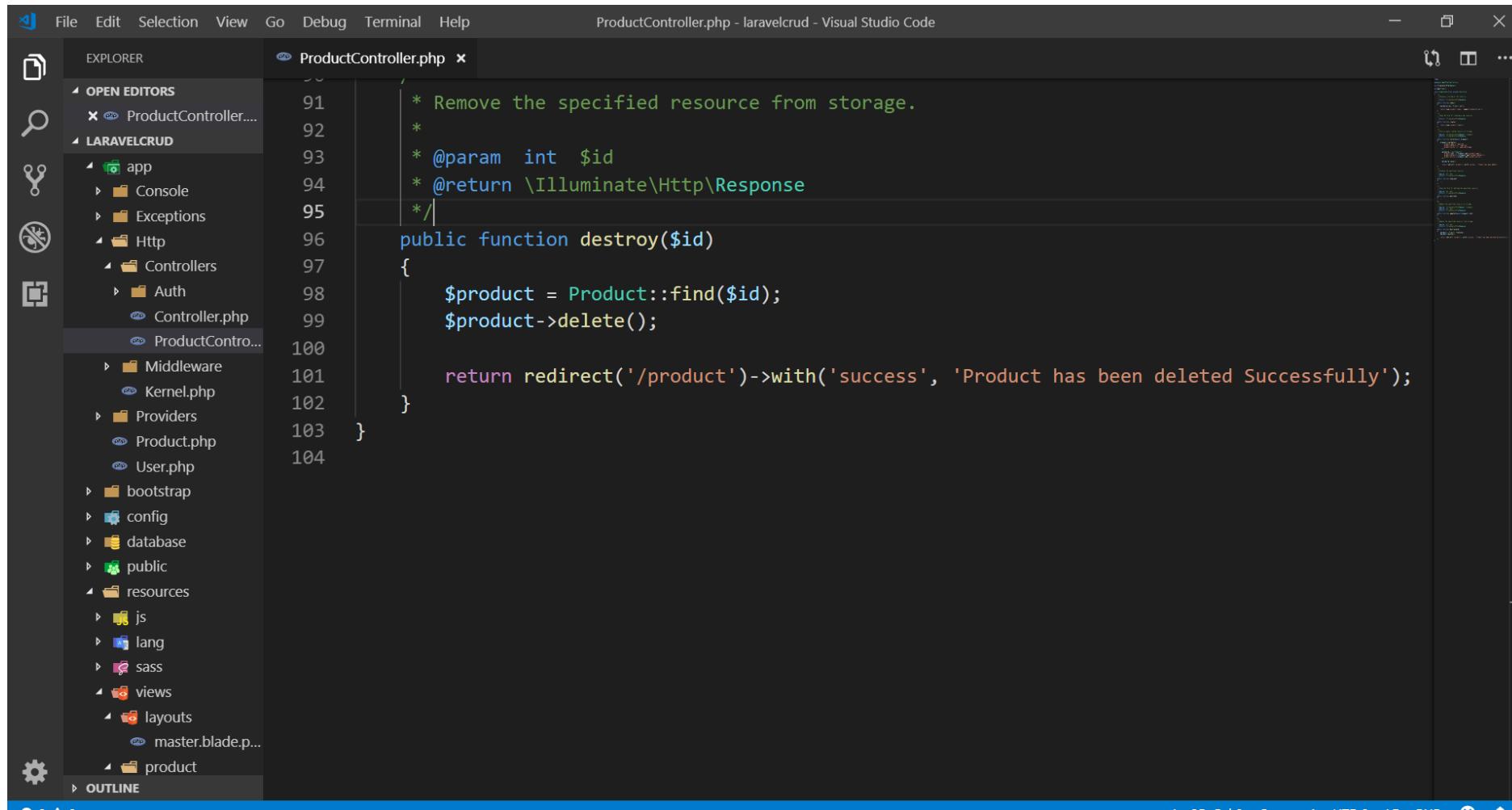
At the top left of the table, there are links [Add](#) and [View](#).



# Delete

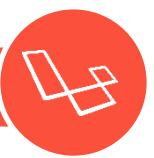


# Delete Record From Table



A screenshot of Visual Studio Code showing the code for `ProductController.php`. The code implements a `destroy` method to delete a product from storage.

```
* Remove the specified resource from storage.  
* @param int $id  
* @return \Illuminate\Http\Response  
*/  
  
public function destroy($id)  
{  
    $product = Product::find($id);  
    $product->delete();  
  
    return redirect('/product')->with('success', 'Product has been deleted Successfully');  
}  
  
101 }  
102 }  
103 }  
104 }
```



# Run product in URL

Laravel CURD

ID	Product Name	Product Price	Product Quantity	Action
1	iPhone XR	iPhone XR	85000	<a href="#">Edit</a> <a href="#">Delete</a>
2	Test	Test	11	<a href="#">Edit</a> <a href="#">Delete</a>

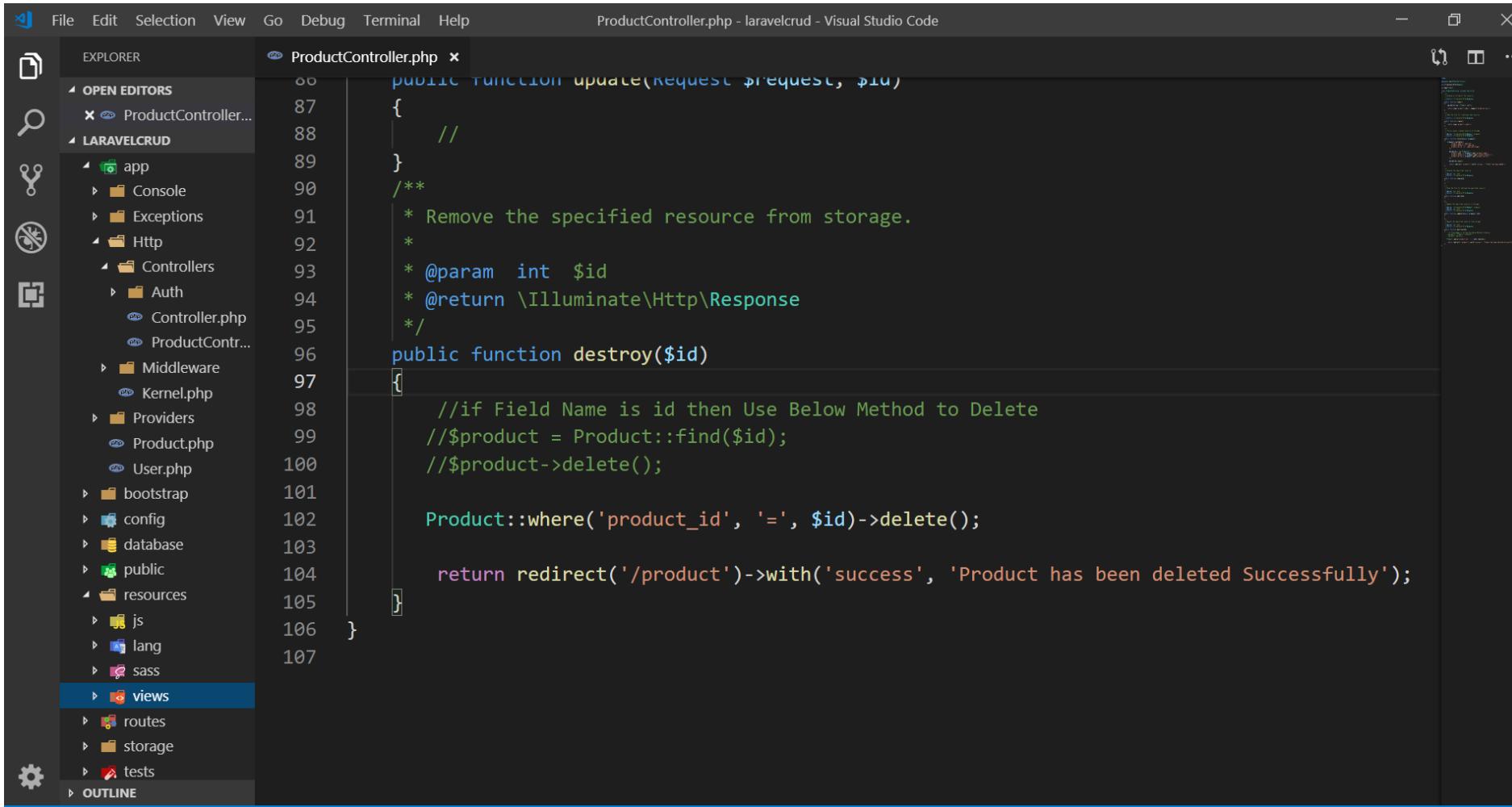


Akash Technolabs

[www.akashsir.com](http://www.akashsir.com)



# Delete Record Method 2



The screenshot shows a Visual Studio Code interface with the following details:

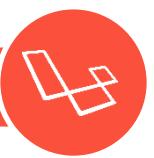
- File Menu:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** ProductController.php - laravelcrud - Visual Studio Code.
- Explorer Panel (Left):** Shows the project structure under "LARAVELCRUD".
  - app
  - Console
  - Exceptions
  - Http
    - Controllers
      - Auth
        - Controller.php
        - ProductContr...
    - Middleware
      - Kernel.php
    - Providers
      - Product.php
      - User.php
  - bootstrap
  - config
  - database
  - public
  - resources
    - js
    - lang
    - sass
  - views
    - selected
  - routes
  - storage
  - tests
- Outline Panel (Bottom Left):** Shows the outline of the current file.
- Code Editor (Right):** Displays the PHP code for the ProductController.php file.

```
public function update(Request $request, $id)
{
    //

    /**
     * Remove the specified resource from storage.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function destroy($id)
    {
        //if Field Name is id then Use Below Method to Delete
        //$product = Product::find($id);
        //$product->delete();

        Product::where('product_id', '=', $id)->delete();

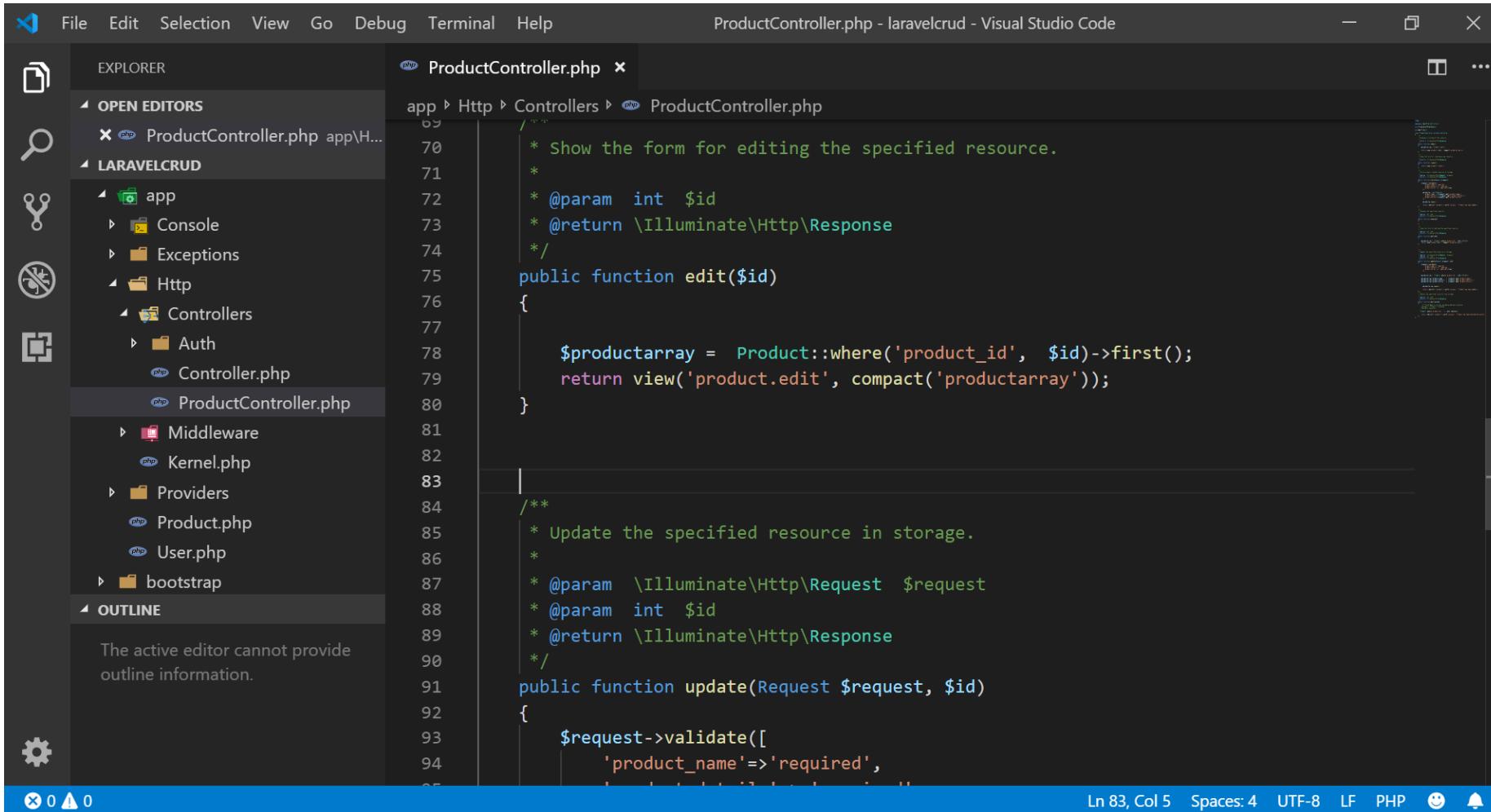
        return redirect('/product')->with('success', 'Product has been deleted Successfully');
    }
}
```



# Edit



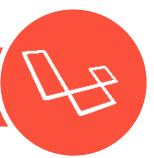
# Edit Method Code



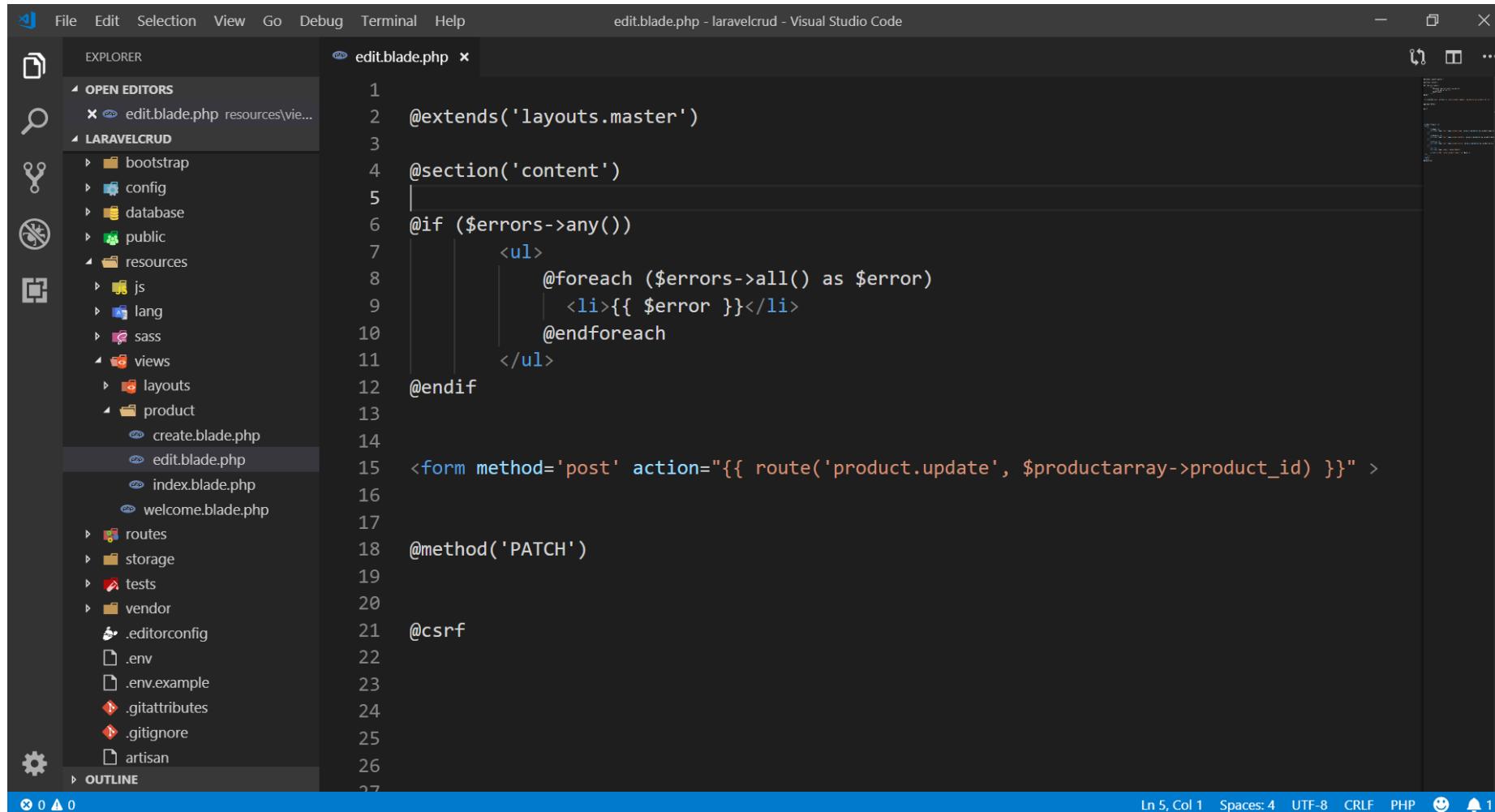
The screenshot shows a Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** ProductController.php - laravelcrud - Visual Studio Code.
- Explorer Panel:** Shows the file structure of a Laravel application. The current file, ProductController.php, is selected in the "OPEN EDITORS" section. Other sections include LARAVELCRUD, app, Http, Controllers, Auth, Middleware, Kernel.php, Providers, Product.php, User.php, and bootstrap.
- Code Editor:** Displays the PHP code for ProductController.php. The code includes methods for editing and updating products.
- Status Bar:** Shows the current line (Ln 83), column (Col 5), spaces used (Spaces: 4), encoding (UTF-8), line separator (LF), PHP, and a few icons.

```
ProductController.php
1 * Show the form for editing the specified resource.
2 *
3 * @param int $id
4 * @return \Illuminate\Http\Response
5 */
6 public function edit($id)
7 {
8     $productarray = Product::where('product_id', $id)->first();
9     return view('product.edit', compact('productarray'));
10 }
11
12 /**
13 * Update the specified resource in storage.
14 *
15 * @param \Illuminate\Http\Request $request
16 * @param int $id
17 * @return \Illuminate\Http\Response
18 */
19 public function update(Request $request, $id)
20 {
21     $request->validate([
22         'product_name'=>'required',
23     ]);
24 }
```



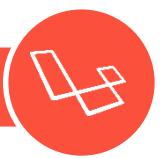
# Create Edit.blade.php in product Folder



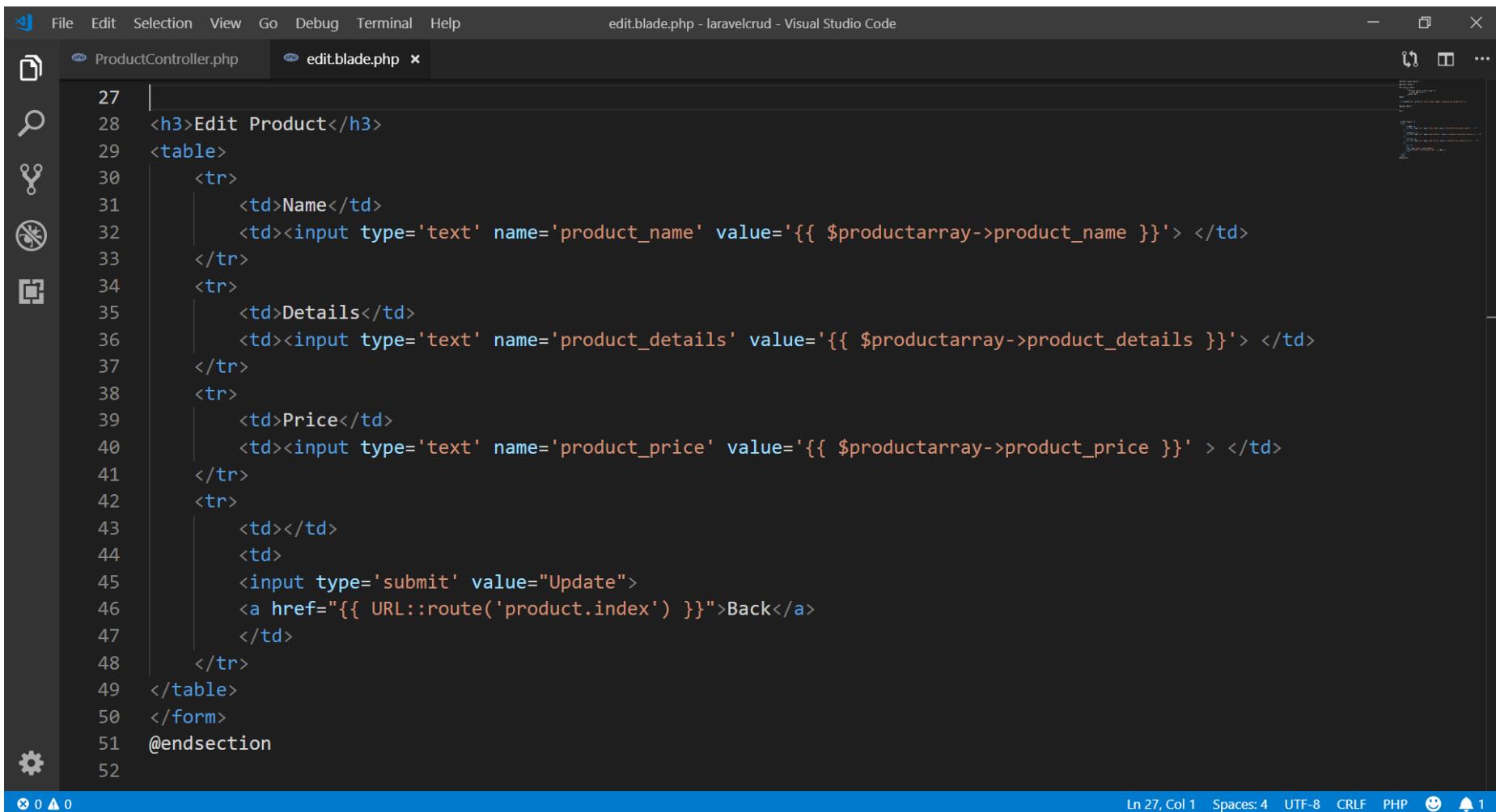
The screenshot shows a Visual Studio Code interface with a dark theme. The left sidebar displays the project structure under 'LARAVELCRUD'. The 'edit.blade.php' file is open in the main editor area. The code is a Blade template for an edit form:

```
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
14 <form method='post' action="{{ route('product.update', $productarray->product_id) }}" >
15
16
17 @method('PATCH')
18
19
20 @csrf
21
22
23
24
25
26
27
```

The status bar at the bottom shows: Ln 5, Col 1 Spaces: 4 UTF-8 CRLF PHP 1



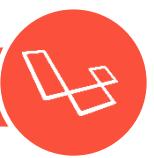
# Edit Form Code



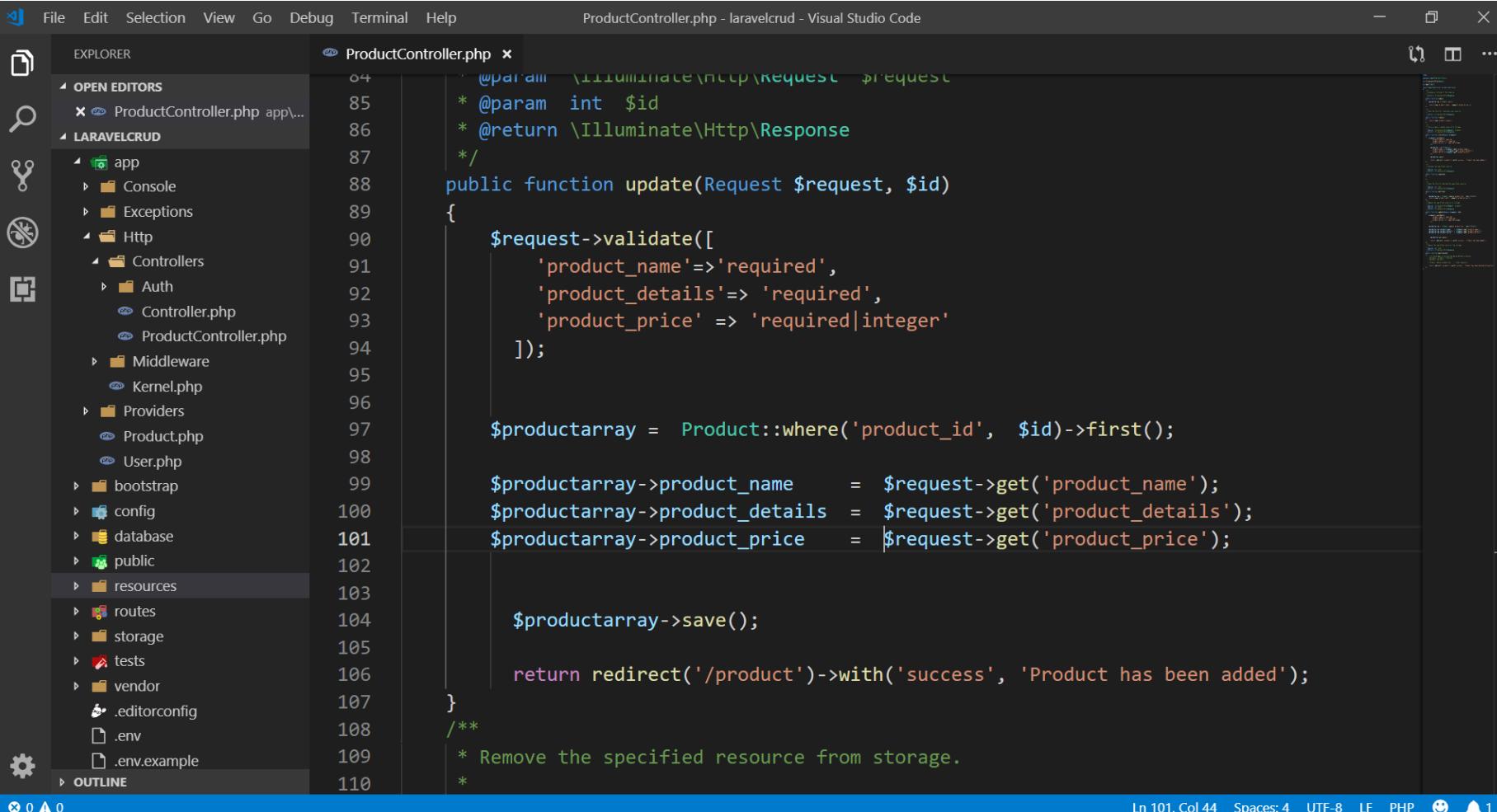
A screenshot of the Visual Studio Code interface showing the code for an edit form. The file is named 'edit.blade.php' and is part of a Laravel CRUD application. The code uses Blade templating syntax to create a table-based form for editing product details.

```
27 <h3>Edit Product</h3>
28 <table>
29   <tr>
30     <td>Name</td>
31     <td><input type='text' name='product_name' value='{{ $productarray->product_name }}'> </td>
32   </tr>
33   <tr>
34     <td>Details</td>
35     <td><input type='text' name='product_details' value='{{ $productarray->product_details }}'> </td>
36   </tr>
37   <tr>
38     <td>Price</td>
39     <td><input type='text' name='product_price' value='{{ $productarray->product_price }}'> </td>
40   </tr>
41   <tr>
42     <td></td>
43     <td>
44       <input type='submit' value="Update">
45       <a href="{{ URL::route('product.index') }}>Back</a>
46     </td>
47   </tr>
48 </table>
49 </form>
50 @endsection
51
52
```

The code includes logic to handle form submission and provide a back link to the index page. The file is saved in the 'laravelcrud' workspace.



# Update Process in Controller



The screenshot shows a Visual Studio Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Debug, Terminal, Help.
- Title Bar:** ProductController.php - laravelcrud - Visual Studio Code.
- Explorer:** Shows the project structure under "LARAVELCRUD". The "resources" folder is currently selected.
- Code Editor:** Displays the PHP code for the update method in ProductController.php. The code performs validation on the request, retrieves an existing product from the database, updates its details, saves the changes, and then redirects to the product index page with a success message.
- Bottom Status Bar:** Shows the current line (Ln 101), column (Col 44), and other settings like spaces, encoding, and file type.

```
    * @param int $id
    * @return \Illuminate\Http\Response
   */
public function update(Request $request, $id)
{
    $request->validate([
        'product_name'=>'required',
        'product_details'=> 'required',
        'product_price' => 'required|integer'
    ]);

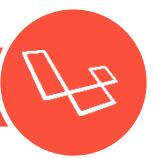
    $productarray = Product::where('product_id', $id)->first();

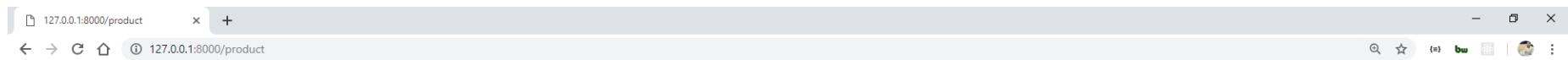
    $productarray->product_name      = $request->get('product_name');
    $productarray->product_details  = $request->get('product_details');
    $productarray->product_price    = $request->get('product_price');

    $productarray->save();

    return redirect('/product')->with('success', 'Product has been added');
}

/**
 * Remove the specified resource from storage.
 *
```





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

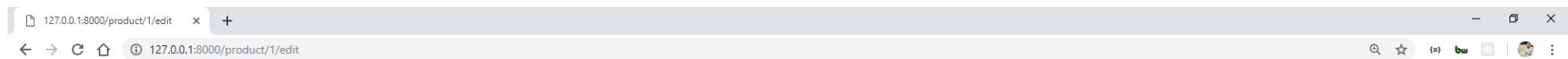
ID	Product Name	Product Price	Product Quantity	Action
1	iPhone XR	iPhone XR	85000	<a href="#">Edit</a> <a href="#">Delete</a>



Akash Technolabs

[www.akashsir.com](http://www.akashsir.com)





## Laravel CURD

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

### Edit Product

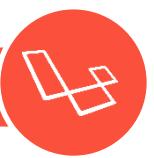
Name	<input type="text" value="iPhone XR"/>
Details	<input type="text" value="iPhone XR"/>
Price	<input type="text" value="85000"/>

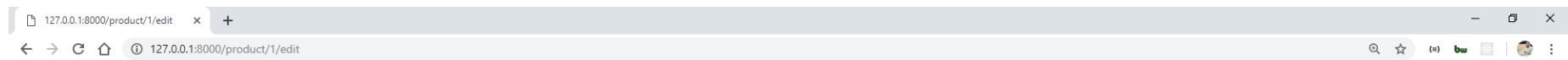
[Update](#) [Back](#)



Akash Technolabs

[www.akashsir.com](http://www.akashsir.com)





## Laravel CURD

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

### Edit Product

Name

Details

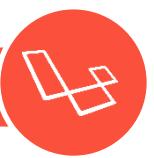
Price

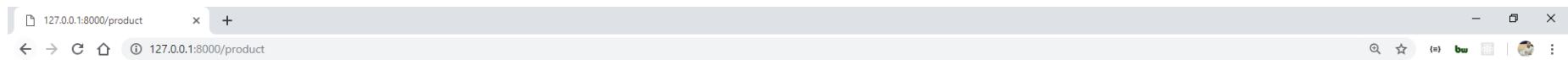
[Update](#) [Back](#)



Akash Technolabs

[www.akashsir.com](http://www.akashsir.com)

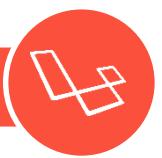




## Laravel CURD

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

ID	Product Name	Product Price	Product Quantity	Action
1	iPhone XR	iPhone XR New Model	95000	<a href="#">Edit</a> <a href="#">Delete</a>



■ Enjoy ☺

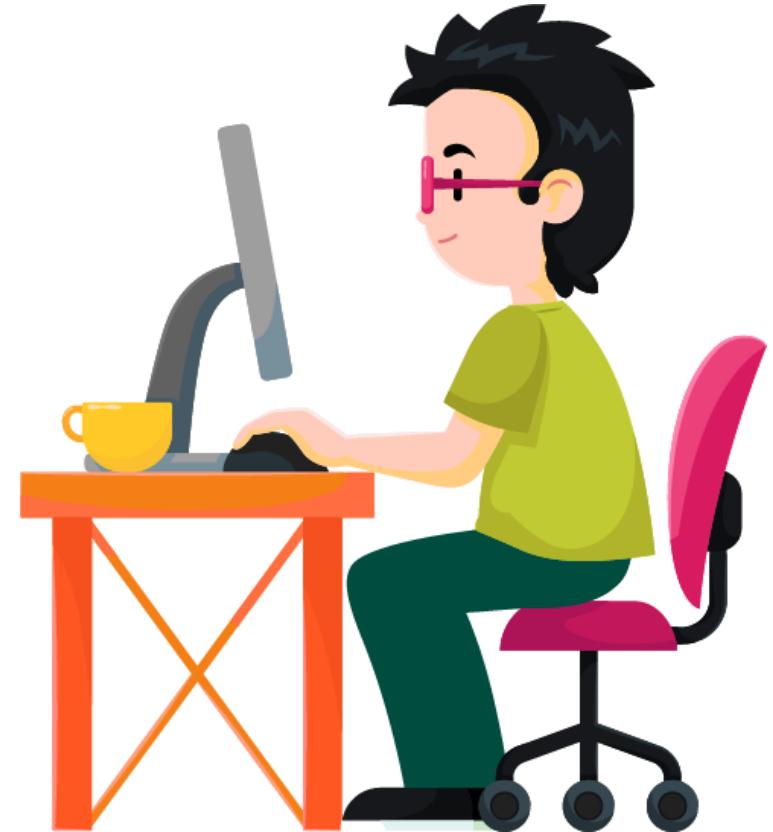


Akash Technolabs

[www.akashsir.com](http://www.akashsir.com)



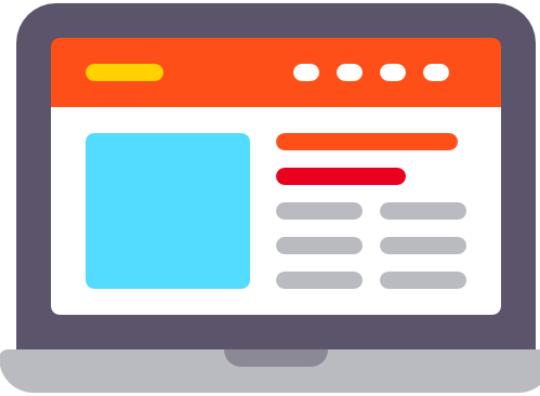
# Get Exclusive Video Tutorials



[www.aptutorials.com](http://www.aptutorials.com)

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





Get More Details

[www.akashsir.com](http://www.akashsir.com)



# If You Liked It !

## Rating Us Now



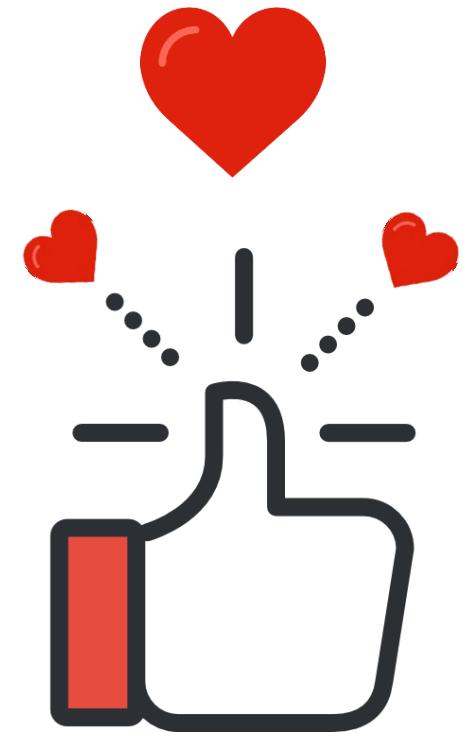
**Just Dial**

[https://www.justdial.com/Ahmedabad/Aakash-Technolabs-Navrangpura-Bus-Stop-Navrangpura/079PXX79-XX79-170615221520-S5C4\\_BZDET](https://www.justdial.com/Ahmedabad/Aakash-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](http://www.akashsir.com)

[www.akashtechnolabs.com](http://www.akashtechnolabs.com)

[www.akashpadhiyar.com](http://www.akashpadhiyar.com)

[www.aptutorials.com](http://www.aptutorials.com)

## # Social Info



Akash.padhiyar



Akashpadhiyar



Akash\_padhiyar



+91 99786-21654



#Akashpadhiyar

#aptutorials