



ក្រុងសាលា
SETEC INSTITUTE

HTmL

ooo

CONVERTING AN HTML TEMPLATE INTO LARAVEL PROJECT

Presented by TRY VORN



Ads

Laravel



Feedback

HTML Template

NiceAdmin

≡ 🔍

4 3  K. Anderson

- [Dashboard](#)
- [Components](#)
- [Forms](#)
- [Tables](#)
- [Charts](#)
- [Icons](#)

PAGES

- [Profile](#)
- [F.A.Q](#)
- [Contact](#)
- [Register](#)

Dashboard

Home / Dashboard

Sales | Today

145

12% increase

Revenue | This Month

\$3,264

8% increase

Customers | This Year

1244

12% decrease

Reports | Today

Recent Activity | Today

32 min • Quia quae rerum explicabo officiis beatae

56 min • Voluptatem blanditiis blanditiis eveniet

2 hrs • Voluptates corrupti molestias voluptatem

1 day • Tempore autem saepe occaecati voluptatem tempore

2 days • Est sit eum reiciendis exercitationem

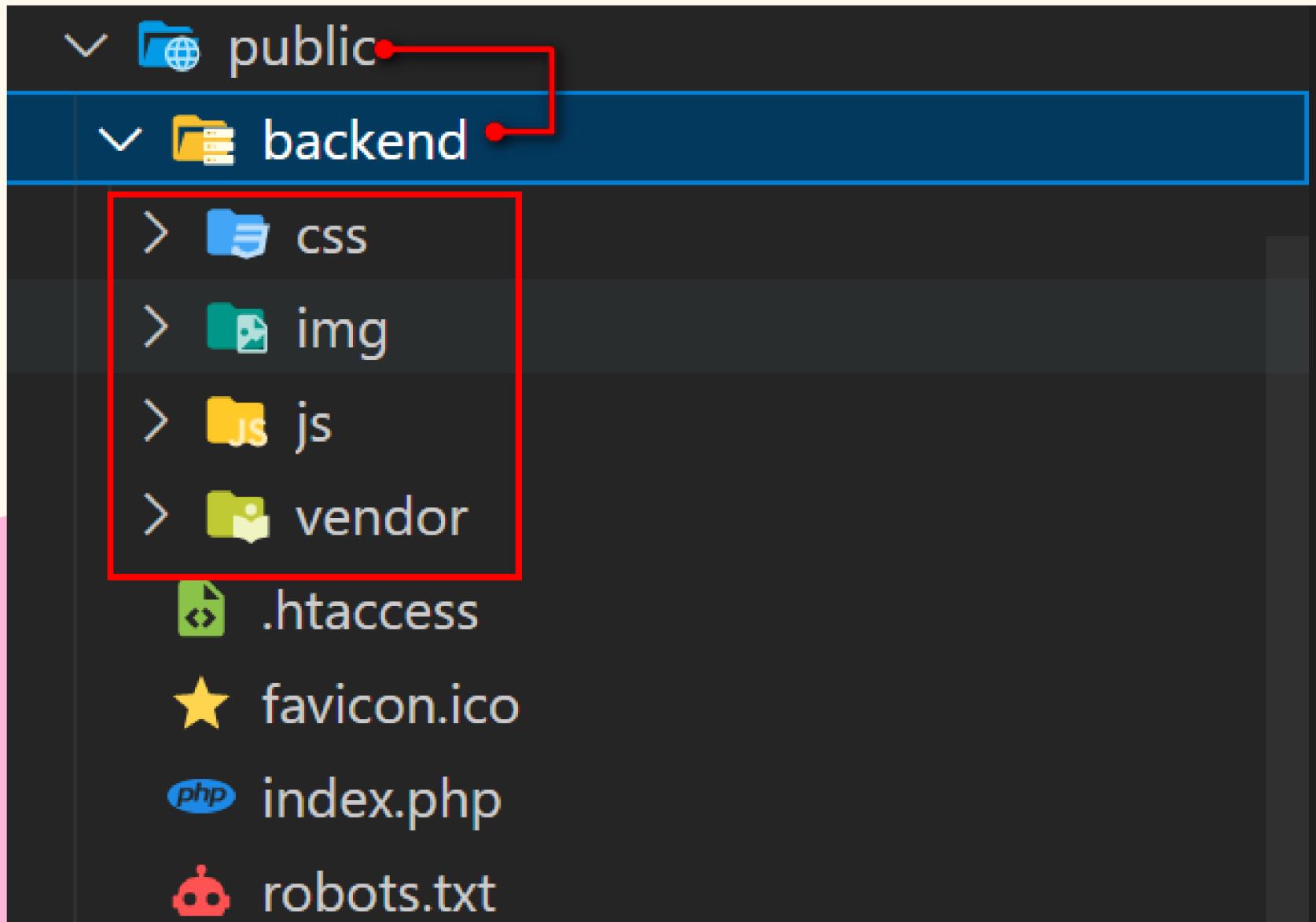
4 weeks • Dicta dolorem harum

HTML Theme conversion in Laravel

You have to follow some basic steps to convert your HTML theme to the Laravel framework

Step 1 – Make assets files to a public folder in Laravel

Basically, you need to copy all the assets files like CSS, Fonts, Images, Jquery files in the **public** folder in your Laravel.



Step 2 – Make a new Controller

The next step is you have to create a new controller in your **Controllers** file.

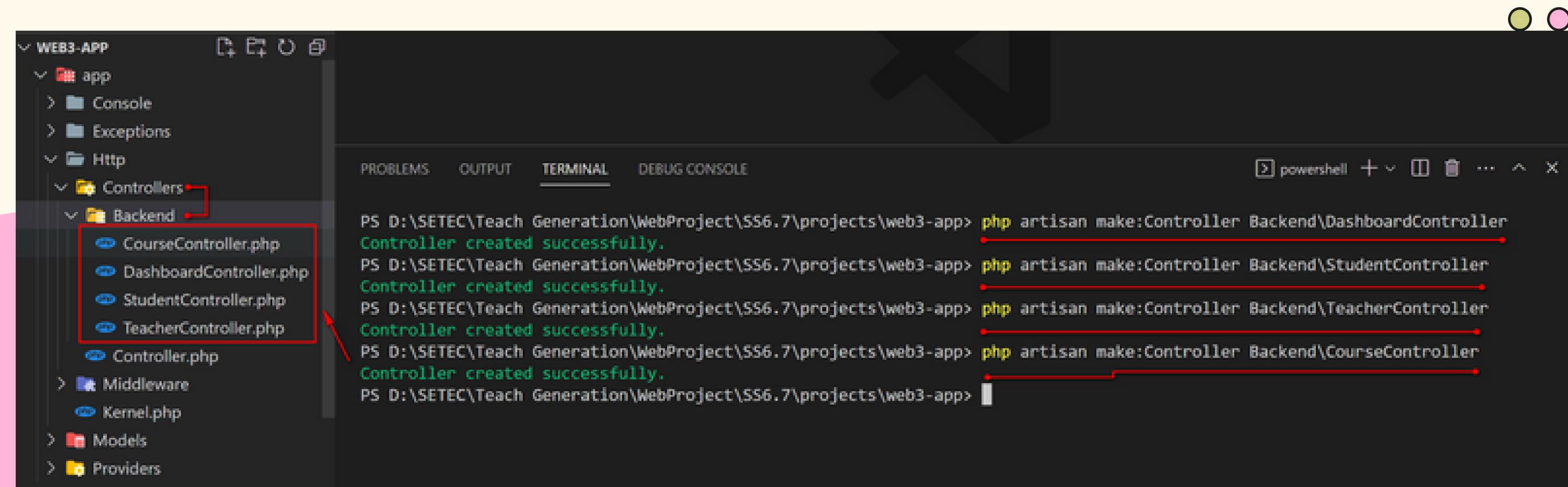
Eg:

php artisan make:Controller Backend\DashboardController

php artisan make:Controller Backend\StudentController

php artisan make:Controller Backend\TeacherController

php artisan make:Controller Backend\CourseController



A screenshot of a terminal window within a code editor interface. The terminal shows the following command and output:
PS D:\SETEC\Teach Generation\WebProject\SS6.7\projects\web3-app> php artisan make:Controller Backend\DashboardController
Controller created successfully.
PS D:\SETEC\Teach Generation\WebProject\SS6.7\projects\web3-app> php artisan make:Controller Backend\StudentController
Controller created successfully.
PS D:\SETEC\Teach Generation\WebProject\SS6.7\projects\web3-app> php artisan make:Controller Backend\TeacherController
Controller created successfully.
PS D:\SETEC\Teach Generation\WebProject\SS6.7\projects\web3-app> php artisan make:Controller Backend\CourseController
Controller created successfully.
PS D:\SETEC\Teach Generation\WebProject\SS6.7\projects\web3-app>

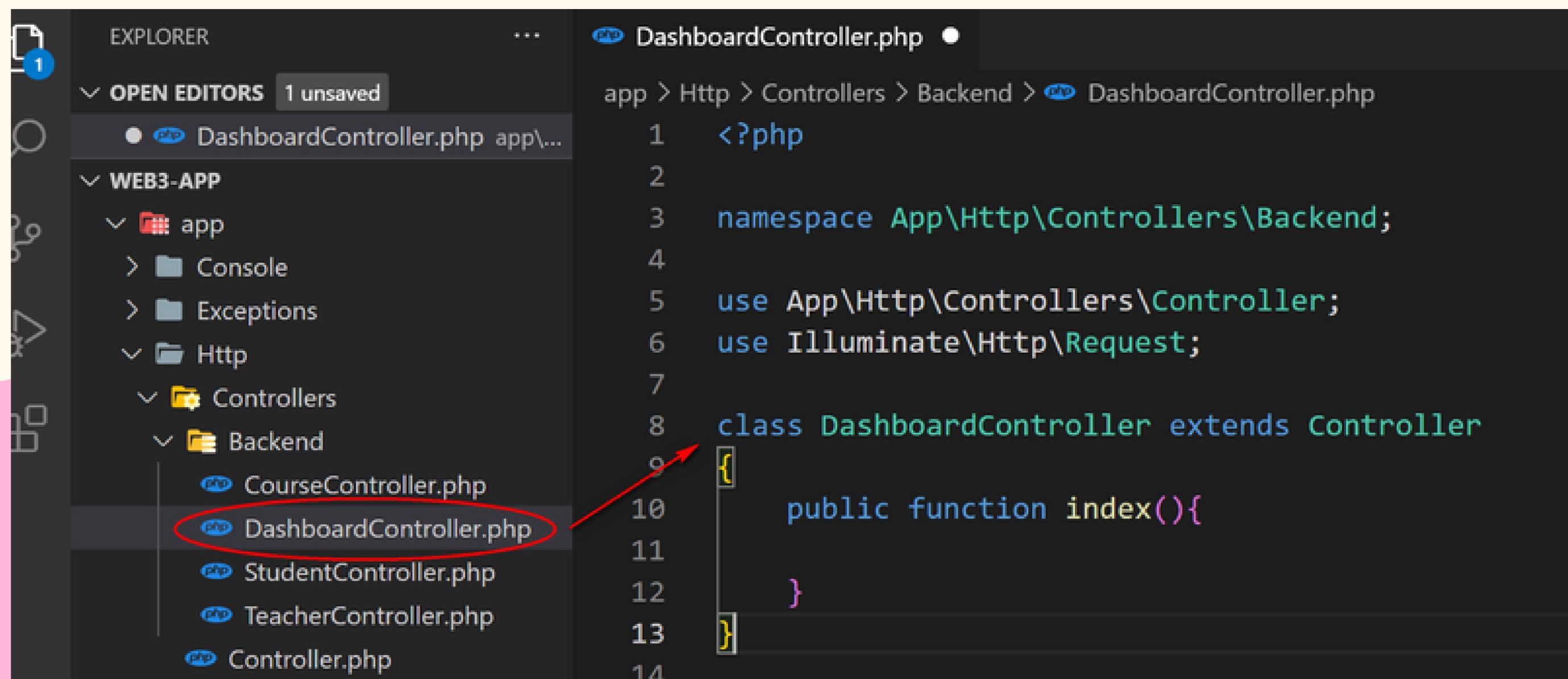
The left sidebar of the code editor shows a project structure with a red box highlighting the **Backend\Controllers** folder. Inside this folder, four files are listed: **CourseController.php**, **DashboardController.php**, **StudentController.php**, and **TeacherController.php**. A red arrow points from the bottom of the terminal window towards these files.



Step 3 – Make all required methods with controllers

Once you create the **controller** the next step would be to create the required methods in that controller class.

Example Dashboard, Student, Teacher, Course



The screenshot shows a code editor interface with the following details:

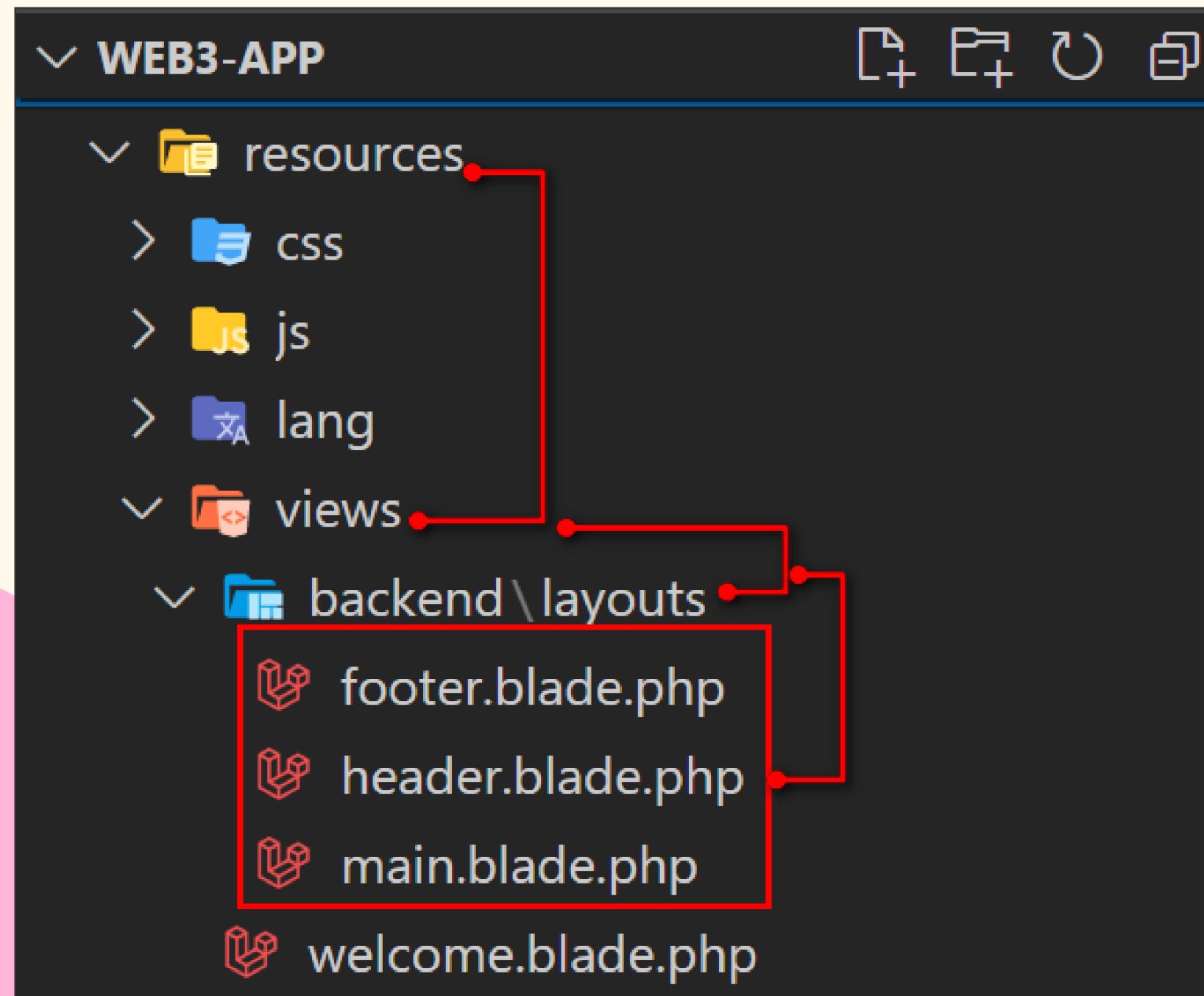
- EXPLORER** panel on the left: Shows the project structure under **WEB3-APP**, specifically the **Http** > **Controllers** > **Backend** folder. It lists several files: **CourseController.php**, **DashboardController.php** (which is highlighted with a red oval), **StudentController.php**, **TeacherController.php**, and **Controller.php**.
- DashboardController.php** file open in the main editor area:

```
1  <?php
2
3  namespace App\Http\Controllers\Backend;
4
5  use App\Http\Controllers\Controller;
6  use Illuminate\Http\Request;
7
8  class DashboardController extends Controller
9  {
10     public function index(){
11
12     }
13 }
```



Step 4 – Views Folder Structure

Create the folder with any name in your layout let's say HTML and create the different **views** files in that folder.



Step 5 – Make a fixed layout for the website

Once, all the methods defined in controller class the next step would be to fix the layout of your page.

Yield is a function used in Laravel to define a particular section in layout and it is permanently used to retrieve content from the child page and transfer it into the master page.

The diagram illustrates the Laravel file structure and the corresponding code for creating a fixed layout. It shows two main windows: a file browser and a code editor.

File Browser: Shows the project structure under 'WEB3-APP'.

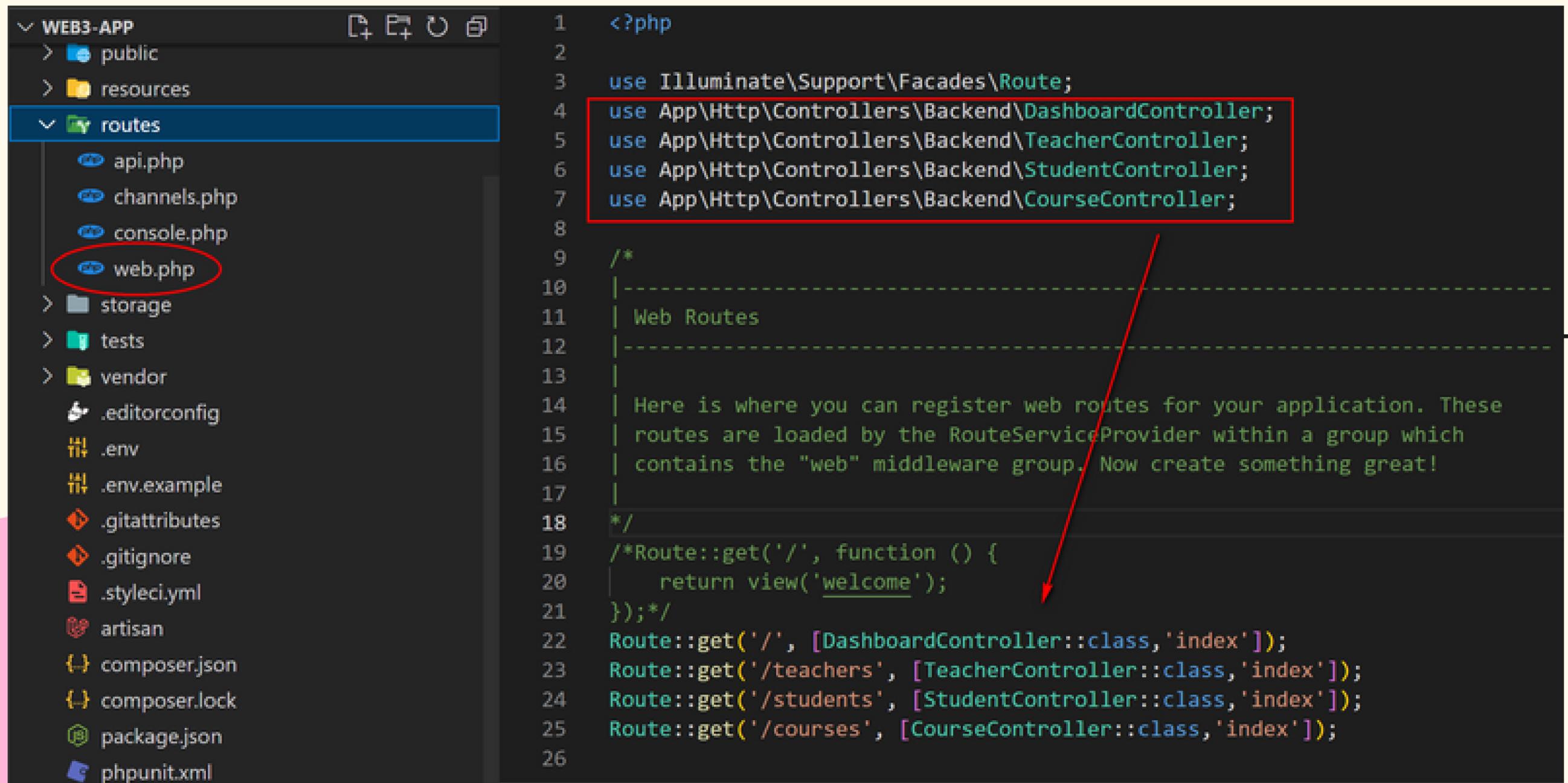
- resources
- css
- js
- lang
- views
- backend
- courses
- layouts
- footer.blade.php
- header.blade.php
- leftsidebar.blade.php
- main.blade.php** (selected)
- students
- teachers
- index.blade.php
- welcome.blade.php

Code Editor (Left): Displays the content of **main.blade.php**.1 @include('backend.layouts.header')
2 @include('backend.layouts.leftsidebar')
3 @yield('main-container')
4 @include('backend.layouts.footer')
5A red arrow points from the 'main.blade.php' entry in the file browser to the line number 5 in the code editor.

Code Editor (Right): Displays the content of a child view, likely 'index.blade.php'.1 @extends('backend.layouts.main')
2 @section("title","Teacher")
3 @section('main-container')
4
5
6 <main id="main" class="main">
7
8 > <div class="pagetitle">...
17 </div><!-- End Page Title -->
18
19 > <section class="section">...
81 </section>
82
83 </main><!-- End #main -->
84
85 @endsectionRed boxes highlight the sections and extends statements, and red arrows point from the file browser entries to these highlighted sections.

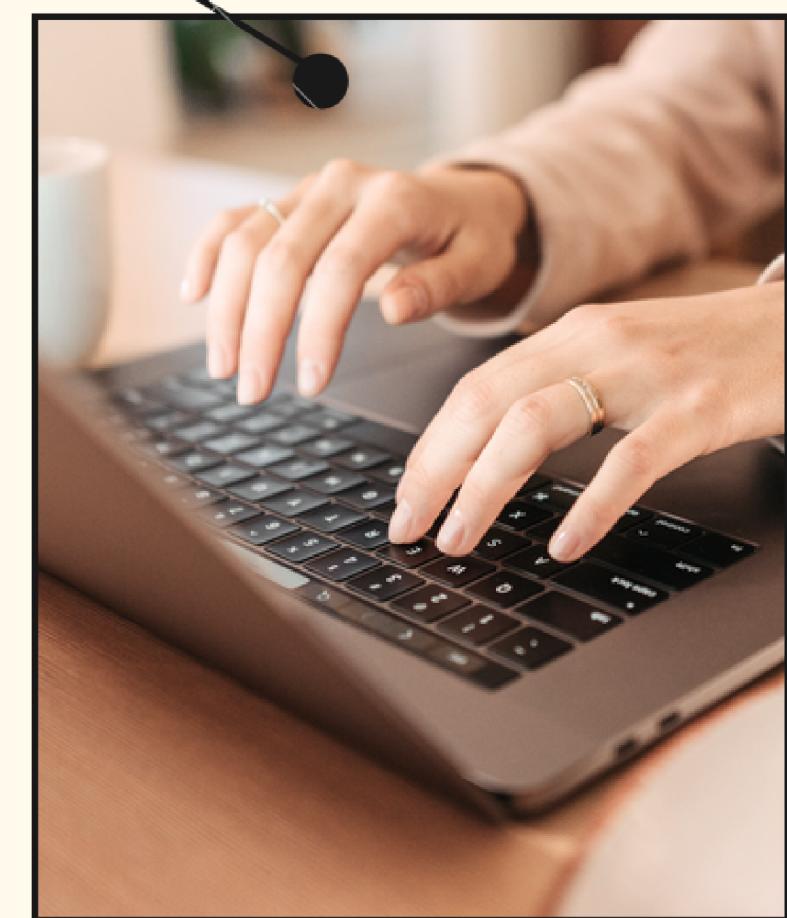
Step – 6 Configure Routes file

The last step is to configure your route file



```
WEB3-APP
public
resources
routes
api.php
channels.php
console.php
web.php
storage
tests
vendor
.editorconfig
.env
.env.example
.gitattributes
.gitignore
.styleci.yml
artisan
composer.json
composer.lock
package.json
phpunit.xml

1 <?php
2
3 use Illuminate\Support\Facades\Route;
4 use App\Http\Controllers\Backend\DashboardController;
5 use App\Http\Controllers\Backend\TeacherController;
6 use App\Http\Controllers\Backend\StudentController;
7 use App\Http\Controllers\Backend\CourseController;
8
9 /*
10 |-----
11 | Web Routes
12 |-----
13 |
14 | Here is where you can register web routes for your application. These
15 | routes are loaded by the RouteServiceProvider within a group which
16 | contains the "web" middleware group. Now create something great!
17 |
18 */
19 Route::get('/', function () {
20     return view('welcome');
21 });*/
22 Route::get('/', [DashboardController::class, 'index']);
23 Route::get('/teachers', [TeacherController::class, 'index']);
24 Route::get('/students', [StudentController::class, 'index']);
25 Route::get('/courses', [CourseController::class, 'index']);
```



Laravel Template

The screenshot displays a Laravel template for a management application. The URL in the browser bar is 127.0.0.1:8000. The page title is "NiceAdmin". The left sidebar contains a navigation menu with the following items:

- Dashboard (selected, highlighted with a red box)
- Manage Teacher
- Manage Student
- Manage Course
- Components
- Forms
- Tables
- Charts
- Icons

PAGES is listed at the bottom of the sidebar.

The main content area is titled "Dashboard" and shows three key metrics:

- Sales | Today**: 145 (12% increase)
- Revenue | This Month**: \$3,264 (8% increase)
- Customers | This Year**: 1244 (12% decrease)

A "Recent Activity" section lists five entries:

- 32 min ago: Quia quae rerum explicabo officiis beatae
- 56 min ago: Voluptatem blanditiis blanditiis eveniet
- 2 hrs ago: Voluptates corrupti molestias voluptatem
- 1 day ago: Tempore autem saepe occaecati voluptatem tempore
- 2 days ago: Est sit eum reiciendis exercitationem

The top right corner shows a notification badge with 4 notifications, a message badge with 3 messages, and a user profile for K. Anderson.