php
Laravel

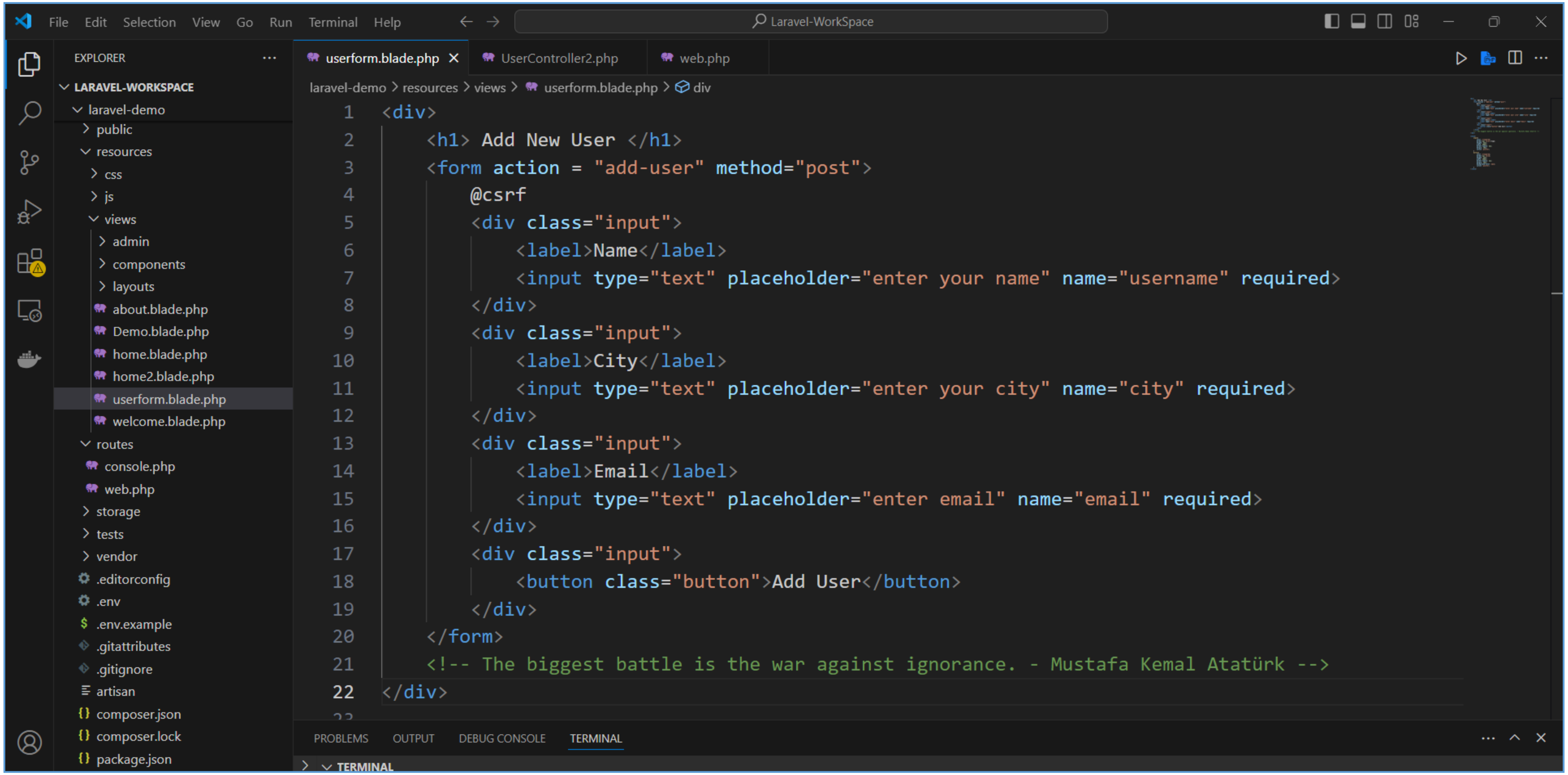
Outlines of Discussions

- Form handling in laravel
- Form Validation
- Named Route in Laravel
- Route Grouping

Form handling in laravel

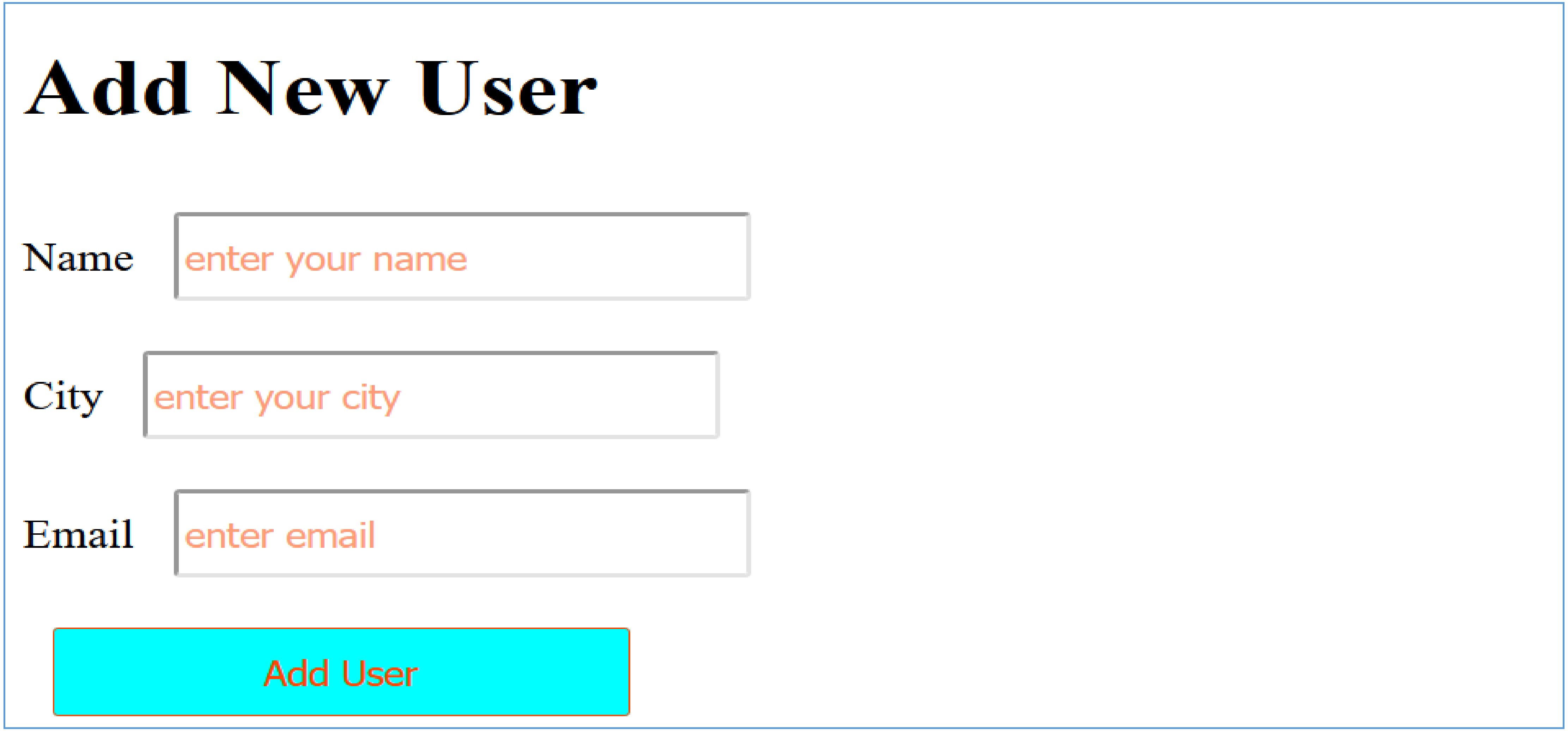
- In Laravel, forms and input fields can be created using **standard HTML** or **Laravel's Blade** syntax.
- Laravel doesn’t have its own syntax for creating forms, but it does offer various **tools** and **helper** methods to make handling forms more manageable.

Cont. ...



```
<div>
    <h1> Add New User </h1>
    <form action = "add-user" method="post">
        @csrf
        <div class="input">
            <label>Name</label>
            <input type="text" placeholder="enter your name" name="username" required>
        </div>
        <div class="input">
            <label>City</label>
            <input type="text" placeholder="enter your city" name="city" required>
        </div>
        <div class="input">
            <label>Email</label>
            <input type="text" placeholder="enter email" name="email" required>
        </div>
        <div class="input">
            <button class="button">Add User</button>
        </div>
    </form>
    <!-- The biggest battle is the war against ignorance. - Mustafa Kemal Atatürk -->
</div>
```

Cont. ...



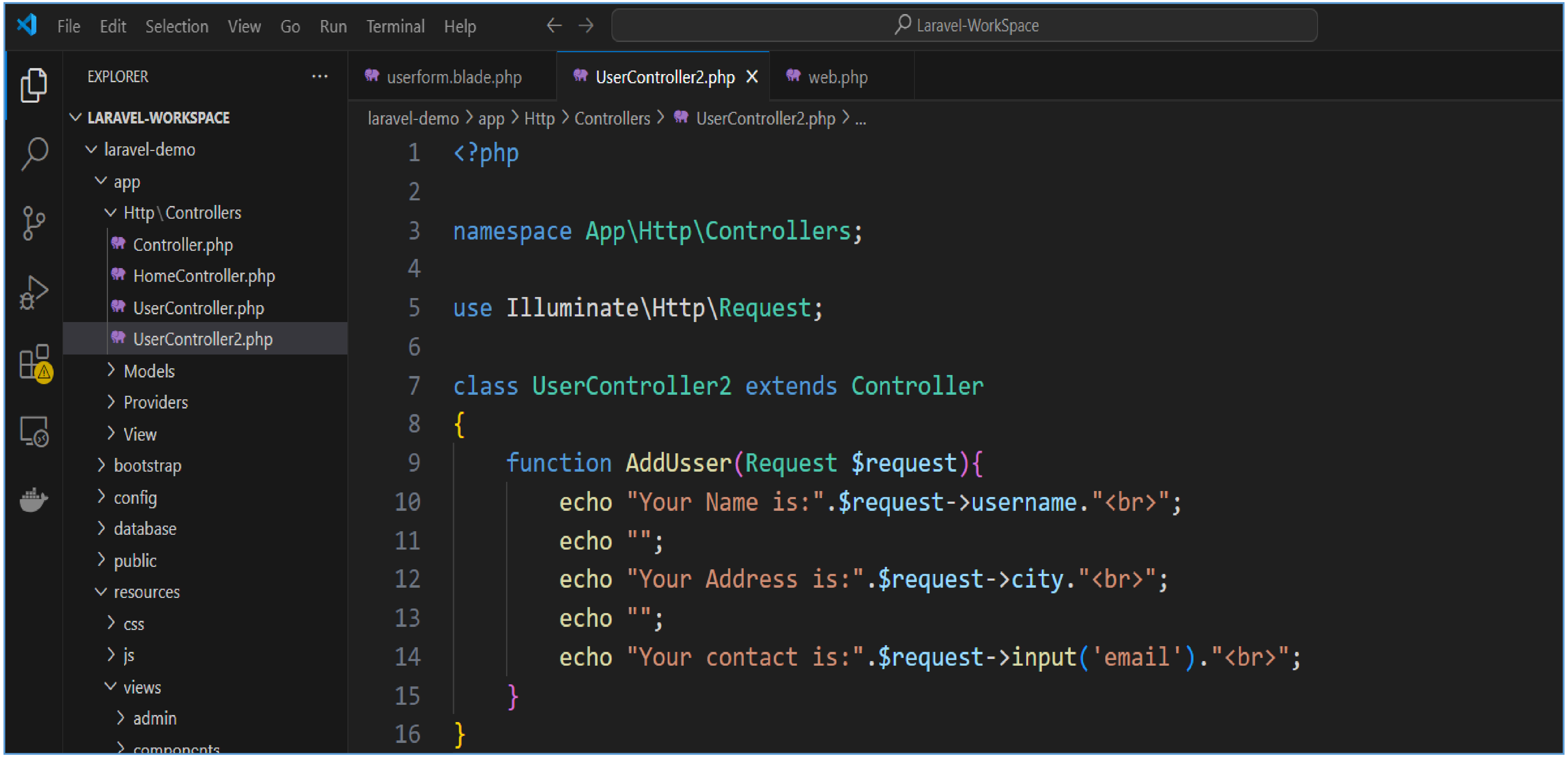
Add New User

Name

City

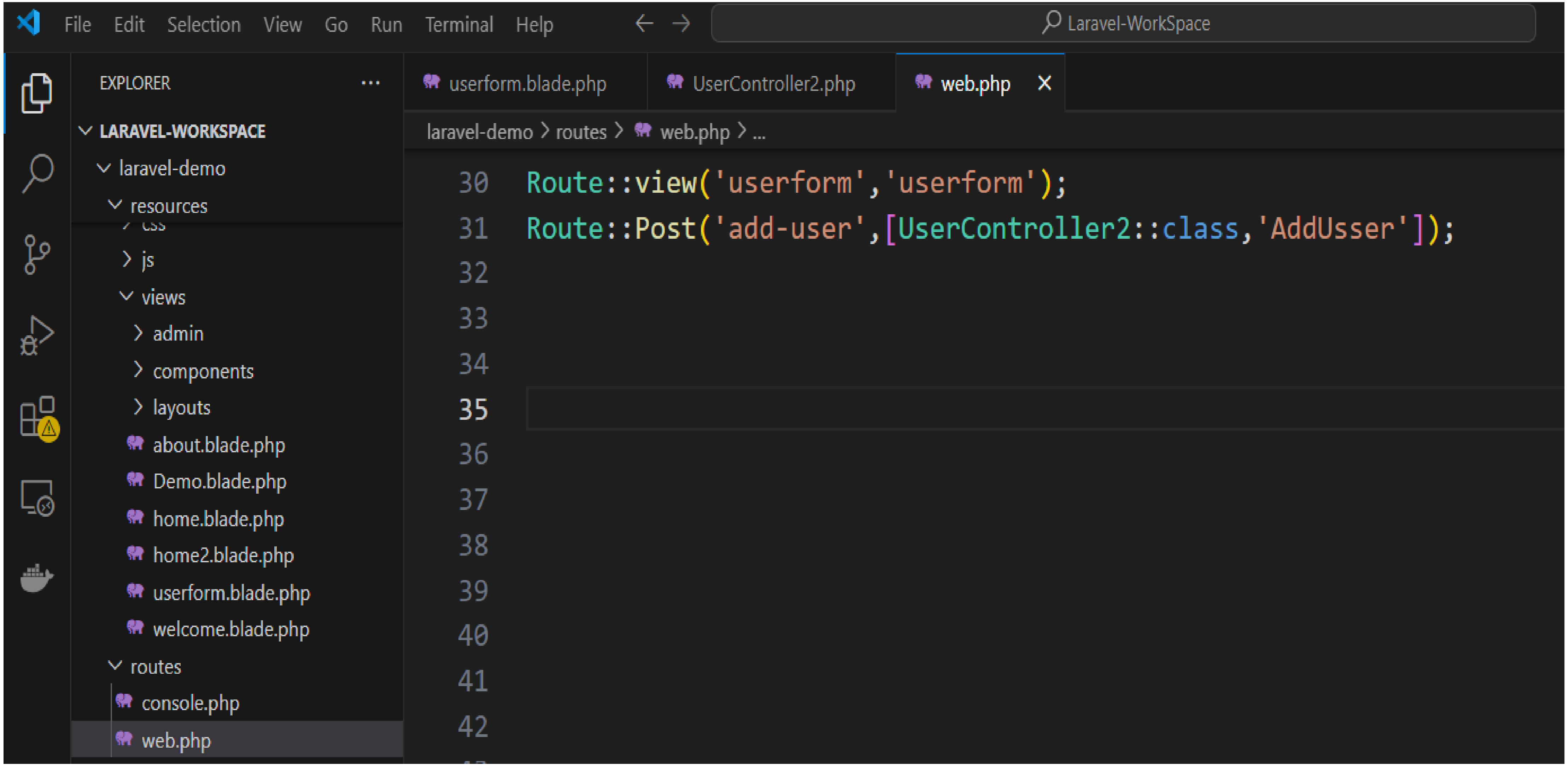
Email

Cont. …



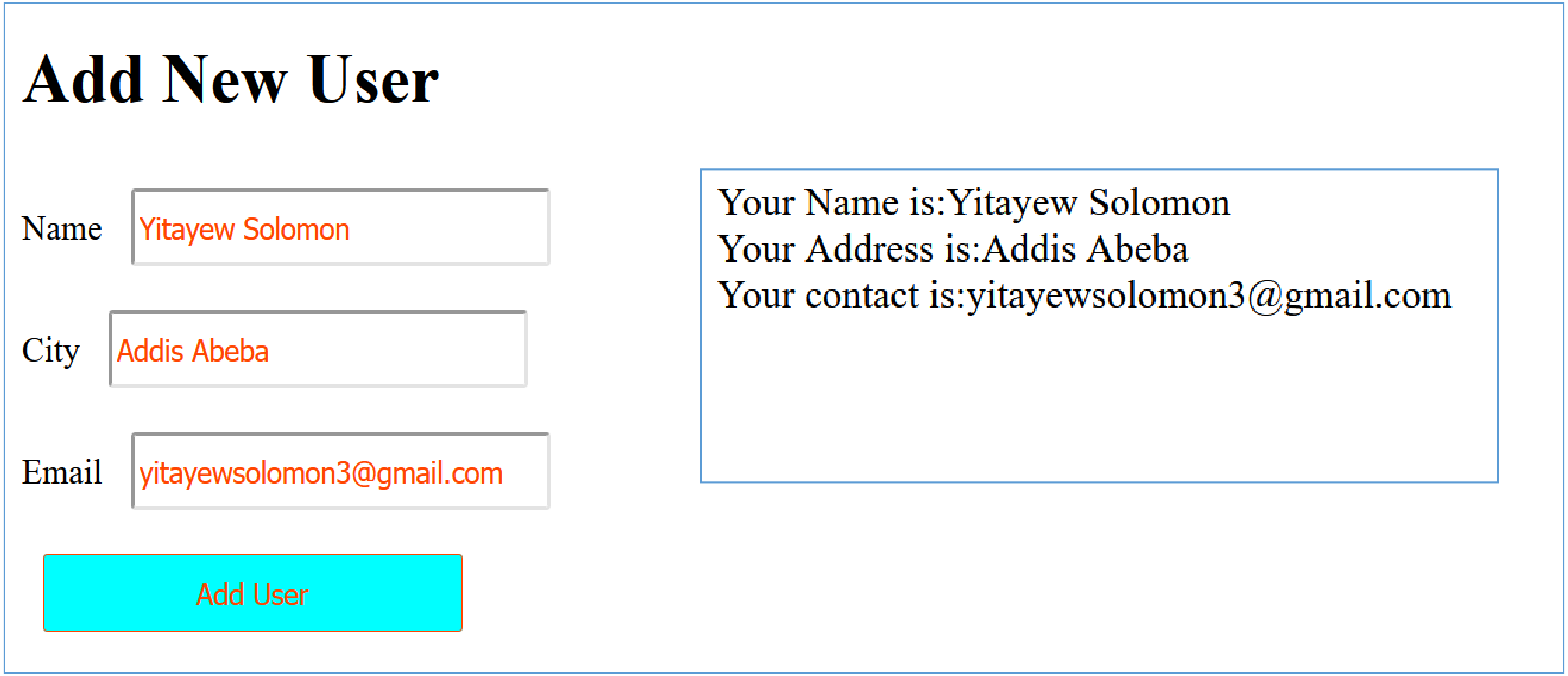
```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class UserController2 extends Controller
{
    function AddUsser(Request $request){
        echo "Your Name is:".$request->username."<br>";
        echo "";
        echo "Your Address is:".$request->city."<br>";
        echo "";
        echo "Your contact is:".$request->input('email')."<br>";
    }
}
```

Cont. …



```php
Route::view('userform','userform');
Route::Post('add-user',[UserController2::class,'AddUsser']);
```

Cont. ...



Add New User

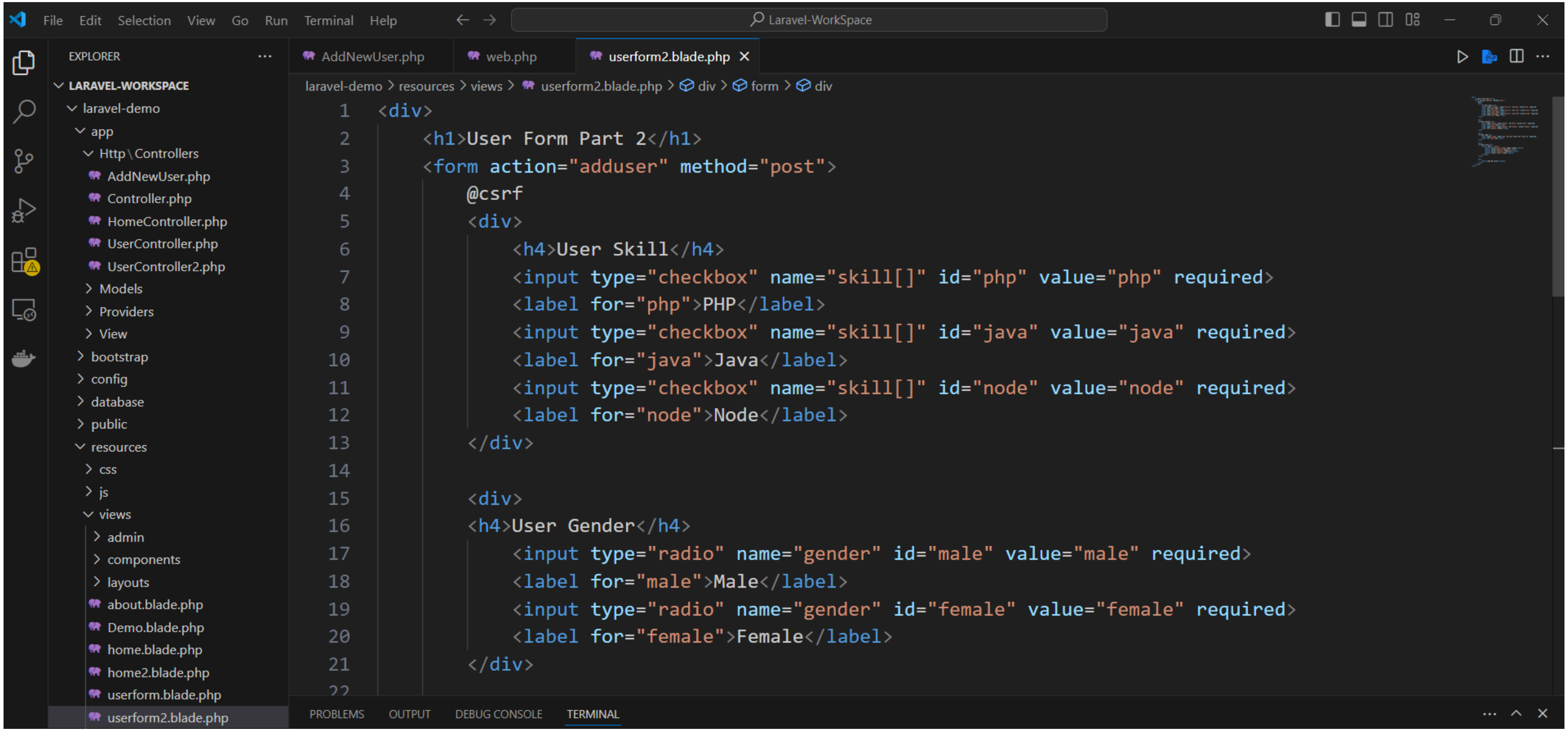
Name: Yitayew Solomon

City: Addis Abeba

Email: yitayewsolomon3@gmail.com

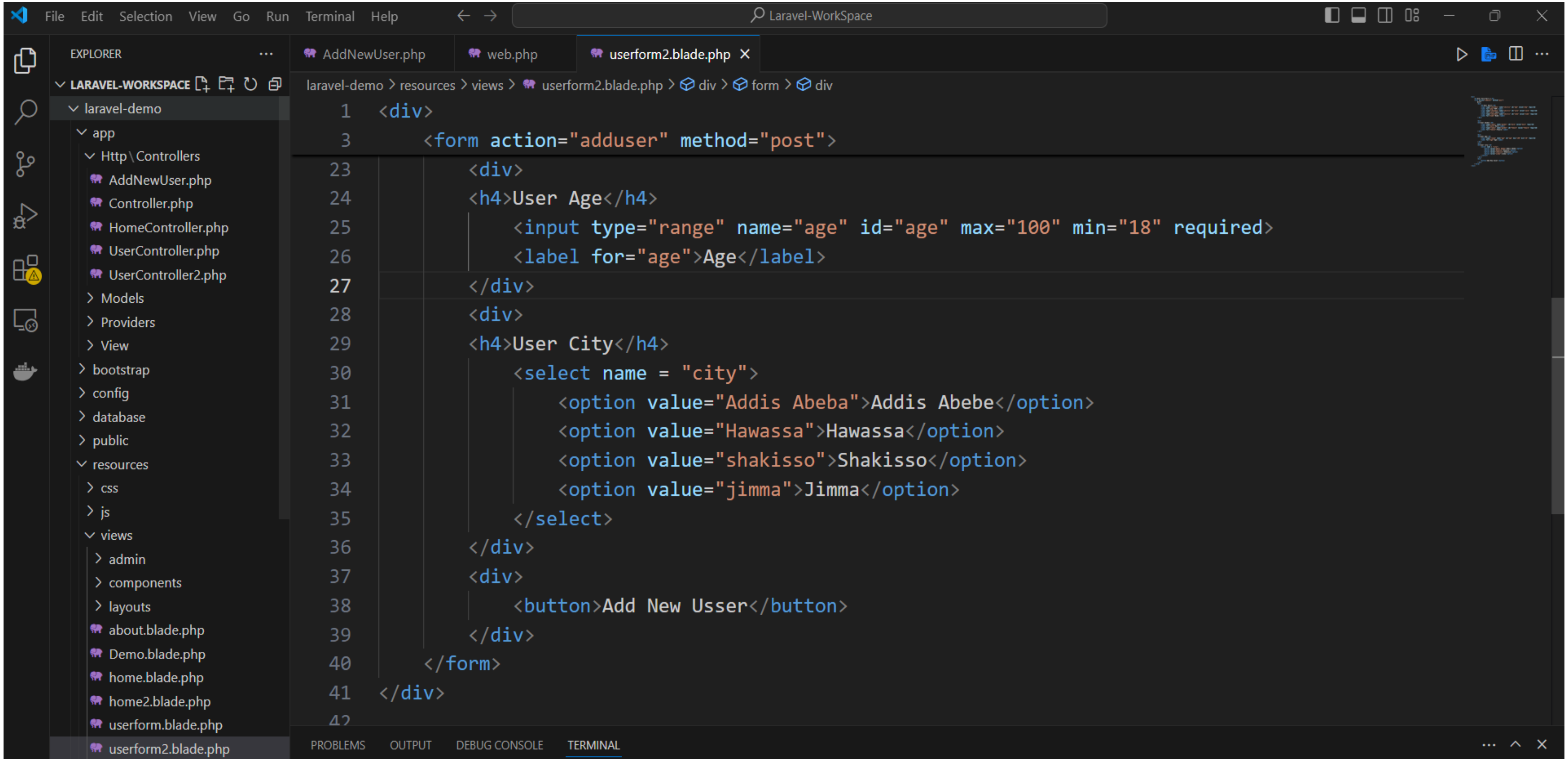
```
Your Name is:Yitayew Solomon
Your Address is:Addis Abeba
Your contact is:yitayewsolomon3@gmail.com
```

More on Forms



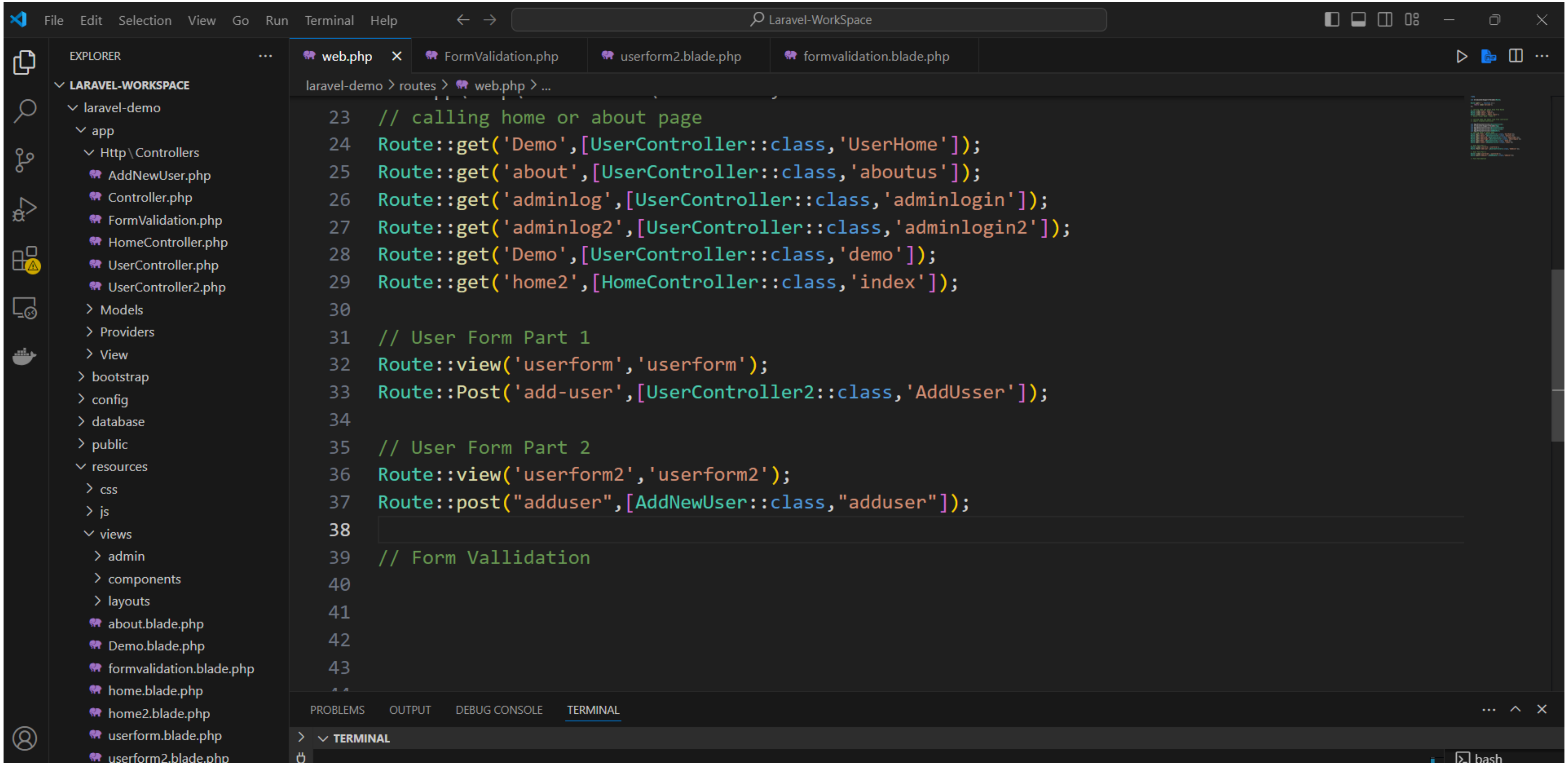
```
<div>
    <h1>User Form Part 2</h1>
    <form action="adduser" method="post">
        @csrf
        <div>
            <h4>User Skill</h4>
            <input type="checkbox" name="skill[]" id="php" value="php" required>
            <label for="php">PHP</label>
            <input type="checkbox" name="skill[]" id="java" value="java" required>
            <label for="java">Java</label>
            <input type="checkbox" name="skill[]" id="node" value="node" required>
            <label for="node">Node</label>
        </div>

        <div>
        <h4>User Gender</h4>
            <input type="radio" name="gender" id="male" value="male" required>
            <label for="male">Male</label>
            <input type="radio" name="gender" id="female" value="female" required>
            <label for="female">Female</label>
        </div>
```

Cont. ...



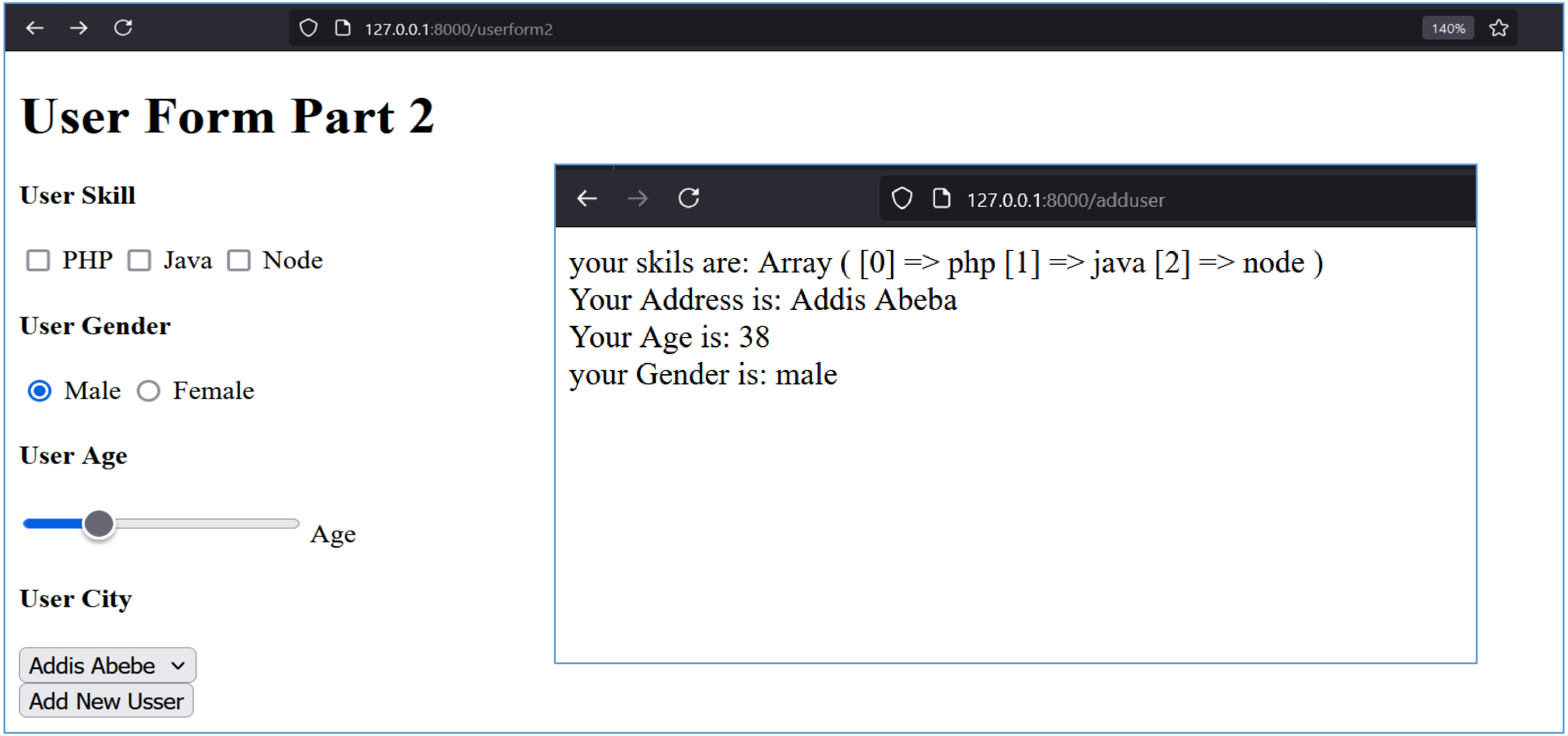
```
<div>
    <form action="adduser" method="post">
        <div>
        <h4>User Age</h4>
            <input type="range" name="age" id="age" max="100" min="18" required>
            <label for="age">Age</label>
        </div>
        <div>
        <h4>User City</h4>
            <select name = "city">
                <option value="Addis Abeba">Addis Abebe</option>
                <option value="Hawassa">Hawassa</option>
                <option value="shakisso">Shakisso</option>
                <option value="jimma">Jimma</option>
            </select>
        </div>
        <div>
            <button>Add New Usser</button>
        </div>
    </form>
</div>
```

Cont. ...



```php
// calling home or about page
Route::get('Demo',[UserController::class,'UserHome']);
Route::get('about',[UserController::class,'aboutus']);
Route::get('adminlog',[UserController::class,'adminlogin']);
Route::get('adminlog2',[UserController::class,'adminlogin2']);
Route::get('Demo',[UserController::class,'demo']);
Route::get('home2',[HomeController::class,'index']);

// User Form Part 1
Route::view('userform','userform');
Route::Post('add-user',[UserController2::class,'AddUsser']);

// User Form Part 2
Route::view('userform2','userform2');
Route::post("adduser",[AddNewUser::class,"adduser"]);

// Form Vallidation
```

Cont. ...



User Form Part 2

User Skill

□ PHP □ Java □ Node

User Gender

● Male ○ Female

User Age

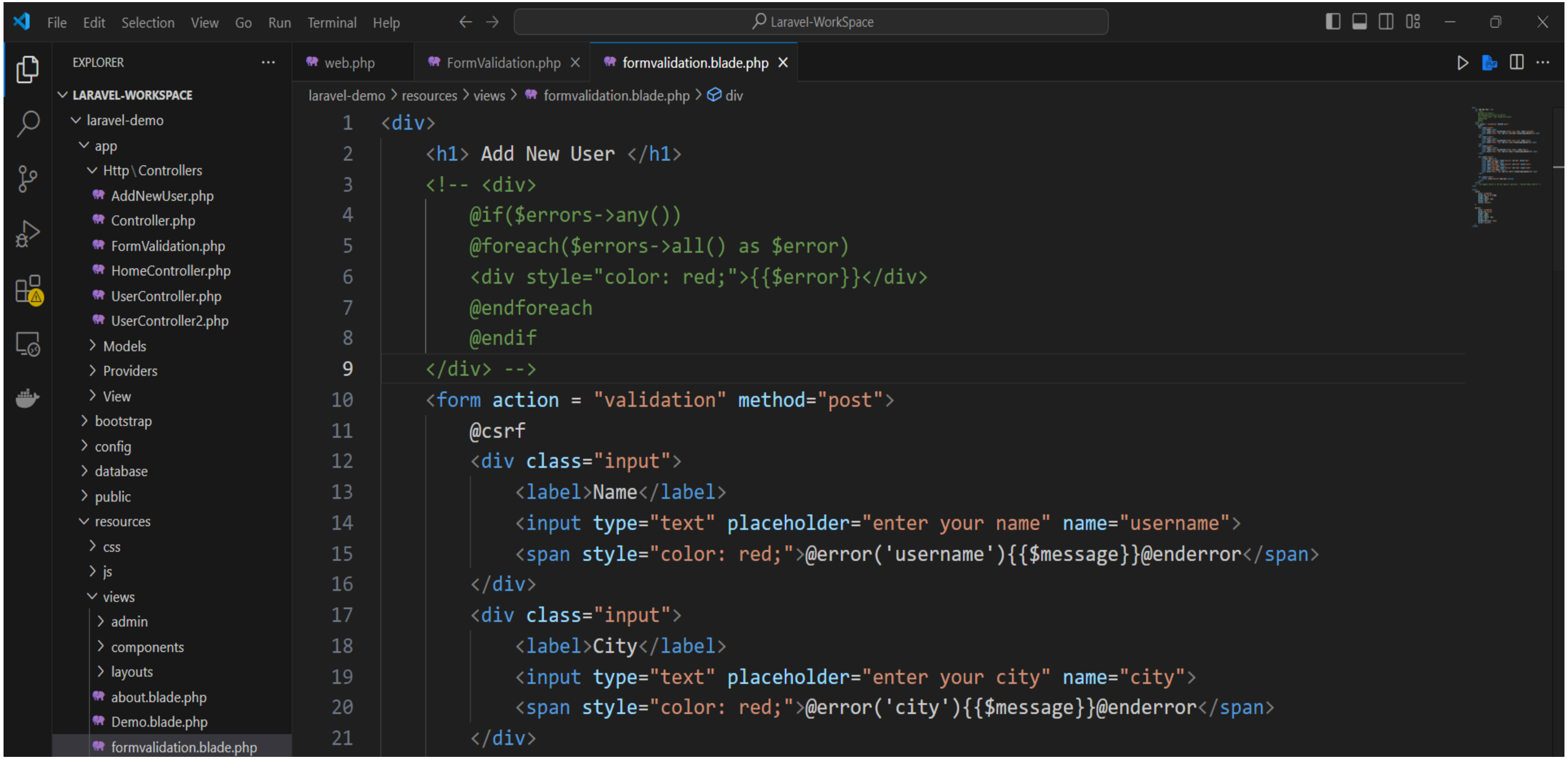
Age

User City

Addis Abebe

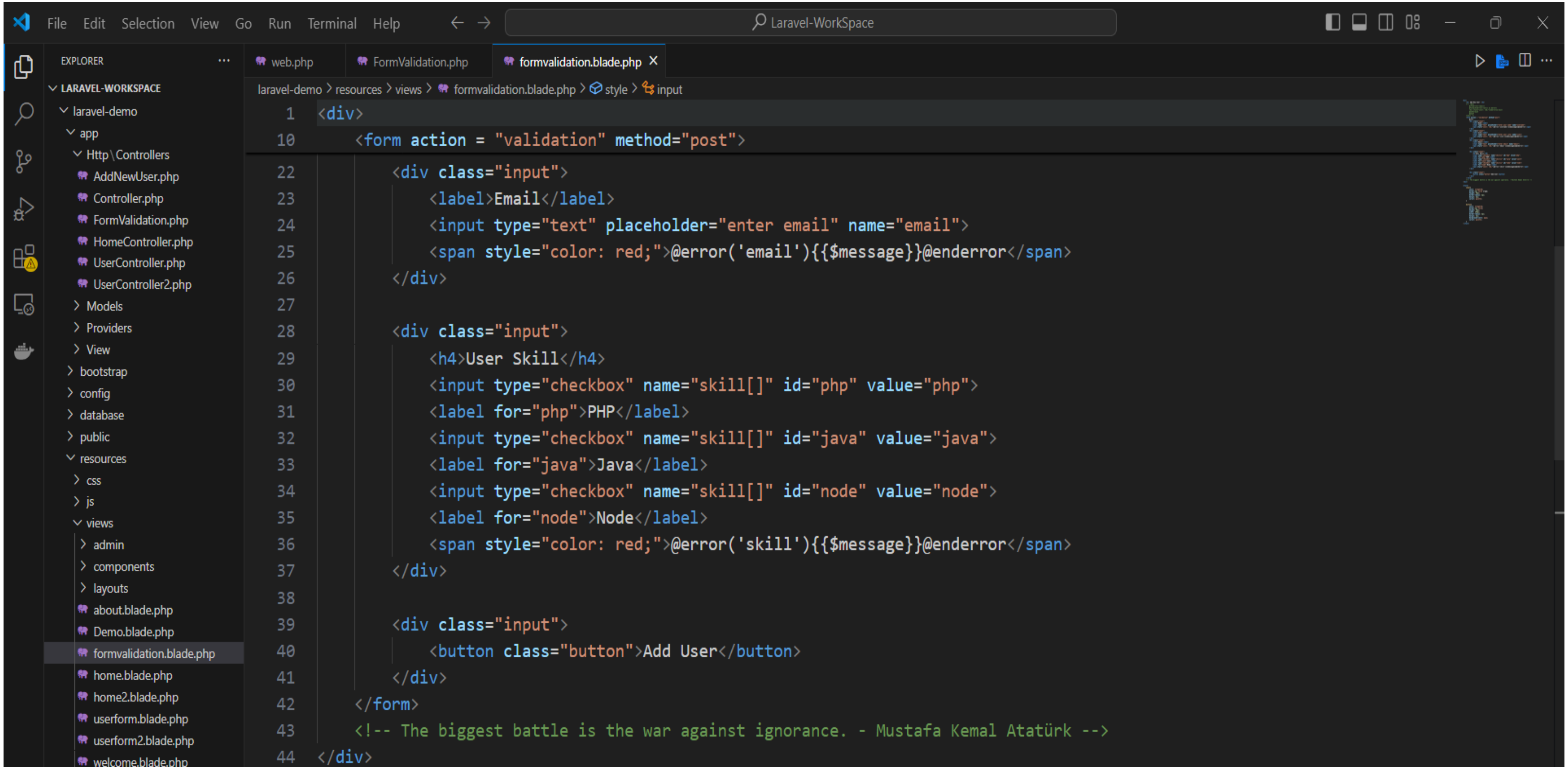
```
your skils are: Array ( [0] => php [1] => java [2] => node )
Your Address is: Addis Abeba
Your Age is: 38
your Gender is: male
```

Form Validation



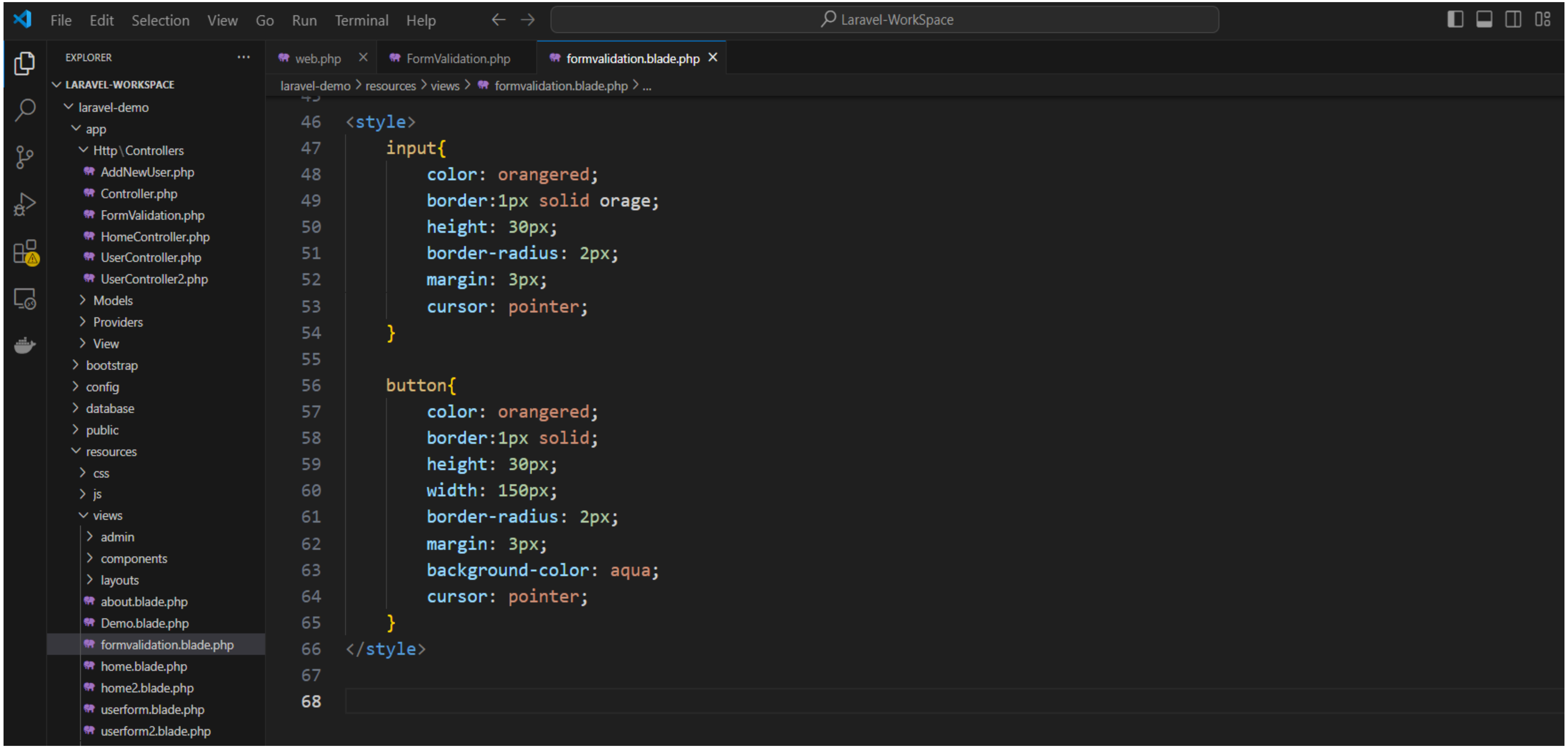
```
<div>
    <h1> Add New User </h1>
    <!-- <div>
        @if($errors->any())
        @foreach($errors->all() as $error)
        <div style="color: red;">{{$error}}</div>
        @endforeach
        @endif
    </div> -->
    <form action = "validation" method="post">
        @csrf
        <div class="input">
            <label>Name</label>
            <input type="text" placeholder="enter your name" name="username">
            <span style="color: red;">@error('username'){{$message}}@enderror</span>
        </div>
        <div class="input">
            <label>City</label>
            <input type="text" placeholder="enter your city" name="city">
            <span style="color: red;">@error('city'){{$message}}@enderror</span>
        </div>
```

Cont. …



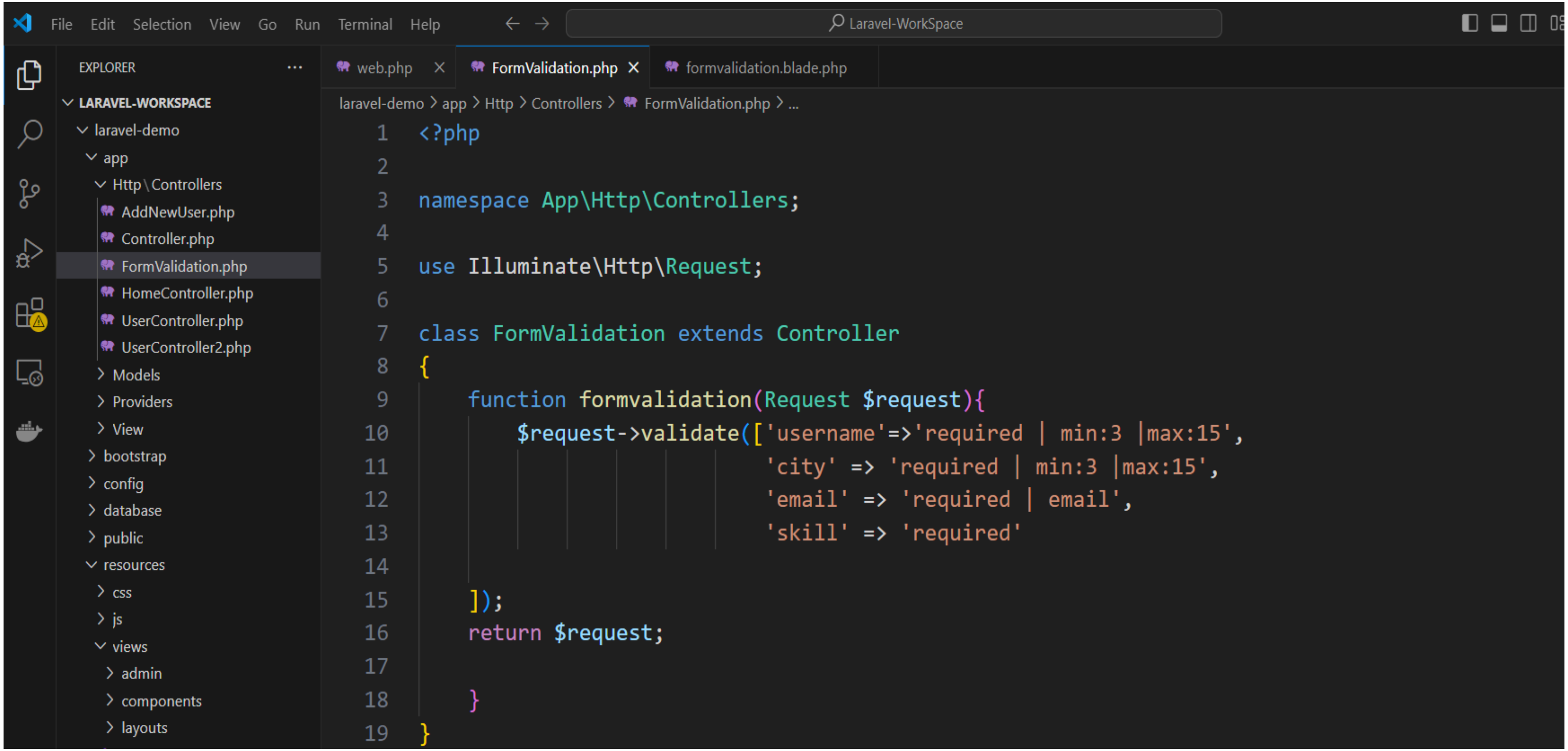
```
<div>
    <form action = "validation" method="post">
        <div class="input">
            <label>Email</label>
            <input type="text" placeholder="enter email" name="email">
            <span style="color: red;">@error('email'){{$message}}@enderror</span>
        </div>

        <div class="input">
            <h4>User Skill</h4>
            <input type="checkbox" name="skill[]" id="php" value="php">
            <label for="php">PHP</label>
            <input type="checkbox" name="skill[]" id="java" value="java">
            <label for="java">Java</label>
            <input type="checkbox" name="skill[]" id="node" value="node">
            <label for="node">Node</label>
            <span style="color: red;">@error('skill'){{$message}}@enderror</span>
        </div>

        <div class="input">
            <button class="button">Add User</button>
        </div>
    </form>
    <!-- The biggest battle is the war against ignorance. - Mustafa Kemal Atatürk -->
</div>
```

Cont. ...



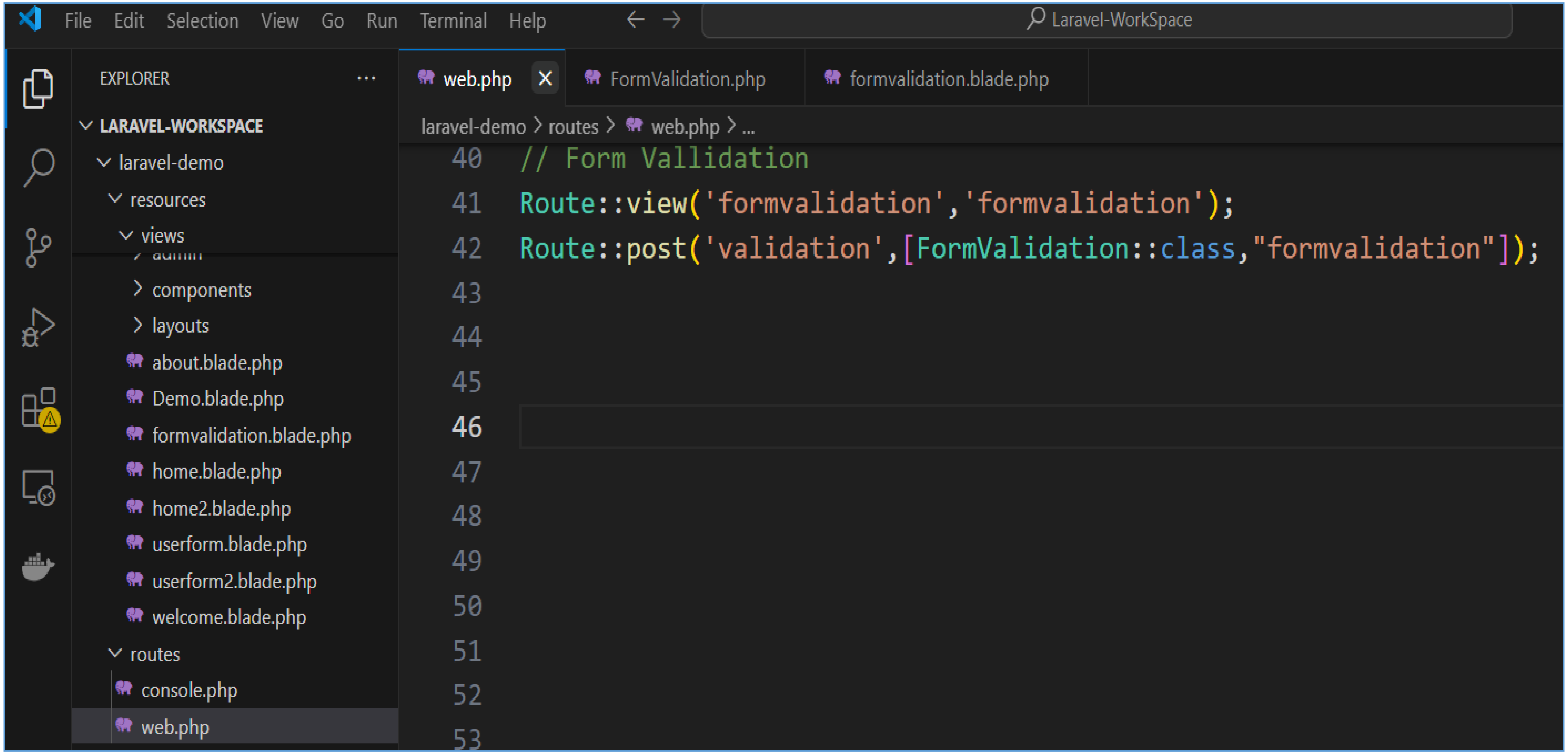
```
<style>
    input{
        color: orangered;
        border:1px solid orage;
        height: 30px;
        border-radius: 2px;
        margin: 3px;
        cursor: pointer;
    }

    button{
        color: orangered;
        border:1px solid;
        height: 30px;
        width: 150px;
        border-radius: 2px;
        margin: 3px;
        background-color: aqua;
        cursor: pointer;
    }
</style>

```

Cont. ...



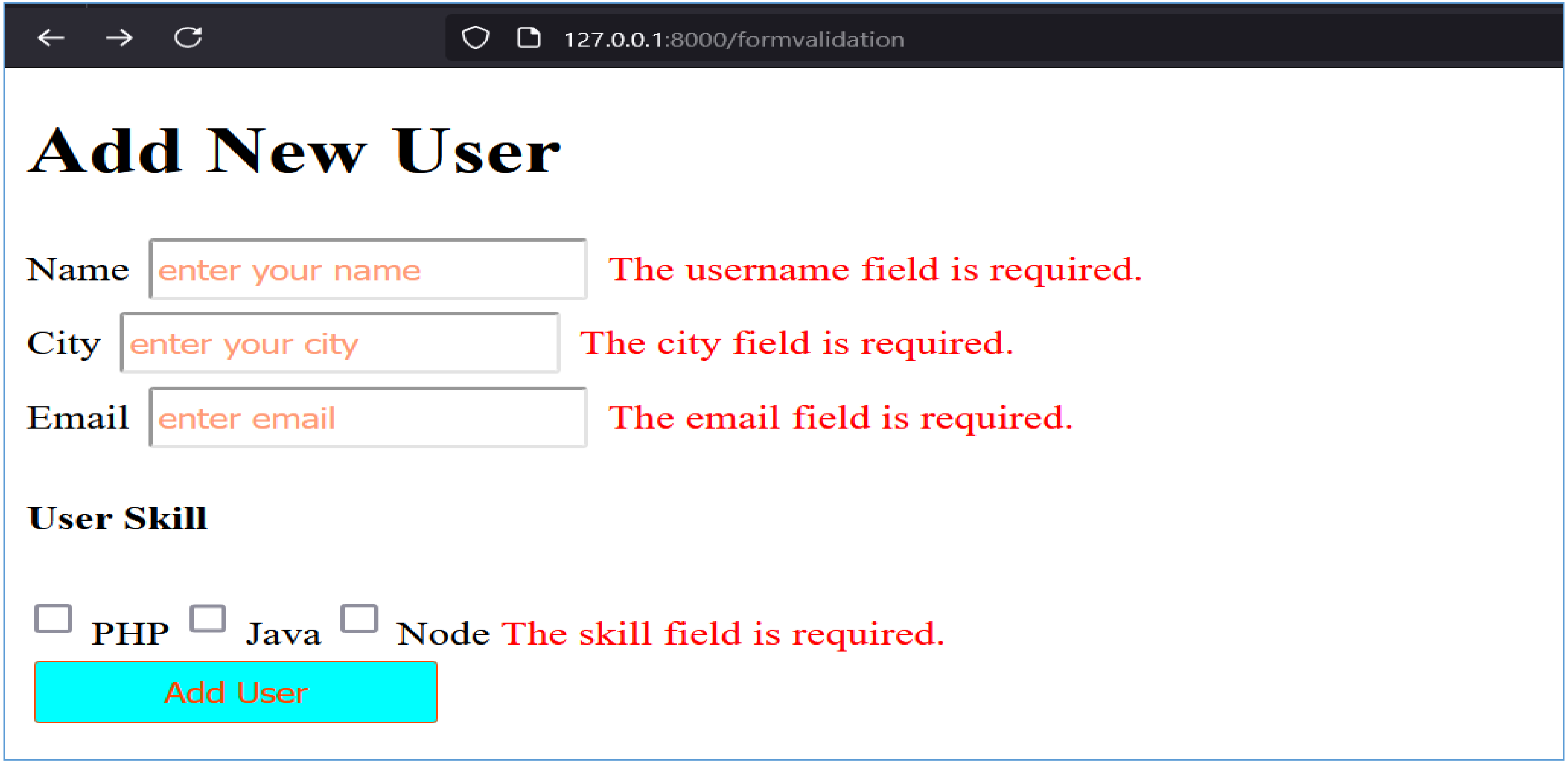
```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class FormValidation extends Controller
{
    function formvalidation(Request $request){
        $request->validate(['username'=>'required | min:3 |max:15',
                            'city' => 'required | min:3 |max:15',
                            'email' => 'required | email',
                            'skill' => 'required'

    ]);
    return $request;

    }
}
```

Cont. ...



```php
// Form Vallidation
Route::view('formvalidation','formvalidation');
Route::post('validation',[FormValidation::class,"formvalidation"]);
```

Cont. ...



127.0.0.1:8000/formvalidation

Add New User

Name enter your name The username field is required.

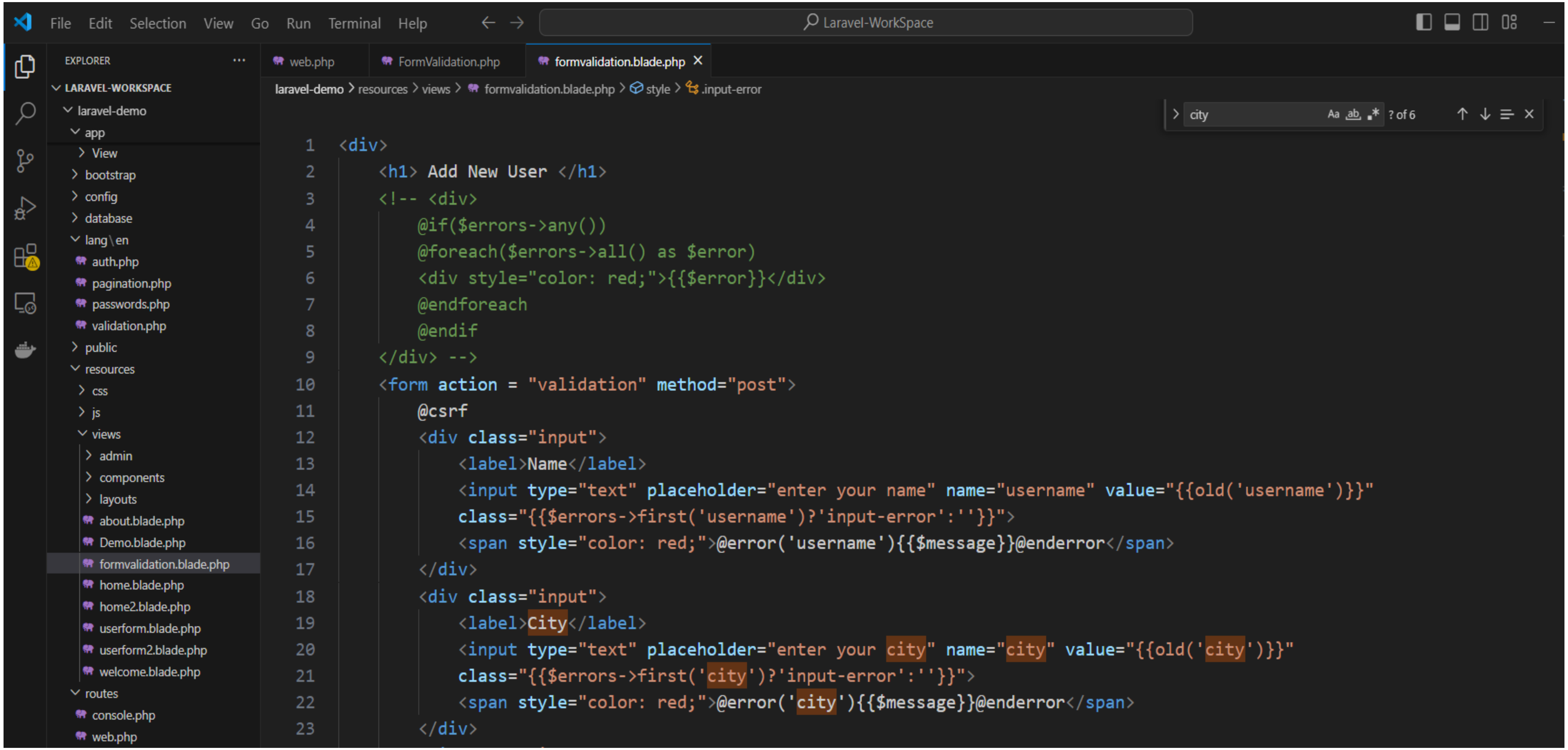
City enter your city The city field is required.

Email enter email The email field is required.

User Skill

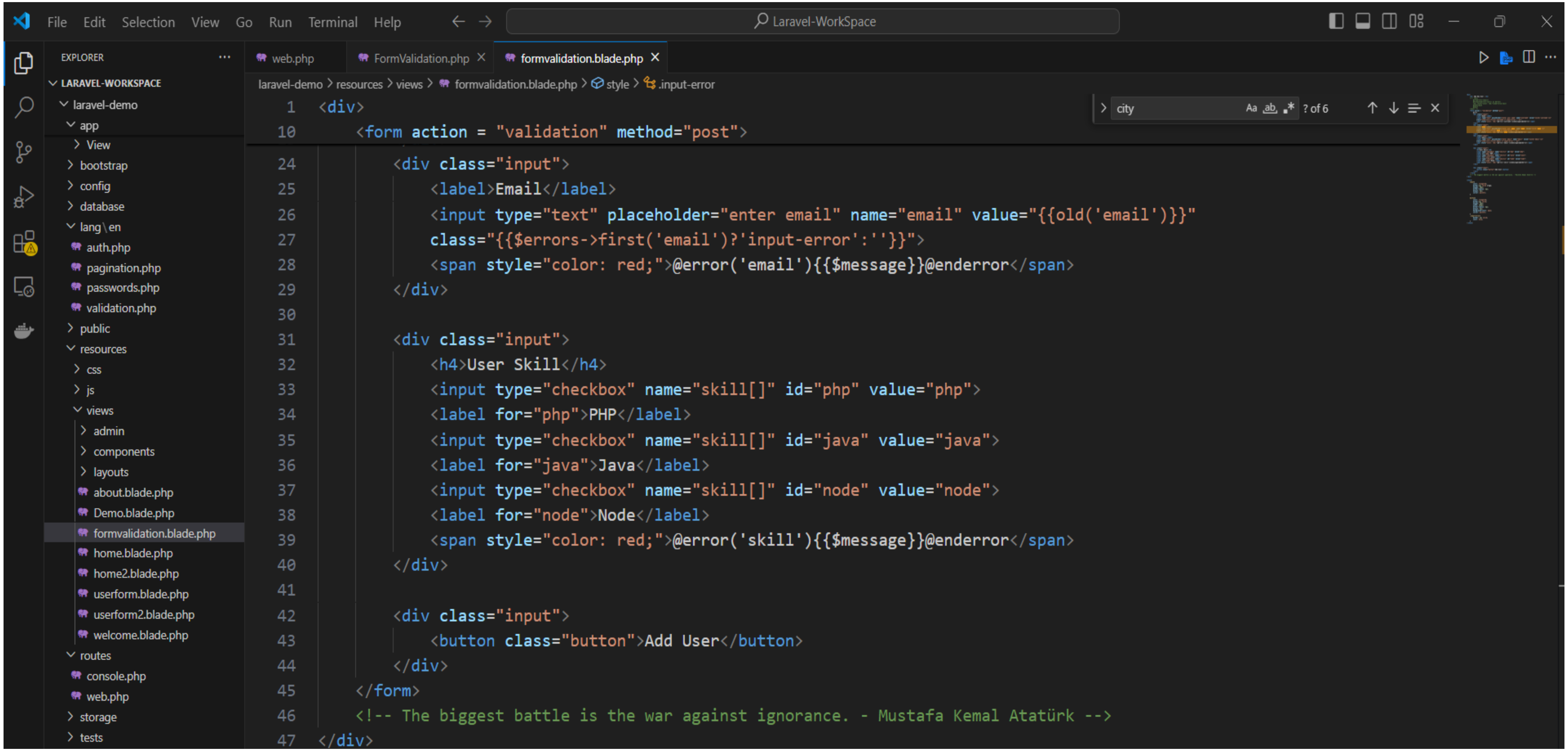
□ PHP □ Java □ Node The skill field is required.

Custom Validation Message



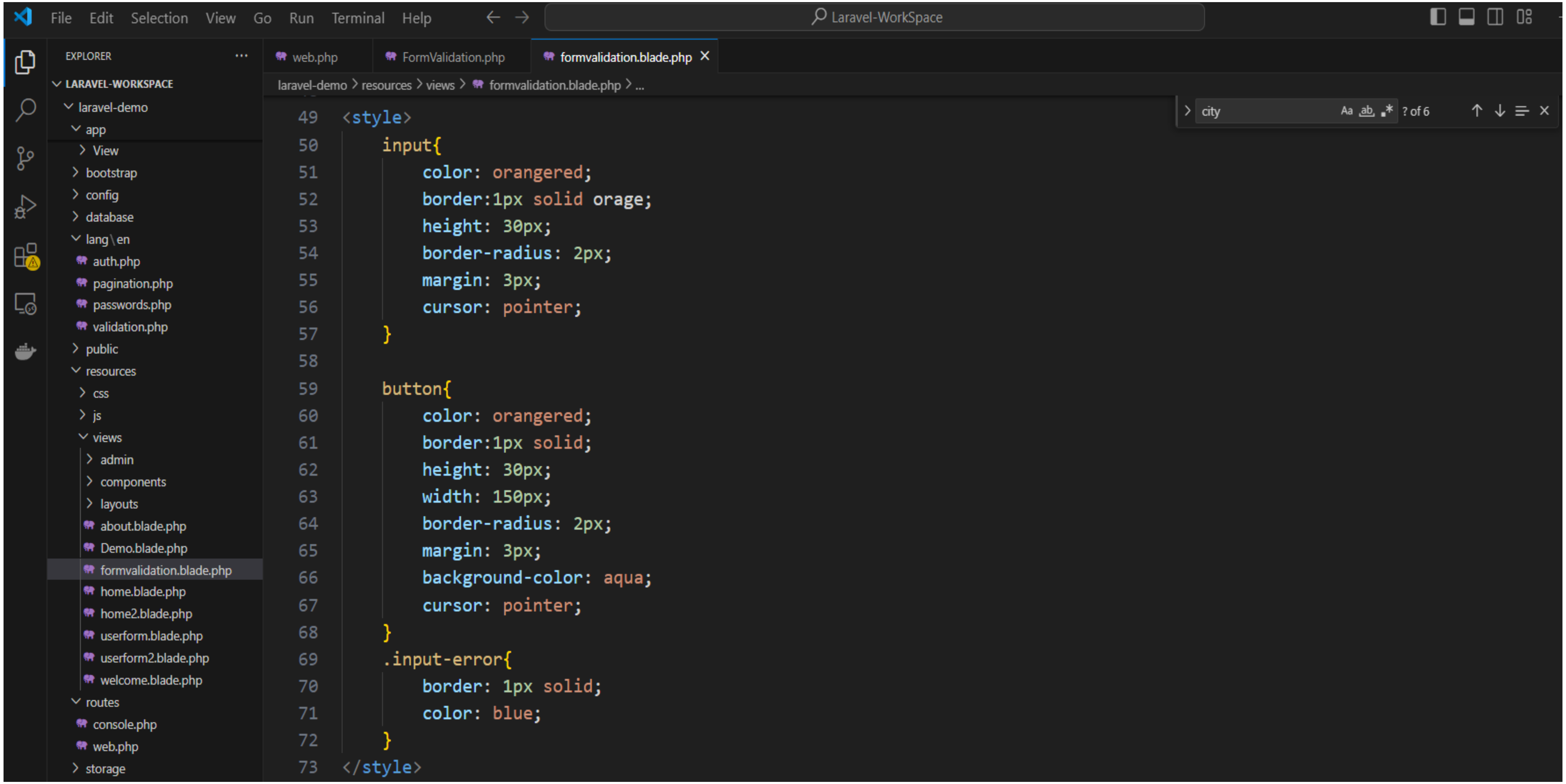
```
<div>
    <h1> Add New User </h1>
    <!-- <div>
        @if($errors->any())
        @foreach($errors->all() as $error)
        <div style="color: red;">{{$error}}</div>
        @endforeach
        @endif
    </div> -->
    <form action = "validation" method="post">
        @csrf
        <div class="input">
            <label>Name</label>
            <input type="text" placeholder="enter your name" name="username" value="{{old('username')}}"
            class="{{$errors->first('username')?'input-error':''}}">
            <span style="color: red;">@error('username'){{$message}}@enderror</span>
        </div>
        <div class="input">
            <label>City</label>
            <input type="text" placeholder="enter your city" name="city" value="{{old('city')}}"
            class="{{$errors->first('city')?'input-error':''}}">
            <span style="color: red;">@error('city'){{$message}}@enderror</span>
        </div>
```

Cont. ...



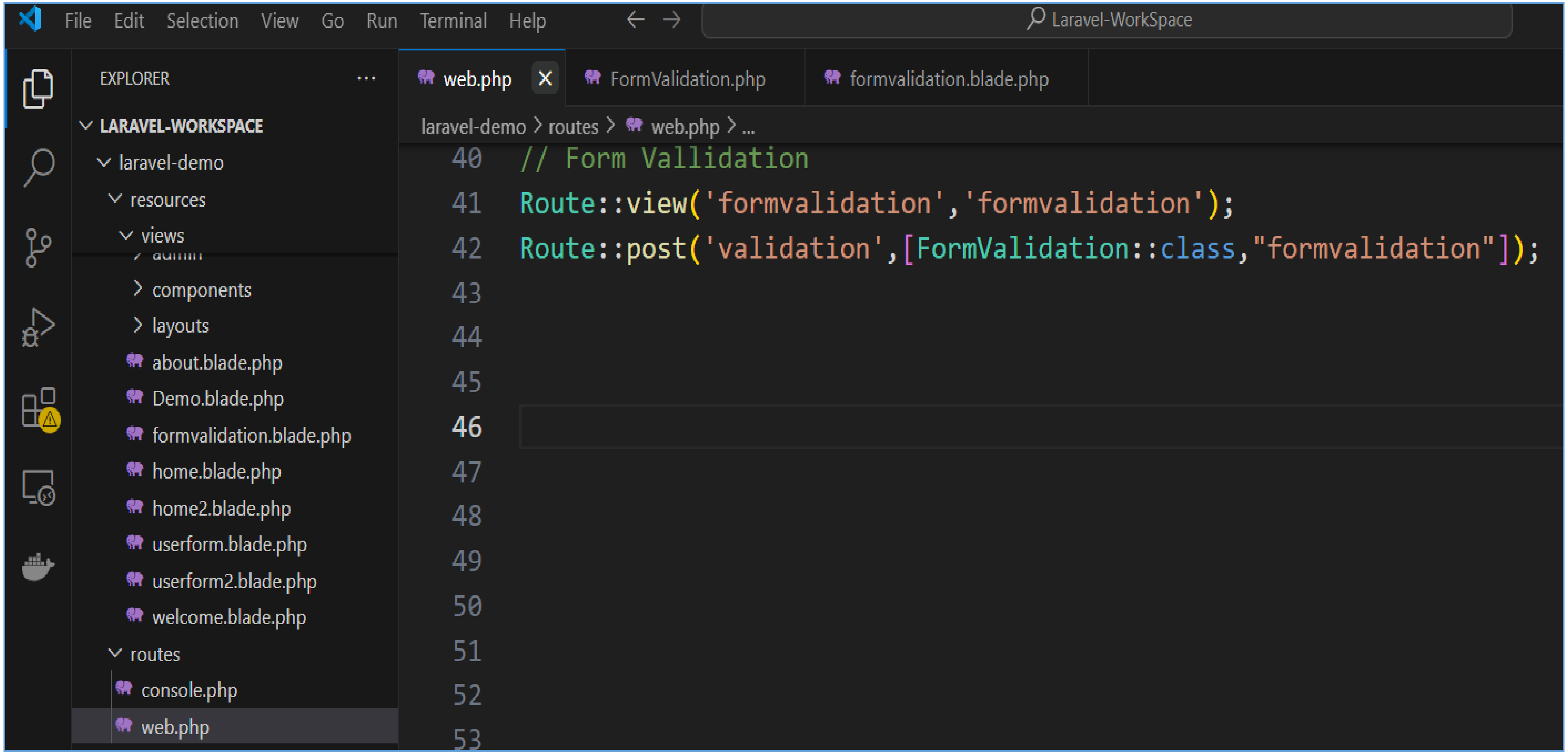
```
<div>
    <form action = "validation" method="post">
        <div class="input">
            <label>Email</label>
            <input type="text" placeholder="enter email" name="email" value="{{old('email')}}"
            class="{{$errors->first('email')?'input-error':''}}">
            <span style="color: red;">@error('email'){{$message}}@enderror</span>
        </div>

        <div class="input">
            <h4>User Skill</h4>
            <input type="checkbox" name="skill[]" id="php" value="php">
            <label for="php">PHP</label>
            <input type="checkbox" name="skill[]" id="java" value="java">
            <label for="java">Java</label>
            <input type="checkbox" name="skill[]" id="node" value="node">
            <label for="node">Node</label>
            <span style="color: red;">@error('skill'){{$message}}@enderror</span>
        </div>

        <div class="input">
            <button class="button">Add User</button>
        </div>
    </form>
    <!-- The biggest battle is the war against ignorance. - Mustafa Kemal Atatürk -->
</div>
```

Cont. ...



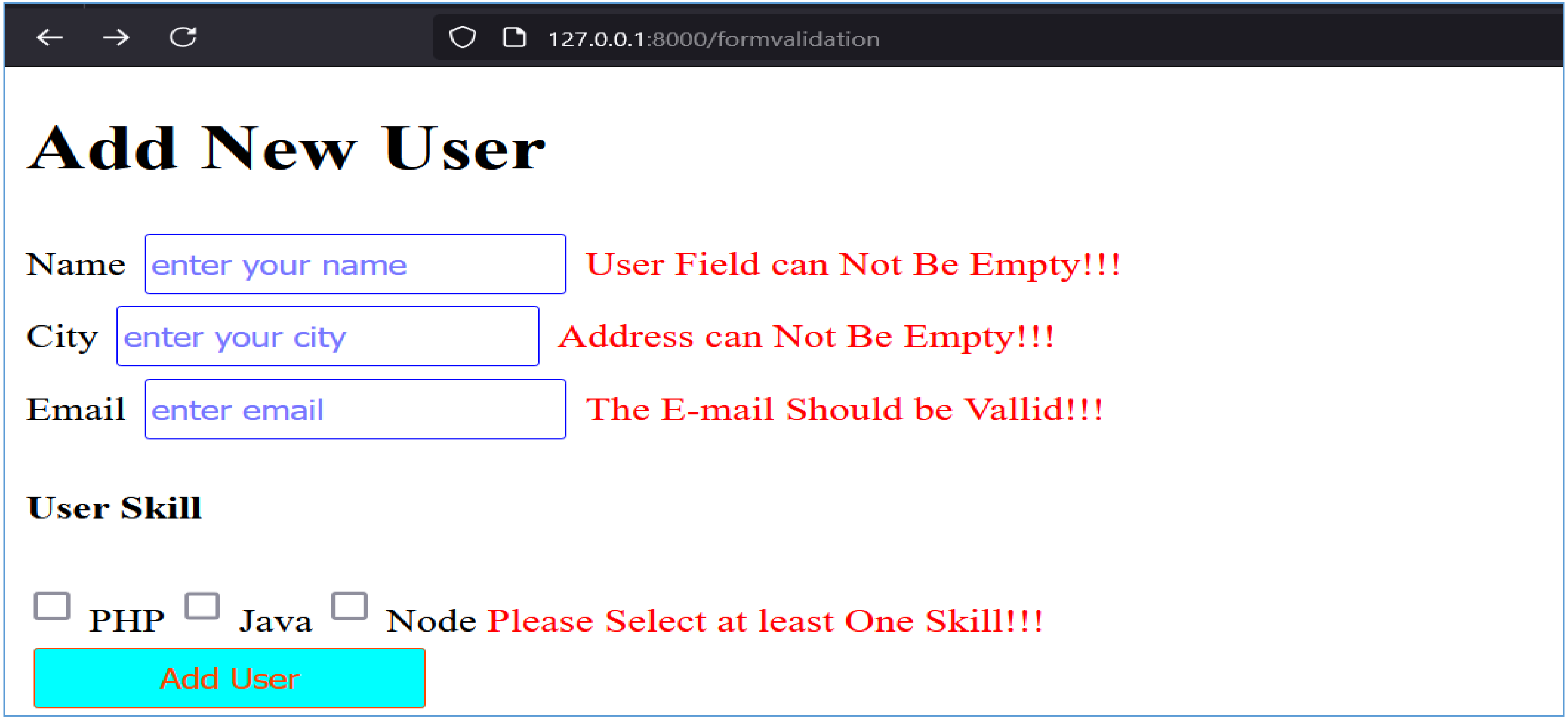
```
<style>
    input{
        color: orangered;
        border:1px solid orage;
        height: 30px;
        border-radius: 2px;
        margin: 3px;
        cursor: pointer;
    }

    button{
        color: orangered;
        border:1px solid;
        height: 30px;
        width: 150px;
        border-radius: 2px;
        margin: 3px;
        background-color: aqua;
        cursor: pointer;
    }
    .input-error{
        border: 1px solid;
        color: blue;
    }
</style>
```

Cont. …



```php
// Form Vallidation
Route::view('formvalidation','formvalidation');
Route::post('validation',[FormValidation::class,"formvalidation"]);
```

Cont. ...



127.0.0.1:8000/formvalidation

Add New User

Name enter your name User Field can Not Be Empty!!!

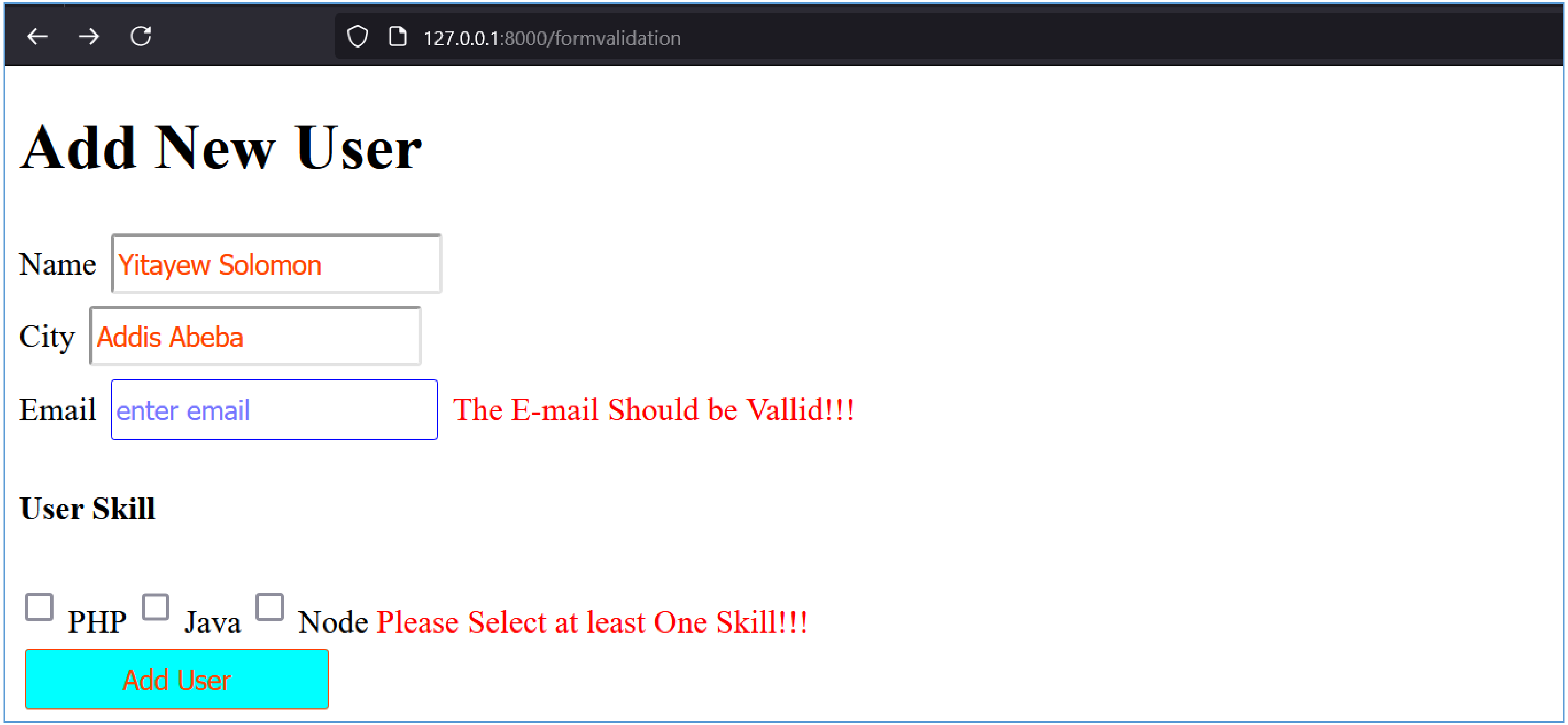
City enter your city Address can Not Be Empty!!!

Email enter email The E-mail Should be Vallid!!!

User Skill

□ PHP □ Java □ Node Please Select at least One Skill!!!

User Data Are Saved



Add New User

Name Yitayew Solomon

City Addis Abeba

Email enter email The E-mail Should be Vallid!!!

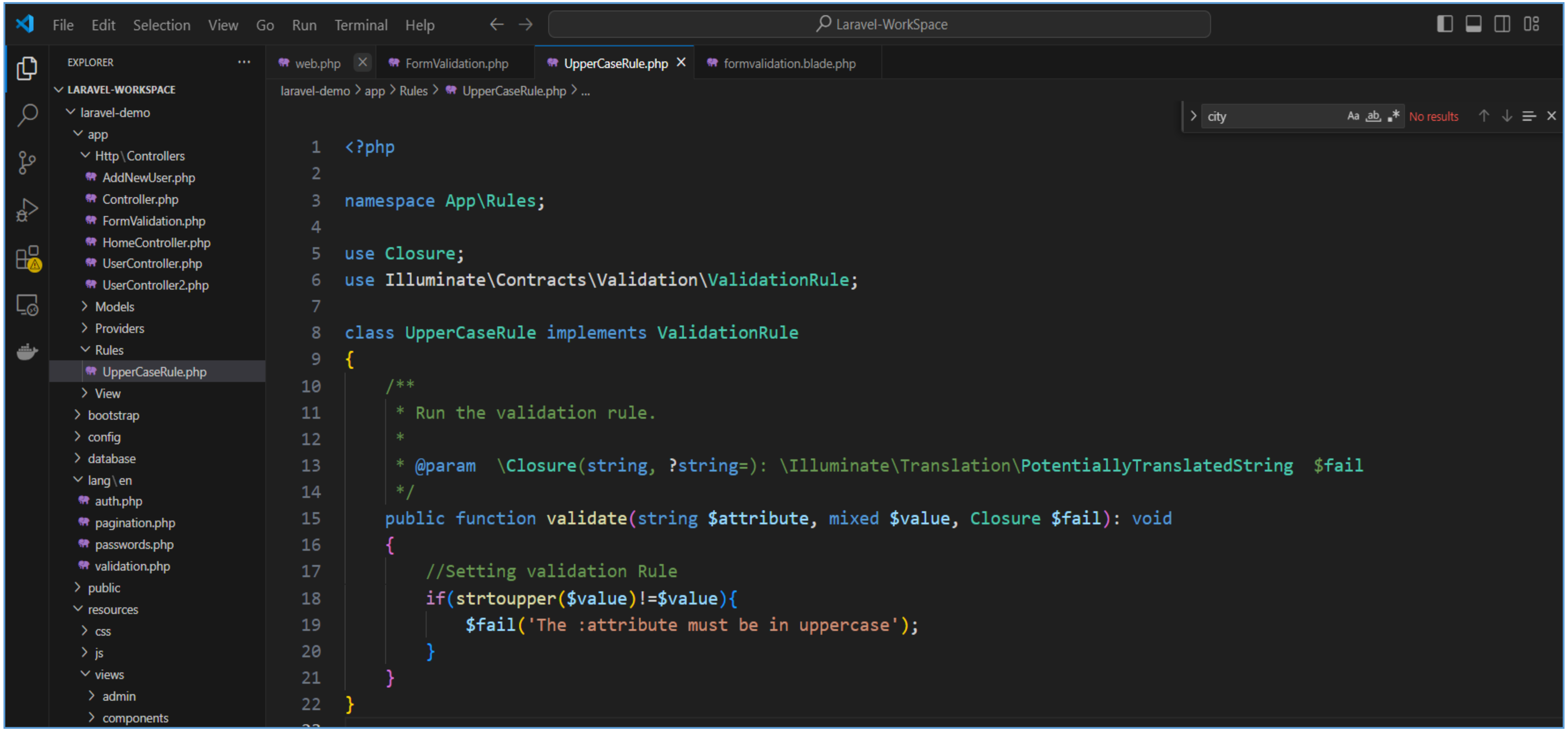
User Skill

□ PHP □ Java □ Node Please Select at least One Skill!!!

Make Own Validation

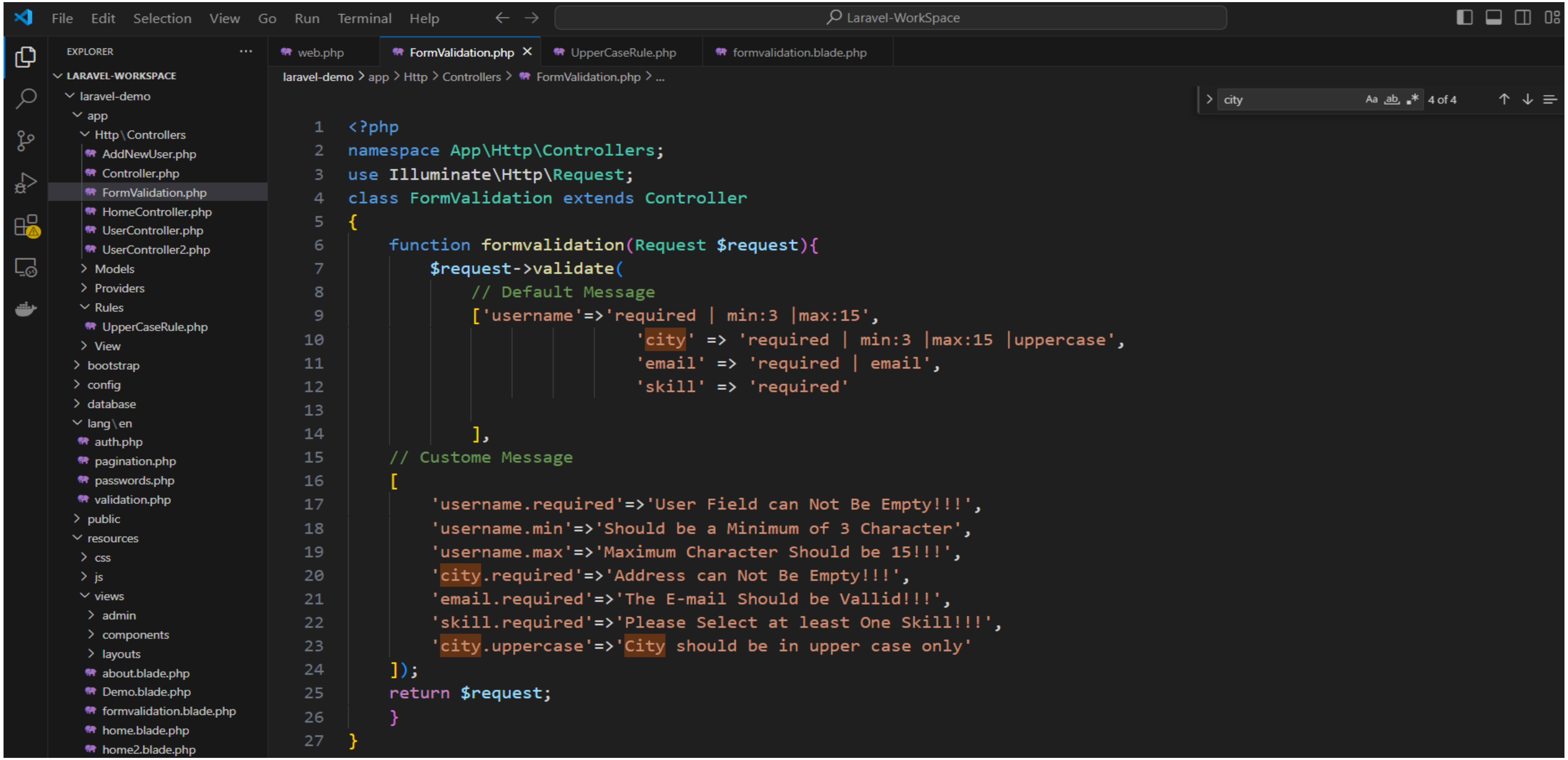
- In Laravel, you can create custom validation rules to handle scenarios that are not covered by the **default validation rules**. This involves creating a **custom validation class** or using **closures** in your validation logic.

Cont. ...



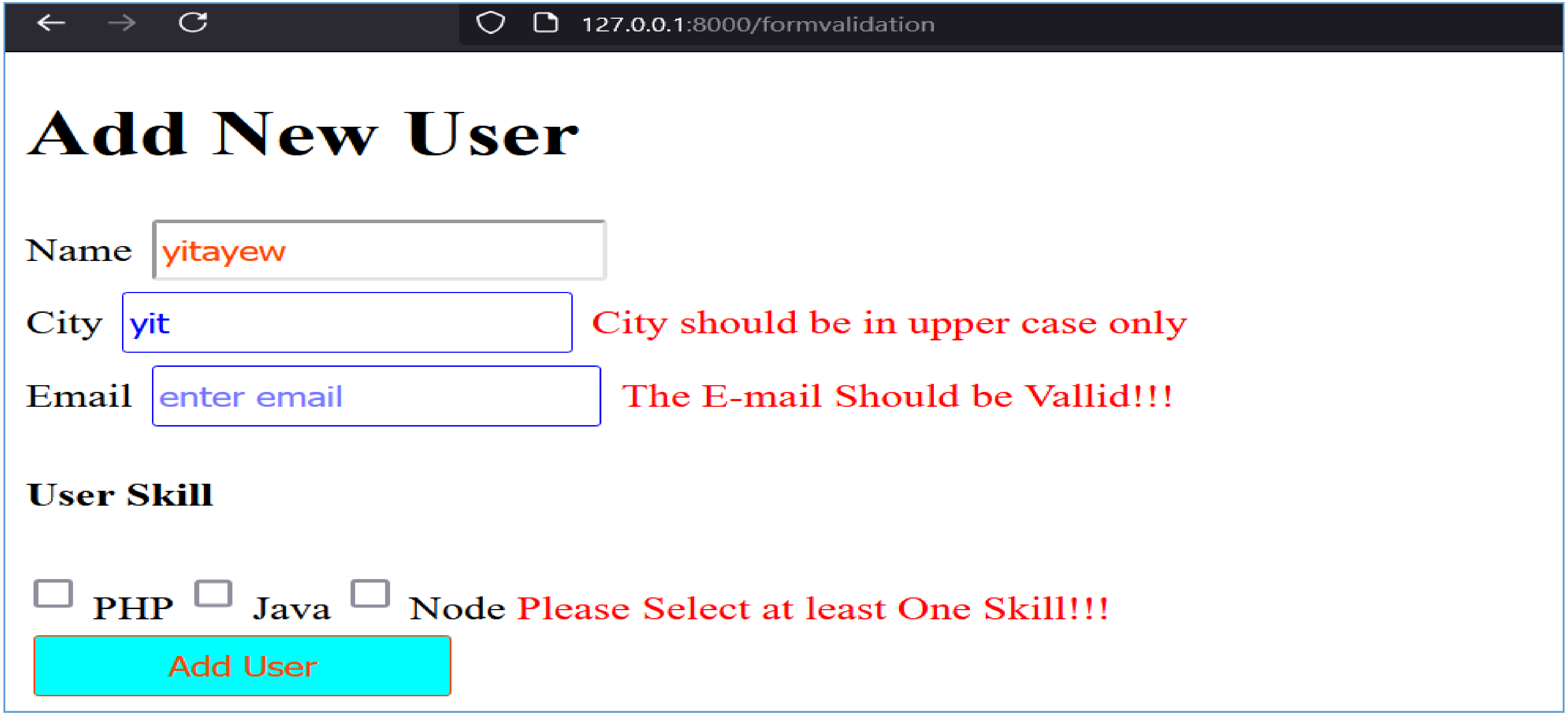
```php
<?php

namespace App\Rules;

use Closure;
use Illuminate\Contracts\Validation\ValidationRule;

class UpperCaseRule implements ValidationRule
{
    /**
     * Run the validation rule.
     *
     * @param  \Closure(string, ?string=): \Illuminate\Translation\PotentiallyTranslatedString  $fail
     */
    public function validate(string $attribute, mixed $value, Closure $fail): void
    {
        //Setting validation Rule
        if(strtoupper($value)!=$value){
            $fail('The :attribute must be in uppercase');
        }
    }
}
```

Cont. …



```php
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
class FormValidation extends Controller
{
    function formvalidation(Request $request){
        $request->validate(
            // Default Message
            ['username'=>'required | min:3 |max:15',
                            'city' => 'required | min:3 |max:15 |uppercase',
                            'email' => 'required | email',
                            'skill' => 'required'

            ],
    // Custome Message
    [
        'username.required'=>'User Field can Not Be Empty!!!',
        'username.min'=>'Should be a Minimum of 3 Character',
        'username.max'=>'Maximum Character Should be 15!!!',
        'city.required'=>'Address can Not Be Empty!!!',
        'email.required'=>'The E-mail Should be Vallid!!!',
        'skill.required'=>'Please Select at least One Skill!!!',
        'city.uppercase'=>'City should be in upper case only'
    ]);
    return $request;
    }
}
```

Cont. ...



Add New User

Name yitayew

City yit City should be in upper case only

Email enter email The E-mail Should be Vallid!!!

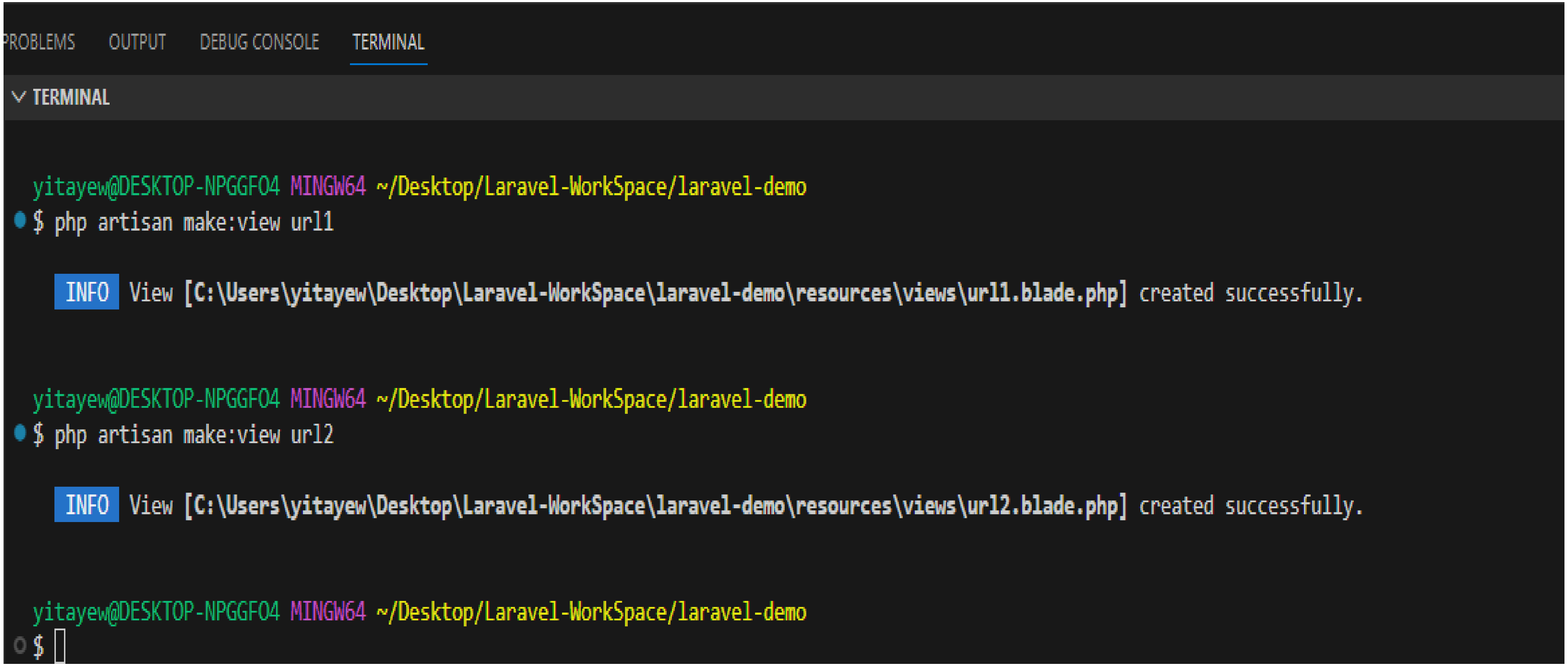
User Skill

□ PHP □ Java □ Node Please Select at least One Skill!!!

URL Generation

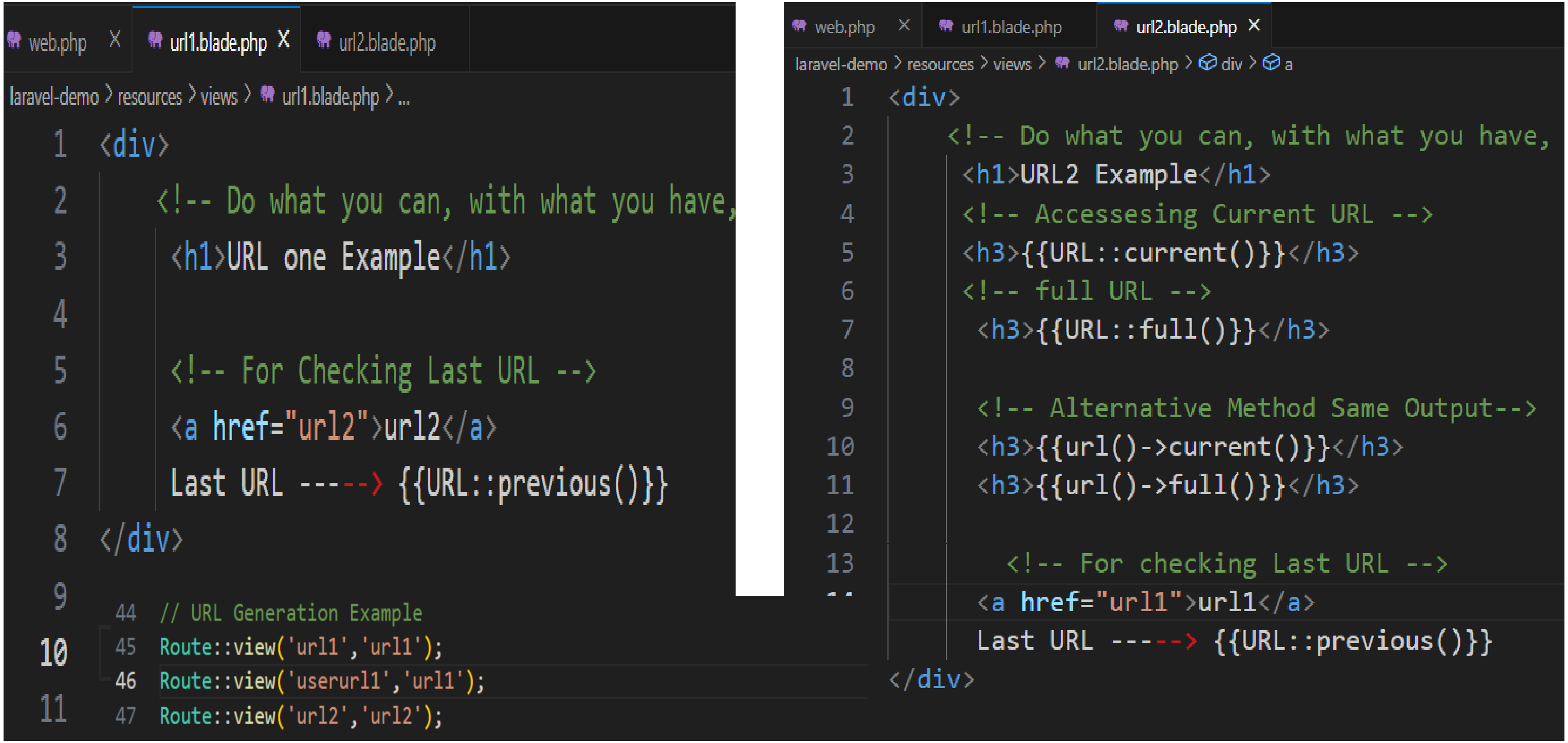
- In Laravel, URL generation is a core feature that allows you to create URLs dynamically within your application, making it easier to manage links and routes.
- Laravel provides several **helper functions** and **methods** to generate URLs.

Creating Views



```
yitayew@DESKTOP-NPGGF04 MINGW64 ~/Desktop/Laravel-WorkSpace/laravel-demo
$ php artisan make:view url1

  INFO  View [C:\Users\yitayew\Desktop\Laravel-WorkSpace\laravel-demo\resources\views\url1.blade.php] created successfully.


yitayew@DESKTOP-NPGGF04 MINGW64 ~/Desktop/Laravel-WorkSpace/laravel-demo
$ php artisan make:view url2

  INFO  View [C:\Users\yitayew\Desktop\Laravel-WorkSpace\laravel-demo\resources\views\url2.blade.php] created successfully.


yitayew@DESKTOP-NPGGF04 MINGW64 ~/Desktop/Laravel-WorkSpace/laravel-demo
$ 
```

Cont. …

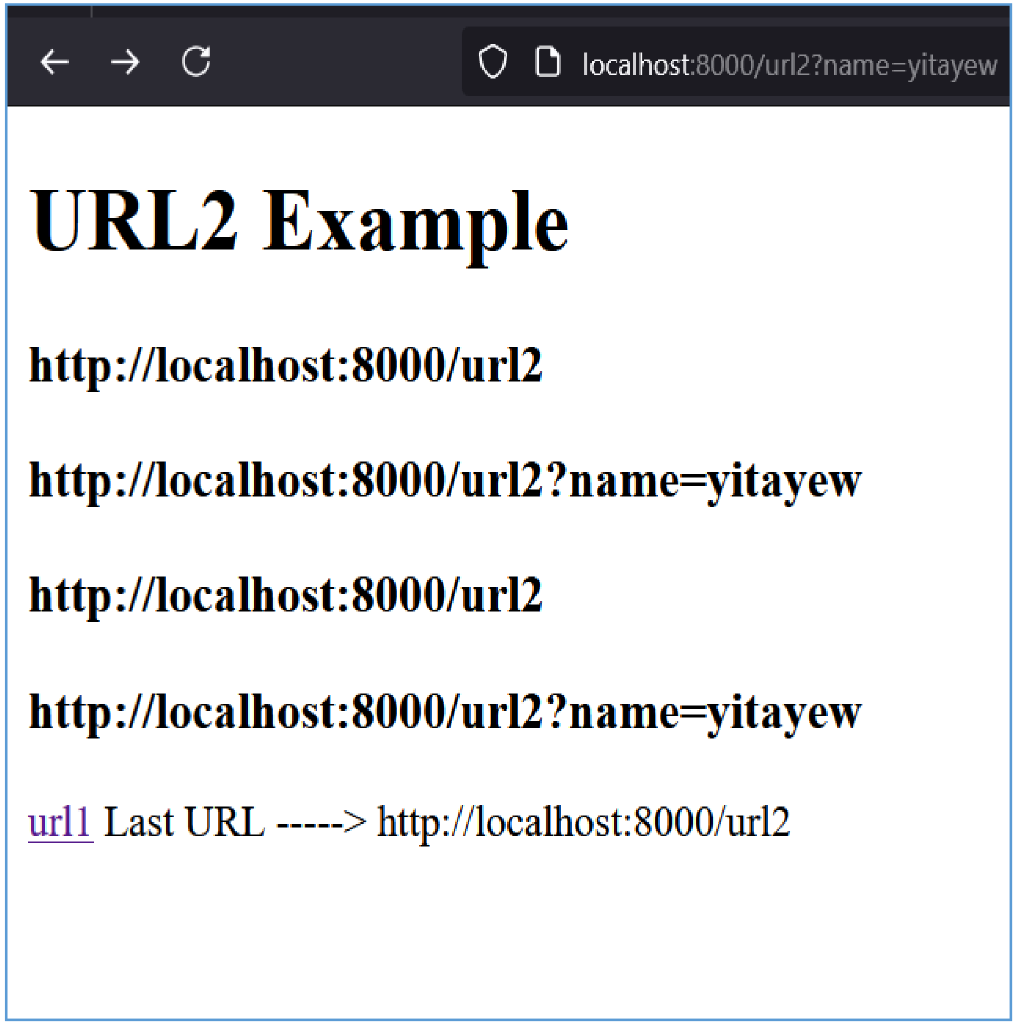

web.php
url1.blade.php
url2.blade.php
laravel-demo › resources › views › url1.blade.php › …
laravel-demo › resources › views › url2.blade.php › div › a

```
<div>
    <!-- Do what you can, with what you have,
     <h1>URL one Example</h1>

     <!-- For Checking Last URL -->
     <a href="url2">url2</a>
     Last URL ----> {{URL::previous()}}
</div>
```

```
// URL Generation Example
Route::view('url1','url1');
Route::view('userurl1','url1');
Route::view('url2','url2');
```

```
<div>
    <!-- Do what you can, with what you have,
     <h1>URL2 Example</h1>
     <!-- Accessesing Current URL -->
     <h3>{{URL::current()}}</h3>
     <!-- full URL -->
      <h3>{{URL::full()}}</h3>

      <!-- Alternative Method Same Output-->
      <h3>{{url()->current()}}</h3>
      <h3>{{url()->full()}}</h3>

        <!-- For checking Last URL -->
      <a href="url1">url1</a>
      Last URL ----> {{URL::previous()}}
</div>
```

Cont. ...



URL2 Example

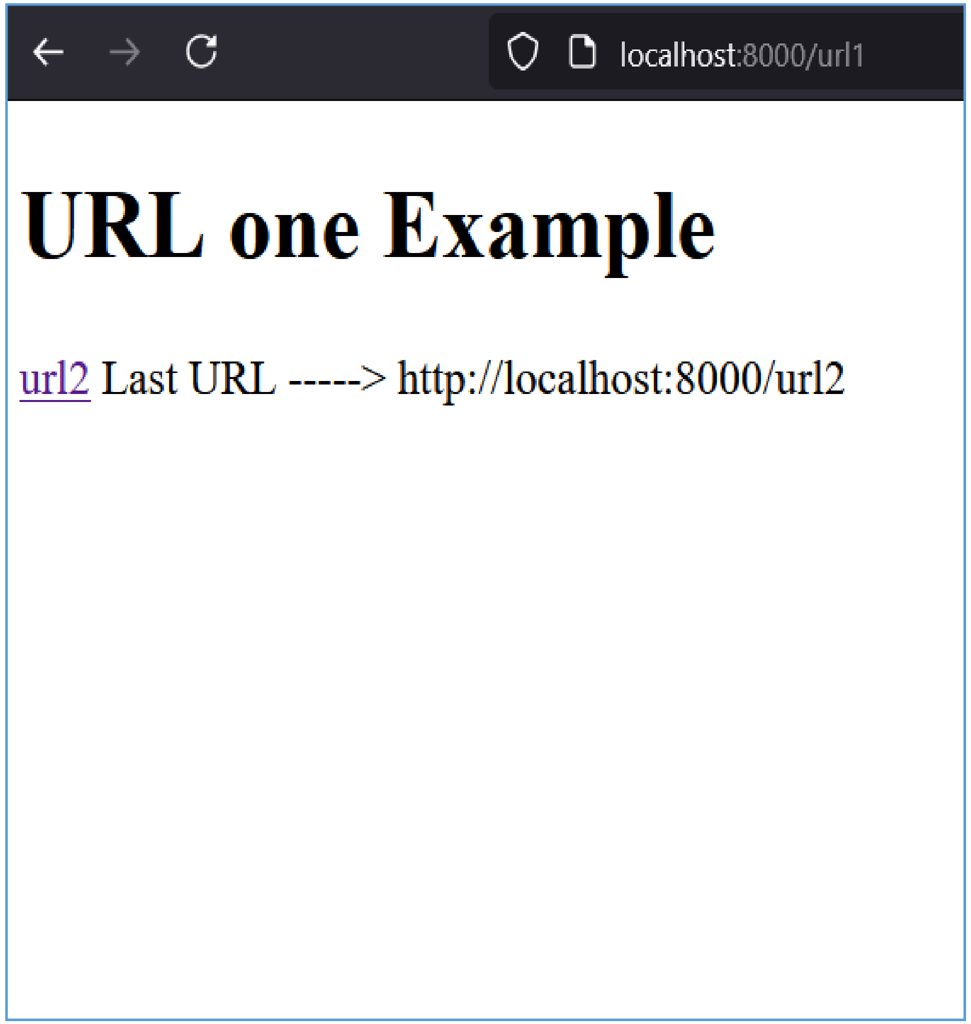
http://localhost:8000/url2

http://localhost:8000/url2?name=yitayew

http://localhost:8000/url2

http://localhost:8000/url2?name=yitayew

url1 Last URL -----> http://localhost:8000/url2



URL one Example

url2 Last URL -----> http://localhost:8000/url2

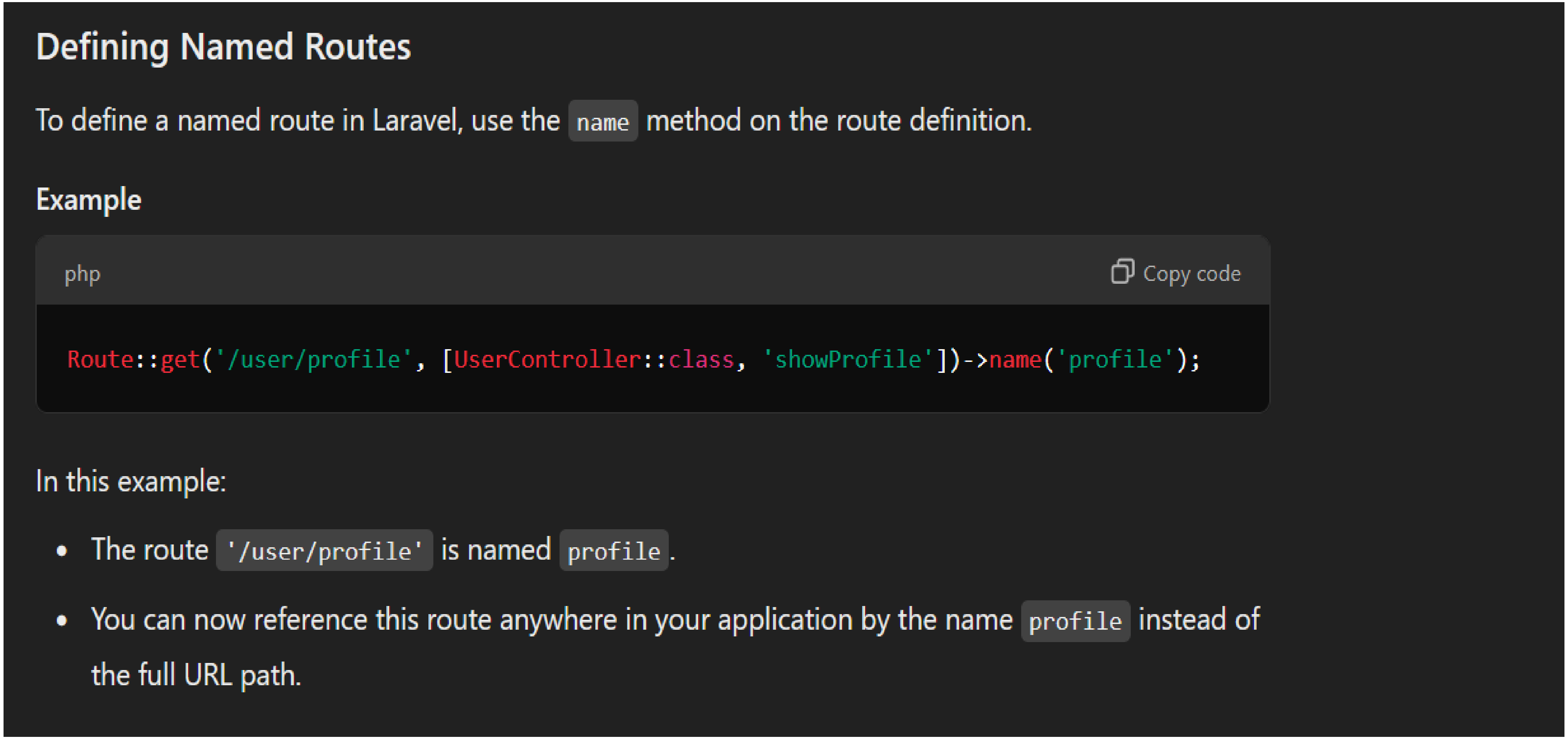
Named Route in Laravel

- A named route in Laravel is a route with a **specific name assigned** to it. Named routes allow you to reference the route by its **name** rather than by its URL path, which makes your code more flexible and easier to maintain.
- Instead of hardcoding URLs throughout your application, you can use named routes to generate URLs or redirects by referring to the route name.

Why Use Named Routes?

- **Maintainability**: If the URL of a route changes, you only need to update the route definition instead of updating every instance of the URL in your application.
- **Readability**: Named routes make it easier to understand the purpose of a route.
- **Flexibility**: Named routes support generating URLs, handling redirects, and passing parameters dynamically.

Defining Named Route



Defining Named Routes

To define a named route in Laravel, use the `name` method on the route definition.

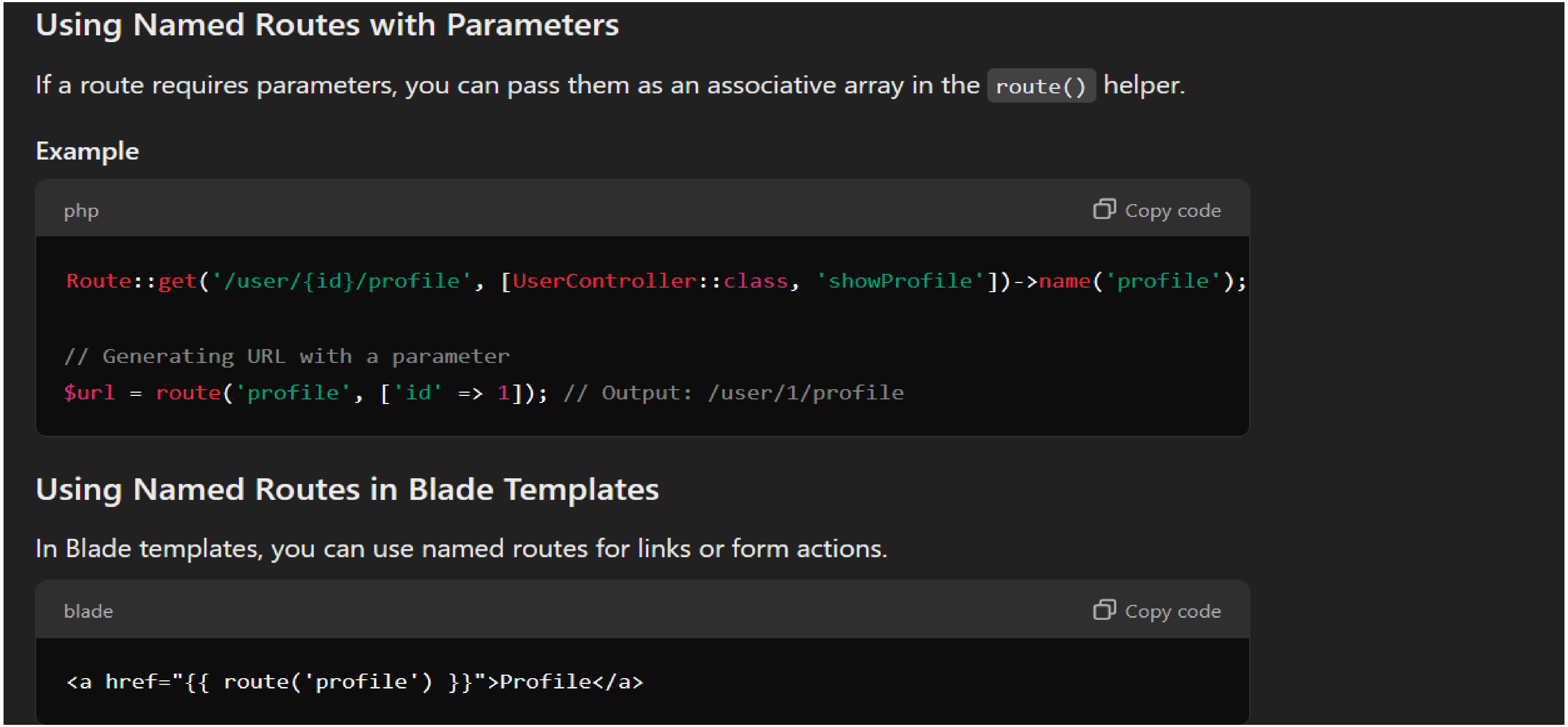
Example

```php
Route::get('/user/profile', [UserController::class, 'showProfile'])->name('profile');
```

In this example:

- The route `'/user/profile'` is named `profile`.
- You can now reference this route anywhere in your application by the name `profile` instead of the full URL path.

Cont. ...



Using Named Routes with Parameters

If a route requires parameters, you can pass them as an associative array in the `route()` helper.

Example

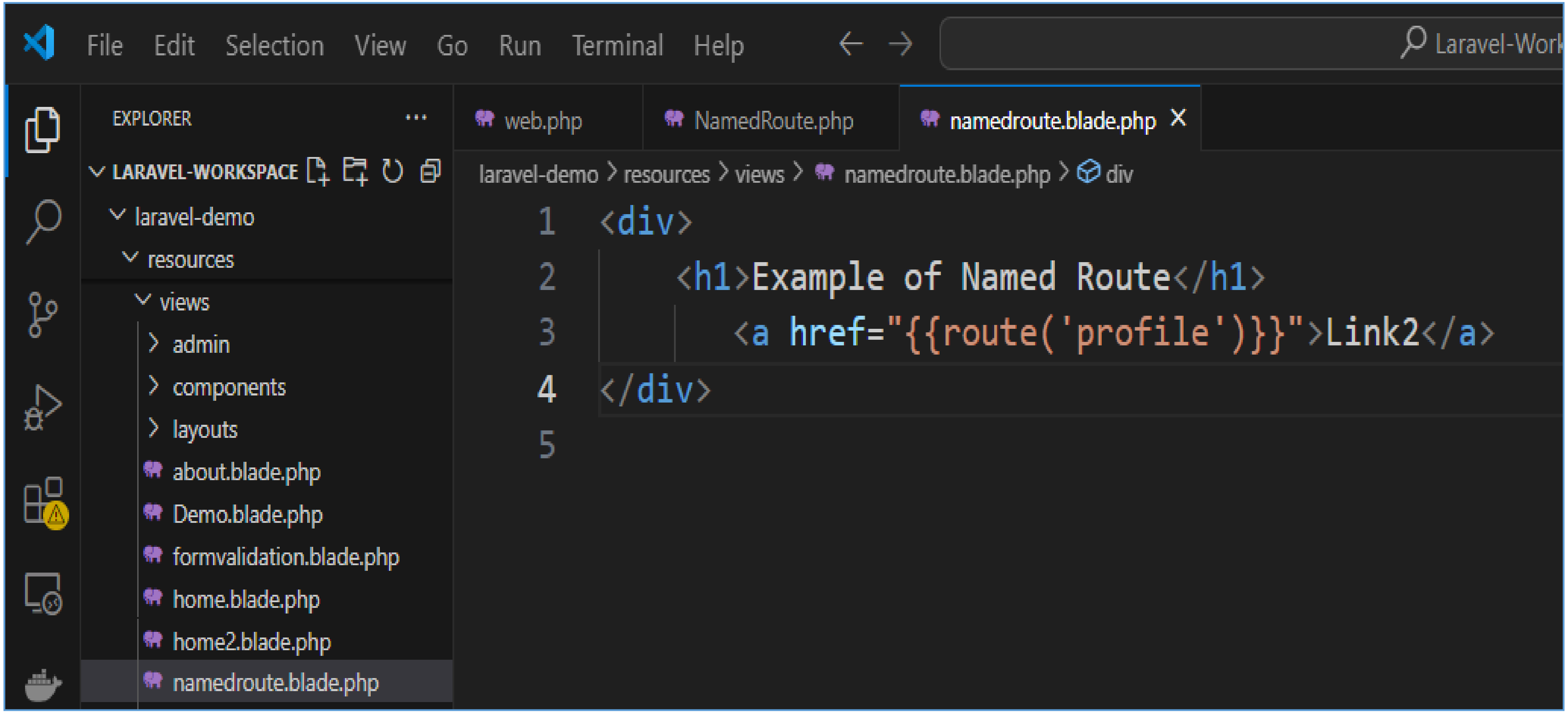
```php
Route::get('/user/{id}/profile', [UserController::class, 'showProfile'])->name('profile');

// Generating URL with a parameter
$url = route('profile', ['id' => 1]); // Output: /user/1/profile
```

Using Named Routes in Blade Templates

In Blade templates, you can use named routes for links or form actions.

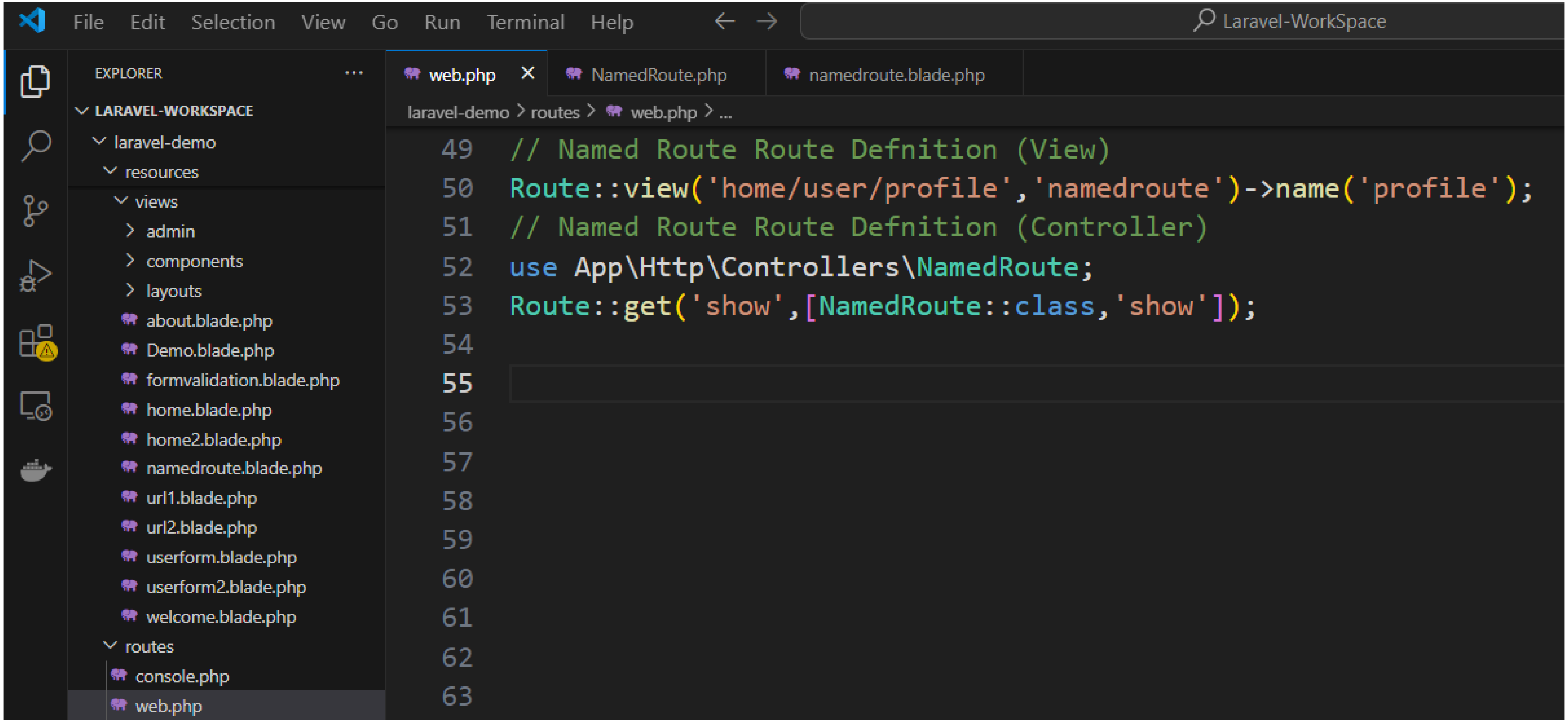
```blade
<a href="{{ route('profile') }}">Profile</a>
```

Example



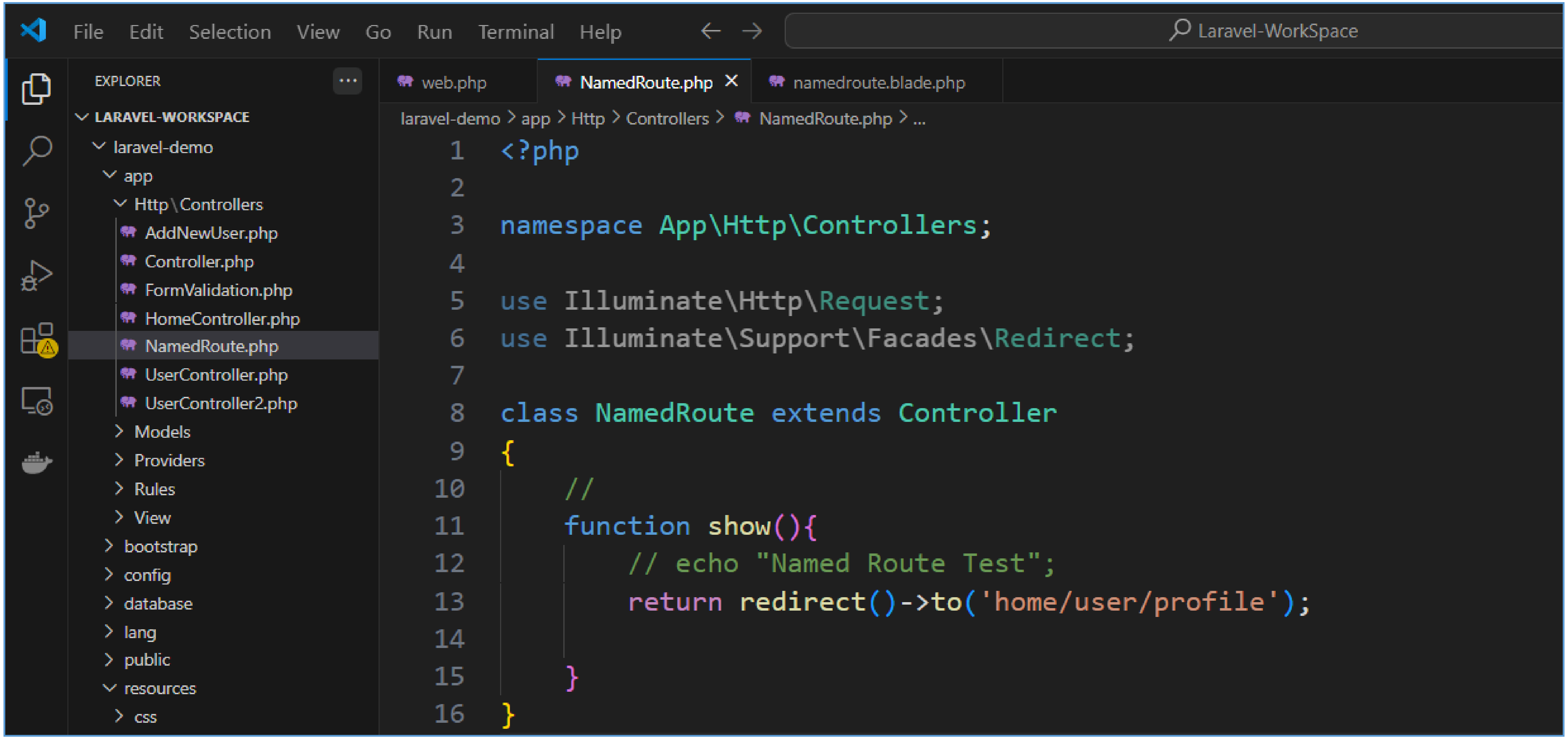
```
<div>
    <h1>Example of Named Route</h1>
        <a href="{{route('profile')}}">Link2</a>
</div>

```

Cont. …



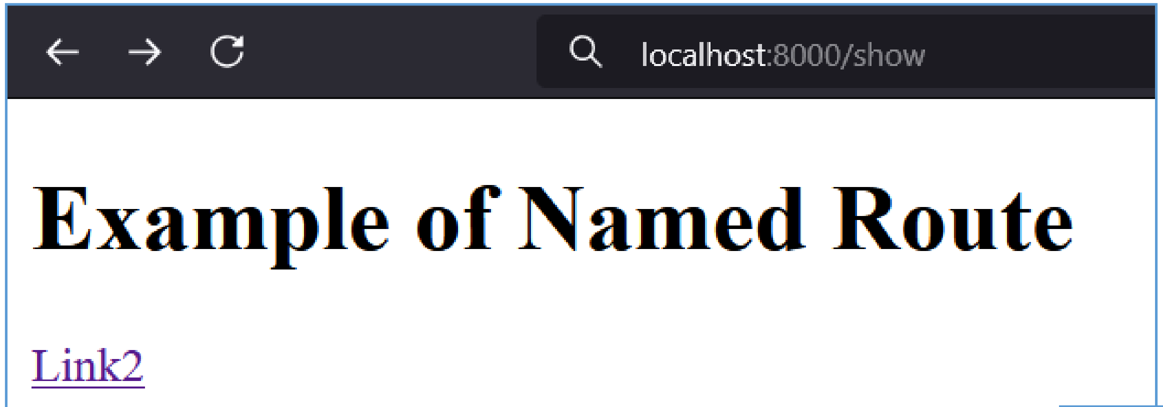
```php
// Named Route Route Defnition (View)
Route::view('home/user/profile','namedroute')->name('profile');
// Named Route Route Defnition (Controller)
use App\Http\Controllers\NamedRoute;
Route::get('show',[NamedRoute::class,'show']);
```

Cont. …



```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Redirect;

class NamedRoute extends Controller
{
    //
    function show(){
        // echo "Named Route Test";
        return redirect()->to('home/user/profile');

    }
}
```

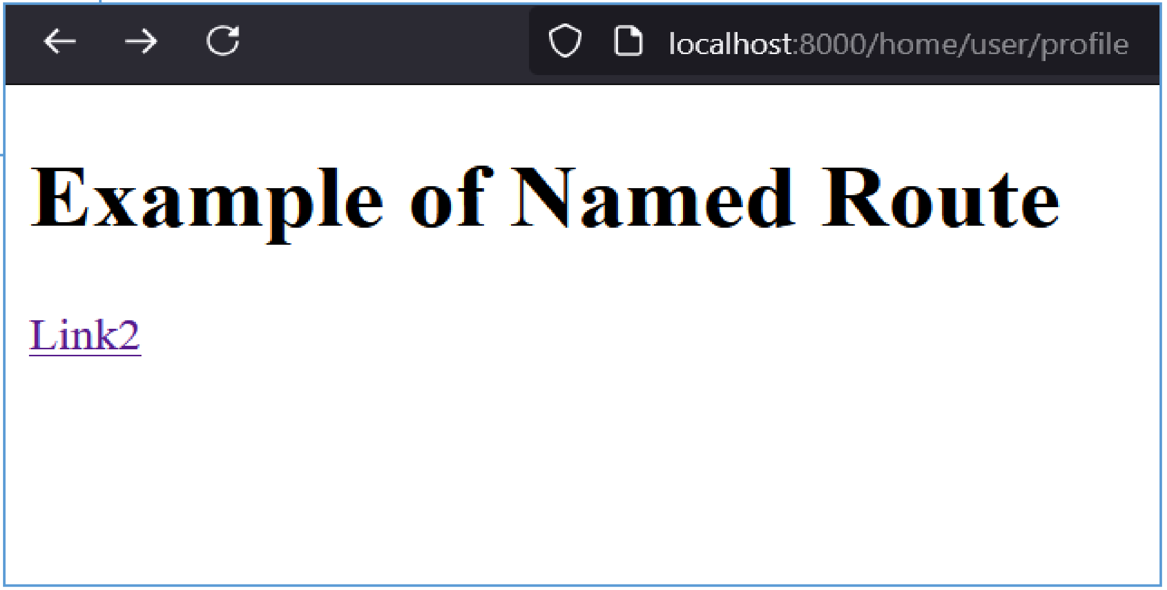
Cont. ...



localhost:8000/show

Example of Named Route

Link2



localhost:8000/home/user/profile

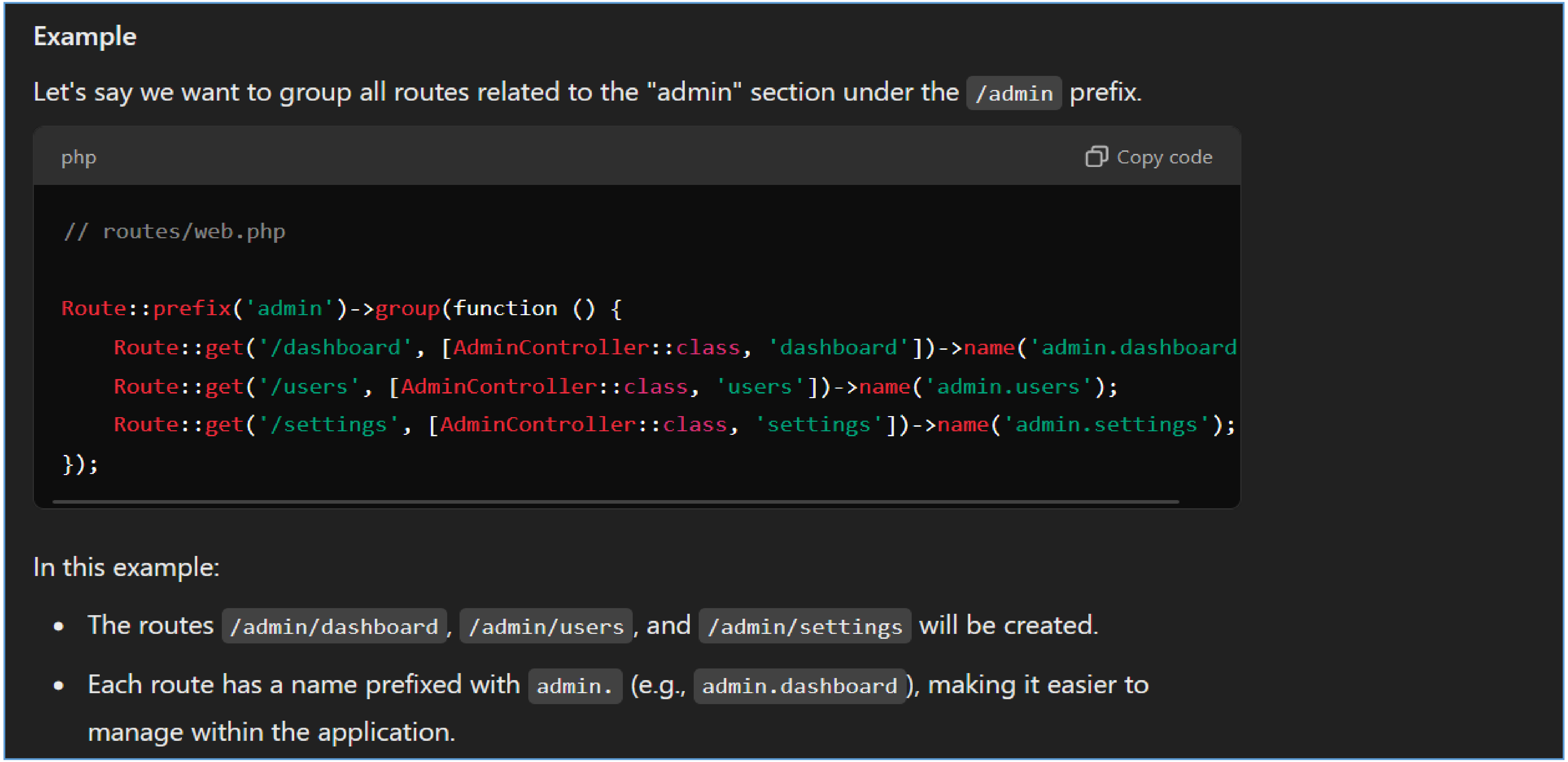
Example of Named Route

Link2

Route Group with a Prefix

- In Laravel, a route group with a prefix allows you to apply a common URL prefix to multiple routes, **simplifying route management**.
- This is useful for **organizing routes** with similar structures, such as routes for the admin panel, API, or user profile sections.

Cont. ...



Example

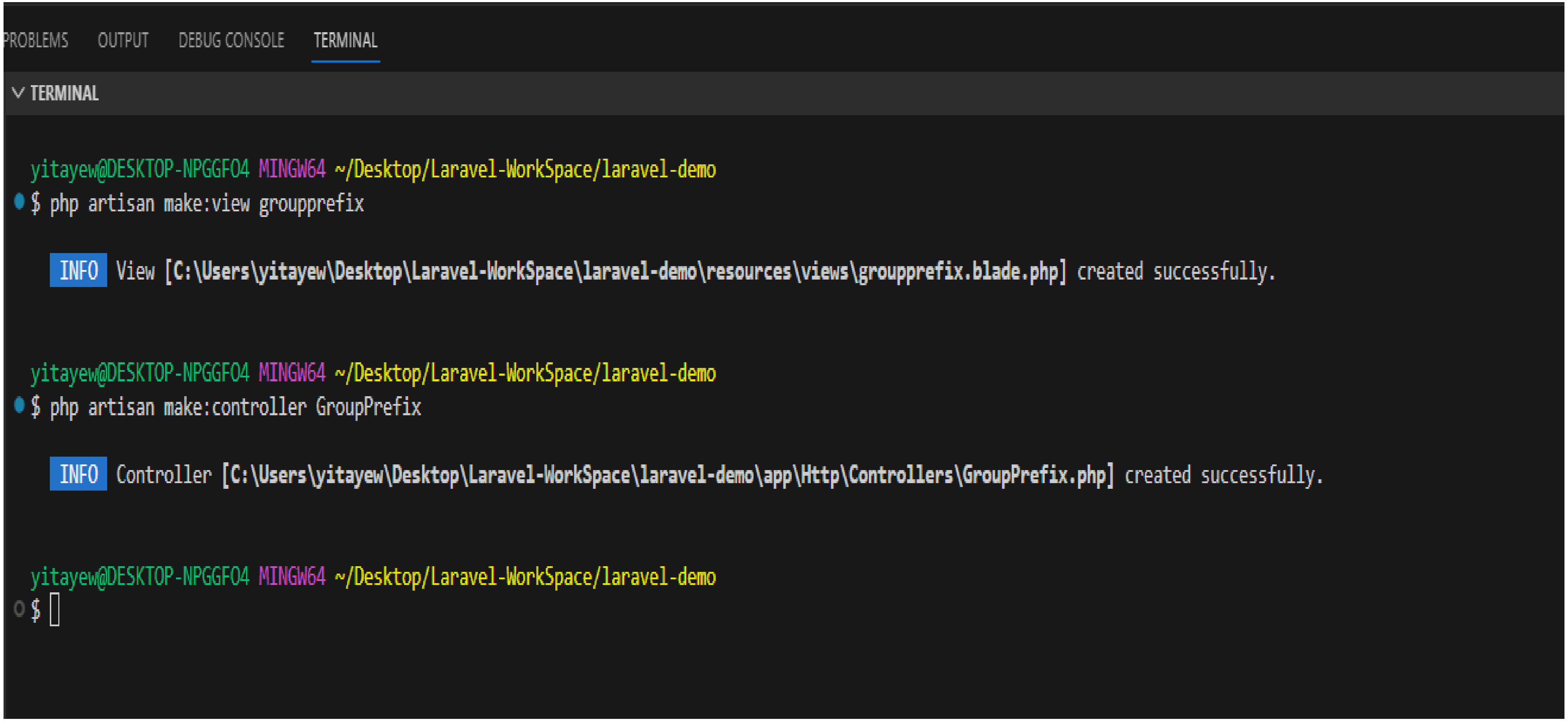
Let's say we want to group all routes related to the "admin" section under the `/admin` prefix.

```php
// routes/web.php

Route::prefix('admin')->group(function () {
    Route::get('/dashboard', [AdminController::class, 'dashboard'])->name('admin.dashboard
    Route::get('/users', [AdminController::class, 'users'])->name('admin.users');
    Route::get('/settings', [AdminController::class, 'settings'])->name('admin.settings');
});
```

In this example:

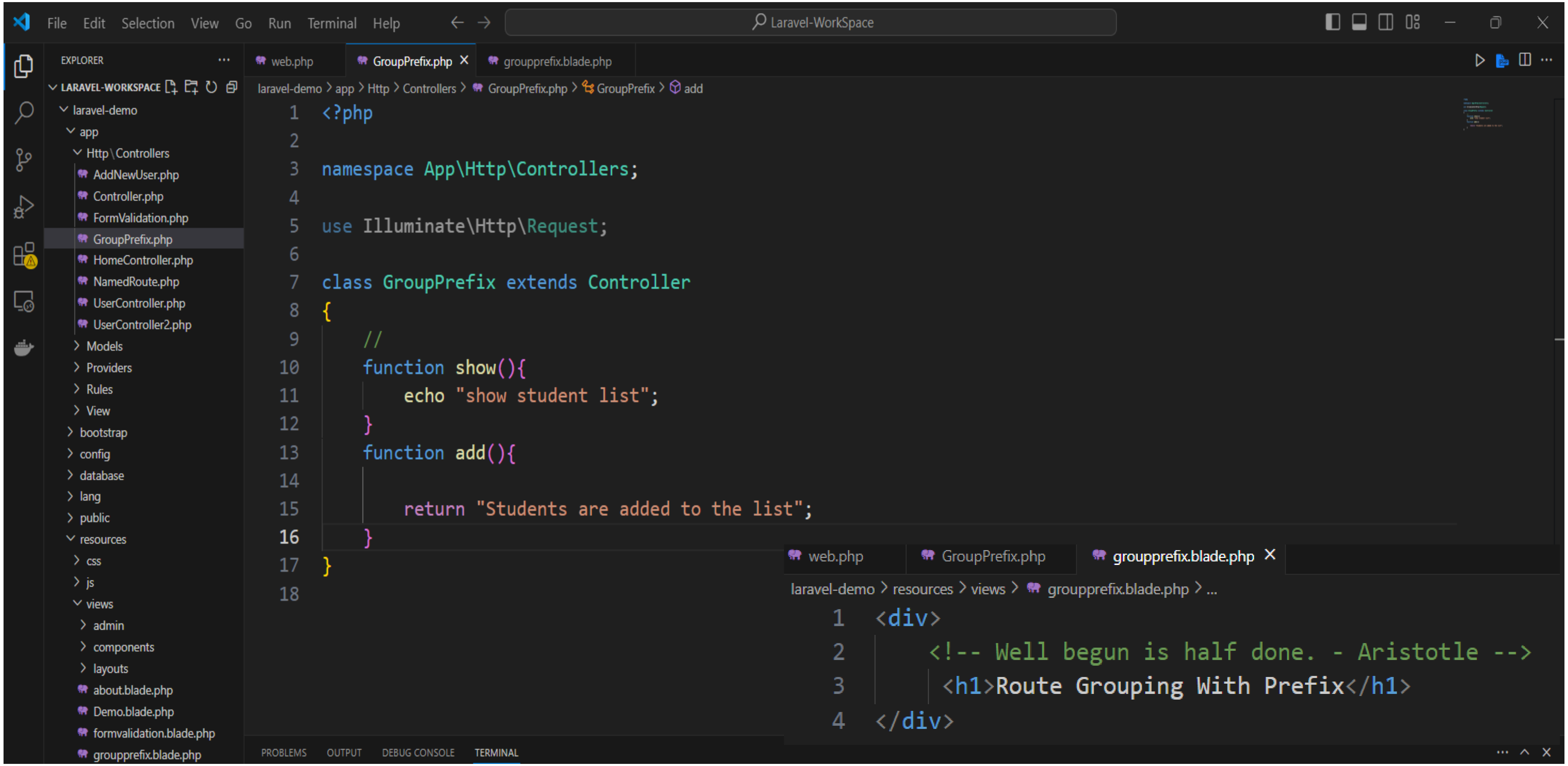
- The routes `/admin/dashboard`, `/admin/users`, and `/admin/settings` will be created.
- Each route has a name prefixed with `admin.` (e.g., `admin.dashboard`), making it easier to manage within the application.

Example



```
yitayew@DESKTOP-NPGGFO4 MINGW64 ~/Desktop/Laravel-WorkSpace/laravel-demo
$ php artisan make:view groupprefix

   INFO  View [C:\Users\yitayew\Desktop\Laravel-WorkSpace\laravel-demo\resources\views\groupprefix.blade.php] created successfully.


yitayew@DESKTOP-NPGGFO4 MINGW64 ~/Desktop/Laravel-WorkSpace/laravel-demo
$ php artisan make:controller GroupPrefix

   INFO  Controller [C:\Users\yitayew\Desktop\Laravel-WorkSpace\laravel-demo\app\Http\Controllers\GroupPrefix.php] created successfully.


yitayew@DESKTOP-NPGGFO4 MINGW64 ~/Desktop/Laravel-WorkSpace/laravel-demo
$ 
```

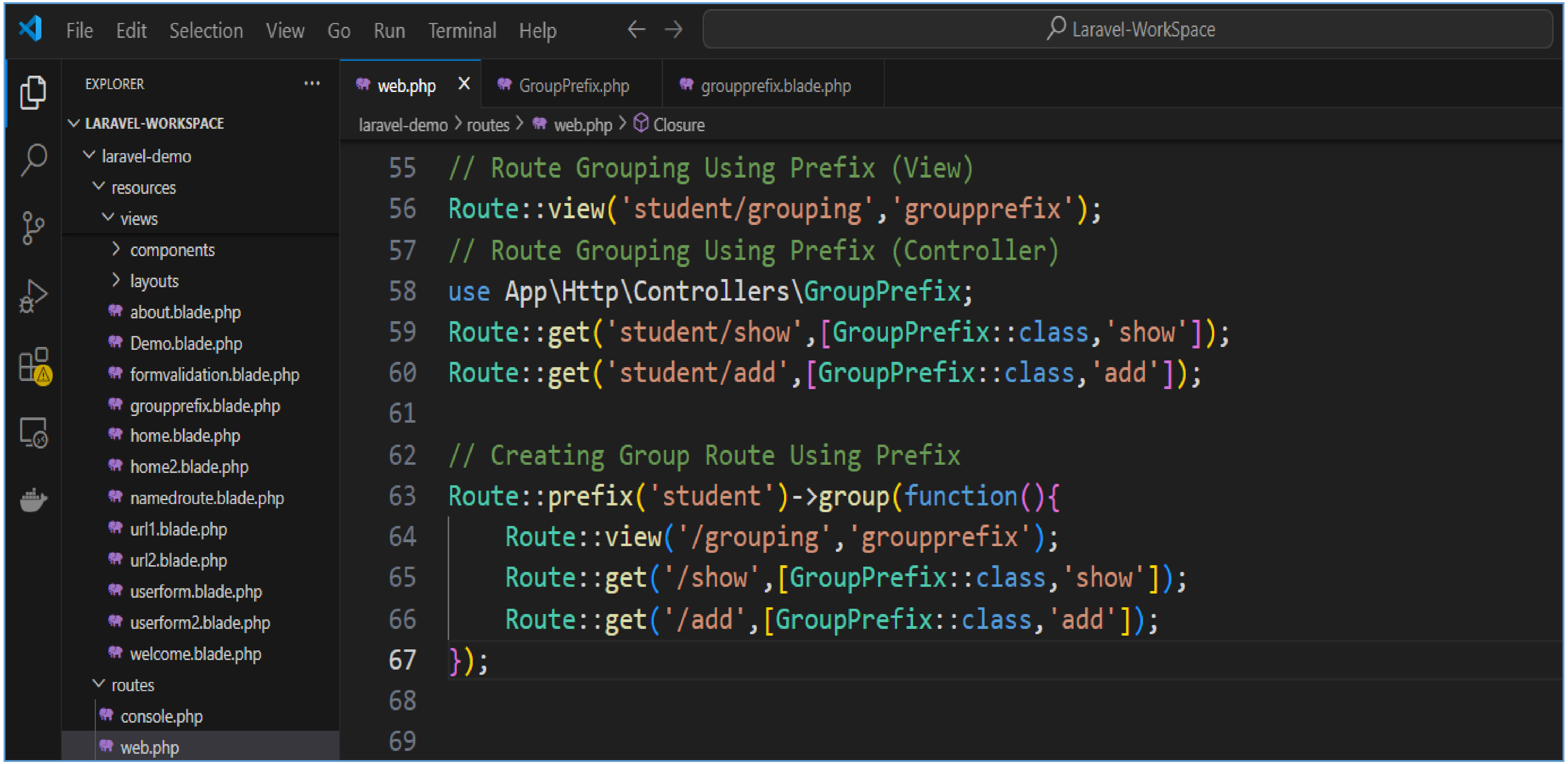
Cont. ...



```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class GroupPrefix extends Controller
{
    //
    function show(){
        echo "show student list";
    }
    function add(){

        return "Students are added to the list";
    }
}

```

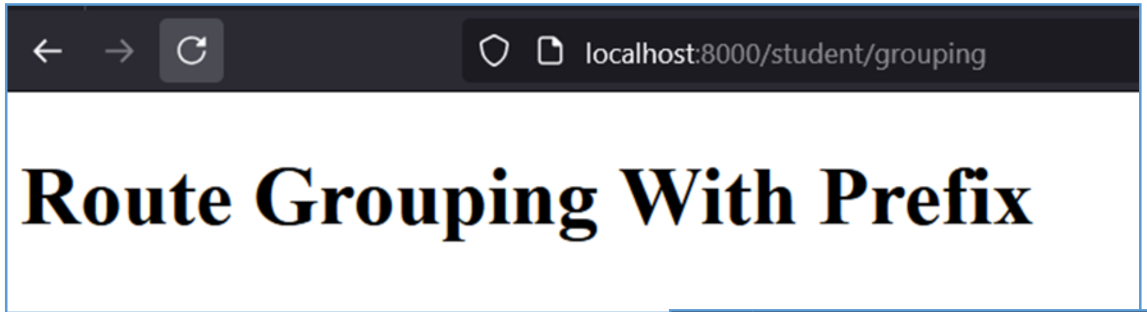
```html
<div>
    <!-- Well begun is half done. - Aristotle -->
    <h1>Route Grouping With Prefix</h1>
</div>
```

Cont. ...

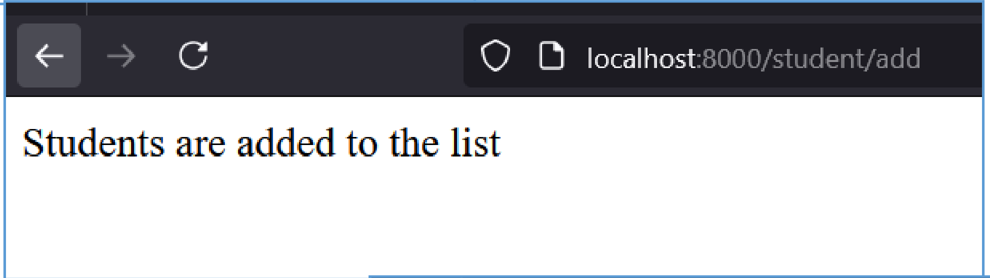


```php
// Route Grouping Using Prefix (View)
Route::view('student/grouping','groupprefix');
// Route Grouping Using Prefix (Controller)
use App\Http\Controllers\GroupPrefix;
Route::get('student/show',[GroupPrefix::class,'show']);
Route::get('student/add',[GroupPrefix::class,'add']);

// Creating Group Route Using Prefix
Route::prefix('student')->group(function(){
    Route::view('/grouping','groupprefix');
    Route::get('/show',[GroupPrefix::class,'show']);
    Route::get('/add',[GroupPrefix::class,'add']);
});


```

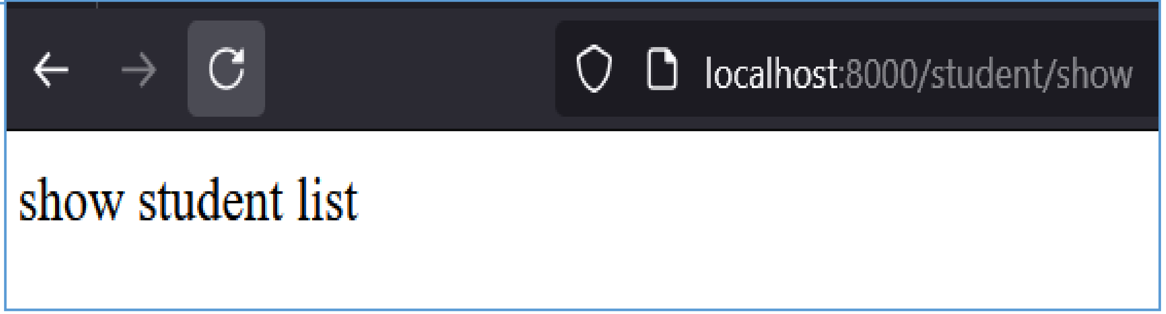
Cont. …



Route Grouping With Prefix



Students are added to the list



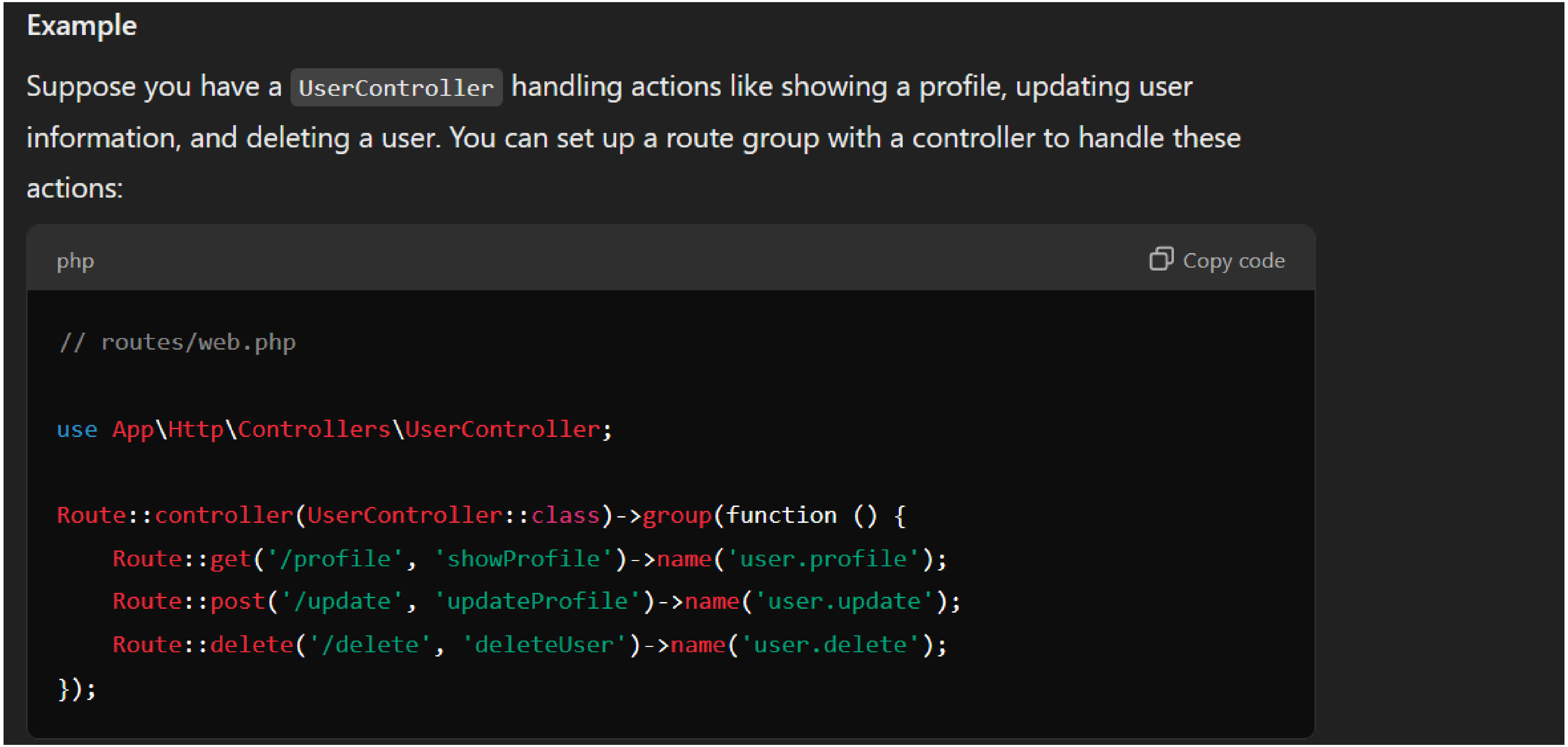
show student list

Route Group With Controller

- In Laravel, you can create route groups with a common controller to streamline routing for multiple methods within the same controller. This is especially useful when multiple actions in a controller share a common path or prefix.

❖**Defining a Route Group with a Controller:** To define a route group with a controller, use the controller method within a route group. This method assigns the specified controller to all routes in the group, simplifying the code and reducing redundancy.

Example

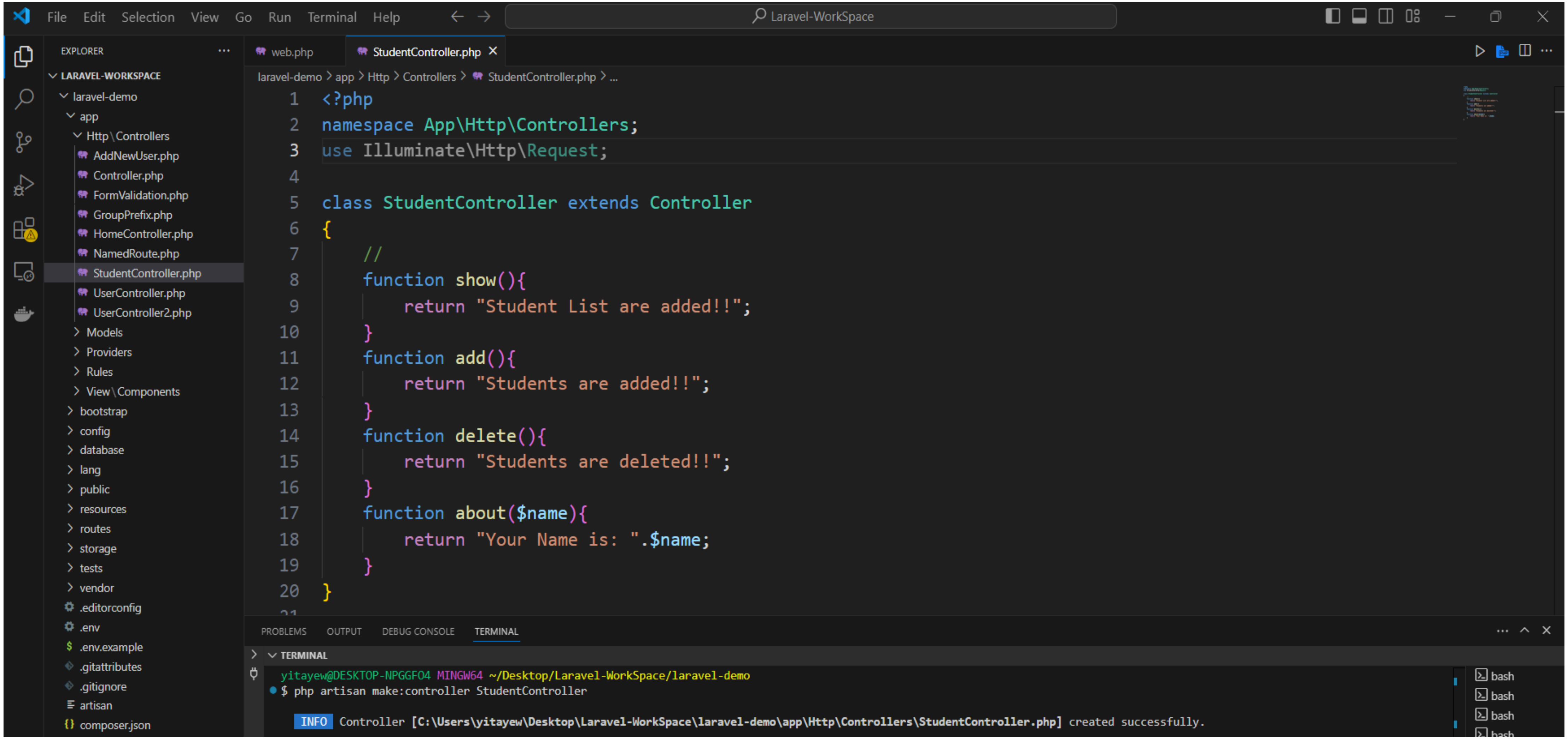


Example

Suppose you have a `UserController` handling actions like showing a profile, updating user information, and deleting a user. You can set up a route group with a controller to handle these actions:

```php
// routes/web.php

use App\Http\Controllers\UserController;

Route::controller(UserController::class)->group(function () {
    Route::get('/profile', 'showProfile')->name('user.profile');
    Route::post('/update', 'updateProfile')->name('user.update');
    Route::delete('/delete', 'deleteUser')->name('user.delete');
});
```

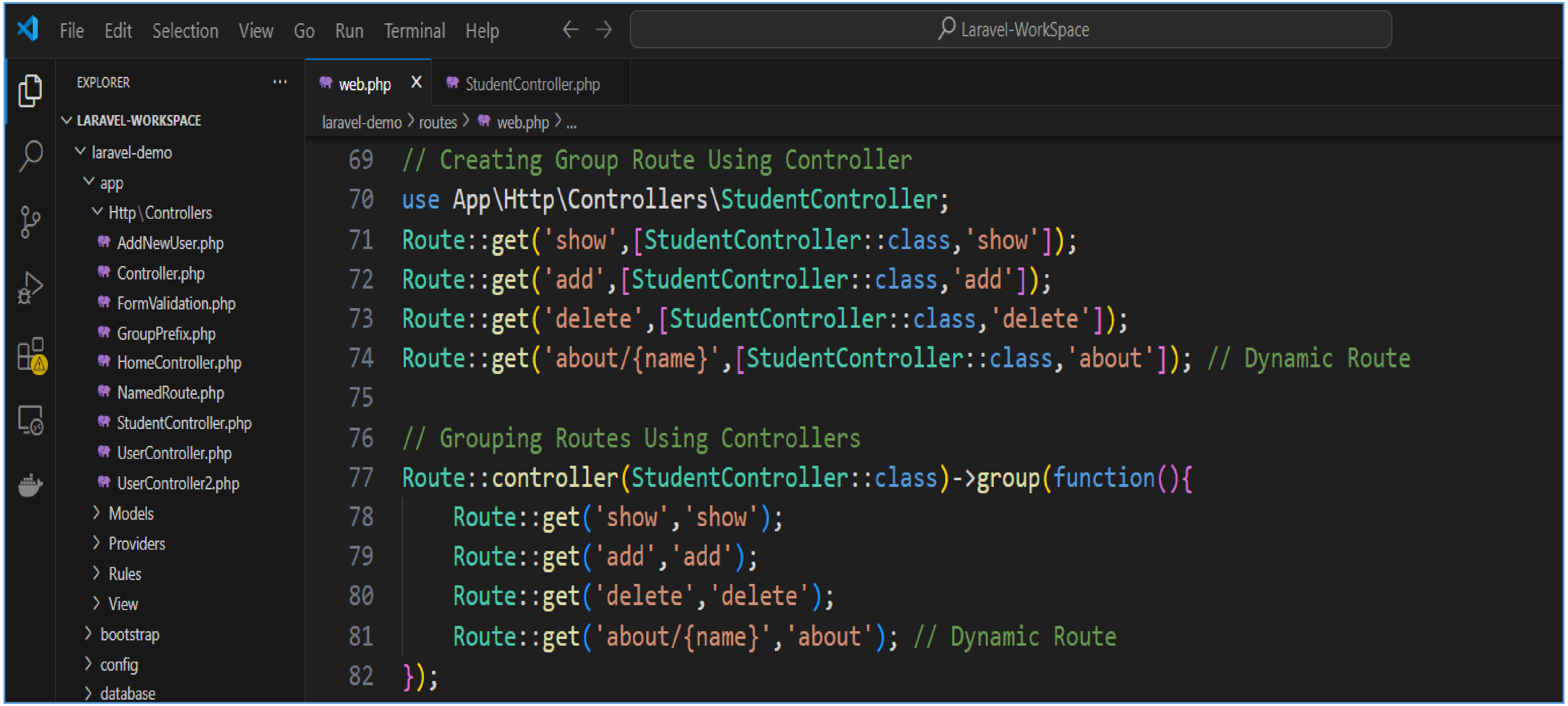
Cont. ...



```php
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;

class StudentController extends Controller
{
    //
    function show(){
        return "Student List are added!!";
    }
    function add(){
        return "Students are added!!";
    }
    function delete(){
        return "Students are deleted!!";
    }
    function about($name){
        return "Your Name is: ".$name;
    }
}
```

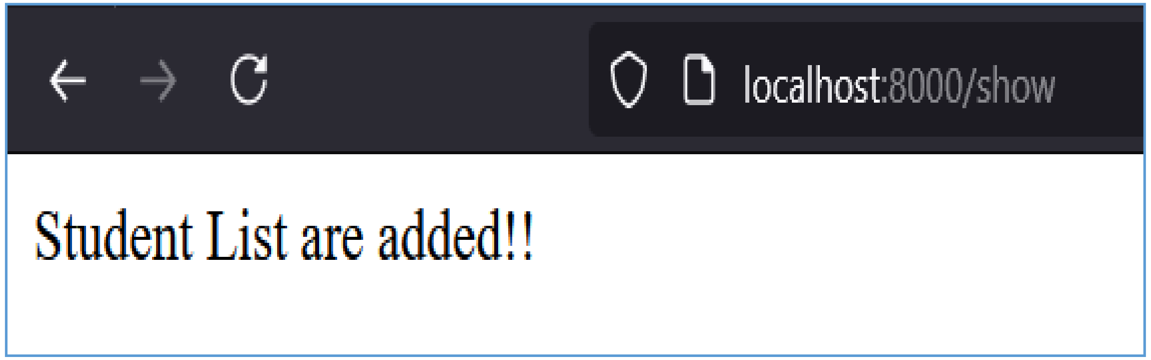
```
yitayew@DESKTOP-NPGGFO4 MINGW64 ~/Desktop/Laravel-WorkSpace/laravel-demo
$ php artisan make:controller StudentController

  INFO  Controller [C:\Users\yitayew\Desktop\Laravel-WorkSpace\laravel-demo\app\Http\Controllers\StudentController.php] created successfully.
```

Cont. …

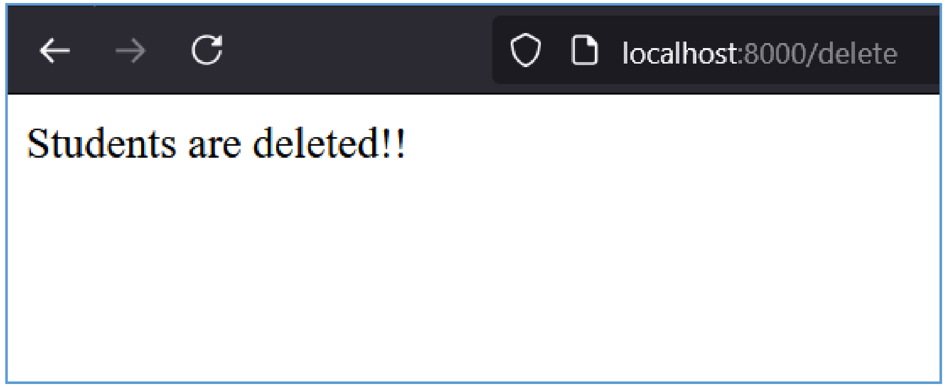


```php
// Creating Group Route Using Controller
use App\Http\Controllers\StudentController;
Route::get('show',[StudentController::class,'show']);
Route::get('add',[StudentController::class,'add']);
Route::get('delete',[StudentController::class,'delete']);
Route::get('about/{name}',[StudentController::class,'about']); // Dynamic Route

// Grouping Routes Using Controllers
Route::controller(StudentController::class)->group(function(){
    Route::get('show','show');
    Route::get('add','add');
    Route::get('delete','delete');
    Route::get('about/{name}','about'); // Dynamic Route
});
```

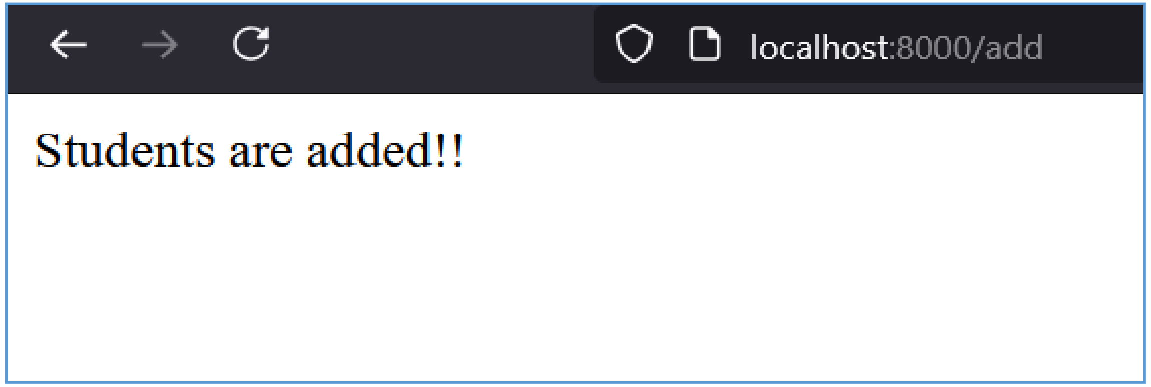
Cont. ...



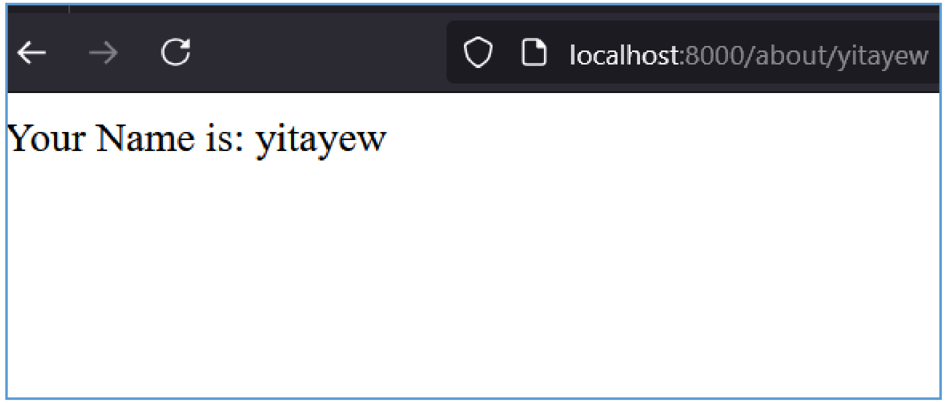
Student List are added!!



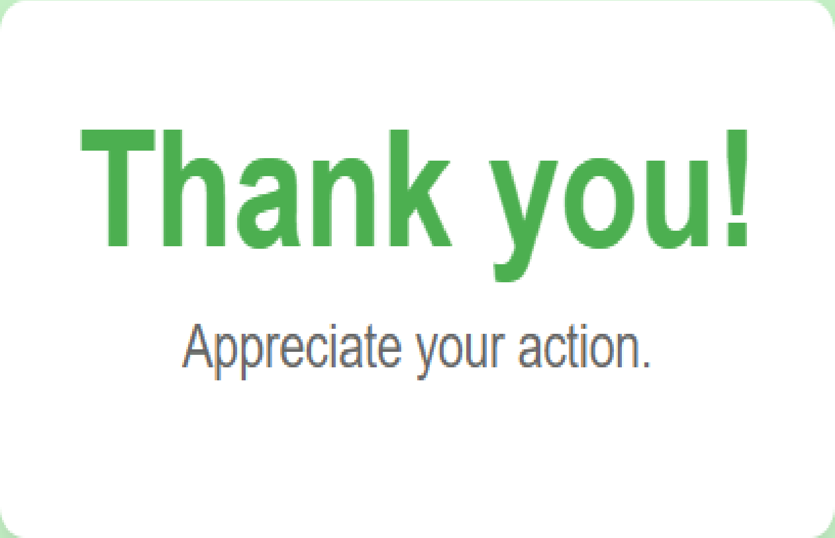
Students are deleted!!



Students are added!!



Your Name is: yitayew



Thank you!

Appreciate your action.