What is Composer in PHP?

If you've been coding in PHP for some time, you'll be aware of how **PHP libraries can help save work** and make code **reusable**. In the past, it was **harder to add libraries to PHP**.

That's where a dependency manager like **Composer comes in**. In fact, before Composer, there was a popular tool called **PEAR which was used to manage PHP extensions and libraries. But it had its own limitations**, which Composer was created to address.

Composer is a **dependency manager for PHP.**

You can simply run **composer update** to get the **latest compatible packages**.

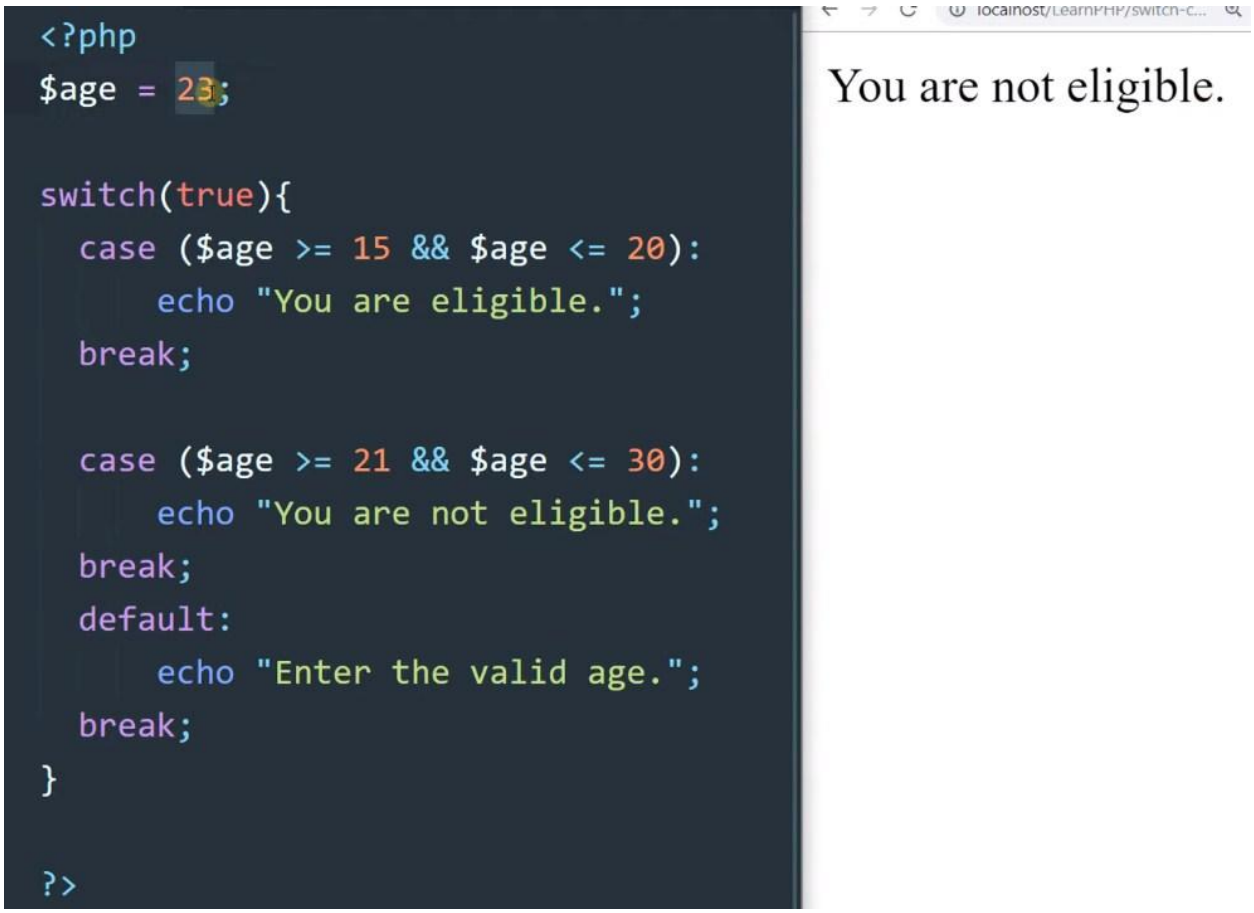
Ex

❖ To ensure that the composer is successfully installed:

$ composer

❖ To install the package:

$ composer require phpmailer


<?php
$age = 23;
switch(true){
case ($age >= 15 && $age <= 20):
echo "You are eligible.";
break;
case ($age >= 21 && $age <= 30):
echo "You are not eligible.";
break;
default:
echo "Enter the valid age.";
break;
}
?>
localhost/LearnPHP/switch-c...
You are not eligible.

PHP Switch Example-

```php
<?php
$num=20;
switch($num){
case 10:
echo("number is equals to 10");
break;
case 20:
echo("number is equal to 20");
break;
case 30:
echo("number is equal to 30");
break;
default:
echo("number is not equal to 10, 20 or 30");
}
?>
```

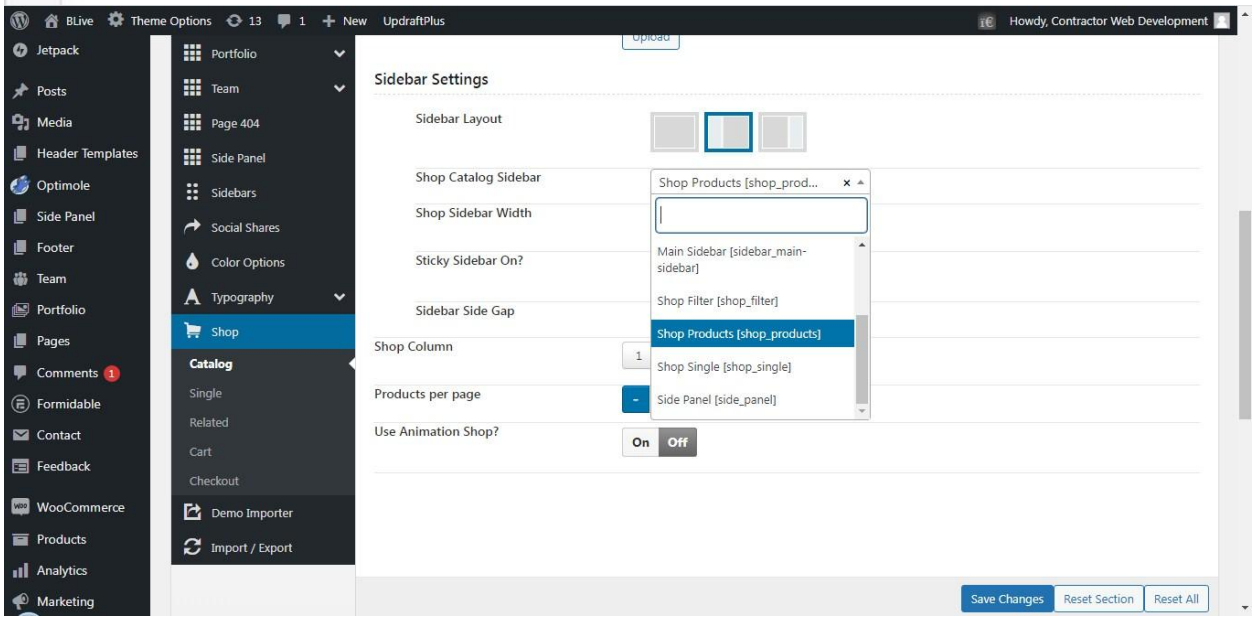
```php
<?php
$day = "Monday";

switch ($day) {
    case "Monday":
        echo "Today is Monday";
        break;
    case "Tuesday":
        echo "Today is Tuesday";
        break;
    case "Wednesday":
        echo "Today is Wednesday";
        break;
    case "Thursday":
        echo "Today is Thursday";
        break;
    case "Friday":
        echo "Today is Friday";
        break;
    case "Saturday":
        echo "Today is Saturday";
        break;
    case "Sunday":
        echo "Today is Sunday";
        break;
    default:
        echo "Not a valid day";
}
?>
```

```php
<?php
$score = 85;

switch (true) {
    case ($score >= 90):
        echo "You got an A";
        break;
    case ($score >= 80 && $score < 90):
        echo "You got a B";
        break;
    case ($score >= 70 && $score < 80):
        echo "You got a C";
        break;
    case ($score >= 60 && $score < 70):
        echo "You got a D";
        break;
    default:
        echo "You got an F";
}
?>
```

```php
add_action('woocommerce_single_product_summary', 'fn_custom_block', 100);
function fn_custom_block() {
    echo '<div class="custom-block">';
    echo do_shortcode('[contact-form-7 id="4772" title="Inquiry Form"]');
    echo '</div>';
}
```



How to code in MVC Framework ?

Controller (controllers/my_controller.php)

```
function total_frogs(){
  $this->load->model("frogs");
  $number_of_frogs= $this->frogs->count_frogs();
  $data['froggies'] = $number_of_frogs;
  $this->load->view("frog_view",$data);
}
```

Model (models/frogs.php)

```
function count_frogs(){
  $this->db->where("type","frog");
  $this->db->from("animals");
  $query = $this->db->get();
  return $query->numb_rows();
}
```

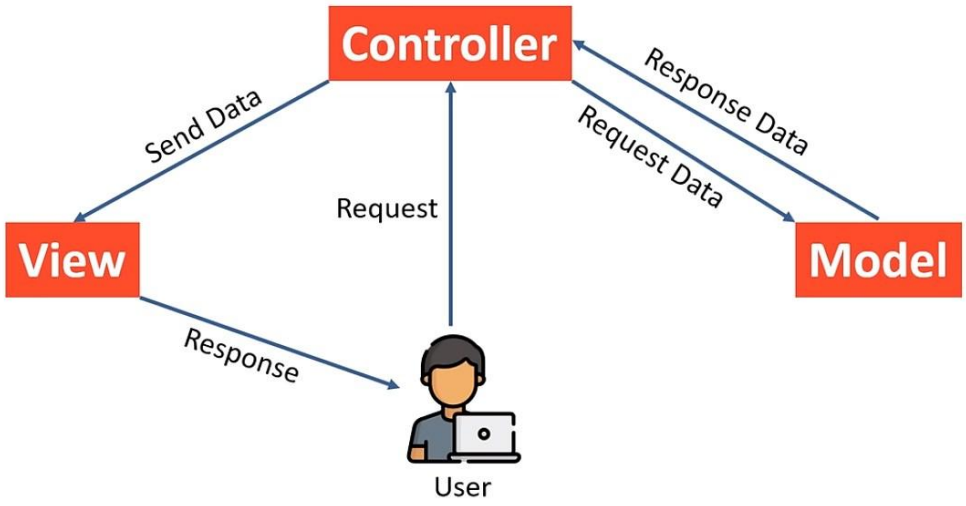
View (views/frog_view.php)

```
<html>
  <body>
    <h1>You've <?=$name;?> frogs in list</h1>
  </body>
</html>
```


Benefits of MVC Framework

- Organized Code
- Independent Block
- Reduces the complexity of Web Applications
- Easy to maintain
- Easy to modify
- Code reusability
- Improved collaboration
- Platform independence

MVC Pattern Work Flow



Popular MVC Frameworks

PHP MVC Framework

- Laravel
- Symfony
- CodeIgniter
- Yii
- CakePHP
- Zend Framework

Other Programming MVC Framework

- Django & Flask (Python)
- Ruby on Rails (Ruby)
- Express.js (JavaScript/Node.js)
- ASP.NET MVC (ASP.net Core)

What is Framework ?

Programming frameworks are sets of pre-written code and libraries that provide a foundation for developing software applications.

Pre-written Code & Library

- Tools
- Components
- Modules

Examples

- Database Component
- Caching
- Pagination
- Session management
- Form Handling
- Security mechanisms
- User authentication
- APIs
- Payment Gateways

Benefits

- Code organization
- Reusability
- Standardization
- Testing & debugging support
- Community and support

Benefits of Laravel Framework

- Open source
- Elegant syntax
- MVC architecture
- Database migration and ORM
- Robust routing system
- Command-line Interface (Composer)
- Powerful template engine (Blade Template)
- Authentication and authorization
- Testing and debugging
- Security (XSS, CSRF, SQL injection)
- Scalability and performance(Redis and Memcached)
- Robust ecosystem and community

What we will learn in this Laravel Series

- Artisan CLI
- Routing
- Views
- Blade Template
- Controllers
- Model
- Database
- Eloquent ORM
- Migration
- Middleware
- Form Validation
- Authentication
- Handling File Upload
- APIs Validation
- CRUD Project
- News Blog Project

Laravel Route

http://localhost/about

↓

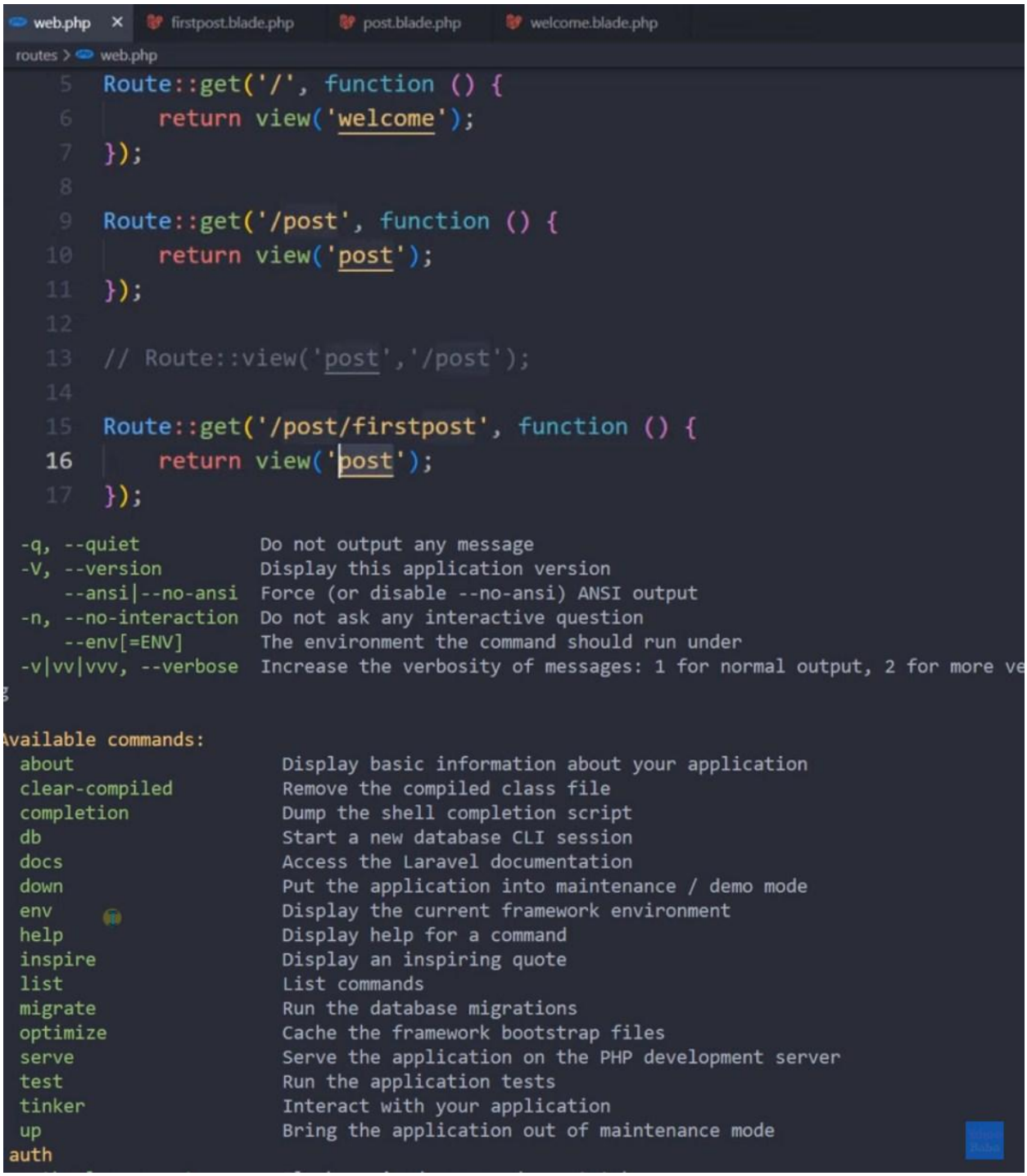
Route Name

↓

hello.php (View File / Controller File)

routes/web.php

```
Route::get('/about', function () {
    return view('hello');
});
```


web.php
firstpost.blade.php
post.blade.php
welcome.blade.php
routes > web.php
Route::get('/', function () {
return view('welcome');
});
Route::get('/post', function () {
return view('post');
});
// Route::view('post','/post');
Route::get('/post/firstpost', function () {
return view('post');
});
-q, --quiet Do not output any message
-V, --version Display this application version
--ansi|--no-ansi Force (or disable --no-ansi) ANSI output
-n, --no-interaction Do not ask any interactive question
--env[=ENV] The environment the command should run under
-v|vv|vvv, --verbose Increase the verbosity of messages: 1 for normal output, 2 for more ve
Available commands:
about Display basic information about your application
clear-compiled Remove the compiled class file
completion Dump the shell completion script
db Start a new database CLI session
docs Access the Laravel documentation
down Put the application into maintenance / demo mode
env Display the current framework environment
help Display help for a command
inspire Display an inspiring quote
list List commands
migrate Run the database migrations
optimize Cache the framework bootstrap files
serve Serve the application on the PHP development server
test Run the application tests
tinker Interact with your application
up Bring the application out of maintenance mode
auth

```
 queue:retry-batch       Retry the failed jobs for a batch
 queue:table             Create a migration for the queue jobs database table
 queue:work              Start processing jobs on the queue as a daemon
route
 route:cache             Create a route cache file for faster route registration
 route:clear             Remove the route cache file
 route:list              List all registered routes
sail
 sail:add                Add a service to an existing Sail installation
 sail:install            Install Laravel Sail's default Docker Compose file
 sail:publish            Publish the Laravel Sail Docker files
sanctum
 sanctum:prune-expired   Prune tokens expired for more than specified number of hours
schedule
 schedule:clear-cache    Delete the cached mutex files created by scheduler
 schedule:list           List all scheduled tasks
 schedule:run            Run the scheduled commands
 schedule:test           Run a scheduled command
 schedule:work           Start the schedule worker
schema
 schema:dump             Dump the given database schema
session
 session:table           Create a migration for the session database table
storage
 storage:link            Create the symbolic links configured for the application
stub
 stub:publish            Publish all stubs that are available for customization
vendor
 vendor:publish          Publish any publishable assets from vendor packages
view
 view:cache              Compile all of the application's Blade templates
```

Laravel Route Parameters

http://localhost/post/10

http://localhost/post/yahoobaba

http://localhost/post/news10

http://localhost/post/@news10

```
Route::get('/post/{id}', function (string $id) {
   return 'User ' . $id;
});
```

web.php

```php
<?php

use Illuminate\Support\Facades\Route;

Route::get('/', function () {
    return view('welcome');
});

// Route::get('/post', function () {
//     return view('post');
// });

Route::view('/post','post');

Route::get('/post/firstpost', function () {
    return view('firstpost');
});
```

web.php

```php
<?php

use Illuminate\Support\Facades\Route;

Route::get('/', function () {
    return view('welcome');
});

Route::get('/post/{id?}', function (string $id = null) {
    if($id){
        return "<h1>Post ID : ". $id ."</h1>";
    }else{
        return "<h1>No ID Found</h1>";
    }


});
```

Laravel Route Constraints

http://localhost/post/10 → whereNumber('id')

http://localhost/post/yahoobaba → whereAlpha('name')

http://localhost/post/news10 → whereAlphaNumeric('name')

http://localhost/post/song → whereIn('category', ['movie', 'song'])

http://localhost/post/@10 → where('id', '[@0-9]+')

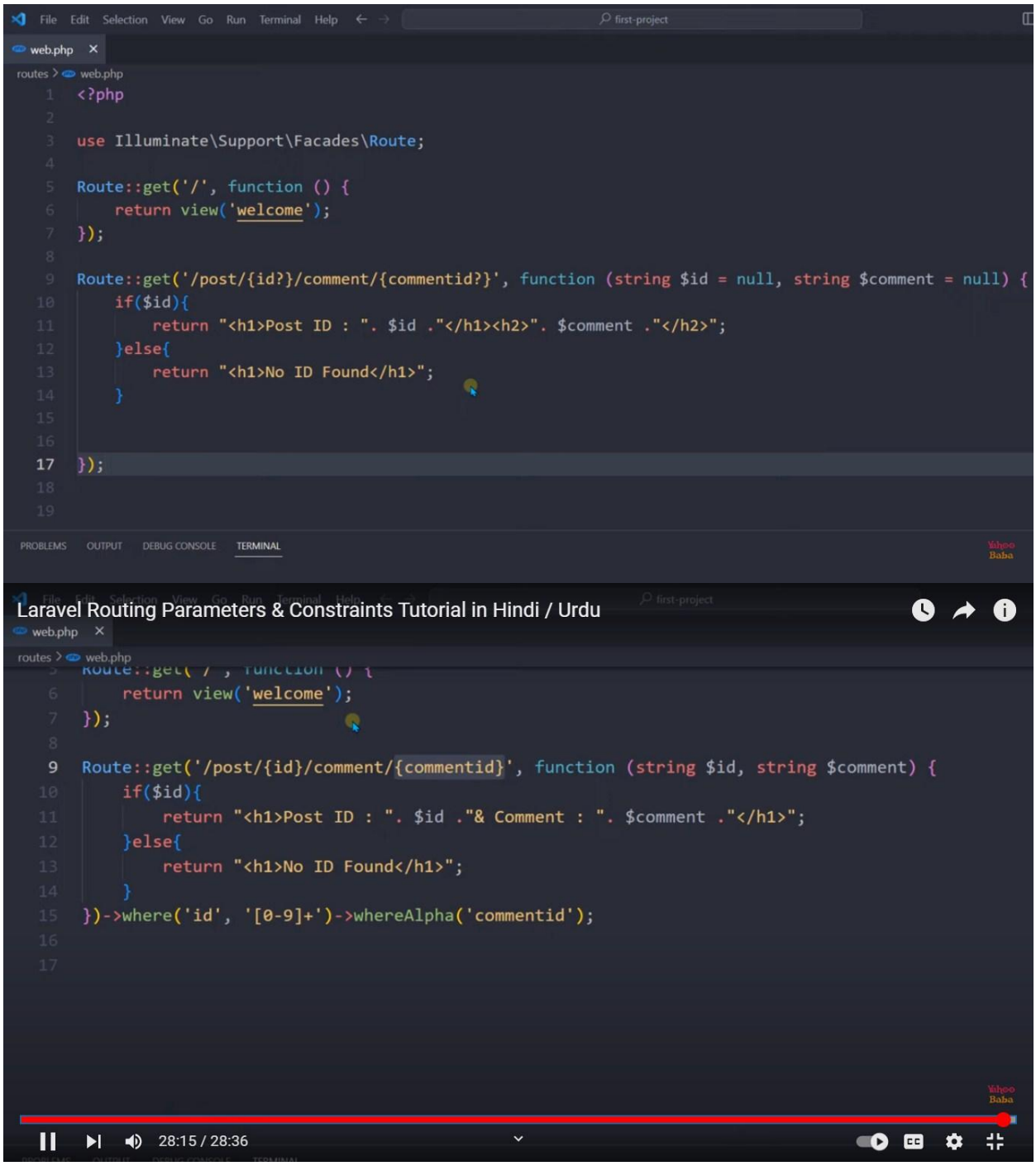
↓

Regular Expressions

Laravel Route Constraints

```
Route::get('/post/{id}', function (string $id) {
    return 'User ' . $id;
})->whereNumber('id');
```

Yahoo Baba www.yahoobaba.net


File Edit Selection View Go Run Terminal Help
first-project
web.php
routes > web.php
<?php
use Illuminate\Support\Facades\Route;
Route::get('/', function () {
return view('welcome');
});
Route::get('/post/{id?}/comment/{commentid?}', function (string $id = null, string $comment = null) {
if($id){
return "<h1>Post ID : ". $id ."</h1><h2>". $comment ."</h2>";
}else{
return "<h1>No ID Found</h1>";
}
});
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Laravel Routing Parameters & Constraints Tutorial in Hindi / Urdu
web.php
routes > web.php
return view('welcome');
});
Route::get('/post/{id}/comment/{commentid}', function (string $id, string $comment) {
if($id){
return "<h1>Post ID : ". $id ."& Comment : ". $comment ."</h1>";
}else{
return "<h1>No ID Found</h1>";
}
})->where('id', '[0-9]+')->whereAlpha('commentid');
28:15 / 28:36

Laravel Named Routes

http://localhost/page/about

```
Route::get('/page/about-us', function () {
    return 'About Page';
})->name('about');
```

first.blade.php

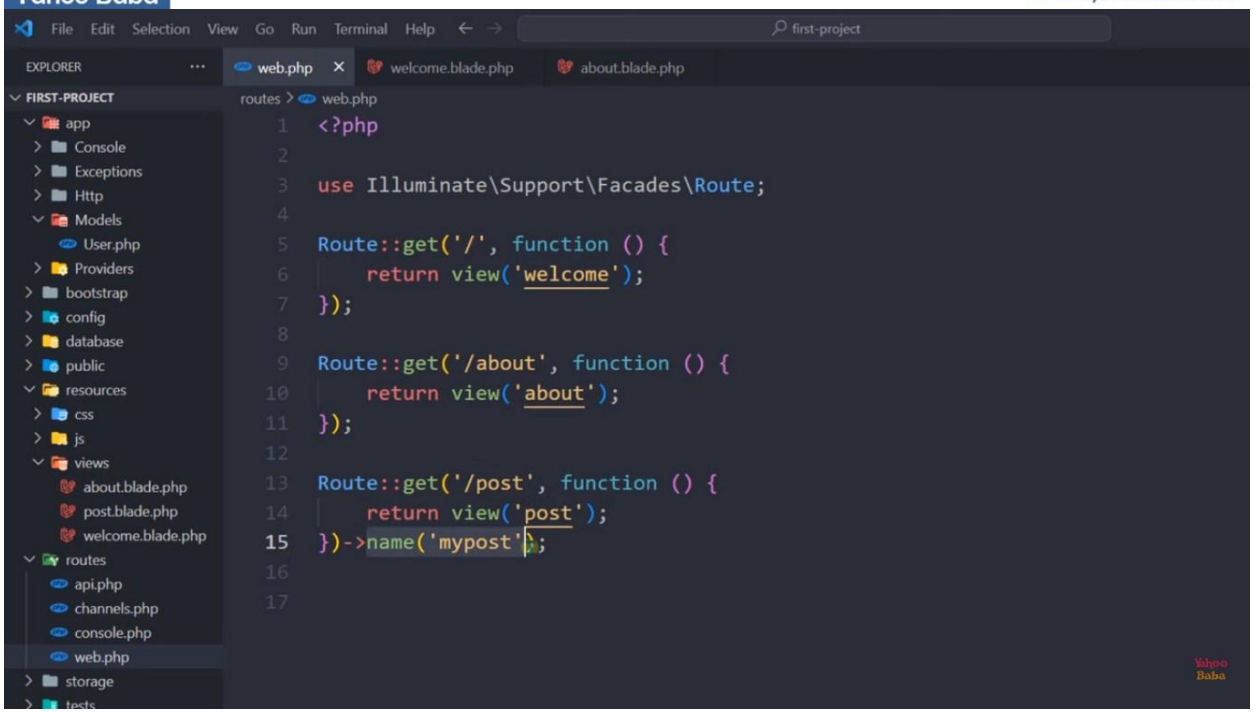
```
<a href='{{ route('about') }}'>About</a>
```

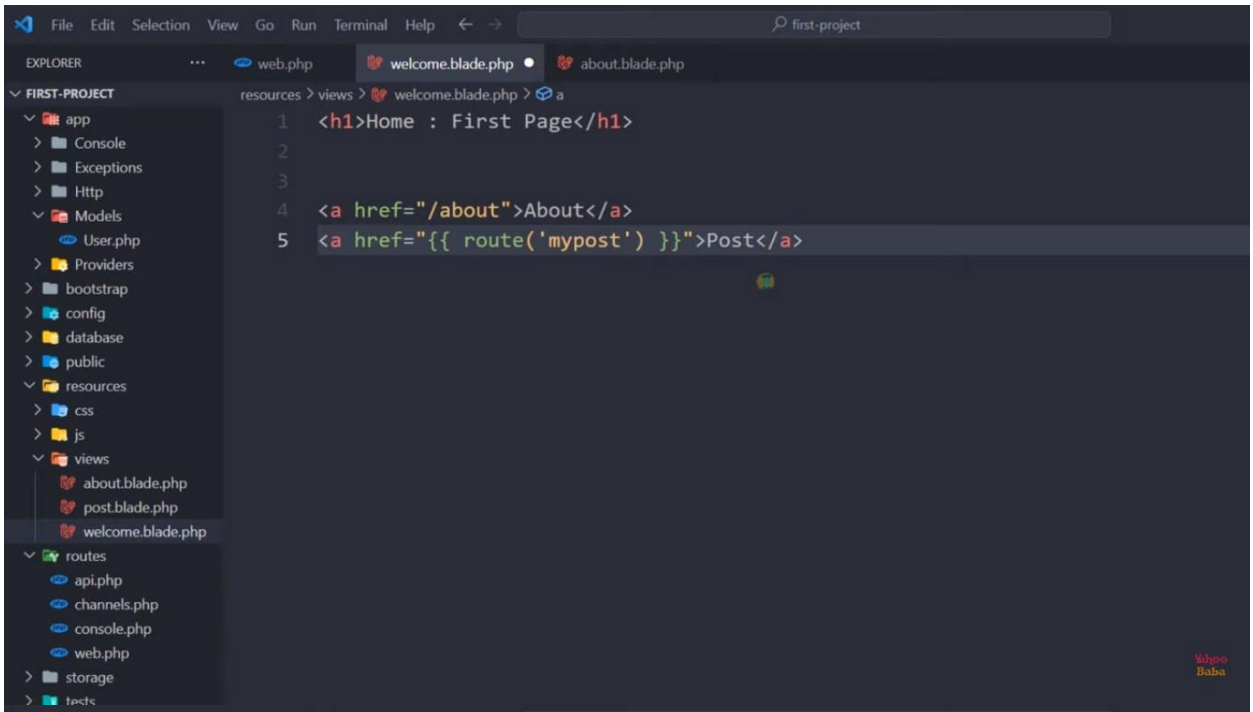
second.blade.php

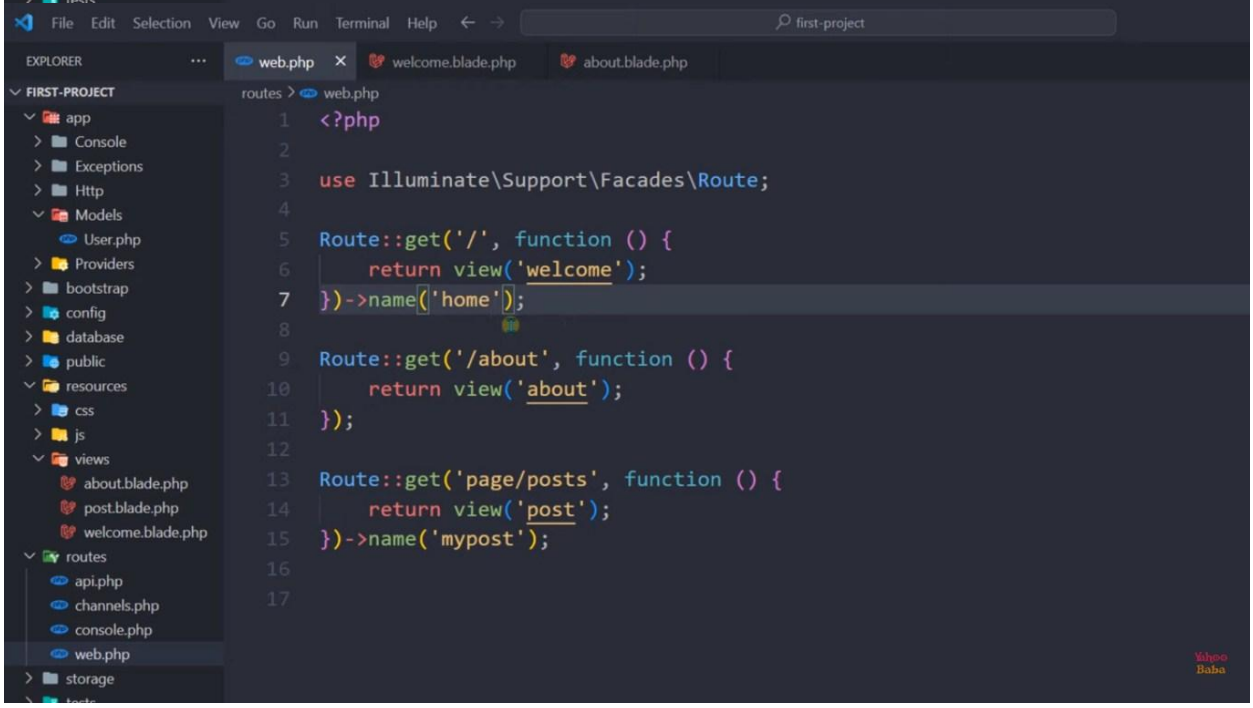
```
<a href='{{ route('about') }}'>About</a>
```

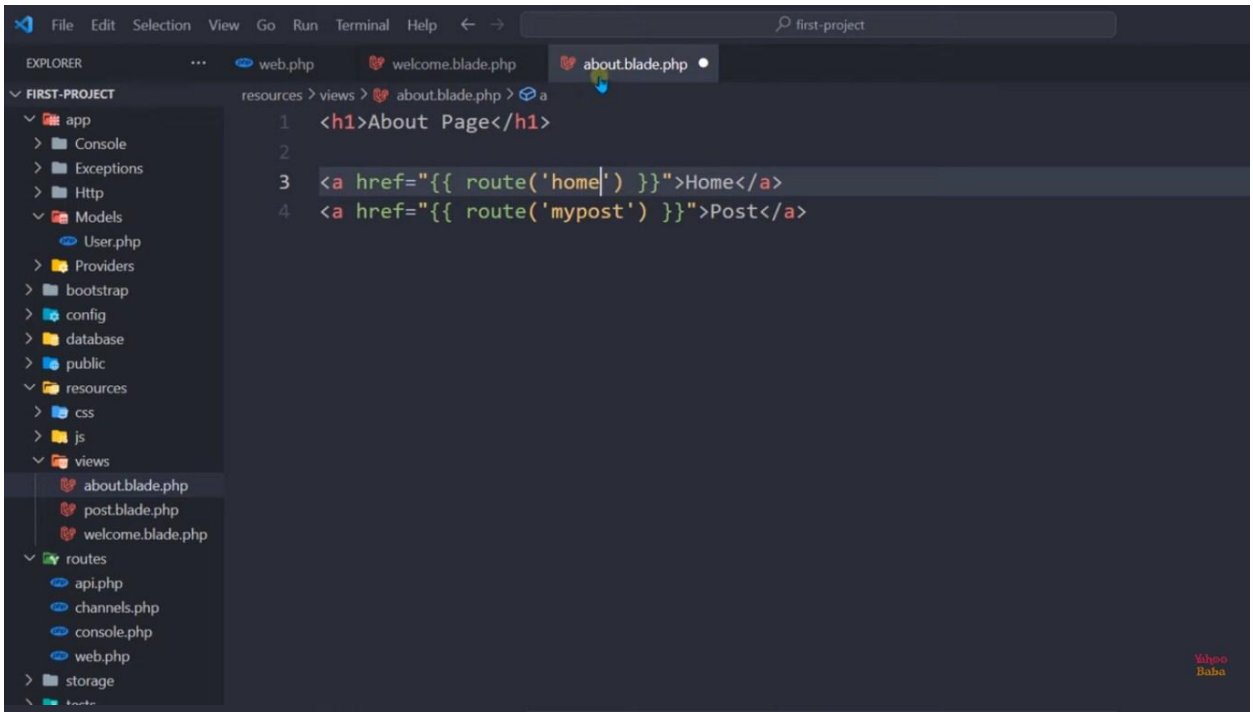
third.blade.php

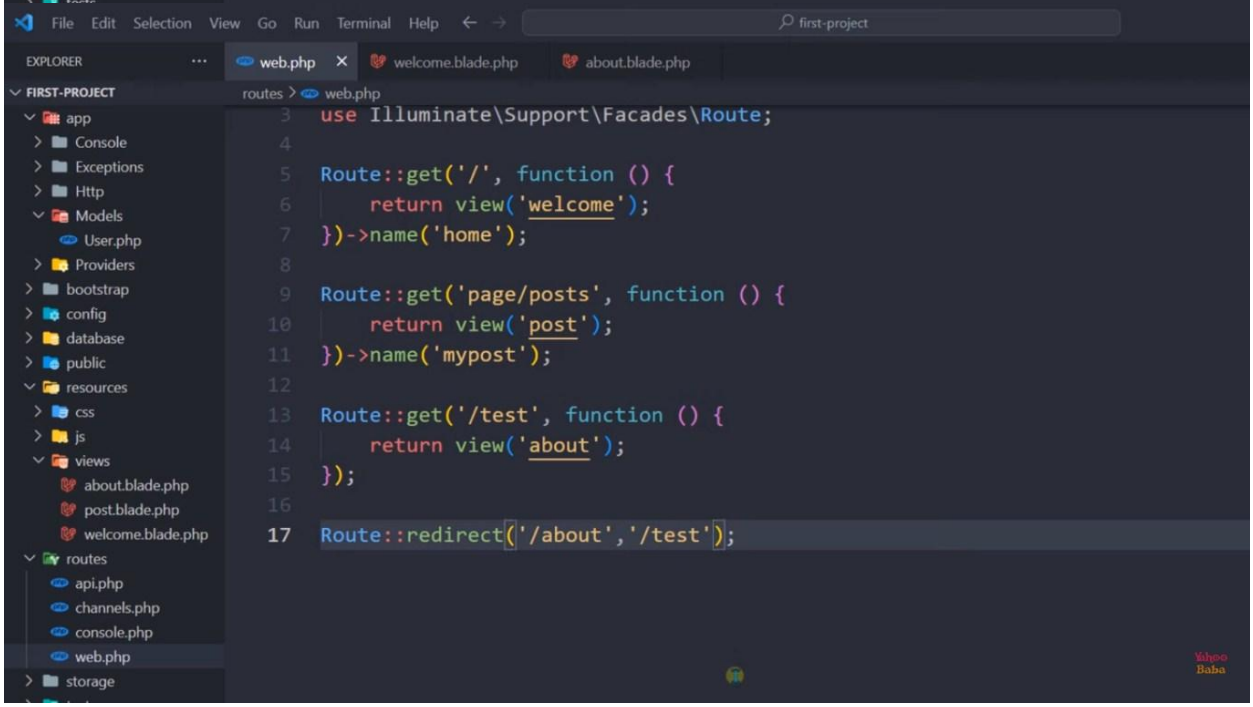
```
<a href='{{ route('about') }}'>About</a>
```


File Edit Selection View Go Run Terminal Help
first-project
EXPLORER
web.php
welcome.blade.php
about.blade.php
FIRST-PROJECT
resources > views > welcome.blade.php > a
app
Console
Exceptions
Http
Models
User.php
Providers
bootstrap
config
database
public
resources
css
js
views
about.blade.php
post.blade.php
welcome.blade.php
routes
api.php
channels.php
console.php
web.php
storage
1 <h1>Home : First Page</h1>
4 About
5 Post
Yahoo Baba


File Edit Selection View Go Run Terminal Help
first-project
EXPLORER
web.php
welcome.blade.php
about.blade.php
FIRST-PROJECT
routes > web.php
app
Console
Exceptions
Http
Models
User.php
Providers
bootstrap
config
database
public
resources
css
js
views
about.blade.php
post.blade.php
welcome.blade.php
routes
api.php
channels.php
console.php
web.php
storage
1 <?php
3 use Illuminate\Support\Facades\Route;
5 Route::get('/', function () {
6 return view('welcome');
7 })->name('home');
9 Route::get('/about', function () {
10 return view('about');
11 });
13 Route::get('page/posts', function () {
14 return view('post');
15 })->name('mypost');
Yahoo Baba

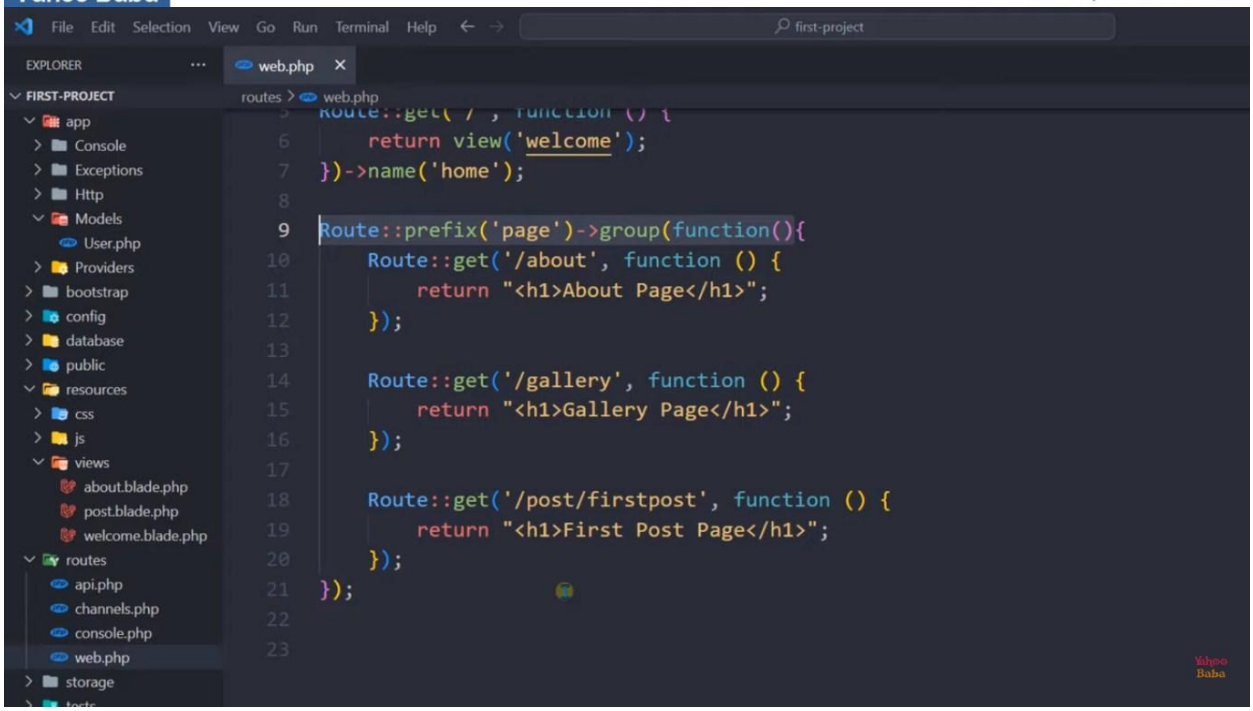

File Edit Selection View Go Run Terminal Help
first-project
EXPLORER
web.php
welcome.blade.php
about.blade.php
FIRST-PROJECT
resources > views > about.blade.php > a
app
Console
Exceptions
Http
Models
User.php
Providers
bootstrap
config
database
public
resources
css
js
views
about.blade.php
post.blade.php
welcome.blade.php
routes
api.php
channels.php
console.php
web.php
storage
1 <h1>About Page</h1>
2
3 Home
4 Post
Yahoo Baba

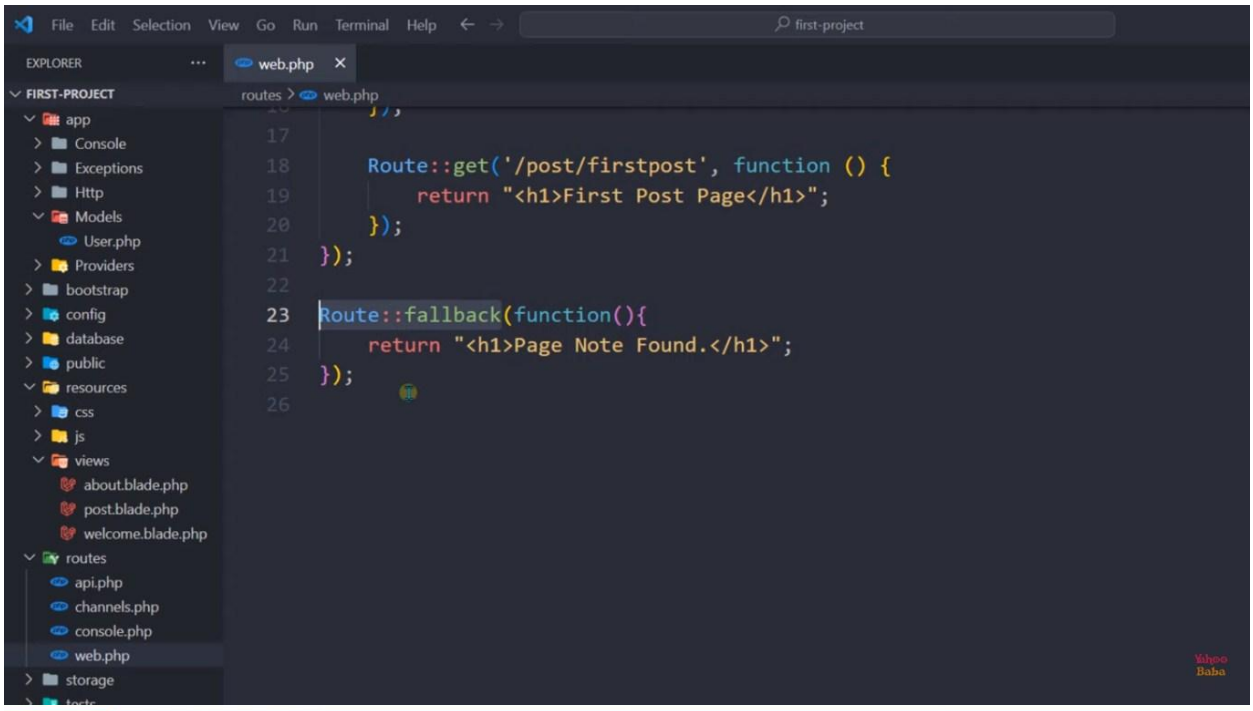

File Edit Selection View Go Run Terminal Help
first-project
EXPLORER
web.php
welcome.blade.php
about.blade.php
FIRST-PROJECT
routes > web.php
app
Console
Exceptions
Http
Models
User.php
Providers
bootstrap
config
database
public
resources
css
js
views
about.blade.php
post.blade.php
welcome.blade.php
routes
api.php
channels.php
console.php
web.php
storage
3 use Illuminate\Support\Facades\Route;
4
5 Route::get('/', function () {
6 return view('welcome');
7 })->name('home');
8
9 Route::get('page/posts', function () {
10 return view('post');
11 })->name('mypost');
12
13 Route::get('/test', function () {
14 return view('about');
15 });
16
17 Route::redirect('/about','/test');
Yahoo Baba

Laravel Route Groups

```
Route::prefix('page')->group(function () {
    Route::get('/post/1', function () {
    });
    Route::get('/about/', function () {
    });
    Route::get('/gallery/', function () {
    });
});
```

→ http://localhost/page/about





What is Blade Template ?

Blade Template

↓

Template Engine Based on PHP

Blade provides a clean and convenient way to create views in Laravel

Benefits : Create Dynamic and Reusable Templates

HTML & PHP

Laravel : Blade Template Syntax

```
<?php
   echo "Hello";
?>
```

```
{{ "Hello" }}
```

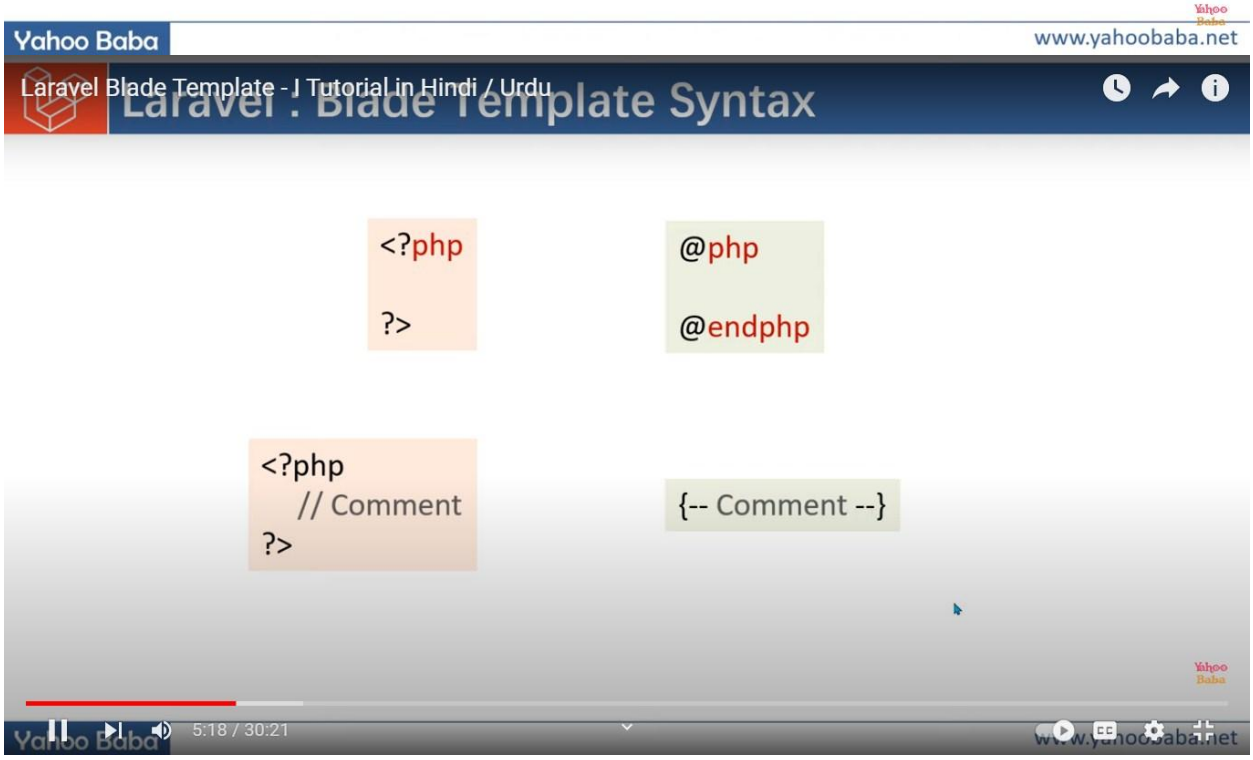
Prevent cross-site scripting (XSS) Attacks

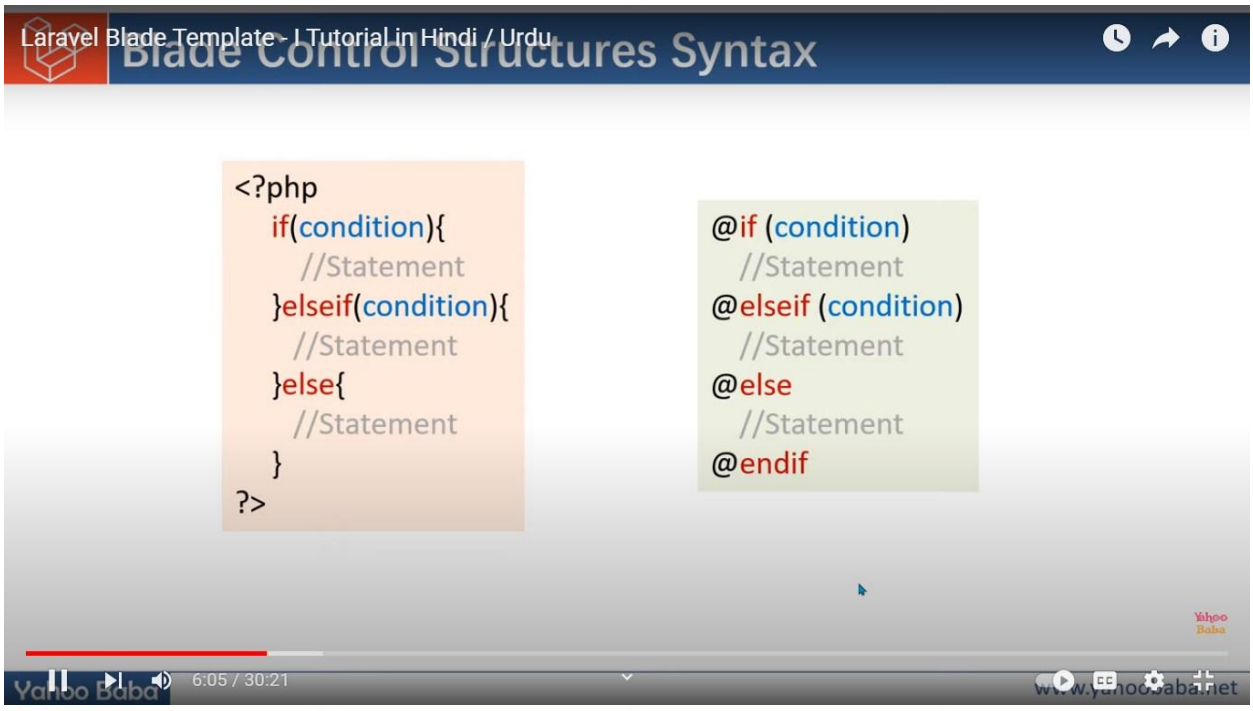
```
<?php
   echo $name;
?>
```

```
{{ $name }}
```

```
<?php
   echo "<h1>Hello</h1>";
?>
```

```
{!! "<h1>Hello</h1>" !!}
```





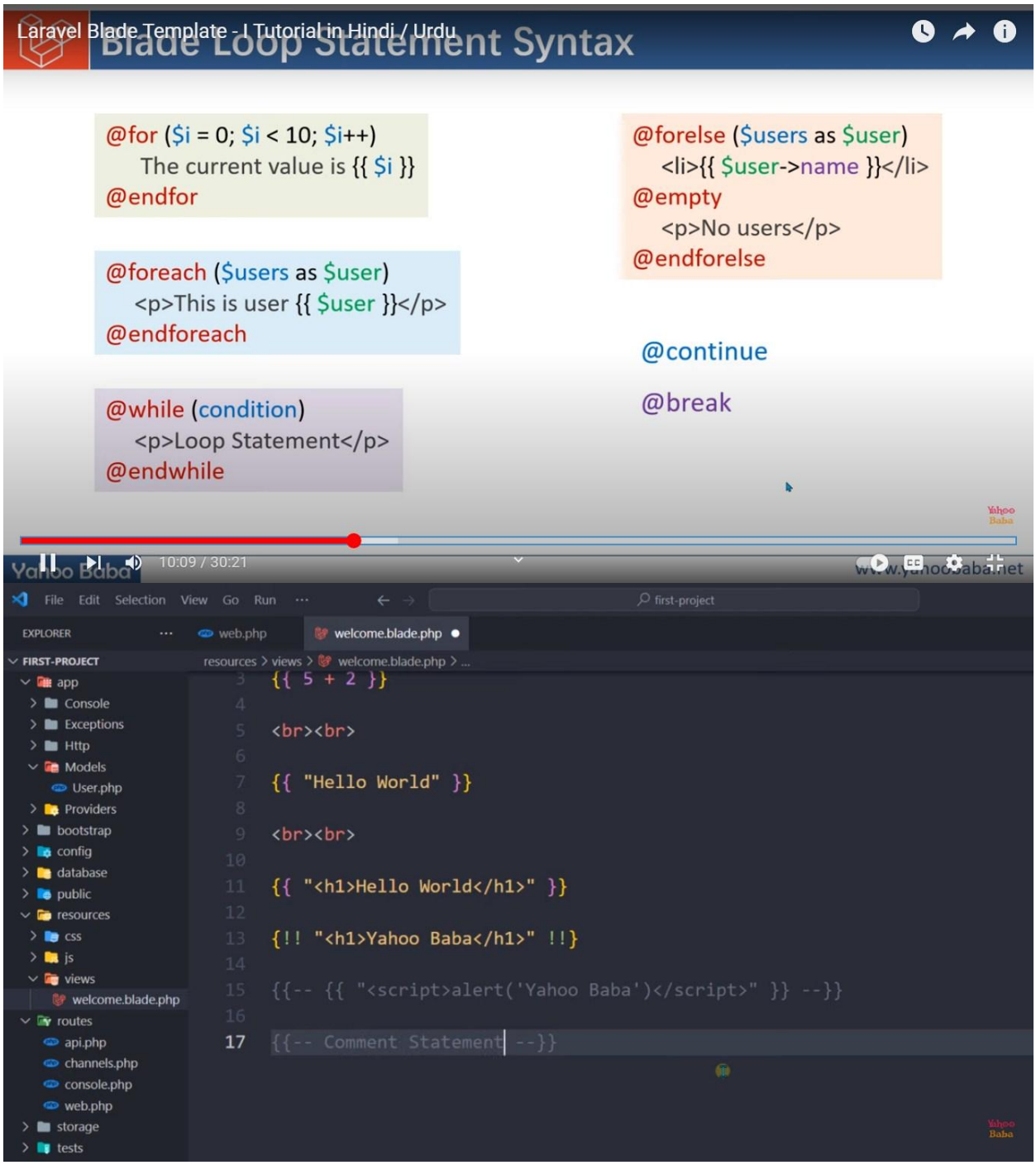
Blade Control Structures Syntax

```
<?php
    if(isset($records)){
        //Statement
    }
?>
```

```
@isset($records)
    // $records is defined and is not null...
@endisset
```

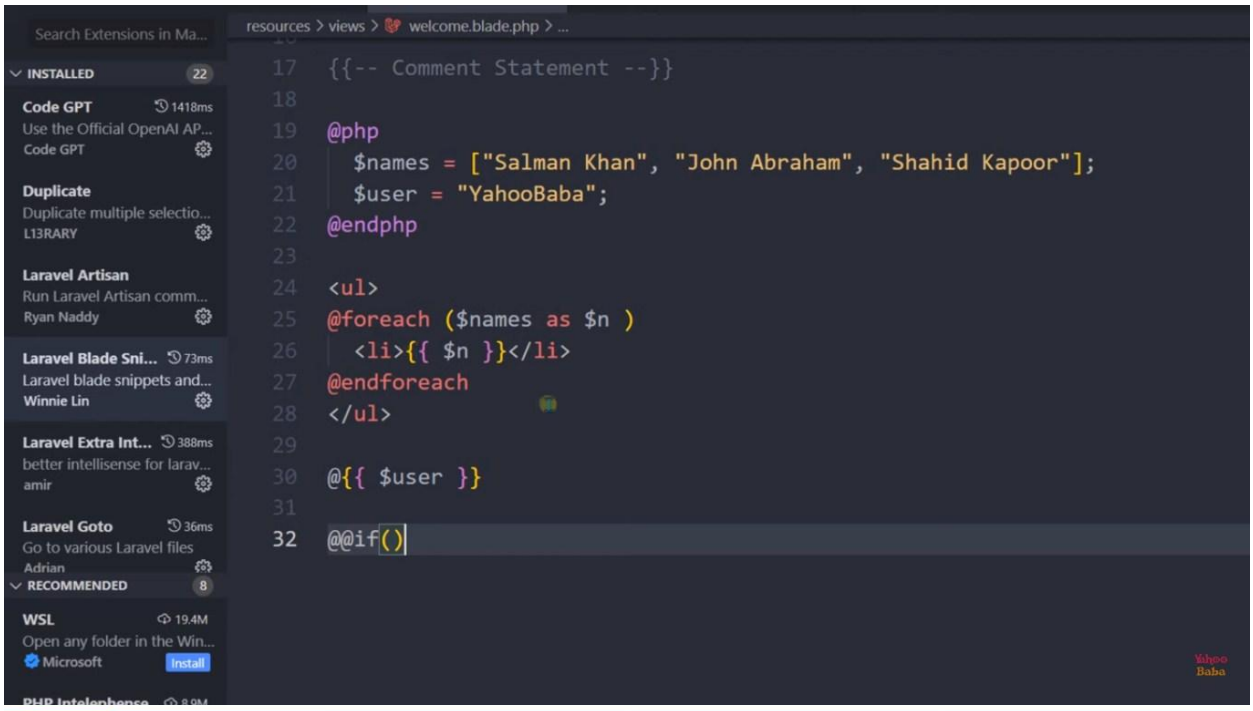
```
<?php
    if(emtpy($records)){
        //Statement
    }
?>
```

```
@empty($records)
    // $records is "empty"...
@endempty
```


Laravel Blade Template - I Tutorial in Hindi / Urdu
Blade Loop Statement Syntax
@for ($i = 0; $i < 10; $i++)
The current value is {{ $i }}
@endfor
@foreach ($users as $user)
This is user {{ $user }}
@endforeach
@while (condition)
Loop Statement
@endwhile
@forelse ($users as $user)
{{ $user->name }}
@empty
No users
@endforelse
@continue
@break
10:09 / 30:21
File Edit Selection View Go Run
first-project
EXPLORER
web.php
welcome.blade.php
FIRST-PROJECT
resources > views > welcome.blade.php > ...
app
Console
Exceptions
Http
Models
User.php
Providers
bootstrap
config
database
public
resources
css
js
views
welcome.blade.php
routes
api.php
channels.php
console.php
web.php
storage
tests
{{ 5 + 2 }}

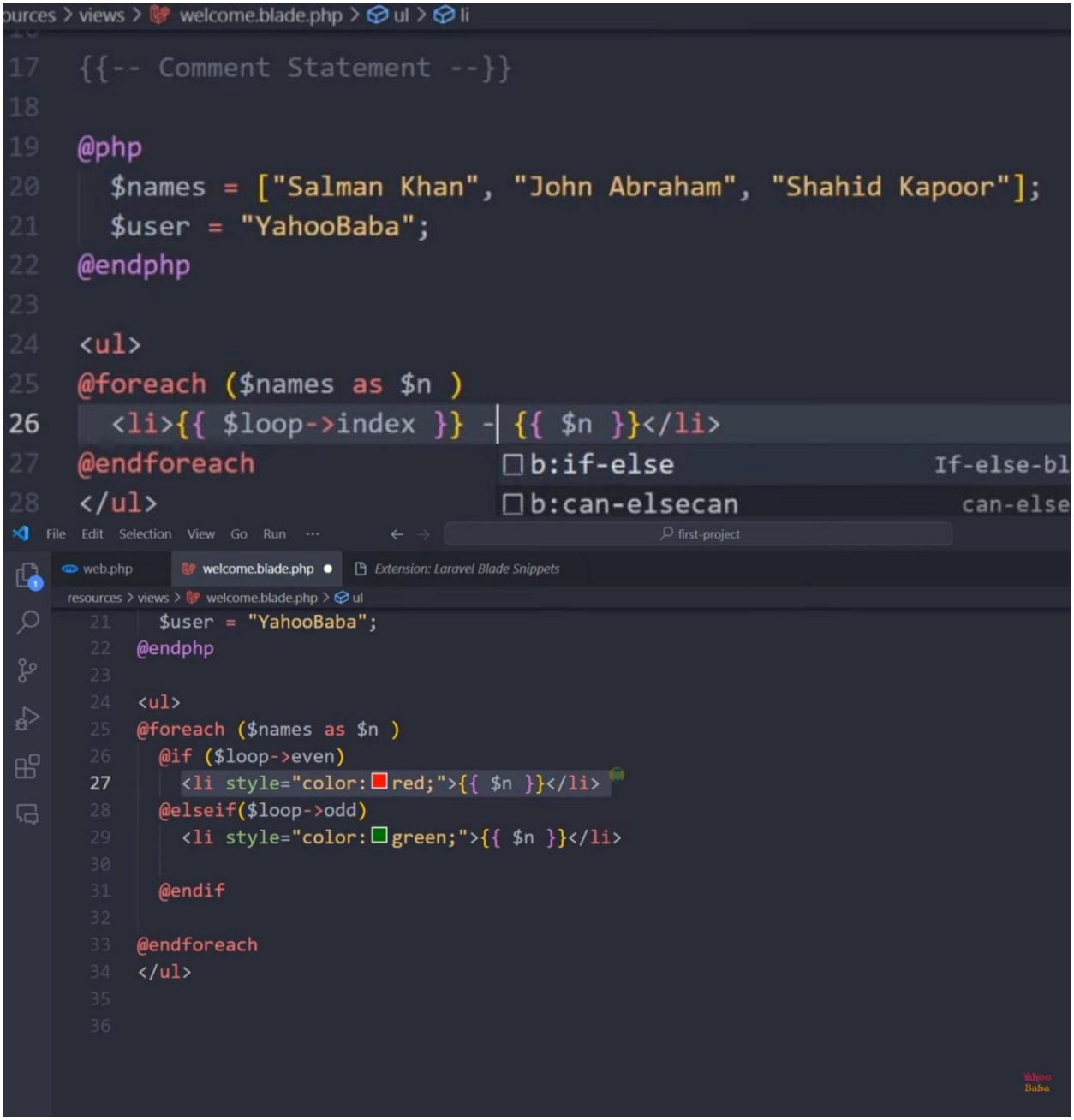
{{ "Hello World" }}

{{ "<h1>Hello World</h1>" }}
{!! "<h1>Yahoo Baba</h1>" !!}
{{-- {{ "<script>alert('Yahoo Baba')</script>" }} --}}
{{-- Comment Statement --}}



Blade Loop Variable for @foreach

Property	Description
$loop->index	The index of the current loop iteration (starts at 0).
$loop->iteration	The current loop iteration (starts at 1).
$loop->remaining	The iterations remaining in the loop.
$loop->count	The total number of items in the array being iterated.
$loop->first	Whether this is the first iteration through the loop.
$loop->last	Whether this is the last iteration through the loop.
$loop->even	Whether this is an even iteration through the loop.
$loop->odd	Whether this is an odd iteration through the loop.
$loop->depth	The nesting level of the current loop.
$loop->parent	When in a nested loop, the parent's loop variable.


ources > views > welcome.blade.php > ul > li
{{-- Comment Statement --}}
@php
$names = ["Salman Khan", "John Abraham", "Shahid Kapoor"];
$user = "YahooBaba";
@endphp

@foreach ($names as $n)
{{ $loop->index }} - {{ $n }}
@endforeach

b:if-else
If-else-bl
b:can-elsecan
can-else
File Edit Selection View Go Run ...
first-project
web.php
welcome.blade.php
Extension: Laravel Blade Snippets
resources > views > welcome.blade.php > ul
$user = "YahooBaba";
@endphp

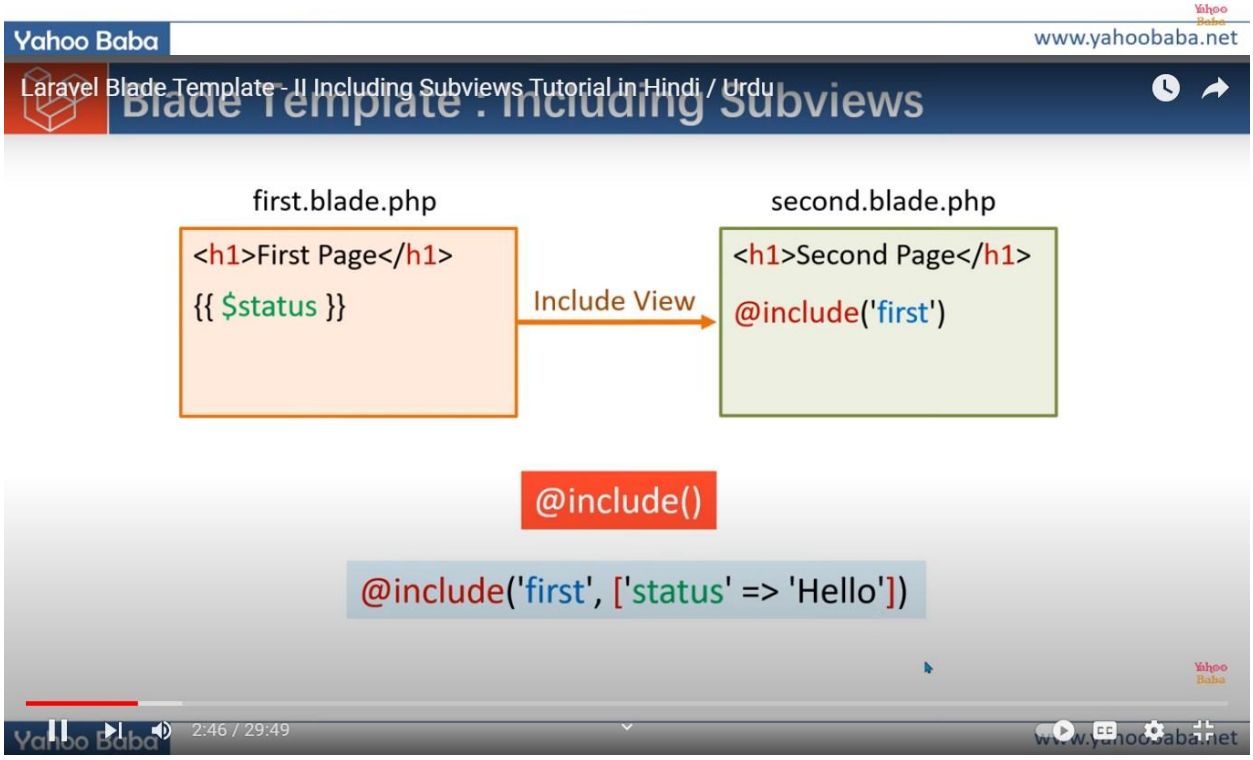
@foreach ($names as $n)
@if ($loop->even)
<li style="color: red;">{{ $n }}
@elseif($loop->odd)
<li style="color: green;">{{ $n }}
@endif
@endforeach

Yahoo Baba

Blade Template Main Directives

- @include
- @section
- @extend
- @yield

Reusable Templates



Blade Template : Including Subviews

first.blade.php

```
<h1>First Page</h1>
{{ $status }}
```

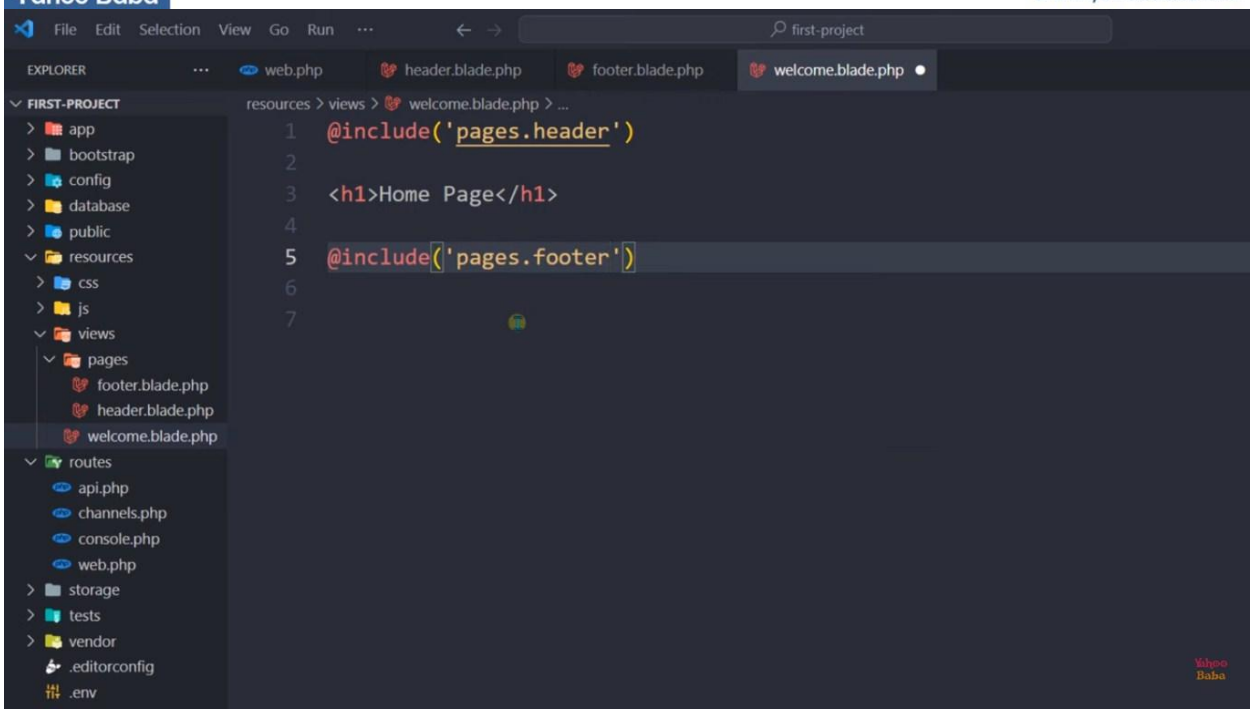
Include View

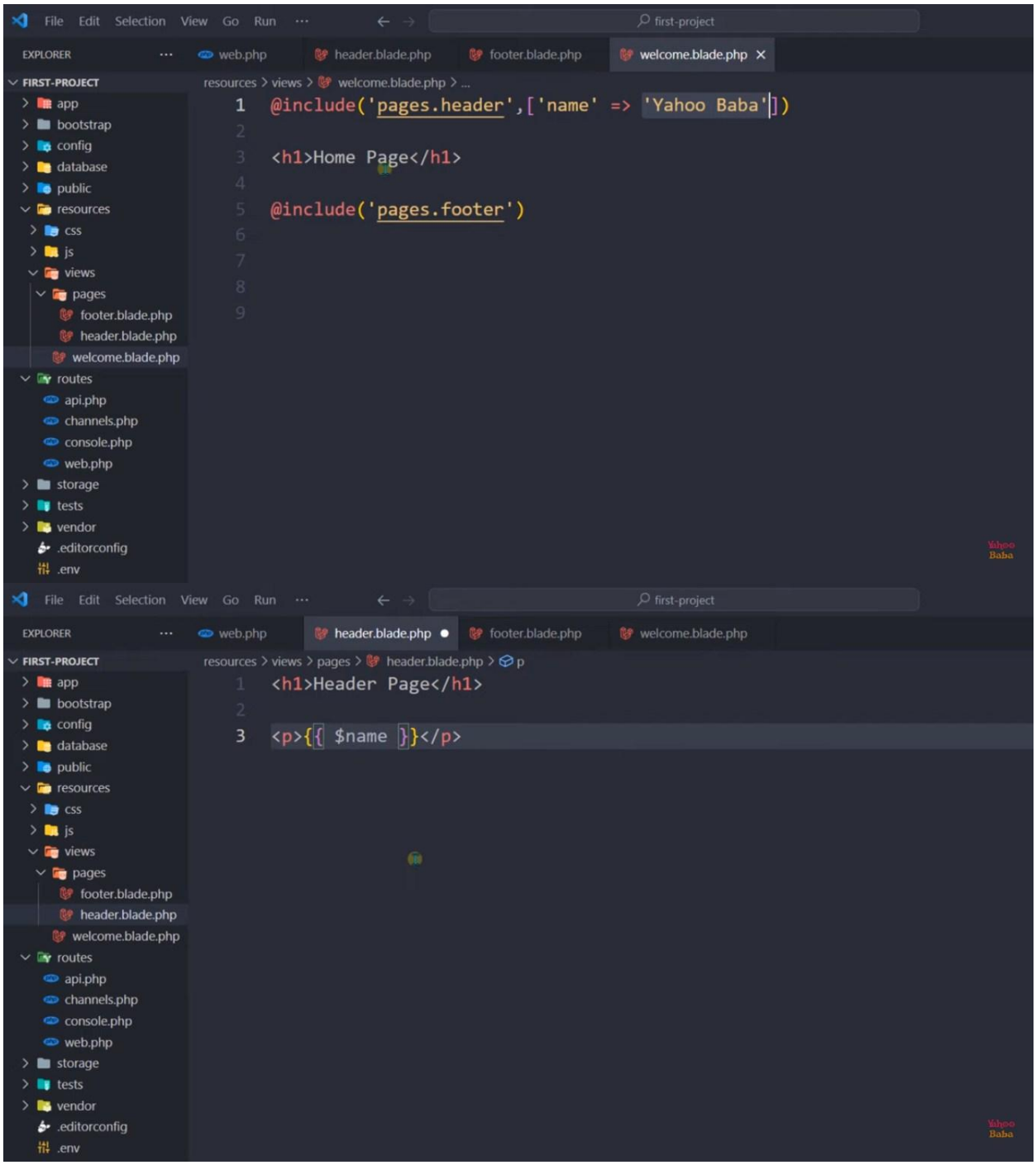
second.blade.php

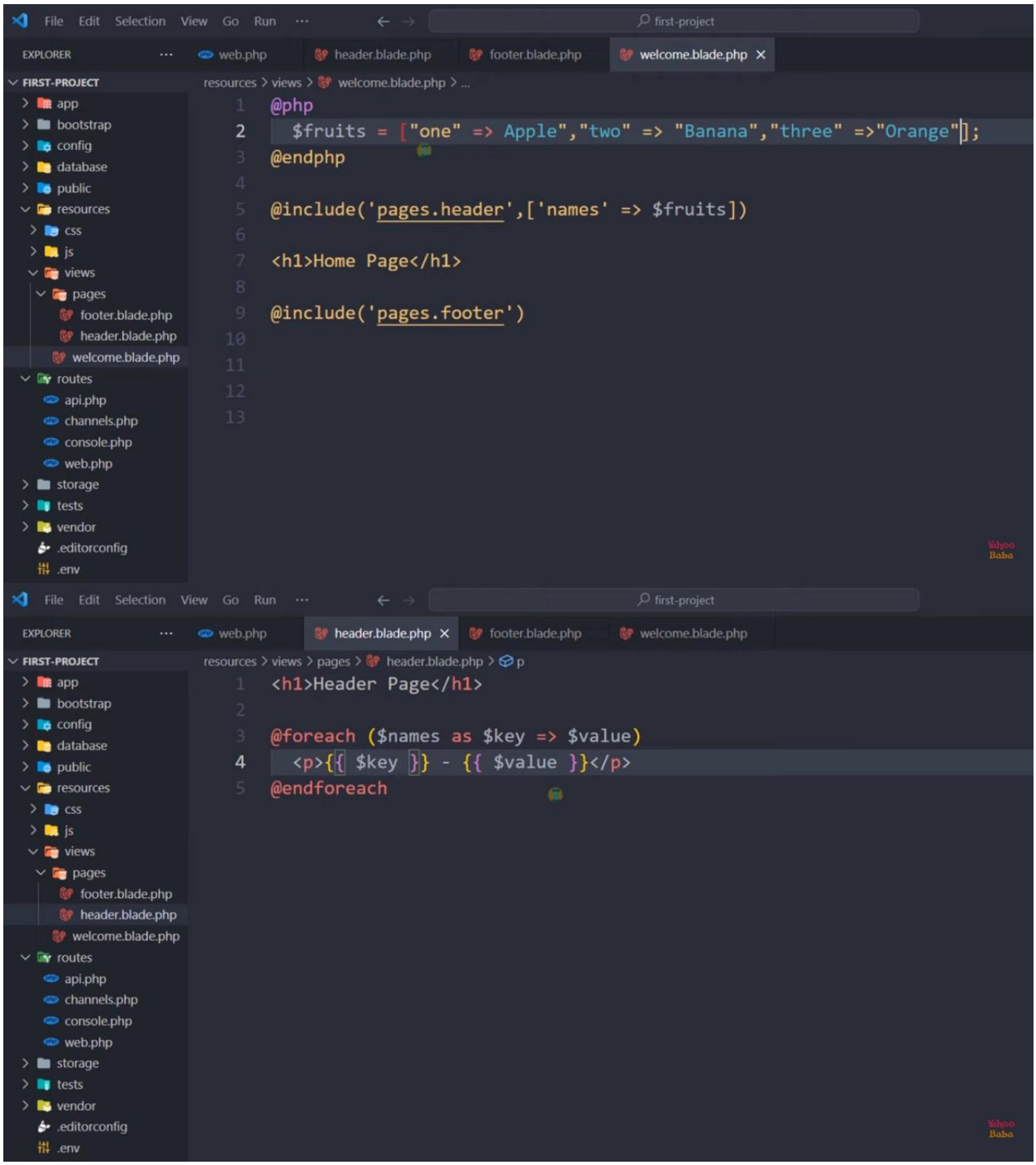
```
<h1>Second Page</h1>
@include('first')
```

@include()

```
@include('first', ['status' => 'Hello'])
```


File Edit Selection View Go Run
first-project
EXPLORER
web.php
header.blade.php
footer.blade.php
welcome.blade.php
FIRST-PROJECT
app
bootstrap
config
database
public
resources
css
js
views
pages
footer.blade.php
header.blade.php
welcome.blade.php
routes
api.php
channels.php
console.php
web.php
storage
tests
vendor
.editorconfig
.env
resources > views > welcome.blade.php > ...
1 @include('pages.header',['name' => 'Yahoo Baba'])
3 <h1>Home Page</h1>
5 @include('pages.footer')
resources > views > pages > header.blade.php > p
1 <h1>Header Page</h1>
3 {{ $name }}
Yahoo Baba

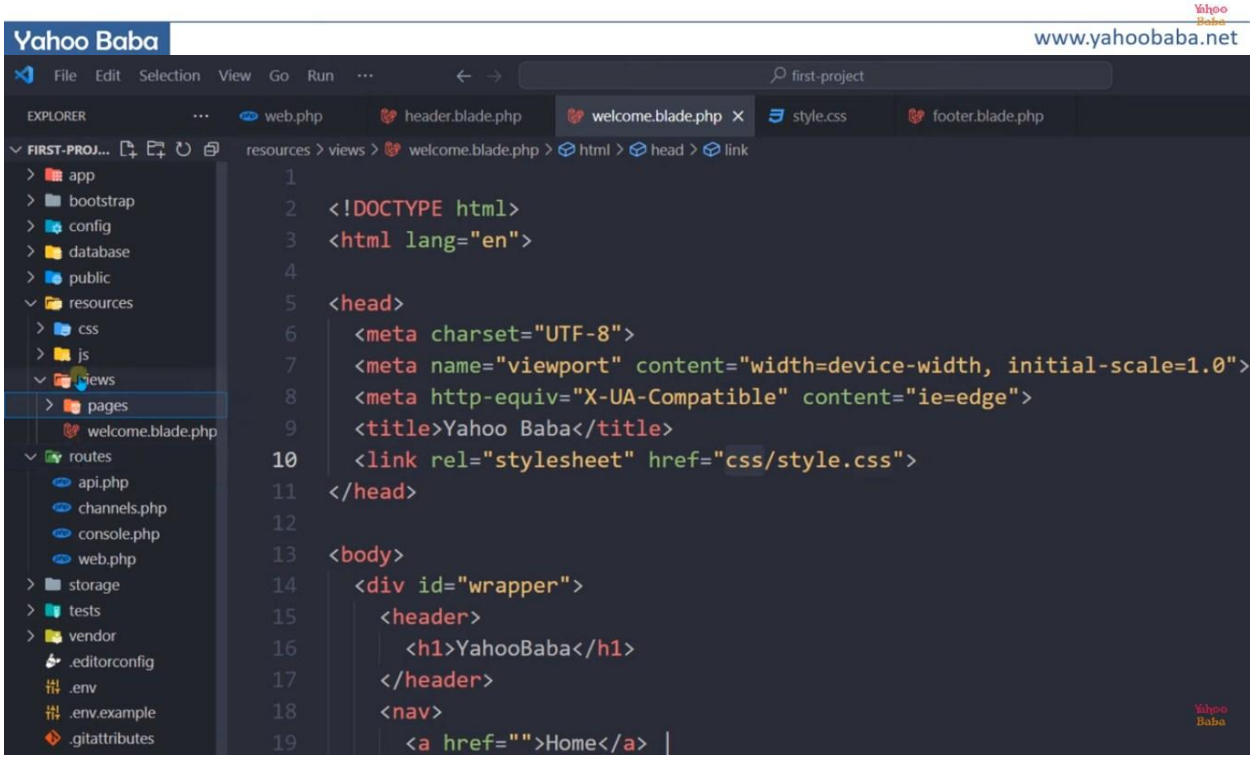

File Edit Selection View Go Run
first-project
EXPLORER
web.php
header.blade.php
footer.blade.php
welcome.blade.php
FIRST-PROJECT
resources > views > welcome.blade.php > ...
app
bootstrap
config
database
public
resources
css
js
views
pages
footer.blade.php
header.blade.php
welcome.blade.php
routes
api.php
channels.php
console.php
web.php
storage
tests
vendor
.editorconfig
.env
@php
$fruits = ["one" => Apple","two" => "Banana","three" =>"Orange"];
@endphp
@include('pages.header',['names' => $fruits])
<h1>Home Page</h1>
@include('pages.footer')
Yahoo Baba
File Edit Selection View Go Run
first-project
EXPLORER
web.php
header.blade.php
footer.blade.php
welcome.blade.php
FIRST-PROJECT
resources > views > pages > header.blade.php > p
app
bootstrap
config
database
public
resources
css
js
views
pages
footer.blade.php
header.blade.php
welcome.blade.php
routes
api.php
channels.php
console.php
web.php
storage
tests
vendor
.editorconfig
.env
<h1>Header Page</h1>
@foreach ($names as $key => $value)
{{ $key }} - {{ $value }}
@endforeach
Yahoo Baba

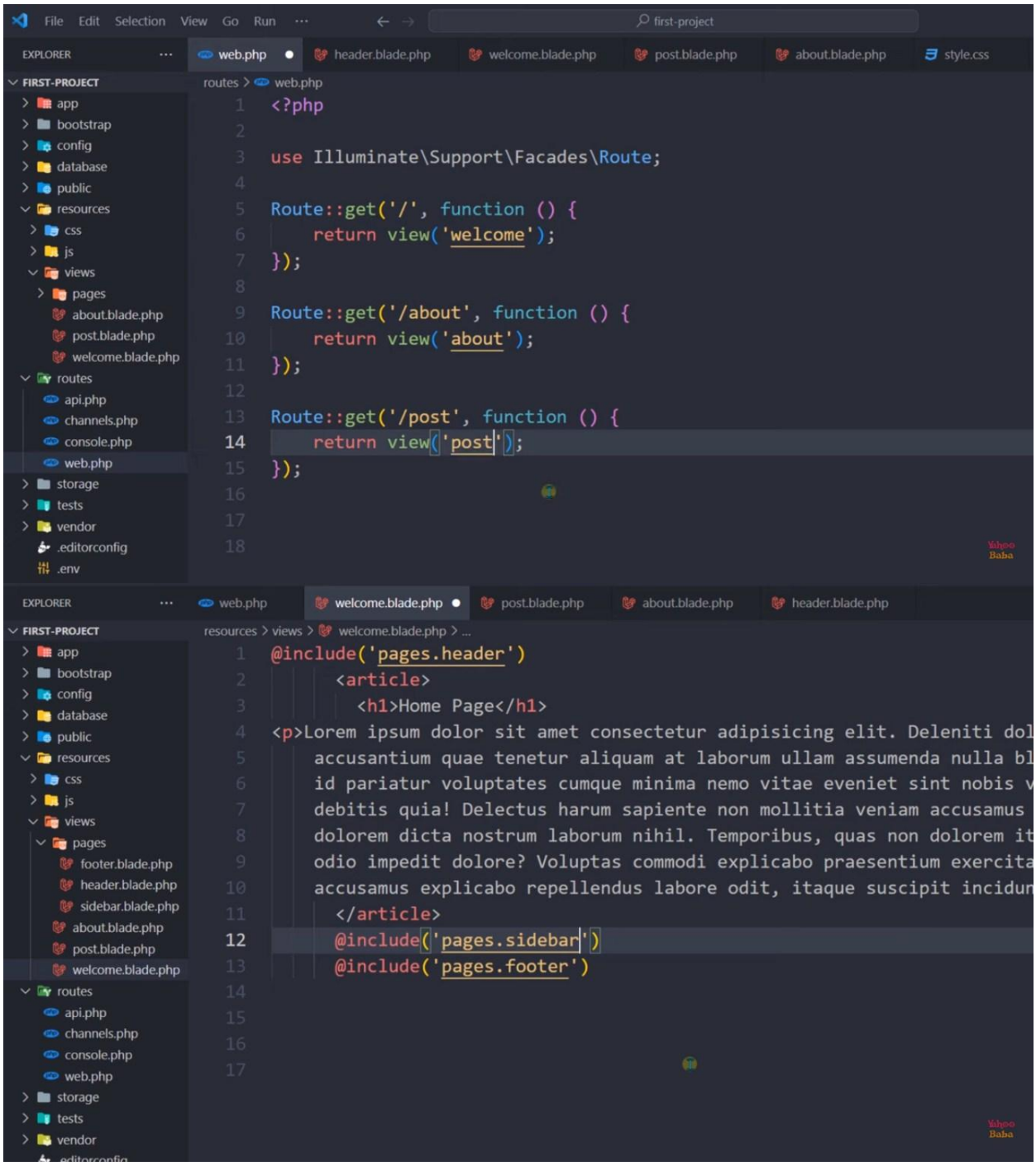
Including Subviews with Conditional Check

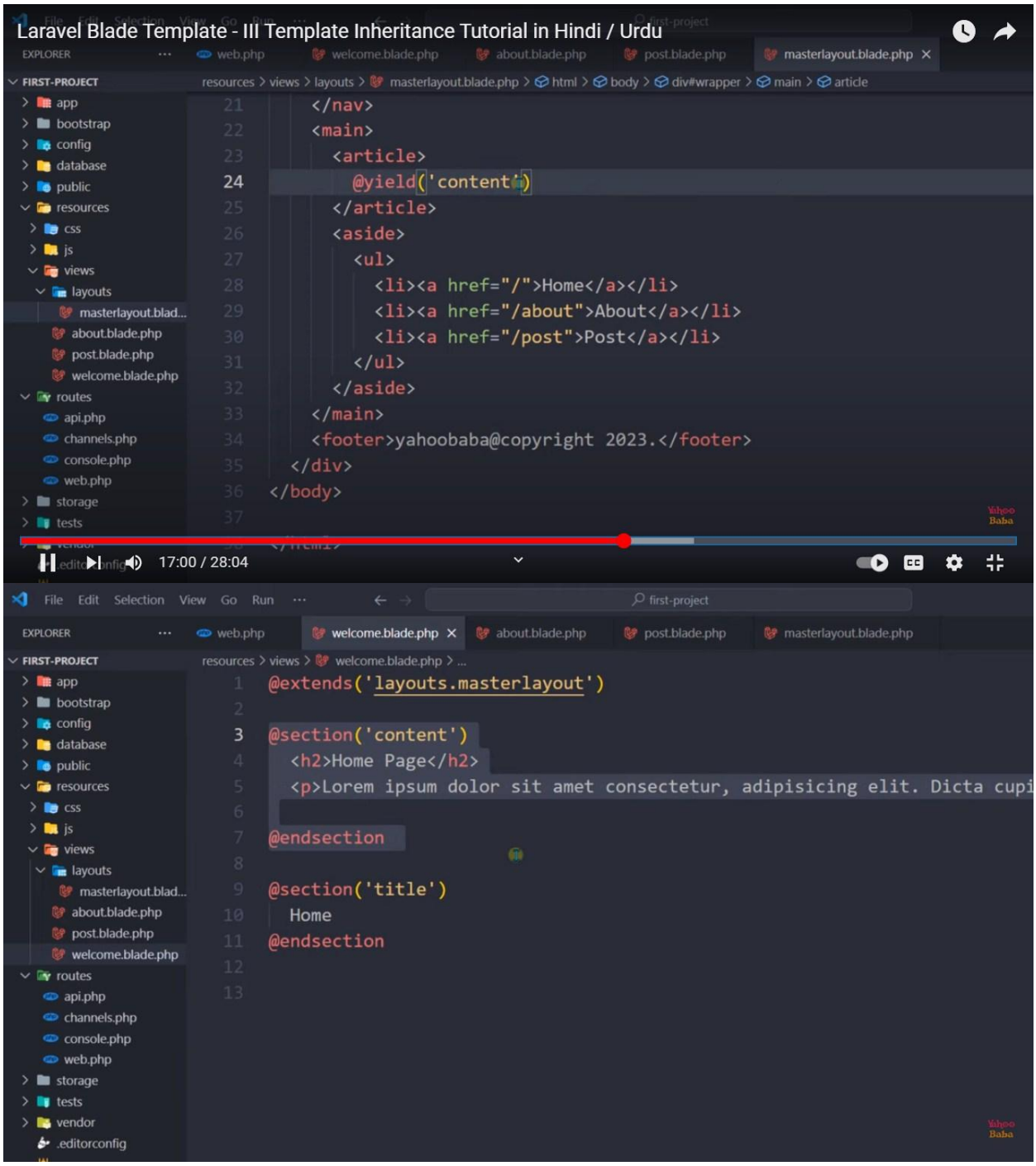
```
@includeWhen (Condition Value,'viewfile', ['status' => 'Hello'])
```

TRUE / FALSE

```
@includeUnless (Condition Value,'viewfile', ['status' => 'Hello'])
```



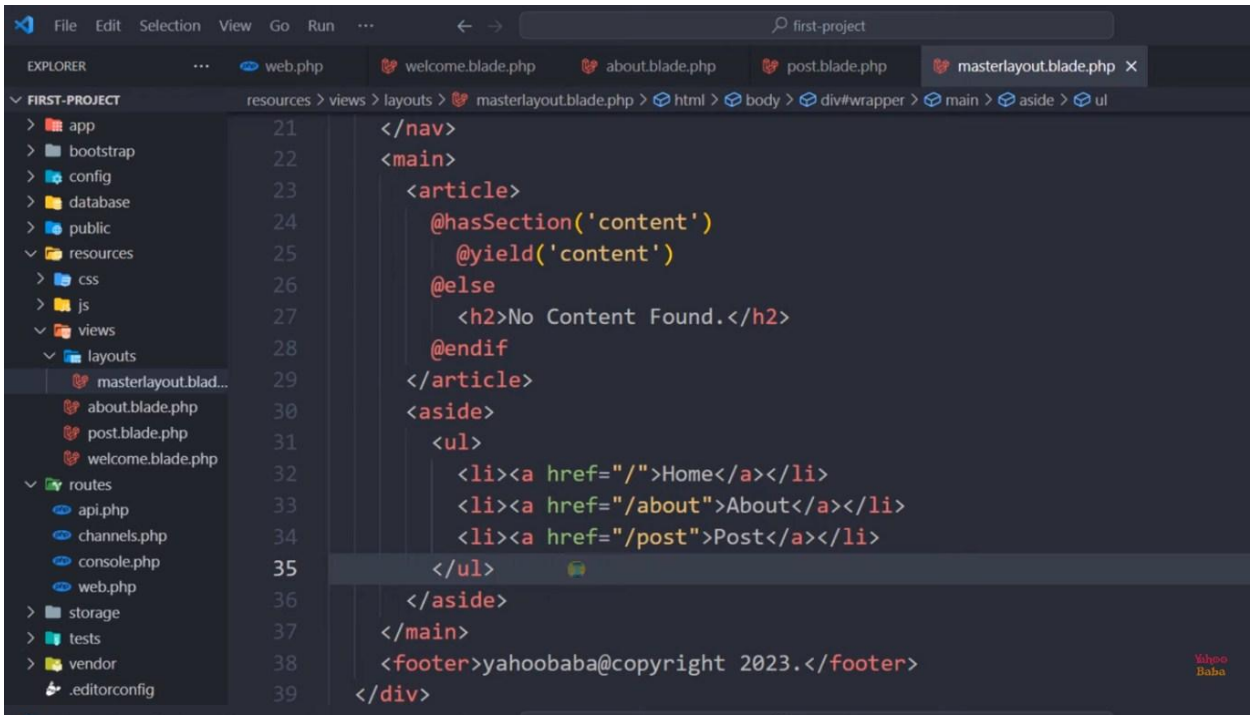

File Edit Selection View Go Run
first-project
EXPLORER
web.php
header.blade.php
welcome.blade.php
post.blade.php
about.blade.php
style.css
FIRST-PROJECT
routes > web.php
app
bootstrap
config
database
public
resources
css
js
views
pages
about.blade.php
post.blade.php
welcome.blade.php
routes
api.php
channels.php
console.php
web.php
storage
tests
vendor
.editorconfig
.env
<?php
use Illuminate\Support\Facades\Route;
Route::get('/', function () {
return view('welcome');
});
Route::get('/about', function () {
return view('about');
});
Route::get('/post', function () {
return view('post');
});
EXPLORER
web.php
welcome.blade.php
post.blade.php
about.blade.php
header.blade.php
FIRST-PROJECT
resources > views > welcome.blade.php > ...
app
bootstrap
config
database
public
resources
css
js
views
pages
footer.blade.php
header.blade.php
sidebar.blade.php
about.blade.php
post.blade.php
welcome.blade.php
routes
api.php
channels.php
console.php
web.php
storage
tests
vendor
@include('pages.header')
<article>
<h1>Home Page</h1>
<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Deleniti dol
accusantium quae tenetur aliquam at laborum ullam assumenda nulla bl
id pariatur voluptates cumque minima nemo vitae eveniet sint nobis v
debitis quia! Delectus harum sapiente non mollitia veniam accusamus
dolorem dicta nostrum laborum nihil. Temporibus, quas non dolorem it
odio impedit dolore? Voluptas commodi explicabo praesentium exercita
accusamus explicabo repellendus labore odit, itaque suscipit incidun
</article>
@include('pages.sidebar')
@include('pages.footer')


Laravel Blade Template - III Template Inheritance Tutorial in Hindi / Urdu
EXPLORER
web.php
welcome.blade.php
about.blade.php
post.blade.php
masterlayout.blade.php
FIRST-PROJECT
resources > views > layouts > masterlayout.blade.php > html > body > div#wrapper > main > article
app
bootstrap
config
database
public
resources
css
js
views
layouts
masterlayout.blad...
about.blade.php
post.blade.php
welcome.blade.php
routes
api.php
channels.php
console.php
web.php
storage
tests
vendor
.editorconfig
</nav>
<main>
<article>
@yield('content')
</article>
<aside>

Home
About
Post

</aside>
</main>
<footer>yahoobaba@copyright 2023.</footer>

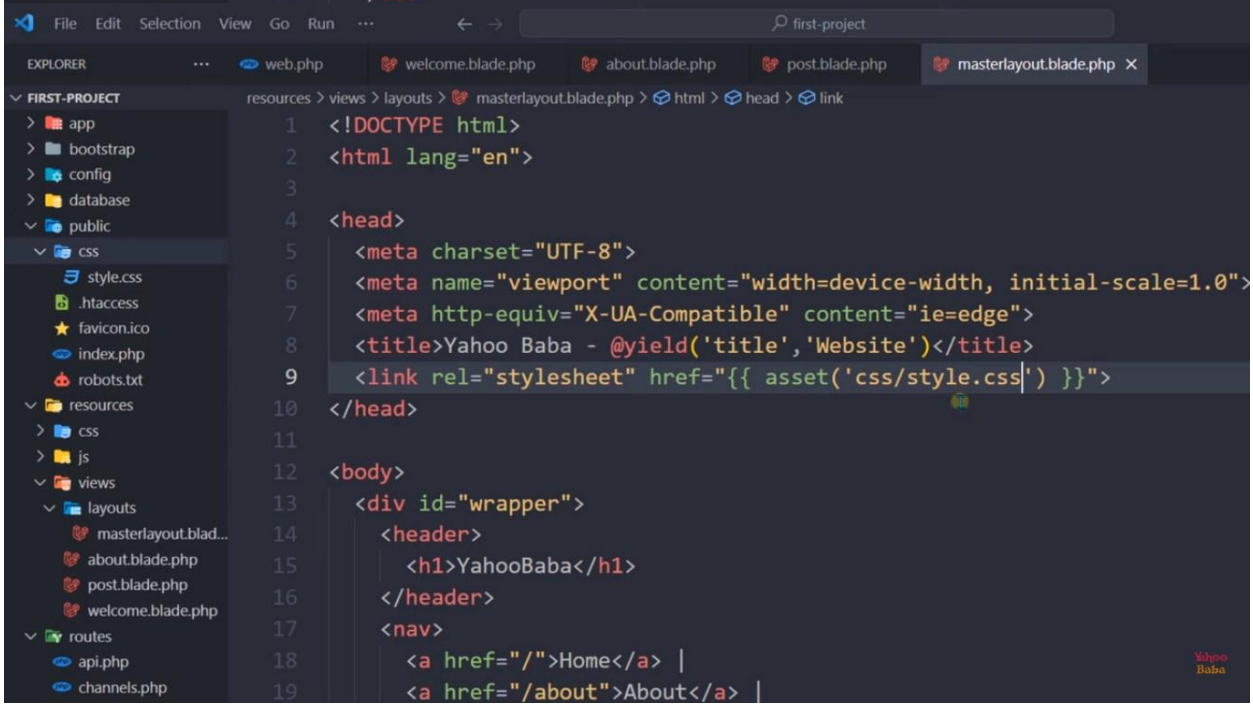
</body>
17:00 / 28:04
File Edit Selection View Go Run
first-project
resources > views > welcome.blade.php > ...
@extends('layouts.masterlayout')
@section('content')
<h2>Home Page</h2>
<p>Lorem ipsum dolor sit amet consectetur, adipisicing elit. Dicta cupi
@endsection
@section('title')
Home
@endsection


File Edit Selection View Go Run
first-project
EXPLORER
web.php
welcome.blade.php
about.blade.php
post.blade.php
masterlayout.blade.php
FIRST-PROJECT
resources > views > layouts > masterlayout.blade.php > html > body > div#wrapper > main > aside > ul
app
bootstrap
config
database
public
resources
css
js
views
layouts
masterlayout.blad...
about.blade.php
post.blade.php
welcome.blade.php
routes
api.php
channels.php
console.php
web.php
storage
tests
vendor
.editorconfig
</nav>
<main>
<article>
@hasSection('content')
@yield('content')
@else
<h2>No Content Found.</h2>
@endif
</article>
<aside>

Home
About
Post

</aside>
</main>
<footer>yahoobaba@copyright 2023.</footer>

Yahoo Baba


File Edit Selection View Go Run
first-project
EXPLORER
web.php
welcome.blade.php
about.blade.php
post.blade.php
masterlayout.blade.php
FIRST-PROJECT
resources > views > layouts > masterlayout.blade.php > html > head > link
app
bootstrap
config
database
public
css
style.css
.htaccess
favicon.ico
index.php
robots.txt
resources
css
js
views
layouts
masterlayout.blad...
about.blade.php
post.blade.php
welcome.blade.php
routes
api.php
channels.php
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Yahoo Baba - @yield('title','Website')</title>
<link rel="stylesheet" href="{{ asset('css/style.css') }}">
</head>
<body>

<header>
<h1>YahooBaba</h1>
</header>
<nav>
Home |
About |
Yahoo Baba

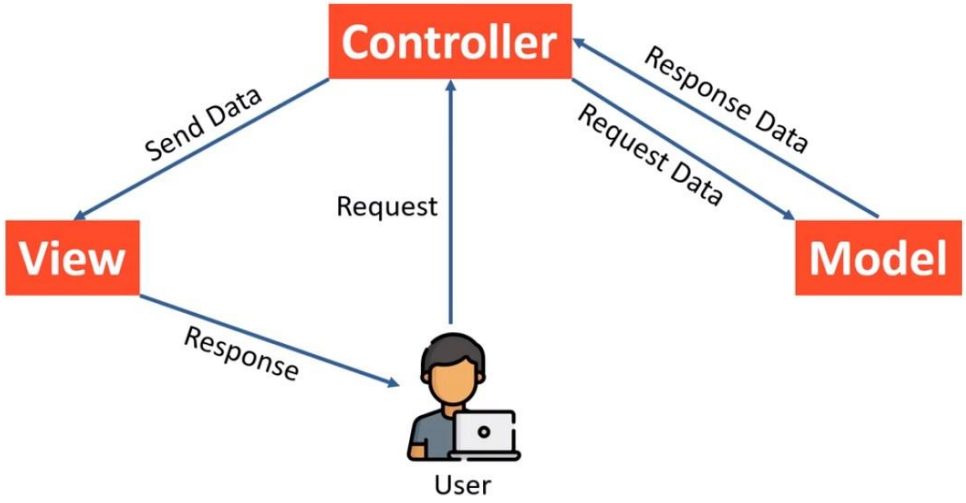
Passing Data : Route to View


```
Route::get('/users', function () {
   $name = 'Yahoo Baba';
   return view('userpage', ['user' => $name] );
});
```

```
<h1>{{ $user }}</h1>
```

userpage.blade.php

Controller Role In MVC Framework



Laravel : Controller

1. Create Controller File

```
php artisan make:controller UserController
```

2. Create Controller Class

```
class UserController extends Controller
{
    public function show()
    {
        return view('user.profile');
    }
}
```

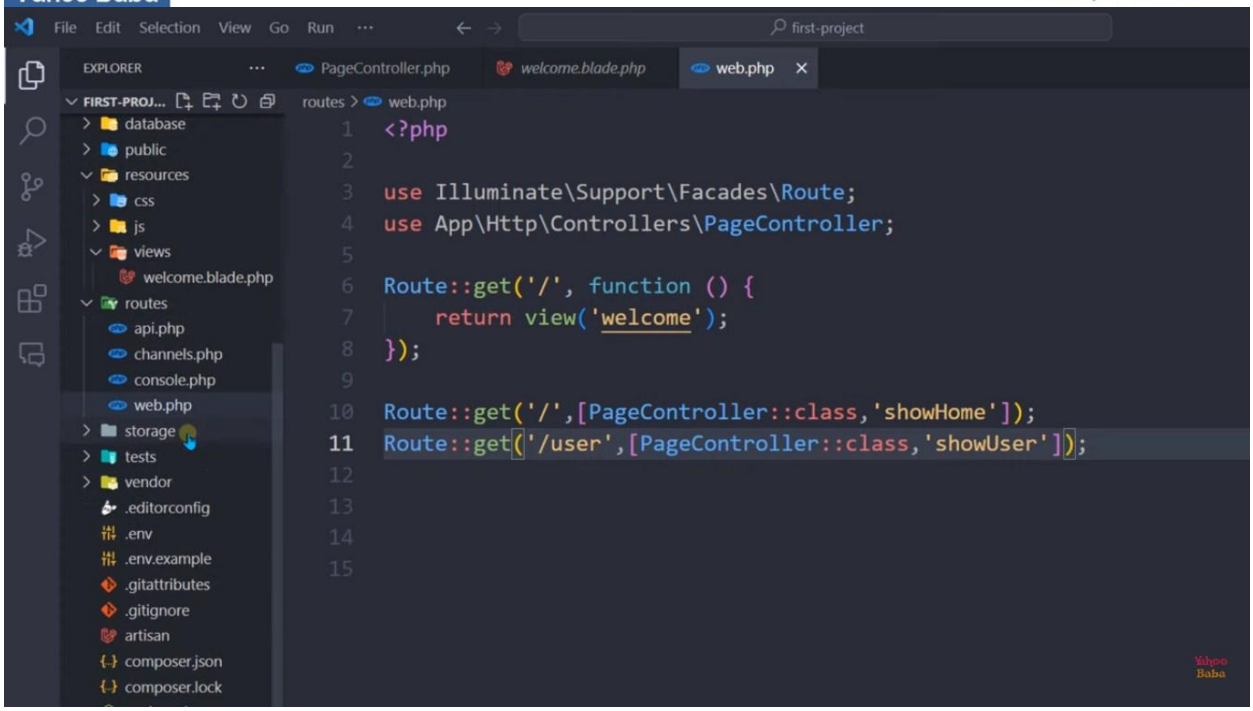
UserController.php

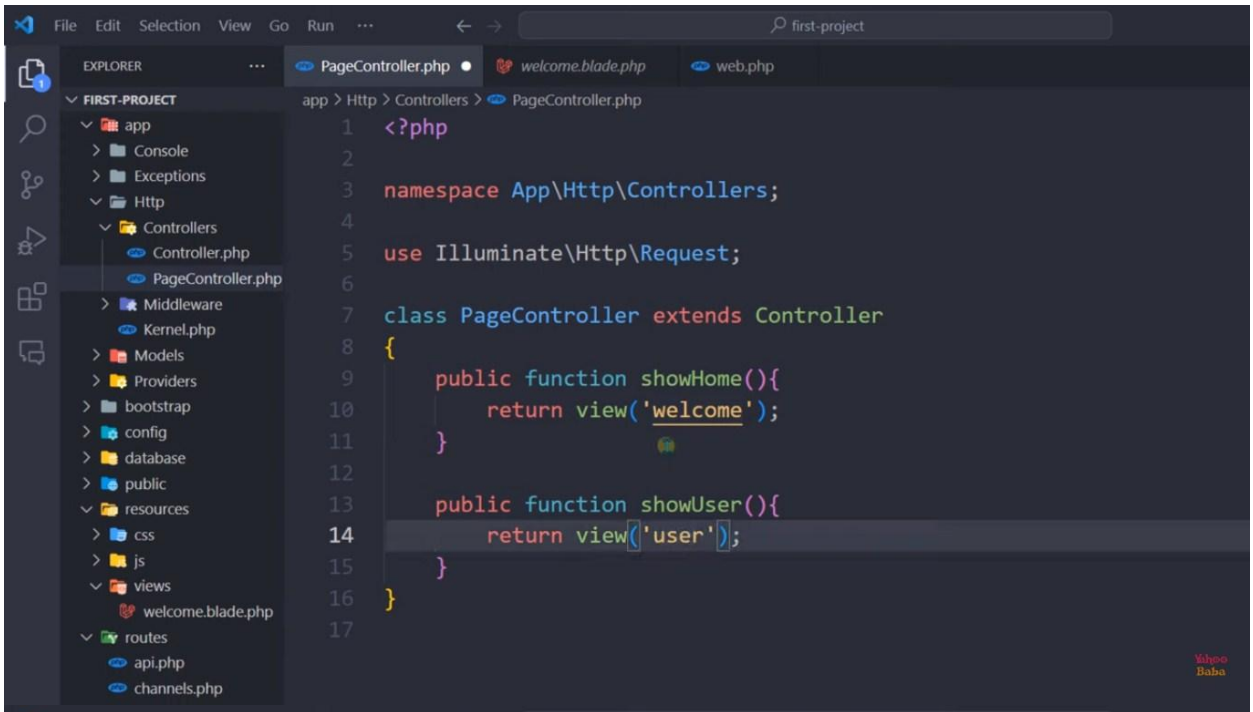
3. Create Route

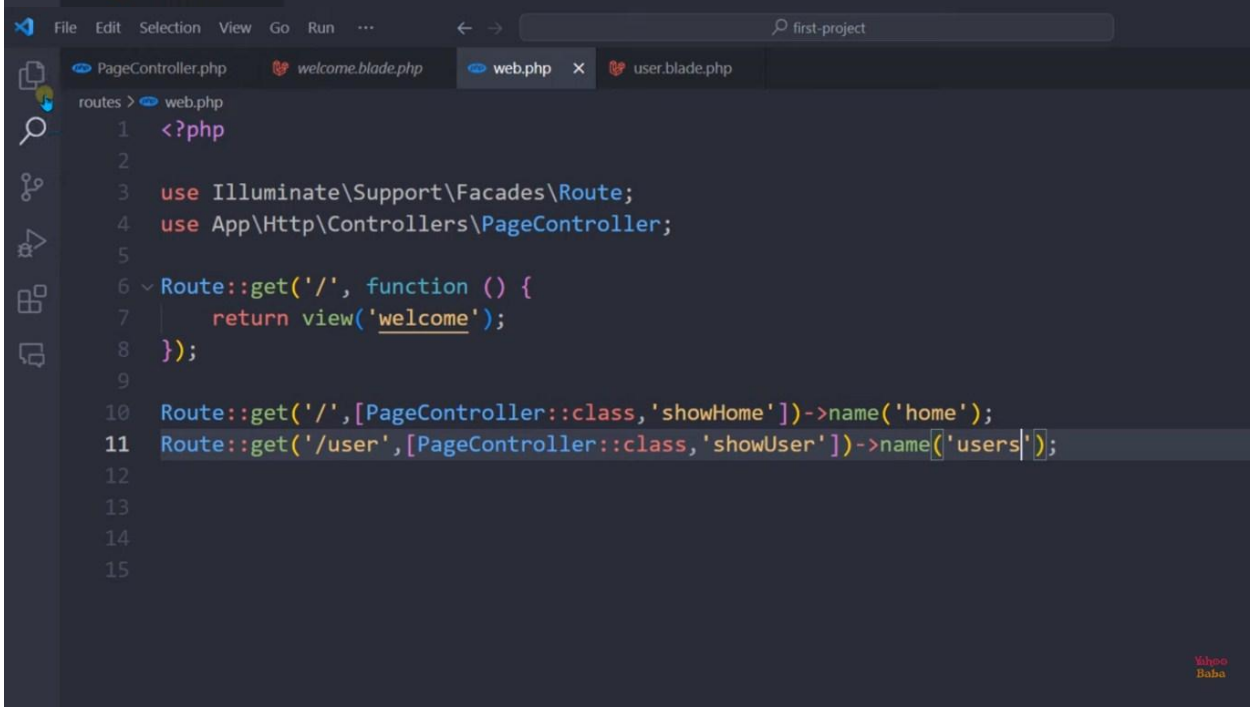
```
use App\Http\Controllers\UserController;

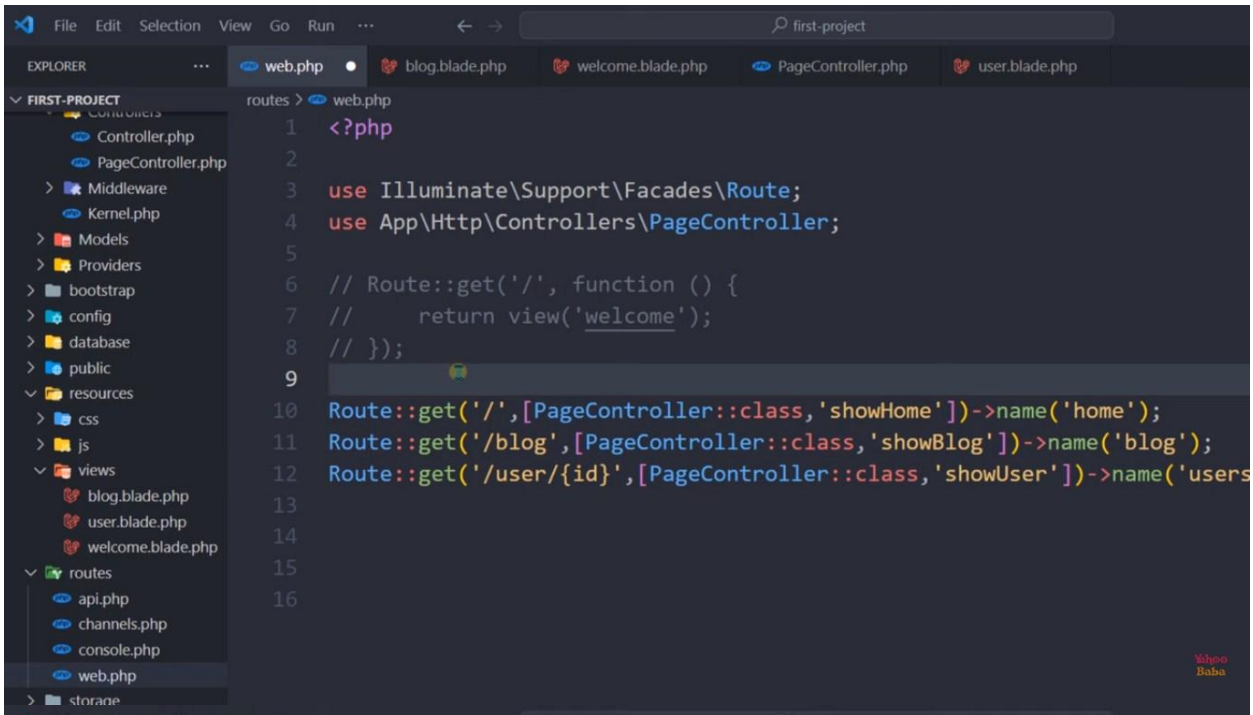
Route::get('/user', [UserController::class, 'show']);
```

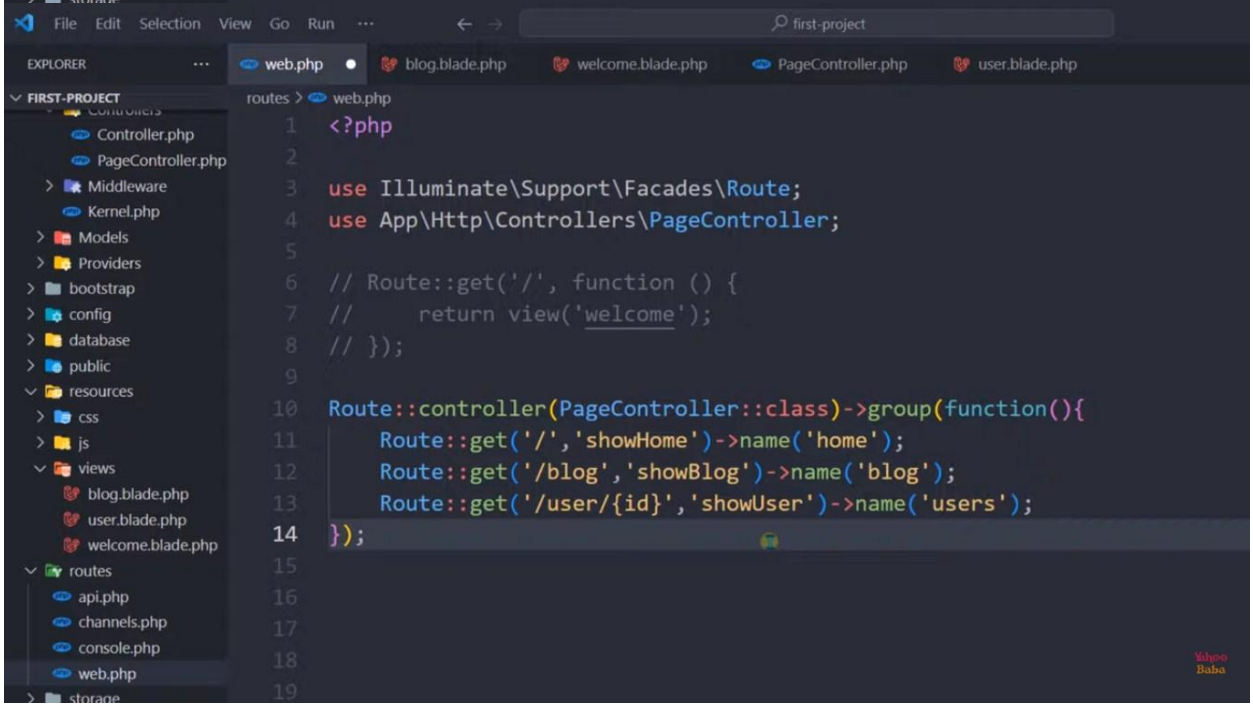
http://localhost/user




File Edit Selection View Go Run ···
first-project
EXPLORER
PageController.php
welcome.blade.php
web.php
FIRST-PROJECT
app
Console
Exceptions
Http
Controllers
Controller.php
PageController.php
Middleware
Kernel.php
Models
Providers
bootstrap
config
database
public
resources
css
js
views
welcome.blade.php
routes
api.php
channels.php
app > Http > Controllers > PageController.php
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
class PageController extends Controller
{
public function showHome(){
return view('welcome');
}
public function showUser(){
return view('user');
}
}


File Edit Selection View Go Run ···
first-project
PageController.php
welcome.blade.php
web.php
user.blade.php
routes > web.php
<?php
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\PageController;
Route::get('/', function () {
return view('welcome');
});
Route::get('/',[PageController::class,'showHome'])->name('home');
Route::get('/user',[PageController::class,'showUser'])->name('users');


File Edit Selection View Go Run
first-project
EXPLORER
web.php
blog.blade.php
welcome.blade.php
PageController.php
user.blade.php
FIRST-PROJECT
routes > web.php
Controller.php
PageController.php
Middleware
Kernel.php
Models
Providers
bootstrap
config
database
public
resources
css
js
views
blog.blade.php
user.blade.php
welcome.blade.php
routes
api.php
channels.php
console.php
web.php
storage
1 <?php
3 use Illuminate\Support\Facades\Route;
4 use App\Http\Controllers\PageController;
6 // Route::get('/', function () {
7 // return view('welcome');
8 // });
10 Route::get('/',[PageController::class,'showHome'])->name('home');
11 Route::get('/blog',[PageController::class,'showBlog'])->name('blog');
12 Route::get('/user/{id}',[PageController::class,'showUser'])->name('users


File Edit Selection View Go Run
first-project
EXPLORER
web.php
blog.blade.php
welcome.blade.php
PageController.php
user.blade.php
FIRST-PROJECT
routes > web.php
Controller.php
PageController.php
Middleware
Kernel.php
Models
Providers
bootstrap
config
database
public
resources
css
js
views
blog.blade.php
user.blade.php
welcome.blade.php
routes
api.php
channels.php
console.php
web.php
storage
1 <?php
3 use Illuminate\Support\Facades\Route;
4 use App\Http\Controllers\PageController;
6 // Route::get('/', function () {
7 // return view('welcome');
8 // });
10 Route::controller(PageController::class)->group(function(){
11 Route::get('/','showHome')->name('home');
12 Route::get('/blog','showBlog')->name('blog');
13 Route::get('/user/{id}','showUser')->name('users');
14 });

Laravel : Single Action Controllers

```
class TestController extends Controller
{
    public function __invoke()
    {
        return view('test');
    }
}
```

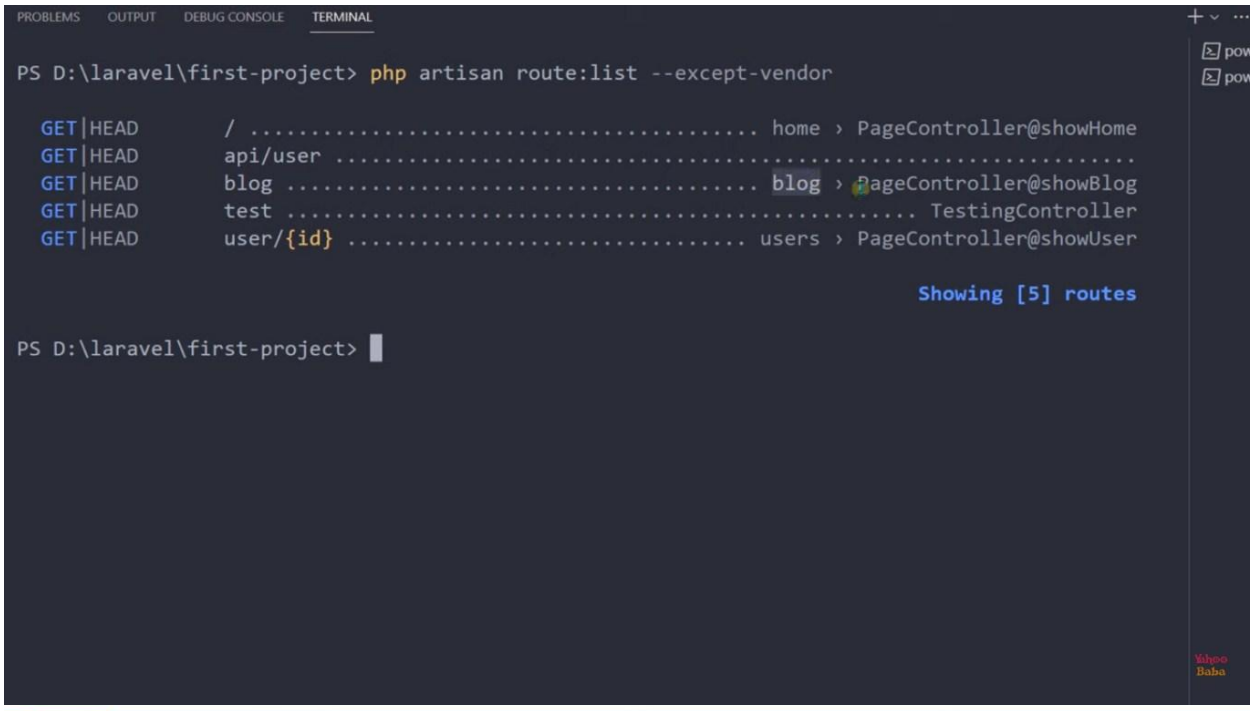
```
Route::get('/test', TestingController::class);
```


Laravel : Single Action Controllers

```
php artisan make:controller TestingController --invokable
or
php artisan make:controller TestingController --i
```

```
class TestController extends Controller
{
    public function __invoke()
    {
        return view('test');
    }
}
```

```
Route::get('/test', TestingController::class);
```



Laravel : Steps to work in Model

1. Create Database — CREATE DATABASE laravel-db or Xampp (Phpmyadmin)
   Projectfolder/.env
2. Create Database Migration (Create tables in database)
   php artisan make:migration create_students_table
   Projectfolder/database/migrations/
3. Seeding (Insert Initial Data in Tables)
4. Create Model

Laravel : Migration

```
CREATE TABLE table_name (
   id INT(),
   name VARCHAR(30),
   city VARCHAR(15),
  ....
);
```

1. `php artisan make:migration create_students_table`

2.
```
return new class extends Migration
{
    public function up(): void
    {
        Schema::create('students', function (Blueprint $table) {
            $table->id();
            $table->string('name', 30);
            $table->string('city', 15);
        });
    }
}
```

3. `php artisan migrate`

Laravel : Artisan Migration Commands

```
php artisan make:migration create_students_table
php artisan migrate

php artisan migrate:rollback
php artisan migrate:reset
php artisan migrate:refresh
```

Laravel : Add to Column with Migration

1. php artisan make:migration update_students_table --table=students

2.
```
return new class extends Migration
{
    public function up(): void
    {
        Schema::table('students', function (Blueprint $table) {
            $table->string('city');

        });
    }
}
```

3. php artisan migrate

Laravel : Modify Column with Migration

```
$table->renameColumn('from', 'to');
```

MySQL < 8.0.3
MariaDB < 10.5.2

```
$table->dropColumn('city');
```

```
$table->dropColumn(['city', 'avatar', 'location']);
```

```
$table->string('name', 50)->change();
```

```
$table->integer('votes')->unsigned()->default(1)->comment('my comment')->change();
```

Laravel : Constraints with Migration

MySQL

- NOT NULL
```
$table->string('email')->nullable();
```
- UNIQUE
```
$table->string('email')->unique();
$table->unique('email');
```
- DEFAULT
```
$table->string('city')->default('Agra');
```
- PRIMARY KEY
```
$table->primary('user_id');
```
- FOREIGN KEY
```
$table->foreign('user_id')->references('id')->on('users');
```
- CHECK

Laravel : Column Modifiers

Modifier	Description
->after('column')	Place the column "after" another column (MySQL).
->autoIncrement()	Set INTEGER columns as auto-incrementing (primary key).
->comment('my comment')	Add a comment to a column (MySQL/PostgreSQL).
->first()	Place the column "first" in the table (MySQL).
->from($integer)	Set the starting value of an auto-incrementing field (MySQL / PostgreSQL).
->invisible()	Make the column "invisible" to SELECT * queries (MySQL).
->unsigned()	Set INTEGER columns as UNSIGNED (MySQL).
->useCurrent()	Set TIMESTAMP columns to use CURRENT_TIMESTAMP as default value.
->useCurrentOnUpdate()	Set TIMESTAMP columns to use CURRENT_TIMESTAMP when a record is updated (MySQL).

Laravel : Primary Key & Foreign Key Constraints

Student Table

Id	Name	Age	City
1	Ram Kumar	19	1
2	Salman Khan	18	2
3	Meera Khan	19	1
4	Sarita Kumari	21	3

City Table

Cid	City
1	Agra
2	Bhopal
3	Delhi

Link

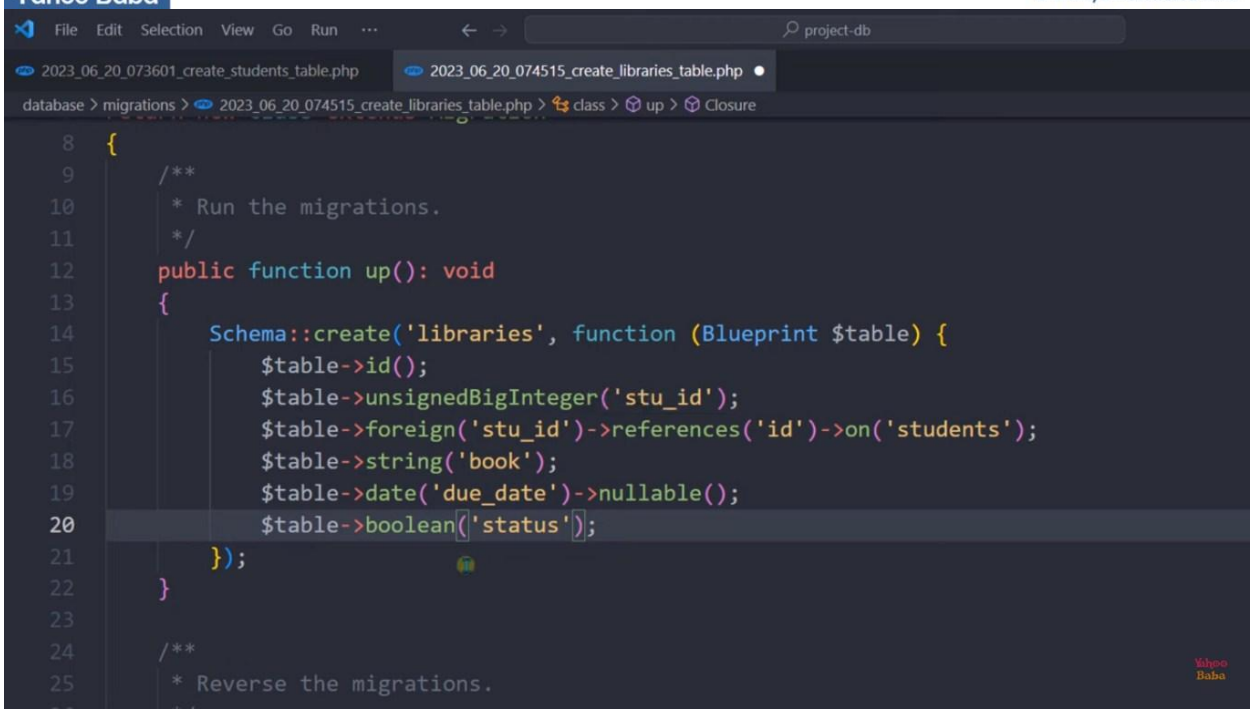
$table->primary('Cid');

PRIMARY KEY

FOREIGN KEY

PRIMARY KEY

$table->foreign('city')->references('Cid')->on('City');



Laravel : Steps to work in Seeder

1. php artisan make:model student
2. php artisan make:seeder StudentSeeder
3.

```
use App\Models\student;

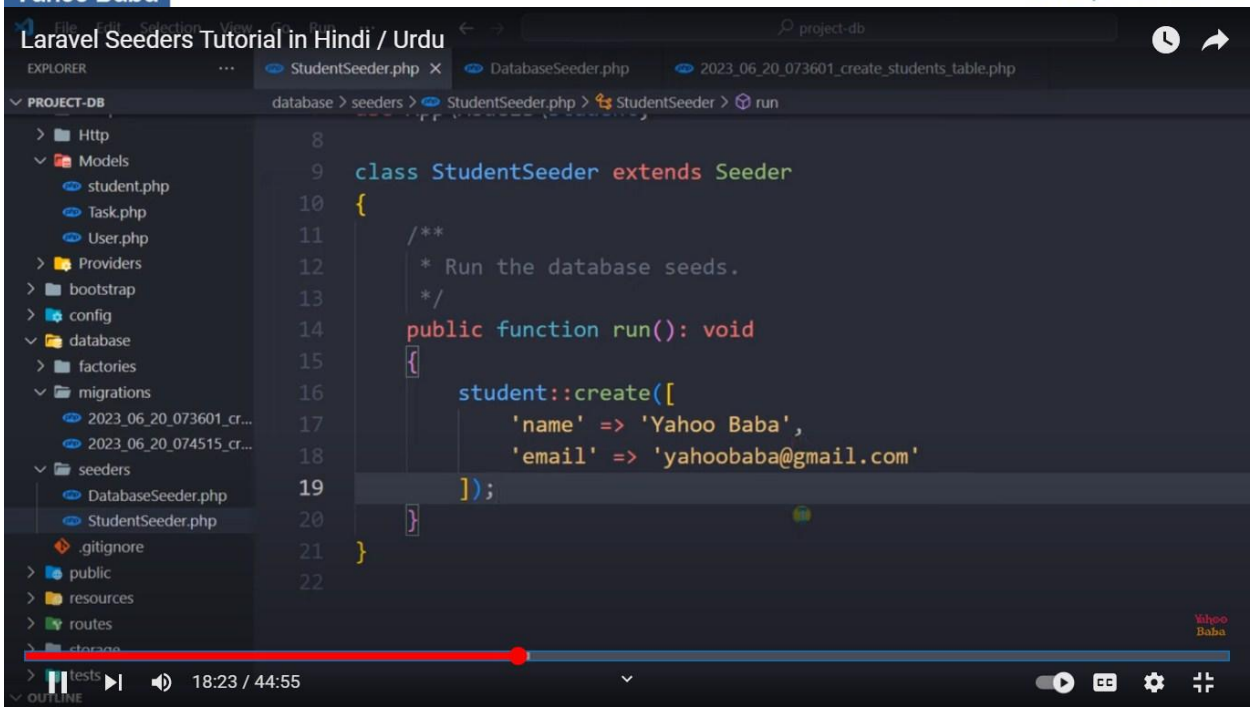
class StudentSeeder extends Seeder
{
    public function run(): void
    {
        student::create([
            'name' => 'Yahoo Baba',
            'email' => 'yahoobaba@gmail.com'
        ]);
    }
}
```

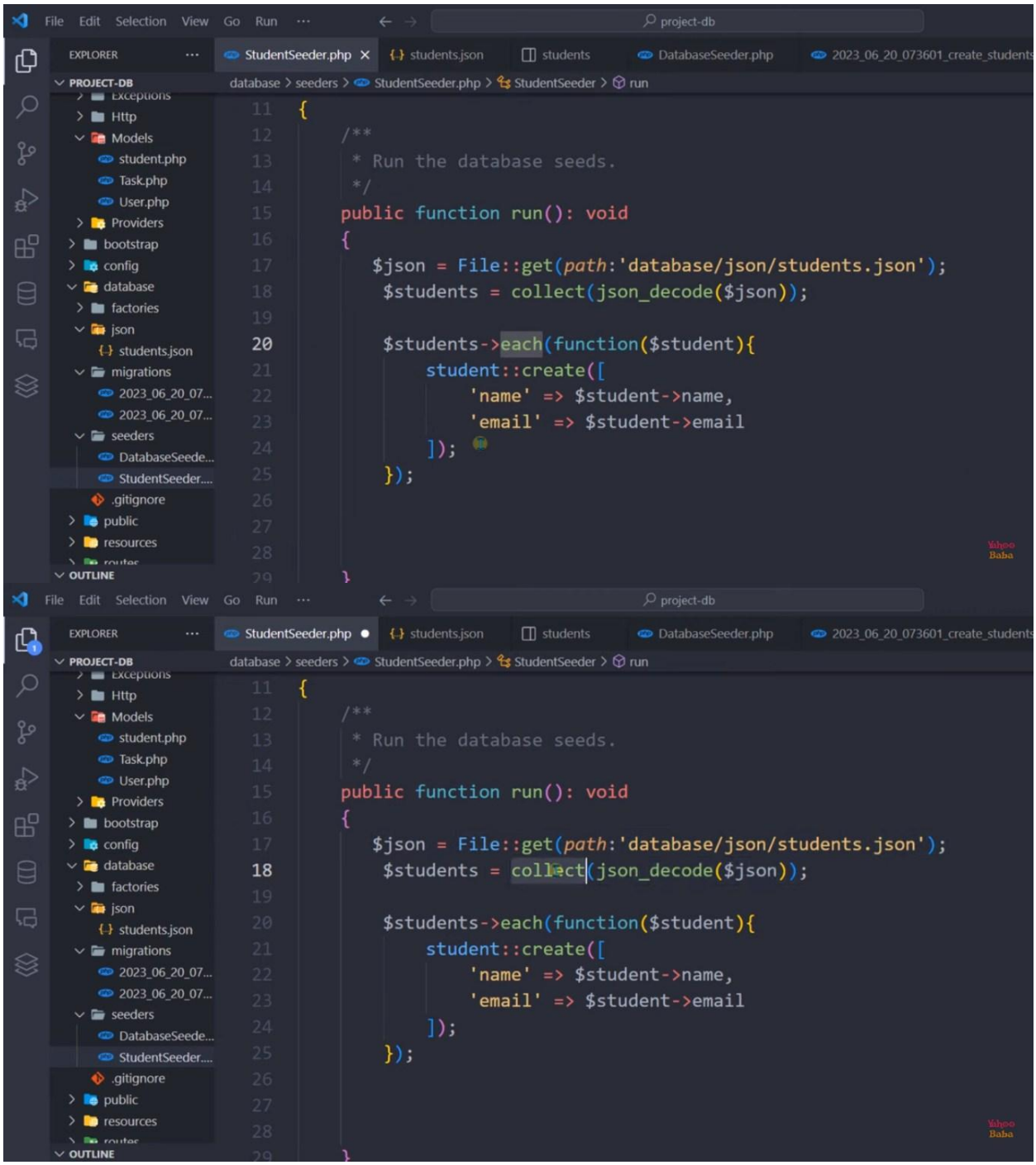
4. Seeders/DatabaseSeeder.php

```
$this->call([
    StudentSeeder::class
]);
```

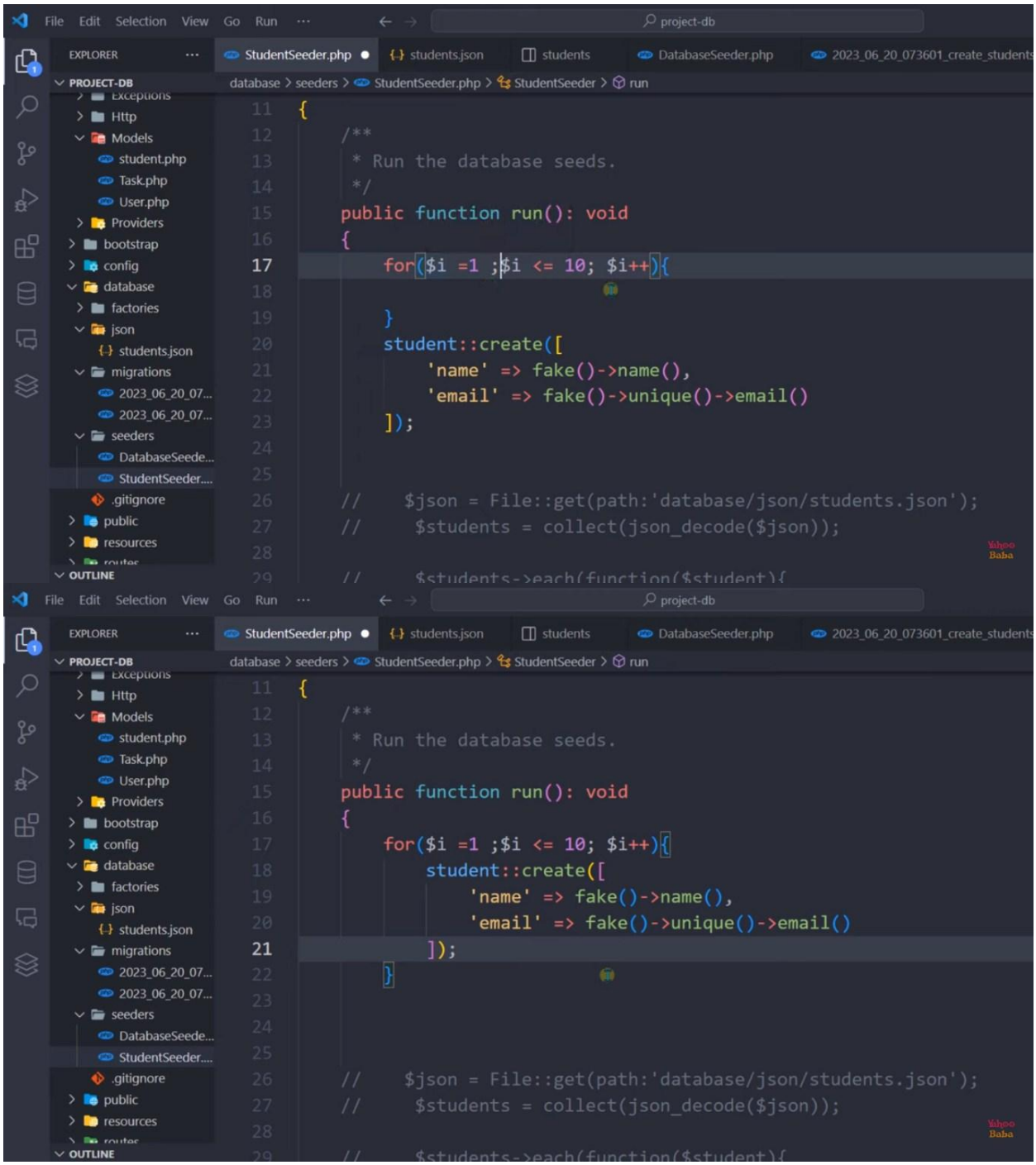
5. php artisan db:seed

Yahoo Baba www.yahoobaba.net





File Edit Selection View Go Run
project-db
EXPLORER
StudentSeeder.php
students.json
students
DatabaseSeeder.php
2023_06_20_073601_create_students
PROJECT-DB
database > seeders > StudentSeeder.php > StudentSeeder > run
Http
Models
student.php
Task.php
User.php
Providers
bootstrap
config
database
factories
json
students.json
migrations
seeders
DatabaseSeede...
StudentSeeder....
.gitignore
public
resources
OUTLINE
{
/**
* Run the database seeds.
*/
public function run(): void
{
$json = File::get(path:'database/json/students.json');
$students = collect(json_decode($json));
$students->each(function($student){
student::create([
'name' => $student->name,
'email' => $student->email
]);
});
}
Yahoo Baba


File Edit Selection View Go Run
project-db
EXPLORER
StudentSeeder.php
students.json
students
DatabaseSeeder.php
2023_06_20_073601_create_students
PROJECT-DB
database > seeders > StudentSeeder.php > StudentSeeder > run
Http
Models
student.php
Task.php
User.php
Providers
bootstrap
config
database
factories
json
students.json
migrations
seeders
DatabaseSeede...
StudentSeeder....
.gitignore
public
resources
OUTLINE
{
/**
* Run the database seeds.
*/
public function run(): void
{
for($i =1 ;$i <= 10; $i++){
}
student::create([
'name' => fake()->name(),
'email' => fake()->unique()->email()
]);
// $json = File::get(path:'database/json/students.json');
// $students = collect(json_decode($json));
Yahyaa Baba

File Edit Selection View Go Run
project-db
EXPLORER
StudentSeeder.php
students.json
students
DatabaseSeeder.php
2023_06_20_073601_create_students
PROJECT-DB
database > seeders > StudentSeeder.php > StudentSeeder > run
Http
Models
student.php
Task.php
User.php
Providers
bootstrap
config
database
factories
json
students.json
migrations
seeders
DatabaseSeede...
StudentSeeder....
.gitignore
public
resources
OUTLINE
{
/**
* Run the database seeds.
*/
public function run(): void
{
for($i =1 ;$i <= 10; $i++){
student::create([
'name' => fake()->name(),
'email' => fake()->unique()->email()
]);
}
// $json = File::get(path:'database/json/students.json');
// $students = collect(json_decode($json));
Yahyaa Baba

Laravel : Artisan Seeders Commands

```
php artisan make:seeder StudentSeeder
```

```
php artisan db:seed
```

```
php artisan db:seed --class=UserSeeder
```

```
php artisan db:seed --force
```

```
php artisan db:seed --class=UserSeeder --force
```


Laravel : Steps to work in Factory

1. `php artisan make:model student`
2. `php artisan make:factory StudentFactory`
3.
```
class StudentFactory extends Factory
{
    public function definition(): array
    {
        return[
            'name' => fake()->name(),
            'email' => fake()->email()
        ];
    }
}
```
4. Seeders/DatabaseSeeder.php
```
use App\Models\student;

student::factory()->count(5)->create();
```
5. `php artisan db:seed`

Laravel : Artisan Factory Commands

```
php artisan make:factory StudentFactory
```

```
php artisan make:factory StudentFactory --model=Student
```
(Factory & Model)

```
php artisan make:model student -f
```
(Model & Factory)

```
php artisan db:seed
```

```
php artisan db:seed --class=UserSeeder
```

```
php artisan migrate:fresh --seed
```
(Migration & Seeding)

Laravel : Steps to work in Query Builder

1. ```
   php artisan make:controller UserController
   ```

2. ```
   use Illuminate\Support\Facades\DB;

   class UserController extends Controller
   {
       public function show()
       {
           $users = DB::table('users')->get();
           return $users;
       }
   }
   ```

   - Read – get()
   - Insert – insert()
   - Update – update()
   - Delete – delete()

3. ```
   use App\Http\Controllers\UserController;

   Route::get('/user', [UserController::class, 'show']);
   ```

   http://localhost/user
   ```

Laravel : Read Data with Query Builder

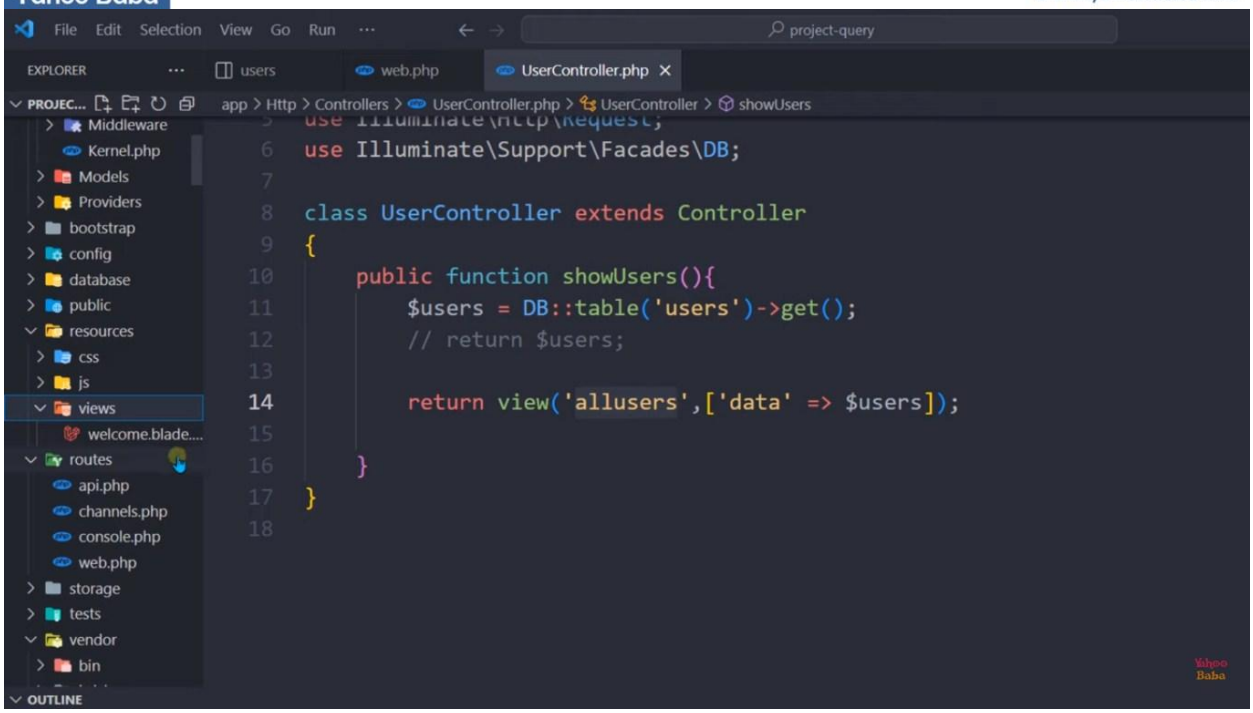
```
SELECT * FROM users
DB::table('users')->get()
```

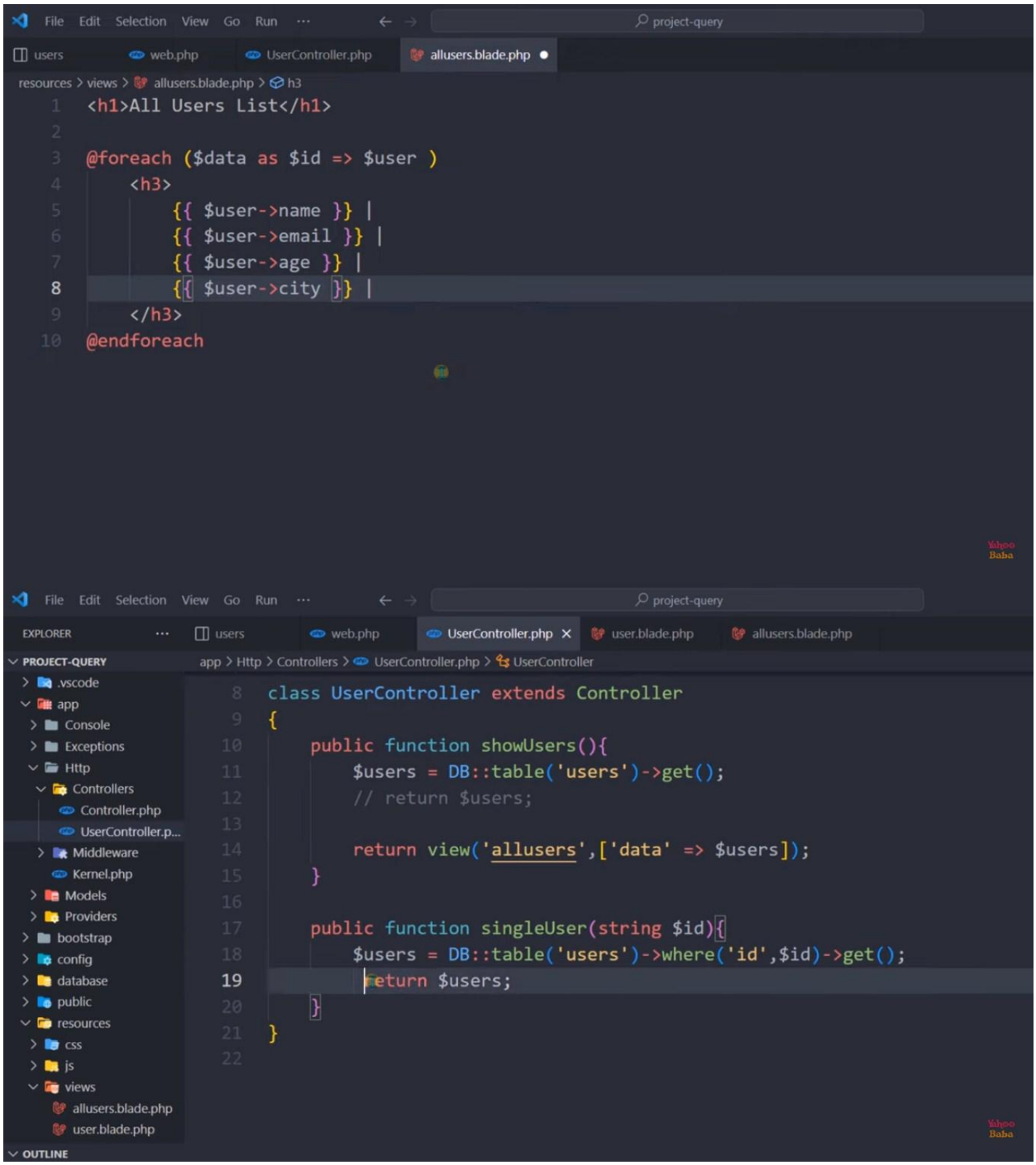
```
SELECT name, city FROM users
DB::table('users') ->select('name', 'city')->get()
```

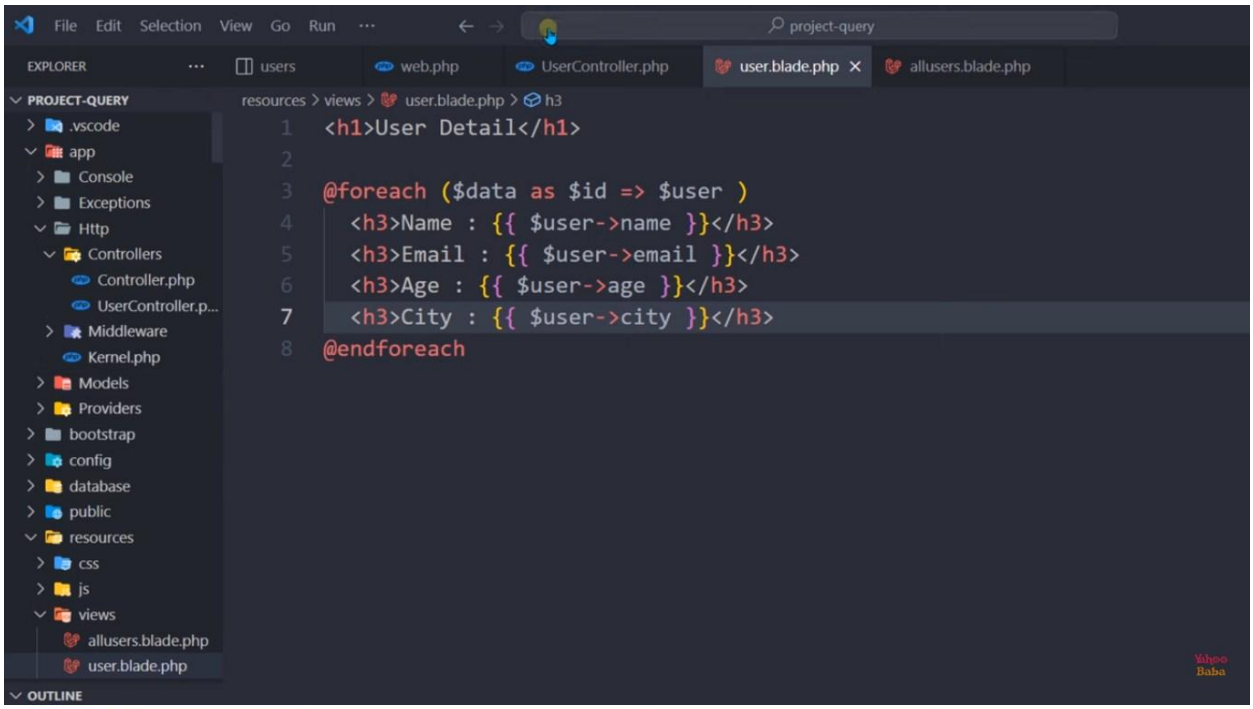
```
SELECT * FROM users WHERE city = 'goa';
DB::table('users') ->where('city', '=', 'goa')->get()
DB::table('users') ->where('city', '=', 'goa') ->where('age', '>', 18)->get()
DB::table('users') ->where('city', '=', 'goa') ->orWhere('age', '>', 18)->get()
```

- whereBetween()
- whereIn()
- whereNull()
- whereMonth()
- whereDay()
- whereYear()
- whereTime()

Yahoo Baba www.yahoobaba.net




File Edit Selection View Go Run
project-query
users
web.php
UserController.php
allusers.blade.php
resources > views > allusers.blade.php > h3
<h1>All Users List</h1>
@foreach ($data as $id => $user)
<h3>
{{ $user->name }} |
{{ $user->email }} |
{{ $user->age }} |
{{ $user->city }} |
</h3>
@endforeach
File Edit Selection View Go Run
project-query
EXPLORER
users
web.php
UserController.php
user.blade.php
allusers.blade.php
PROJECT-QUERY
app > Http > Controllers > UserController.php > UserController
.vscode
app
Console
Exceptions
Http
Controllers
Controller.php
UserController.p...
Middleware
Kernel.php
Models
Providers
bootstrap
config
database
public
resources
css
js
views
allusers.blade.php
user.blade.php
OUTLINE
class UserController extends Controller
{
public function showUsers(){
$users = DB::table('users')->get();
// return $users;
return view('allusers',['data' => $users]);
}
public function singleUser(string $id){
$users = DB::table('users')->where('id',$id)->get();
return $users;
}
}



Laravel : Query Builder

```
use Illuminate\Support\Facades\DB;
```

Add New Record

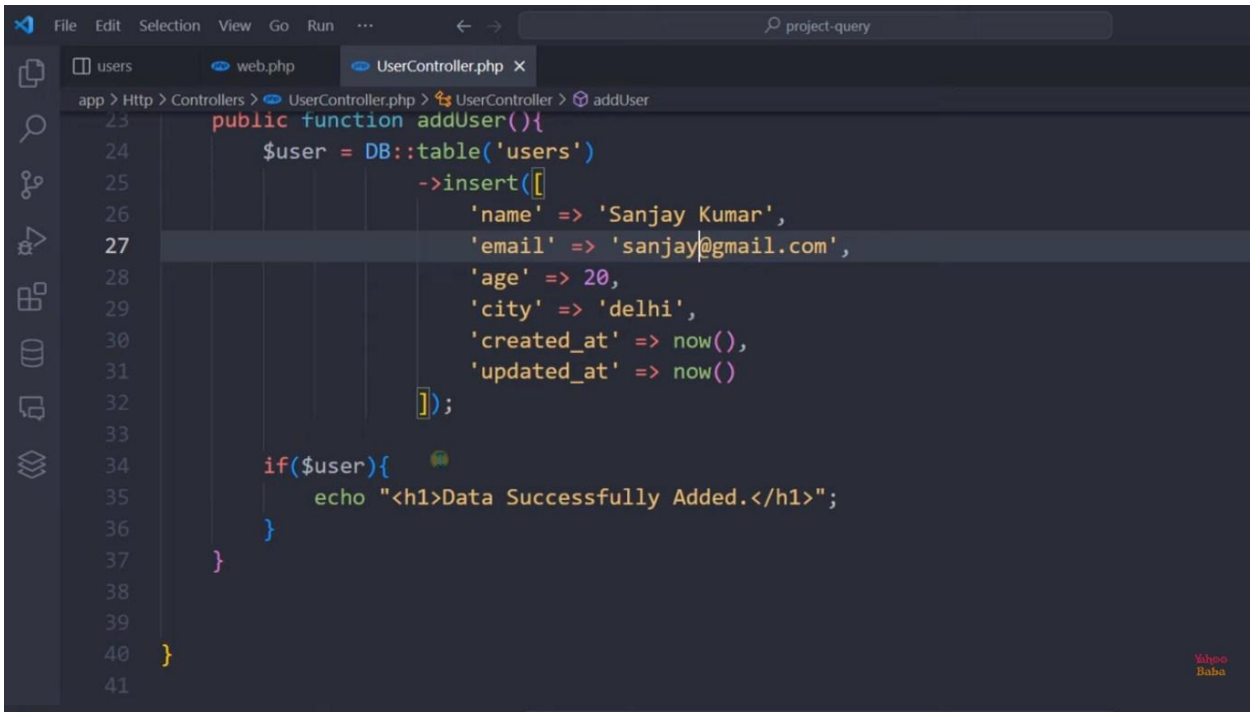
```
DB::table('users')->insert([
    'name' => 'Yahoo Baba',
    'email' => 'yahoobaba@email.com',
]);
```

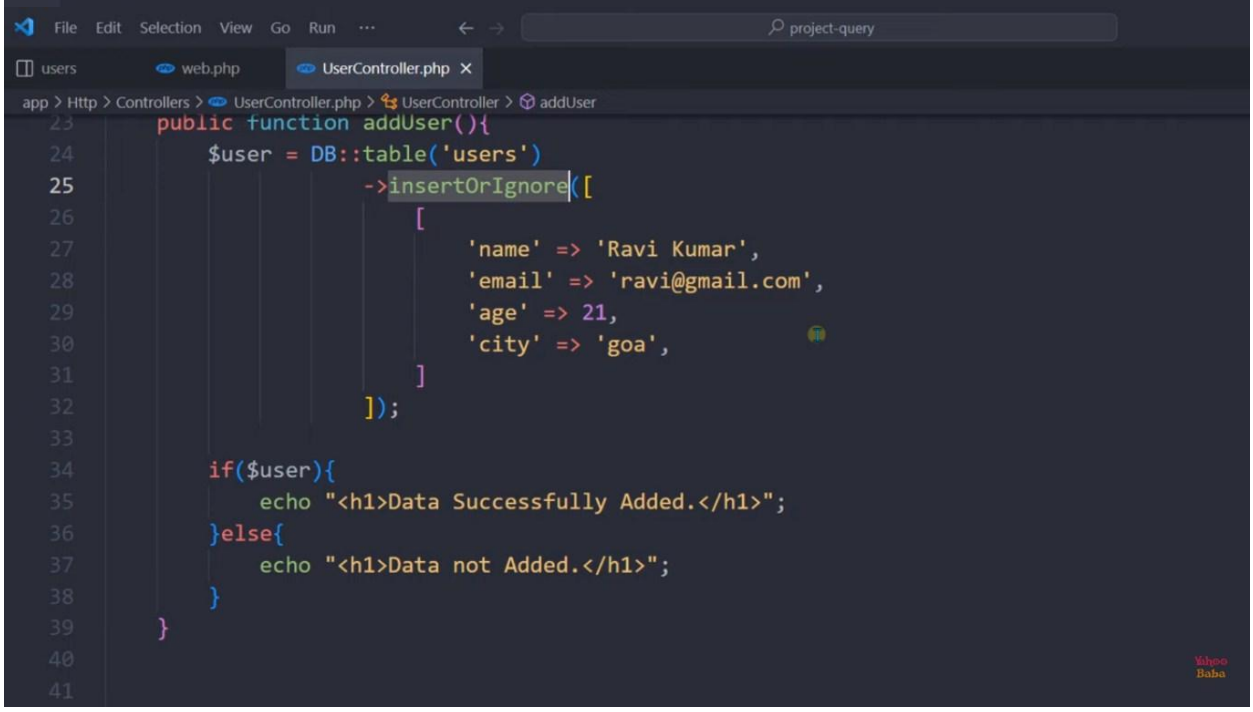
Delete Record

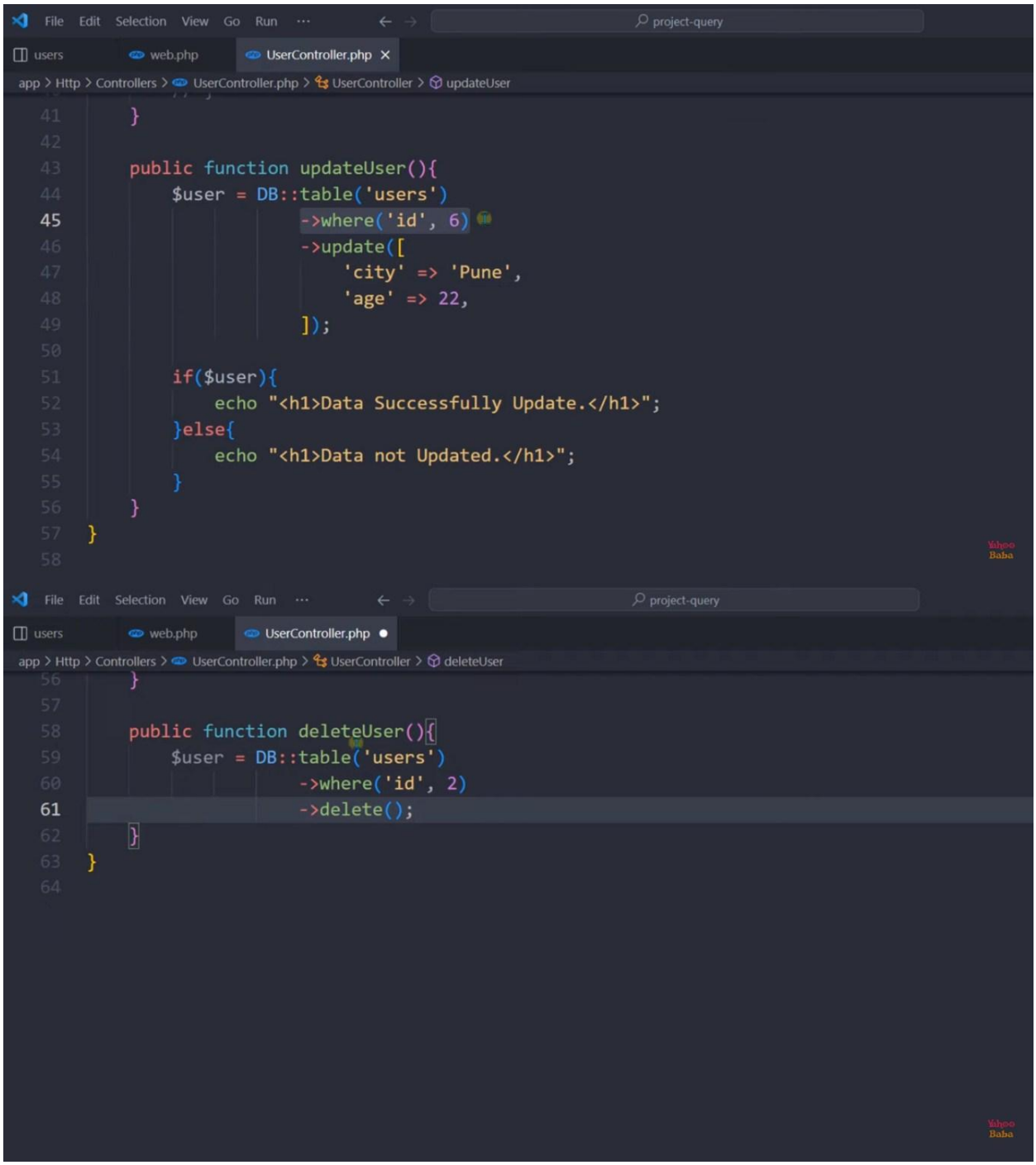
```
DB::table('users')
    ->where('id', 1)
    ->delete();
```

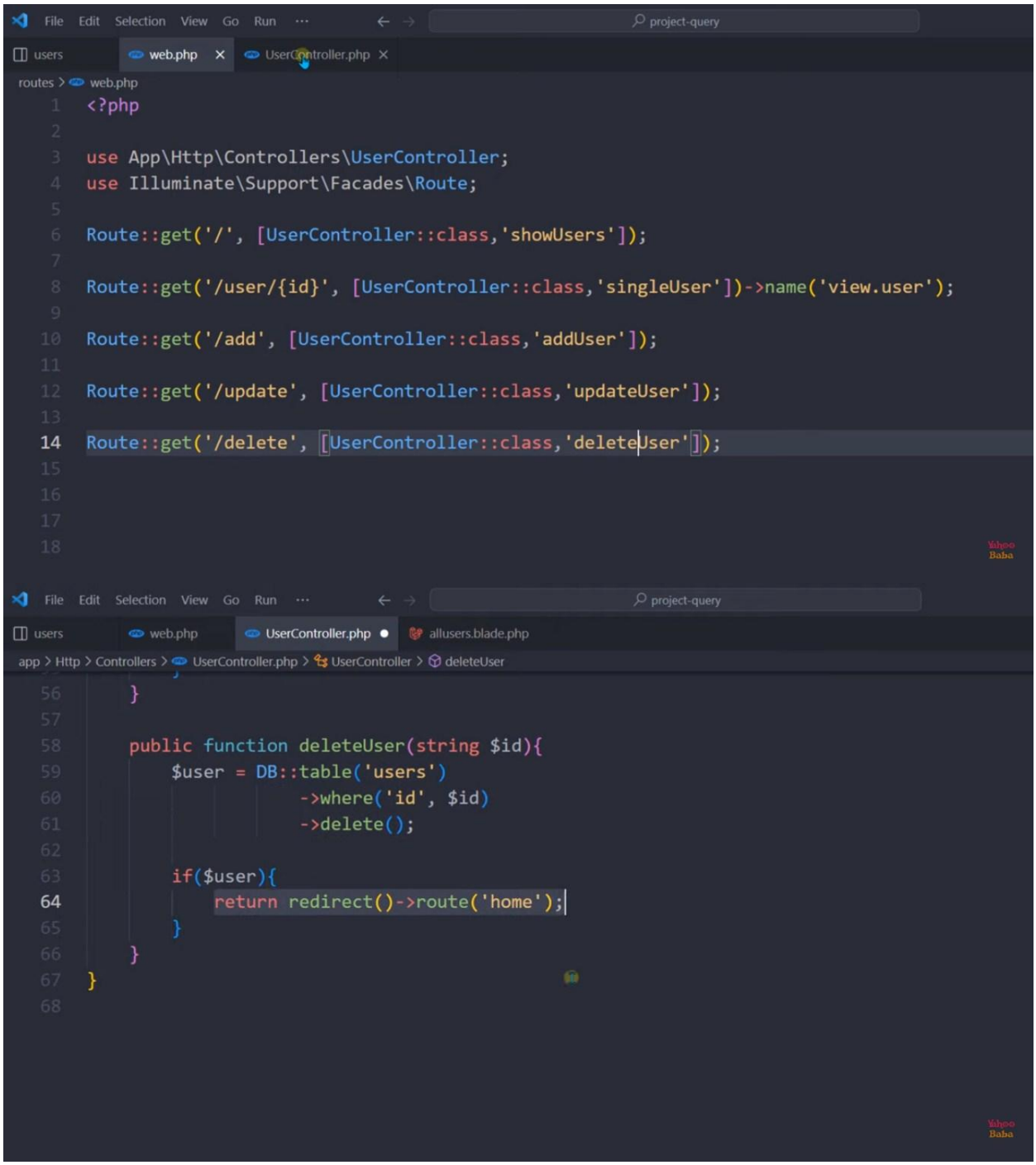
Update Existing Record

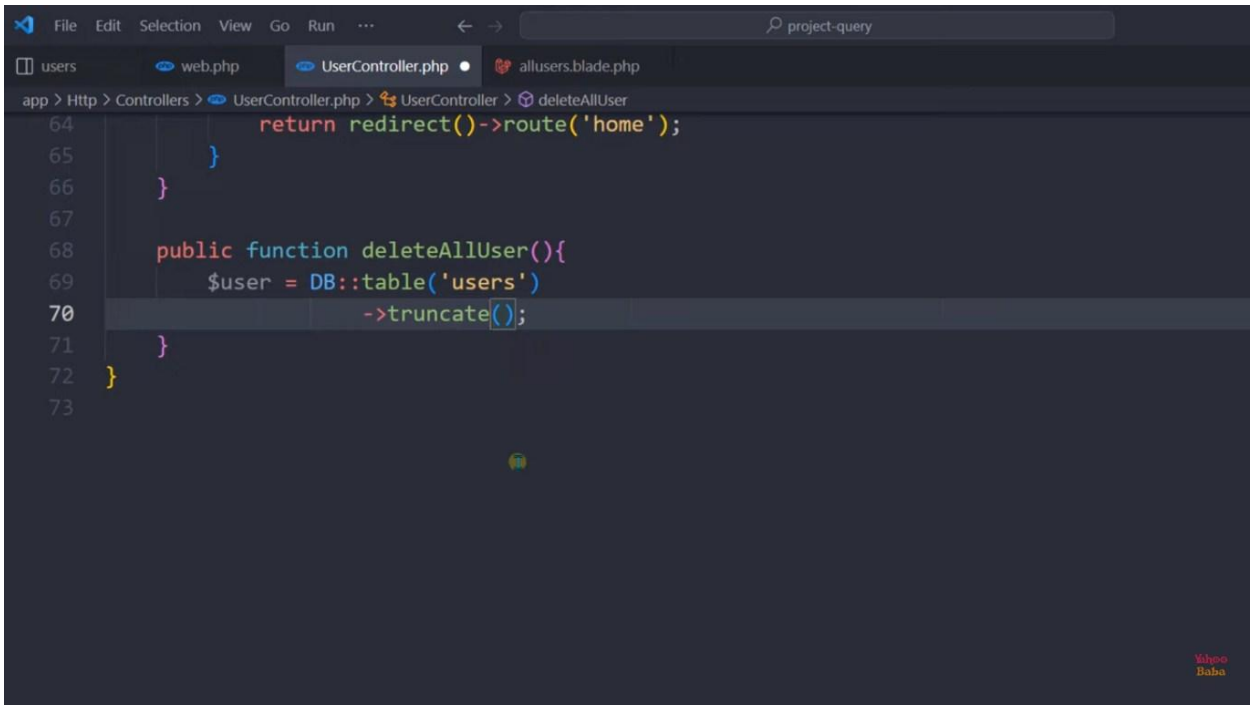
```
DB::table('users')
    ->where('id', 1)
    ->update([
        'city' => 'Agra',
    ]);
```


File Edit Selection View Go Run ···
project-query
users
web.php
UserController.php
app > Http > Controllers > UserController.php > UserController > addUser
public function addUser(){
$user = DB::table('users')
->insert([
'name' => 'Sanjay Kumar',
'email' => 'sanjay@gmail.com',
'age' => 20,
'city' => 'delhi',
'created_at' => now(),
'updated_at' => now()
]);
if($user){
echo "<h1>Data Successfully Added.</h1>";
}
}
}

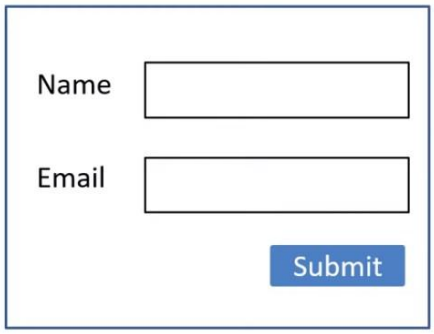

File Edit Selection View Go Run ···
project-query
users
web.php
UserController.php
app > Http > Controllers > UserController.php > UserController > addUser
public function addUser(){
$user = DB::table('users')
->insertOrIgnore([
[
'name' => 'Ravi Kumar',
'email' => 'ravi@gmail.com',
'age' => 21,
'city' => 'goa',
]
]);
if($user){
echo "<h1>Data Successfully Added.</h1>";
}else{
echo "<h1>Data not Added.</h1>";
}
}


File Edit Selection View Go Run ···
project-query
users
web.php
UserController.php
app > Http > Controllers > UserController.php > UserController > updateUser
}
public function updateUser(){
$user = DB::table('users')
->where('id', 6)
->update([
'city' => 'Pune',
'age' => 22,
]);
if($user){
echo "<h1>Data Successfully Update.</h1>";
}else{
echo "<h1>Data not Updated.</h1>";
}
}
}
File Edit Selection View Go Run ···
project-query
users
web.php
UserController.php
app > Http > Controllers > UserController.php > UserController > deleteUser
}
public function deleteUser(){
$user = DB::table('users')
->where('id', 2)
->delete();
}
}


File Edit Selection View Go Run
project-query
users
web.php
UserController.php
routes > web.php
<?php
use App\Http\Controllers\UserController;
use Illuminate\Support\Facades\Route;
Route::get('/', [UserController::class,'showUsers']);
Route::get('/user/{id}', [UserController::class,'singleUser'])->name('view.user');
Route::get('/add', [UserController::class,'addUser']);
Route::get('/update', [UserController::class,'updateUser']);
Route::get('/delete', [UserController::class,'deleteUser']);
Yahoo Baba
File Edit Selection View Go Run
project-query
users
web.php
UserController.php
allusers.blade.php
app > Http > Controllers > UserController.php > UserController > deleteUser
}
public function deleteUser(string $id){
$user = DB::table('users')
->where('id', $id)
->delete();
if($user){
return redirect()->route('home');
}
}
}
Yahoo Baba


File Edit Selection View Go Run ···
project-query
users
web.php
UserController.php
allusers.blade.php
app > Http > Controllers > UserController.php > UserController > deleteAllUser
return redirect()->route('home');
}
}
public function deleteAllUser(){
$user = DB::table('users')
->truncate();
}
}
Yahoo Baba

Laravel : Query Builder with HTML Forms




```
<form action="/addUser" method="POST">
    @csrf
    <input type="text" name="username">
    <input type="text" name="userpassword">
    <input type="submit">
</form>
```

```
Route::post('/add', 'addUser')
```

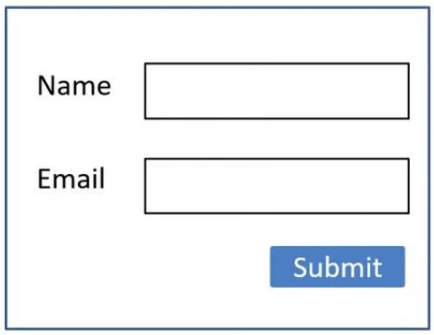
```
public function addUser(Request $req){

}
```

Cross-site Request Forgeries

malicious exploit

Laravel : Query Builder with HTML Forms




```
<form action="/addUser" method="POST">
    @csrf
    <input type="text" name="username">
    <input type="text" name="userpassword">
    <input type="submit">
</form>
```

```
Route::post('/add', 'addUser')
```

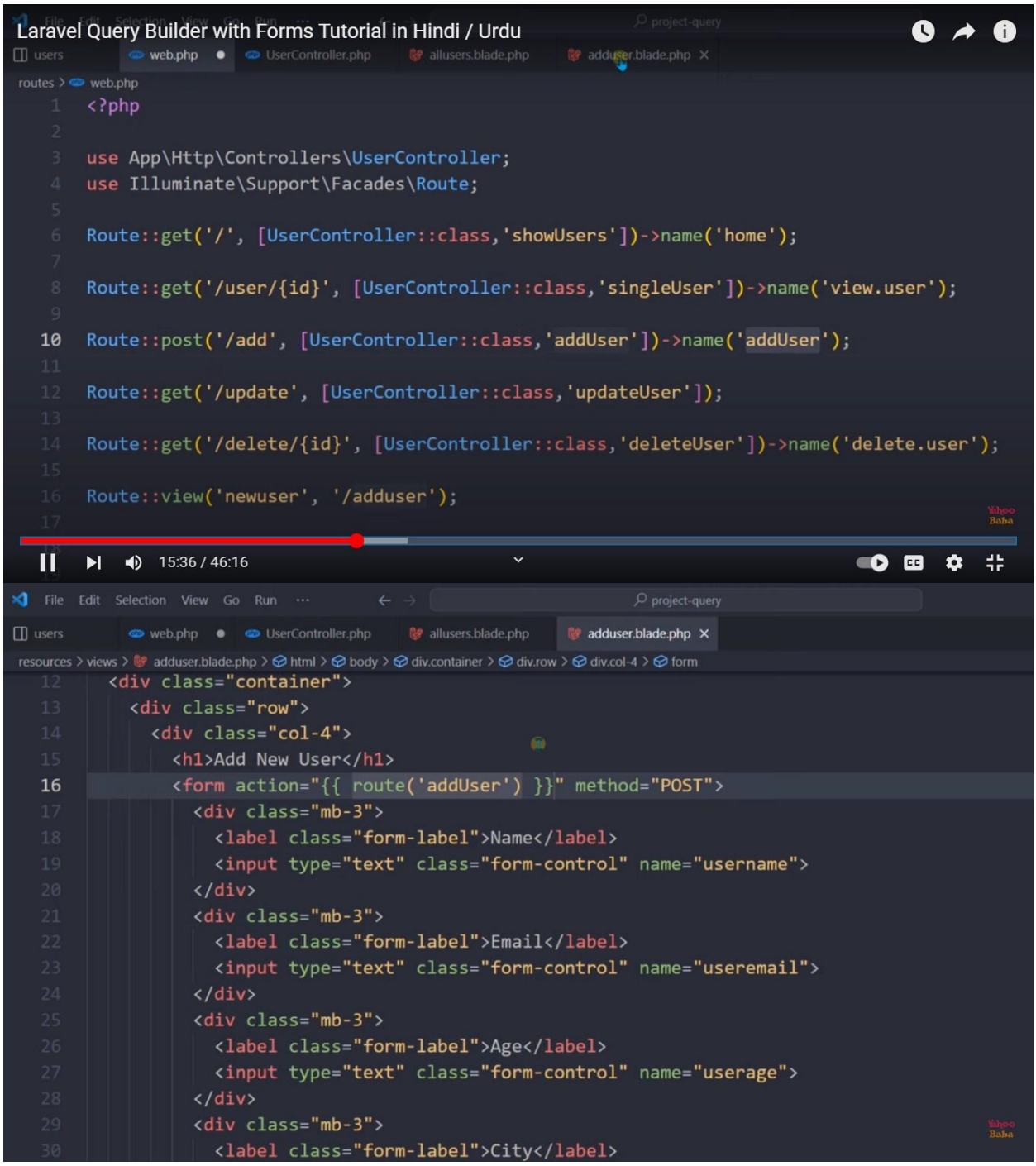
```
public function addUser(Request $req){

}
```

Cross-site Request Forgeries

malicious exploit

Yahoo Baba
www.yahoobaba.net


Laravel Query Builder with Forms Tutorial in Hindi / Urdu
users
web.php
UserController.php
allusers.blade.php
adduser.blade.php
routes > web.php
<?php
use App\Http\Controllers\UserController;
use Illuminate\Support\Facades\Route;
Route::get('/', [UserController::class,'showUsers'])->name('home');
Route::get('/user/{id}', [UserController::class,'singleUser'])->name('view.user');
Route::post('/add', [UserController::class,'addUser'])->name('addUser');
Route::get('/update', [UserController::class,'updateUser']);
Route::get('/delete/{id}', [UserController::class,'deleteUser'])->name('delete.user');
Route::view('newuser', '/adduser');
15:36 / 46:16
File Edit Selection View Go Run
project-query
resources > views > adduser.blade.php > html > body > div.container > div.row > div.col-4 > form

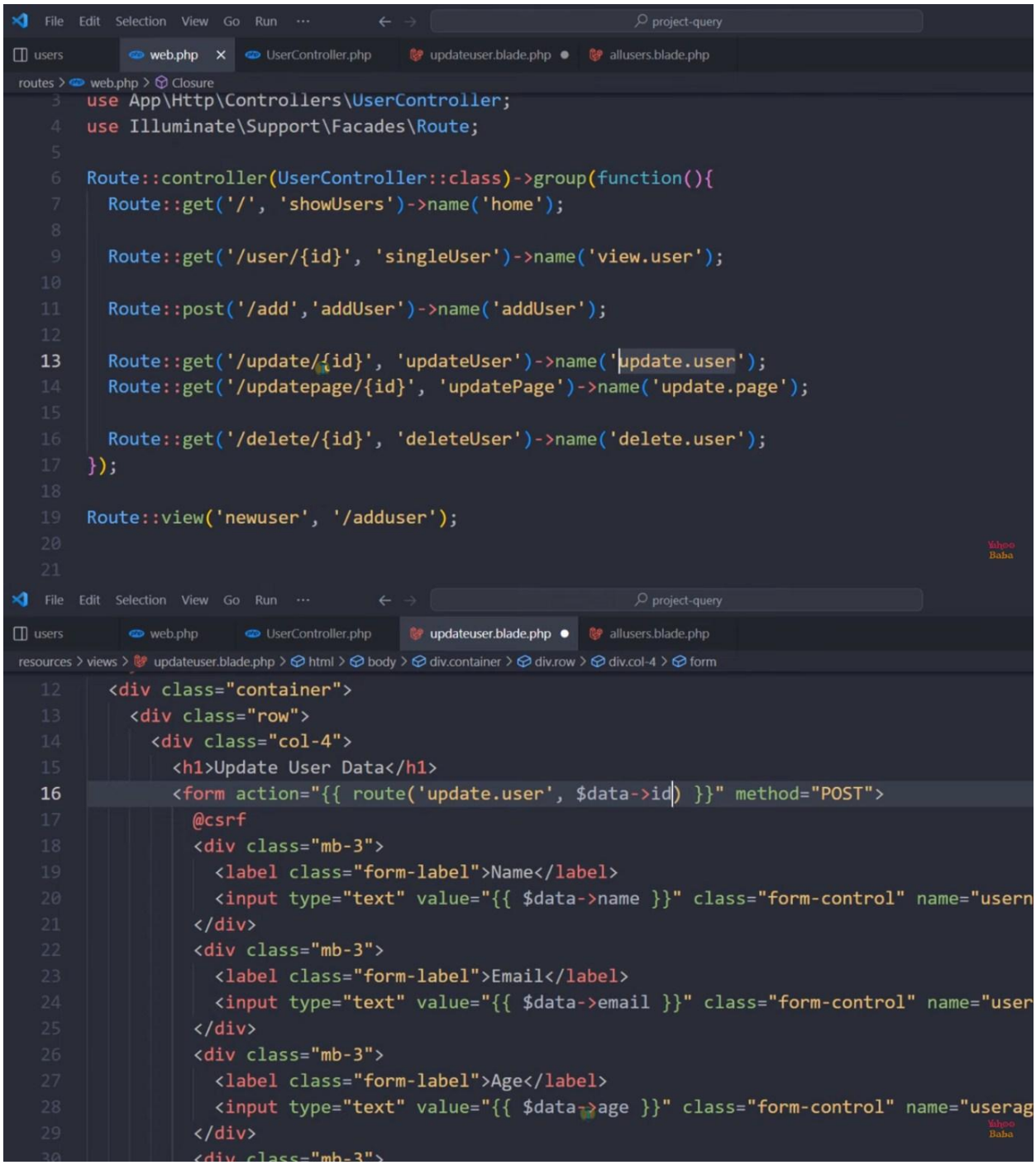
<h1>Add New User</h1>
<form action="{{ route('addUser') }}" method="POST">

<label class="form-label">Name</label>
<input type="text" class="form-control" name="username">

<label class="form-label">Email</label>
<input type="text" class="form-control" name="useremail">

<label class="form-label">Age</label>
<input type="text" class="form-control" name="userage">

<label class="form-label">City</label>

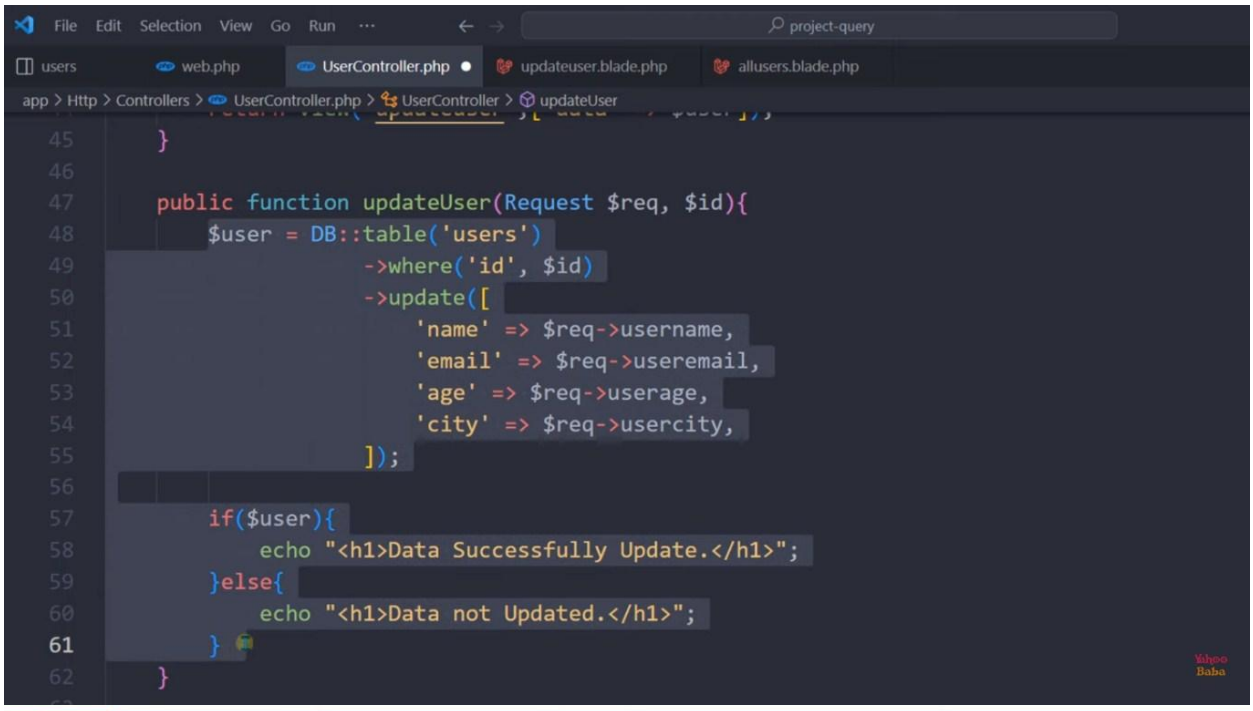

File Edit Selection View Go Run ···
project-query
users
web.php
UserController.php
updateuser.blade.php
allusers.blade.php
routes > web.php > Closure
use App\Http\Controllers\UserController;
use Illuminate\Support\Facades\Route;
Route::controller(UserController::class)->group(function(){
Route::get('/', 'showUsers')->name('home');
Route::get('/user/{id}', 'singleUser')->name('view.user');
Route::post('/add','addUser')->name('addUser');
Route::get('/update/{id}', 'updateUser')->name('update.user');
Route::get('/updatepage/{id}', 'updatePage')->name('update.page');
Route::get('/delete/{id}', 'deleteUser')->name('delete.user');
});
Route::view('newuser', '/adduser');
File Edit Selection View Go Run ···
project-query
users
web.php
UserController.php
updateuser.blade.php
allusers.blade.php
resources > views > updateuser.blade.php > html > body > div.container > div.row > div.col-4 > form

<h1>Update User Data</h1>
<form action="{{ route('update.user', $data->id) }}" method="POST">
@csrf

<label class="form-label">Name</label>
<input type="text" value="{{ $data->name }}" class="form-control" name="usern

<label class="form-label">Email</label>
<input type="text" value="{{ $data->email }}" class="form-control" name="user

<label class="form-label">Age</label>
<input type="text" value="{{ $data->age }}" class="form-control" name="userag



Laravel : Router Methods

Route::get(); → Read

Route::post(); → Add, Update, Delete

Route::put(); → Update

Route::patch(); → Update

Route::delete(); → Delete

Route::options();

```
<form method="POST">
    @csrf
    @method('PUT')

</form>
```

```
Route::match(['get', 'post'], '/', function () {
  // ...
});
```

```
Route::any('/', function () {
  // ...
});
```

Laravel : Pagination Methods

1. Paginate()

```
DB::table('users')->paginate(5)
```

< 1 2 3 4 >

2. simplePaginate()

```
DB::table('users')->simplePaginate(5)
```

Previous Next

Blade File :

```
{{ $data->links() }}
```

3. cursorPaginate()

```
DB::table('users')->orderBy('id')->cursorPaginate(5)
```


Laravel : Form Validation

Name

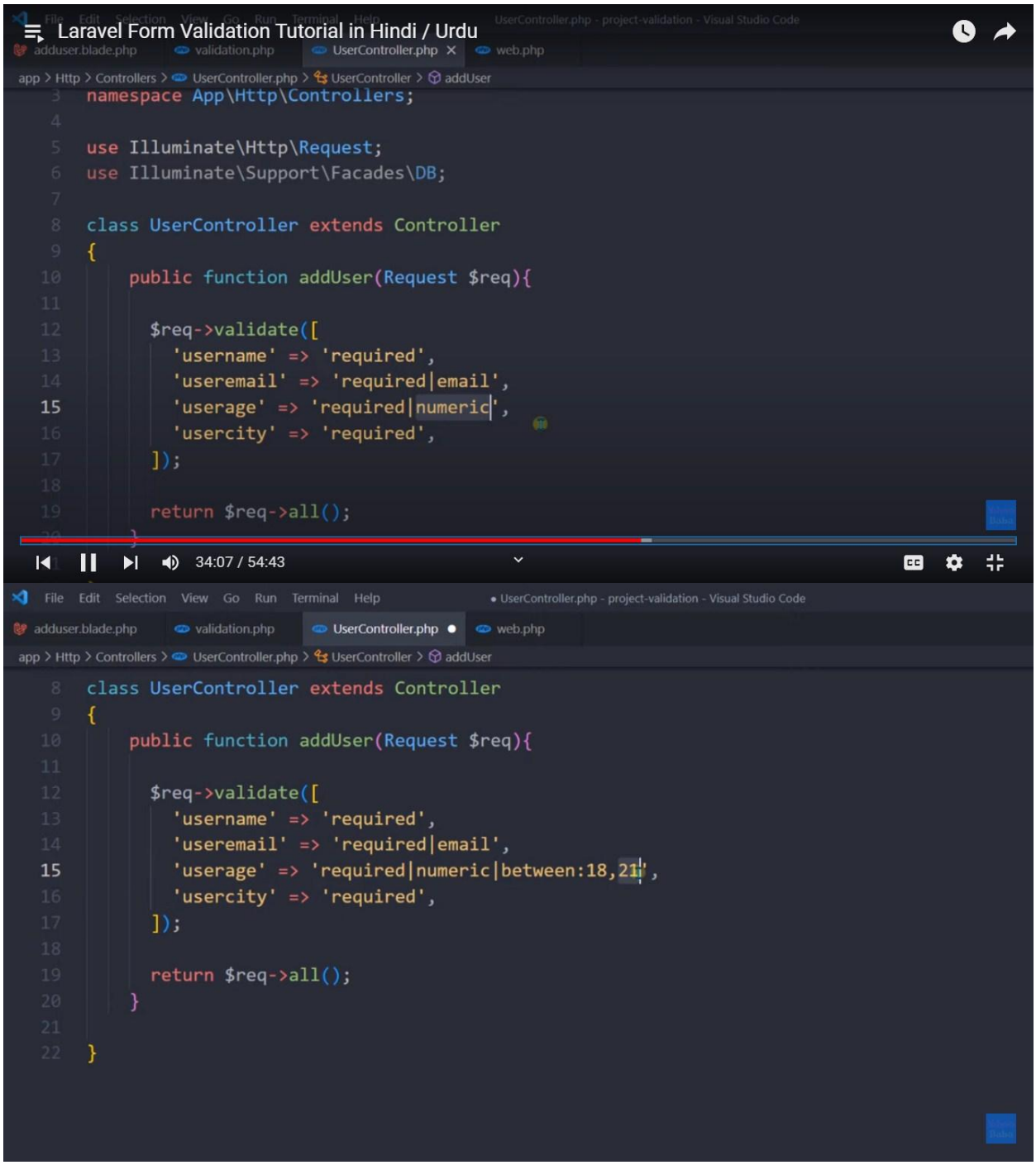
Email

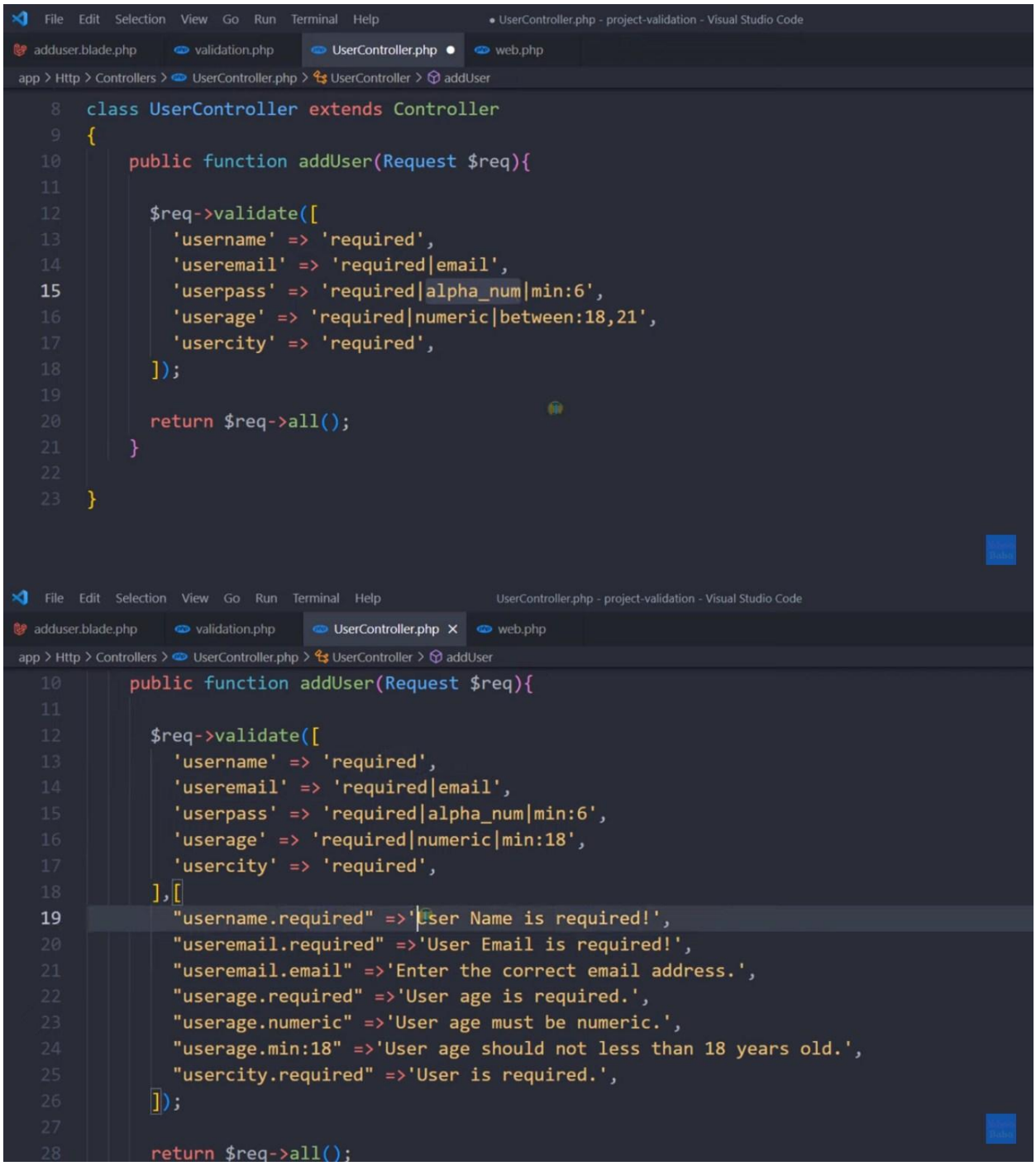
Submit

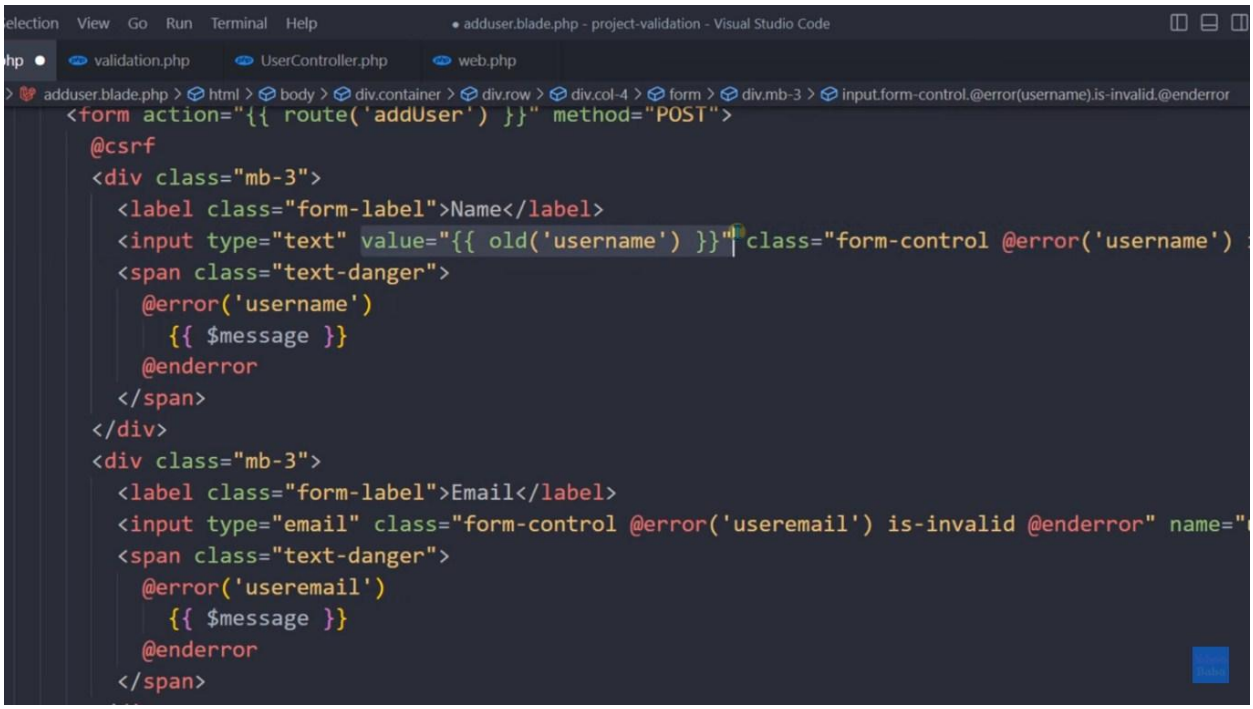
```
public function addUser(Request $req){

    $req->validate([
        'username' => 'required',
        'useremail' => 'required|email',
    ]);

}
```


Laravel Form Validation Tutorial in Hindi / Urdu
UserController.php - project-validation - Visual Studio Code
adduser.blade.php
validation.php
UserController.php
web.php
app > Http > Controllers > UserController.php > UserController > addUser
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
class UserController extends Controller
{
public function addUser(Request $req){
$req->validate([
'username' => 'required',
'useremail' => 'required|email',
'userage' => 'required|numeric',
'usercity' => 'required',
]);
return $req->all();
34:07 / 54:43
File Edit Selection View Go Run Terminal Help
• UserController.php - project-validation - Visual Studio Code
'userage' => 'required|numeric|between:18,21',


File Edit Selection View Go Run Terminal Help
• UserController.php - project-validation - Visual Studio Code
adduser.blade.php
validation.php
UserController.php
web.php
app > Http > Controllers > UserController.php > UserController > addUser
class UserController extends Controller
{
public function addUser(Request $req){
$req->validate([
'username' => 'required',
'useremail' => 'required|email',
'userpass' => 'required|alpha_num|min:6',
'userage' => 'required|numeric|between:18,21',
'usercity' => 'required',
]);
return $req->all();
}
}
File Edit Selection View Go Run Terminal Help
UserController.php - project-validation - Visual Studio Code
adduser.blade.php
validation.php
UserController.php
web.php
app > Http > Controllers > UserController.php > UserController > addUser
public function addUser(Request $req){
$req->validate([
'username' => 'required',
'useremail' => 'required|email',
'userpass' => 'required|alpha_num|min:6',
'userage' => 'required|numeric|min:18',
'usercity' => 'required',
],[
"username.required" =>'User Name is required!',
"useremail.required" =>'User Email is required!',
"useremail.email" =>'Enter the correct email address.',
"userage.required" =>'User age is required.',
"userage.numeric" =>'User age must be numeric.',
"userage.min:18" =>'User age should not less than 18 years old.',
"usercity.required" =>'User is required.',
]);
return $req->all();


Selection View Go Run Terminal Help
• adduser.blade.php - project-validation - Visual Studio Code
validation.php
UserController.php
web.php
adduser.blade.php > html > body > div.container > div.row > div.col-4 > form > div.mb-3 > input.form-control.@error(username).is-invalid.@enderror
<form action="{{ route('addUser') }}" method="POST">
@csrf

<label class="form-label">Name</label>
<input type="text" value="{{ old('username') }}" class="form-control @error('username')

@error('username')
{{ $message }}
@enderror

<label class="form-label">Email</label>
<input type="email" class="form-control @error('useremail') is-invalid @enderror" name="

@error('useremail')
{{ $message }}
@enderror