

# Laravel

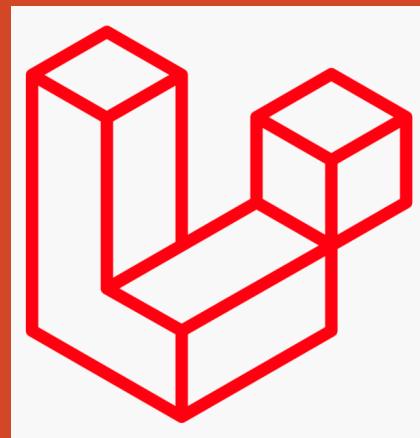




# Laravel 01

## Bina Projek

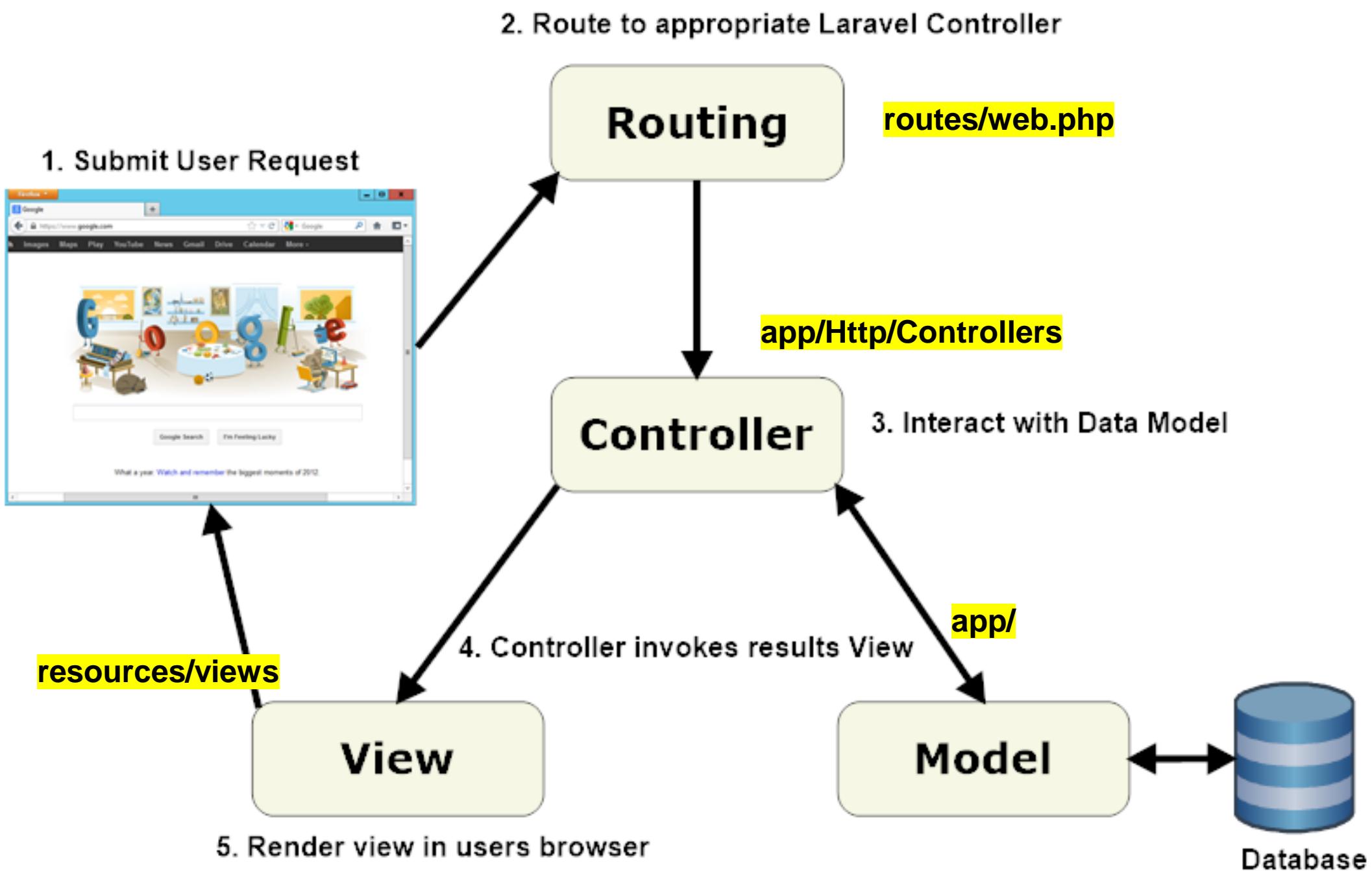
### Laravel Pemula



# Module Outcome

## Day 1

- | Day 1                      | What will be needed |
|----------------------------|---------------------|
| • Create Laravel Project   | • Laragon           |
| • Working with Database.   | • phpMyAdmin        |
| • Working with Model       | • VS Code           |
| • Working with Controllers |                     |
| • Working with Views       |                     |



# Create Laravel First Project

Install Laravel by issuing the Composer create-project command in your terminal

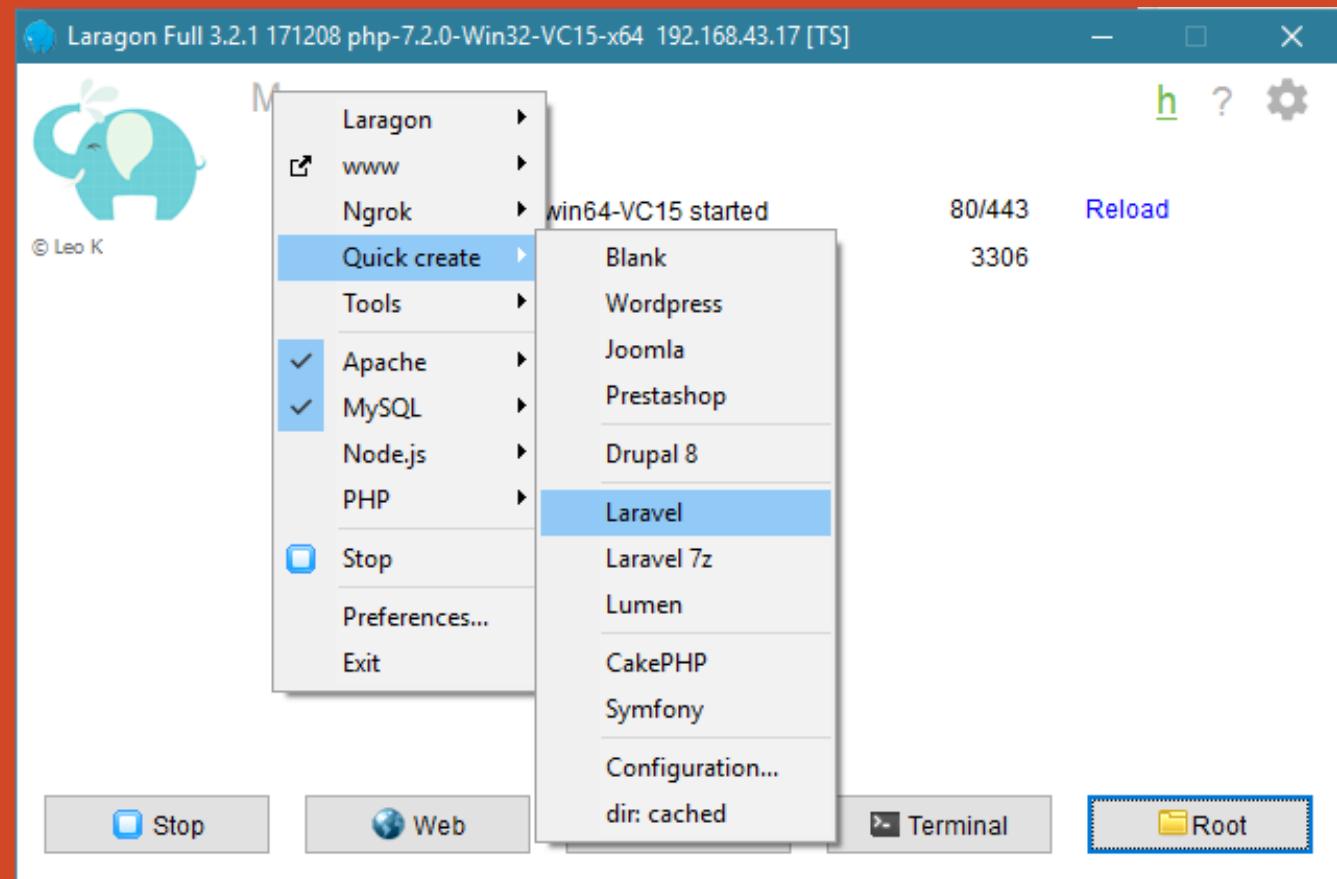
```
composer create-project --prefer-dist laravel/laravel project-name
```

# Laravel

[DOCS](#)[LARACASTS](#)[NEWS](#)[BLOG](#)[NOVA](#)[FORGE](#)[VAPOR](#)[GITHUB](#)

# laragon.org

With Laragon?  
Never been EASIER.



# Laragon

## What is Laragon?

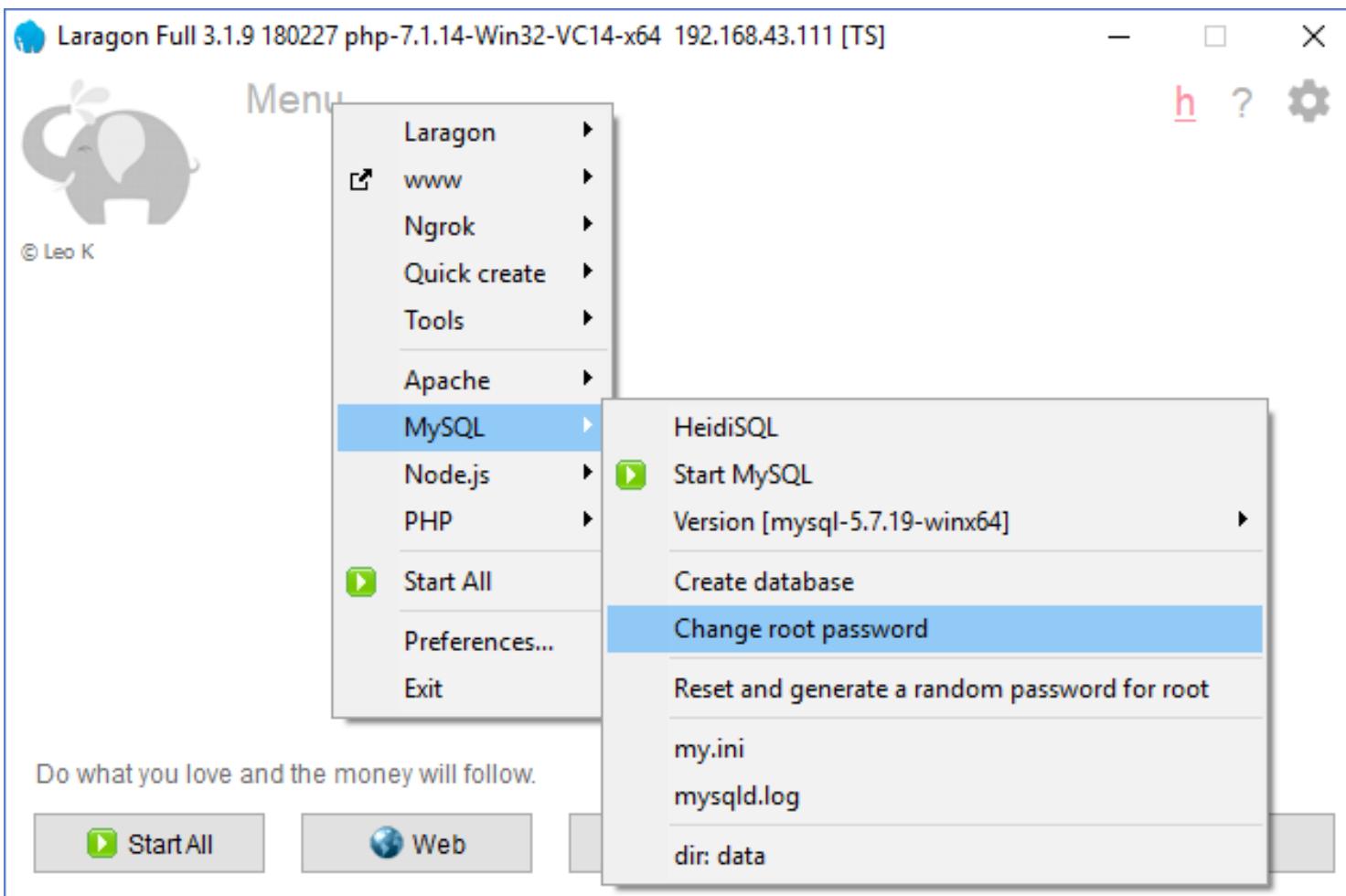
Laragon is a portable, isolated, fast & powerful universal development environment for PHP, Node.js, Python, Java, Go, Ruby. It is fast, lightweight, easy-to-use and easy-to-extend.

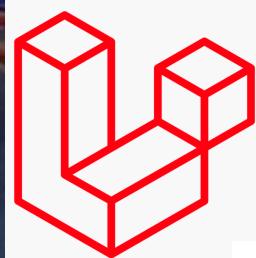
## Why Laragon?

Laragon is great for building and managing modern web applications. It is focused on performance - designed around stability, simplicity, flexibility and freedom.

Laragon improves web development. Developers all over the world are using Laragon to make apps quickly and easily. It is used by thousands of developers with loves. You can check out the Testimonials to see how users think of Laragon and it's features page for more details.

# Change DB root password





# Laravel t02

Cara mudah tambah laman  
login & daftar pengguna



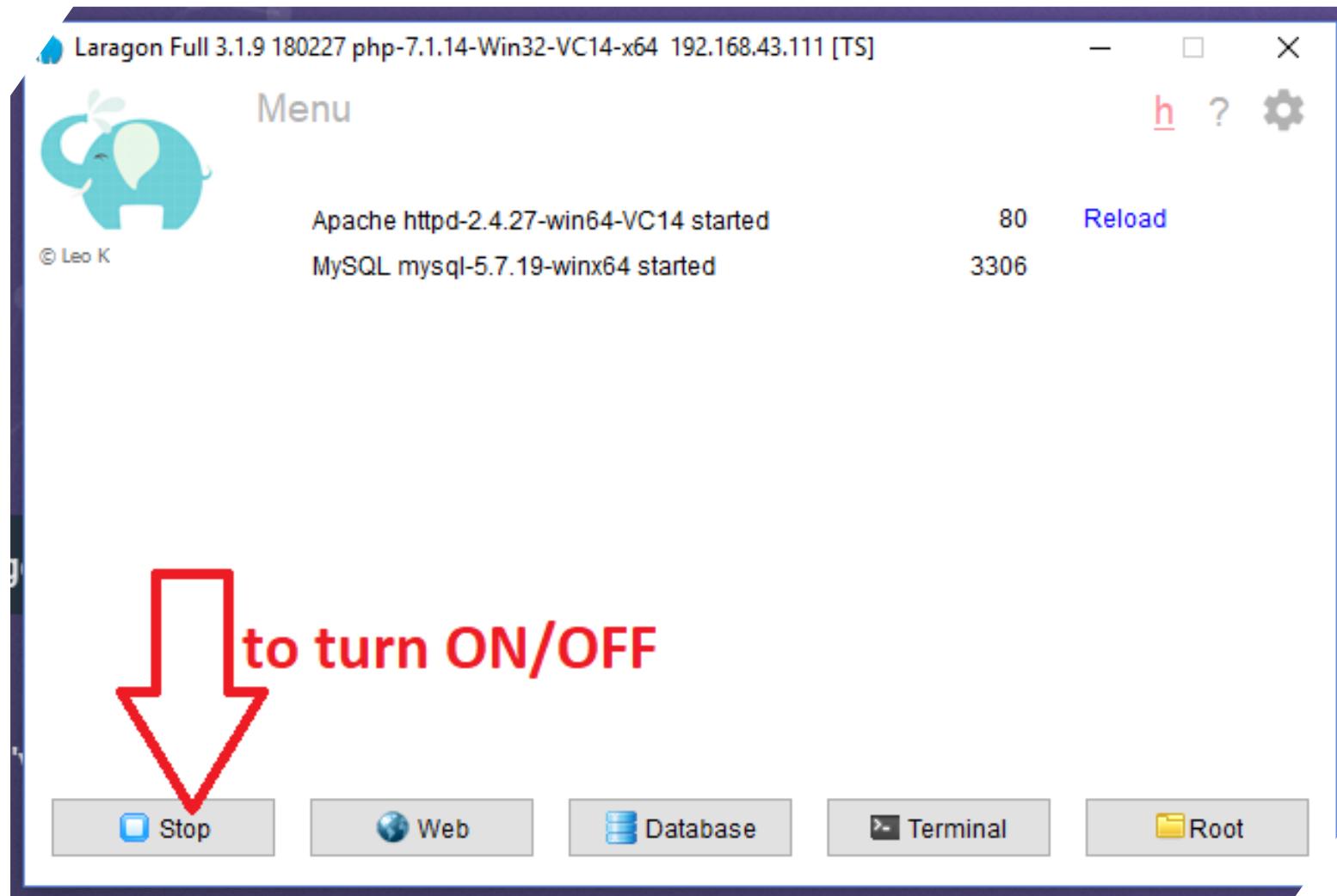
# Laravel t02

## Tambah laman login & daftar pengguna

[fstm.kuis.edu.my/blog/laravel](http://fstm.kuis.edu.my/blog/laravel)

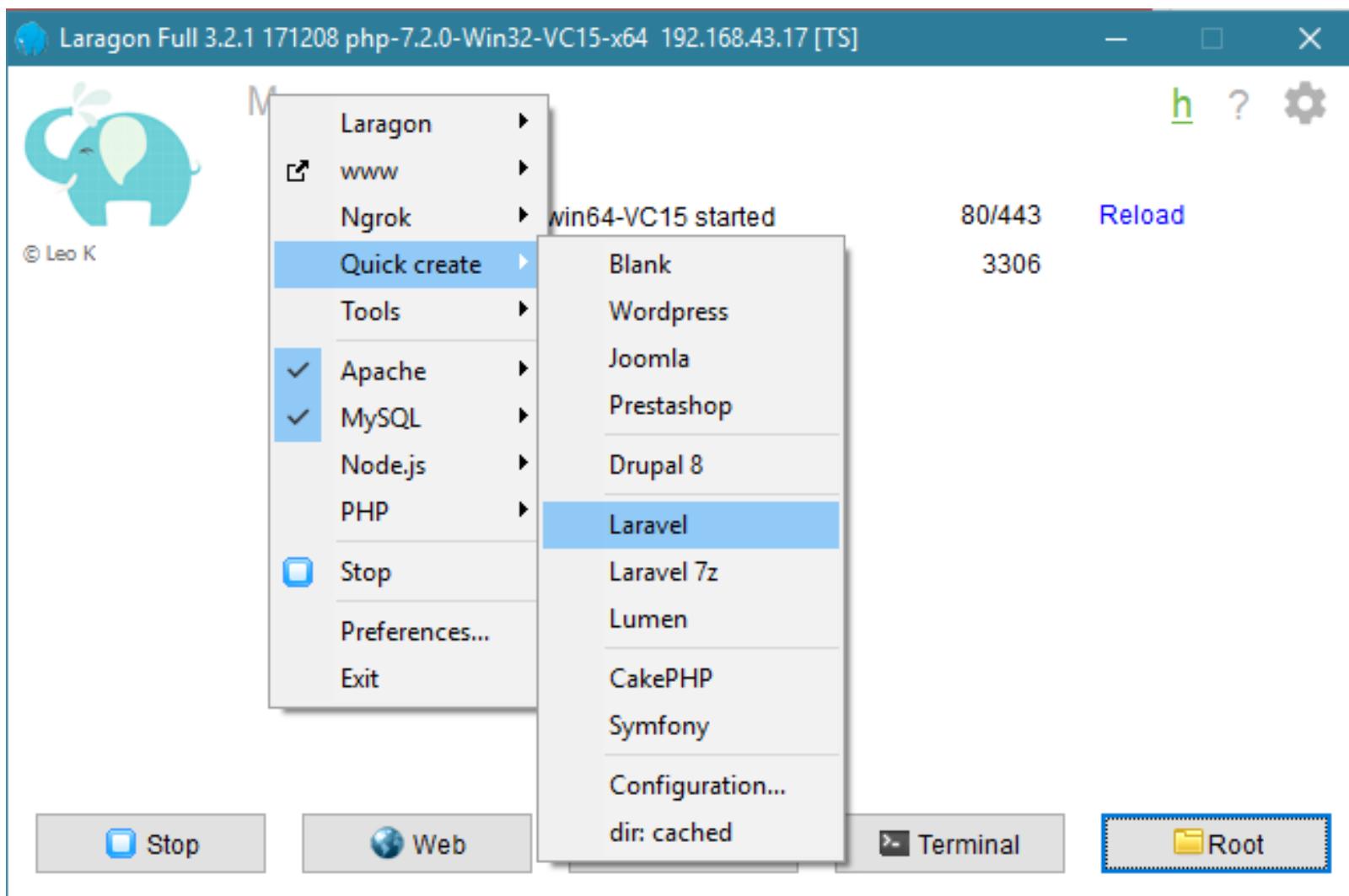


# Turn On Laragon



By default Laragon will serve port 80 for Apache and port 3306 for DB

# Create Project



# Laravel project with Pretty URL

The terminal window shows the command `@php artisan key:generate --ansi` being run, followed by a success message: `Application key set successfully.`. A note is displayed: `***** NOTE: Now, you can use pretty url for your awesome project :)`. The browser window shows a Laravel application with a header containing "T02 Laravel FSTM" and "Authentication". Red arrows point from the terminal output to the browser window, indicating the connection between the generated key and the pretty URL setup.

```
C:\WINDOWS\SYSTEM32\cmd.exe - cmd.exe
71
72
73 @php artisan key:generate --ansi
74 Application key set successfully.
75 *****
76 ***** NOTE: Now, you can use pretty url for your
77 ***** awesome project :)
78 *****
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
C:\laragon\www\lara7>
```

(Laragon) Project path: C:/laragon/www/lara7  
(Laragon) Pretty url: http://lara7.test

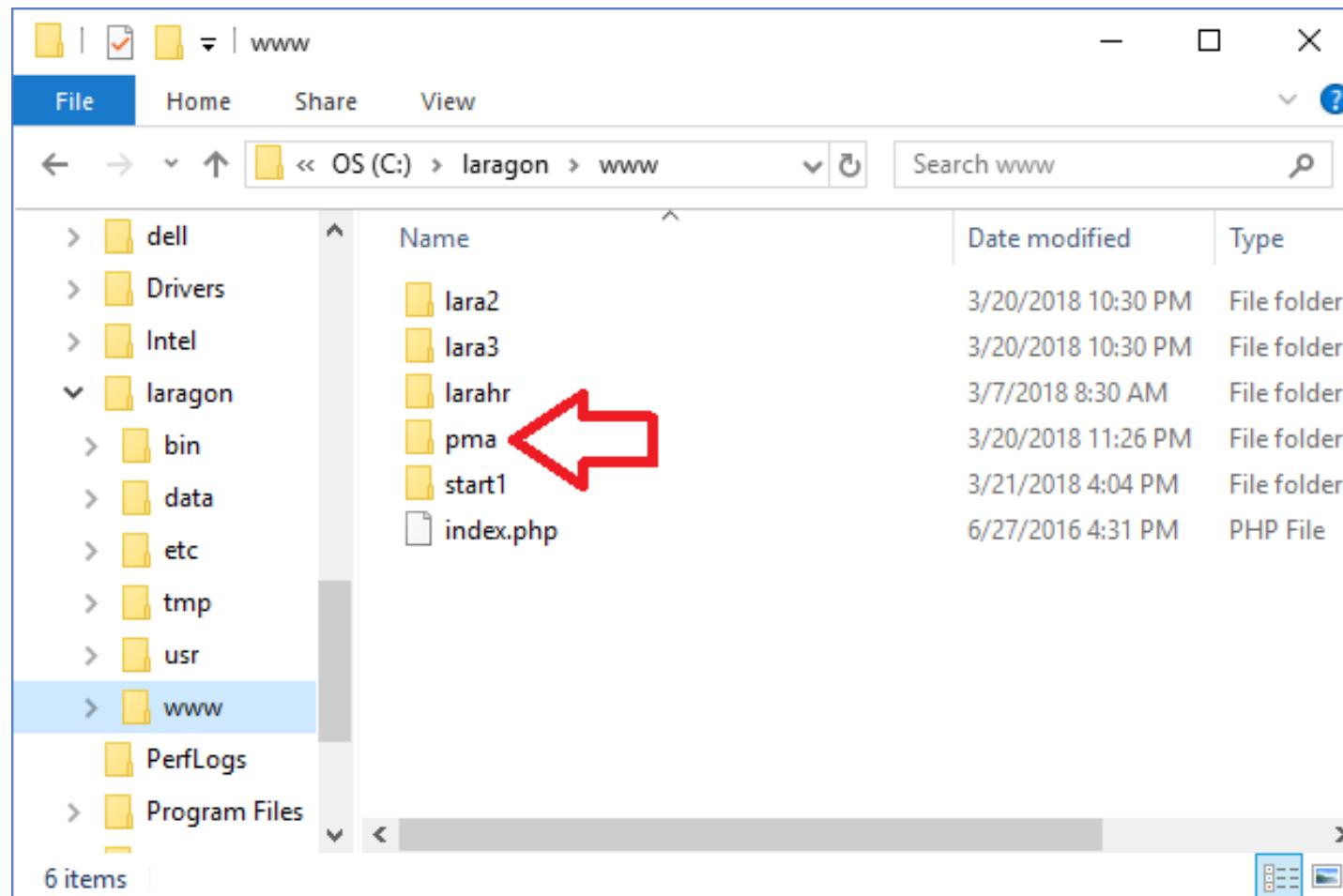
```
<a href="{{ url('/home') }}>Home</a>
@else
<a href="{{ route('auth.login') }}>Login</a>
@if(Route::has('register'))
<a href="{{ route('auth.register') }}>Register</a>
@endif
@endif
@endauth
@endauth
</div>
<div class="content">
<div class="title m-b-md">
T02 Laravel FSTM<br>
Authentication
</div>
```

Test the Pretty url in browser

# Update DB credentials in .env

```
8  
9  DB_CONNECTION=mysql  
10 DB_HOST=127.0.0.1  
11 DB_PORT=3306  
12 DB_DATABASE=start1  
13 DB_USERNAME=root  
14 DB_PASSWORD=abc123  
15
```

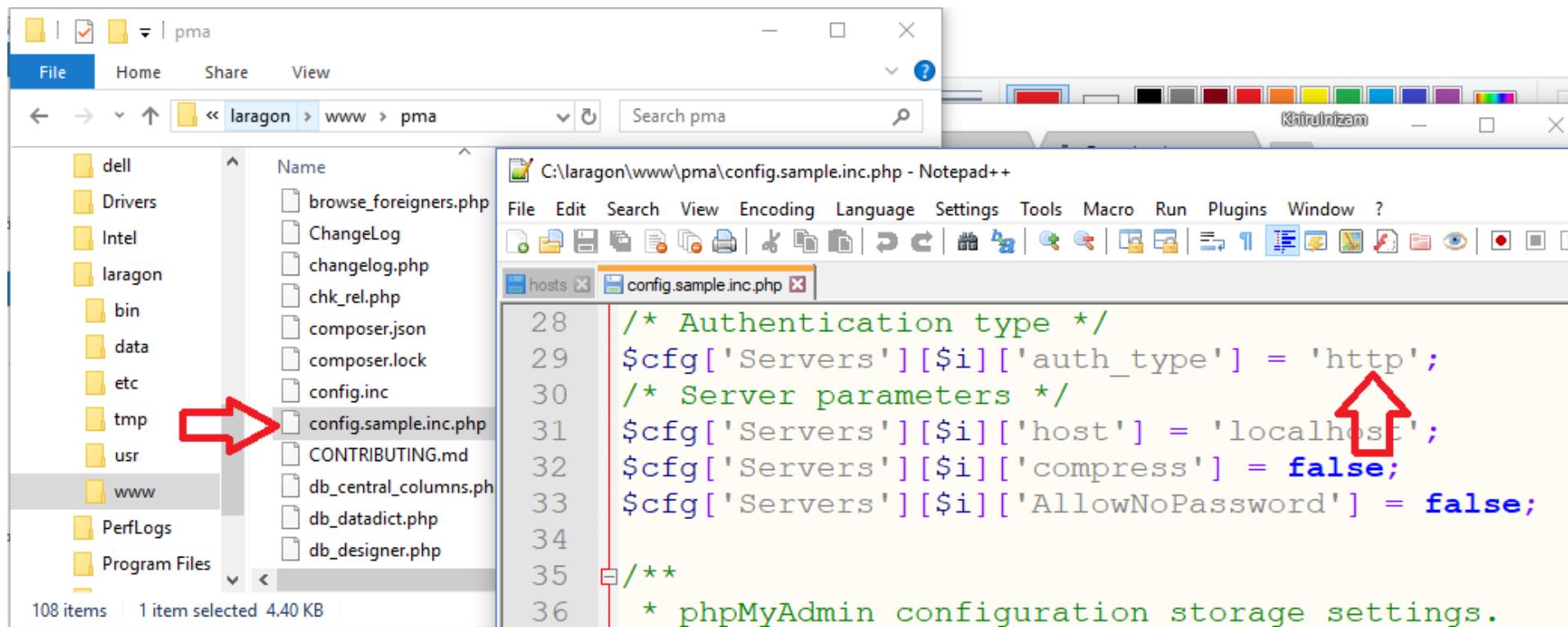
# Download and install phpmyadmin.Net



Try access

<http://localhost/pma>

# Phpmyadmin config.inc



# Authentication Module

Laravel makes implementing authentication very simple. In fact, almost everything is configured for you out of the box.

```
composer require laravel/ui --dev  
  
php artisan ui vue --auth  
  
npm install  
  
npm run dev
```

# Authentication Module

The screenshot shows a code editor with a file open and a terminal window below it.

**Code Editor Content:**

```
14
15
16
17
18
19
20
21
22
23
```

```
public function up()
{
    Schema::create('users', function (Blueprint $table)
    {
        $table->increments('id');
        $table->string('name');
        $table->string('email')->unique();
        $table->string('password');
        $table->rememberToken();
        $table->timestamps();
    });
}
```

**Terminal Window Content:**

```
C:\laragon\www\start1>php artisan migrate
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table
```

A red arrow points from the terminal window to the command `php artisan migrate`.

Login

E-Mail Address

Password

Remember Me

[Forgot Your Password?](#)

Reset Password

E-Mail Address

Register

Name

E-Mail Address

Password

Confirm Password



# T03 Pangkalan data – migration, seeder & factory

[bit.ly/laravelkerul](http://bit.ly/laravelkerul)





1. Submit User Request

2. Route to appropriate Laravel Controller

**Routing**

**routes/web.php**

**Controller**

**app/Http/Controllers**

3. Interact with Data Model

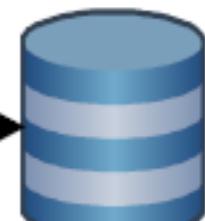
**View**

4. Controller invokes results View

**resources/views**

**Model**

**app/**



**Database**

5. Render view in users browser

Laragon Full 4.0.16 190914 php-7.2.19-Win32-VC15-x64 [TS] 192.168.1.43

Menu



© Leo K

Apache httpd-2.4.35-win64-VC15 started  
MySQL mysql-5.7.24-winx64 started

80  
3306  
Reload

h ?

Want to do

Send to appropriate Laravel Controller

**Routing**

**routes/web.php**

**app/Http/Controllers**

**Controller**

**3. Interact with Data Model**

Search

1. lara7

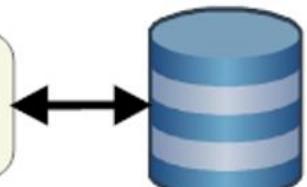
C:\laragon\www\lara7  
resources\views

**View**

**4. Controller invokes results View**

**Model**

**app/**



**Database**

**5. Render view in users browser**

# Working with Database

## Database Migrations

Migrations are like version control for your database.

## Make Migrations

```
php artisan make:migration create_trainings_table
```

```
public function up()
{
    Schema::create('trainings', function (Blueprint $table) {
        $table->bigIncrements('id');
        $table->string('title');
        $table->text('description');
        $table->string('trainer');
        $table->timestamps();

        $table->unsignedBigInteger('user_id');
        $table->foreign('user_id')->references('id')->on('users');
    });
}
```

# php artisan migrate

## Run Migrations

```
php artisan migrate
```

# Model

Models typically live in the app directory.

All Eloquent models  
extend Illuminate\Database\Eloquent\Model class.

## Model

```
php artisan make:model Training
```

```
<?php  
  
namespace App;  
  
use Illuminate\Database\Eloquent\Model;  
  
class Training extends Model  
{  
  
}
```

# Model

```
protected $fillable = [  
    'title', 'description', 'trainer', 'user_id', 'attachment'  
];
```

```
public function user()  
{  
    return $this->belongsTo('App\User');
```

```
protected $table= 'articles';
```

```
//define $article->submitted_date  
public function getSubmittedDateAttribute(){  
    return $this->created_at->format('d/m/Y');  
}
```

```
public function scopePublished($query){  
    return $query->where('published',1);  
}
```

Model used to define fillable attributes.

Model also can be used to define relationship

Model also can be used to specify custom table.

Model also can be used to define mutator

Model also can be used to define local scope

# Working with Database

## Database Seeds

Laravel includes a simple method of seeding your database with test data using seed classes.

On [database/seeds/TrainingsTableSeeder.php](#)

```
public function run()
{
    DB::table('trainings')->insert([
        'title' => 'Laravel Training 6 Days(Advanced)',
        'description' => 'The 6 Days training focus on enhancement API to Restful Architecture',
        'trainer' => 'Khirulnizam',
        'user_id' => factory('App\User')->create()->id,
    ]);
}
```

## Make Seeder

```
php artisan make:seeder TrainingsTableSeeder
```

On [database/seeds/DatabaseSeeder.php](#)

```
public function run()
{
    $this->call(TrainingsTableSeeder::class);
}
```

# Working with Database : run seeder

## Run Seeder

```
php artisan db:seed
```

# Working with Database

## Database Factories

Laravel has a feature called model factories that allows you to build fake data for your models.

### Make Factory

```
php artisan make:factory TrainingFactory
```

```
use App\Training;
use Faker\Generator as Faker;

$factory->define(Training::class, function (Faker $faker) {
    return [
        'title' => $faker->sentence(5),
        'description' => $faker->text(),
        'trainer' => $faker->name,
        'user_id' => factory('App\User')->create()->id,
    ];
});
```

### Run Factory

```
php artisan tinker
factory('App\Training',10)->create();
```

## How to create and run migrations, seeder and factory?

Here are the example.

### Make Migrations

```
php artisan make:migration create_trainings_table  
php artisan make:migration alter_articles_table_add_attachment
```

### Run Migrations

```
php artisan migrate
```

### Make Seeder

```
php artisan make:seeder TrainingsTableSeeder
```

### Run Seeder

```
php artisan db:seed
```

### Make Factory

```
php artisan make:factory TrainingFactory
```

### Run Factory

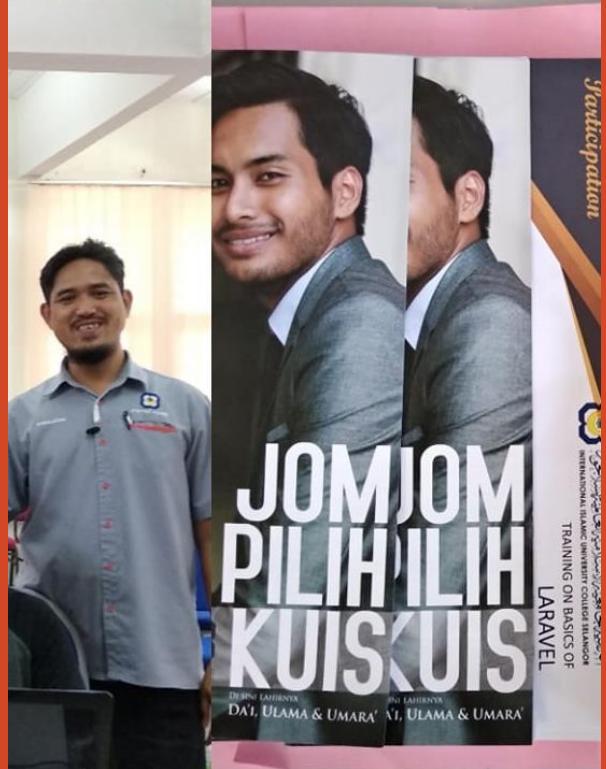
```
php artisan tinker  
factory('App\Training',10)->create();
```

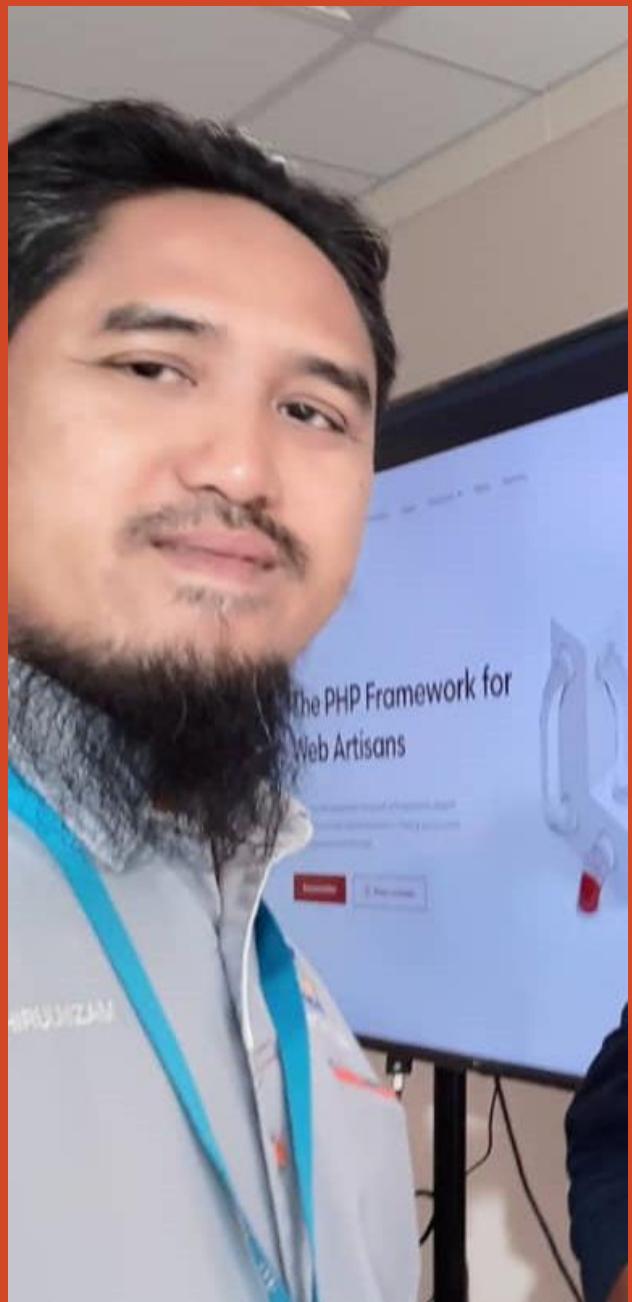
# Laravel t04

# Views

## (create form)

[fstm.kuis.edu.my/blog/laravel](http://fstm.kuis.edu.my/blog/laravel)





# Laravel t05

## Views

### (save new record)

[fstm.kuis.edu.my/blog/laravel](http://fstm.kuis.edu.my/blog/laravel)





# Laravel t06

## Views

### (list records in index)

[fstm.kuis.edu.my/blog/laravel](http://fstm.kuis.edu.my/blog/laravel)



1. Submit User Request



2. Route to appropriate Laravel Controller

**Routing**

`routes/web.php`

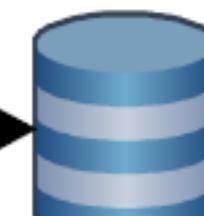
**Controller**

`app/Http/Controllers`

3. Interact with Data Model

`app/`

**Model**



Database

`resources/views`

**View**

4. Controller invokes results View

5. Render view in users browser

# Create new record interface (create.blade.php)

## Create Training

Create Training

Title

Description

Trainer

[Create New Training](#) [Cancel](#)

# Training index.blade.php

## Training Index

Training Index

Training Index

ID	Title	Actions		
1	Laravel Beginner to Intermediate	Show	Edit	Delete
2	PHP & Mysql Web System	Show	Edit	Delete

# Views

Views contain the HTML served by your application and separate your controller / application logic from your presentation logic.

Views are stored in the resources/views directory

1. Go to resources/views
2. Create training folder
3. Inside training folder, create:
  1. create.blade.php
  2. index.blade.php

```
✓ resources
  > js
  > lang
  > sass
  ✓ views
    > admin
    > auth
    > layouts
    ✓ trainings
      └─ create.blade.php
      └─ index.blade.php
```

# Views – Create View

Views are stored in the resources/views directory

1. Go to

`resources/views/trainings/create.blade.php`

```
@extends('layouts.app')

@section('content')
<div class="container">
    <div class="row justify-content-center">
        <div class="col-md-8">
            <div class="card">
                <div class="card-header">Create Training</div>

                <div class="card-body">
                    <form action="" method="POST">
                        @csrf
                        <div class="form-group">
                            <label for="title">Title</label>
                            <input type="text" class="form-control" name="title">
                        </div>
                        <div class="form-group">
                            <label for="description">Description</label>
                            <textarea name="description" cols="20" rows="10" class="form-control"></textarea>
                        </div>
                        <div class="form-group">
                            <label for="trainer">Trainer</label>
                            <input type="text" class="form-control" name="trainer">
                        </div>
                        <div class="form-group">
                            <button type="submit" class="btn btn-primary">Create New Training</button>
                            <a href="" class="btn btn-link">Cancel</a>
                        </div>
                    </form>
                </div>

            </div>
        </div>
    </div>
</div>
@endsection
```

# Views – Index View

Views are stored in the resources/views directory

1. Go to resources/views/trainings/index.blade.php

```
@extends('layouts.app')

@section('content')
<div class="container">
    <div class="row justify-content-center">
        <div class="col-md-8">
            <div class="card">
                <div class="card-header">Training Index</div>

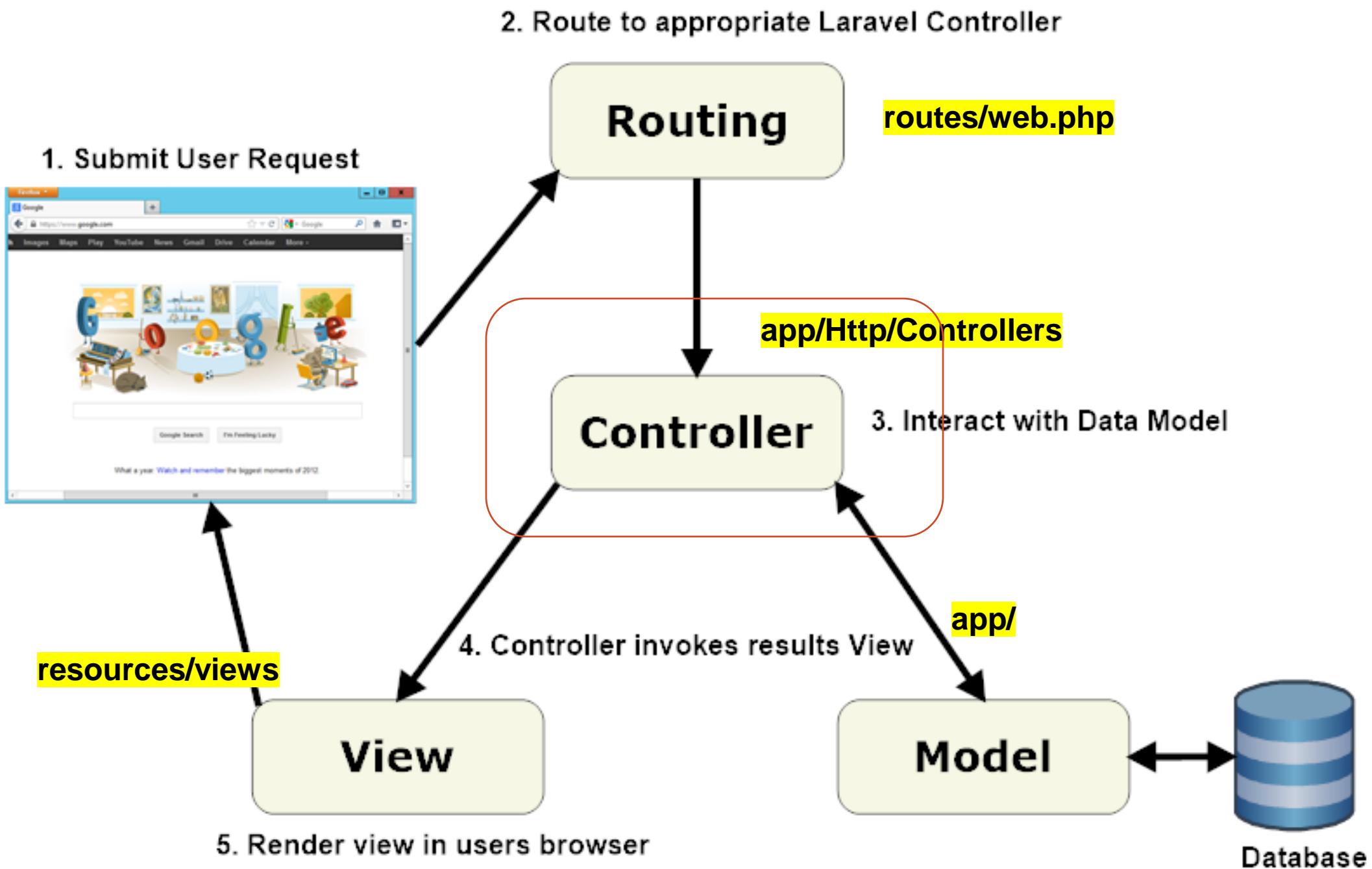
                <div class="card-body">
                    <table class="table">
                        <thead>
                            <tr>
                                <th>ID</th>
                                <th>Title</th>
                                <th>Creator</th>
                                <th>Actions</th>
                            </tr>
                        </thead>
                        <tbody>
                            @foreach ($trainings as $training)
                                <tr>
                                    <td>{{ $training->id}}</td>
                                    <td>{{ $training->title}}</td>
                                    <td>{{ $training->user_id}}</td>
                                    <td>
                                        <a href="" class="btn btn-primary">Show</a>
                                        <a href="" class="btn btn-success">Edit</a>
                                        <a href="" class="btn btn-danger"
                                            onclick="return confirm('Are you sure?')">Delete</a>
                                    </td>
                                </tr>
                            @endforeach
                        </tbody>
                    </table>
                </div>
            </div>
        </div>
    </div>
</div>
@endsection
```

# Laravel t07

# Controller

<http://fstm.kuis.edu.my/blog/laravel>





# Controller

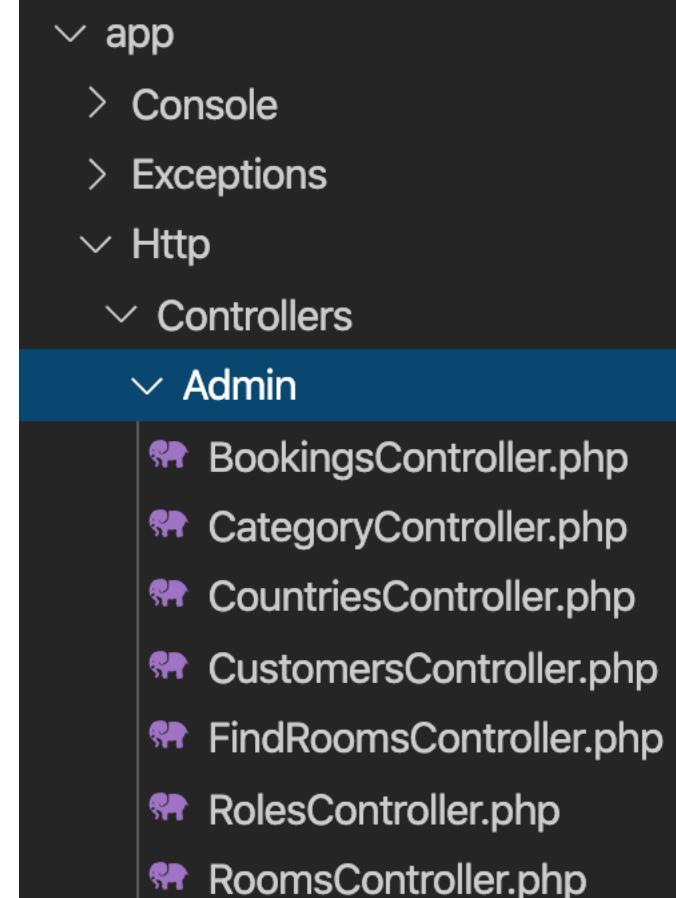
Instead of defining all of your request handling logic as Closures in route files, you may wish to organize this behavior using Controller classes.

Controllers can group related request handling logic into a single class.

Controllers are stored in the `.app/Http/Controllers` directory.

## Make Controller

```
php artisan make:controller TrainingController
```



# Controller

7 method in controllers

## Make Controller

```
C:\laragon\www\lara7
λ php artisan make:controller TrainingController --resource
```

### Action

index

create

store

show

edit

update

destroy

# Controller - Create

Description	URL	Controller Function	View File
Show the form to create a new training.	GET example.com/trainings/create	create()	app/views/trainings/create.blade.php

```
public function create()
{
    return view('trainings.create');
}
```

# Controller - Index

Description	URL	Controller Function	View File
Default page for showing all the training.	GET example.com/trainings	index()	app/views/trainings/index.blade.php

```
use App\Training;

class TrainingController extends Controller
{
    //
    public function index()
    {
        $trainings = Training::all();
        return view('trainings.index')->with(compact('trainings'));
    }
}
```

# Controller - Store

Description	URL	Controller Function
Process the create form submit and save the training to the database.	POST example.com/trainings/create	store()

```
//Method 2
$user = auth()->user();
$article = $user->trainings()->create($request->only('title','description','trainer'));
```

```
use Illuminate\Support\Facades\Auth;

class TrainingController extends Controller
{
    /**
     * @ -17,4 +18,17 @@ public function create()
     {
        return view('trainings.create');
    }

    public function store(Request $request)
    {
        //Method 1
        $training = new Training();
        $training->title = $request->get('title');
        $training->description = $request->get('description');
        $training->trainer = $request->get('trainer');
        $training->user_id = Auth::id();
        $training->save();

        return redirect('/trainings');
    }
}
```

# Laravel t09 Routing

Youtube : [bit.ly/laravelkerul](https://bit.ly/laravelkerul)



1. Submit User Request



2. Route to appropriate Laravel Controller

**Routing**

**routes/web.php**

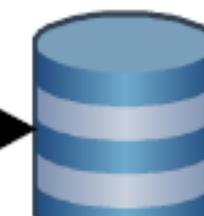
**app/Http/Controllers**

**Controller**

3. Interact with Data Model

**app/**

**Model**



**Database**

**resources/views**

**View**

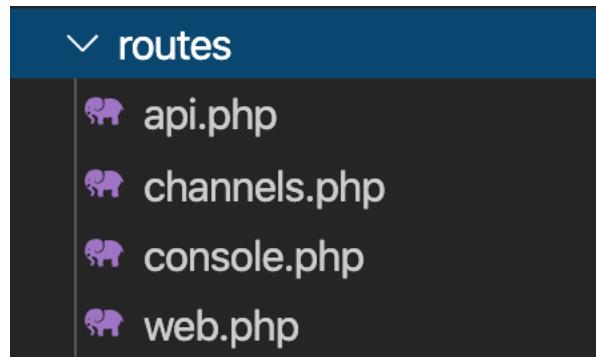
4. Controller invokes results View

5. Render view in users browser

# Routing

All Laravel routes are defined in your route files, which are located in the routes directory.

<https://laravel.com/docs/6.x/routing>.

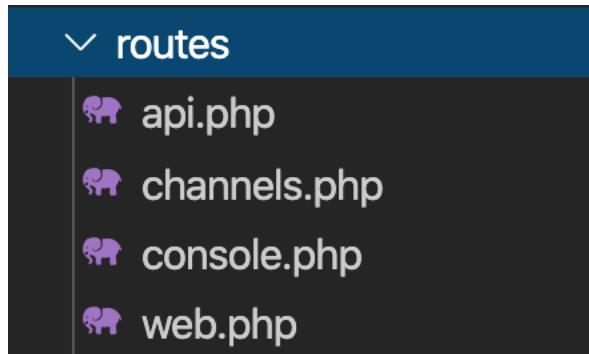


```
Route::get($uri, $callback);  
Route::post($uri, $callback);  
Route::put($uri, $callback);  
Route::patch($uri, $callback);  
Route::delete($uri, $callback);  
Route::options($uri, $callback);
```

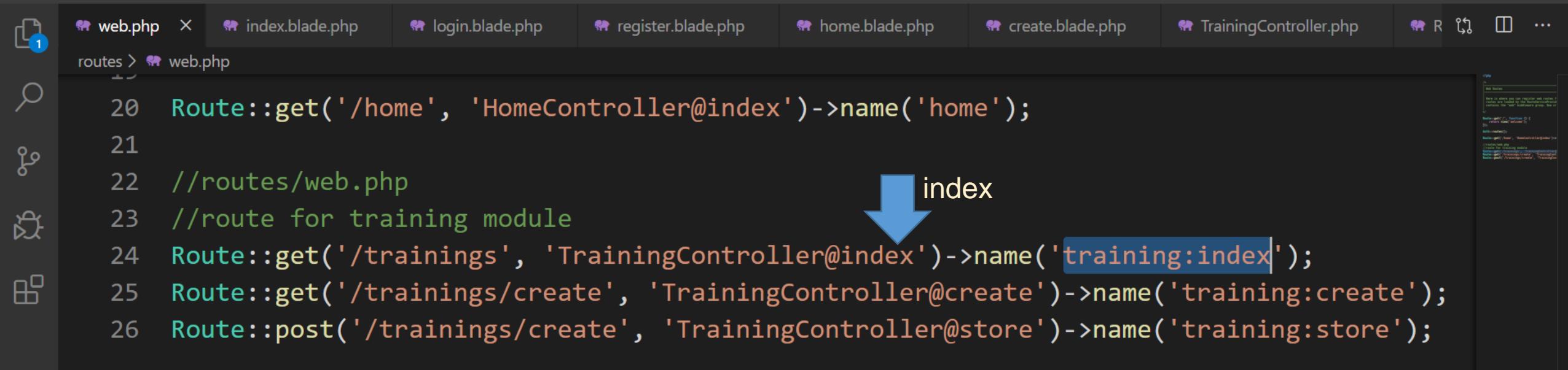
# Simple Routing

All Laravel routes are defined in your route files, which are located in the routes directory.

<https://laravel.com/docs/6.x/routing>.



# Simple routing ( routes/web.php )



```
routes > web.php
1  web.php X  index.blade.php  login.blade.php  register.blade.php  home.blade.php  create.blade.php  TrainingController.php  R  ⌂  ...
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20 Route::get('/home', 'HomeController@index')->name('home');
21
22 //routes/web.php
23 //route for training module
24 Route::get('/trainings', 'TrainingController@index')->name('training:index');
25 Route::get('/trainings/create', 'TrainingController@create')->name('training:create');
26 Route::post('/trainings/create', 'TrainingController@store')->name('training:store');
```

A blue arrow points from the word "index" in the route definition at line 20 to the word "index" in the name() method call at line 24.

# Simple Routing

```
php artisan route:list
```

GET HEAD   trainings	training:index	App\Http\Controllers\TrainingController@index	web,auth
GET HEAD   trainings/create	training:create	App\Http\Controllers\TrainingController@create	web,auth
POST   trainings/create	training:store	App\Http\Controllers\TrainingController@store	web,auth

# php artisan route:list

Domain	Method	URI	Name	Action	Middleware
	GET HEAD	/		Closure	web
	GET HEAD	api/user		Closure	api,auth:api
	GET HEAD	home	home	App\Http\Controllers\HomeController@index	web,auth
	GET HEAD	login	login	App\Http\Controllers\Auth\LoginController@showLoginForm	web,guest
	POST	login		App\Http\Controllers\Auth\LoginController@login	web,guest
	POST	logout	logout	App\Http\Controllers\Auth\LoginController@logout	web
	POST	password/confirm		App\Http\Controllers\Auth\ConfirmPasswordController@confirm	web,auth
	GET HEAD	password/confirm	password.confirm	App\Http\Controllers\Auth\ConfirmPasswordController@showConfirmForm	web,auth
	POST	password/email	password.email	App\Http\Controllers\Auth\ForgotPasswordController@sendResetLinkEmail	web
	POST	password/reset	password.update	App\Http\Controllers\Auth\ResetPasswordController@reset	web
	GET HEAD	password/reset	password.request	App\Http\Controllers\Auth\ForgotPasswordController@showLinkRequestForm	web
	GET HEAD	password/reset/{token}	password.reset	App\Http\Controllers\Auth\ResetPasswordController@showResetForm	web
	POST	register		App\Http\Controllers\Auth\RegisterController@register	web,guest
	GET HEAD	register	register	App\Http\Controllers\Auth\RegisterController@showRegistrationForm	web,guest
	GET HEAD	trainings	training:index	App\Http\Controllers\TrainingController@index	web
	GET HEAD	trainings/create	training:create	App\Http\Controllers\TrainingController@create	web
	POST	trainings/create	training:store	App\Http\Controllers\TrainingController@store	web

## Action

---

index

---

create

---

store

---

show

---

edit

---

update

---

destroy

# Advanced ROUTING: group & named

```
routes > 🐾 web.php

24 //Route::get('/trainings', 'TrainingController@index')->name('training:index');
25 //Route::get('/trainings/create', 'TrainingController@create')->name('training:create');
26 //Route::post('/trainings/create', 'TrainingController@store')->name('training:store');
27
28 //group routing
29 //
30 Route::group([
31     'middleware'=>'auth',//lock this route group to only authenticated users
32     'prefix'=>'trainings',
33     'as'=>'training:',
34 ],function(){
35     Route::get('/', 'TrainingController@index')->name('index');
36     Route::get('/create', 'TrainingController@create')->name('create');
37     Route::post('/create', 'TrainingController@store')->name('store');
38     Route::get('/{training}', 'TrainingController@show')->name('show');
39     Route::get('/{training}/edit', 'TrainingController@edit')->name('edit');
40     Route::post('/{training}', 'TrainingController@update')->name('update');
41     Route::post('/{training}/delete', 'TrainingController@delete')->name('delete');
42 });

});
```

# php artisan route:list

```
C:\laragon\www\lara7
λ php artisan route:list
```

Domain	Method	URI	Name	Action	Middleware
	GET HEAD	/		Closure	web
	GET HEAD	api/user		Closure	api,auth:api
	GET HEAD	home	home	App\Http\Controllers\HomeController@index	web,auth
	GET HEAD	login	login	App\Http\Controllers\Auth\LoginController@showLoginForm	web,guest
	POST	login		App\Http\Controllers\Auth\LoginController@login	web,guest
	POST	logout	logout	App\Http\Controllers\Auth\LoginController@logout	web
	POST	password/confirm		App\Http\Controllers\Auth\ConfirmPasswordController@confirm	web,auth
	GET HEAD	password/confirm	password.confirm	App\Http\Controllers\Auth\ConfirmPasswordController@showConfirmForm	web,auth
	POST	password/email	password.email	App\Http\Controllers\Auth\ForgotPasswordController@sendResetLinkEmail	web
	POST	password/reset	password.update	App\Http\Controllers\Auth\ResetPasswordController@reset	web
	GET HEAD	password/reset	password.request	App\Http\Controllers\Auth\ForgotPasswordController@showLinkRequestForm	web
	GET HEAD	password/reset/{token}	password.reset	App\Http\Controllers\Auth\ResetPasswordController@showResetForm	web
	POST	register		App\Http\Controllers\Auth\RegisterController@register	web,guest
	GET HEAD	register	register	App\Http\Controllers\Auth\RegisterController@showRegistrationForm	web,guest
	GET HEAD	trainings	training:index	App\Http\Controllers\TrainingController@index	web,auth
	GET HEAD	trainings/create	training:create	App\Http\Controllers\TrainingController@create	web,auth
	POST	trainings/create	training:store	App\Http\Controllers\TrainingController@store	web,auth
	GET HEAD	trainings/{training}	training:show	App\Http\Controllers\TrainingController@store('create');	web,auth
	POST	trainings/{training}	training:update	App\Http\Controllers\TrainingController@store('store');	web,auth
	POST	trainings/{training}/delete	training:delete	App\Http\Controllers\TrainingController@store('show');	web,auth
	GET HEAD	trainings/{training}/edit	training:edit	App\Http\Controllers\TrainingController@store('edit');	web,auth



# Laravel t07

# Search

Tarmizi Sanusi  
Khirulnizam Abd Rahman  
[fstm.kuis.edu.my/blog/laravel](http://fstm.kuis.edu.my/blog/laravel)



# Search

1. Create form to input keyword
2. Query the keyword that match with title

Training Index			
ID	Title	Creator	Actions
13	Laravel Training 6 Days(Advanced)	Prof. Francesco Haag IV	<button>Show</button> <button>Edit</button> <button>Delete</button>
18	Laravel Medkad	Vincent Harding	<button>Show</button> <button>Edit</button> <button>Delete</button>

① [127.0.0.1:8000/trainings?keyword=laravel](http://127.0.0.1:8000/trainings?keyword=laravel)

# Search - Form

```
<div class="card-header">Training Index
<div class="float-right">
    <form
        class="d-none d-md-inline-block form-inline ml-auto mr-0 mr-md-3 my-2 my-md-0"
        method="GET"
        action="{{ route('training:index') }}"
    >
        <div class="input-group">
            <input
                class="form-control"
                type="text" placeholder="Search for..."
                aria-label="Search" aria-describedby="basic-addon2"
                name="keyword"
                value="{{ request()->get('keyword') }}"
            />
            <div class="input-group-append">
                <button class="btn btn-primary" type="submit"><i class="fas fa-search"></i></button>
            </div>
        </div>
    </form>
    <a href="{{ route('training:create') }}" class="btn btn-primary">+ Create New Training </a>
</div>
</div>
```

Using GET method to  
named route **training:index**

# Simple routing ( routes/web.php )

```
18 Auth::routes();  
19  
20 Route::get('/home', 'HomeController@index')->name('home');  
21  
22 //routes/web.php  
23 //route for training module  
24 Route::get('/trainings', 'TrainingController@index')->name('training');  
25 Route::get('/trainings/create', 'TrainingController@create')->name('training:create');  
26 Route::post('/trainings/create', 'TrainingController@store')->name('training:store');
```



The diagram illustrates the mapping between a route name and a controller method. A blue downward-pointing arrow originates from the word "index" in the line "Route::get('/trainings', 'TrainingController@index')->name('training');". It points to the second "index" in the controller name "TrainingController@index". This visual cue helps in understanding how the route name "trainings" is mapped to the controller's index action.

# Simple Routing

```
php artisan route:list
```

GET HEAD   trainings	training:index	App\Http\Controllers\TrainingController@index	web,auth
GET HEAD   trainings/create	training:create	App\Http\Controllers\TrainingController@create	web,auth
POST   trainings/create	training:store	App\Http\Controllers\TrainingController@store	web,auth

# Search – Index Method

Check if there is keyword, if there is keyword on query string, run query to search column where like title.

If no keyword, the method will return collection rows from trainings table.

```
public function index(Request $request)
{
    if($request->get('keyword') != null)
    {
        $keyword = $request->get('keyword');
        $trainings = Training::where('title', 'LIKE', '%' . $keyword . '%')->get();
    }
    else
    {
        $trainings = Training::all();
    }
    return view('trainings.index')->with(compact('trainings'));
}
```



# Laravel t08 Pagination

Tarmizi Sanusi  
Khirulnizam Abd Rahman

[fstm.kuis.edu.my/blog/laravel](http://fstm.kuis.edu.my/blog/laravel)

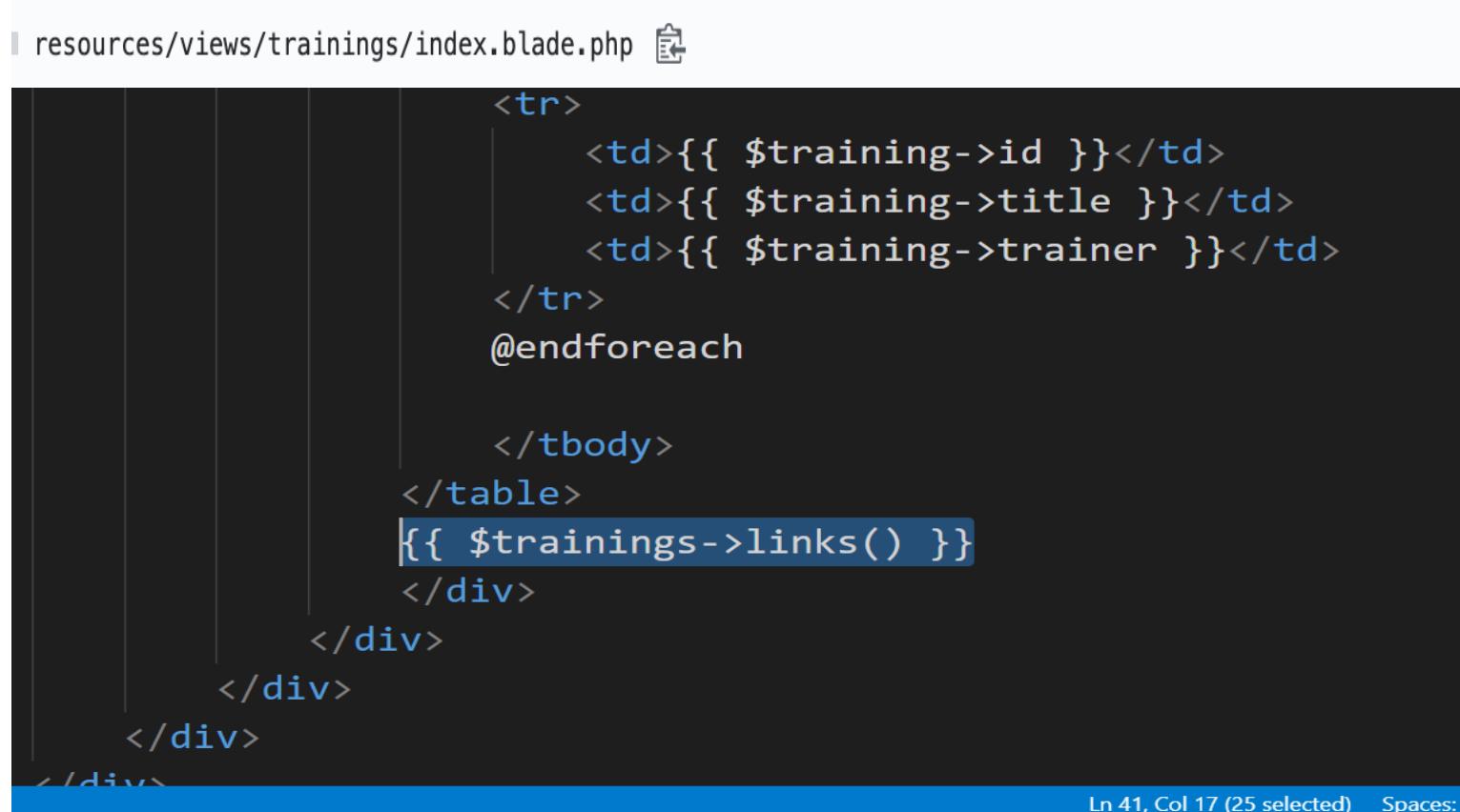


Training Index		
<input type="text"/>		<button>Search</button>
<b>id</b>	<b>title</b>	<b>trainer</b>
1	Laravel Basics	Khirulnizam Abd Rahman
2	Laravel Intermediate	Tarmizi Sanusi
3	Laravel Advanced with Unit Test	Nasrul Hazim
4	Ipsa molestiae enim atque voluptatem.	Dr. Imani Feil
5	Tempore dicta eos ut sint.	Frida White

[«](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [»](#)

# Search - Pagination

The paginate method automatically takes care of setting the proper limit and offset based on the current page being viewed by the user and append with keywords.



A screenshot of a code editor displaying a portion of a Blade template file named `index.blade.php`. The code is written in PHP and HTML. It includes a `@foreach` loop to iterate over `$trainings`, rendering each item in a table row (`<tr>`). Inside the loop, three `<td>` tags are used to display the `id`, `title`, and `trainer` of each training. After the loop, there is a closing `</tbody>`, `</table>`, and `</div>`. The URL `https://github.com/samtarmizi/laravel-spr/commit/6b0d8b3c64bd42c58af97fc49b2c48d983273f1d` is visible at the bottom of the editor window.

```
resources/views/trainings/index.blade.php   
  
<tr>  
    <td>{{ $training->id }}</td>  
    <td>{{ $training->title }}</td>  
    <td>{{ $training->trainer }}</td>  
</tr>  
@endforeach  
  
</tbody>  
</table>  
{{ $trainings->links() }}  
</div>  
</div>  
</div>
```

Ln 41, Col 17 (25 selected) Spaces: 4

<https://github.com/samtarmizi/laravel-spr/commit/6b0d8b3c64bd42c58af97fc49b2c48d983273f1d>  
http://fstm.kuis.edu.my/blog/laravel/

# TrainingController@index

```
if($request->get('keyword') != null)
{
    $keyword = $request->get('keyword');
    $trainings = Training::where('title','LIKE','%'.$keyword.'%')->get();
    $trainings = Training::where('title','LIKE','%'.$keyword.'%')->paginate(5);
}
else
{
    $trainings = Training::all();
    $trainings = Training::paginate(5);
}
```

# Hacks

Creating Index - <https://github.com/samtarmizi/laravel-spr/commit/1c925853f61da081f0e3ce298e58a4a423b66040>

Creating Create - <https://github.com/samtarmizi/laravel-spr/commit/7a5b5f3b643bcb8a265af3bb9f4e94e3aef95aa1>

Creating Store - <https://github.com/samtarmizi/laravel-spr/commit/2513b55c65bc4d27afa5004144ee2c7d41b3cbd7>



# Laravel t10

# npm not found

## [fstm.kuis.edu.my/blog/laravel](http://fstm.kuis.edu.my/blog/laravel)



# npm not found



```
Command Prompt

D:\>npm -v
'npm' is not recognized as an internal or external command,
operable program or batch file.

D:\>npm install
'npm' is not recognized as an internal or external command,
operable program or batch file.

D:\>
```

# Authentication Module

Laravel makes implementing authentication very simple. In fact, almost everything is configured for you out of the box.

```
composer require laravel/ui --dev
```

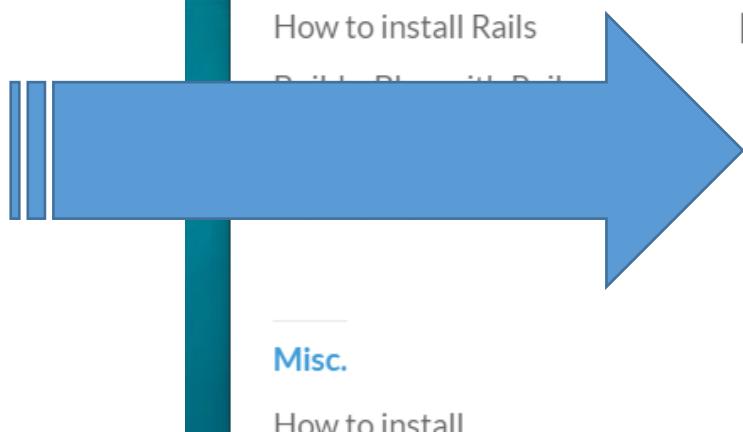
```
php artisan ui vue --auth
```

```
npm install
```

```
npm run dev
```

# Laragon

# Includes NODEJS



The screenshot shows a Microsoft Edge browser window with the URL [laragon.org/download/index.html](http://laragon.org/download/index.html). The page title is "Download". On the left sidebar, there are links for "Other versions", "Migrate to Laragon" (with sub-links for WAMP, XAMPP, and others), "Tutorials" (with sub-links for Rails, Python, Node.js, and Java), and "Misc." (with sub-links for Rails, Python, Node.js, and Java). A large blue arrow points from the "Tutorials" section towards the "Edition" section. The main content area starts with a paragraph about Laragon's features, followed by a section titled "Benefits of Laragon" which lists various tools like git, phpmyadmin, Node.js, MongoDB, Python, Django, Flask, Postgres, Ruby, Java, and Go. It also notes that users can download from GitHub. Below this is a section titled "Edition" with two bullet points: one for "Laragon Full" (Apache 2.4, Nginx, MySQL 5.7, PHP 7.2, Redis, Memcached, Node.js 11, npm, yarn, git) and another for "Laragon Lite" (which excludes Node.js, npm, yarn, git but includes "Tools > Quick add"). Download links for both editions are provided at the bottom of each section.

Download | Laragon - portable, is X

← → C [laragon.org/download/index.html](http://laragon.org/download/index.html) ☆ S F I P A :

**Other versions**

Version 4.0 (Win 32-bit)

---

**Migrate to Laragon**

Migrate from WAMP  
Migrate from XAMPP  
Migrate from others

---

**Tutorials**

How to install Rails  
Python  
Node.js  
Java

---

**Misc.**

How to install Rails  
Python  
Node.js  
Java

**Download** Edit

---

Laragon is a universal development environment. It has many features to make you more productive:

**Benefits of Laragon**

After downloading, You can add **git, phpmyadmin, Node.js/MongoDB, Python/Django/Flask/Postgres, Ruby, Java, Go** using “Tools > Quick add”

Note: [You can also download from GitHub](#)

---

**Edition**

- **Laragon Full:** Apache 2.4, Nginx, MySQL 5.7, PHP 7.2, Redis, Memcached, Node.js 11, npm, yarn, git,...
- **Laragon Lite:** Don't include Node.js 11, npm, yarn, git but you can add them easily using “Tools > Quick add”

[Download Laragon - Full \(130 MB\)](#)

[Download Laragon - Lite \(85 MB\)](#)

---

**Contents**

[Edition](#)  
[Why Laragon?](#)  
[Tutorials](#)  
[Back to Top](#)

# Terminal in Laragon

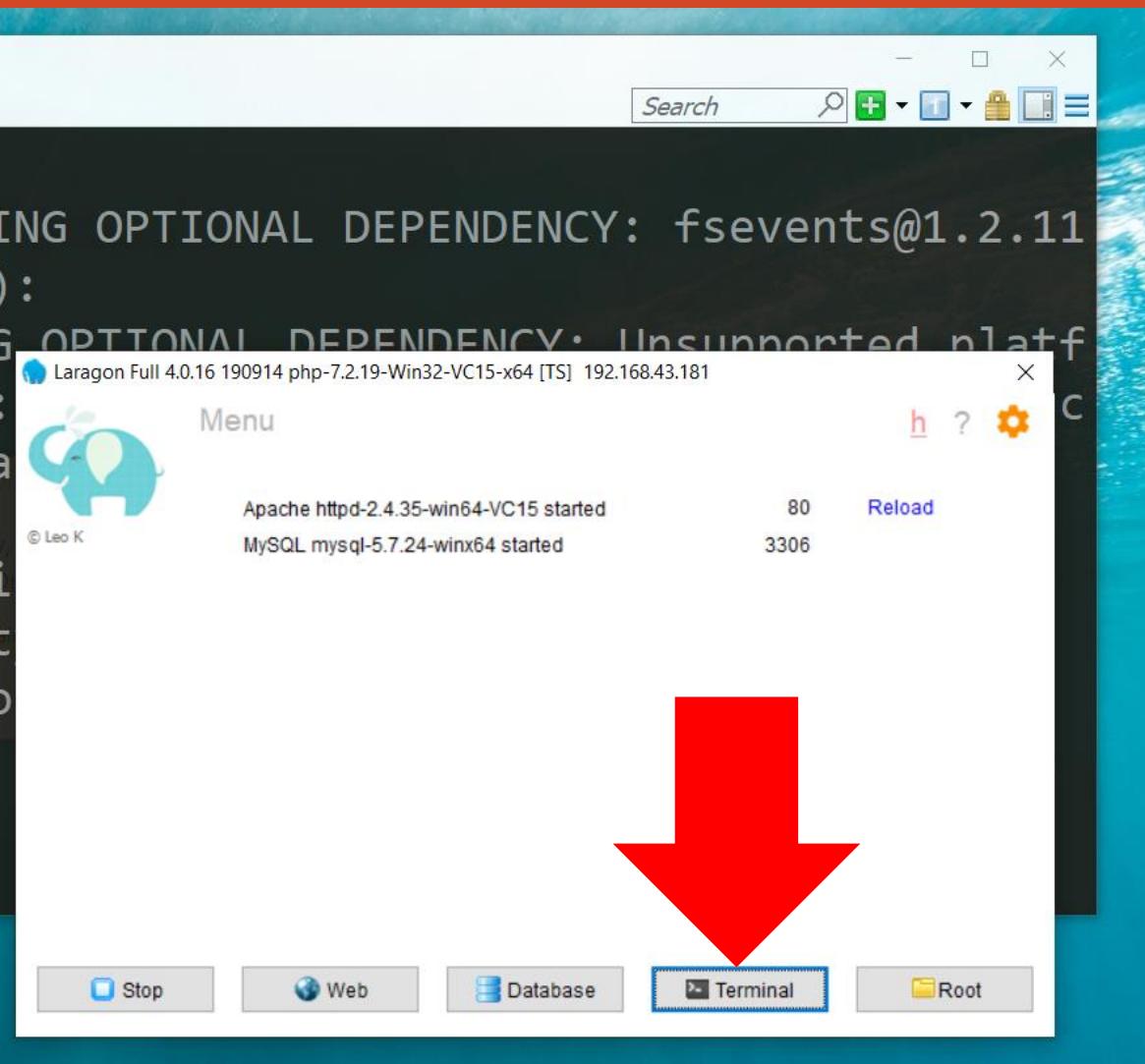
Laravel makes

lf.

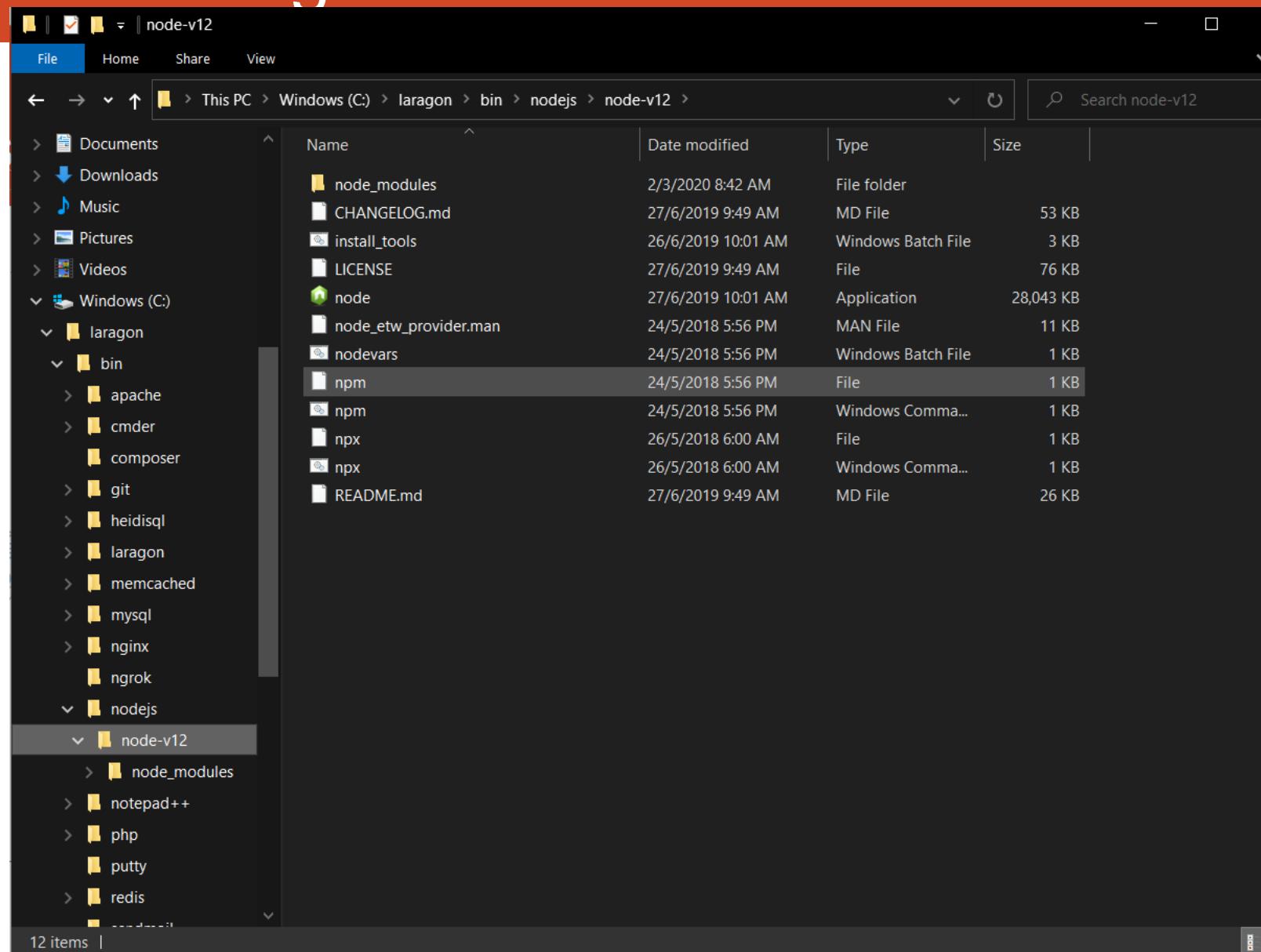
```
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.11  
  (node_modules\fsevents):  
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform  
  for fsevents@1.2.11:  
  current: {"os": "win32", "a
```

```
audited 17591 packages in  
found 1 moderate severity  
  run `npm audit fix` to
```

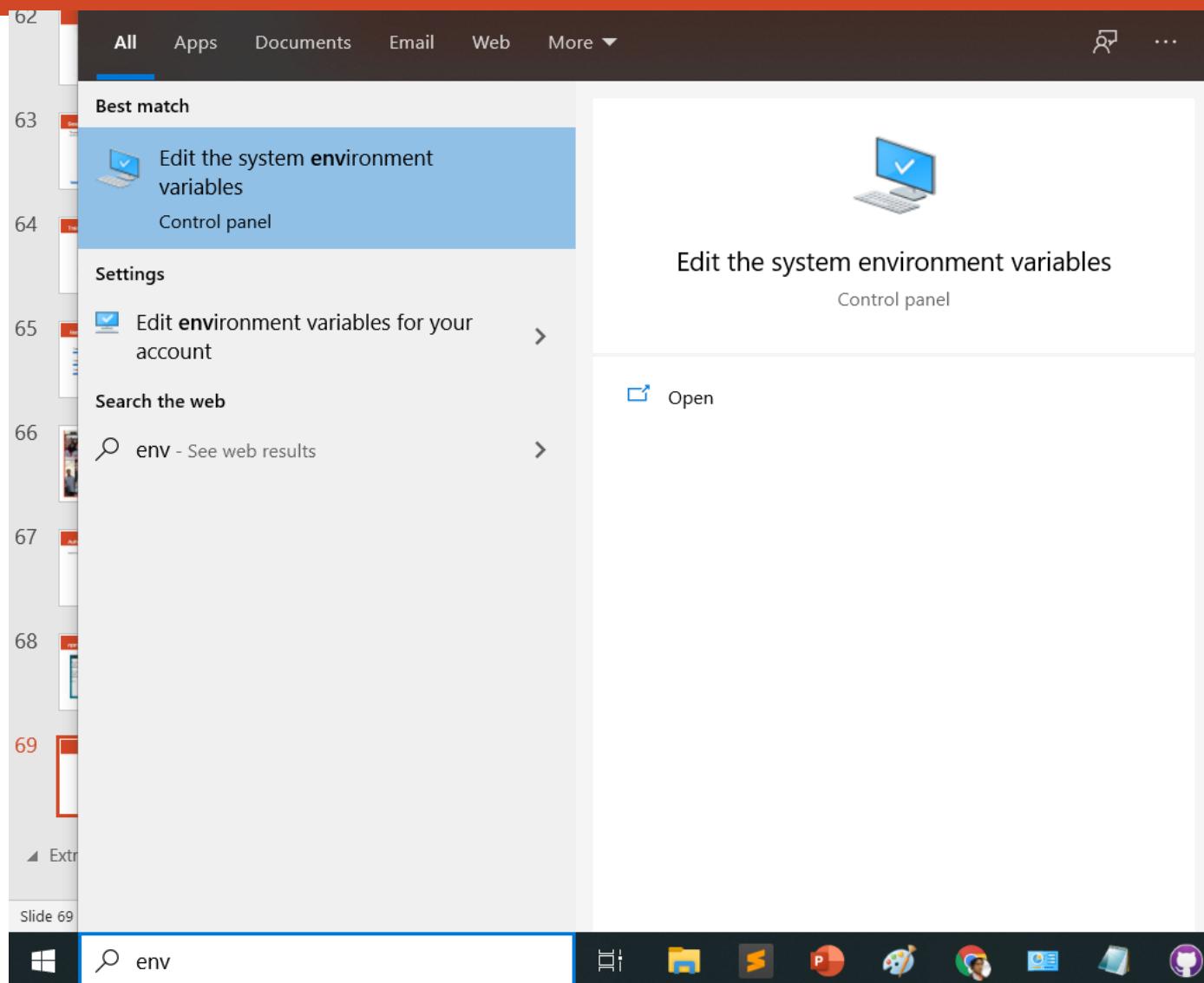
```
C:\laragon\www\lara7  
λ npm install
```



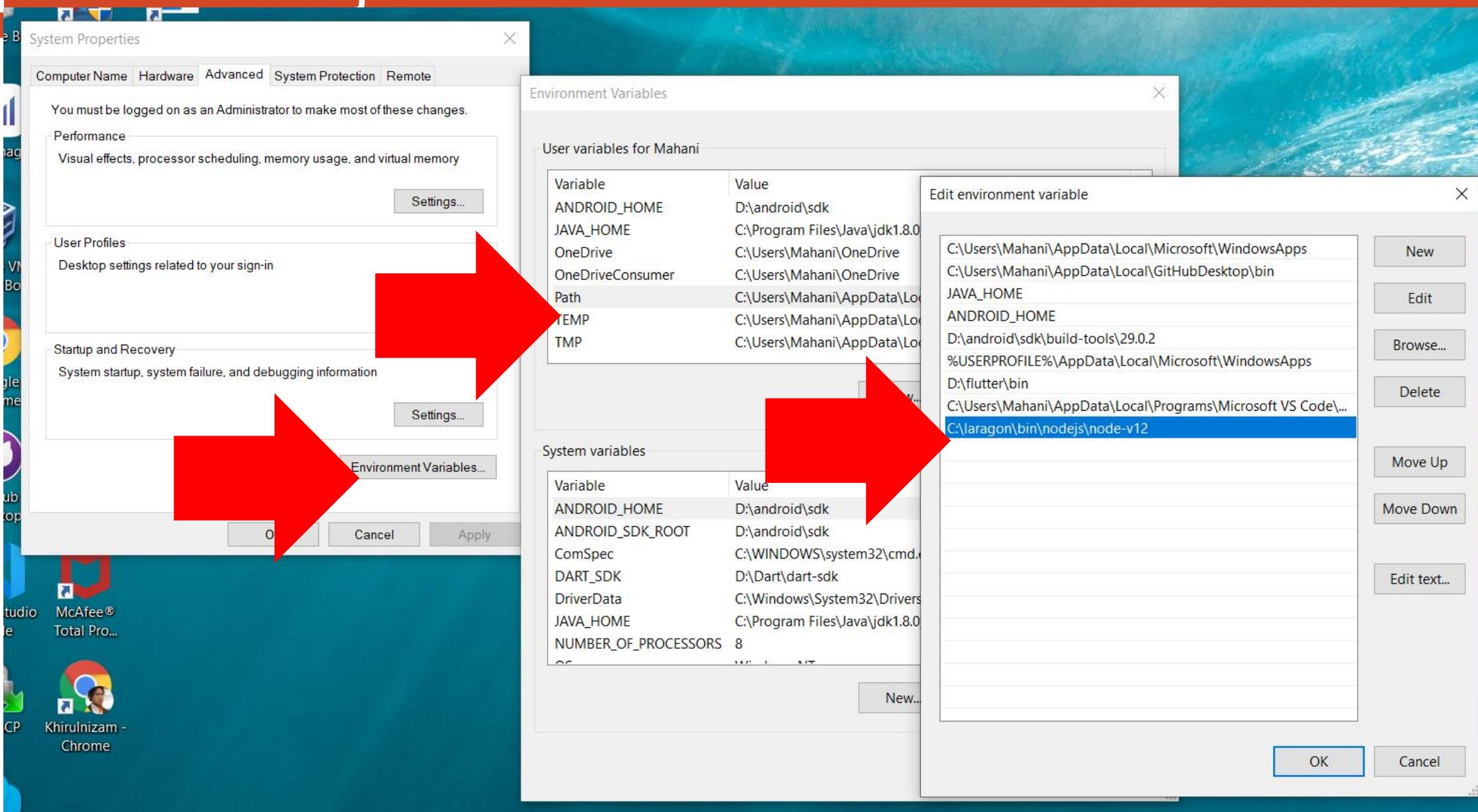
# Locate nodejs in Laragon



# Environment variables



# Set path to nodejs



# NODEJS.ORG

The screenshot shows the official Node.js website at [nodejs.org/en/](https://nodejs.org/en/). The page features a dark header with the Node.js logo and navigation links for HOME, ABOUT, DOWNLOADS, DOCS, GET INVOLVED, SECURITY, and NEWS. Below the header, a large green button labeled "Download for Windows (x64)" is highlighted with a red rounded rectangle. This button has two variants: "12.16.1 LTS" (Recommended For Most Users) and "13.11.0 Current" (Latest Features). Both variants include links to "Other Downloads", "Changelog", and "API Docs". A note below the buttons suggests looking at the "Long Term Support (LTS) schedule". At the bottom, there's a call to action to sign up for the "Node.js Everywhere" newsletter.

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

Download for Windows (x64)

**12.16.1 LTS**  
Recommended For Most Users

**13.11.0 Current**  
Latest Features

Other Downloads | Changelog | API Docs    Other Downloads | Changelog | API Docs

Or have a look at the [Long Term Support \(LTS\) schedule](#).

Sign up for [Node.js Everywhere](#), the official Node.js Monthly Newsletter.



LaravelFSTM  
@KVBalikPulau

# Laravel t11

## Middleware

## enforced Auth

[fstm.kuis.edu.my/blog/laravel](http://fstm.kuis.edu.my/blog/laravel)



# Middleware

Middleware provide a convenient mechanism for filtering HTTP requests entering your application.

For example, Laravel includes a middleware that verifies the user of your application is authenticated. If the user is not authenticated, the middleware will redirect the user to the login screen.

```
Route::get('/home', 'HomeController@index')->name('home')->middleware('auth');
```

```
public function __construct()  
{  
    $this->middleware('auth');  
}
```

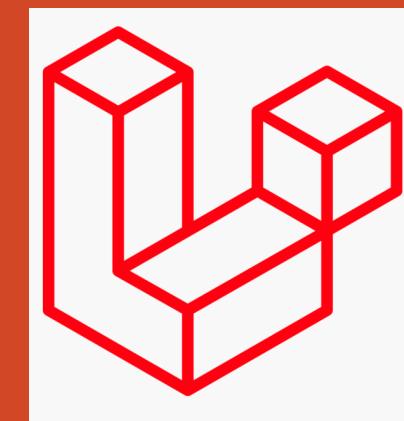
# Add to dashboard menu

The screenshot shows a Sublime Text editor window with several tabs open. The active tab is 'app.blade.php' located at C:\laragon\www\lara7\resources\views\layouts\app.blade.php. The code in this tab is a Laravel blade template for a dashboard menu. The code includes a body section with a main navigation bar and a container for menu items. Two specific menu items are added: one for 'Insert' (with a href to 'trainings/create') and another for 'Listing/Search' (with a href to 'trainings'). The code uses Laravel's URL helper function {{ url() }} for generating URLs.

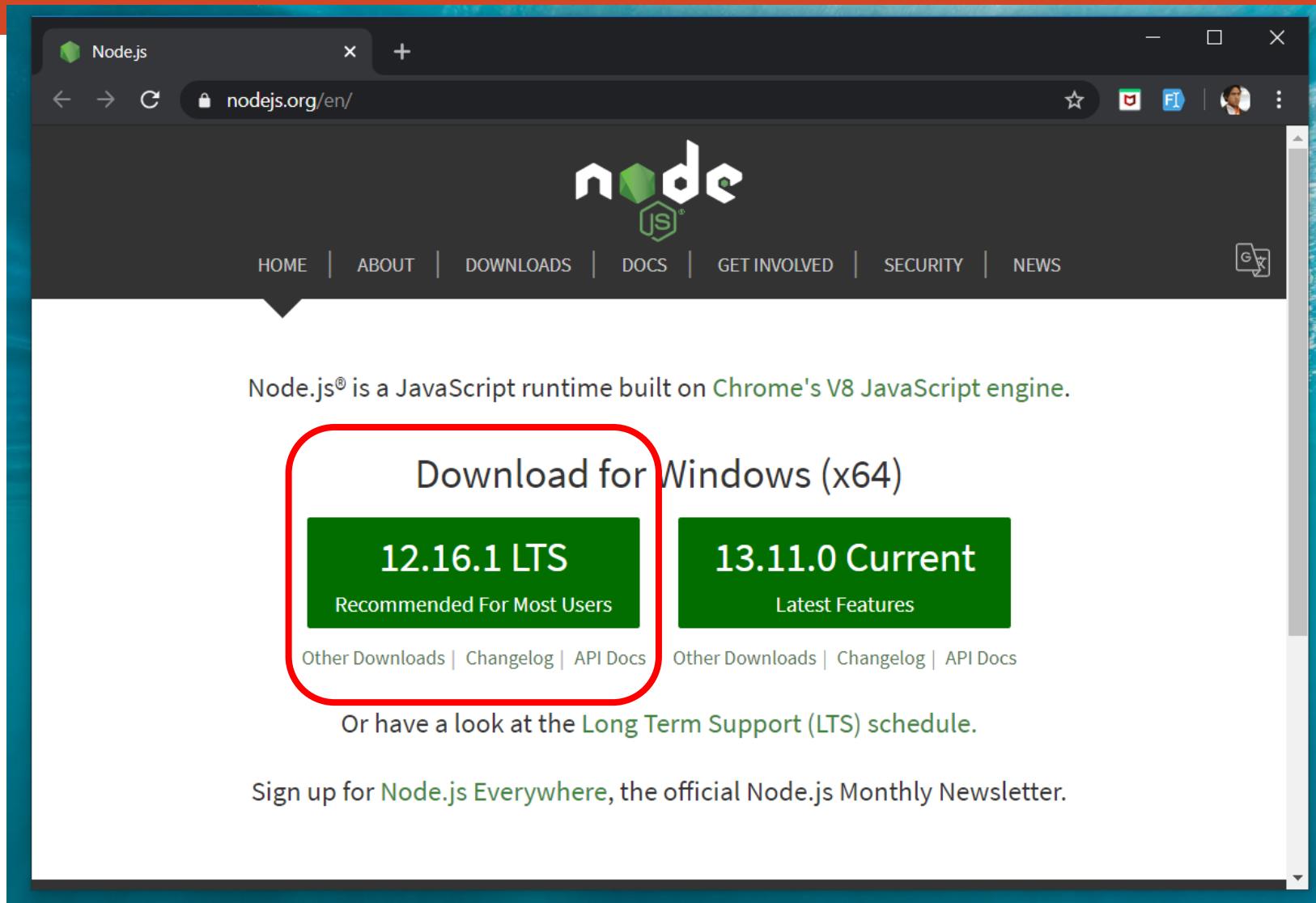
```
22 <body>
23     <div id="app">
24         <nav class="navbar navbar-expand-md navbar-light
25             bg-white shadow-sm">
26             <div class="container">
27                 <a class="navbar-brand" href="{{ url('/') }}">
28                     {{ config('app.name', 'FSTM Laravel') }}
29                 </a>
30                 <!-- menu -->
31                 <a href="{{ url('trainings/create') }}">Insert
32                 </a>
33                 ||
34                 <a href="{{ url('trainings') }}">Listing/Search
35                 </a>
36             <!-- end menu -->
```

Laravel t12

# Test Laravel project from Github repo



# NODEJS.ORG



The screenshot shows the official Node.js website at [nodejs.org/en/](https://nodejs.org/en/). The page features a dark header with the Node.js logo and navigation links for HOME, ABOUT, DOWNLOADS, DOCS, GET INVOLVED, SECURITY, and NEWS. A large green button labeled "Download for Windows (x64)" is highlighted with a red border. Below it are two green buttons: "12.16.1 LTS" (labeled "Recommended For Most Users") and "13.11.0 Current" (labeled "Latest Features"). At the bottom, there's a link to the "Long Term Support (LTS) schedule".

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

Download for Windows (x64)

**12.16.1 LTS**  
Recommended For Most Users

**13.11.0 Current**  
Latest Features

Other Downloads | Changelog | API Docs   Other Downloads | Changelog | API Docs

Or have a look at the [Long Term Support \(LTS\) schedule](#).

Sign up for [Node.js Everywhere](#), the official Node.js Monthly Newsletter.

# Github.com/khirulnizam/lara7

github.com/khirulnizam/lara7

1 commit 1 branch 0 packages 0 releases 1 contributor

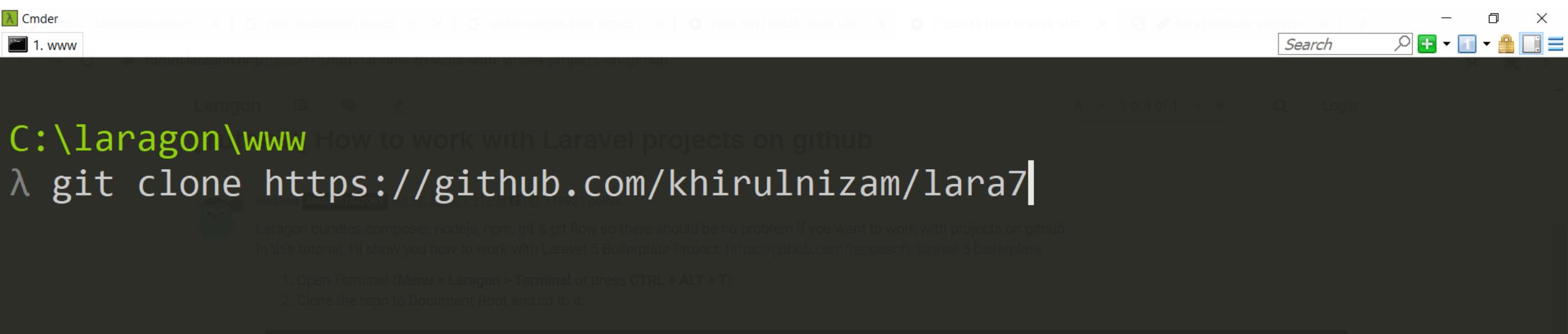
Branch: master ▾ New pull request Find file Clone or download ▾

File/Folder	Commit Message	Last Commit
app	Initial commit	16 minutes ago
bootstrap	Initial commit	16 minutes ago
config	Initial commit	16 minutes ago
database	Initial commit	16 minutes ago
public	Initial commit	16 minutes ago
resources	Initial commit	16 minutes ago
routes	Initial commit	16 minutes ago

Clone with HTTPS ⓘ  
Use Git or checkout with SVN using the web URL.  
<https://github.com/khirulnizam/lara7.git> 

[Open in Desktop](#) [Download ZIP](#) 16 minutes ago

# Git clone



A screenshot of a web browser window. In the background, a terminal window titled 'Cmder' is visible, showing the command 'git clone https://github.com/khirulnizam/lara7'. The browser window has a search bar with 'Search' and a list of tabs. The active tab shows a GitHub page for a tutorial on working with Laravel projects on GitHub.

```
C:\laragon\www How to work with Laravel projects on github
λ git clone https://github.com/khirulnizam/lara7
```

Laragon bundles composer, nodejs, npm, git & git flow so there should be no problem if you want to work with projects on github.  
In this tutorial, I'll show you how to work with Laravel 5 Boilerplate Project: <https://github.com/rappasoft/laravel-5-boilerplate>

1. Open Terminal (**Menu > Laragon > Terminal or press CTRL + ALT + T**)
2. Clone the repo to Document Root and cd to it:

# composer & npm install

The screenshot shows a Cmder terminal window with the title bar "Cmder" and tab "1. larat". The terminal content is as follows:

```
C:\laragon\www
λ cd larat

C:\laragon\www\larat
λ composer install

C:\laragon\www\larat
λ npm install
```

The background of the terminal window features a dark theme with a faint watermark of the Laragon logo.

## .env file

```
6  
7 LOG_CHANNEL=stack  
8  
9 DB_CONNECTION=mysql  
10 DB_HOST=127.0.0.1  
11 DB_PORT=3306  
12 DB_DATABASE=lara7  
13 DB_USERNAME=root  
14 DB_PASSWORD=abcd1234  
15
```

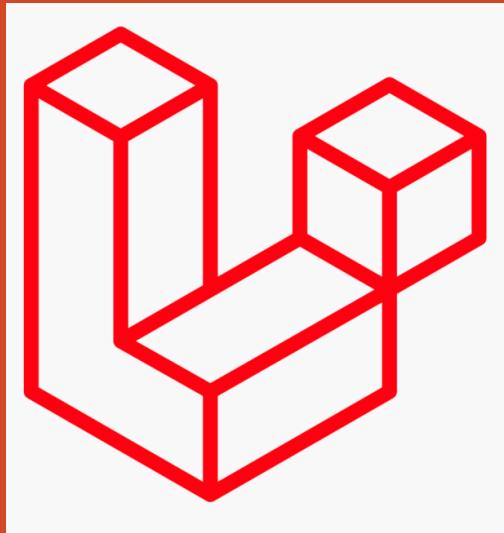
# php artisan migrate & seed

Inside Laravel project folder

php artisan key:generate

php artisan migrate

php artisan db:seed



# Laravel t13

## Show records

### (Paparan semua data dalam rekod)

# Table with lots of column

The screenshot shows the HeidiSQL Portable interface connected to a MySQL database named 'mycompanyhr'. The current table is 'employee' with 29 rows total. The interface includes a left sidebar for navigating databases and tables, and a top menu bar with various options like File, Edit, Search, Tools, Go to, Help, and a 'Donate' button.

EMPNO	FIRSTNAME	MIDINT	LASTNAME	WORKDEPT	PHONENO	HIREDATE	JOB	EDLEV...	GENDER	BIRTHDATE	SALARY	BONUS	COMM
000341	KHIRULNIZAM	B	ABD RAHMAN	C01	1111	2000-01-29	PROGRAMM	17	M	1976-08-18	36,000.00	6,000.00	2,000.00
000020	MICHAEL	L	THOMPSONS	C01	2222	1973-10-10	MANAGER	18	M	1948-02-02	41,250.00	800.00	3,300.00
000050	JOHN	B	GEYER	E01	6789	1949-08-17	MANAGER	16	M	1925-09-15	40,175.00	800.00	3,214.00
000070	EVA	D	PULASKI	D21	7831	1980-09-30	MANAGER	16	F	1953-05-26	36,170.00	700.00	2,893.00
000090	EILEEN	W	HENDERSON	E11	5498	1970-08-15	MANAGER	16	F	1941-05-15	29,750.00	600.00	2,380.00
000100	THEODORE	Q	SPENSER	E21	0972	1980-06-19	MANAGER	14	M	1956-12-18	26,150.00	500.00	2,092.00
000110	VINCENZO	G	LUCCHESI	A00	3490	1958-05-16	SALESREP	19	M	1929-11-05	46,500.00	900.00	3,720.00
000120	SEAN		O'CONNELL	A00	2167	1963-12-05	CLERK	14	M	1942-10-18	29,250.00	600.00	2,340.00
000130	DOLORES	M	QUINTANA	C01	4578	1971-07-28	ANALYST	16	F	1925-09-15	23,800.00	500.00	1,904.00
000140	HEATHER	A	NICHOLLS	C01	1793	1976-12-15	ANALYST	18	F	1946-01-19	28,420.00	600.00	2,274.00
000150	BRUCE		ADAMSON	D11	4510	1972-02-12	DESIGNER	16	M	1947-05-17	25,280.00	500.00	2,022.00
000160	ELIZABETH	R	PIANKA	D11	3782	1977-10-11	DESIGNER	17	F	1955-04-12	22,250.00	400.00	1,780.00
000180	MARILYN	S	SCOUTTEN	D11	1682	1973-07-07	DESIGNER	17	F	1949-02-21	21,340.00	500.00	1,707.00
000190	JAMES	H	WALKER	D11	2986	1974-07-26	DESIGNER	16	M	1952-06-25	20,450.00	400.00	1,636.00
000200	DAVID		BROWN	D11	4501	1966-03-03	DESIGNER	16	M	1941-05-29	27,740.00	600.00	2,217.00
000210	WILLIAM	T	JONES	D11	0942	1979-04-11	DESIGNER	17	M	1953-02-23	18,270.00	400.00	1,462.00
000220	JENNIFER	K	LUTZ	D11	0672	1968-08-29	DESIGNER	18	F	1948-03-19	29,840.00	600.00	2,387.00
000230	JAMES	J	JEFFERSON	D21	2094	1966-11-21	CLERK	14	M	1935-05-30	22,180.00	400.00	1,774.00
000240	SALVATORE	M	MARINO	D21	3780	1979-12-05	CLERK	17	M	1954-03-31	28,760.00	600.00	2,301.00
000250	DANIEL	S	SMITH	D21	0961	1969-10-30	CLERK	15	M	1939-11-12	19,180.00	400.00	1,534.00
000260	SYBIL	P	JOHNSON	D21	8953	1975-09-11	CLERK	16	F	1936-10-05	17,250.00	300.00	1,380.00
000270	MARIA	L	PEREZ	D21	9001	1980-09-30	CLERK	15	F	1953-05-26	27,380.00	500.00	2,190.00
000280	ETHEL	R	SCHNEIDER	E11	8997	1967-03-24	OPERATOR	17	F	1936-03-28	26,250.00	500.00	2,100.00
000290	JOHN	R	PARKER	E11	4502	1980-05-30	OPERATOR	12	M	1946-07-09	15,340.00	300.00	1,227.00
000300	PHILIP	X	SMITH	E11	2095	1972-06-19	OPERATOR	14	M	1936-10-27	17,750.00	400.00	1,420.00
000310	MAUDE	F	SETRIGHT	E11	3332	1964-09-12	OPERATOR	12	F	1931-04-21	15,900.00	300.00	1,272.00
000320	RAMLAL	V	MEHTA	E21	9990	1965-07-07	FIELDREP	16	M	1932-08-11	19,950.00	400.00	1,596.00

# Github.com/khirulnizam/lara7

Slides  
&  
codes

The screenshot shows a GitHub repository page for 'lara7'. At the top, it displays summary statistics: 1 commit, 1 branch, 0 packages, 0 releases, and 1 contributor. Below this, there are buttons for 'Branch: master' (with a dropdown arrow), 'New pull request', 'Find file', and 'Clone or download' (in a green button). A list of files and their commit details follows:

File	Commit Details	Last Commit
app	Initial commit	16 minutes ago
bootstrap	Initial commit	16 minutes ago
config	Initial commit	16 minutes ago
database	Initial commit	16 minutes ago
public	Initial commit	16 minutes ago
resources	Initial commit	16 minutes ago
routes	Initial commit	16 minutes ago

To the right of the file list, there is a 'Clone with HTTPS' section with a URL: <https://github.com/khirulnizam/lara7.git>. Below this are two buttons: 'Open in Desktop' and 'Download ZIP', with 'Download ZIP' being highlighted in blue.

# DISCLAIMER

Codes are for demo purposes. Examples showed here are only to validate the proof of concepts. Security aspects the demonstrated examples are beyond the coverage of these video tutorials. You are advised to explore / search after advanced/security topics of PHP web programming as these tutorials only for beginner level.

Suitable to view in DESKTOP

# Views

Views are stored in the resources/views directory

1. Go to resources/views
2. Go to trainings folder, create:
  1. show.blade.php
  2. edit.blade.php

```
<resources>
  > js
  > lang
  > sass
  > views
    > admin
    > auth
    > layouts
  > trainings
    => create.blade.php
    => edit.blade.php
    => index.blade.php
    => show.blade.php
```

# Controller

7 method in controllers

## Make Controller

```
php artisan make:controller TrainingController
```

### Action

---

-index

---

-create

---

-store

---

show

---

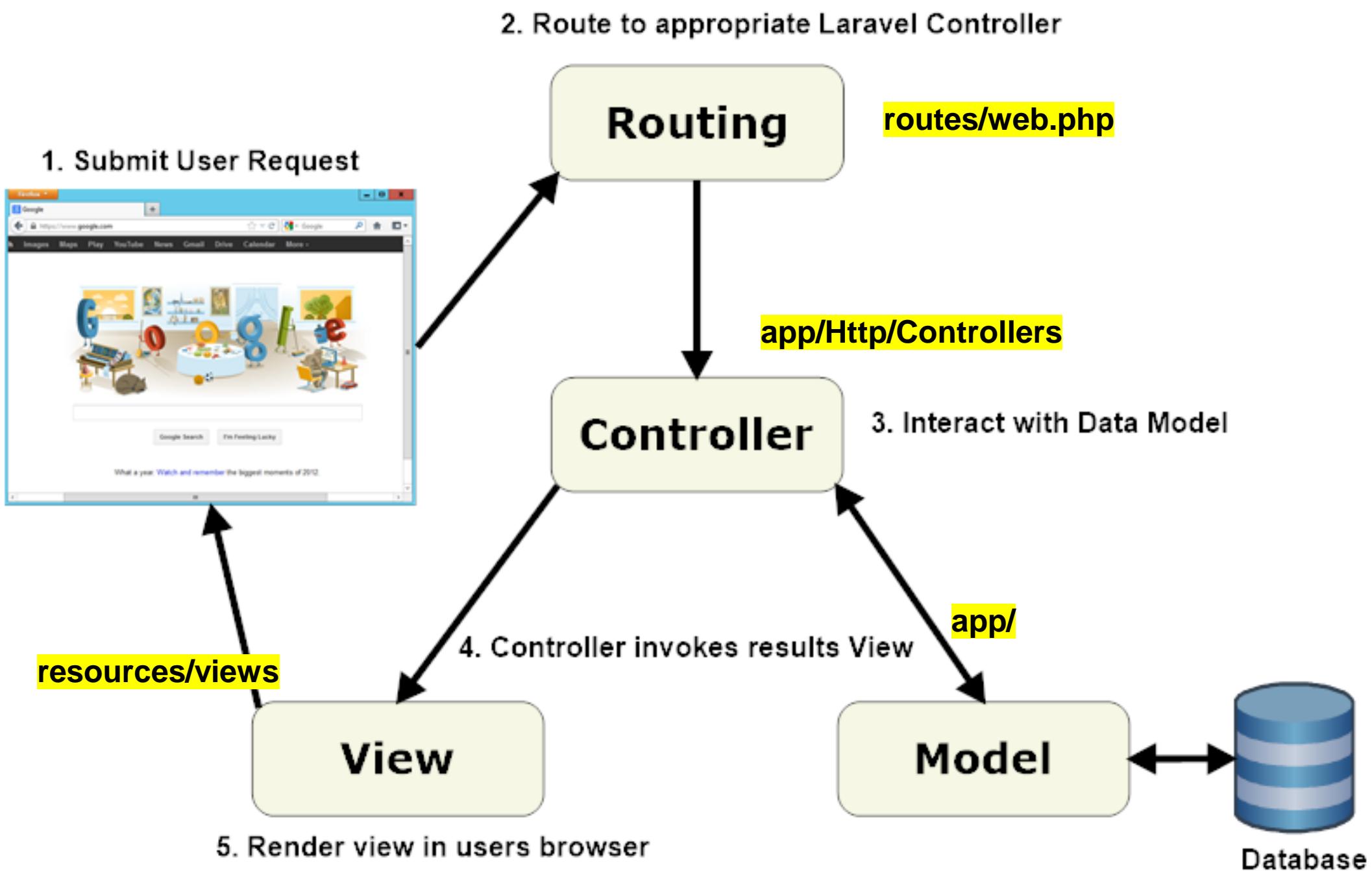
edit

---

update

---

destroy



# Controller - Show

Description	URL	Controller	View File
		Function	
Show one of the trainings.	GET example.com/trainings/{id}	show()	app/views/trainings/show.blade.php

```
public function show($id)
{
    $training=Training::find($id);
    return view('trainings.show')->with(compact('training'));
}
```

# Views – Show View

Views are stored in the resources/views directory

1. Go to resources/views/trainings/show.blade.php

# Github.com/khirulnizam/lara7

```
@extends('layouts.app')

@section('content')


Show Training



Title



Description
{{ $training->title }}



Trainer



Back to Training Index


```

# Github.com/khirulnizam/lara7

Slides  
&  
codes

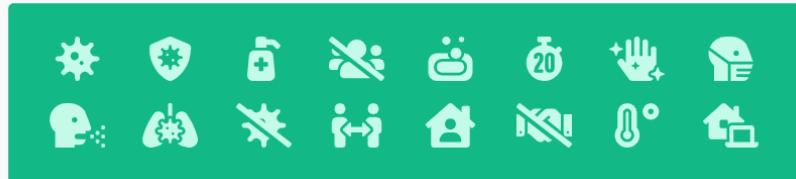
The screenshot shows a GitHub repository page for the user 'khirulnizam' with the repository name 'lara7'. The page displays the following statistics:

- 1 commit
- 1 branch
- 0 packages
- 0 releases
- 1 contributor

The repository structure on the left includes:

- khirulnizam Initial commit
- app
- bootstrap
- config
- database
- public
- resources
- routes

On the right, there is a 'Clone with HTTPS' section with a URL: <https://github.com/khirulnizam/lara7.git>. Below it are two buttons: 'Open in Desktop' and 'Download ZIP'. The 'Download ZIP' button is highlighted in blue. The commits listed on the right are all 'Initial commit' and occurred 16 minutes ago.



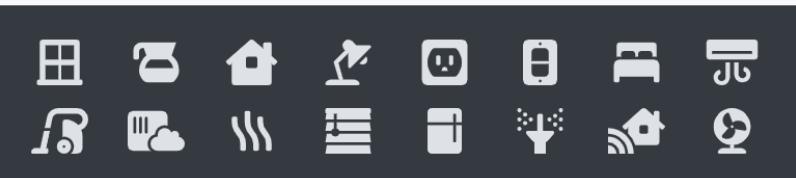
COVID-19: Help spread awareness and not the virus with these free icons!



Our new Science Fiction icons are ready to beam up.



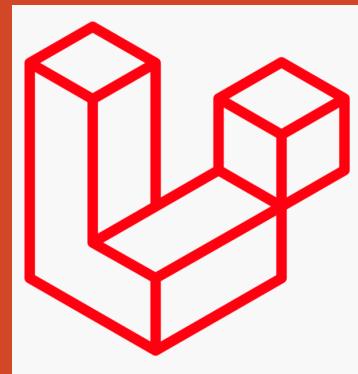
Ready for a close encounter? Check out our new space icons.



Hey Siri, turn up the heat on our new household icons.

# Laravel t14

## Font-awesome thru CDN (Pasang icon font-awesome guna CDN)



# Github.com/khirulnizam/lara7

Slides  
&  
codes

github.com/khirulnizam/lara7

1 commit 1 branch 0 packages 0 releases 1 contributor

Branch: master New pull request Find file Clone or download

khirulnizam Initial commit

app Initial commit

bootstrap Initial commit

config Initial commit

database Initial commit

public Initial commit

resources Initial commit

routes Initial commit

Clone with HTTPS ⓘ  
Use Git or checkout with SVN using the web URL.  
<https://github.com/khirulnizam/lara7.git>

16 minutes ago

# CDN – CONTENT DELIVERY NETWORK

- CDN – load library/ framework from public host server
- to provide high availability and performance
- **<https://www.bootstrapcdn.com/fontawesome/>**

## Font Awesome

**Font Awesome CSS**

```
https://stackpath.bootstrapcdn.com/fontawesome/4.7.0/css/font-awesome.min.css
```

Click to copy ▾

**HTML**

```
<link href="https://stackpath.bootstrapcdn.com/fontawesome/4.7.0/css/font-awesome.min.css" rel="stylesheet"/>
```

Copied text to clipboard

**Pug**

```
link(href="https://stackpath.bootstrapcdn.com/fontawesome/4.7.0/css/font-awesome.min.css", rel="stylesheet")
```

Click to copy

**Haml**

```
%link{href: "https://stackpath.bootstrapcdn.com/fontawesome/4.7.0/css/font-awesome.min.css", rel: "stylesheet"}
```

Click to copy

# FONTAWESOME IN SIMPLE HTML

Icons Chart

[Icons Currency](#)

[Icons Directional](#)

[Icons File Type](#)

[Icons Form](#)

[Icons Gender](#)

[Icons Hand](#)

[Icons Medical](#)

[Icons Payment](#)

[Icons Spinner](#)

**Icons Text**

[Icons Transportation](#)

[Icons Video](#)

[Icons Web Application](#)

**Bootstrap**

[Icons BS Glyphicons](#)

**Google**

[Google Icons Intro](#)

[Icons Action](#)

[Icons Alert](#)

## Text Icons

The table below shows all Font Awesome Text icons:

Icon	Description	Example
	fa fa-align-center	<a href="#">Try it</a>
	fa fa-align-justify	<a href="#">Try it</a>
	fa fa-align-left	<a href="#">Try it</a>
	fa fa-align-right	<a href="#">Try it</a>
	fa fa-bold	<a href="#">Try it</a>
	fa fa-chain	<a href="#">Try it</a>
	fa fa-chain-broken	<a href="#">Try it</a>
	fa fa-clipboard	<a href="#">Try it</a>
	fa fa-columns	<a href="#">Try it</a>
	fa fa-copy	<a href="#">Try it</a>

# Import font-awesome from CDN

```
<html>
<head>
<title>Font Awesome Icons</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
</head>
<body>

<h1>fa fa-file-text</h1>

<i class="fa fa-file-text"></i>
<i class="fa fa-file-text" style="font-size:24px"></i>
<i class="fa fa-file-text" style="font-size:36px;"></i>
<i class="fa fa-file-text" style="font-size:48px;color:red"></i>
<br>

<p>Used on a button:</p>
<button style="font-size:24px">Show <i class="fa fa-file-text"></i>
</button>

</body>
</html>
```

## Tag: <i></i>

```
<i class="fa fa-file-text"></i>
```

<p>Used on a button:</p>

```
<button>
```

Show <**i class="fa fa-file-text"**></i>

```
</button>
```

**fa fa-file-text**



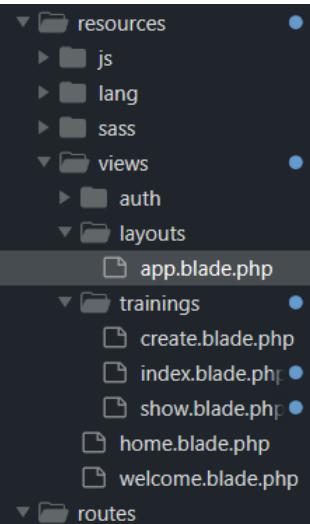
Used on a button:



Unicode:



# Import font-awesome CDN in Laravel layout



18

19

20

```
<link href="https://stackpath.bootstrapcdn.com/  
font-awesome/4.7.0/css/font-awesome.min.css" rel="  
stylesheet" integrity="sha384-wvfXpqpZZVQGK6TAh5PVlGO  
fQNHSoD2xbE+QkPxCAF1NEevoEH3S10sibVcOQVnN"  
crossorigin="anonymous">
```

- <https://www.bootstrapcdn.com/fontawesome/>

## Font Awesome

### Font Awesome CSS

```
https://stackpath.bootstrapcdn.com/fontawesome/4.7.0/css/font-awesome.min.css
```

Click to copy

### HTML

```
<link href="https://stackpath.bootstrapcdn.com/fontawesome/4.7.0/css/font-awesome.min.css" rel="styl
```

Copied text to clipboard

### Pug

```
link(href="https://stackpath.bootstrapcdn.com/fontawesome/4.7.0/css/font-awesome.min.css", rel="style
```

Click to copy

### Haml

```
%link{href: "https://stackpath.bootstrapcdn.com/fontawesome/4.7.0/css/font-awesome.min.css", rel: "styl
```

Click to copy

# Implement in buttons

Training Index

<b>id</b>	<b>Title</b>	<b>Task</b>
1	Laravel Basics	
2	Laravel Intermediate	
3	Laravel Advanced with Unit Test	

# Github.com/khirulnizam/lara7

Slides  
&  
codes

github.com/khirulnizam/lara7

1 commit 1 branch 0 packages 0 releases 1 contributor

Branch: master New pull request Find file Clone or download

khirulnizam Initial commit

app Initial commit

bootstrap Initial commit

config Initial commit

database Initial commit

public Initial commit

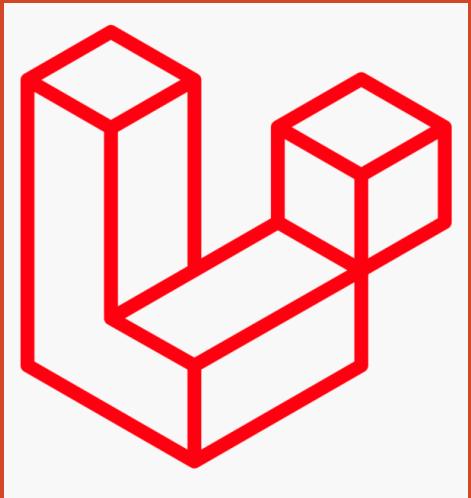
resources Initial commit

routes Initial commit

Clone with HTTPS ⓘ  
Use Git or checkout with SVN using the web URL.  
<https://github.com/khirulnizam/lara7.git>

16 minutes ago

16 minutes ago 16 minutes ago 16 minutes ago 16 minutes ago

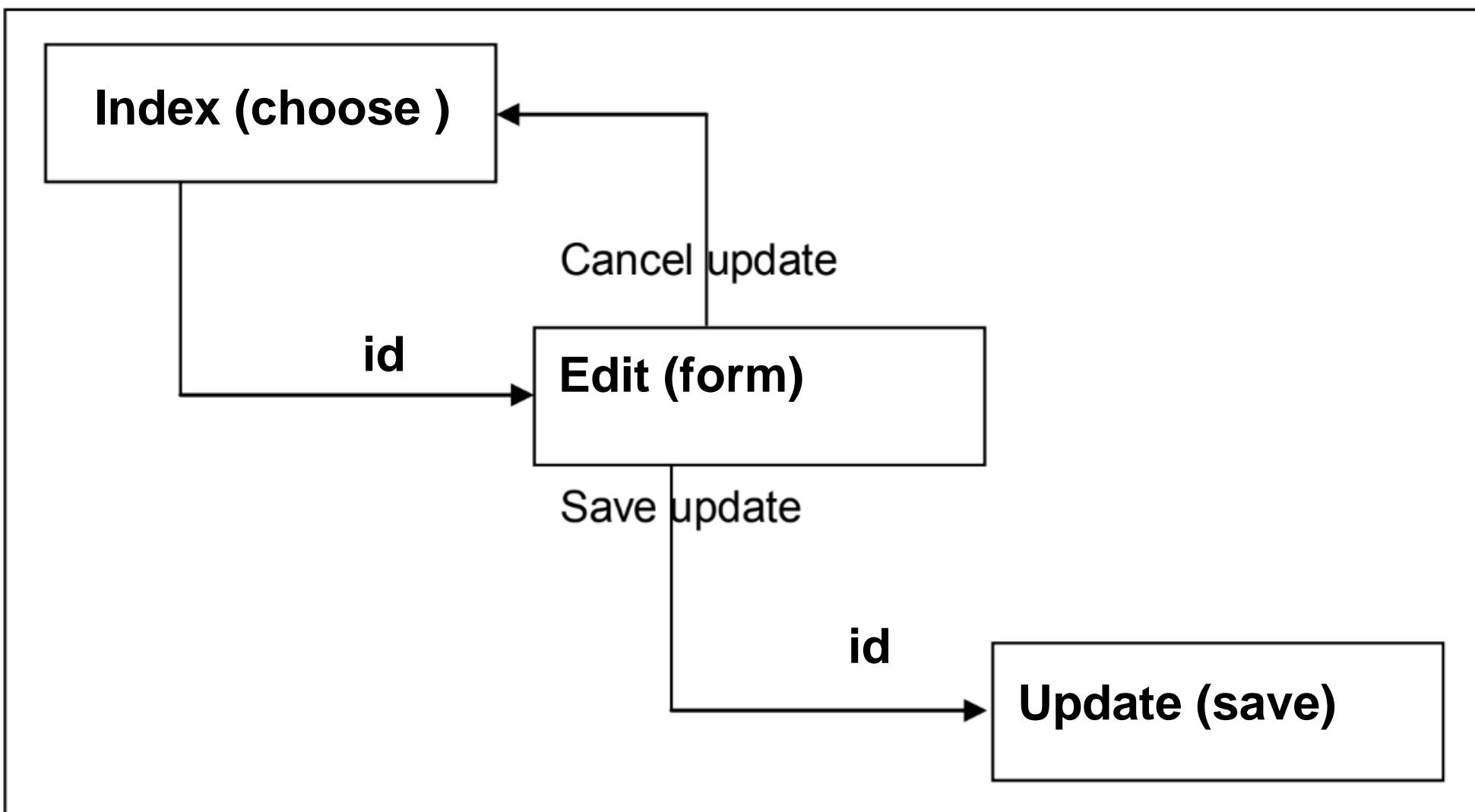


# Laravel t15

# Edit/Update

(Kemaskini data  
dalam rekod)

# UPDATE RECORD PROCESS



# DISCLAIMER

Codes are for demo purposes. Examples showed here are only to validate the proof of concepts. Security aspects the demonstrated examples are beyond the coverage of these video tutorials. You are advised to explore / search after advanced/security topics of PHP web programming as these tutorials only for beginner level.

Suitable to view in DESKTOP

# Github.com/khirulnizam/lara7

Slides &  
codes

T12 Tutorial  
clone/download  
github repo

The screenshot shows a GitHub repository page for 'lara7'. At the top, there's a header with navigation icons, a search bar, and user information. Below the header, the repository summary shows: 1 commit, 1 branch, 0 packages, 0 releases, and 1 contributor (khirulnizam). There are buttons for 'Branch: master' (with a dropdown arrow), 'New pull request', 'Find file', and 'Clone or download' (in green). The main content area displays a list of files and their commit history:

File	Commit Message	Date
app	Initial commit	16 minutes ago
bootstrap	Initial commit	16 minutes ago
config	Initial commit	16 minutes ago
database	Initial commit	16 minutes ago
public	Initial commit	16 minutes ago
resources	Initial commit	16 minutes ago
routes	Initial commit	16 minutes ago

To the right of the file list, there's a 'Clone with HTTPS' section with a URL (https://github.com/khirulnizam/lara7.git) and a clipboard icon. Below that are 'Open in Desktop' and 'Download ZIP' buttons, both in blue. A timestamp '16 minutes ago' is also present.

# Controller - Edit

Description	URL	Controller	View File
			Function
Pull a training from the database and allow editing.	GET example.com/trainings/{id}/edit	edit()	app/views/trainings/edit.blade.php

```
public function edit(Training $training)
{
    return view('trainings.edit')->with(compact('training'));
}
```

# Views – Edit View

Views are stored in the  
resources/views directory

Go to

resources/views/trainings/edit.blade.php

```

@extends('layouts.app')

@section('content')


<div class="row justify-content-center">
        <div class="col-md-8">
            <div class="card">
                <div class="card-header">Edit Training</div>

                <div class="card-body">
                    <form action="" method="POST">
                        @csrf
                        <div class="form-group">
                            <label for="title">Title</label>
                            <input type="text" class="form-control" name="title" value="{{ $training->title }}>
                        </div>
                        <div class="form-group">
                            <label for="description">Description</label>
                            <textarea name="description" cols="20" rows="10" class="form-control">{{ $training->title }}</textarea>
                        </div>
                        <div class="form-group">
                            <label for="trainer">Trainer</label>
                            <input type="text" class="form-control" name="trainer" value="{{ $training->trainer }}>
                        </div>
                        <div class="form-group">
                            <button type="submit" class="btn btn-primary">Update Training</button>
                            <a href="{{ route('training:index') }}" class="btn btn-link">Cancel</a>
                        </div>
                </div>
            </div>
        </div>
    </div>
@endsection


```

# Controller - Update

Description	URL	Controller Function	View File
Process the create form submit and save the training to the database.	POST example.com/trainings/{id}	update()	NONE



The screenshot shows a code editor with a file named `TrainingController.php` open. The file contains PHP code for a controller. The code includes a `update()` method that takes a `Request` object as a parameter. Inside the method, it finds a training record by its ID, updates it with fields from the request, and then returns a redirect response with a success message. The code editor has syntax highlighting and line numbers. A vertical orange bar highlights the `update()` method.

```
public function update(Request $request, $id)
{
    //save edited record
    $training = Training::find($id);
    $training->update($request->only(
        'title', 'description', 'trainer'));

    return redirect('/trainings')->with('success',
        'Training info has been updated!!');
}
```

http://fstm.kuis.edu.my/blog/laravel

# Output screen

Update Record

Title

Description

Trainer

**Save Update**

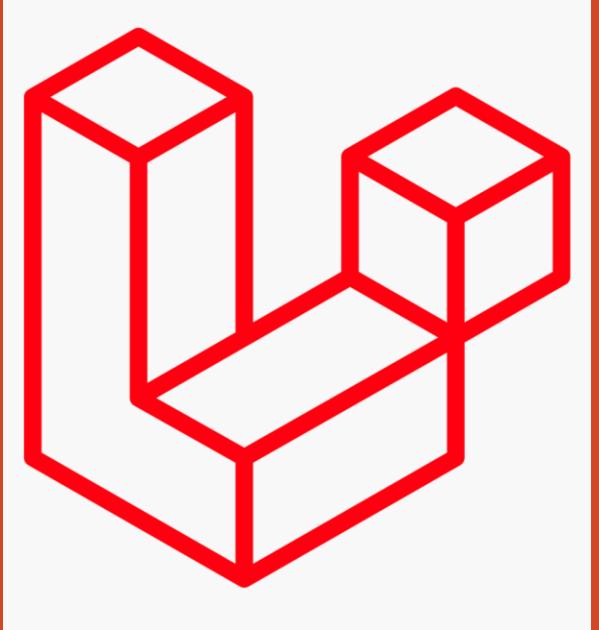
Training Index		
<input type="text"/>		
Training record has been updated		
Title		
Task		
Laravel Basics 1		
Laravel Intermediate		
Laravel Too Advanced		



# Laravel t16

## Delete records

(Padam rekod)



# DISCLAIMER

Codes are for demo purposes. Examples showed here are only to validate the proof of concepts. Security aspects the demonstrated examples are beyond the coverage of these video tutorials. You are advised to explore / search after advanced/security topics of PHP web programming as these tutorials only for beginner level.

Suitable to view in DESKTOP

# Github.com/khirulnizam/lara7

Slides &  
codes

T12 Tutorial  
clone/download  
github repo

The screenshot shows a GitHub repository page for the user 'khirulnizam' with the repository name 'lara7'. The page displays the following statistics:

- 1 commit
- 1 branch
- 0 packages
- 0 releases
- 1 contributor

Below the stats, there are buttons for 'Branch: master' (with a dropdown arrow), 'New pull request', 'Find file', and 'Clone or download' (in green). The repository structure is listed on the left, showing:

- khirulnizam Initial commit
- app Initial commit
- bootstrap Initial commit
- config Initial commit
- database Initial commit
- public Initial commit
- resources Initial commit
- routes Initial commit

On the right, there is a 'Clone with HTTPS' section with a URL: <https://github.com/khirulnizam/lara7.git>. Below it are two buttons: 'Open in Desktop' and 'Download ZIP' (highlighted in blue). The 'Download ZIP' button was last updated 16 minutes ago.

# Overview

Insert||Listing/Search

lara7.test says

Are you sure to delete the record (id:1) ?

OK

Cancel

## Training Index



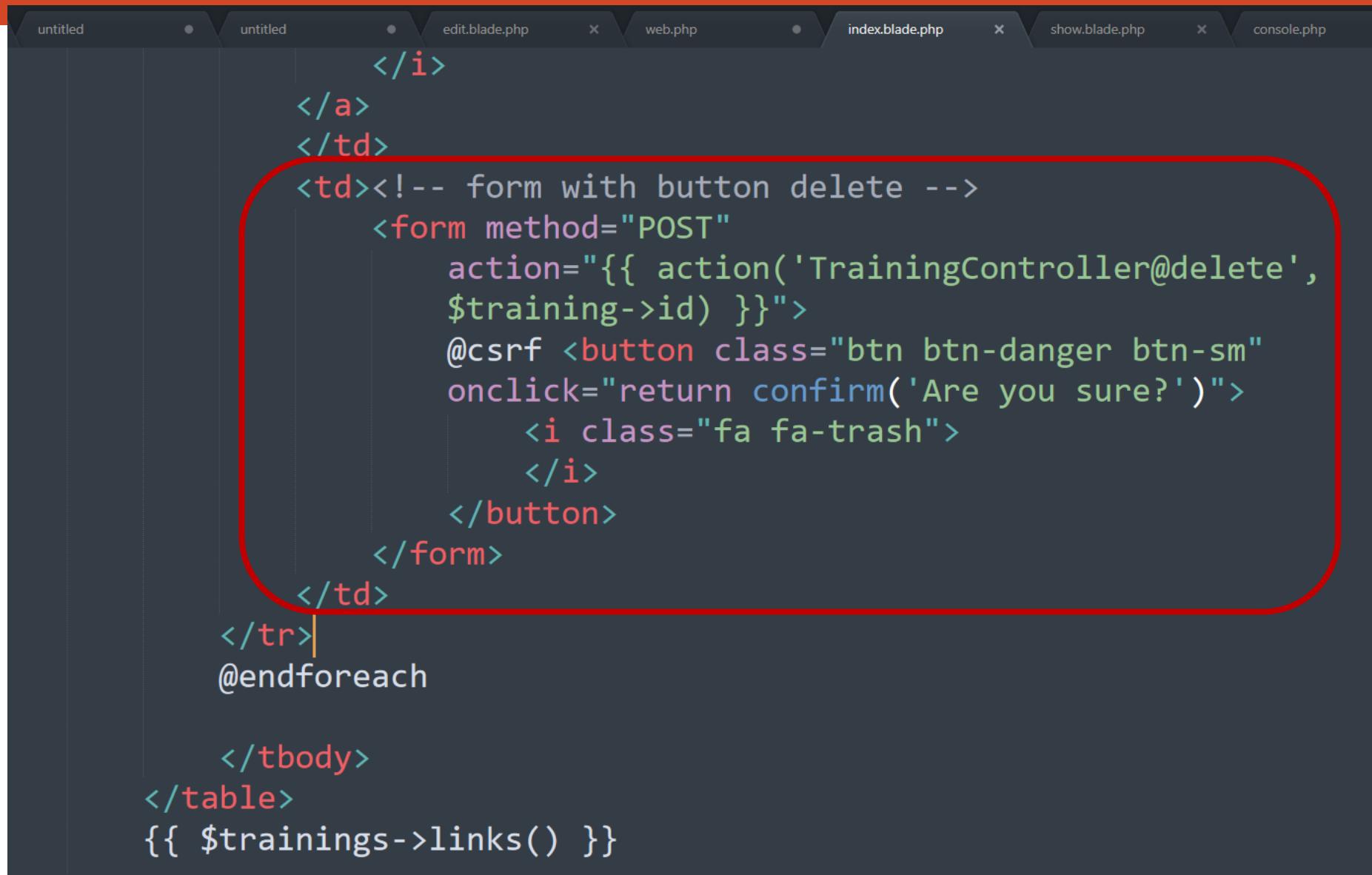
id	Title	Task
----	-------	------

1	Laravel Basics 1	
---	------------------	--

2	Laravel Intermediate	
---	----------------------	--

3	Laravel Too Advanced	
---	----------------------	--

# Delete button inside FORM



The screenshot shows a code editor with multiple tabs at the top: 'untitled', 'untitled', 'edit.blade.php', 'web.php', 'index.blade.php' (which is the active tab), 'show.blade.php', and 'console.php'. The code in the 'index.blade.php' tab is highlighted with a red oval. It contains Blade template syntax and PHP code. The highlighted section starts with a form inside a table cell:

```
<td><!-- form with button delete -->
<form method="POST"
      action="{{ action('TrainingController@delete',
      $training->id) }}"
      @csrf <button class="btn btn-danger btn-sm"
      onclick="return confirm('Are you sure?')">
        <i class="fa fa-trash">
        </i>
      </button>
    </form>
</td>
```

Below this, there's a closing tag and an tag. The code continues with a foreach loop:

```
</tr>
@endforeach
```

At the bottom of the visible code, there are closing and tags, followed by a PHP variable:

```
</tbody>
</table>
{{ $trainings->links() }}
```

# Routing – Group and Named Route

All Laravel routes are defined in your route files,  
which are located in the routes directory.

## routes

- api.php
- channels.php
- console.php
- web.php

<https://laravel.com/docs/7.x/routing>.

```
//group routing
//for training module
Route::group([
    'middleware'=>'auth',//lock this route group to only authenticated users
    'prefix'=>'trainings',
    'as'=>'training:',
],function(){
    Route::get('/', 'TrainingController@index')->name('index');
    Route::get('/create', 'TrainingController@create')->name('create');
    Route::post('/create', 'TrainingController@store')->name('store');
    Route::get('/{training}', 'TrainingController@show')->name('show');
    Route::get('/{training}/edit', 'TrainingController@edit')->name('edit');
    Route::post('/{training}', 'TrainingController@update')->name('update');
    Route::post('/{training}/delete', 'TrainingController@delete')->name('delete');
});
```

# Routing – Group and Named Route

```
php artisan route:list
```

[https://laravel.com/docs/7.x/routing.](https://laravel.com/docs/7.x/routing)

20	GET HEAD	/trainings	training:index	App\Http\Controllers\TrainingController@index	web,auth
	GET HEAD	/trainings/create	training:create	App\Http\Controllers\TrainingController@create	web,auth
	POST	/trainings/create	training@store	App\Http\Controllers\TrainingController@store	web,auth
	GET HEAD	/trainings/{training}	training@show	App\Http\Controllers\TrainingController@show	web,auth
	POST	/trainings/{training}	training@update	App\Http\Controllers\TrainingController@update	web,auth
21	POST	/trainings/{training}/delete	training(delete)	App\Http\Controllers\TrainingController@delete	web,auth
	GET HEAD	/trainings/{training}/edit	training:edit	App\Http\Controllers\TrainingController@edit	web,auth

```
D:\laragon\www\lara7 (master)
λ php artisan route:list
```

http://fstm.kuis.edu.my/blog/laravel

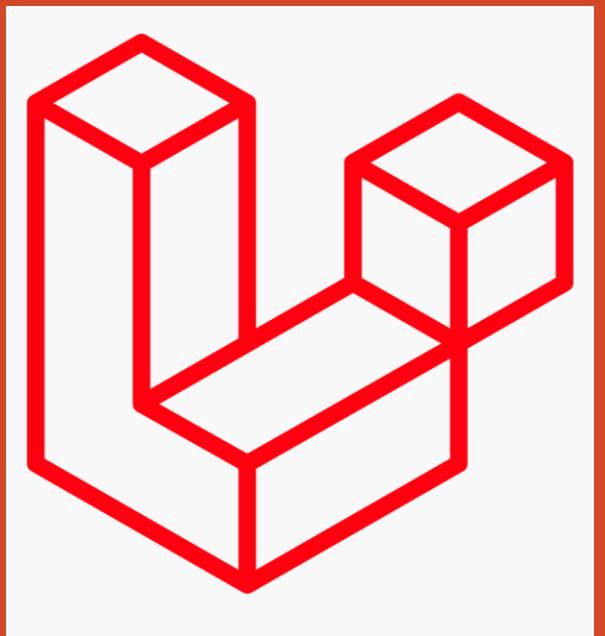
118

# Controller – Destroy/Delete

Description	URL	Controller Function	View File
Delete the training from DB.	GET example.com/trainings{id}/delete	delete()	NONE

```
public function delete(Training $training)
{
    $training->delete();

    return redirect('/trainings');
}
```



# Laravel t17 @csrf

(Built-in security token  
for post method)  
[bit.ly/fstmtubelaravel](http://bit.ly/fstmtubelaravel)

# DISCLAIMER

Codes are for demo purposes. Examples showed here are only to validate the proof of concepts. Security aspects the demonstrated examples are beyond the coverage of these video tutorials. You are advised to explore / search after advanced/security topics of PHP web programming as these tutorials only for beginner level.

Suitable to view in DESKTOP

# Github.com/khirulnizam/lara7

Slides &  
codes

T12 Tutorial  
clone/download  
github repo

The screenshot shows a GitHub repository page for 'lara7'. At the top, there's a header with navigation icons, a search bar, and user information. Below the header, the repository summary shows: 1 commit, 1 branch, 0 packages, 0 releases, and 1 contributor (khirulnizam). A 'Branch: master' dropdown and a 'New pull request' button are also present. On the right, there are 'Find file' and 'Clone or download' buttons. The main content area displays a list of files and their commit history:

File	Commit Message	Time
app	Initial commit	16 minutes ago
bootstrap	Initial commit	16 minutes ago
config	Initial commit	16 minutes ago
database	Initial commit	16 minutes ago
public	Initial commit	16 minutes ago
resources	Initial commit	16 minutes ago
routes	Initial commit	16 minutes ago

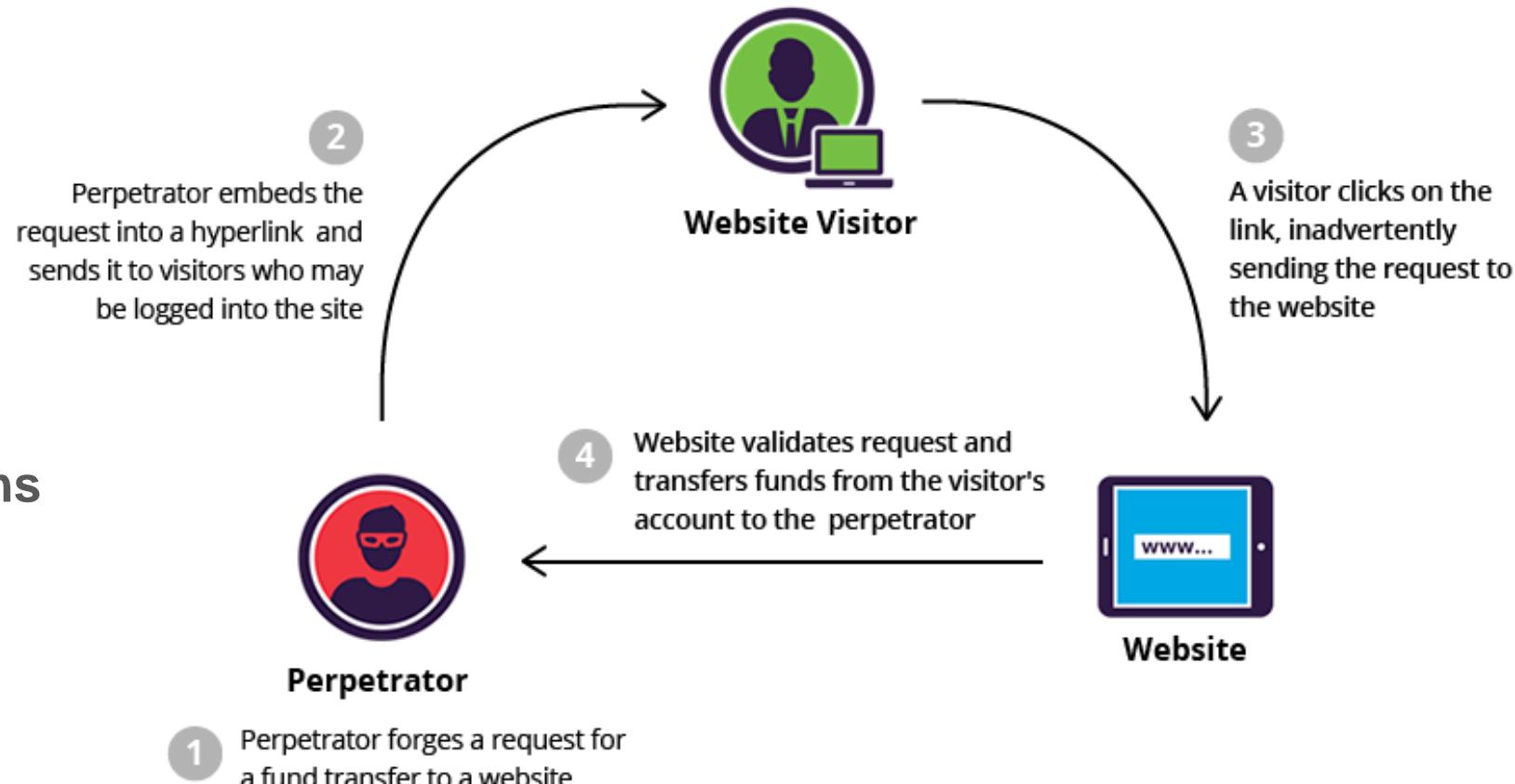
To the right of the file list, there's a 'Clone with HTTPS' section with a URL (https://github.com/khirulnizam/lara7.git) and a clipboard icon. Below that are 'Open in Desktop' and 'Download ZIP' buttons, both timestamped '16 minutes ago'.

# What is CSRF?

OWASP.org

## Cross-Site Request Forgery

- is an attack that forces an end user to **execute unwanted actions** on a web application in which they're currently authenticated.



# Implementation in Laravel blade

```
<!-- form start -->
<form action="{{action('TrainingController@update',
$training->id )}}" method="POST">
```

```
@csrf
```

```
Title
```

```
<input name="title" type="text"
class="form-control"
```

```
← → C ⓘ Not secure | view-source:lara7.test/trainings/1/edit
```

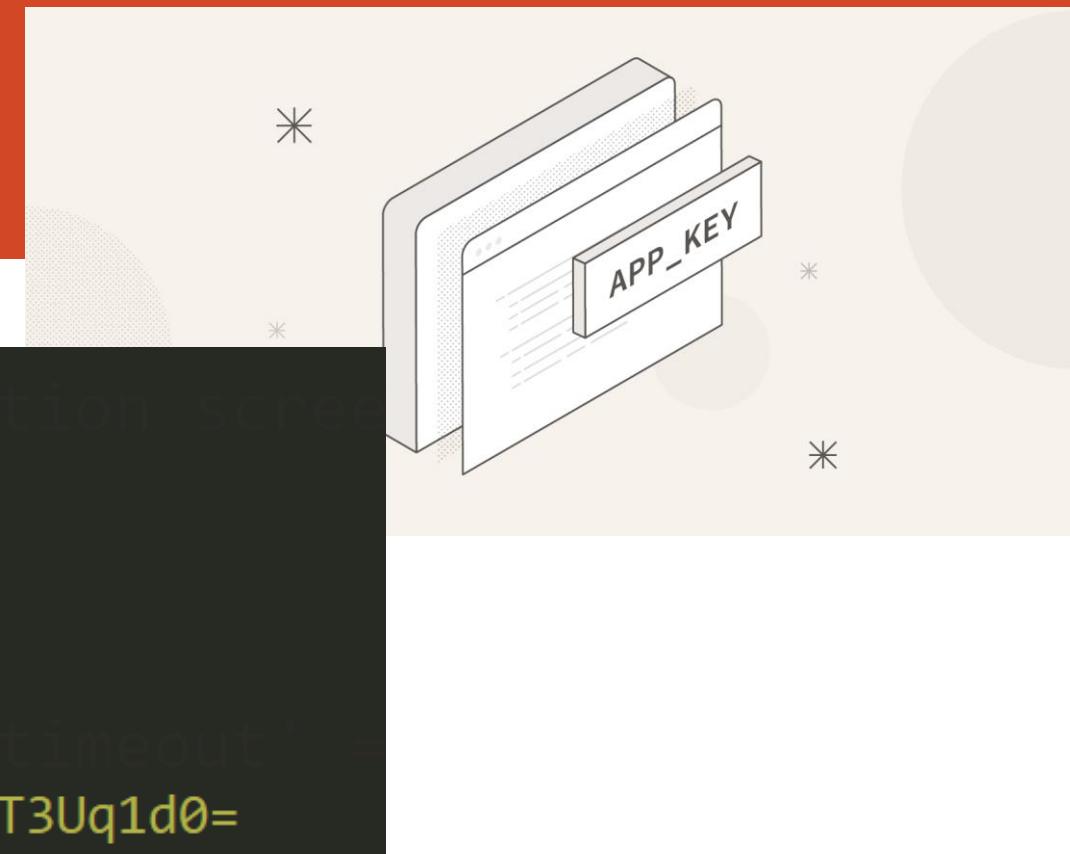
```
81
82 <!-- form start -->
83 <form action="http://lara7.test/trainings/1" method="POST">
84   <input type="hidden" name="_token"
85     value="r4SPb6vSY1tu03UHiZUphq2HsvogIJZYJjlBq2PX">
86   Title
87   <input name="title" type="text"
88     class="form-control"
89     value="Laravel Basics 1"
90   >
```

# Location of csrf app\_key

Use to hash security token

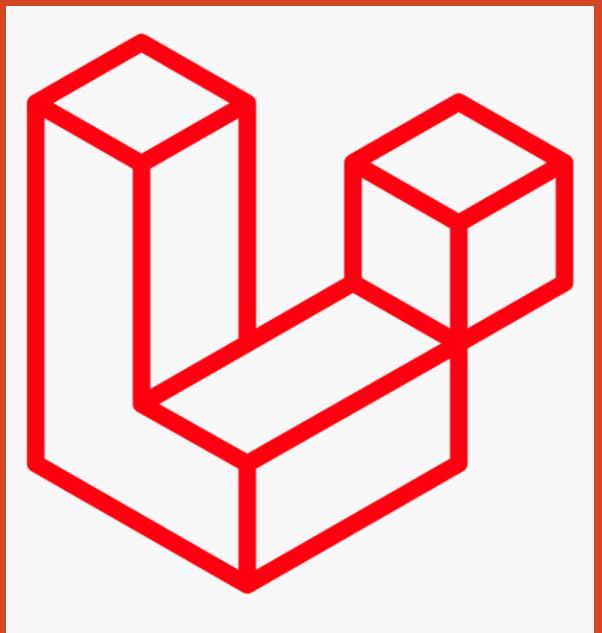
```
D:\laragon\www\lara7 (master)
λ php artisan key:generate
Application key set successfully.
```

```
D:\laragon\www\lara7 (master)
λ php artisan key:generate --show
base64:64tnzwWAKX9tvEIrYeOK+aQ+3A06SUtvvdzh0T3Uq1d0=
```



.env   x   TrainingController.php   x   untitled   ●   untitled   ●   edit.blade.php   x   app.php   x   auth.php   x   web.php   ●   index.blade.php   x

```
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:1sxP7anB8mPwR0m3oNUCHQiiGbngiCrrWNWPx0fD3mI=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
```



# Laravel t18

## Apply Bootstrap Layout (Pasang layout Bootstrap baru)

# Output overview

Search your prefered training...

Search

id	title	trainer	
1	Laravel Basics	Khirulnizam Abd Rahman	Show
2	Laravel Intermediate	Tarmizi Sanusi	Show
3	Laravel Advanced with Unit Test	Nasrul Hazim	Show

# DISCLAIMER

Codes are for demo purposes. Examples showed here are only to validate the proof of concepts. Security aspects the demonstrated examples are beyond the coverage of these video tutorials. You are advised to explore / search after advanced/security topics of PHP web programming as these tutorials only for beginner level.

Suitable to view in DESKTOP

# Github.com/khirulnizam/lara7

Slides &  
codes

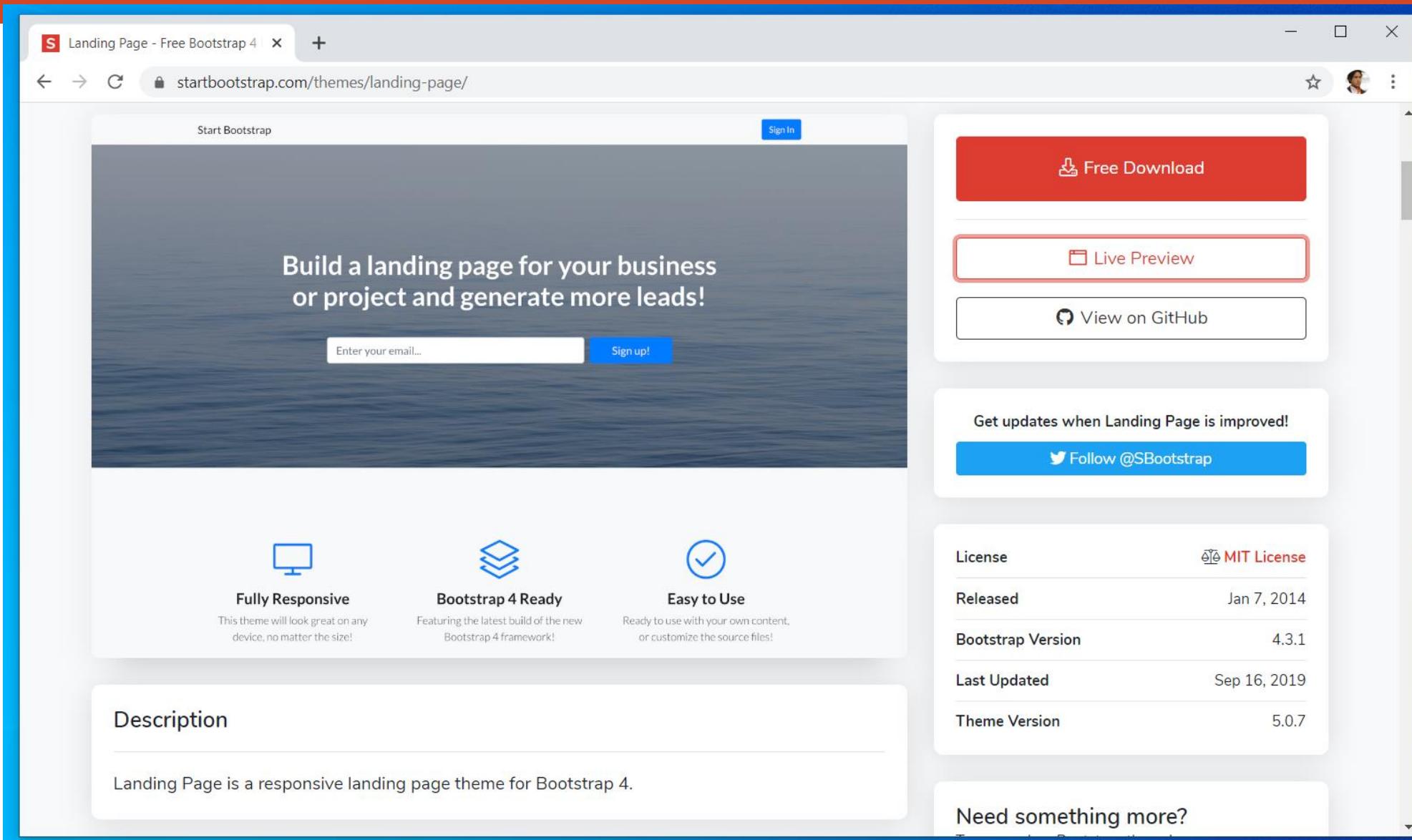
T12 Tutorial  
clone/download  
github repo

The screenshot shows a GitHub repository page for 'lara7'. At the top, there's a header with navigation icons, a search bar, and user information. Below the header, the repository summary shows: 1 commit, 1 branch, 0 packages, 0 releases, and 1 contributor (khirulnizam). A 'Branch: master' dropdown and a 'New pull request' button are also present. On the right, there are 'Find file' and 'Clone or download' buttons. The main content area displays a list of files and their commit history:

File	Commit Message	Time
app	Initial commit	16 minutes ago
bootstrap	Initial commit	16 minutes ago
config	Initial commit	16 minutes ago
database	Initial commit	16 minutes ago
public	Initial commit	16 minutes ago
resources	Initial commit	16 minutes ago
routes	Initial commit	16 minutes ago

To the right of the file list, there's a 'Clone with HTTPS' section with a URL (https://github.com/khirulnizam/lara7.git) and a 'Download ZIP' button.

# Theme from <https://startbootstrap.com/themes/landing-page/>



The screenshot shows a web browser displaying the "Landing Page - Free Bootstrap 4" theme from startbootstrap.com. The page has a dark blue header with the "Start Bootstrap" logo and a "Sign In" button. The main content area features a large image of ocean waves and the text "Build a landing page for your business or project and generate more leads!". Below this is a form with an "Enter your email..." input field and a "Sign up!" button. To the right, there's a sidebar with download options ("Free Download", "Live Preview", "View on GitHub"), social media links ("Follow @SBootstrap"), and a table of metadata including license, release date, bootstrap version, last update, and theme version. At the bottom, there's a "Description" section and a "Need something more?" link.

Landing Page - Free Bootstrap 4

startbootstrap.com/themes/landing-page/

Start Bootstrap

Sign In

Build a landing page for your business or project and generate more leads!

Enter your email... Sign up!

Free Download

Live Preview

View on GitHub

Get updates when Landing Page is improved!

Follow @SBootstrap

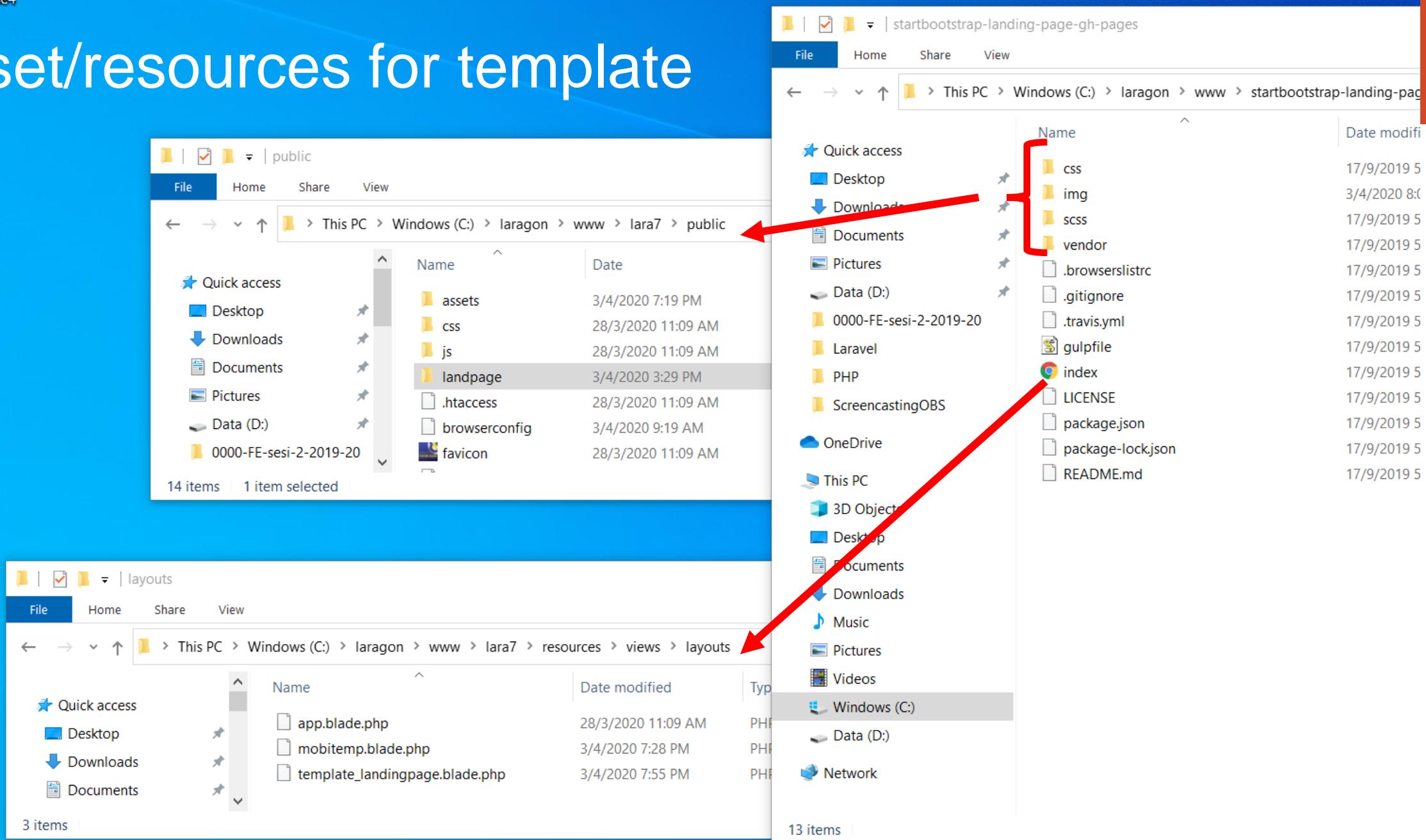
License	<a href="#">MIT License</a>
Released	Jan 7, 2014
Bootstrap Version	4.3.1
Last Updated	Sep 16, 2019
Theme Version	5.0.7

Description

Landing Page is a responsive landing page theme for Bootstrap 4.

Need something more?

# Asset/resources for template



# Blade asset( ) function – apply on both css & js

The image shows a code editor interface with a sidebar on the left displaying a file tree. The tree includes files like app.blade.php, mobitemp.blade.php, template\_landingpage.blade.php, trainings (create.blade.php, index.blade.php), routes (api.php, channels.php, console.php, web.php), storage, tests, vendor, .editorconfig, .env, .gitattributes, .gitignore, .styleci.yml, artisan, composer.json, composer.lock, and several presentation files (PPTX) and package files.

```
21    <!-- Custom fonts for this template -->
22    <link href="{{asset('landpage/vendor/
23        fontawesome-free/css/all.min.css')}}" rel="stylesheet">
24    <link href="{{asset('landpage/vendor/
25        simple-line-icons/css/simple-line-icons.css')}}" rel="stylesheet" type="text/css">
26    <link href="https://fonts.googleapis.com/css?family=Lato:300,400,700,300italic,400italic,700italic"
27        rel="stylesheet" type="text/css">
28
29    </head>
```

# @yield( )

The screenshot shows a code editor with several tabs at the top: welcome.blade.php, template\_landingpage.blade.php (which is the active tab), web.php, landingpage.blade.php, index.blade.php, TrainingController.php, and home.php. On the left, there's a sidebar with project navigation: sass, views (auth, layouts, trainings, routes), storage, tests, vendor, .editorconfig, .env, .gitattributes, .gitignore, .styleci.yml, artisan, composer.json, and composer.lock.

The main content area displays the following code:

```
77 </header>
78
79 <!-- Main content - display welcome and search
80 training here -->
81 @yield('public_content')
82
83 <! end main content -->
84
85 <!-- Footer -->
86 <footer class="footer bg-light">
87   <div class="container">
88     <div class="row">
```

The line `@yield('public_content')` is highlighted with a red rounded rectangle. The word `search` in the line above it is also highlighted with a light blue rounded rectangle.

# Your ICT Training partner

[Search](#)

id	title	trainer	
1	Laravel Basics	Khirulnizam Abd Rahman	<a href="#">Show</a>
2	Laravel Intermediate	Tarmizi Sanusi	<a href="#">Show</a>
3	Laravel Advanced with Unit Test	Nasrul Hazim	<a href="#">Show</a>

@yield( )

[About](#) · [Contact](#) · [Terms of Use](#) · [Privacy Policy](#)

© FSTM KUIS 2020. All Rights Reserved.



# Output overview

Search your prefered training...

Search

id	title	trainer	
1	Laravel Basics	Khirulnizam Abd Rahman	Show
2	Laravel Intermediate	Tarmizi Sanusi	Show
3	Laravel Advanced with Unit Test	Nasrul Hazim	Show

# landingpage.blade.php

C:\laragon\www\lara7\resources\views\landingpage.blade.php • (lara7) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

views

- auth
- layouts
  - app.blade.php
  - mobitemp.blade.php
  - template\_landingpage.blade.php
- trainings
  - create.blade.php
  - index.blade.php
  - Dapatkan percuma 20+ playlist tut...
  - home.blade.php
  - landingpage.blade.php
  - welcome.blade.php
- routes
  - api.php
  - channels.php
  - console.php
  - web.php
- storage
- tests
- vendor
  - .editorconfig
  - .env
  - .gitattributes
  - .gitignore
  - /\* .styleci.yml
  - artisan
  - /\* composer.json
  - composer.lock
  - Laravel Training Youtube Tutorial Set2.pptx
  - Laravel Training Youtube Tutorial.pptx
  - /\* package-lock.json
  - /\* package.json
  - phpunit.xml
  - README.md
  - server.php
  - /\* webpack.mix.js

# route/web.php

```
• | 15 //the landing page route
• | 16 Route::get('/', function () {
• | 17     //welcome /public search
• | 18     $trainings = Training::all();
• | 19     return view('landingpage')->with(compact(
• | 20         'trainings'));
• | 21     });

  routes
    api.php
    channels.php
    console.php
    web.php
  storage
  tests
  vendor
    .editorconfig
    .env
    .gitattributes
    .gitignore
    /* .styleci.yml
  artisan
  /* composer.json
```

# Github.com/khirulnizam/lara7

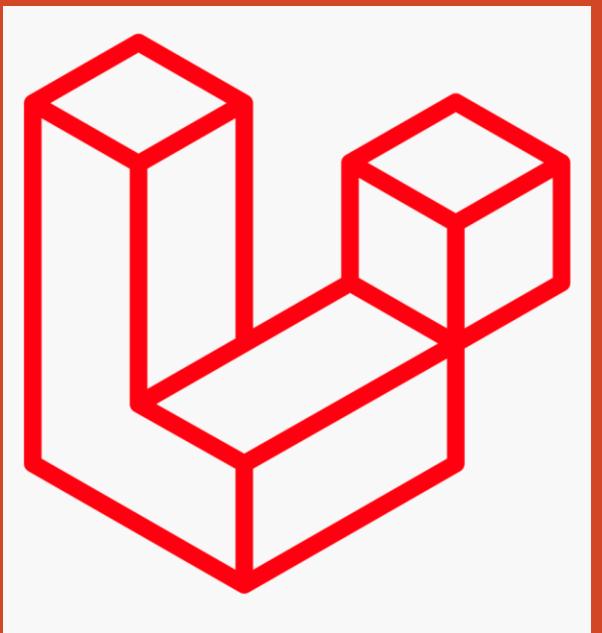
Slides &  
codes

T12 Tutorial  
clone/download  
github repo

The screenshot shows a GitHub repository page for the user 'khirulnizam' with the repository name 'lara7'. The top navigation bar includes links for Home, Create new repository, and Search. The repository summary bar shows 1 commit, 1 branch, 0 packages, 0 releases, and 1 contributor. Below this, there are buttons for Branch: master (with a dropdown arrow), New pull request, Find file, and Clone or download (in green). The main content area displays a list of files and their commit history:

File	Commit Message	Date
app	Initial commit	16 minutes ago
bootstrap	Initial commit	16 minutes ago
config	Initial commit	16 minutes ago
database	Initial commit	16 minutes ago
public	Initial commit	16 minutes ago
resources	Initial commit	16 minutes ago
routes	Initial commit	16 minutes ago

To the right of the file list, there is a 'Clone with HTTPS' section containing the URL <https://github.com/khirulnizam/lara7.git> and a copy icon. Below this are two buttons: 'Open in Desktop' and 'Download ZIP' (highlighted in blue).



# Laravel t19

## Apply Bootstrap on Admin Dashboard

### (Pasang layout Bootstrap baru pada Panel Admin)

# Output overview

The screenshot shows a web browser window titled "FSTM KUIS - template Sbadmin2". The URL is "Not secure | lara7.test/trainings". The page is titled "FSTM KUIS Admin Panel" and displays a "Training Index". On the left, there is a sidebar with "FSTM ADMIN 2" branding and links for "Dashboard", "Insert new record", and "Listing/Search". The main content area has a search bar with a "Search" button. A table lists three training entries:

id	title	tasks
1	Laravel Basics	
2	Laravel Intermediate	
3	Laravel Advanced with Unit Test	

At the bottom, it says "Copyright © FSTM KUIS 2020".

# DISCLAIMER

Codes are for demo purposes. Examples showed here are only to validate the proof of concepts. Security aspects the demonstrated examples are beyond the coverage of these video tutorials. You are advised to explore / search after advanced/security topics of PHP web programming as these tutorials only for beginner level.

Suitable to view in DESKTOP

# Github.com/khirulnizam/lara7

Slides &  
codes

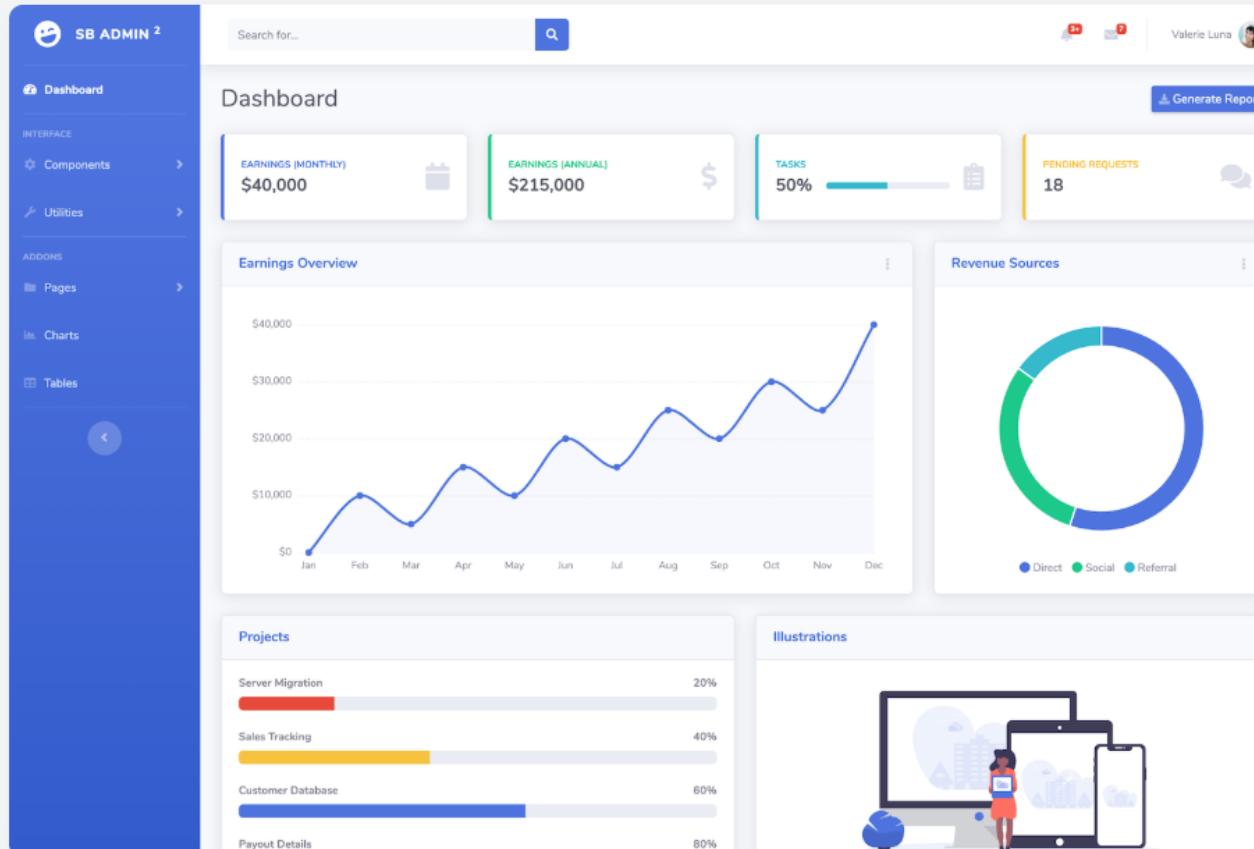
T12 Tutorial  
clone/download  
github repo

The screenshot shows a GitHub repository page for the user 'khirulnizam' with the repository name 'lara7'. The top navigation bar includes links for Home, Create new, and Search. The repository summary bar shows 1 commit, 1 branch, 0 packages, 0 releases, and 1 contributor. Below this, there are buttons for Branch: master (with a dropdown arrow), New pull request, Find file, and Clone or download (in green). The main content area displays a list of files and their commit history:

File	Commit Message	Date
app	Initial commit	16 minutes ago
bootstrap	Initial commit	16 minutes ago
config	Initial commit	16 minutes ago
database	Initial commit	16 minutes ago
public	Initial commit	16 minutes ago
resources	Initial commit	16 minutes ago
routes	Initial commit	16 minutes ago

To the right of the file list, there is a 'Clone with HTTPS' section containing the URL <https://github.com/khirulnizam/lara7.git> and a copy icon. Below this are two buttons: 'Open in Desktop' and 'Download ZIP' (highlighted in blue).

# Theme from <https://startbootstrap.com/themes/sb-admin-2/>



The dashboard features a left sidebar with navigation links for Dashboard, Components, Utilities, Pages, Charts, and Tables. The main area includes a search bar, user profile, and a 'Generate Report' button. Key metrics displayed are Earnings Monthly (\$40,000), Earnings Annual (\$215,000), Tasks (50%), and Pending Requests (18). It also includes a line chart for Earnings Overview and a donut chart for Revenue Sources (Direct, Social, Referral).

### Description

SB Admin 2 is a free, open source, Bootstrap 4 based admin theme perfect for quickly creating dashboards and web applications. It's modern design style with subtle shadows and a card-

 [Free Download](#)

 [Live Preview](#)

 [View on GitHub](#)

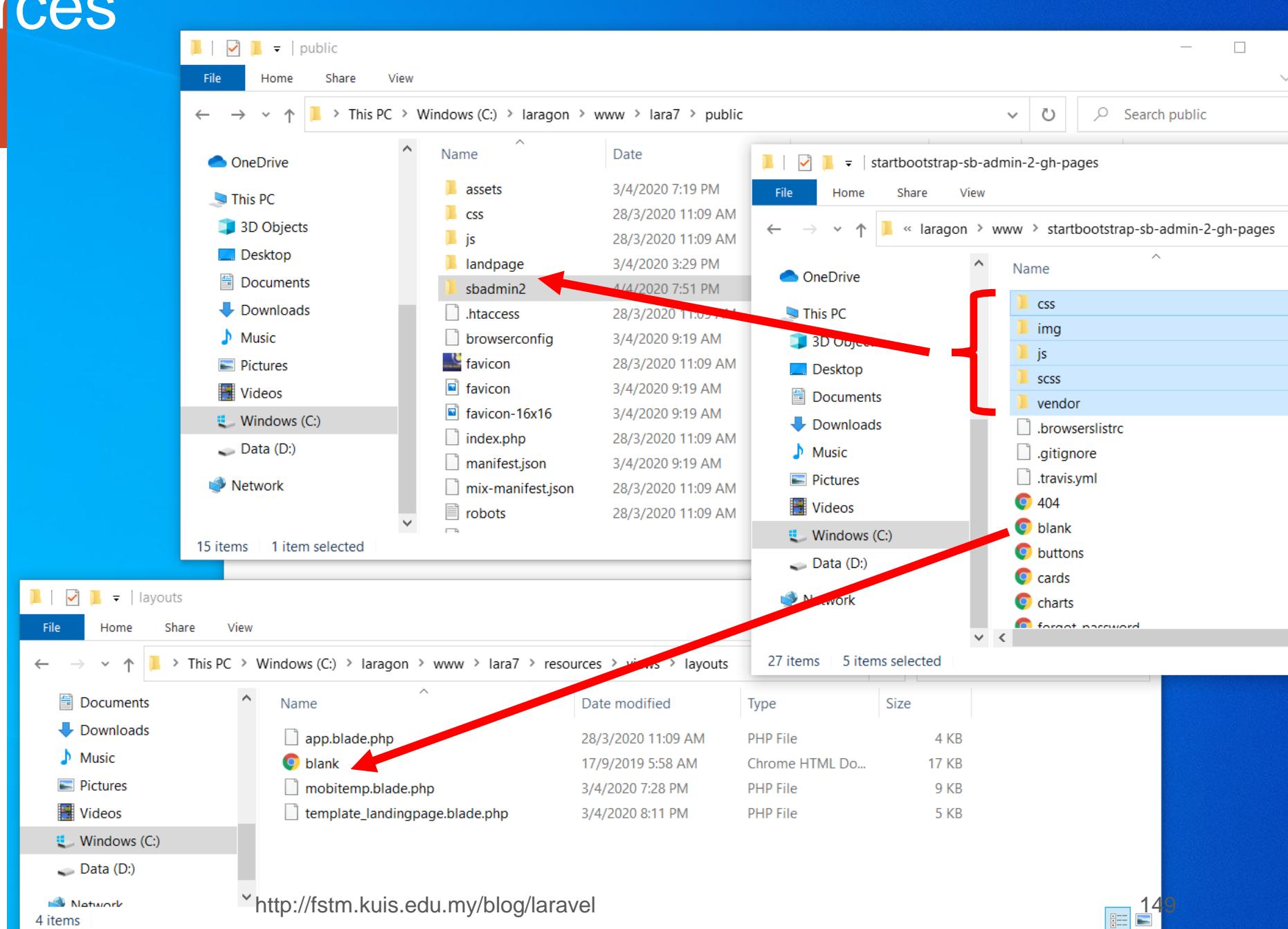
Get updates when SB Admin 2 is improved!

 [Follow @SBootstrap](#)

<b>License</b>	 <a href="#">MIT License</a>
<b>Released</b>	Jan 14, 2014
<b>Bootstrap Version</b>	4.3.1
<b>Last Updated</b>	Sep 16, 2019
<b>Theme Version</b>	4.0.7

[View on GitHub](#)

# Asset/resources for template



# First time display the new template

[SB Admin<sup>2</sup>](#)

- 
- [Dashboard](#)
- 

Interface

- [Components](#)

Custom Components:

[Buttons](#) [Cards](#)

- [Utilities](#)

Custom Utilities:

[Colors](#) [Borders](#) [Animations](#) [Other](#)

---

Addons

- [Pages](#)

Login Screens:

[Login](#) [Register](#) [Forgot Password](#)

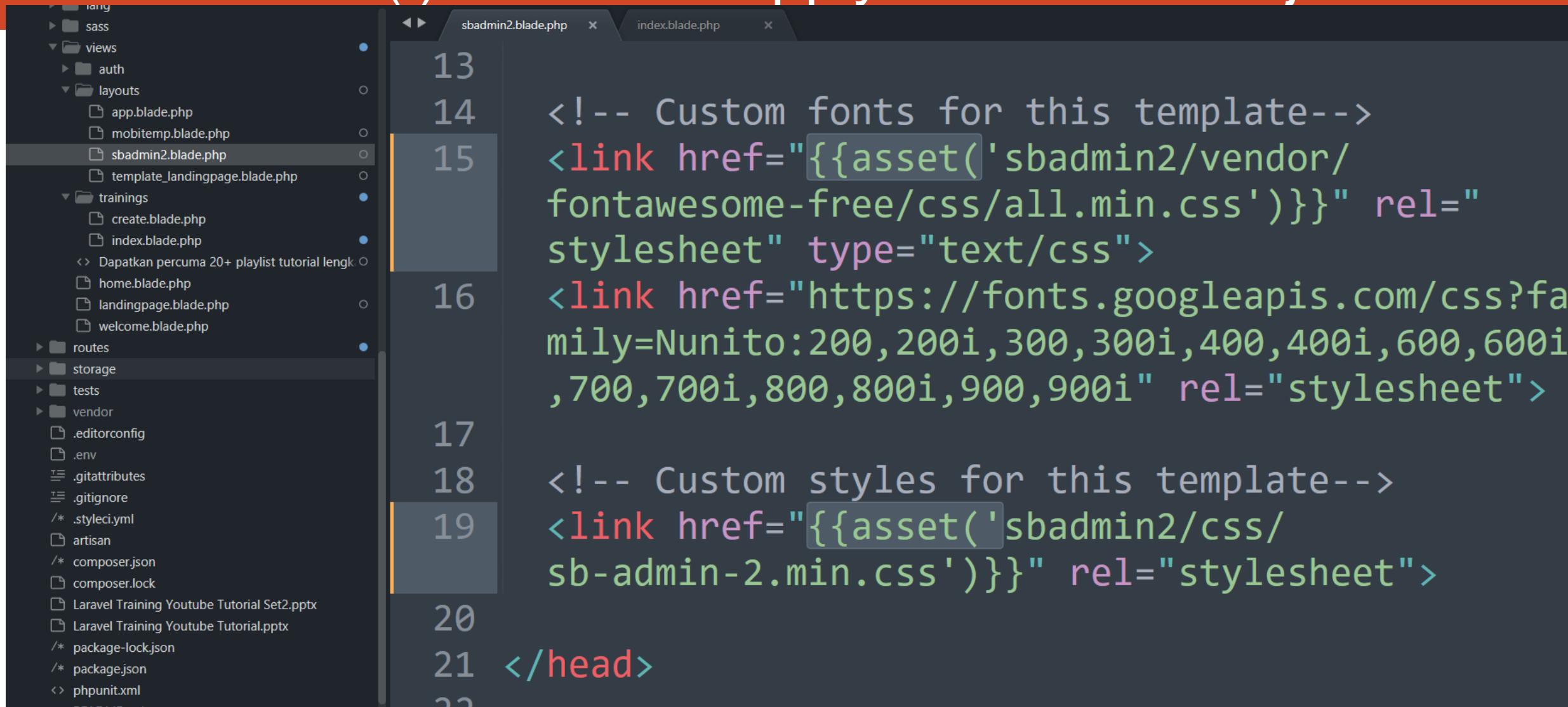
Other Pages:

[404 Page](#) [Blank Page](#)

- [Charts](#)

[Email](#)

# Blade asset( ) function – apply on both css & js

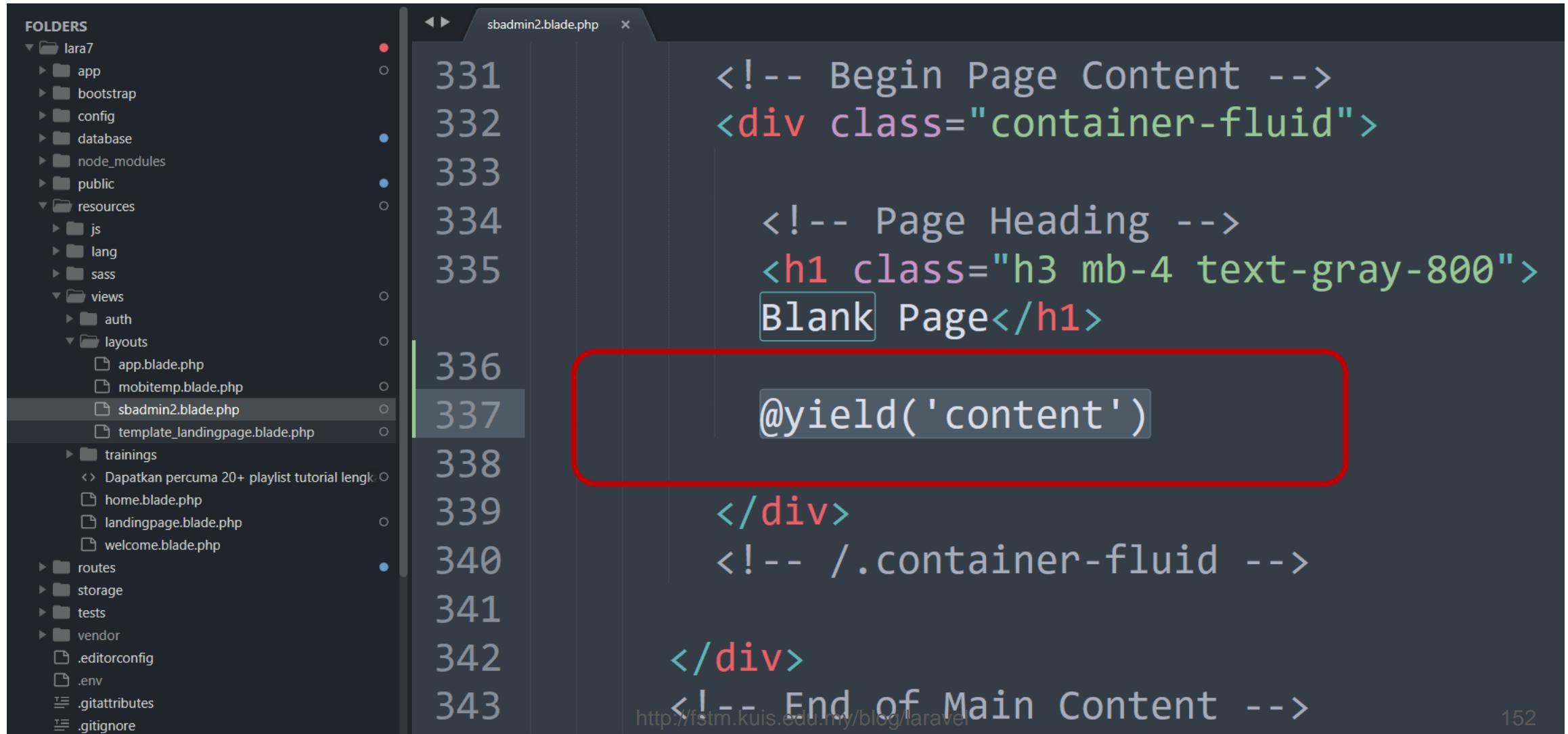


The image shows a code editor interface with a sidebar containing a file tree and two tabs at the top: 'sbadmin2.blade.php' and 'index.blade.php'. The 'index.blade.php' tab is active, displaying the following code:

```
13 <!-- Custom fonts for this template-->
14 <link href="{{asset('sbadmin2/vendor/
15 fontawesome-free/css/all.min.css')}}" rel="stylesheet" type="text/css">
16 <link href="https://fonts.googleapis.com/css?family=Nunito:200,200i,300,300i,400,400i,600,600i
17 ,700,700i,800,800i,900,900i" rel="stylesheet">
18 <!-- Custom styles for this template-->
19 <link href="{{asset('sbadmin2/css/
20 sb-admin-2.min.css')}}" rel="stylesheet">
21 </head>
22
```

The code uses the Blade {{asset()}} function to include CSS files. Lines 15 and 19 are highlighted with a light gray background.

# @yield( )



The image shows a code editor interface with a dark theme. On the left is a sidebar titled "FOLDERS" containing a file tree of a Laravel project. The main area shows a file named "sbadmin2.blade.php". The code in the file is:

```
<!-- Begin Page Content -->
<div class="container-fluid">

    <!-- Page Heading -->
    <h1 class="h3 mb-4 text-gray-800">
        Blank Page</h1>

    @yield('content')

</div>
<!-- /.container-fluid -->

</div>
<!-- End of Main Content -->
```

A red rectangular box highlights the line of code: `@yield('content')`. The URL <http://fstm.kuis.edu.my/blog/laravel> is visible at the bottom of the editor.

# @yield( )

The screenshot shows a web browser window titled "FSTM KUIS - template Sbadmin2" displaying the "Training Index" page of the "FSTM KUIS Admin Panel". The page features a sidebar on the left with options like "Dashboard", "Insert new record", and "Listing/Search". The main content area has a heading "FSTM KUIS Admin Panel" and a large red placeholder text "@yield( )". A red box highlights this placeholder, and two red arrows point from it towards the right side of the screen, where a modal window titled "Training Index" is displayed. This modal lists three training entries with columns for "id", "title", and "tasks".

id	title	tasks
1	Laravel Basics	
2	Laravel Intermediate	
3	Laravel Advanced with Unit Test	

http://fstm.kuis.edu.my/blog/laravel

153

# Output overview

The screenshot shows a web browser window titled "FSTM KUIS - template Sbadmin2" with the URL "lara7.test/trainings". The page is titled "FSTM KUIS Admin Panel" and displays a "Training Index". On the left, there is a sidebar with icons for Dashboard, Insert new record, and Listing/Search. The main content area has a header "Training Index" with a search bar and a "Search" button. Below is a table with three rows:

id	title	tasks
1	Laravel Basics	
2	Laravel Intermediate	
3	Laravel Advanced with Unit Test	

At the bottom right of the page, it says "Copyright © FSTM KUIS 2020".

# index.blade.php

C:\laragon\www\lara7\resources\views\trainings\index.blade.php (lara7) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

The screenshot shows the Sublime Text interface with the file 'index.blade.php' open. The left sidebar displays the project structure:

- auth
- layouts
  - app.blade.php
  - mobitemp.blade.php
  - sbadmin2.blade.php
  - template\_landingpage.blade.php
- trainings
  - create.blade.php
  - index.blade.php
  - home.blade.php
  - landingpage.blade.php
  - welcome.blade.php
- routes
- storage
- tests
- vendor
  - .editorconfig
  - .env
  - .gitattributes
  - .gitignore
  - .styleci.yml
  - artisan
  - composer.json
  - composer.lock
  - Laravel Training Youtube Tutorial Set2.pptx
  - Laravel Training Youtube Tutorial.pptx
  - package-lock.json
  - package.json
  - phpunit.xml
  - README.md
  - server.php
  - webpack.mix.js

# Github.com/khirulnizam/lara7

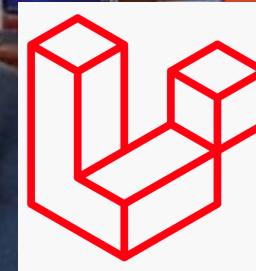
Slides &  
codes

T12 Tutorial  
clone/download  
github repo

The screenshot shows a GitHub repository page for the user 'khirulnizam' with the repository name 'lara7'. The top navigation bar includes links for Home, Create new repository, and Search. The repository summary bar shows 1 commit, 1 branch, 0 packages, 0 releases, and 1 contributor. Below this, there are buttons for Branch: master (with a dropdown arrow), New pull request, Find file, and Clone or download (in green). The main content area displays a list of files and their commit history:

File	Commit Message	Date
app	Initial commit	16 minutes ago
bootstrap	Initial commit	16 minutes ago
config	Initial commit	16 minutes ago
database	Initial commit	16 minutes ago
public	Initial commit	16 minutes ago
resources	Initial commit	16 minutes ago
routes	Initial commit	16 minutes ago

To the right of the file list, there is a 'Clone with HTTPS' section containing the URL <https://github.com/khirulnizam/lara7.git> and a copy icon. Below this are two buttons: 'Open in Desktop' and 'Download ZIP' (highlighted in blue).



**Laravel t20**  
**Muat naik fail gambar**

# Upload image

FSTM KUIS - template Sbadmin2 Not secure | lara7.test/trainings/create

**Insert New Record**

Title: Flutter Basics to Intermediate

Description: Develop Multiplatform Apps using Flutter & Android Studio

Trainer: EnKerul

Poster image filename: 2020-04-07-Flutter Basics to Intermediate.png

Upload training poster (png, jpg, img file only with max 2MB filesize)

Choose File: bannerBMI.png

Save new Record

Copyright © FSTM KUIS 2020

**Upload image**

FSTM KUIS - template Sbadmin2 Not secure | lara7.test/trainings/10

**FSTM ADMIN 2**

Dashboard

Insert new record

Listing/Search

Show all data Record

Title: Flutter Basics to Intermediate

Description: Develop Multiplatform Apps using Flutter & Android Studio

Trainer: EnKerul

Poster image filename: 2020-04-07-Flutter Basics to Intermediate.png



**Show uploaded image**

Back

# DISCLAIMER

Codes are for demo purposes. Examples showed here are only to validate the proof of concepts. Security aspects the demonstrated examples are beyond the coverage of these video tutorials. You are advised to explore / search after advanced/security topics of PHP web programming as these tutorials only for beginner level.

Suitable to view in DESKTOP

# Github.com/khirulnizam/lara7

Slides &  
codes

T12 Tutorial  
clone/download  
github repo

The screenshot shows a GitHub repository page for the user 'khirulnizam' with the repository name 'lara7'. The top navigation bar includes links for Home, Create new repository, and Search. The repository summary bar shows 1 commit, 1 branch, 0 packages, 0 releases, and 1 contributor. Below this, there are buttons for Branch: master (with a dropdown arrow), New pull request, Find file, and Clone or download (in green). The main content area displays a list of files and their commit history:

File	Commit Message	Date
Initial commit	khirulnizam Initial commit	16 minutes ago
app	Initial commit	16 minutes ago
bootstrap	Initial commit	16 minutes ago
config	Initial commit	16 minutes ago
database	Initial commit	16 minutes ago
public	Initial commit	16 minutes ago
resources	Initial commit	16 minutes ago
routes	Initial commit	16 minutes ago

On the right side, there is a 'Clone with HTTPS' section with a URL field containing <https://github.com/khirulnizam/lara7.git> and a copy icon. Below it are 'Open in Desktop' and 'Download ZIP' buttons, both in blue. The entire screenshot is framed by a blue border.

# Add column to table trainings

Laragon\lara7test\trainings - HeidiSQL Portable 10.2.0.5599

File Edit Search Tools Go to Help

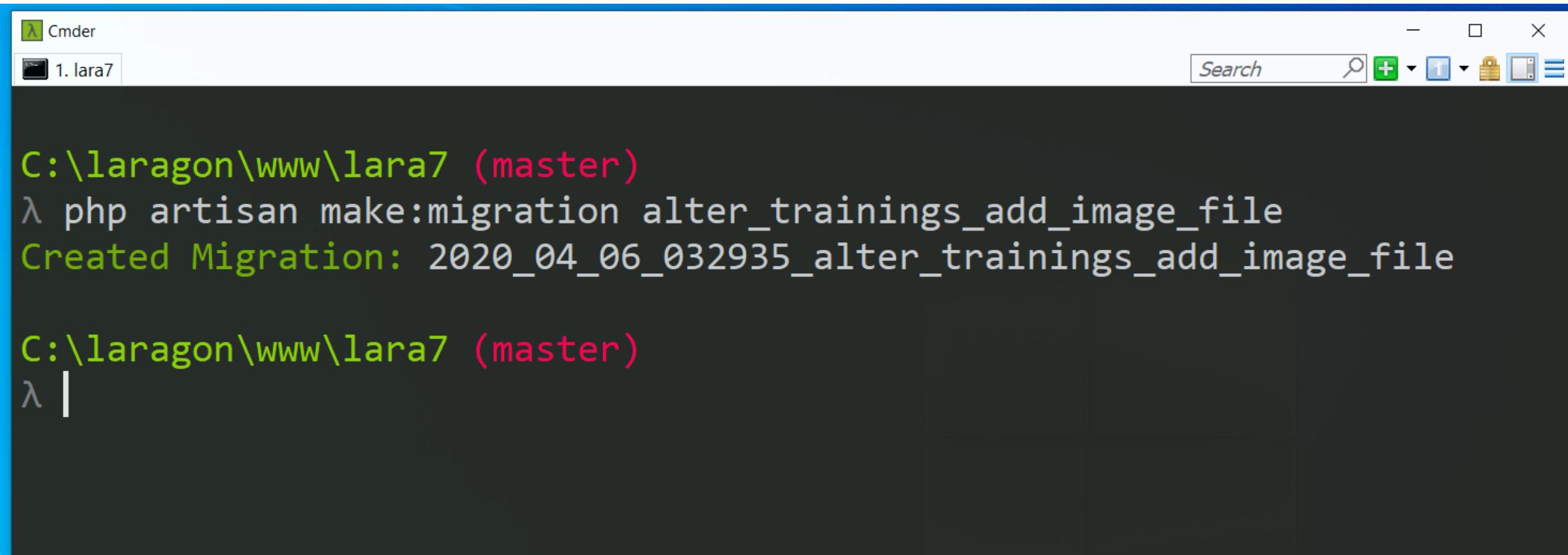
Database filter Table filter Host: localhost Database: lara7test Table: trainings Data Query

	id	title	description	trainer	filename
1	1	Laravel Basics	2 days Laravel course	Khirulnizam Abd ...	(NULL)
2	2	Laravel Intermediate	2 days Laravel course...	Tarmizi Sanusi	(NULL)
3	3	Laravel Advanced wit...	2 days Laravel course...	Nasrul Hazim	(NULL)
4	4	Laravel API 2	Training to generate ...	Kerul Rahman2	2020-04-06-Laravel API 2...
5	5	Open Broadcasting S...	Open Broadcasting S...	En Kerul	2020-04-06-Open Broadca...
6	6	Test 2 WO Pic	Upload record WO Pic	Mr WHO	(NULL)
8	8	Flutter 101	Flutter to develop mu...	Mr Mano	2020-04-06-Flutter 101.jpeg
9	9	Ionic 101	Basic Ionic training to...	enKerul	2020-04-06-Ionic 101.png



Field to hold image's filename

# Additional field to table trainings



A screenshot of a terminal window titled "Cmder". The window has a dark theme with light-colored text. It shows the command line interface for Laravel migrations. The user is in the directory "C:\laragon\www\lara7" and runs the command "php artisan make:migration alter\_trainings\_add\_image\_file". The response indicates that a migration named "2020\_04\_06\_032935\_alter\_trainings\_add\_image\_file" was created. The terminal prompt ends with a cursor at the start of a new line.

```
C:\laragon\www\lara7 (master)
λ php artisan make:migration alter_trainings_add_image_file
Created Migration: 2020_04_06_032935_alter_trainings_add_image_file

C:\laragon\www\lara7 (master)
λ |
```

# function up() – additional migration file

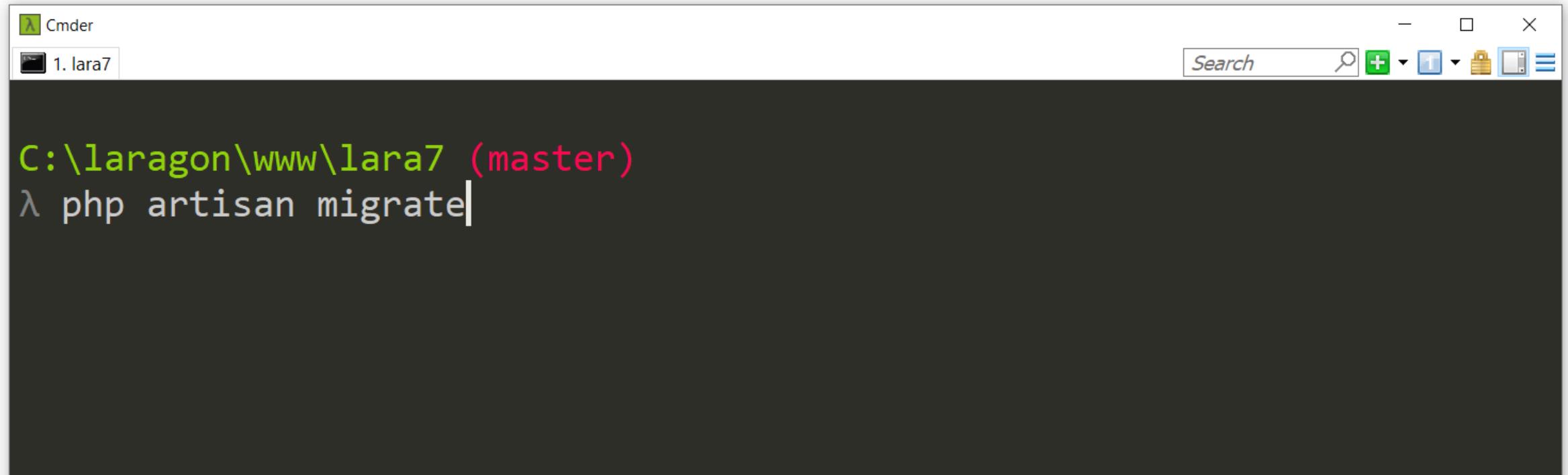
The screenshot shows a Sublime Text editor window with the following details:

- File Path:** C:\laragon\www\lara7\database\migrations\2020\_04\_06\_032935\_alter\_trainings\_add\_image\_file.php
- Editor Title:** 2020\_04\_06\_032935\_alter\_trainings\_add\_image\_file.php
- Folders:** A sidebar titled "FOLDERS" lists project directories: lara7, app, bootstrap, config, database, factories, migrations, seeds, .gitignore, node\_modules, public, resources, routes, storage, tests, vendor, .editorconfig, and env.
- Code Content:** The migration file contains PHP code for a "Schema::table('trainings')".

```
public function up()
{
    //add fields to hold image file
    Schema::table('trainings',
        function (Blueprint $table) {
            $table->string('filename')->nullable();
            $table->string('mime')->nullable();
            $table->string('original_fname')->nullable();

            //for next training table relationship
            $table->integer('idadmin')->nullable();
        });
}
```

# php artisan migrate



A screenshot of a Cmder terminal window. The title bar says "Cmder". The tab bar shows "1. lara7". The search bar has "Search" and other icons. The main area shows the command line:

```
C:\laragon\www\lara7 (master)
λ php artisan migrate|
```

# create.blade.php – to select and upload image

Insert New Record

Title  
Ionic 101

Description  
Basic Ionic training to develop Hybrid apps

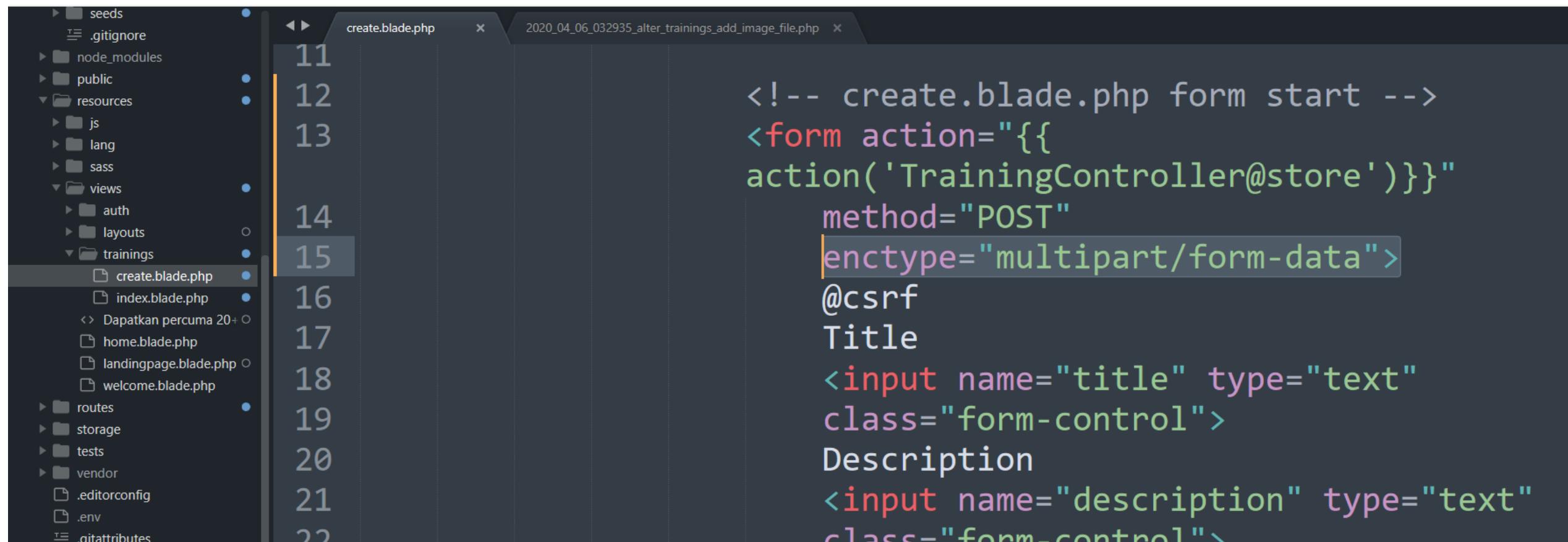
Trainer  
enKerul

Upload training poster (png, jpg, img file only with max 2MB filesize)

bannerFlutter.png

 **Upload image**

# Add enctype in create.blade.php



```
<!-- create.blade.php form start -->
<form action="{!!
action('TrainingController@store')}">
    method="POST"
    |enctype="multipart/form-data">
        @csrf
        Title
        <input name="title" type="text"
            class="form-control">
        Description
        <input name="description" type="text"
            class="form-control">
```

# Add input type="file" in create.blade.php

```
► config  
► database  
► node_modules  
► public  
▼ resources  
  ► js  
  ► lang  
  ► sass  
  ▼ views  
    ► auth  
    ► layouts  
    ▼ trainings  
      □ create.blade.php  
      □ index.blade.php  
      <> Dapatkan percuma  
      □ home.blade.php  
      □ landingpage.blade.php  
      □ welcome.blade.php  
  ► routes  
  ► storage  
  ► tests  
  ► vendor  
    □ .editorconfig  
    □ .env  
    ▄ .gitattributes  
    ▄ .gitignore  
    /* .styleci.yml  
    □ artisan  
    /* composer.json
```

```
23 Trainer  
24   <input name="trainer" type="text"  
25     class="form-control">  
26  
27 <!-- create.blade.php add file input -->  
28 Upload training poster  
29   <input type="file"  
30     name="attachment">  
31   <button type="submit" class="btn  
32     btn-primary">  
33     Submit Record</button>  
34   </form>  
35 <!-- form ends -->  
36   </div>  
37 </div>
```

# app/Training.php - Add fillable for the fields

The screenshot shows a code editor interface with a sidebar labeled "FOLDERS" on the left. The "app" folder is expanded, showing subfolders like "Console", "Exceptions", "Http", "Controllers", "Middleware", "Providers", and "User". Inside the "Controllers" folder, a file named "Training.php" is selected and highlighted with a blue background.

The main editor area displays the PHP code for the "Training" model:

```
1 <?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class Training extends Model
8 {
9     //Training Model at app/Training.php
10    //add the new fields
11    protected $fillable = [
12        'title', 'description', 'trainer',
13        'filename','idadmin'//add the new fields
14    ];
15
16    protected $table='trainings';
17 }
18
```

The line number 10 is highlighted with a yellow background, indicating where the new code was added.

# Validate uploaded file type

### Insert New Record

• Only jpeg, png, bmp image file are allowed. Do make sure file size is not larger than 2MB

Title

Description

Trainer

Upload training poster (png, jpg, img file only with max 2MB filesize)  
 Choose File No file chosen

# TrainingController@store – validate file type

The screenshot shows a code editor interface with a dark theme. On the left, there's a sidebar titled 'FOLDERS' listing various Laravel project components like 'lara7', 'app', 'Http', 'Providers', etc. The main area displays a PHP file named 'TrainingController.php'. The code is as follows:

```
57 //validate uploaded file type
58 //TrainingCOnroller@store
59 if ($request->hasFile('attachment')){
60     $this->validate($request, [
61         'attachment' => 'mimes:jpeg,png,bmp|max:2048',
62     ],
63         //validateion error messages
64         $errors = [
65             'required' => 'The :attribute field is
66                         required.',
67             'mimes' => 'Only jpeg, png, bmp image file
68                         are allowed.
69                         Do make sure file size is not larger
70                         than 2MB'
71         ]);
72     );
73 };
```

The code implements a validation rule for the 'attachment' file upload. It checks if the file exists, then uses the `$this->validate` method to define validation rules and error messages. The rules specify that the file must be a jpeg, png, or bmp image and its size must not exceed 2MB.

# TrainingController@store – continued...

The screenshot shows a code editor interface with a dark theme. On the left, there's a sidebar titled "FOLDERS" listing various Laravel project files and directories. The main area displays a PHP file named "TrainingController.php". The code is a continuation of a method, specifically the "store" action. It includes logic for generating a filename from the current date and the request title, reading an attached file using Storage and File classes, setting the training's filename, saving the training record, and finally redirecting to the index page with a success message.

```
//example
//filename=2020-04-06-LaravelTraining101.png
$filename=date('Y-m-d').'-' . $request->get('title').
'.' . $request->attachment->
    getClientOriginalExtension();
//save attaced file to storage
Storage::disk('public')->put($filename,
    File::get($request->attachment));

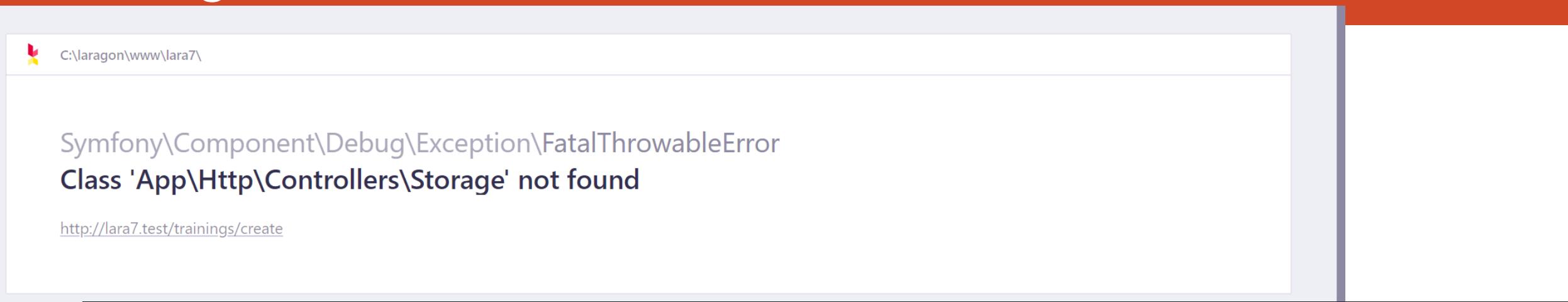
$training->filename=$filename;
}

$training->save();

//redirect to index
return redirect('/trainings')
    ->with('success','New record saved');

}//end TrainingController@store
```

# Storage not found



The screenshot shows a code editor with the following details:

- File path: TrainingController.php
- Code content:

```
1 <?php
2
3 namespace App\Http\Controllers;
4 use Illuminate\Http\Request;
5 use App\Training;
6 //add this to use Storage::disk() function
7 use Illuminate\Support\Facades\Storage;
8 use Illuminate\Support\Facades\File;
```
- File tree on the left:
  - FOLDERS:
    - lara7
    - app
      - Console
      - Exceptions
      - Http
        - Controllers
  - File list:
    - C.php
    - Controller.php
    - HomeController.php
    - NewController.php
    - TrainingController.php

# Configuring Filesystem for Storage config/filesystems.php

The screenshot shows a code editor with multiple tabs at the top: Kernel.php, filesystems.php (which is the active tab), Training.php, create.blade.php, TrainingController.php, User.php, and 2020\_04\_06\_03. The left sidebar displays the project structure, including providers, bootstrap, config (with app.php, auth.php, broadcasting.php, cache.php, database.php, and filesystems.php selected), hashing.php, logging.php, mail.php, queue.php, services.php, session.php, view.php, database, node\_modules, public, resources (with js, lang, sass, views, and auth selected), and auth.

```
//file location config/filesystems.php

44
45
46 'disks' => [
47
48     'local' => [
49         'driver' => 'local',
50         'root' => storage_path('app'),
51     ],
52
53     'public' => [
54         'driver' => 'local',
55         'root' => storage_path('app/public'),
56         'url' => env('APP_URL').'/storage',
57         'visibility' => 'public',
58     ],
59 ];
```

# File Storage

## Create Symbolic Link

The public disk is intended for files that are going to be publicly accessible.

By default, the public disk uses the local driver and stores these files in `storage/app/public`.

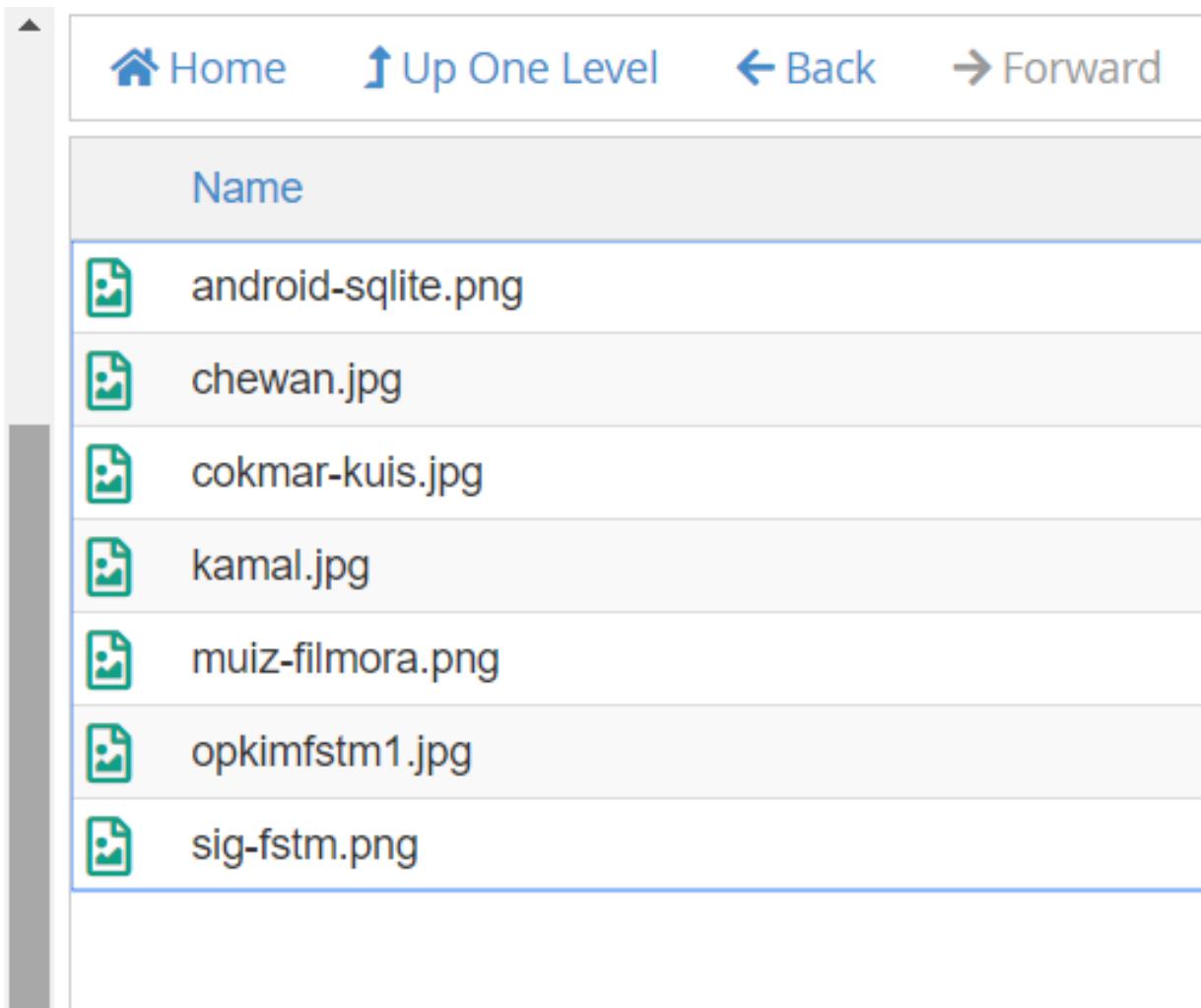
To make them accessible from the web, you should create a symbolic link from `public/storage` to `storage/app/public`.

To create the symbolic link, you may use the `storage:link` Artisan command:

```
php artisan storage:link
```

# Virtual link of public file storage

```
logs  
+ mail  
+ public_ftp  
- public_html  
  - larafstm  
    + css  
    fonts  
    + images  
    + img  
    + js  
    + sadmin  
    + storage  
    + vendors  
+ ionicpwa  
+ js
```



# Physical link of public file storage

The screenshot shows a file explorer interface. On the left, the directory structure of a Laravel application is displayed:

- (/home/---)
- cache
- etc
- larafstm
  - + app
  - + bootstrap
    - config
  - + database
  - + public
  - + resources
    - routes
  - storage
    - app
      - public
        - uploads
  - + framework
  - logs
  - + tests
  - logs
  - + mail
  - + public\_ftp
  - public\_html
  - larafstm
    - + css
    - fonts
    - + images
    - + img

android-sqlite.png
chewan.jpg
cokmar-kuis.jpg
kamal.jpg
kursus-it-opkim.fstm.png
muiz-filmora.png
opkimfstm1.jpg
sig-fstm.png

## The actual physical location of the file storage

To make them accessible from the web, you should create a symbolic link from **public/storage** to **storage/app/public**.

# create.blade.php - Display validation errors

```
create.blade.php x TrainingController.php x filesystems.php x Training.php x show.blade.php x register.blade.php x login.blade.php x sadmin2.blade.php x .env
7 <div class="card">
8   <div class="card-header"><h3>Insert New Record</h3></div>
9
10  <div class="card-body">
11    <!-- show validation error messages -->
12    @if ($errors->any())
13      <div class="alert alert-danger">
14        <ul>
15          @foreach ($errors->all() as $error)
16            <li>{{ $error }}</li>
17          @endforeach
18        </ul>
19      </div>
20    @endif
21
```

# Github.com/khirulnizam/lara7

Slides &  
codes

T12 Tutorial  
clone/download  
github repo

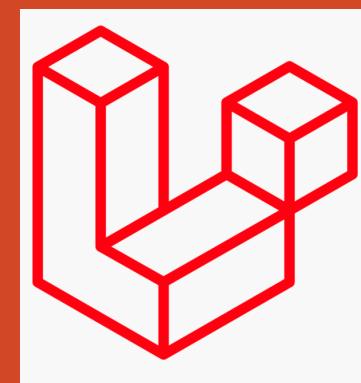
The screenshot shows a GitHub repository page for 'lara7'. At the top, there's a header with navigation icons, a search bar, and user information. Below the header, the repository summary shows: 1 commit, 1 branch, 0 packages, 0 releases, and 1 contributor (khirulnizam). There are buttons for 'Branch: master' (with a dropdown arrow), 'New pull request', 'Find file', and 'Clone or download' (in green). The main content area displays a list of files and their commit history:

File	Commit Message	Date
app	Initial commit	16 minutes ago
bootstrap	Initial commit	16 minutes ago
config	Initial commit	16 minutes ago
database	Initial commit	16 minutes ago
public	Initial commit	16 minutes ago
resources	Initial commit	16 minutes ago
routes	Initial commit	16 minutes ago

To the right of the file list, there's a 'Clone with HTTPS' section with a URL (https://github.com/khirulnizam/lara7.git) and a clipboard icon. Below that are 'Open in Desktop' and 'Download ZIP' buttons, both in blue. A vertical scrollbar is visible on the right side of the page.

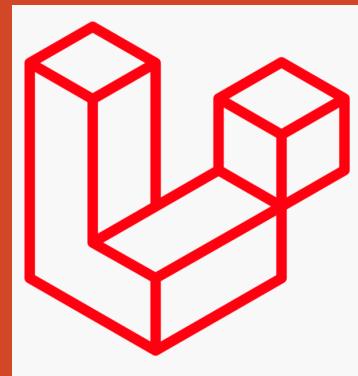
# Laravel t21

## Table relationship (Jadual dengan perhubungan 1-Many)



# Laravel t22

Insert record with drop  
down data option  
**(Masukkan data baharu  
dengan jadual berhubungan)**



# Laravel t23

## Upload record with file (Muat naik sebarang fail)

