



# آموزش LARAVEL

رضا آواره

# مباحث

Laravel •

# Laravel

- اصول لاراوول بر مبنای php و شی گزایی هست
- Mvc
- پوشه بندی با معماری درست

**LARAVEL**

# framework

- Library چیست
- کدی که کسی نوشته در کد خودم استفاده میکنم
- فریم ورک چیست ؟
- چند کتابخانه کنار هم با معماری درست
- چرا بهش نیاز داریم ؟
- سریعتر و راحت تر



LARAVEL

# Why laravel

- سرعت بالا
- ایدیت های سریع
- جامعه های زیاد

**LARAVEL**

# Start...

• نصب xampp با مهندسی next!!!

## LARAVEL

# بررسی نصب php

• Cmd

• `Php -v`

- اگر خطا داد پوشه php داخل xamp
- راست کلیک روی mycomputer بعد properties بعد advanced system
- setting بعد environment variables
- Path ادرس را اضافه میکنیم



LARAVEL

# composer

- Getcomposer.org
- پیرای packagist.org
- مهندسی next!!

• در path اسلاید قبل باید اضافه شود!!!



LARAVEL



# نصب لاراول

`composer create-project laravel/laravel example-app` •

**LARAVEL**

`composer global require laravel/installer` •  
`laravel new example-app` •

# artisan

- دستیار لاراوول
- فایل Artisan داخل فولدر اصلی

- Php artisan
- لیست دستورات

- Php artisan down
- در حال توسعه

- Php artisan up
- فعال سازی مجدد

- Php artisan serve
- فعال سازی بدون htdocs...

LARAVEL

# Routing

- سیستم مسیر دهی در کنار mvc
- فایل htaccess در public سپس index.php
- در پوشه route فایل web.php

**LARAVEL**

# Routing

`Route::get("address", [controller::class, 'method']);` •

`Route::get("address", function() { });` •

• /الزامی نیست

**LARAVEL**

# Data in routes

```
Route::get('.../{x}',function($x){return $x;}); •  
Route::get('.../{x}/{y}',function($x,$y){return $x;}); •
```

**LARAVEL**

# Route group

- `Route::prefix('/a')->group(function(){`
- `Route::get('address', [controller::class, 'method']);`
- `Route::get('address', [controller::class, 'method']);`
  
- `});`



LARAVEL

# Views

- Return view('...')
  - (helper function)
- Resources->views->....
  - Blade->موتور قالب ساز
  - .blade.php
  - html کد



LARAVEL

# Pass data to views

- In php
  - `return view('test',['aa'=>"ss"]);`
  - `return view('welcome',compact('t'));`
- In blade
  - `<?=$aa?>`
- Bug xss
  - `return view('test',['aa'=>"<script>alert('aaa')</script>"]);`
- In blade
  - `{{ $t }}`



LARAVEL



• Blade کامپایل میشود و در Storage و views

**LARAVEL**

# دستورات کنترلی در blade

- @if(...)
- @elseif()
- @endif
  
- @empty()
- @else
- @endempty



LARAVEL

# foreach

- @foreach(\$objs as \$obj)
- {{\$obj}}
- @endforeach
  
- @php
- @endphp
  
- <https://laravel.com/docs/9.x/blade#loops>



LARAVEL

# Master layouts

- پوشه layout
- فایل master
- در master قسمتی که تغییر میکند با `@yield('name')`
- ارث بری صفحات دیگر با `@extends('layout.amaster')`
- در صفحات دیگر با `@endsection`.....`@section('name')`
  - `@section('title','aaaa')`
- بخشهای ثابت و مستقل مثل هدر و فوتر و منو و ... در فایل جداگانه و در فایل اصلی `@include()` میکنیم

LARAVEL

# Static files

- اگر فایلی در public باشد بدون توجه به route باز میکند
- برای استایل ها و فایلها و عکسها ... پوشه بندی در public
- `{{asset('css/bootstrap.min.css')}}}`



LARAVEL

# Database

- در فایل `env`. قسمتهای `DB_` را ویرایش میکنیم
- پوشه `Config` فایل `database`

• تغییرات `env` باید `serve` را مجدد اجرا کرد

LARAVEL

# Query

- `$q=DB::table('')->get();`
- `->find(...)` بـ `id`
  
- `->where('مقدار','ستون')`
  
- `->irderBy('id','desc')`
  
- Show=>
- `Dd($q)`
- `$q->id`



LARAVEL

# Query insert

- `DB::table('...')->insert([`
  - `'id'=>'...'`,
- `]);`



LARAVEL



# migration

- ابزاری برای مدیریت جداول

- `Php artisan migrate`

- جداول پوشه `migrate` را ایجاد میکند

- `Php artisan migrate:status`

- چه جداولی ایجاد شده



LARAVEL



# LARAVEL

• ساخت migrate

• `Php artisan make:migrate create_users_table`

• ساختار class

• method up

• برای اجرا

• Method down

• حذف

- `$table->type('name');`
- `String('name',length)`

• `Php artisan migrate:rollback`

• آخرین تغییرات برمیگردد

• `Php artisan migrate:reset`

• برمیگردد حالت اول

• `Php artisan migrate:refresh`

• اول ریست بعد ساختن



LARAVEL

# Add column

• `Php artisan make:migration add_new_field_to_articles_table --table=articles`

- در `up` ستون را مینویسیم
- در `down`
- `$table->dropColumn("")`

LARAVEL

# Change column

• `Php artisan make:migration change_column_from_articles_table --table=articles`

• Up

- تغییر سایز
- `change()` اخراج

• `renameColumn('a', 'b')` تغییر نام

• Down

• قبلیا

• `Composer require` پکیج

LARAVEL

# Seeder

• اطلاعات fake و اولیه

• `Php artisan make:seeder UserTableSeeder`

**LARAVEL**

- `Foreach(range(1,10) as $item){`
- `DB::table('users')->insert([`
  - `'name'=>'name$item',`
- `])`
- `}`
- `Php artisan db:seed --class=UserTableSeeder`
- ها...Faker

# Eloquent

• برای MVC

• ORM

• Object relational mode

• `Php artisan make:model Article`

• `=Article::where()->orderBy()->get()`

**LARAVEL**

# Form

• اگر از post استفاده شود CSRF جهت امنیت

@CSRF •

• چک inspect

• request('name')

• \$user=new User();

• \$user->name=request('name')

• \$user->save();

LARAVEL



```
User::create([  
    'name'=>request('name'),  
    'family'=>request('family')  
]);
```

در مدل •

```
protected $fillable=['name','family']; •
```

```
return redirect('/users') •
```

LARAVEL

# validation

- `$validator = Validator::make(request()->all() , [`
- `'name'=>'required'`
- `]);`
- `If($validator->fails())`
  - `Return redirect()->back()->with Errors($validator);`
- Blade
- `If($errors->any())`
  - `Foreach($errors->all() as $error)`

LARAVEL

- `$validator = Validator::make(request()->all() , [`
- `'name'=>'required'`
- `])->validate();`



**LARAVEL**

# Custom message

- `$validator = Validator::make(request()->all() , [`
- `'name'=>'required'`
- `], [`
- `'name.required'=>'متن'`
- `)]->validate();`

LARAVEL

- در resource و بخش lang در en پیامهای انگلیسی هست
- میتوانیم فایل Laravel-lang فارسی دانلود کنیم و کپی کنیم
- جهت فعالسازی در config فایل app قسمت locale را fa میزاریم



LARAVEL

# Edit

- `if(!$user)`
  - `Abort(404)`
- `findOrFail()`

• مثل ایجاد کرد یا از `save` یا از `update`



LARAVEL

# Delete

`$user->delete()` •

**LARAVEL**