

1. دانسته های خود از لاراول را بیان کنید و توضیح دهید پوشه های آن چه کاربردی دارد؟

لاراول یک چارچوب توسعه وب بسیار قدرتمند است که با زبان برنامه نویسی PHP توسعه یافته است. این چارچوب به توسعه دهندگان امکانات زیادی را برای ایجاد برنامه های وب کارآمد و با کیفیت ارائه می دهد. در زیر به برخی از ویژگی ها و توانمندی های لاراول به زبان فارسی و به صورت افقی اشاره خواهیم کرد:

۱. معماری MVC: لاراول از معماری مدل-ویو-کنترلر (MVC) استفاده می کند. این معماری به توسعه دهندگان این امکان را می دهد که کد خود را به قسمت های منطق (مدل)، نمایش (ویو) و کنترل (کنترلر) تقسیم کنند که باعث سهولت در مدیریت و توسعه کد می شود.

۲. Eloquent ORM: لاراول از ORM قدرتمند به نام Eloquent برای تعامل با پایگاه داده استفاده می کند. این ابزار به توسعه دهندگان این امکان را می دهد تا با استفاده از کد PHP با پایگاه داده کار کنند بدون این که نیاز به نوشتن کوئری های SQL داشته باشند.

۳. سیستم مسیریابی: لاراول دارای یک سیستم مسیریابی کامل است که به شما این امکان را می دهد تا درخواست ها را به کنترلر ها هدایت کنید و مسیرها را به راحتی مدیریت کنید.

۴. فراخوانی HTTP برای RESTful API: لاراول به شکل بسیار خوبی از فراخوانی های HTTP برای پیاده سازی سرویس های وب و API استفاده می کند و پشتیبانی کامل از RESTful API را فراهم می کند.

۵. کشاورزی: لاراول از ویژگی های کشاورزی برای بهبود عملکرد و سرعت برنامه استفاده می کند.

۶. مهاجرت و توسعه پایگاه داده: با استفاده از مهاجرت ها، توسعه دهندگان می توانند تغییرات در پایگاه داده را مدیریت کرده و ارتقاء یا کاهش نسخه پایگاه داده را به راحتی انجام دهند.

۷. سیستم قدرتمند توسعه یافته جهت تست: لاراول دارای ابزارهای تست مثل PHPUnit است که به توسعه دهندگان این امکان را می دهد تا برنامه های خود را با استفاده از تست های واحد (unit tests) و تست های اجتماعی (feature tests) آزمایش کنند.

۸. فایل های عمومی و ایستا: لاراول از فولدر public جهت ذخیره فایل های عمومی و ایستا (مانند تصاویر و فایل های CSS، جاوااسکریپت (استفاده می کند.

۹. Composer. و مدیریت پکیج ها: لاراول از Composer برای مدیریت پکیج ها و وابستگی ها استفاده می کند که کار توسعه دهندگان را ساده تر می کند.

همچنین، لاراول دارای جامعه ای فعال و حاضر به همکاری است که به توسعه دهندگان امکان پشتیبانی، آموزش، و به اشتراک گذاری منابع را فراهم می کند. این ویژگی ها به لاراول جاذبه زیادی برای توسعه دهندگان وب ایجاد کرده اند.

بله، حتما. لاراول یک چارچوب توسعه وب بسیار قدرتمند است که با زبان برنامه نویسی PHP توسعه یافته است. این چارچوب به توسعه دهندگان امکانات زیادی را برای ایجاد برنامه های وب کارآمد و با کیفیت ارائه می دهد. در زیر به برخی از ویژگی ها و توانمندی های لاراول به زبان فارسی و به صورت افقی اشاره خواهیم کرد:

۱. معماری MVC: لاراول از معماری مدل-ویو-کنترلر (MVC) استفاده می کند. این معماری به توسعه دهندگان این امکان را می دهد که کد خود را به قسمت های منطق (مدل)، نمایش (ویو) و کنترل (کنترلر) تقسیم کنند که باعث سهولت در مدیریت و توسعه کد می شود.

۲. Eloquent ORM: لاراول از ORM قدرتمند به نام Eloquent برای تعامل با پایگاه داده استفاده می کند. این ابزار به توسعه دهندگان این امکان را می دهد تا با استفاده از کد PHP با پایگاه داده کار کنند بدون این که نیاز به نوشتن کوئری های SQL داشته باشند.

۳. سیستم مسیریابی: لاراول دارای یک سیستم مسیریابی کامل است که به شما این امکان را می دهد تا درخواست ها را به کنترلر ها هدایت کنید و مسیر ها را به راحتی مدیریت کنید.

۴. ****فراخوانی HTTP برای RESTful API:**** لاراول به شکل بسیار خوبی از فراخوانی‌های HTTP برای پیاده‌سازی سرویس‌های وب و API استفاده می‌کند و پشتیبانی کامل از RESTful API را فراهم می‌کند.

۶. **مهاجرت و توسعه پایگاه داده با استفاده از مهاجرت‌ها،** توسعه‌دهندگان می‌توانند تغییرات در پایگاه داده را مدیریت کرده و ارتقاء یا کاهش نسخه پایگاه داده را به راحتی انجام دهند.

۷. **سیستم قدرتمند توسعه‌یافته جهت تست لاراول** دارای ابزارهای تست مثل PHPUnit است که به توسعه‌دهندگان این امکان را می‌دهد تا برنامه‌های خود را با استفاده از تست‌های واحد (unit tests) و تست‌های اجتماعی (feature tests) آزمایش کنند.

۸. **فایل‌های عمومی و ایستا لاراول** از فولدر `public` جهت ذخیره فایل‌های عمومی و ایستا (مانند تصاویر و فایل‌های CSS، جاوااسکریپت) استفاده می‌کند.

۹. **Composer و مدیریت پکیج‌ها لاراول** از Composer برای مدیریت پکیج‌ها و وابستگی‌ها استفاده می‌کند که کار توسعه‌دهندگان را ساده‌تر می‌کند.

همچنین، لاراول دارای جامعه‌ای فعال و حاضر به همکاری است که به توسعه‌دهندگان امکان پشتیبانی، آموزش، و به اشتراک گذاری منابع را فراهم می‌کند. این ویژگی‌ها به لاراول جاذبه زیادی برای توسعه‌دهندگان وب ایجاد کرده‌اند.

2. Artisan در لاراول چیست؟

Artisan در لاراول یک ابزار خط فرمان (Command-Line Interface) یا (CLI) است که توسط این چارچوب ارائه شده است و به توسعه‌دهندگان امکان اجرای دستورات مختلف و انجام وظایف مختلف در حین توسعه برنامه را می‌دهد. این ابزار برای انجام کارهای متنوعی از جمله مدیریت پایگاه داده، اجرای تست‌ها، ایجاد مدل‌ها و کنترلرها، و بسیاری از وظایف دیگر استفاده می‌شود.

دستورات Artisan می‌توانند به صورت خودکار ایجاد شده و توسط توسعه‌دهندگان نیز گسترش داده شوند. برخی از دستورات معروف Artisan عبارتند از:

php artisan migrate: این دستور برای ایجاد یا اعمال تغییرات به پایگاه داده بر اساس فایل‌های مهاجرت (migration) استفاده می‌شود.

php artisan make:controller: با استفاده از این دستور می‌توانید یک کنترلر جدید ایجاد کنید.

php artisan make:model: این دستور برای ایجاد یک مدل جدید استفاده می‌شود.

php artisan serve: با اجرای این دستور، یک سرور توسعه به صورت محلی راه اندازی می‌شود تا بتوانید برنامه خود را در محیط توسعه تست کنید.

php artisan tinker: این دستور یک محیط REPL (Read-Eval-Print Loop) فراهم می‌کند که به شما این امکان را می‌دهد تا با مدل‌ها و دیگر اجزای برنامه تعامل داشته و تست اجرایی انجام دهید.

php artisan make:middleware: برای ایجاد میان‌افزارها (middleware) استفاده می‌شود.

Artisan به توسعه‌دهندگان این امکان را می‌دهد که با استفاده از خط فرمان و دستورات این ابزار، بسیاری از وظایف توسعه و مدیریت برنامه‌های Laravel را به صورت سریع و کارآمد انجام دهند.

3. engine templating چیست، معروف ترین هارا نام ببرید، لاراول از کدام استفاده

میکنند؟

موتور تمپلیتینگ یا Engine Templating یک قسمت اساسی در فریمورک‌های وب است که برای نمایش داده‌ها به کاربران و ایجاد صفحات وب استفاده می‌شود. این موتورها امکان افزودن دینامیکیت به صفحات وب را فراهم می‌کنند. برخی از موتورهای تمپلیتینگ معروف عبارتند از:

Blade (Laravel): این موتور تمپلیتینگ به صورت خاص برای فریمورک لاراول طراحی شده است. با ویژگی‌هایی مانند دستورات کنترلی، ارائه‌ی داده، و امکانات دیگر، Blade به توسعه‌دهندگان این امکان را می‌دهد که قالب‌ها را بازیابی کنند و داده‌ها را به آنها اضافه کنند.

Jinja2 (Python/Flask): این موتور تمپلیتینگ برای زبان برنامه‌نویسی پایتون و فریمورک Flask طراحی شده است. از آن به عنوان یک قسمت اصلی در ساختار قالب Flask استفاده می‌شود.

Twig (PHP/Symfony): Twig یک موتور تمپلیتینگ برای زبان برنامه‌نویسی PHP است و به عنوان موتور تمپلیت اصلی در فریمورک Symfony به کار می‌رود.

EJS (JavaScript): Embedded JavaScript یا EJS یک موتور تمپلیتینگ برای زبان جاوااسکریپت است که از آن برای ایجاد صفحات وب دینامیک استفاده می‌شود.

Handlebars (JavaScript): این موتور تمپلیتینگ برای زبان جاوااسکریپت است و ویژگی‌هایی مانند قابلیت گسترش و سادگی در استفاده دارد.

هر کدام از این موتورها ویژگی‌ها و ساختارهای خاص به خود را دارند و با زبان‌ها و فریمورک‌های خاصی همخوانی دارند. لاراول از موتور Blade استفاده می‌کند که به طور اختصاصی برای نیازها و ساختارهای لاراول طراحی شده و امکانات منحصر به فردی برای توسعه‌دهندگان لاراول فراهم می‌کند.

4. چرخه حیات درخواست در لاراول را توضیح دهید.

درخواست: (Request)

کلیه درخواست‌ها از طریق مرورگر یا سایر وسایل ارتباطی به سرور Laravel می‌آیند.

Laravel درخواست‌ها را بررسی می‌کند و بر اساس نوع درخواست (GET)، POST و غیره (و مسیر مشخص می‌کند که کدام کنترلر و اقدامی باید انجام شود.

مسیریابی: (Routing)

مسیرها به کمک فایل web.php یا api.php یا تنظیمات مسیرهای دیگر در Laravel تعیین می‌شوند.

Laravel مسیرها را تجزیه و تحلیل کرده و به کنترلرهای مرتبط هدایت می‌کند.

کنترلر: (Controller)

کنترلرها مسئول پردازش و اجرای دستورات مرتبط با درخواست هستند.

اگر درخواستی به یک کنترلر متصل شده باشد، متدهای مختلف کنترلر بر اساس نوع درخواست اجرا می‌شوند.

مدل: (Model)

مدل‌ها به تعامل با پایگاه داده اختصاص دارند. آن‌ها اطلاعات مورد نیاز را از پایگاه داده استخراج و به کنترلر ارائه می‌دهند.

مهاجرت (Migration) و الگوهای موجودیت: (Entity)

در صورت نیاز به تغییرات در پایگاه داده، مهاجرت‌ها برای اعمال تغییرات به کمک دستور php artisan migrate استفاده می‌شوند.

الگوهای موجودیت (Entity) نشان‌دهنده ساختار جدول‌های پایگاه داده هستند.

میان‌افزار: (Middleware)

میان افزارها در Laravel به توسعه دهندگان این امکان را می دهند که درخواست ها را تغییر دهند یا بررسی های اضافی انجام دهند قبل از رسیدن به کنترلر.

مشاهده: (View)

مشاهدات (Views) نقش درخواست برای نمایش اطلاعات به کاربران را ایفا می کنند.

داده های از کنترلر به مشاهدات ارسال می شوند و در صورت لزوم از آنها برای نمایش به کاربر استفاده می شود.

پاسخ: (Response)

پس از اجرای مراحل بالا، Laravel یک پاسخ به مرورگر ارسال می کند.

این پاسخ شامل HTML برای نمایش صفحه، اطلاعات JSON برای API و غیره می شود.

هر یک از این مراحل در چرخه حیات درخواست در Laravel به توسعه دهندگان این امکان را می دهد که به صورت سازمان دهی و اداره شده به توسعه برنامه بپردازند و عملکرد هر قسمت را به صورت جداگانه تعریف و پیش بینی کنند.

5. روت های Resource چه کاربردی دارند؟

روت های Resource در فریمورک Laravel برای ایجاد مسیرها و کنترلرهای مرتبط با عملیات CRUD (Create، Read، Update، Delete بر روی یک منبع) معمولاً یک مدل (Eloquent) استفاده می شوند. این روت ها به صورت خودکار مسیرها و متدهای مرتبط با عملیات CRUD را ایجاد می کنند.

در واقع، با استفاده از روت های Resource، می توانید به راحتی یک سری از کنترلرها و مسیرها را برای مدیریت داده ها ایجاد کنید، بدون این که خودتان به صورت دستی کلیه مسیرها و متدها را تعیین کنید. این کار باعث می شود که کد شما ساده تر و قابل فهم تر باشد.

برای فهم بهتر، نگاهی به یک مثال از روت‌های Resource بندازیم. فرض کنید یک مدل به نام `Post` داریم که قابلیت ایجاد، نمایش، ویرایش، و حذف پست‌ها را دارد. با استفاده از روت‌های Resource، می‌توانیم به صورت زیر مسیرها را تعریف کنیم:

```
Route::resource('posts', 'PostController');
```

این یک خط کد ساده است که به Laravel می‌گوید که تمام مسیرها و متدهای مربوط به CRUD برای منبع `Post` را ایجاد کند. بنابراین، مسیرها ممکن است به این صورت باشند:

`GET /posts` - نمایش لیست تمام پست‌ها

`GET /posts/create` - نمایش فرم ایجاد پست جدید

`POST /posts` - ایجاد یک پست جدید

`GET /posts/{id}` - نمایش جزئیات یک پست

`GET /posts/{id}/edit` - نمایش فرم ویرایش پست

`PUT /posts/{id}` - یا: `PATCH /posts/{id}` ویرایش یک پست

`DELETE /posts/{id}` - حذف یک پست

این ساختار Resource در Laravel به توسعه‌دهندگان این امکان را می‌دهد که با حداقل تکرار کد، عملیات CRUD را بر روی منابع خود پیاده‌سازی کنند و به طراحی خوانا و قابل نگهداری برنامه کمک کند.

6. روت‌های Named چه کاربردی دارند؟

روت‌های Named در Laravel برای تعیین نام خاص به یک مسیر و دسترسی به آن مسیر از سایر قسمت‌های برنامه استفاده می‌شوند. با استفاده از این نام‌ها، می‌توان به راحتی به مسیرها ارجاع داده یا آنها را در قسمت‌های مختلف برنامه مورد استفاده قرار داد.

برخی از موارد استفاده از روت‌های Named:

1. تولید URL

- با استفاده از نام‌های روت، می‌توان به راحتی URL مرتبط با یک مسیر را ایجاد کرد. این کار به جای استفاده از URL‌های ثابت، کمک می‌کند تا در صورت تغییر در ساختار URL، تغییرات کمتری لازم باشد.

2. استفاده در کنترلرها

- در کنترلرها ممکن است نیاز باشد به یک مسیر ارجاع داده شود. با استفاده از نام روت، این کار به صورت خواناتر و قابل نگهداری‌تر انجام می‌شود.

3. استفاده در ویوها

- در ویوها نیز ممکن است به مسیرها نیاز باشد. استفاده از نام‌های روت در اینجا نیز مزایای مشابه با کنترلرها دارد.

4. استفاده در میان‌افزارها Middleware

- در میان‌افزارها نیز ممکن است نیاز به ارجاع به یک مسیر باشد. نام‌های روت در اینجا نیز می‌توانند کاربرد داشته باشند.

برای تعریف یک نام برای روت، می‌توانید از تابع `name`` در تعریف مسیرها استفاده کنید. مثال:

```
Route::get('/user/profile', 'UserProfileController@show')->name('profile');
```

در این مثال، یک نام به روت اختصاص داده شده است که `profile`` نامیده شده است. حالا می‌توانید از این نام در سایر قسمت‌های برنامه برای ارجاع به این مسیر استفاده کنید.

7. تفاوت {!!} و {{}} در Blade چیست؟

در Blade، `{{}}` برای نمایش محتویات متغیرها یا ارزیابی عبارات PHP استفاده می‌شود. این نوع نشانگر معمولی است و محتوای داخل آن به صورت HTML-escaped یعنی HTML entities برای کاراکترهای خاص ایجاد

می شود (نمایش داده می شود. به عبارت دیگر، محتویات داخل `{{{` از نظر امنیتی اسکیپ می شوند تا از حملات XSS جلوگیری شود.

اما `!!!` یا `!!!` از نوع `!!!` نشانگر است و این نوع نشانگر به عنوان "نشانگر غیر-اسکیپ" یا "نشانگر RAW" شناخته می شود. با استفاده از `!!!`، محتویات داخل آن به صورت raw بدون اسکیپ (HTML نمایش داده می شود. این بدان معناست که هر چیزی که داخل `!!!` قرار دارد، به صورت خام (بدون تغییر) نمایش داده می شود.

مثال:

```
<p>{{ $variable }}</p>
```

```
<p>{!! $variable !!}</p>
```

استفاده از `!!!` باید با دقت انجام شود، زیرا این کار ممکن است به افزایش خطر حملات XSS منجر شود. در صورتی که مطمئن هستید که متغیرها مطمئن و تمیز هستند، می توانید از `!!!` استفاده کنید، اما در مواردی که اطمینان ندارید، بهتر است از `{{{` استفاده کنید تا از اسکیپ HTML به منظور امنیت بیشتر استفاده کنید.

8. آدرس `{cat}/{id}/{title}` به صورت ادامه در نظر بگیرید

Title ? alphanit.

id ? numeric

Cat ? cat has to be one of sport, economy, values.

اعتبار سنجی های الزم را انجام دهید و پس از تایید صفحه نمایش `/cat{/}` را `php.blade.index` نمایش دهید که متن

Title => {title},id=>{id},cat=>{cat}

در آن نمایش داده شده است

برای اعتبار سنجی آدرس مطلوب و نمایش صفحه در Blade در Laravel ، می توانید از Route Parameters برای دریافت مقادیر مختلف استفاده کنید. همچنین، می توانید از validation rules برای اعتبار سنجی اطلاعات ورودی استفاده کنید.

ابتدا، باید Route و Controller را تعریف کنید. در اینجا یک مثال تعریف شده است:

```
Route::get('/php.blade.index/{cat}', 'YourController@index')->where('cat',  
'sport|economy|values');
```

سپس در کنترلر مربوطه:(`YourController`)

```
public function index($cat)  
{  
    return view('php.blade.index', compact('cat'));  
}
```

حالا باید validation rules را در کنترلر اعمال کنیم:

```
public function index($cat)  
{  
    $rules = [  
        'cat' => 'in:sport,economy,values',  
    ];  
  
    $validator = Validator::make(['cat' => $cat], $rules);  
  
    if ($validator->fails()) {
```

```
}  
  
return view('php.blade.index', compact('cat'));  
}
```

در نهایت، در فایل `php.blade.index.blade.php` می‌توانید اطلاعات را نمایش دهید:

```
Title => alphabit, id => {id}, cat => {{ $cat }}
```

توجه داشته باشید که `{id}` باید از درخواست یا سایر روش‌های شما برای دریافت مقدار id برای صفحه استفاده شود. در مثال فوق، من از آن به عنوان یک placeholder استفاده کردم، شما می‌توانید آن را با مقدار مورد نظر خود جایگزین کنید.