سوال 1

لاراول محبوب ترین فرم ورک **PHP** می باشد که تحت معماری **MVC** بنا شده است و برای توسعه نرم افزار های تحت وب طراحی شده است، با اینکه تحت سرور نوشته شده است ولی شما میتوانید فرانت اند خود را هم از طریق ابزار های قوی که در اختیار شما می گذارد از قیبل **Mix** و **Blade** آن را مدیریت و توسعه دهید.

یکی از مزیت های دیگر لاراول این است که برای مدیریت و وابستگی پکیچ های خود از  [HYPERLINK "https://getcomposer.org/download/"Compose](HYPERLINK%20%22https://getcomposer.org/download/%22Compose)r استفاده می کند و شما میتوانید براحتی یک پکیج از بیرون در پروژه لاراول خود نصب و استفاده کنید.

همچنین از نظر امنیت از سیستم اهراز هویت پشتیبانی می کند که کار کردن با این فرم ورک را جذاب می کند و همچنین ویژگی ها و امکانات کاربردی دیگر برای کار با دیتابیس از قبیل **ORM** و **Eloquent** و **Artisan** که میتوانید خط فرمان لاراول را به خوبی مدیریت کنید و خیلی از وپژگی های منحصر به فرد دیگری که کار کردن با این فرم ورک را جذاب می کند.

سه تا از ویژگی ها دیگر لاراول به نظرم باید به آنها اشاره کنم مورد اول در حال رشد **(Progressive)** بود آن است یعنی به طور مداوم این فرم ورک در حال تغییر می باشد و باید دانش خودتون نسبت بهش بروز کنید. مورد دوم مقیاس پذیری **(Scalable)** خب همانطور که می دانید ما هرچه بتونیم با منابع کمتر عملکرد بالاتری داشته باشیم خیلی برای ما بصرفه تر و بهتر است خب با پشتیبانی و بروز رسانی های **PHP** و با فراهم ساختن امکان سیستم های کش مثل **Redis** ما میتوانیم میلیون ها درخواست را در ماه مدیریت کنیم.

**هر پوشه در لاراول وظیفه چه کاری را دارد‌ ؟**

**پوشه app**

هسته اصلی لاراول که در آن تمام کتابخانه و سرویس های اجرا می شوند در این پوشه می باشد.

**پوشه bootstrap**

هنگام راه اندازی پروژه اولین پوشه که شروع **(root)** به اجرای بقیه اجزا می کند می باشد که درون خود یک فایل به نام **app.php** است که وظیفه راه اندازی فرم ورک را بر عهده دارد و پوشه ای به نام **cache** که فرم ورک فایل های را برای افزایش سرعت، تولید میکند. فایل های از قبیل تنظیمات یا روت ها و سرویس ها می باشد.

**پوشه config**

تمام تنظیمات و پیکربندی های لاراول از قبیل دیتابیس، کش، سشن، هش و غیره ... در این اینجا می باشد.

**پوشه database**

تمام فایل های مربوط به **migartion** ها و مدیریت کلاس **Factory** و **Seeder** در اینجا وجود دارد.

**پوشه public**

این پوشه عمومی هدف تمام درخواست های است که از سمت کلاینت به سمت اپلیکیشن شما ارسال می شود می باشد که توسط **htaccess** مدیریت شده است و همچنین فایل های ثابتی از قبیل **CSS** و **JS** در خود نگه میدارد.

**پوشه routes**

به صورت پیش فرض تمام روت های که شما تعریف می کنید در این پوشه قرار دارد که شامل این فایل های **web.php, api.php, channels.php, console.php** می باشد پیکربندی های فایل **web.php** در داخل سرویس **RouteServiceProvider** تعریف شده اند که بعنوان مثال همه آنها از **middleware** به نام **web** گذر می کنند و برای تمام روت ها اعمال می شود.

**پوشه resources**

این پوشه شامل views های است که در پروژه خود تعریف می کنید همچنین تمام کد های برای **css** و **javascript** می نویسید درون این قسمت قرار دارد و بعد از کامپایل شدن به پوشه **public** منتقل می شود. همچین تمام زبانهای که ممکن است در اپ خود استفاده کنید در پوشه به نام **lang** وجود دارد.

**پوشه storage**

این پوشه شامل تمام **logs, caches, sessions** و فایل های کامپایل شده **blade** می باشد که توسط برنامه شما ایجاد شده.

**پوشه tests**

این پوشه شامل تست های است که توسط **PHPUnit** برای سیستم خود نوشته اید.

**پوشه vendor**

این پوشه شامل تمام پکیج هایی می باشد که توسط **composer** برای وابستگی های پروژه خود نصب کرده اید.

سوال 2

است که به عنوان command Line Interface یک رابط کاربری خط فرمان یا همان PHP Artisan

یک ابزار کمکی در فریم ورک لاراول به کار برده می شود. دستوراتی که در Artisan وجود دارد این امکان را فراهم می‌کنند که اپلیکیشن‌های مختلف به سادگی ساخته شوند. به طور خلاصه **artisan** یک رابط خط فرمان **CLI** اختصاصی لاراول می باشد که سرعت فوق العاده ای به توسعه برنامه ما می بخشد.

سوال 3

موتور قالب به شما امکان می دهد از فایل های قالب ثابت static template در برنامه خود استفاده کنید. در زمان اجرا، موتور قالب متغیرهای موجود در فایل قالب را با مقادیر واقعی جایگزین می‌کند و قالب را به یک فایل HTML ارسال شده برای مشتری تبدیل می‌کند. این رویکرد طراحی یک صفحه HTML را آسان تر می کند.

مشهورترین template engine ها عبارتند از :

Twig , Smarty , Dwoo , Blade , Volt , Plates , Mustache

لاراول از Blade استفاده می کند.

سوال 4

چرخه حیات درخواست

اولین قدم

نقطه ورودی برای کل درخواست هایی که به یک برنامه لاراول ارسال می شود، فایل public/index.php است. همه درخواست ها بر اساس تنظیمات و پیکربندی وب سرور ([Apache](https://www.apache.org/) / [Nginx](https://www.nginx.com/)) به این فایل منتقل می شوند. این فایل کد زیادی ندارد و فقط یک شروع کننده برای کل فریم ورک است. فایل public/index.php تعاریف مربوط به autoloader که توسط  [HYPERLINK "https://getcomposer.org/"Composer](HYPERLINK%20%22https://getcomposer.org/%22Composer) تولید شده اند را بارگذاری کرده و یک نمونه از bootstrap/app.php را دریافت می کند.

دومین قدم

درخواست ورودی بعد از گذشتن از فایل public/index.php بر اساس نوع درخواست به کرنل HTTP یا Console ارسال می شود. همه درخواست های ورودی به برنامه از داخل این دو کرنل عبور می کنند. کرنل HTTP در داخل فایل app/Http/Kernel.php و کرنل Console در داخل فایل app/Console/Kernel.php قرار دارد. کرنل HTTP از کلاس  [HYPERLINK "https://github.com/laravel/framework/blob/5.8/src/Illuminate/Foundation/Http/Kernel.php"Illuminate\Foundation\Http\Kernel](HYPERLINK%20%22https://github.com/laravel/framework/blob/5.8/src/Illuminate/Foundation/Http/Kernel.php%22Illuminate%5CFoundation%5CHttp%5CKernel) ارث بری می کند. این کلاس یک آرایه به نام bootstrappers دارد که موارد موجود در آن قبل از اجرا شدن درخواست، اجرا می شوند. به طور کلی این کلاس پیکربندی های مربوط به خود فریم ورک را انجام می دهد و نیاز نیست کاری انجام دهید.

کرنل HTTP لیستی از Middleware ها را تعریف می کند که همه درخواست ها قبل از هندل شدن توسط برنامه از داخل آن ها عبور می کند. Middleware های مربوط به احراز هویت، بررسی توکن  [HYPERLINK "https://en.wikipedia.org/wiki/Cross-site\_request\_forgery"CSRF](HYPERLINK%20%22https://en.wikipedia.org/wiki/Cross-site_request_forgery%22CSRF) و غیره در این بخش تعریف شده اند.

ارائه دهندگان سرویس (Service Providers)

یکی از عملیات مهم مربوط به آماده‌سازی کرنل، بارگذاری ارائه دهندگان سرویس برای برنامه است. تمام سرویس های مورد استفاده در برنامه در داخل فایل config/app.php و آرایه providers تنظیم و پیکربندی می شوند. لاراول در هنگام اجرا هر کدام از سرویس های مشخص شده در آرایه providers را نمونه سازی می کند. بعد از نمونه سازی متد register که در همه سرویس ها وجود دارد فراخوانی می شود و زمانی که همه سرویس ها ثبت شدند، متد boot هر سرویس فراخوانی می شود.

ارائه دهندگان سرویس مسئول آماده‌سازی بخش های مختلف فریم ورک مانند دیتابیس، صف، اعتبار سنجی، مسیریابی و غیره هستند. اساسا ویژگی های اصلی مربوط به لاراول توسط ارائه دهندگان سرویس پیکربندی و آماده می شود.

مسیریابی (Routing)

یکی از مهم ترین ارائه دهندگان سرویس در لاراول، App\Providers\RouteServiceProvider است که فایل های مربوط به مسیرهای برنامه را که در [دایرکتوری HYPERLINK "https://sourcesara.com/laravel-directory-structure/" HYPERLINK "https://sourcesara.com/laravel-directory-structure/"routes](https://sourcesara.com/laravel-directory-structure/) قرار دارند، بارگذاری می کند. بعد از آنکه بخش های مختلف برنامه آماده شد (به اصطلاح بوت استرپ شد)، درخواست ارسال شده به برنامه برای مدیریت شدن به روتر فرستاده می شود. سپس روتر درخواست را به یک مسیر یا کنترلر ارسال می کند و همچنین middleware های لازم را اجرا می کند.

میان افزارها (Middleware) مکانیزمی برای فیلتر کردن درخواست های HTTP ارسالی به برنامه هستند. برای مثال لاراول یک میان افزار برای احراز هویت دارد که بررسی میکند که آیا کاربر احراز هویت شده است یا خیر. اگر کاربر احراز هویت نشده باشد، آن را به صفحه لاگین هدایت می کند و در غیر این صورت اجازه می دهد تا درخواست عبور کند.

ارسال پاسخ درخواست

زمانی که مراحل فوق انجام و درخواست دریافتی پردازش شد، متد handle مربوط به کرنل HTTP پاسخ آن را به فایل index.php باز میگرداند و در فایل index.php متد send مربوط به پاسخ فراخوانی می شود. این متد محتوای پاسخ را به مرورگر کاربر ارسال می کند.

سوال 5

معمولا ما یک سری عملیات های خاص رو تحت وب انجام میدیم ، به طور مثال در مورد مقالات:

* به وجود آوردن یک مقاله ( درخواست POST)
* خواندن یک مقاله ( درخواست GET)
* ویرایش یک مقاله ( درخواست PUT/PATCH)
* حذف یک مقاله ( درخواست DELETE)

که اصطلاحا بهش CRUD میگن. حالا وقتی که از Resource Route استفاده میکنیم ، مثل حالت زیر :

Route::resource('articles', 'ArticleController');

در این حالت لاراول Route هایی که برای عملیات CRUD لازمه رو اضافه میکنه ، یعنی با یک خط کد 7 تا Route رو تعریف میکنه :

index , create , store , show , edit , update , destroy

سوال 6

برای Routing در لاراول ، هنگامی که Route‌ها را نام‌گذاری می‌کنیم، می‌توانیم به راحتی آن‌ها را در هر صفحه‌ای فراخوانی کنیم. برای تعریف یک نام مشخص برای Route مورد نظر از متد name استفاده می‌کنیم.

سوال 7

هر دو برای نمایش اطلاعات به کارمیرن {{ }} برای جلوگیری از حمله XSS بکار میره. وقتی از اون برای نمایش اطلاعات استفاده می کنیم ، اطلاعات ابتدا از تابع htmlspecialchars عبور میکنه و بعد نمایش داده میشه.

{{ "<script>alert('Hello')</script>" }}

خروجی کد بالا بصورت زیر است :

&lt;script&gt;alert(&#039;Hello&#039;)&lt;/script&gt;

اما اگر به جای {{ }} از {!! !!} استفاده کنیم ، اطلاعات خام و بدون تبدیل نمایش داده میشن:

{!! "<script>alert('Hello')</script>" !!}

با اجرا شدن کد بالا ، محتویات درون تگ script اجرا میشه و اینجا یک alert جاوااسکریپتی می گیریم. پس باید در نظر داشته باشیم زمانی که می خوایم اطلاعاتی رو نمایش بدیم که از صحت اون اطمینان نداریم ، مثلا فرمی که یک کاربر ثبت کرده ، بهتره از {{ }} استفاده کنیم.