Ques. What is Laravel?

- Laravel is free open source “PHP framework” based on MVC design pattern.
- It is created by **Taylor Otwell**
- Laravel provides expressive and elegant syntax that helps in creating a wonderful web application easily and quickly.
- The first version of laravel is released on 9 June 2011.
- The latest version of Laravel is **9.0**

Ques. What is the Features of Laravel?

- Inbuilt CRSF (cross-site request forgery) Protection.
- Inbuilt paginations
- middleware
- Eloquent ORM
- Query builder available
- Reverse Routing
- Restful Controllers
- Migration
- Database Seeding/Database Migration
- Autamatic Pagination
- Unit Testing
- Homestead
- Query builder
- Route caching
- IOC (Inverse of Control) Container Or service container.

↥ Back to Top

Ques. What are pros and cons of using Laravel Framework?

Pros of using Laravel Framework

1. Laravel framework has in-built lightweight blade template engine to speed up compiling task and create layouts with dynamic content easily.
2. Hassles code reusability.
3. Eloquent ORM with PHP active record implementation
4. Built in command line tool “Artisan“ for creating a code skeleton , database structure and build their migration

Cons of using laravel Framework

1. Development process requires you to work with standards and should have real understanding of programming
2. Laravel is new framework and composer is not so strong in compare to npm (for node.js), ruby gems and python pip.
3. Development in laravel is not so fast in compare to ruby on rails.

4. Laravel is lightweight so it has less inbuilt support in compare to django and rails. But this problem can be solved by integrating third party tools, but for large and very custom websites it may be a tedious task.

⬆ Back to Top

Ques. What are available databases supported by Laravel?

- PostgreSQL
- SQL Server
- SQLite
- MySQL

⬆ Back to Top

Ques. What are the steps to install Laravel with composer?

Laravel installation steps:-

- Download composer from https://getcomposer.org/download (if you don't have a composer on your system)
- Open cmd
- Goto your htdocs folder.
- C:\xampp\htdocs> composer create-project laravel/laravel projectname

```
If you install some particular version, then you can use
composer create-project laravel/laravel project name "5.6"
If you did not mention any particular version, then it will install with the
latest version.
```

⬆ Back to Top

Ques. What is composer?

- Composer is the package manager for the framework.
- It helps in adding new packages from the huge community into your laravel application.
- Composer is a tool for managing dependency in PHP. It allows you to declare the libraries on which your project depends on and will manage (install/update) them for you Laravel utilizes Composer to manage its dependencies.
- It is an application-level package manager for PHP. It provides a standard format for managing PHP software dependencies and libraries. Example:-

```
composer requires laravel/passport
```

⬆ Back to Top

Ques. What is the templating engine used in Laravel?

- The templating engine used in Laravel is **Blade**.
- **Displaying data** If you want to print the value of a variable, then you can do so by simply enclosing the variable within the curly brackets.
  Syntax:-

```
{{$variable}};
```

↥ **Back to Top**

Ques. How to put Laravel applications in maintenance mode?

Laravel applications can be put into maintenance mode using the below command:

```
php artisan down
```

And can put the application again on live using the below command:

```
php artisan up
```

↥ **Back to Top**

Ques. What is a Route?

- A route is basically an endpoint specified by a URI (Uniform Resource Identifier). It acts as a pointer in Laravel application.
- Most commonly, a route simply points to a method on a controller and also dictates which HTTP methods are able to hit that URI.

↥ **Back to Top**

Ques. Why use Route?

Routes are stored inside files under the /routes folder inside the project's root directory. By default, there are a few different files corresponding to the different "sides" of the application ("sides" comes from the hexagonal architecture methodology).

↥ **Back to Top**

Ques. Explain Events in laravel?

- An event is an action or occurrence recognized by a program that may be handled by the program or code. Laravel events provides a simple observer implementation, that allowing you to subscribe and listen for various events/actions that occur in your application.
- All Event classes are generally stored in the app/Events directory, while their listeners are stored in app/Listeners of your application.

⬆ Back to Top

Ques. Explain validations in laravel?

In Programming validations are a handy way to ensure that your data is always in a clean and expected format before it gets into your database.
Laravel provides several different ways to validate your application incoming data.By default Laravel's base controller class uses a **ValidatesRequests trait** which provides a convenient method to validate all incoming HTTP requests coming from client.You can also validate data in laravel by creating Form Request.

```
Laravel validation Example

$validatedData = $request->validate([
    'name' => 'required|max:255',
    'username' => 'required|alpha_num',
    'age' => 'required|numeric',
  ]);
```

⬆ Back to Top

Ques. What is PHP artisan List out some artisan commands?

Artisan is the command-line tool for Laravel to help the developer build the application.
Here are the list of some artisan commands:-

- php artisan list
- php artisan help
- php artisan tinker
- php artisan make
- php artisan -versian
- php artisan make model model_name
- php artisan make controller controller_name

⬆ Back to Top

Ques. List Some default packages provided by Laravel 5.6?

- Cashier
- Envoy
- Passport
- Scout
- Socialite
- Horizon
- Telescope

⬆ Back to Top

Ques. What are named routes in Laravel?

Named routing is another amazing feature of Laravel framework. Named routes allow referring to routes when generating redirects or Urls more comfortably.
You can specify named routes by chaining the name method onto the route definition:

```
Route::get('user/profile', function () {
     //
})->name('profile');
```

You can specify route names for controller actions:

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

Once you have assigned a name to your routes, you may use the route's name when generating URLs or redirects via the global route function:

```
// Generating URLs...
  $url = route('profile');
// Generating Redirects...
  return redirect()->route('profile');
```

⬆ Back to Top

Ques. What is database migration and How to create migration via artisan?

- Migrations are used to create database schemas in Laravel.
- In migration files, we store which table to create, update or delete.

```
// creating Migration
php artisan make:migration create_users_table
```

⬆ Back to Top

Ques. What are service providers?

Service Providers are central place where all laravel application is bootstrapped . Your application as well all Laravel core services are also bootstrapped by service providers.
All service providers extend the Illuminate\Support\ServiceProvider class. Most service providers contain a register and a boot method. Within the register method, you should only bind things into the service container. You should never attempt to register any event listeners, routes, or any other piece of functionality within the register method.

⬆ Back to Top

Ques. Explain Laravel’s service container?

One of the most powerful feature of Laravel is its Service Container. It is a powerful tool for resolving class dependencies and performing dependency injection in Laravel.
Dependency injection is a fancy phrase that essentially means class dependencies are “injected” into the class via the constructor or, in some cases, “setter” methods.

⬆ Back to Top

Ques. What is dependency injection in Laravel?

In software engineering, dependency injection is a technique whereby one object supplies the dependencies of another object. A dependency is an object that can be used (a service). An injection is the passing of a dependency to a dependent object (a client) that would use it. The service is made part of the client’s state.[1] Passing the service to the client, rather than allowing a client to build or find the service, is the fundamental requirement of the pattern.
You can do dependency injection via Constructor, setter and property injection.

⬆ Back to Top

Ques. What are Laravel Contract’s?

Laravel’s Contracts are nothing but a set of interfaces that define the core services provided by the Laravel framework.

⬆ Back to Top

Ques. Explain Facades in Laravel?

Laravel Facades provides a static like an interface to classes that are available in the application’s service container. Laravel self-ships with many facades which provide access to almost all features of Laravel ’s. Laravel facades serve as “static proxies” to underlying classes in the service container and provide benefits of a terse, expressive syntax while maintaining more testability and flexibility than traditional static methods of classes. All of Laravel’s facades are defined in the Illuminate\Support\Facades namespace. You can easily access a facade like so:

```
use Illuminate\Support\Facades\Cache;
Route::get('/cache', function () {
  return Cache::get('key');
  });
```

⬆ Back to Top

Ques. What are Laravel eloquent?

Laravel’s Eloquent ORM is simple Active Record implementation for working with your database. Laravel provide many different ways to interact with your database, Eloquent is most notable of them. Each database table has a corresponding “Model” which is used to interact with that table. Models allow you to query for

data in your tables, as well as insert new records into the table.
Below is sample usage for querying and inserting new records in Database with Eloquent.

```
// Querying or finding records from products table where tag is 'new'
$products= Product::where('tag','new');
// Inserting new record
$product =new Product;
$product->title="Iphone 7";
$product->price="$700";
$product->tag='iphone';
$product->save();
```

⬆ Back to Top

Ques. How to enable query log in Laravel?

Use the enableQueryLog method to enable query log in Laravel

```
DB::connection()->enableQueryLog();
You can get array of the executed queries by using getQueryLog method:
$queries = DB::getQueryLog();
```

⬆ Back to Top

Ques. What is reverse routing in Laravel?

- Laravel reverse routing is generating URL's based on route declarations. Reverse routing makes your application so much more flexible. It defines a relationship between links and Laravel routes. When a link is created by using names of existing routes, appropriate Uri's are created automatically by Laravel. Here is an example of reverse routing.
- Reverse routing is a method of generating URL based on symbol or name. It makes your Laravel application flexible.

```
// route declaration<br>
Route::get('login', 'users@login');
```

Using reverse routing we can create a link to it and pass in any parameters that we have defined. Optional parameters, if not supplied, are removed from the generated link.

```
{{ HTML::link_to_action('users@login') }}
```

It will automatically generate an Url like http://xyz.com/login in view.

⬆ Back to Top

Ques. How to turn off CRSF protection for specific route in Laravel?

To turn off CRSF protection in Laravel add following codes in “app/Http/Middleware/VerifyCsrfToken.php”

```
//add an array of Routes to skip CSRF check
private $exceptUrls = ['controller/route1', 'controller/route2'];
//modify this function
public function handle($request, Closure $next) {
//add this condition foreach($this->exceptUrls as $route) {
if ($request->is($route)) {
return $next($request);
} }
return parent::handle($request, $next);
}
```

⬆ Back to Top

Ques. What are traits in Laravel?

PHP Traits are simply a group of methods that you want include within another class. A Trait, like an abstract class cannot be instantiated by itself.Trait are created to reduce the limitations of single inheritance in PHP by enabling a developer to reuse sets of methods freely in several independent classes living in different class hierarchies.

Here is an example of trait.

```
trait Sharable {
public function share($item)
{
   return 'share this item';
} }
```

You could then include this Trait within other classes like this:

```
class Post {
  use Sharable;
 }
class Comment {
use Sharable;
}
```

Now if you were to create new objects out of these classes you would find that they both have the share() method available:

```
$post = new Post;
echo $post->share(''); // 'share this item'
$comment = new Comment;
echo $comment->share(''); // 'share this item'
```

⬆ Back to Top

Ques. Does Laravel support caching?

Yes, Laravel supports popular caching backends like Memcached and Redis.
By default, Laravel is configured to use the file cache driver, which stores the serialized, cached objects in the file system.For large projects, it is recommended to use Memcached or Redis.

⬆ Back to Top

Ques. How To Clear Cache In Laravel?

Clear database cache

```
php artisan cache:clear
```

Clear config cache

```
php artisan config:cache
```

Laravel Clear Cache on Shared Hosting Server

```
Route::get('/clear-cache', function() {
    $exitCode = Artisan::call('cache:clear');
    return "Cache is cleared";
});
```

⬆ Back to Top

Ques. What is HTTP middleware?

HTTP middleware is a technique for filtering HTTP requests. Laravel includes a middleware that checks whether application user is authenticated or not.

⬆ Back to Top

Ques. Explain Laravel's Middleware?

Middleware acts as a middleman between request and response. It is a type of filtering mechanism. For example, Laravel includes a middleware that verifies whether the user of the application is authenticated or not. If the user is authenticated, he will be redirected to the home page otherwise, he will be redirected to the login page.
$nbsp;
In Laravel, middleware operates as a bridge and filtering mechanism between a request and response. It verifies the authentication of the application users and redirects them according to the authentication results. We can create a middleware in Laravel by executing the following command.
Example: If a user is not authenticated and it is trying to access the dashboard then, the middleware will redirect that user to the login page.

```
Now UserMiddelware.php file will create in
```

There are two types of Middleware in Laravel.

Global Middleware: will run on every HTTP request of the application.
Route Middleware: will be assigned to a specific route.

⬆ Back to Top

Ques. What is Lumen?

Lumen is PHP micro-framework that built on Laravel’s top components.It is created by Taylor Otwell. It is perfect option for building Laravel based micro-services and fast REST API’s. It’s one of the fastest micro-frameworks available.
You can install Lumen using composer by running below command

```
composer create-project --prefer-dist laravel/lumen blog
```

⬆ Back to Top

Ques. Explain Bundles in Laravel?

- In Laravel, bundles are also called packages. Packages are the primary way to extend the functionality of Laravel. Packages might be anything from a great way to work with dates like Carbon, or an entire BDD testing framework like Behat.In Laravel, you can create your custom packages too.
- In Laravel, bundles are referred to as packages. These packages are used to increase the functionality of Laravel. A package can have views, configuration, migrations, routes, and tasks.

⬆ Back to Top

Ques. How to use custom table in Laravel Modal?

You can use custom table in Laravel by overriding protected $table property of Eloquent.

```
class User extends Eloquent{
   protected $table="my_user_table";
}
```

⬆ Back to Top

Ques. List types of relationships available in Laravel Eloquent?

Below are types of relationships supported by Laravel Eloquent ORM.

- One To One
- One To Many
- One To Many (Inverse)
- Many To Many
- Has Many Through
- Polymorphic Relations
- Many To Many Polymorphic Relations

⬆ Back to Top

Ques. Why are migrations necessary?

Migrations are necessary because:

- Without migrations, database consistency when sharing an app is almost impossible, especially as more and more people collaborate on the web app.
- Your production database needs to be synced as well.

⬆ Back to Top

Ques. Provide System requirements for installation of Laravel framework?

In order to install Laravel, make sure your server meets the following requirements:

- PHP >= 7.1.3
- OpenSSL PHP Extension
- PDO PHP Extension
- Mbstring PHP Extension
- Tokenizer PHP Extension
- XML PHP Extension
- Ctype PHP Extension
- JSON PHP Extension

⬆ Back to Top

Ques. List some Aggregates methods provided by query builder in Laravel?

- count()
- max()

- min()
- avg()
- sum()

⬆ Back to Top

Ques. How to check request is ajax or not?

In Laravel, we can use $request->ajax() method to check request is ajax or not.

```
public function saveData(Request $request)
 {
    if($request->ajax()){
    return "Request is of Ajax Type";
  }
    return "Request is of Http type";
}
```

⬆ Back to Top

How To Get User Details By Id Or Email In Laravel?

Get user by email in Laravel 5

```
public function getUserByEmail(Request $request){
    $email =$request->input('email');
    $user = User::where('email',$email)->first();
    return $user;
}
```

Get user by id in Laravel 5

```
public function getUserById(Request $request){
  $id =$request->input('id');
    $user =  User::find($id);
    return $user;
}
```

```
What are named routes in Laravel?
Route::get('user/profile', 'UserController@showProfile')->name('profile');
What is database migration. How to create migration via artisan ?
Migrations are like version control for your database, that’s allow your team to
easily modify and share the application’s database schema. Migrations are
typically paired with Laravel’s schema builder to easily build your application’s
```

```
database schema.
Use below commands to create migration data via artisan.
php artisan make:migration create_users_table
What are service providers ?
Service Providers are central place where all laravel application is bootstrapped
. Your application as well all Laravel core services are also bootstrapped by
service providers.
All service providers extend the Illuminate\Support\ServiceProvider class.
Explain Laravel’s service container ?
One of the most powerful feature of Laravel is its Service Container. It is a
powerful tool for resolving class dependencies and performing dependency injection
in Laravel.
Dependency injection is a fancy phrase that essentially means class dependencies
are “injected” into the class via the constructor or, in some cases, “setter”
methods.
What are Laravel Contract’s ?
Laravel’s Contracts are nothing but a set of interfaces that define the core
services provided by the Laravel framework.
What are Laravel eloquent?
Laravel’s Eloquent ORM is simple Active Record implementation for working with
your database. Laravel provide many different ways to interact with your database,
Eloquent is most notable of them. Each database table has a corresponding “Model”
which is used to interact with that table. Models allow you to query for data in
your tables, as well as insert new records into the table.
Below is sample usage for querying and inserting new records in Database with
Eloquent.

// Querying or finding records from products table where tag is 'new'
$products= Product::where('tag','new');
// Inserting new record
 $product =new Product;
 $product->title="Iphone 7";
 $product->price="$700";
 $product->tag='iphone';
 $product->save();
How to enable query log in Laravel ?
Use the enableQueryLog method to enable query log in Laravel

DB::connection()->enableQueryLog();
You can get array of the executed queries by using getQueryLog method:
$queries = DB::getQueryLog();
How to turn off CRSF protection for specific route in Laravel?
To turn off CRSF protection in Laravel add following codes in
“app/Http/Middleware/VerifyCsrfToken.php”

//add an array of Routes to skip CSRF check
private $exceptUrls = ['controller/route1', 'controller/route2'];
 //modify this function
public function handle($request, Closure $next) {
 //add this condition foreach($this->exceptUrls as $route) {
 if ($request->is($route)) {
  return $next($request);
 }
}
return parent::handle($request, $next);
}
Does Laravel support caching?
Yes, Laravel supports popular caching backends like Memcached and Redis.
By default, Laravel is configured to use the file cache driver, which stores the
serialized, cached objects in the file system.For large projects, it is
recommended to use Memcached or Redis.
Explain Laravel’s Middleware?
As the name suggests, Middleware acts as a middleman between request and response.
It is a type of filtering mechanism. For example, Laravel includes a middleware
that verifies whether the user of the application is authenticated or not. If the
user is authenticated, he will be redirected to the home page otherwise, he will
be redirected to the login page.
There are two types of Middleware in Laravel.
Global Middleware: will run on every HTTP request of the application.
Route Middleware: will be assigned to a specific route.
How to use custom table in Laravel Modal ?
You can use custom table in Laravel by overriding protected $table property of
Eloquent.

Below is sample uses

class User extends Eloquent{
 protected $table="my_user_table";

}
List some Aggregates methods provided by query builder in Laravel ?
Posted by Sharad Jaiswal
• count()
• max()
• min()
• avg()
• sum()
• Class - This is a programmer-defined data type, which includes local functions
as well as local data. You can think of a class as a template for making many
instances of the same kind (or class) of object.
• Object − An individual instance of the data structure defined by a class. You
define a class once and then make many objects that belong to it. Objects are also
known as instance.
• Member Variable − These are the variables defined inside a class. This data will
be invisible to the outside of the class and can be accessed via member functions.
These variables are called attribute of the object once an object is created.
• Member function − These are the function defined inside a class and are used to
access object data.
• Inheritance − When a class is defined by inheriting existing function of a
parent class then it is called inheritance. Here child class will inherit all or
few member functions and variables of a parent class.
• Parent class − A class that is inherited from by another class. This is also
called a base class or super class.
• Child Class − A class that inherits from another class. This is also called a
subclass or derived class.
• Polymorphism − This is an object oriented concept where same function can be
used for different purposes. For example function name will remain same but it
make take different number of arguments and can do different task.
```

```
• Overloading – a type of polymorphism in which some or all of operators have
different implementations depending on the types of their arguments. Similarly
functions can also be overloaded with different implementation.
• Data Abstraction – Any representation of data in which the implementation
details are hidden (abstracted).
• Encapsulation – refers to a concept where we encapsulate all the data and member
functions together to form an object.
• Constructor – refers to a special type of function which will be called
automatically whenever there is an object formation from a class.
• Destructor – refers to a special type of function which will be called
automatically whenever an object is deleted or goes out of scope.
```