**What is PHP**

**PHP** was at first created as a simple scripting platform called "**Personal Home Page**". Nowadays **PHP** is an alternative of the Microsoft's Active Server Pages (**ASP**) technology.

**PHP** is an open source server-side language which is used for creating dynamic web pages. It can be embedded into **HTML**. **PHP** is usually used in conjunction with a MySQL database on Linux/UNIX web servers. It is probably the most popular scripting language.

**PHP** is a widely-used general-purpose scripting language and interpreter that is freely available. A full explanation of all the **PHP** functions, complete user manual and lots of tutorials can be found on the [PHP's official page](http://www.php.net/).

**Different Version of PHP**

<https://en.wikipedia.org/wiki/PHP>

**What’s New in PHP 7.1+**

### iterable pseudo type

### Closure from callable

### Square bracket syntax for list() aka Symmetric array destructuring

### Support for keys in list

### Class constant visibility

### Nullable types

### Void functions

### Catching Multiple Exception Types

### Too few arguments exception.

**What is LARAVEL?**

**?????????????????**

**What is a Framework?**

**Software Framework**

According to [wikipedia](https://en.wikipedia.org/wiki/Software_framework), a software framework is an abstraction in which software providing generic functionality can be selectively changed by additional user-written code, thus providing application-specific software. A software framework provides a standard way to build and deploy applications.

In simple terms, a software framework is a structure which you can use to build something. It allows you to use different components, communicate to external APIs and define the structure of your application.

A software framework allows you to:

1. **Reduce Complexity**: A software framework make it easier to deal with complex situations and break them into different chunks.
2. **Clean Code**: A software framework helps you to write clean and usable code.
3. **Testing and Debugging**: A software framework make testing easier and debugging fun.
4. **Coding Guidelines**: A software framework forces strict guideline to follow for your team mates that promote the consistent codebase with less bugs.
5. **Scaling**: A software framework helps you to scale on demand. No matter how fast you application grow, frameworks make it easier to scale on demand.

**Web Framework**

A web framework or web application framework is just another software framework which helps you to build Web Applications including web services and APIs. A web framework enables you to make components that are re-usable and perform daily tasks more easily.

### PHP Framework

A PHP framework is a set of PHP classes which help you build the web application with ease.

**What is Composer?**

Composer is an application-level package manager for the PHP programming language that provides a standard format for managing dependencies of PHP software and required libraries.

**What is Laravel?**

Laravel is one of the most popular PHP framework used across the globe to build web application ranging from small to big projects. Laravel is the choice of professional developers because of its performance, features, and scalability.

Laravel follows MVC (Model View Controller) structure which makes it easy to learn and rapidly prototype the web applications. Laravel takes the pain out of web application development by providing built-in features like authentication, mail, routing, sessions, and the list goes on.

# Key Features of Laravel Framework

Laravel framework possesses a large ecosystem which includes features like instant deployment, routing, ORM, DB query, Routing, Templating and list goes one.

# 1. Dependency Management

Dependency management is one of the best features of Laravel, understanding the functionality of the service container (IoC) is the core part to learn modern web applications. In Laravel, IoC (Inversion of Control) or Service Container is the most powerful tool to manage class dependencies.

Dependency Injection is a method to remove the hard-coded classes and injecting them using a tool like Composer.

# 2. Modularity

Modularity is the degree to which a web application’s components can be separated and recombined. You can split the business logic into different modules which all work together to make a web application functional.

Laravel is designed to make modular application, even Laravel itself is a collection of components. Using modular structure you can design and develop a large-scale enterprise application with ease. Laravel provides very simple guidelines to create modules or packages in Laravel.

# 3. Authentication

Authentication is an integral part of any modern web application. Writing authentication in other frameworks such as Codeigniter can take a lot of your development time. Laravel provides authentication out of the box, with running a simple command you can create a fully functional authentication system.

Laravel also provide handy documentation to implement your own authentication.

# 4. Caching

Caching is a technique to store data in a temporary storage location and can be retrieved quickly when needed. Mostly caching is used to increase the performance of the application. Laravel cache almost all data from view to routes. Which helps Laravel to decrease the processing time thus increasing the performance?

# 5. Routing

Routing in Laravel very easy to understand and much like the Ruby on Rails framework. Laravel routing can be used to create a restful application with ease. You can group routes, name them, apply filters to them and bind your model data to them.

Laravel routes can use to create search engines friendly URLs with great flexibility and control.

# 6. Restful Controllers

Laravel’s restful controllers enable you to separate logic behind serving the GET or POST requests. You can also create resource controllers which can be used to create CRUD easily. Later you can bind resource controller to route to serve all CRUD routes automatically.

# 7. Testing and Debugging

Laravel comes with PHPUnit out of the box to test your applications. Laravel was built with testing in mind thus support testing and debugging too much extent.

# 8. Template Engine

**Blade** is the templating engine of Laravel. Blade provides handful amount of helper function to format your data within views. Blade also implement the template inheritance which you can use to create complex layouts. All Blade templates use file extension .blade.php.

# 9. Database Query Builder

Laravel’s database query builder provides a convenient way to create database queries. It comes with tons of helper functions which you can use to filter down your data. You can easily implement complex queries using joins in Laravel.

Laravel’s query builder syntax is easy to understand and make writing database queries fun.

# 10. Eloquent ORM

Laravel is based on Eloquent ORM which provides support for almost all database engines. It works perfectly with MySQL and SQLite. Laravel provides a comprehensive documentation to all Eloquent functions.

# 11. File System

Laravel provides the support for multiple file systems. You can use a local file system or any cloud-based like Amazon S3. You can select which file system you want to use by changing the options in the config/filesystems.php file.

# 12. Security

Laravel provides an intuitive way to create secure web applications. Laravel store all passwords as hashed instead of plain text passwords. Laravel use BCrytp to hash passwords. Laravel provides security from SQL injection attacks and also escape all user inputs to avoid the injection of any script tags.

# 13. Artisan

Laravel’s command line tool is called Artisan. Laravel comes with dozens of pre-built commands which you can run on command line interface to perform tasks. You can avoid most repetitive task during development process using this tool.

# 14. Migration System

Laravel provides migration system just like Ruby on Rails to create database structures. You can use PHP language to write migration which will create your database structure instead of using SQL. You can create databases, tables, and indices using the migrations. If you want to change table column you don’t have to repeat creating table again instead you can run a new migration.

# 15. Envoy

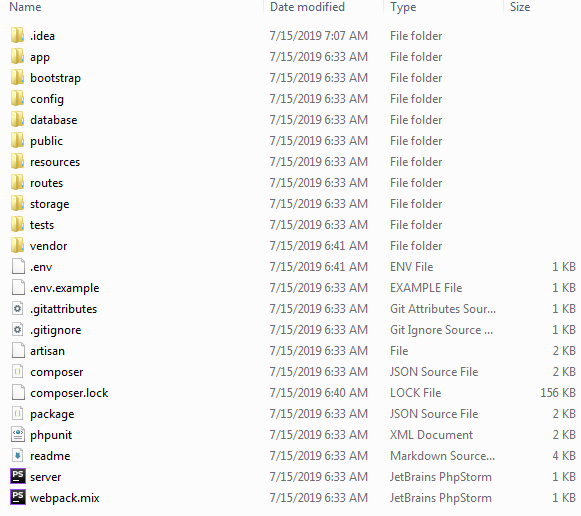
Using Envoy you can run the most common task on your remote servers from within your application. It also allows you to set up tasks for deployment.

# 16. Localization

Laravel enable you to create a multilingual application at the same time you are building your application. You can access the strings in different languages using trans() helper.

Understanding Laravel Directory Structure

This is how the default Laravel Directory Structure looks like



The Folders

**app:***app* is where the majority of your application code resides in. Model, Controllers , Middleware’s, Service Providers. This is the folder you will be working in the majority of your business Logic development.

**bootstrap** : This folder contains the files that Laravel Framework use to boot the framework when it starts.

**config :** This folder contains the different configurations files related to database, file-system, logging etc.  Each configuration file returns an array which denotes the configuration values the framework needs to use for your application.

**database :**This is where the database migration files and the seeds live.

**public :**This is the public facing directory. It contains the index.php file which is the first file to be instantiated by Laravel framework. It also contains the public facing assets such as css, images and js.

**resources :**This is where the non-PHP files that are needed for other scripts live. Views, language-files, Sass / Less and source javascript files live here.

**routes**: This is where all your application routes go in. HTTP web routes, console routes, api routes all go in here.

**storage:**Cache file, log files and other compiled system files live here.

**tests** :  This is where the application tests live. Unit test, Feature test, Automation Dusk tests all go here.

**vendor** :  This is where *composer*install its dependencies. This folder is git-ignored that means when you commit your source in git this folder is ignored. It is expected to download the dependencies via composer on your remote servers.

**The Files**

The root directory also contains the following files

**.env and .env.example**: There are the files that contains environment variables. These are the variables that you expect to be changes with each environment. Therefore .env file is git-ignored. You are suppose to have a separate *.env* file on each environment with different set of variables. .env.example is a template file provided to help you create a .env file

**artisan :**This file allows you to run the artisan command from command line on the project directory root.

**.gitignore and .gitattributes**: These are git configuration files.

**composer.json and composer.lock :** There are configuration files for Composer. Composer.json contains the project package dependencies and composer.lock is non-editable file and it locks dependencies of your project to a known state.

**package.json :**This is like composer.json for your front-end dependencies.

**phpunit.xml** : This is configuration file for PHPUnit, the tool Laravel uses for testing out of the box.

**readme.md** : File to give out basic information about the Laravel Framework.

**server.php** : is a backup server that tries to allow less-capable servers to still preview the Laravel application.

**webpack.mix.js** : Mix provides a clean, fluent API for defining some Webpack build steps for your Laravel application.