101550305
Advanced Web Application Development
UNIT – 1

- Introduction to Laravel
  - What is Larave?
  - What is Framework?
  - What is MVC Architecture?
  - Advantages of Laravel
  - Features of Laravel
  - Laravel Installation
  - Application Structure
- Route and Controllers

Developed By: Dr. Ashish Joshi (ISTAR, Vallabh Vidyanagar)

101550305
Advanced Web Application Development
UNIT – 1

- Introduction to Laravel
  - What is Larave?
  - What is Framework?
  - What is MVC Architecture?
  - Advantages of Laravel
  - Features of Laravel
  - Laravel Installation
  - Application Structure

- Route and Controllers

Developed By: Dr. Ashish Joshi (ISTAR, Vallabh Vidyanagar)

1.1 Introduction to Laravel

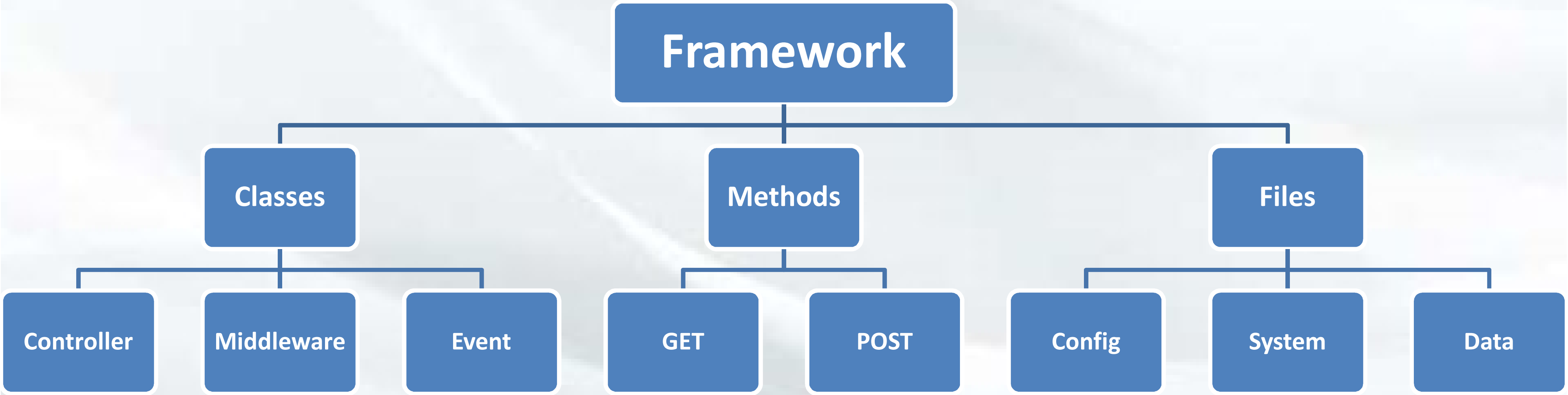
- Laravel is an MVC framework with bundles, migrations(a set of instructions that define the changes you want to make to your database schema), and Artisan CLI(Command Line Interface).
- Laravel offers a robust set of tools and an application architecture that incorporates many of the best features of frameworks like CodeIgniter, Yii, ASP.NET MVC, Ruby on Rails and others.
- Laravel is an Open Source framework. It has a very rich set of features which will boost the speed of Web Development.
- If you familiar with Core PHP and Advanced PHP, Laravel will make your task easier.
- It will save a lot time if you are planning to develop a website from scratch.
- Not only that, the website built in Laravel is also secure. It prevents the various attacks that can take place on websites.

1.2 What is Framework?

Laravel is a **PHP framework** that uses the **MVC architecture**.

What Framework?

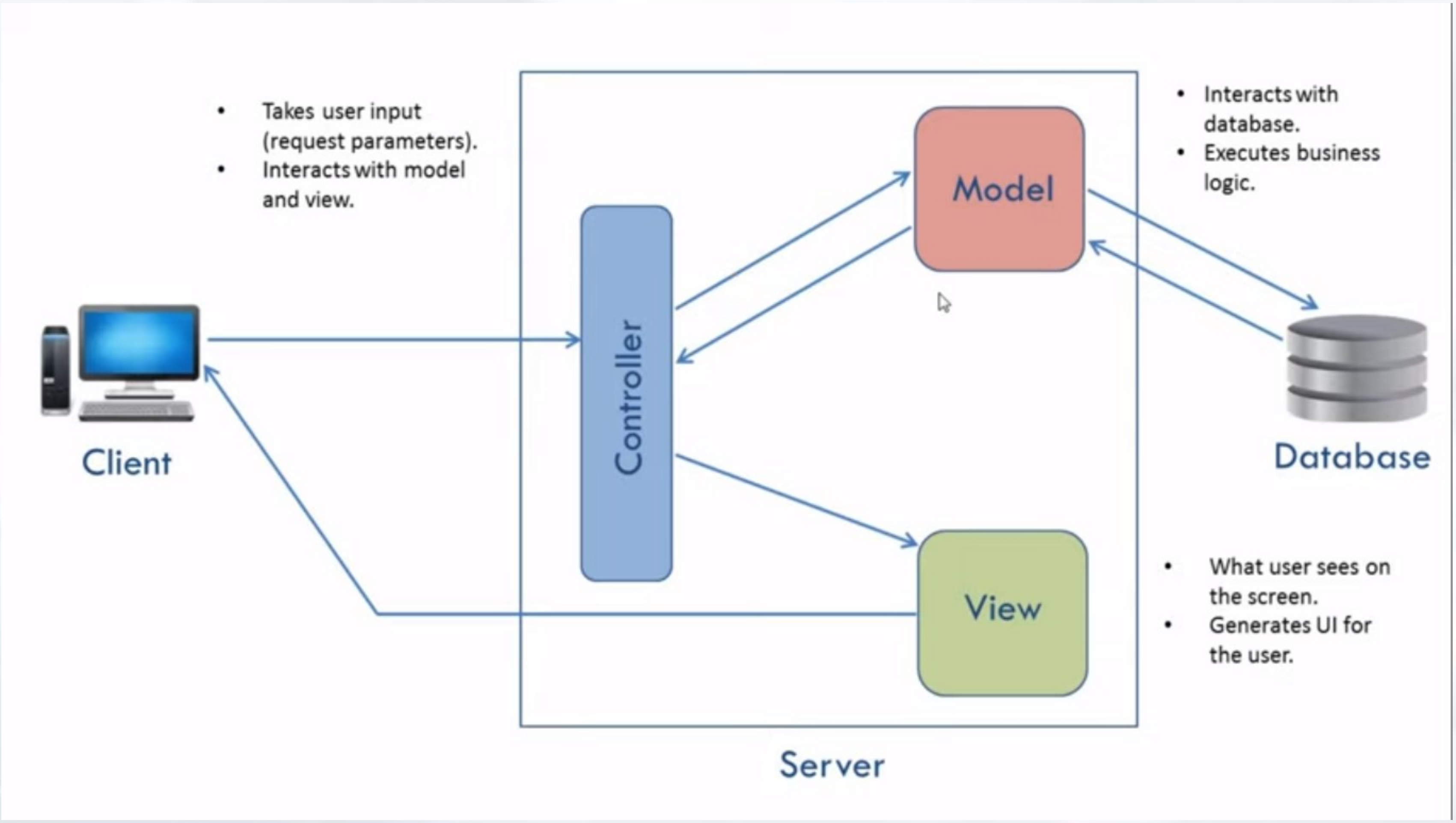
It is the collection of methods, classes, or files that the programmer uses, and they can also extend its functionality by using their code.



1.3 What is MVC Architecture

What is MVC Architecture?

It is the specific design pattern that the framework follows. Laravel is following the MVC architecture.



- Takes user input (request parameters).
- Interacts with model and view.
- Interacts with database.
- Executes business logic.
Model
Controller
Client
Database
View
- What user sees on the screen.
- Generates UI for the user.
Server

1.3 What is MVC Architecture

M: 'M' stands for **Model**. A model is a class that deals with a database. For example, if we have users in an application then we will have users model that deals with a database to query the table of users if we have users model, then we will also have a users table. We conclude from the example that the model is going to have a table for that specific model.

V: 'V' stands for **View**. A view is a class that deals with an HTML. Everything that we can see on the application in the browser is the view or the representation.

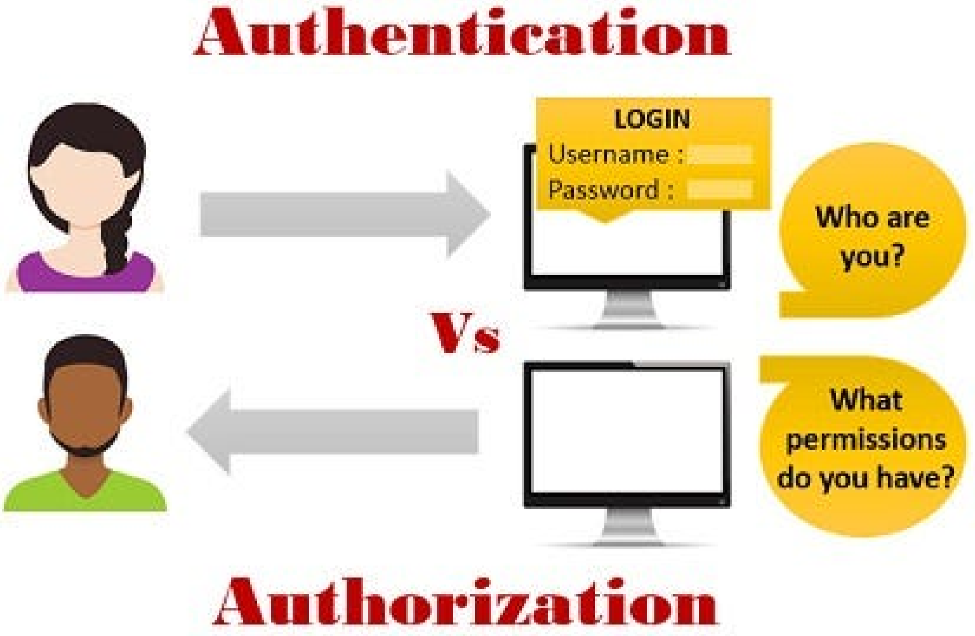
C: 'C' stands for **Controller**. A controller is the middle-man that deals with both model and view. A controller is the class that retrieves the data from the model and sends the data to the view class.

1.4 Advantages of Laravel

Advantages of Laravel

Creating authorization and authentication systems

Every owner of the web application makes sure that unauthorized users do not access secured or paid resources. It provides a simple way of implementing authentication. It also provides a simple way of organizing the authorization logic and control access to resources.



1.4 Advantages of Laravel

Integration with tools

Laravel is integrated with many tools that build a faster app. It is not only necessary to build the app but also to create a faster app. Integration with the caching back end is one of the major steps to improve the performance of a web app. Laravel is integrated with some popular cache back ends such as **Redis**, and **Memcached**. (cache is store and retrieve data quickly for future use)

Mail service integration

Laravel is integrated with the Mail Service. This service is used to send notifications to the user's emails. It provides a clean and simple API that allows you to send the email quickly through a local or cloud-based service of your choice.

1.4 Advantages of Laravel

Handling exception and configuration error

Handling exception and configuration errors are the major factors on the app's usability. The manners in which the software app handles the errors have a huge impact on the user's satisfaction and the app's usability. The organization does not want to lose their customers, so for them, Laravel is the best choice.

1.4 Advantages of Laravel

Automation testing work

Testing a product is very important to make sure that the software runs without any errors, bugs, and crashes. We know that automation testing is less time-consuming than manual testing, so automation testing is preferred over the manual testing. Laravel is developed with testing in mind.

Separation of business logic code from presentation code

The separation between business logic code and presentation code allows the HTML layout designers to change the look without interacting with the developers. A bug can be resolved by the developers faster if the separation is provided between the business logic code and presentation code. We know that Laravel follows the **MVC architecture**, so separation is already done.

1.4 Advantages of Laravel

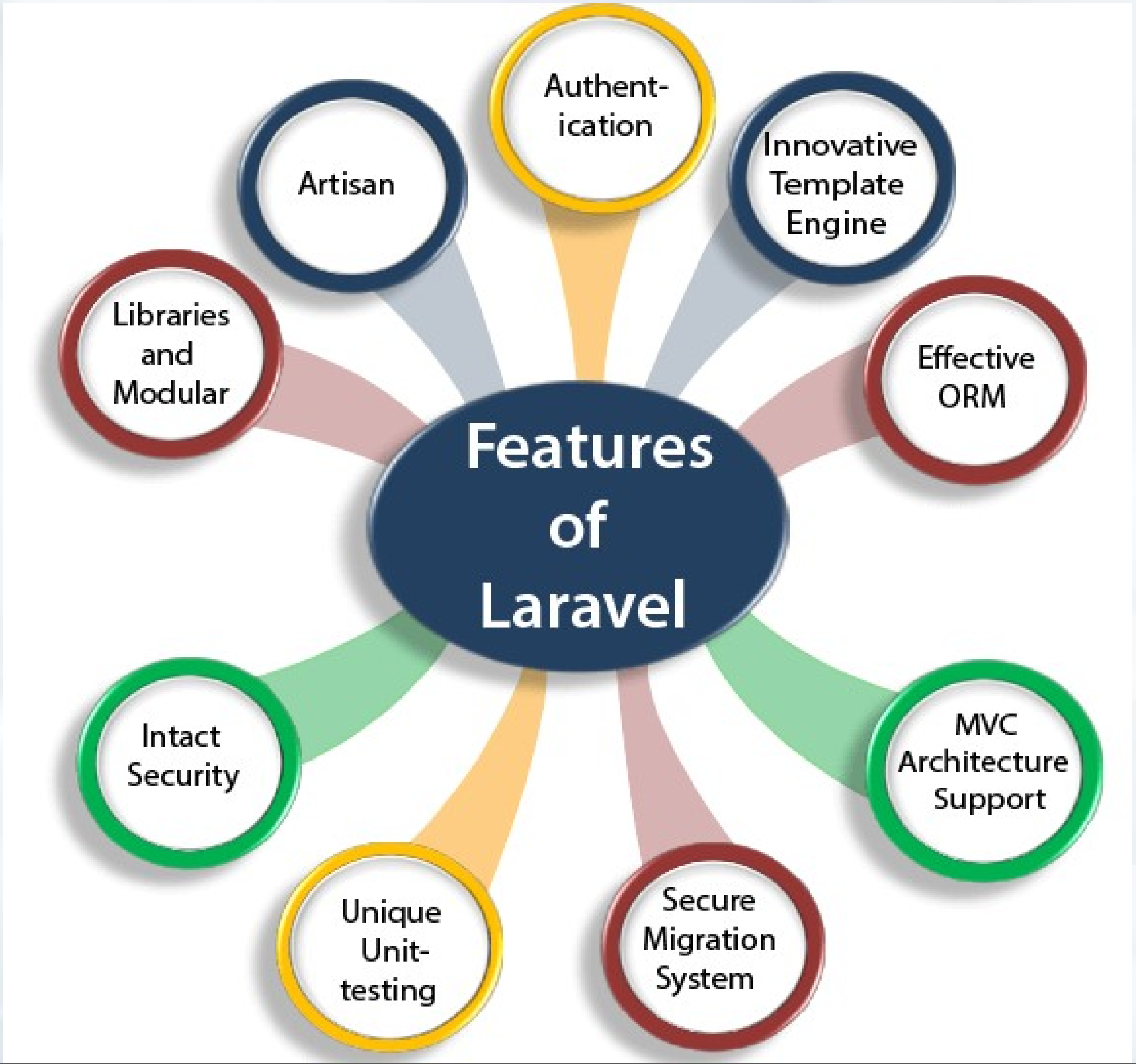
Fixing most common technical vulnerabilities

The **security vulnerability** is the most important example in web application development. An American organization, i.e., OWASP Foundation, defines the most important security vulnerabilities such as SQL injection, cross-site request forgery, cross-site scripting, etc. Developers need to consider these vulnerabilities and fix them before delivery. Laravel is a secure framework as it protects the web application against all the security vulnerabilities.

Scheduling tasks configuration and management

The web app requires some task scheduling mechanism to perform the tasks in time for example, when to send out the emails to the subscribers or when to clean up the database tables at the end of the day. To schedule the tasks, developers need first to create the **Cron entry** for each task, but **Laravel command scheduler** defines a command schedule which requires a single entry on the server.

1.5 Features of Laravel



1.5 Features of Laravel

Authentication

Authentication is the most important factor in a web application, and developers need to spend a lot of time writing the authentication code. Laravel makes a simpler authentication when Laravel is updated to Laravel 5. Laravel contains an inbuilt authentication system, you only need to configure models, views, and controllers to make the application work.

Innovative Template Engine

Laravel provides an innovative template engine which allows the developers to create the dynamic website. The available widgets in Laravel can be used to create solid structures for an application.

1.5 Features of Laravel

Effective ORM

Laravel contains an inbuilt ORM with easy PHP Active Record implementation. An effective ORM allows the developers to query the database tables by using the simple PHP syntax without writing any SQL code. It provides easy integration between the developers and database tables by giving each of the tables with their corresponding models.

Object-relational mapping (ORM) is a technique that connects code with relational databases. It uses metadata descriptors to create a layer between object-oriented program (OOP) code and the database.

MVC Architecture Support

Laravel supports MVC architecture. It provides faster development process as in MVC; one programmer can work on the view while other is working on the controller to create the business logic for the web application. It provides multiple views for a model, and code duplication is also avoided as it separates the business logic from the presentation logic.

1.5 Features of Laravel

Secure Migration System

Laravel framework can expand the database without allowing the developers to put much effort every time to make changes, and the migration process of Laravel is very secure and full-proof. In the whole process, **php code** is used rather than **SQL code**.

Laravel Migration is a set of instructions that define the changes you want to make to your database schema. These changes can include creating new tables, altering existing tables, adding or modifying columns, and seeding the database with initial data.

Unique Unit-testing

Laravel provides a unique unit-testing. Laravel framework can run several test cases to check whether the changes harm the web app or not. In Laravel, developers can also write the test cases in their own code.

1.5 Features of Laravel

Intact Security

Application security is one of the most important factors in web application development. While developing an application, a programmer needs to take effective ways to secure the application. Laravel has an inbuilt web application security, i.e., it itself takes care of the security of an application. It uses "Bcrypt Hashing Algorithm" to generate the salted password means that the password is saved as an encrypted password in a database, not in the form of a plain text.

1.5 Features of Laravel

Libraries and Modular

Laravel is very popular as some Object-oriented libraries, and pre-installed libraries are added in this framework, these pre-installed libraries are not added in other **php frameworks**. One of the most popular libraries is an **authentication library** that contains some useful features such as password reset, monitoring active users, Bcrypt hashing, and CSRF(Cross-site request forgeries) protection.

Note: Cross-site request forgeries are a type of malicious exploit whereby unauthorized commands are performed on behalf of an authenticated user.

1.5 Features of Laravel

Artisan

Laravel framework provides a built-in tool for a command-line known as **Artisan** that performs the repetitive programming tasks that do not allow the php developers to perform manually. These artisans can also be used to create the skeleton code, database structure, and their migration, so it makes it easy to manage the database of the system. It also generates the MVC files through the command line. Artisan also allows the developers to create their own commands.

Composer

Composer is a tool which includes all the dependencies and libraries. It allows a user to create a project with respect to the mentioned framework (for example, those used in Laravel installation). Third party libraries can be installed easily with help of composer.

All the dependencies are noted in **composer.json** file which is placed in the source folder.

1.6 Installation of Laravel

1. Install Xampp
2. download composer from https://getcomposer.org/download/
3. install composer
4. check with command line -> enter command "composer" enter
5. install laravel->composer global require laravel/installer->restart cmd

Note: Modify php.ini (;extension=zip). Remove semicolon

6. select proper folder & create app -> laravel new <app-name>
7. change directory -> cd <app-name>
8. run app -> php artisan serve

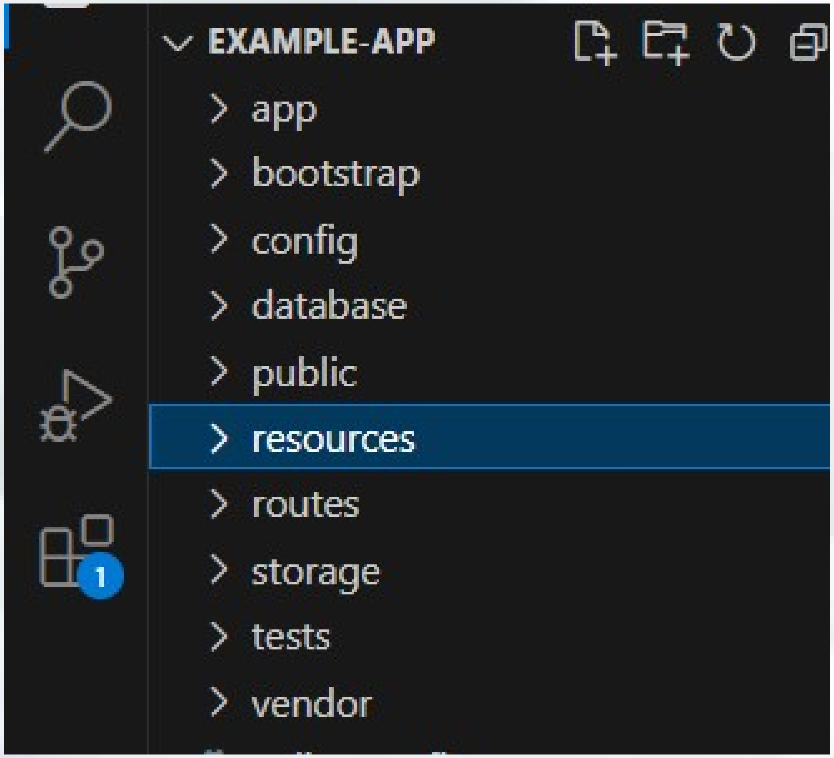
Note: if internal server error, execute following command(start the mysql server)

```
php artisan migrate
php artisan migrate:status
```

1.7 Application Structure in Laravel

Introduction

The default Laravel application structure is intended to provide a great starting point for both large and small applications. But you are free to organize your application however you like. Laravel imposes almost no restrictions on where any given class is located - as long as Composer can autoload the class.



EXAMPLE-APP
- app
- bootstrap
- config
- database
- public
- resources
- routes
- storage
- tests
- vendor

1.7 Application Structure in Laravel

The Root Directory

The App Directory

The app directory contains the core code of your application. We'll explore this directory in more detail soon; however, almost all of the classes in your application will be in this directory.

The Bootstrap Directory

The bootstrap directory contains the app.php file which bootstraps the framework. This directory also houses a cache directory which contains framework generated files for performance optimization such as the route and services cache files.

The Config Directory

The config directory, as the name implies, contains all of your application's configuration files. It's a great idea to read through all of these files and familiarize yourself with all of the options available to you.

1.7 Application Structure in Laravel

The Database Directory

The database directory contains your database migrations, model factories, and seeds. If you wish, you may also use this directory to hold an SQLite database.

The Public Directory

The public directory contains the index.php file, which is the entry point for all requests entering your application and configures autoloading. This directory also houses your assets such as images, JavaScript, and CSS.

The Resources Directory

The resources directory contains your views as well as your raw, un-compiled assets such as CSS or JavaScript.

1.7 Application Structure in Laravel

The Routes Directory

The routes directory contains all of the route definitions for your application. By default, two route files are included with Laravel: web.php and console.php.

The web.php file contains routes that Laravel places in the web middleware group, which provides session state, CSRF protection, and cookie encryption. If your application does not offer a stateless, RESTful API then all your routes will most likely be defined in the web.php file.

The console.php file is where you may define all of your closure based console commands. Each closure is bound to a command instance allowing a simple approach to interacting with each command's IO methods. Even though this file does not define HTTP routes, it defines console based entry points (routes) into your application. You may also schedule tasks in the console.php file.

1.7 Application Structure in Laravel

The Storage Directory

The storage directory contains your logs, compiled Blade templates, file based sessions, file caches, and other files generated by the framework. This directory is segregated into app, framework, and logs directories. The app directory may be used to store any files generated by your application. The framework directory is used to store framework generated files and caches. Finally, the logs directory contains your application's log files.

The storage/app/public directory may be used to store user-generated files, such as profile avatars, that should be publicly accessible. You should create a symbolic link at public/storage which points to this directory. You may create the link using the php artisan storage:link Artisan command.

1.7 Application Structure in Laravel

The Tests Directory

The tests directory contains your automated tests. Example Pest or PHPUnit unit tests and feature tests are provided out of the box. Each test class should be suffixed with the word Test. You may run your tests using the /vendor/bin/pest or /vendor/bin/phpunit commands. Or, if you would like a more detailed and beautiful representation of your test results, you may run your tests using the php artisan test Artisan command.

The Vendor Directory

The vendor directory contains your Composer dependencies.