Laravel

Part 1: Laravel Installation

Working with Laravel, we have to configure our working environment first. We have to follow the following steps to make our environment ready to start Laravel.

Step one is Working Environment

01. First install the composer from https://getcomposer.org/download/

02. As a localhost install Laragon from: https://laragon.org/download/

03. For code editor I am using VS Code Editor from: https://code.visualstudio.com/

04. For Database I am using phpMyAdmin-5.2.0-all-languages from: https://www.phpmyadmin.net/

Step two Working with Laravel:

There are two way to install Laravel.

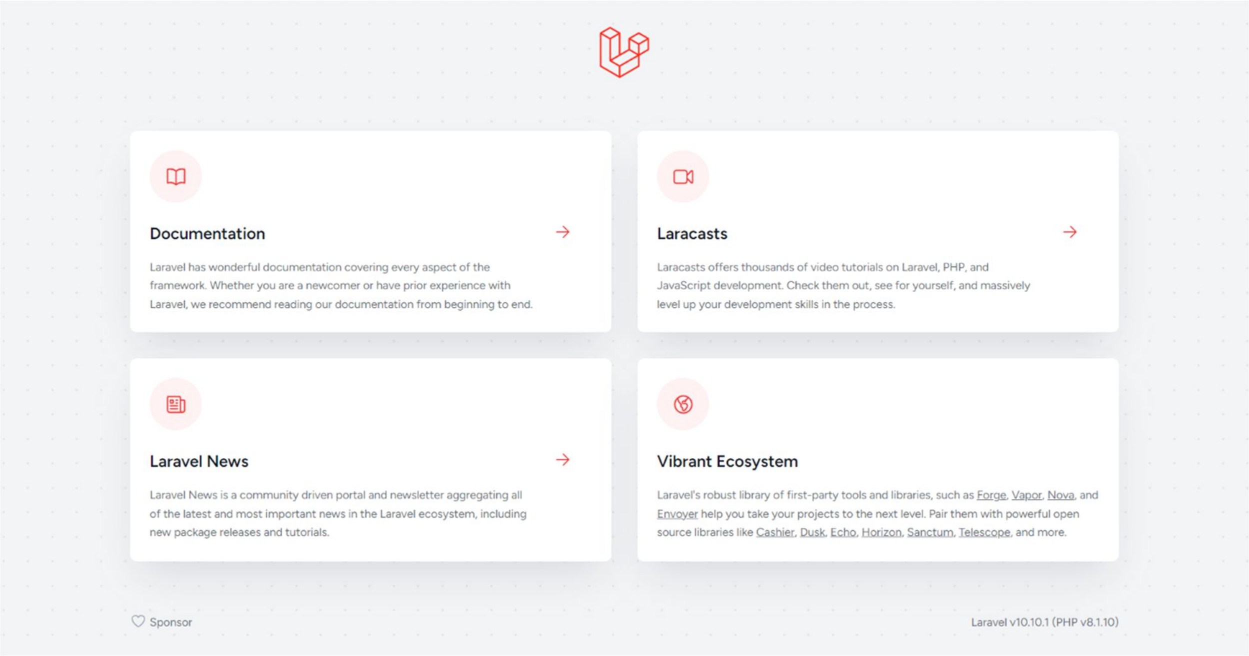
# via the Composer create-project command:

```
composer create-project laravel/laravel example-app
```

# globally installing the Laravel installer via Composer:

```
composer global require laravel/installer

laravel new example-app
```



Documentation

Laravel has wonderful documentation covering every aspect of the framework. Whether you are a newcomer or have prior experience with Laravel, we recommend reading our documentation from beginning to end.

Laracasts

Laracasts offers thousands of video tutorials on Laravel, PHP, and JavaScript development. Check them out, see for yourself, and massively level up your development skills in the process.

Laravel News

Laravel News is a community driven portal and newsletter aggregating all of the latest and most important news in the Laravel ecosystem, including new package releases and tutorials.

Vibrant Ecosystem

Laravel's robust library of first-party tools and libraries, such as Forge, Vapor, Nova, and Envoyer help you take your projects to the next level. Pair them with powerful open source libraries like Cashier, Dusk, Echo, Horizon, Sanctum, Telescope, and more.

♡ Sponsor

Laravel v10.10.1 (PHP v8.1.10)

Part 2: Laravel Folder Structure

In a Laravel project, the purpose of each of the following folders is as follows:

app: This folder contains the core application code. It includes the models, controllers, and other classes that define the business logic and functionality of the application.

bootstrap: The bootstrap folder contains the application's bootstrap files, which are responsible for initializing the framework and its components. It includes files for autoloading, error handling, and configuration loading.

config: The config folder holds configuration files for various aspects of the Laravel application. It includes files for database configuration, mail settings, service providers, and more. You can modify these files to customize the behavior of your application.

database: This folder is used to manage the application's database-related files. It contains migrations, which are used to create and modify database tables, as well as seeders, which allow you to populate the database with initial data.

public: The public folder is the web root of your Laravel application. It contains the entry point for incoming HTTP requests, the index.php file, as well as assets like CSS, JavaScript, and image files that can be accessed directly by users.

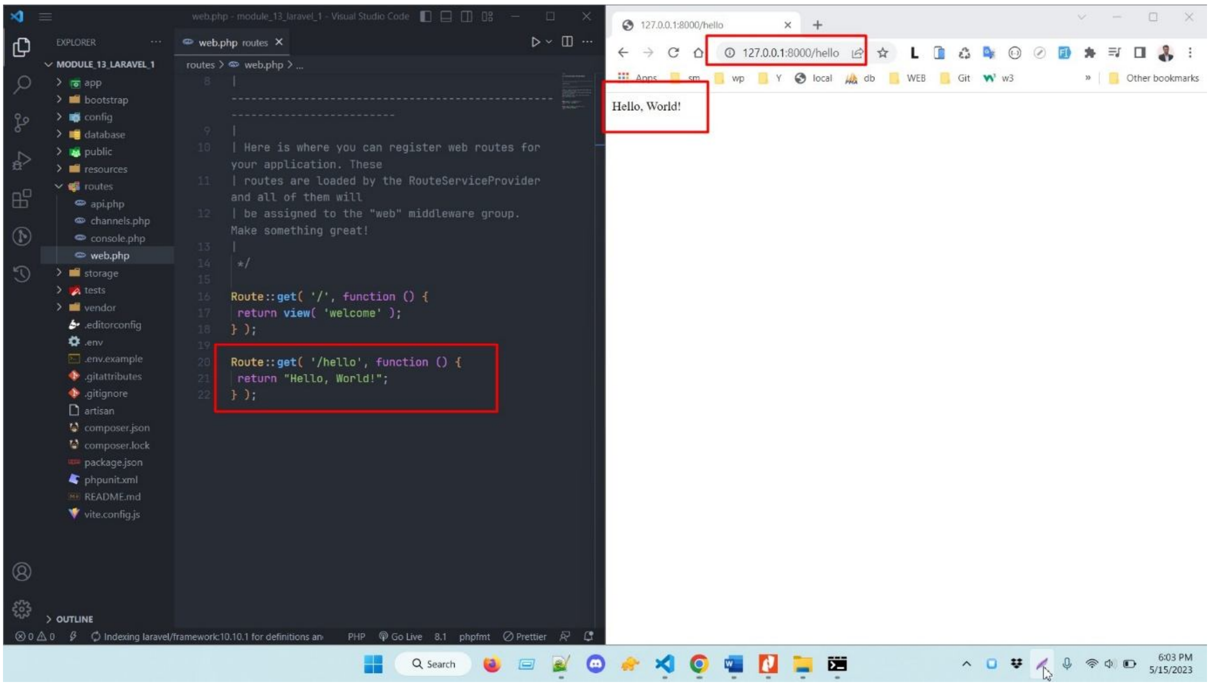
resources: The resources folder holds various resource files used by the application, such as views, language files, and assets that need to be compiled, such as SCSS or JavaScript files. Views are stored in the views subfolder and are responsible for rendering HTML templates.

routes: The routes folder contains route definition files that determine how the application responds to different HTTP requests. It includes web.php for handling web requests and api.php for handling API requests. You can define routes, middleware, and route groups in these files.

storage: The storage folder is used for storing various temporary and long-term storage files generated by the application. This includes logs, session files, cache files, and uploaded files. It is also the default location for Laravel's file-based caching and session storage.

tests: The tests folder contains the automated tests for the application. It includes test cases that cover different parts of the application's functionality. Laravel provides a testing framework that allows you to write and run tests easily.

vendor: The vendor folder is created and managed by Composer, a dependency management tool used by Laravel. It contains all the third-party libraries and packages that your application depends on. Composer installs and updates these dependencies based on the requirements defined in the composer.json file.



```
|
---------------------------------------------------------------------------
|
| Here is where you can register web routes for your application. These
| routes are loaded by the RouteServiceProvider and all of them will
| be assigned to the "web" middleware group. Make something great!
|
 */

Route::get( '/', function () {
 return view( 'welcome' );
} );

Route::get( '/hello', function () {
 return "Hello, World!";
} );
```

127.0.0.1:8000/hello

```
Hello, World!
```