

Web development with php & laravel

Ostad.app

Assignment of Module - 13

Assignment Title: Laravel Installation and Folder Structure Assignment

Name : MD. RAQIBUL ISLAM HOWLADER

Batch No : 01

Ostad login No : 01681533509

WhatsApp No : 01782333171

Email : raqibulislam907@gmail.com

Submit date: 15-05-2023

The Root Directory

App:

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.

Bootstrap:

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. You should not typically need to modify any files within this directory.

Config:

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.

Database:

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.

Public:

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.

Resources:

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

Routes:

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

The web.php file contains routes that the RouteServiceProvider 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 api.php file contains routes that the RouteServiceProvider places in the api middleware group. These routes are intended to be stateless, so requests entering the application through these routes are intended to be authenticated via tokens and will not have access to session state.

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.

The channels.php file is where you may register all of the event broadcasting channels that your application supports.

Storage:

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.

Tests:

The tests directory contains your automated tests. Example 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 phpunit or php 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.

Vendor:

The vendor directory contains your Composer dependencies.

Description of the steps which i took to complete the assignment:

1. At first took open the laravel Official documentation on browser.
2. According the documentation, Installed composer in my customs file location into Xampp Folder. Then Installed Laravel in the same directory.
3. Started to laravel project by project name like laravel documentation.
4. At present, Laravel project run smoothly and fast by using Xampp control panal.
5. It seems to me a easy method.