Part 1: Laravel Installation

```
PS D:\xampp\htdocs\www\ostad.prac\module-13> composer create-project laravel/laravel ost-ass-mod13.wplara
Creating a "laravel/laravel" project at "./ost-ass-mod13.wplara"
Info from https://repo.packagist.org: #StandWithUkraine
Installing laravel/laravel (v10.2.0)
  - Downloading laravel/laravel (v10.2.0)
  - Installing laravel/laravel (v10.2.0): Extracting archive
Created project in D:\xampp\htdocs\www\ostad.prac\module-13\ost-ass-mod13.wplara
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 107 installs, 0 updates, 0 removals
  - Locking brick/math (0.11.0)
  - Locking dflydev/dot-access-data (v3.0.2)
  - Locking doctrine/inflector (2.0.6)
```

```
> Windows PowerShell

  laravel/sail ............................................................... DONE
  laravel/sanctum ............................................................ DONE
  laravel/tinker ............................................................. DONE
  nesbot/carbon .............................................................. DONE
  nunomaduro/collision ....................................................... DONE
  nunomaduro/termwind ........................................................ DONE
  spatie/laravel-ignition .................................................... DONE

81 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi --force

   INFO  No publishable resources for tag [laravel-assets].

> @php artisan key:generate --ansi

   INFO  Application key set successfully.

PS D:\xampp\htdocs\www\ostad.prac\module-13>
```
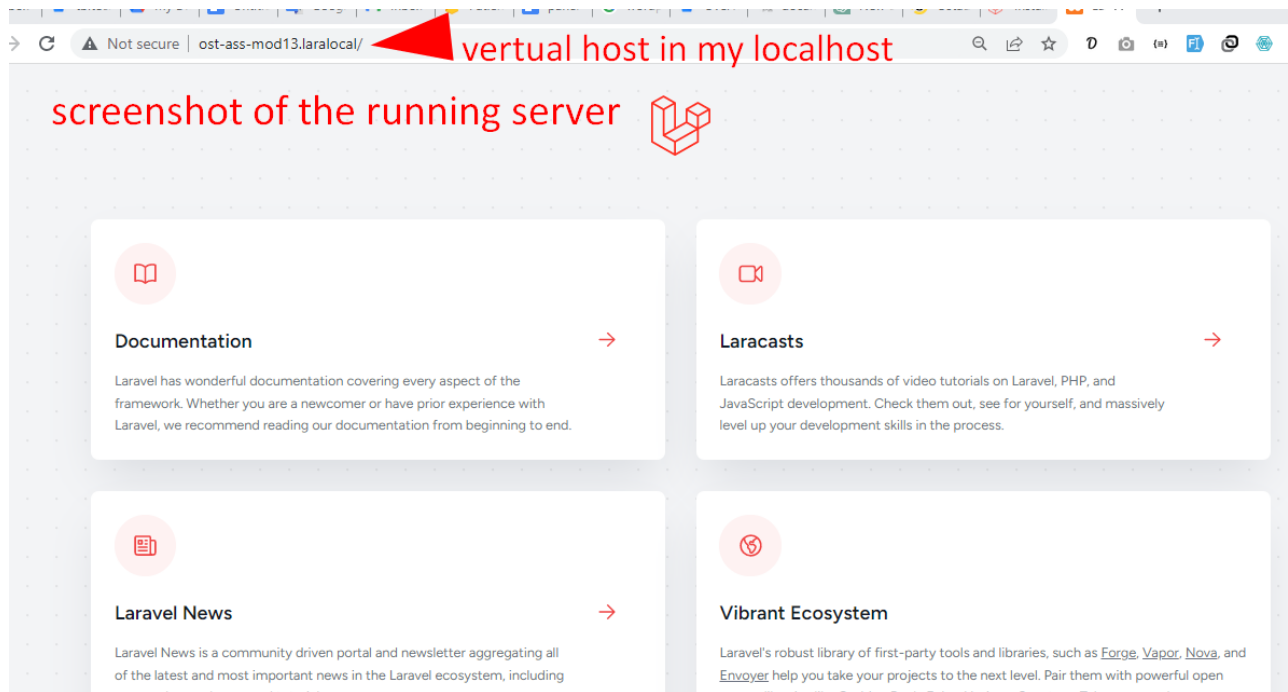
Not secure | ost-ass-mod13.laralocal/ ← vertual host in my localhost

screenshot of the running server

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

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

Part 2: Laravel Folder Structure

Description of the purpose of each of the folders in a typical Laravel project:

app: This folder contains the core application files. It includes the models, controllers, and other PHP classes that define the application's business logic and functionality.

bootstrap: The bootstrap folder contains files responsible for bootstrapping the Laravel framework and setting up the application. It includes the app.php file that initializes the application, as well as other configuration files related to environment setup.

config: The config folder contains various configuration files for the Laravel application. It includes files like app.php, database.php, mail.php, and others, where you can define settings and configurations specific to your application.

database: The database folder contains files related to database management. It includes migrations, which are used to create and modify database tables, as well as seeders to populate the database with sample data.

public: The public folder is the web root of your Laravel application. It contains the index.php file, which serves as the entry point for all HTTP requests. Additionally, it typically contains assets such as CSS, JavaScript, and image files that are directly accessible by the client.

resources: The resources folder contains files related to frontend assets and views. It includes subdirectories for views, language files, and assets like CSS, JavaScript, and images. The views directory holds the Blade templates used for generating HTML responses.

routes: The routes folder contains route definitions for the application. It includes web.php and api.php files by default, where you can define routes for web pages and API endpoints, respectively. This folder also includes a routes directory where you can organize routes into separate files if needed.

storage: The storage folder is used for storing application-specific files such as logs, cached views, uploaded files, and other temporary files. It includes subdirectories like app, framework, logs, and others for organizing different types of files.

tests: The tests folder is used for writing automated tests for your Laravel application. It includes subdirectories for unit tests, feature tests, and other types of tests. Laravel provides tools and frameworks like PHPUnit for testing your application's functionality.

vendor: The vendor folder is created and managed by Composer. It contains all the dependencies installed via Composer for your Laravel project. This folder includes the source code of external packages and libraries that your application depends on.

These folders collectively provide the structure for organizing and managing the different aspects of a Laravel application, separating concerns, and facilitating development, configuration, and maintenance.

New route in my Laravel project that displays a simple "Hello, World!" message
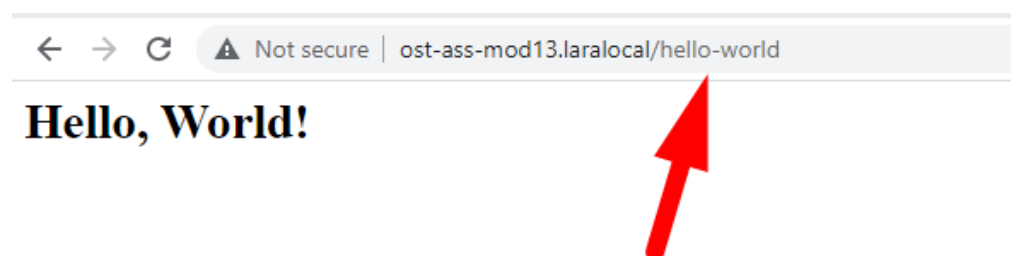
```php
6   |--------------------------------------------------------
7   | Web Routes
8   |--------------------------------------------------------
9   |
10  | Here is where you can register web routes for your
11  | routes are loaded by the RouteServiceProvider and a
12  | be assigned to the "web" middleware group. Make sor
13  |
14  */

16  Route::get('/', function () {
17      return view('welcome');
18  });

20  Route::get('hello-world/', function () {
21      echo "<h1>Hello, World!</h1>";
22  });
23
```