

# ASSIGNMENT 13

DOCUMENT

## Part-I Laravel Installation

### Steps:

#### 1. Composer Installation:

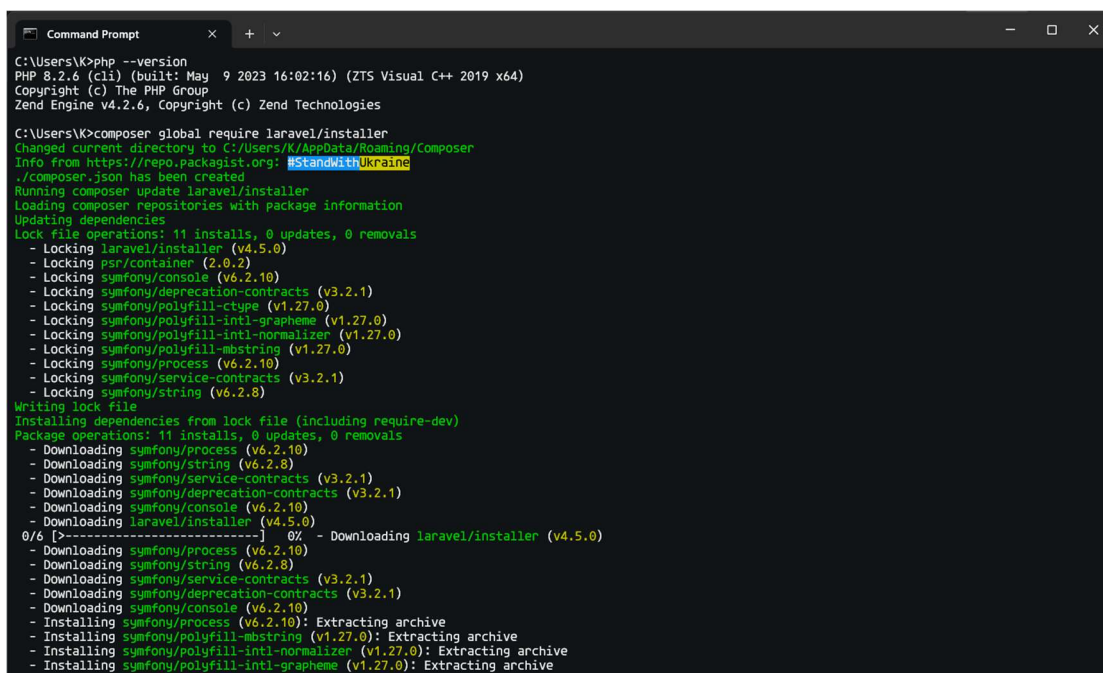
:

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php composer-setup.php
```

Composer could also be installed using an installer executable file on Windows operating system.

#### 2. Laravel Installation using composer:

Composer require Laravel/installer



```
Command Prompt
C:\Users\K>php --version
PHP 8.2.6 (cli) (built: May  9 2023 16:02:16) (ZTS Visual C++ 2019 x64)
Copyright (c) The PHP Group
Zend Engine v4.2.6, Copyright (c) Zend Technologies

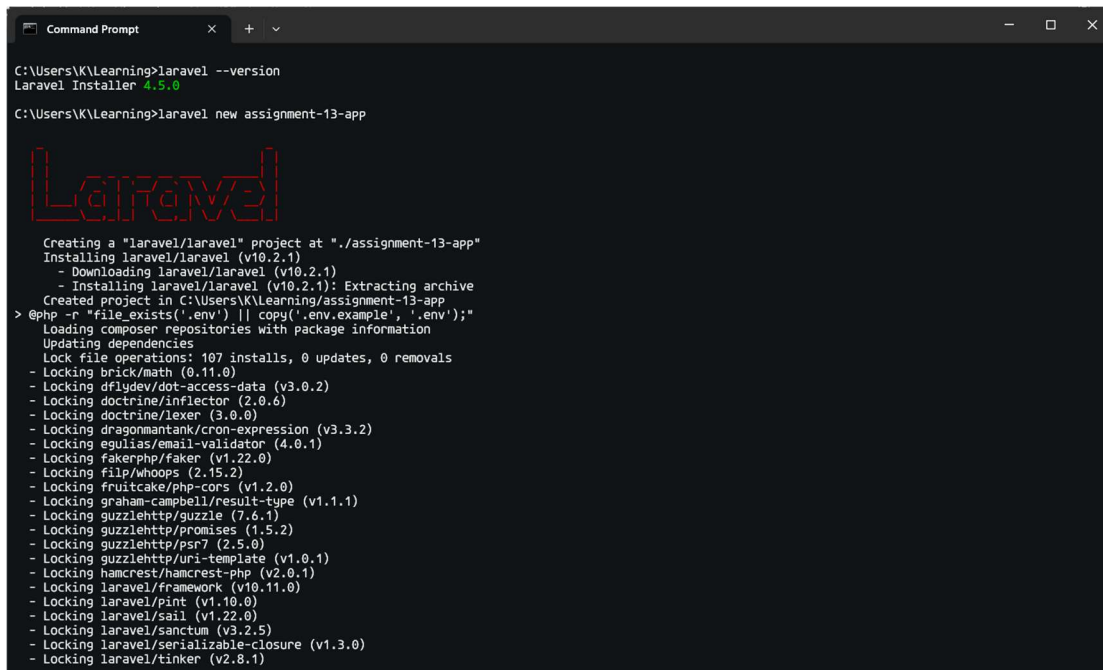
C:\Users\K>composer global require laravel/installer
Changed current directory to C:/Users/K/AppData/Roaming/Composer
Info from https://repo.packagist.org: #StandWithUkraine
./composer.json has been created
Running composer update laravel/installer
Loading composer repositories with package information
Updating dependencies
Lock file operations: 11 installs, 0 updates, 0 removals
- Locking laravel/installer (v4.5.0)
- Locking psr/container (2.0.2)
- Locking symfony/console (v6.2.10)
- Locking symfony/deprecation-contracts (v3.2.1)
- Locking symfony/polyfill-ctype (v1.27.0)
- Locking symfony/polyfill-intl-grapheme (v1.27.0)
- Locking symfony/polyfill-intl-normalizer (v1.27.0)
- Locking symfony/polyfill-mbstring (v1.27.0)
- Locking symfony/process (v6.2.10)
- Locking symfony/service-contracts (v3.2.1)
- Locking symfony/string (v6.2.8)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 11 installs, 0 updates, 0 removals
- Downloading symfony/process (v6.2.10)
- Downloading symfony/string (v6.2.8)
- Downloading symfony/service-contracts (v3.2.1)
- Downloading symfony/deprecation-contracts (v3.2.1)
- Downloading symfony/console (v6.2.10)
- Downloading laravel/installer (v4.5.0)
0/6 [>-----] 0% - Downloading laravel/installer (v4.5.0)
- Downloading symfony/process (v6.2.10)
- Downloading symfony/string (v6.2.8)
- Downloading symfony/service-contracts (v3.2.1)
- Downloading symfony/deprecation-contracts (v3.2.1)
- Downloading symfony/console (v6.2.10)
- Installing symfony/process (v6.2.10): Extracting archive
- Installing symfony/polyfill-mbstring (v1.27.0): Extracting archive
- Installing symfony/polyfill-intl-normalizer (v1.27.0): Extracting archive
- Installing symfony/polyfill-intl-grapheme (v1.27.0): Extracting archive
```

### 3. Verify that Laravel is installed and working correctly:

```
Laravel --version
```

### 4. Creating a new Laravel project:

```
Laravel new assignment-13-app
```



```
Command Prompt
C:\Users\K\Learning>laravel --version
Laravel Installer 4.5.0

C:\Users\K\Learning>laravel new assignment-13-app

Laravel

Creating a "laravel/laravel" project at "./assignment-13-app"
Installing laravel/laravel (v10.2.1)
- Downloading laravel/laravel (v10.2.1)
- Installing laravel/laravel (v10.2.1): Extracting archive
Created project in C:\Users\K\Learning\assignment-13-app
> @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)
- Locking doctrine/lexer (3.0.0)
- Locking dragonmantank/cron-expression (v3.3.2)
- Locking egulias/email-validator (4.0.1)
- Locking fakerphp/faker (v1.22.0)
- Locking filp/whoops (2.15.2)
- Locking fruitcake/php-cors (v1.2.0)
- Locking graham-campbell/result-type (v1.1.1)
- Locking guzzlehttp/guzzle (7.6.1)
- Locking guzzlehttp/promises (1.5.2)
- Locking guzzlehttp/psr7 (2.5.0)
- Locking guzzlehttp/uri-template (v1.0.1)
- Locking hamcrest/hamcrest-php (v2.0.1)
- Locking laravel/framework (v10.11.0)
- Locking laravel/pint (v1.10.0)
- Locking laravel/sail (v1.22.0)
- Locking laravel/sanctum (v3.2.5)
- Locking laravel/serializable-closure (v1.3.0)
- Locking laravel/tinker (v2.8.1)
```

```
Command Prompt
66/107 [=====>-----] 61%
67/107 [=====>-----] 62%
70/107 [=====>-----] 65%
73/107 [=====>-----] 68%
76/107 [=====>-----] 71%
82/107 [=====>-----] 76%
87/107 [=====>-----] 81%
91/107 [=====>-----] 85%
95/107 [=====>-----] 88%
98/107 [=====>-----] 91%
104/107 [=====>-----] 97%
106/107 [=====>-----] 99%
107/107 [=====] 100%

71 package suggestions were added by new dependencies, use `composer suggest` to see details.
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi

[INFO] Discovering packages.

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].

No security vulnerability advisories found
> @php artisan key:generate --ansi

[INFO] Application key set successfully.
[INFO] Application ready! Build something amazing.

C:\Users\K\Learning>
```

## 5. Running the development server:

```
cd assignment-13-app
php artisan serve
```

```
Command Prompt  Command Prompt - php artis
C:\Users\K\Learning\assignment-13-app>php artisan about

Environment .....
Application Name ..... Laravel
Laravel Version ..... 10.11.0
PHP Version ..... 8.2.6
Composer Version ..... 2.5.5
Environment ..... local
Debug Mode ..... ENABLED
URL ..... localhost
Maintenance Mode ..... OFF

Cache .....
Config ..... NOT CACHED
Events ..... NOT CACHED
Routes ..... NOT CACHED
Views ..... NOT CACHED

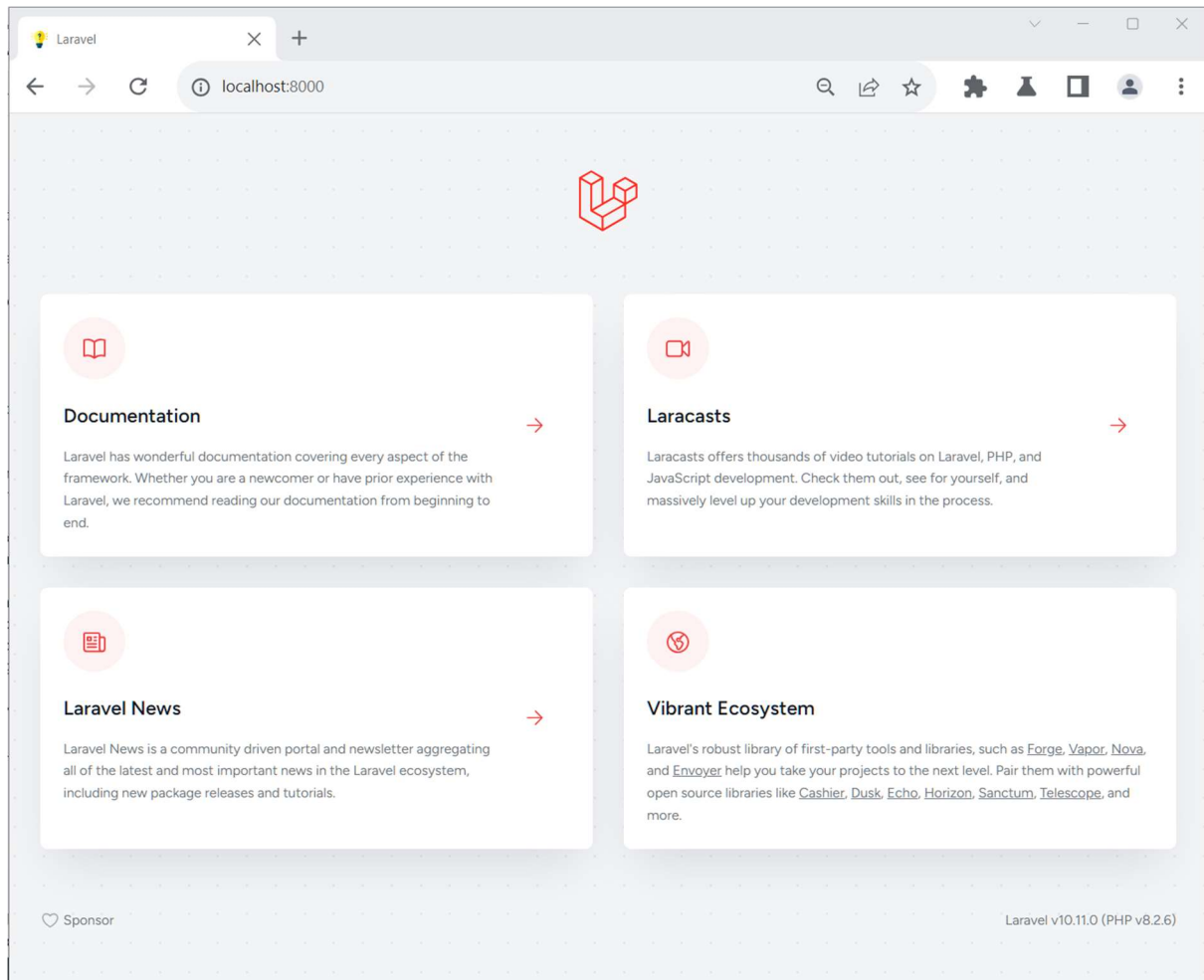
Drivers .....
Broadcasting ..... log
Cache ..... file
Database ..... mysql
Logs ..... stack / single
Mail ..... smtp
Queue ..... sync
Session ..... file

C:\Users\K\Learning\assignment-13-app>php artisan serve

[INFO] Server running on [http://127.0.0.1:8000].

Press Ctrl+C to stop the server
```

## Screenshot of development server running:



## Part-II

### Laravel Folder Structure

#### App Directory

The app directory contains the core code of the application. almost all of the classes in the application will be in this directory.

##### The Http Directory

The Http directory contains the controllers, middleware, and form requests. Almost all of the logic to handle requests entering the application will be placed in this directory.

##### The Models Directory

The Models directory contains all of the Eloquent model classes. The Eloquent ORM included with Laravel provides a beautiful, simple ActiveRecord implementation for working with the database. Each database table has a corresponding "Model" which is used to interact with that table. Models allow you to query for data in the tables, as well as insert new records into the table.

#### Bootstrap Directory

The bootstrap directory contains the `app.php` file which bootstraps the framework.

#### Config Directory

The config directory, as the name implies, contains all of the application's configuration files.

#### Database Directory

The database directory contains the database migrations, model factories, and seeds. Seeding is a process of programmatically populating the database with dummy data described in factory.

#### Public Directory

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

## Resources Directory

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

## Routes Directory

The routes directory contains all of the route definitions for the 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 the application does not offer a stateless, RESTful API then all the 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 the 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 the application.

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

## Storage Directory

The storage directory contains the 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 the application. The framework directory is used to store framework generated files and caches. Finally, the logs directory contains the 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 Directory

The tests directory contains the 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 the tests using the `phpunit` or `php vendor/bin/phpunit` commands. Or, if you would like a more detailed and beautiful representation of the test results, you may run the tests using the `php artisan test` Artisan command.

## Vendor Directory

The vendor directory contains the Composer dependencies.

## Creating a new route in the Laravel project that displays a simple “Hello, World!” message:

File: ‘`routes/web.php`’ adding the following code:

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

## Screenshot of the running route:

