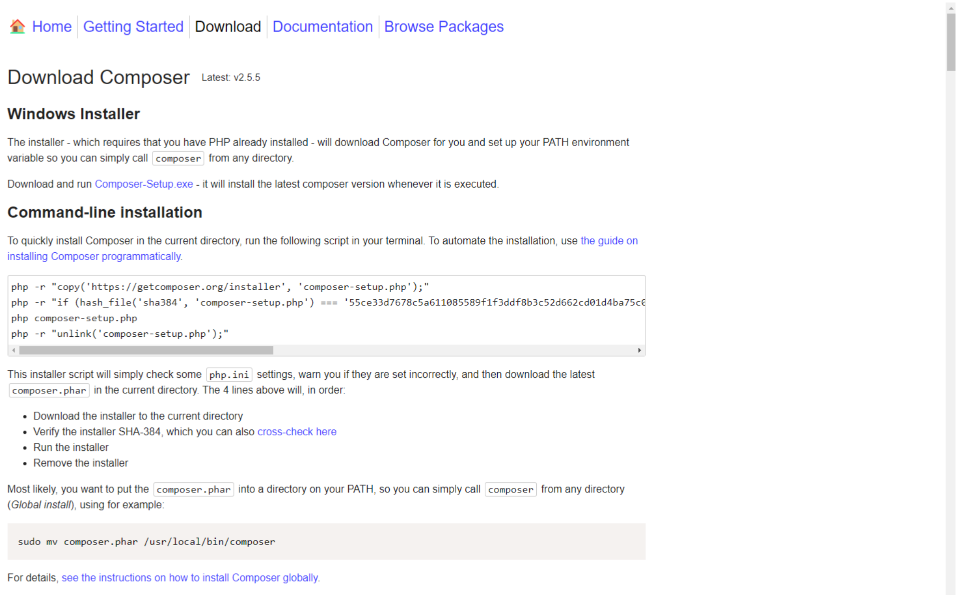
PART 1:

Step 1: Download and Install Composer. (Screenshot added)



Home | Getting Started | Download | Documentation | Browse Packages

Download Composer Latest: v2.5.5

Windows Installer

The installer - which requires that you have PHP already installed - will download Composer for you and set up your PATH environment variable so you can simply call `composer` from any directory.

Download and run Composer-Setup.exe - it will install the latest composer version whenever it is executed.

Command-line installation

To quickly install Composer in the current directory, run the following script in your terminal. To automate the installation, use the guide on installing Composer programmatically.

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') === '55ce33d7678c5a611085589f1f3ddf8b3c52d662cd01d4ba75c
php composer-setup.php
php -r "unlink('composer-setup.php');"
```

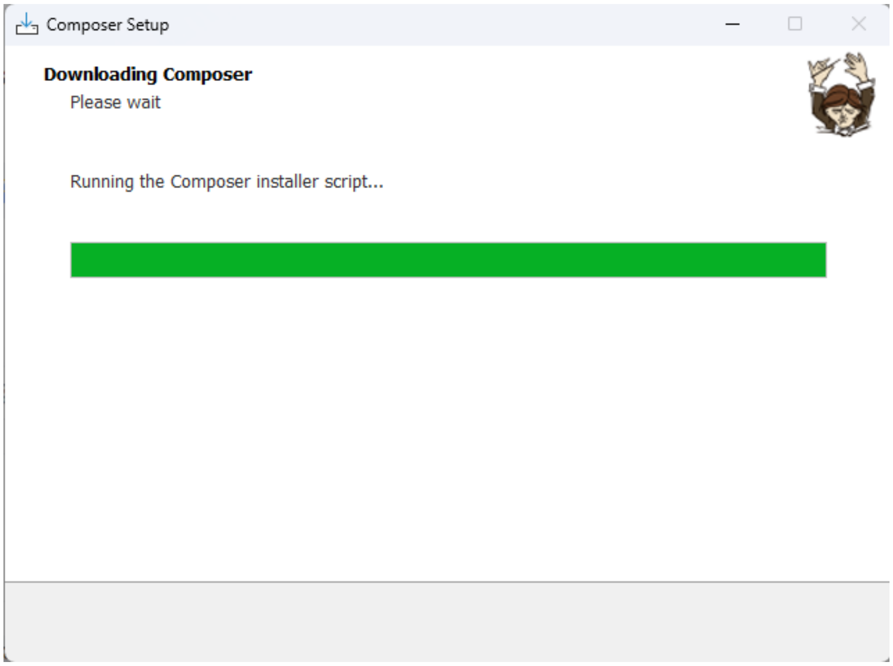
This installer script will simply check some `php.ini` settings, warn you if they are set incorrectly, and then download the latest `composer.phar` in the current directory. The 4 lines above will, in order:

- Download the installer to the current directory
- Verify the installer SHA-384, which you can also cross-check here
- Run the installer
- Remove the installer

Most likely, you want to put the `composer.phar` into a directory on your PATH, so you can simply call `composer` from any directory (*Global install*), using for example:

```
sudo mv composer.phar /usr/local/bin/composer
```

For details, see the instructions on how to install Composer globally.



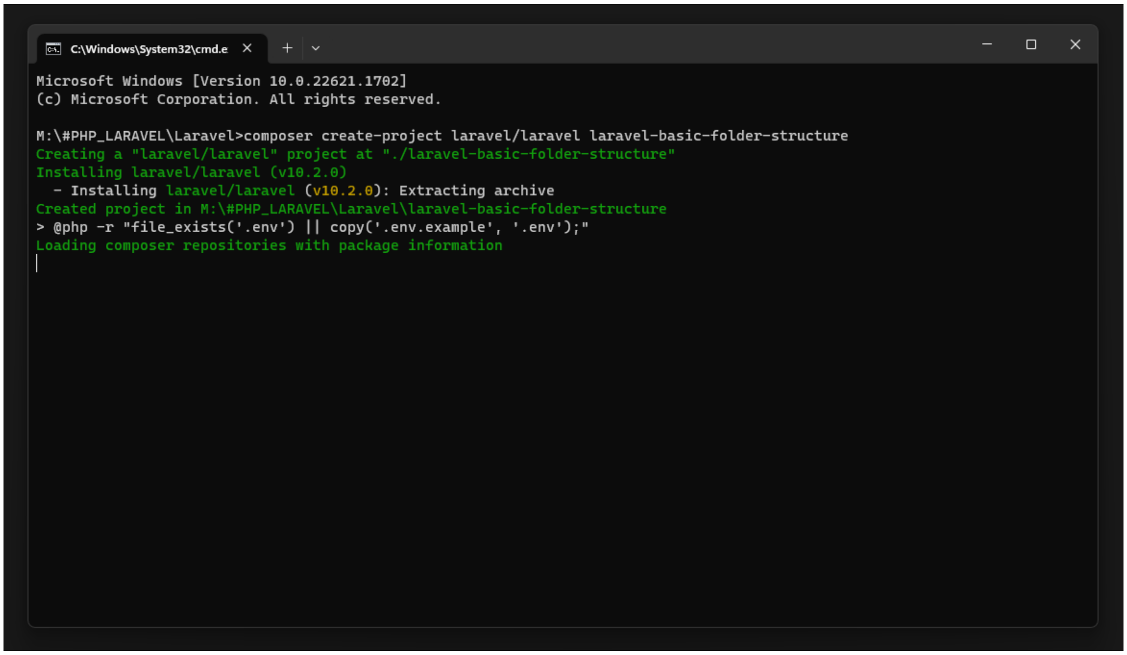
Downloading Composer

Please wait

Running the Composer installer script...

Step 2: Create Latest version of Laravel Project using (Screenshot added)

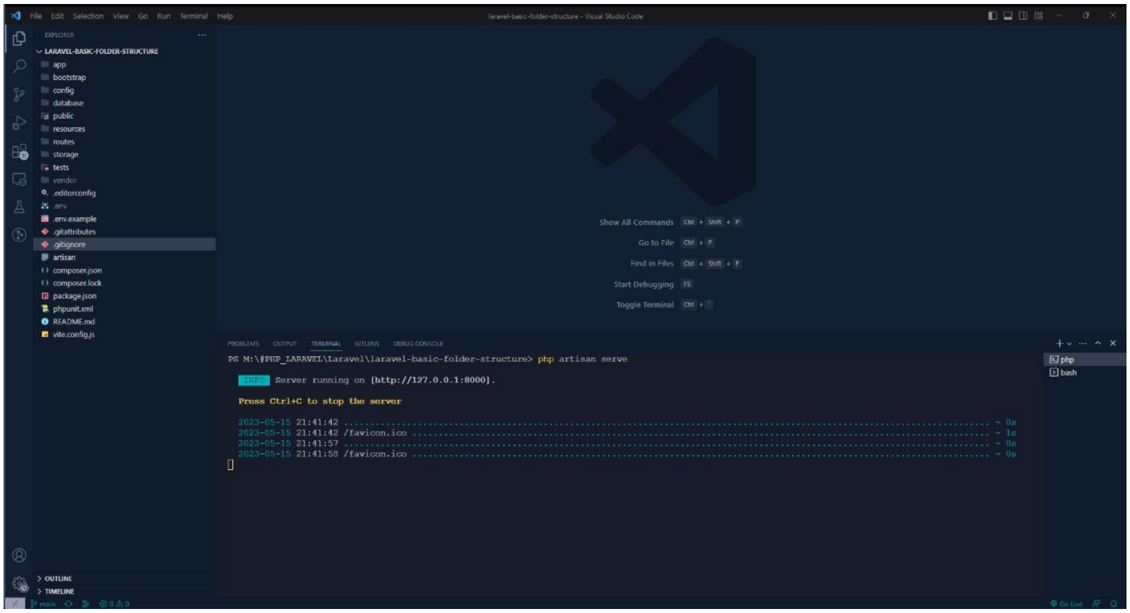
“composer create-project laravel/laravel laravel-basic-folder-structure”



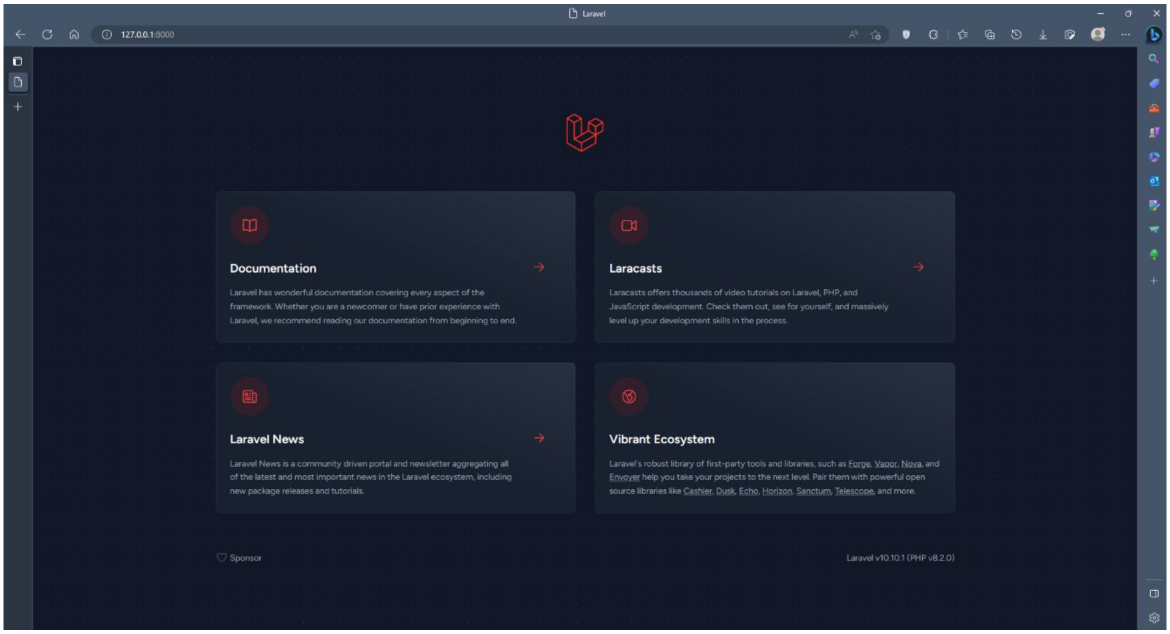
```
Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. All rights reserved.

M:\#PHP_LARAVEL\Laravel>composer create-project laravel/laravel laravel-basic-folder-structure
Creating a "laravel/laravel" project at "./laravel-basic-folder-structure"
Installing laravel/laravel (v10.2.0)
  - Installing laravel/laravel (v10.2.0): Extracting archive
Created project in M:\#PHP_LARAVEL\Laravel\laravel-basic-folder-structure
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
```

Step 3: Run the project using (Screenshot added)

“php artisan serve”



```
PS M:\#PHP_LARAVEL\Laravel\laravel-basic-folder-structure> php artisan serve

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

  Press Ctrl+C to stop the server

  2023-05-15 21:41:42 ..................................................................................... ~ 0s
  2023-05-15 21:41:42 /favicon.ico ........................................................................ ~ 1s
  2023-05-15 21:41:57 ..................................................................................... ~ 0s
  2023-05-15 21:41:58 /favicon.ico ........................................................................ ~ 0s
```



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

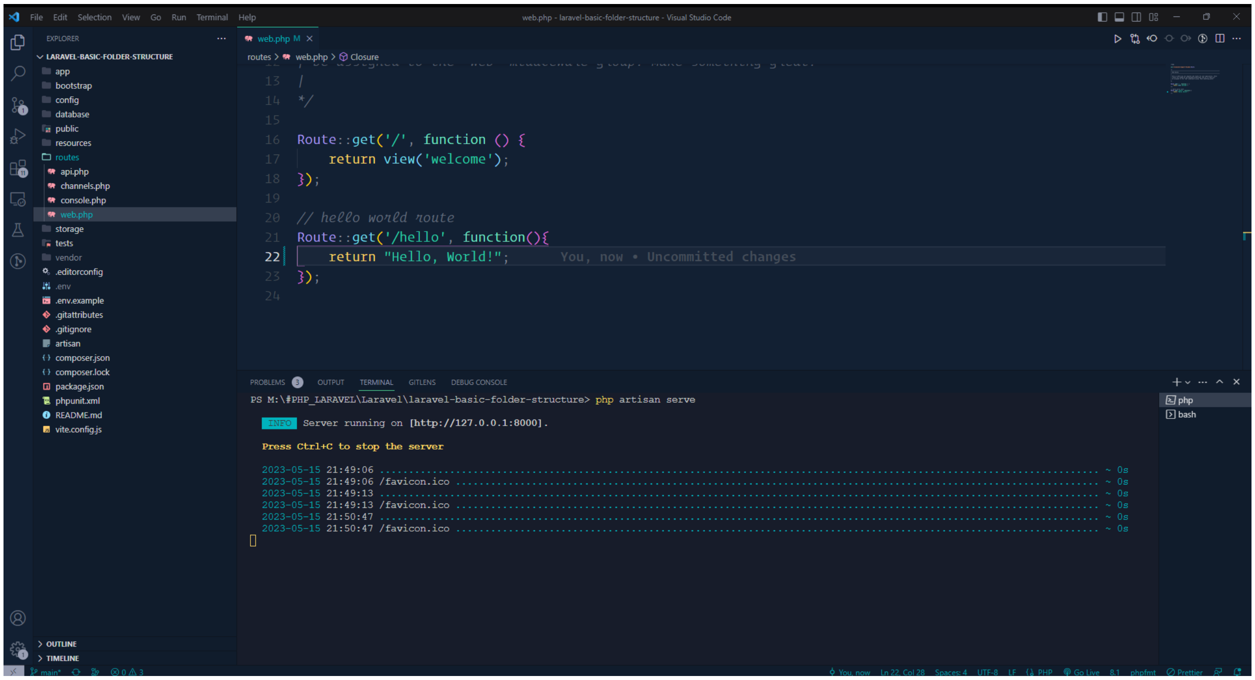
Laravel News is a community driven portal and newsletter aggregating all of the latest and most important news in the Laravel ecosystem, including new package releases and tutorials.

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.

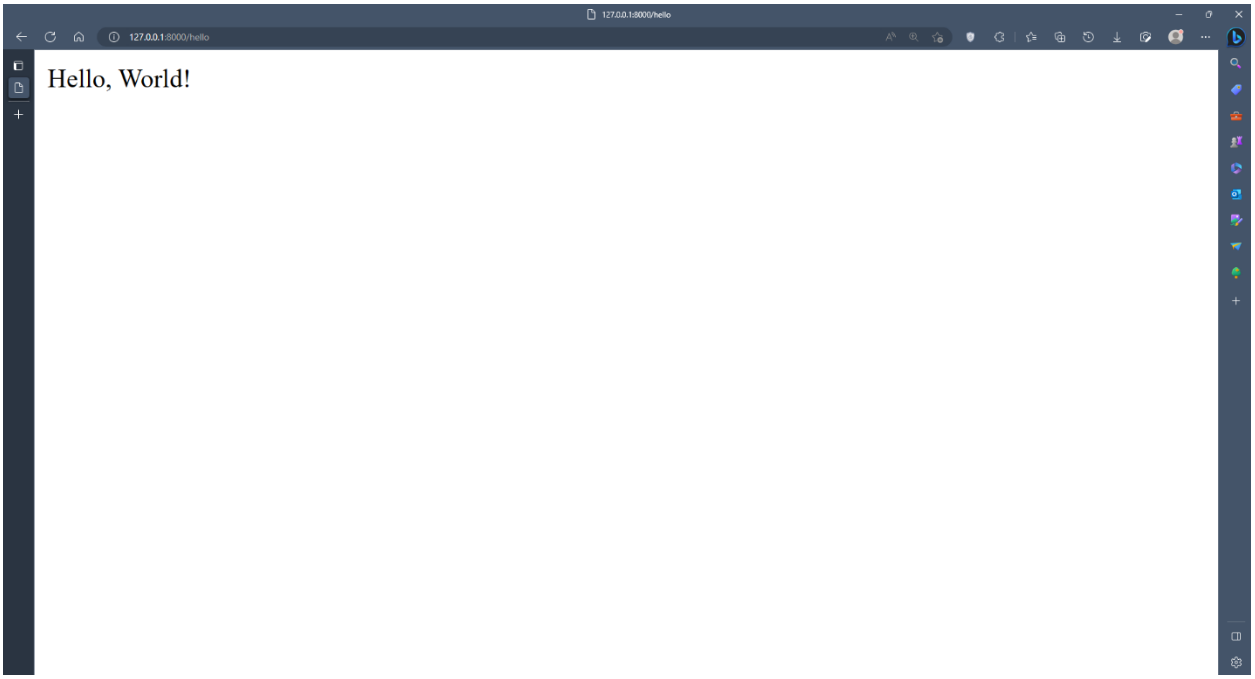
PART 2:

Add new route for displaying “Hello, World!” in **routes/web.php** file (Screenshot added)



```php
| be assigned to the "web" middleware group. Make something great!
|
*/

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

// hello world route
Route::get('/hello', function(){
    return "Hello, World!";        You, now • Uncommitted changes
});
```

```
PS M:\#PHP_LARAVEL\Laravel\laravel-basic-folder-structure> php artisan serve

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

  Press Ctrl+C to stop the server

  2023-05-15 21:49:06 ....................................................................................................... ~ 0s
  2023-05-15 21:49:06 /favicon.ico ........................................................................................ ~ 0s
  2023-05-15 21:49:13 ....................................................................................................... ~ 0s
  2023-05-15 21:49:13 /favicon.ico ........................................................................................ ~ 0s
  2023-05-15 21:50:47 ....................................................................................................... ~ 0s
  2023-05-15 21:50:47 /favicon.ico ........................................................................................ ~ 0s
```



Hello, World!

Screenshot of the running route.

Description of the purpose of each following folder in the Laravel Project:

app: app folder is the most important folder for us as we will work most of the time. In app folder there are important folders like: Http, Models, Providers, Exceptions, Console. From which, Http and Models are very important for us. All of the works related to controller of MVC framework is kept here. It also has Middleware folder and Kernel.php file. In the middleware folder, all the middleware code is kept. In the kernel.php file, code related to security feature, middleware, configuration for web and api is kept. All the works related to models of MVC framework is kept in the Models folder.

bootstrap: some auto generated settings are kept here in bootstrap folder. This folder contains scripts that is used to start up the framework. It has app.php file and cache folder.

database: code related to database operation are kept here. Most used folder in database folder is migration folder.

public: This folder is the application's document root. Index.php file of this folder is the entry point for all the requests entering the application.

resources: This folder contains the views of MVC framework of the project and the assets such as CSS and JS.

routes: This folder contains all the routing definitions that will be used in the project. api.php and web.php are two most important files here.

storage: We upload and download different files; all these files is kept here. This folder contains app, framework, logs folder.

tests: If we work with testcases those are kept in this folder. Also, automated tests are kept here.

vendor: This folder contains composer dependencies. This is auto generated. “composer create” is used to generate this folder if it doesn’t exist in the project folder.