

Part-1:

1.Install Composer: First, you need to install Composer, a dependency manager for PHP. Visit the official Composer website (<https://getcomposer.org/>) and follow the installation instructions for your operating system.

2.Open a terminal or command prompt: Once Composer is installed, open a terminal or command prompt window.

3.Install Laravel: Run the following command to create a new Laravel project:

composer global require laravel/installer

This command installs the Laravel installer globally on your system, allowing you to create new Laravel projects from anywhere.

4. Verify the installation: After the installation is complete, you can verify that Laravel is installed correctly by running the following command:

laravel new project-name

Replace "project-name" with the desired name for your project. This command will create a new Laravel project in a folder with the specified name.

5. Start the development server: Once the project is created, navigate into the project folder using the cd command. For example:

cd project-name

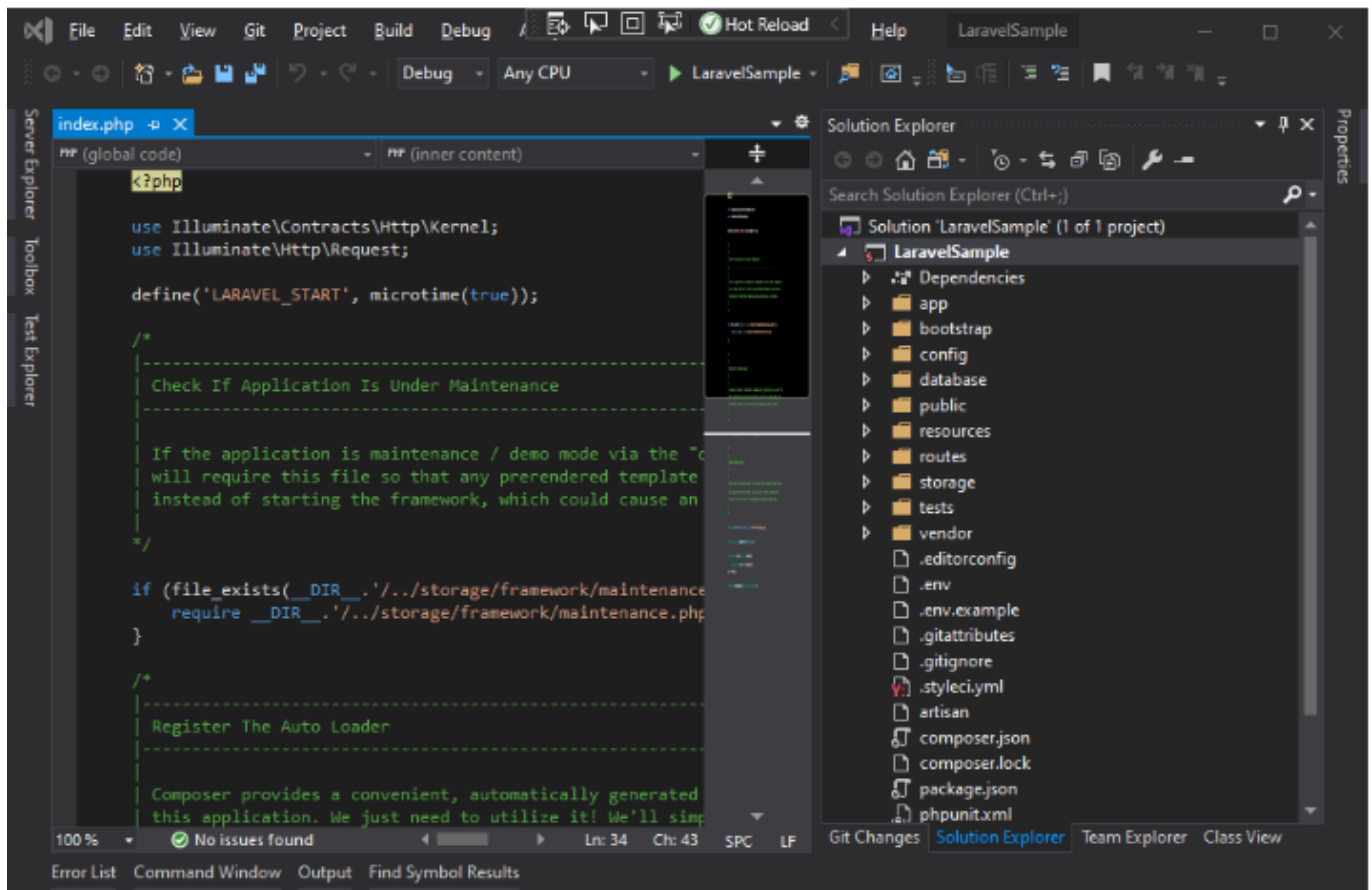
Then, run the following command to start the development server:

php artisan serve

Then output here:

Laravel development server started: <http://127.0.0.1:8000>

6. Verify the running server: Open your web browser and visit the URL shown in the previous step (e.g., <http://127.0.0.1:8000>). If Laravel is installed and running correctly, you should see the default Laravel welcome page.



Part-2:

1.**app**: This folder contains the core application files, including models, controllers, and other PHP classes that define the business logic of your application.

2.**bootstrap**: The bootstrap folder contains the files responsible for bootstrapping the Laravel framework and initializing the application.

3.**config**: The config folder contains various configuration files for your application, including database settings, application settings, and other configuration options.

4.**database**: This folder is used for database-related files, including database migrations, seeders, and factories.

5.**public**: The public folder contains the publicly accessible files of your application, including the index.php file that serves as the entry point for all incoming requests.

6.resources: This folder contains non-PHP resources such as views, language files, and assets like CSS, JavaScript, and images.

7.routes: The routes folder contains the route definitions for your application. It includes web routes (accessible via HTTP) and API routes (typically used for building RESTful APIs).

8.storage: The storage folder is used for storing various files generated by your application, including logs, compiled views, cache files, and uploaded files.

9.tests: This folder is dedicated to storing automated tests for your application, including unit tests, feature tests, and integration tests.

10.vendor: The vendor folder is created and managed by Composer, Laravel's dependency management tool. It contains all the third-party libraries and packages your application depends on.

To create a new route in a Laravel project that displays a "Hello, World!" message, you can follow these steps:

1.Open the routes/web.php file in your Laravel project.

2.Add the following route definition to the file:

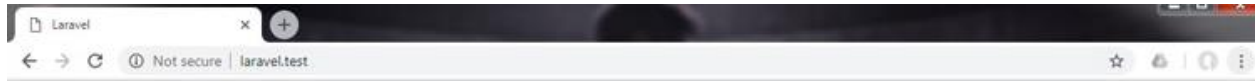
```
Route::get('/hello', function () {  
    return 'Hello, World!';  
});
```

3.Save the file.

4.Start or refresh your Laravel development server.

5.Open your web browser and visit <http://localhost:8000/hello> (assuming you're using the default Laravel development server).

6.You should see the "Hello, World!" message displayed in your browser.



Hello World

This is a sample text in english language.