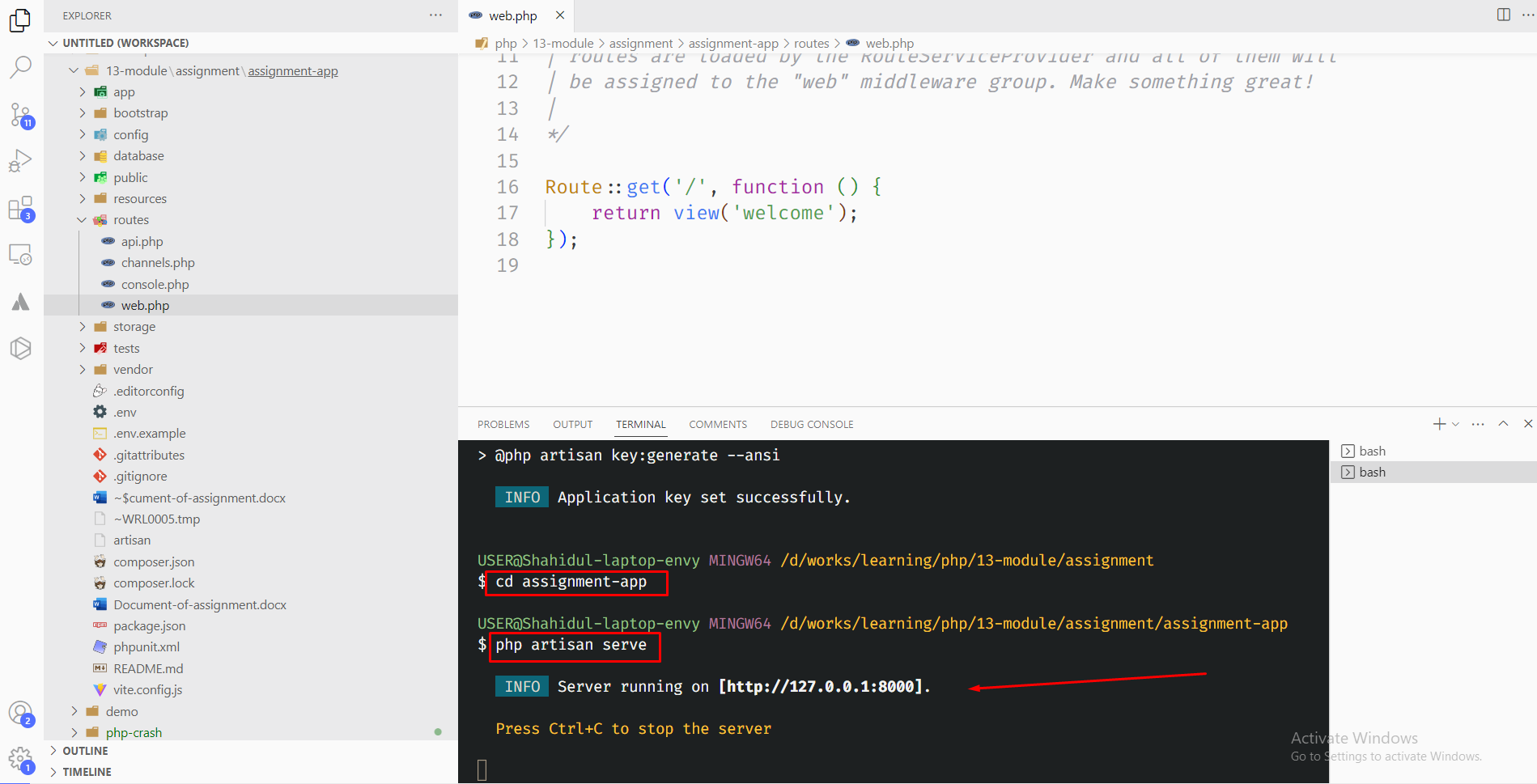
1. Part one: Document of the steps to install Laravel.
2. First, make sure that you have Composer installed on your system. If you don't have it installed already, you can download and install it.
3. Open up your terminal or command prompt and navigate to the directory where you want to install Laravel.
4. Run the following command to create a new Laravel project using Composer:  
   `composer create-project laravel/laravel project-name` Replace project-name with the name of your project.
5. Once the installation is complete, you can navigate to the project directory and start using Laravel.

**Alternatively, you can also install the latest version of Laravel using the Laravel installer. Here are the steps:**

1. First, make sure that you have the Laravel installer installed on your system. If you don't have it installed already, you can install it using Composer by running the following command:  
   ` composer global require laravel/installer`
2. Open up your terminal or command prompt and navigate to the directory where you want to install Laravel.
3. Run the following command to create a new Laravel project using the Laravel installer:  
   ` laravel new project-name` Replace project-name with the name of your project.
4. Once the installation is complete, you can navigate to the project directory and start using Laravel.

That's it! We have successfully installed the latest version of Laravel using either Composer or the Laravel installer.

Here is the screenshot of the running server with command:  


App screenshot from Browser:

A screenshot of a computer

Description automatically generated with medium confidence

2. Part two: Laravel Folder Structure

Here's a brief description of the purpose of each of the following folders in a Laravel project:

app: This folder contains the core logic and code for your Laravel application, including models, controllers, and views.

bootstrap: This folder contains the code that bootstraps the Laravel framework, including the app.php file that sets up the application and loads the environment-specific configuration files.

config: This folder contains the configuration files for your Laravel application, including settings for the database, mail, caching, and other components.

database: This folder contains the database migration files and seeds for your Laravel application, as well as any database-related code, such as models or query builders.

public: This folder contains the publicly accessible files for your Laravel application, including the index.php file that serves as the entry point to your application.

resources: This folder contains the assets and templates for your Laravel application, including views, JavaScript and CSS files, and language files.

routes: This folder contains the route definitions for your Laravel application, including the web.php file that defines the routes for the web interface, and the api.php file that defines the routes for the application's API.

storage: This folder contains the application's cache, logs, and other temporary files that are generated during runtime.

tests: This folder contains the unit and integration tests for your Laravel application, including PHPUnit tests and feature tests.

vendor: This folder contains the third-party packages and dependencies for your Laravel application, including the Laravel framework itself, as well as any additional packages installed via Composer.

Overall, each of these folders serves a specific purpose in a Laravel project and helps organize the code and files in a logical and maintainable way.

Create a new route in your Laravel project that displays a simple "Hello, World!" message. Take a screenshot of the running route.

Here is the browser screenshot: A screen shot of a computer

Description automatically generated with low confidence

And here is the screenshot of web.php file:

A picture containing text, font, screenshot, line

Description automatically generated