**Part 1: Laravel Installation**

Followin steps are used for installing Laravel onto my system –

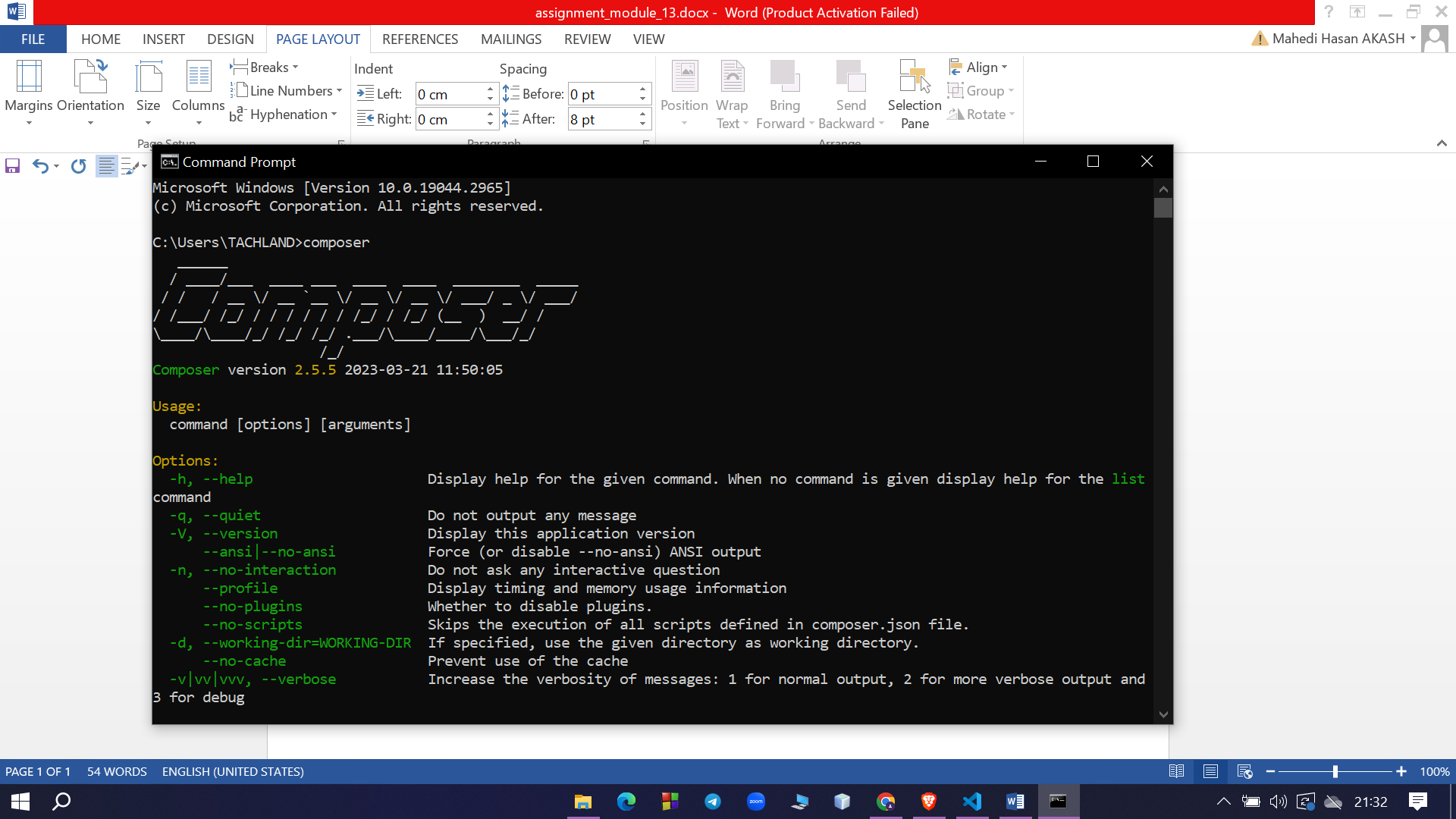
**Step-1**

Visiting the following URL, I downloaded composer and installed it on my system.

[**https://getcomposer.org/download/**](https://getcomposer.org/download/)

**Step-2**

After the Composer is installed, I checked the installation by typing the Composer command in the command prompt as shown in the following screenshot.

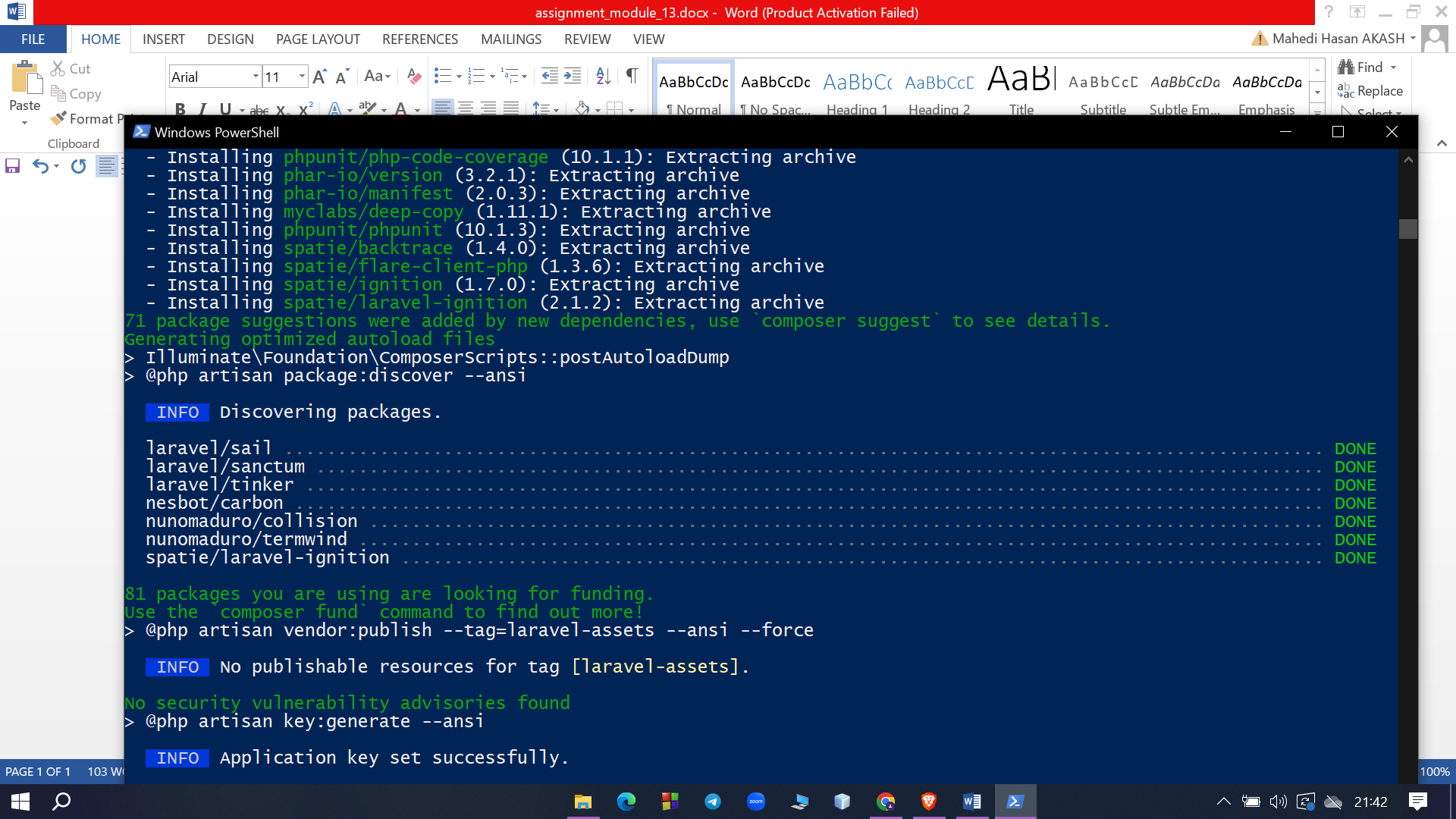


**Step-3**

I created a new directory where I want to create a new Laravel project. After that, I opened a command prompt in that new directory and typed the following command there to create a Laravel project named ‘first\_project’

**composer create-project laravel/laravel first\_project**

The output of the command is as shown below –



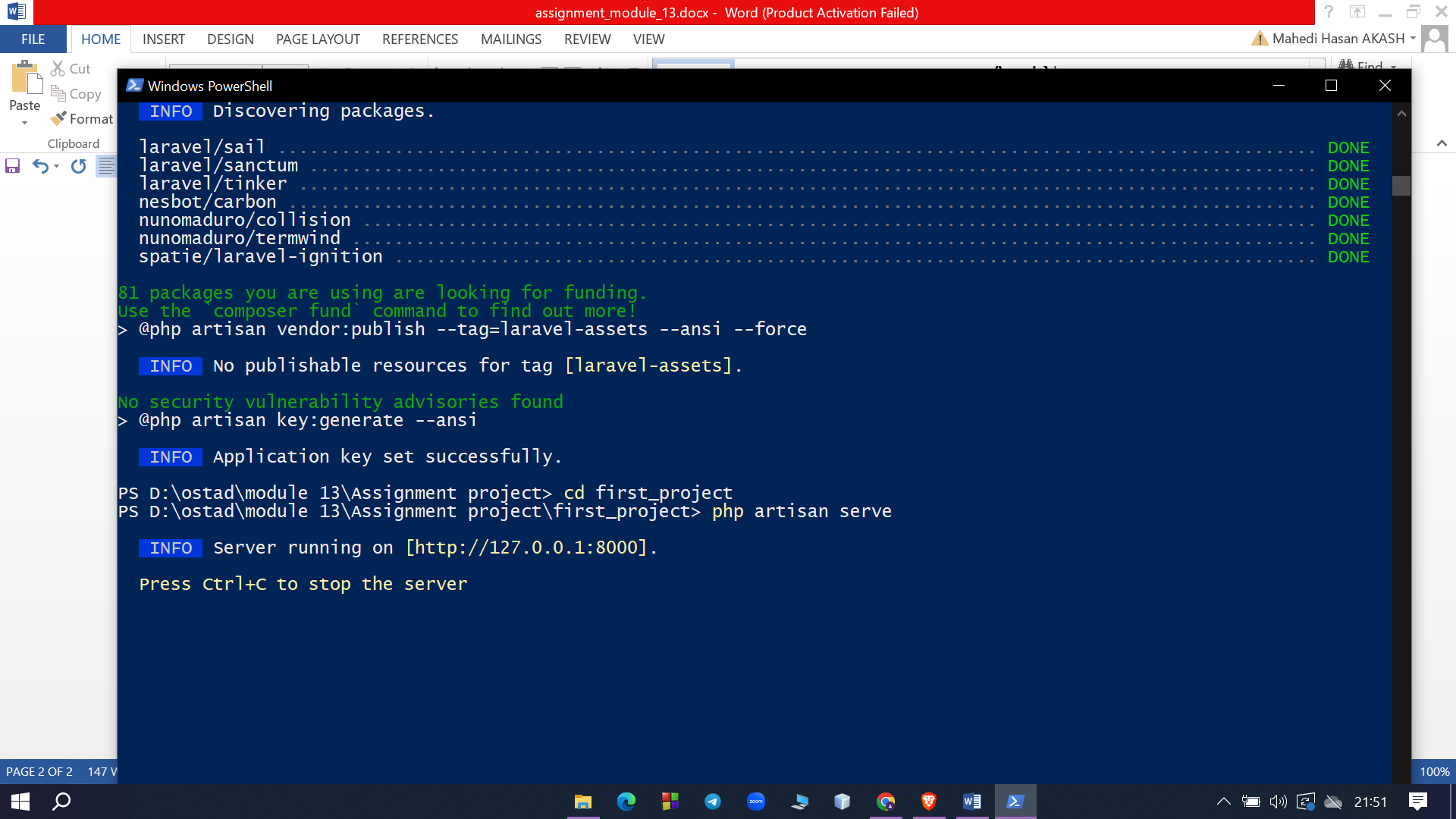
**Step-4**

After that, I changed the directory of the command promt to the directory of the new project.Then I started the development server by executing the following command.

**php artisan serve**

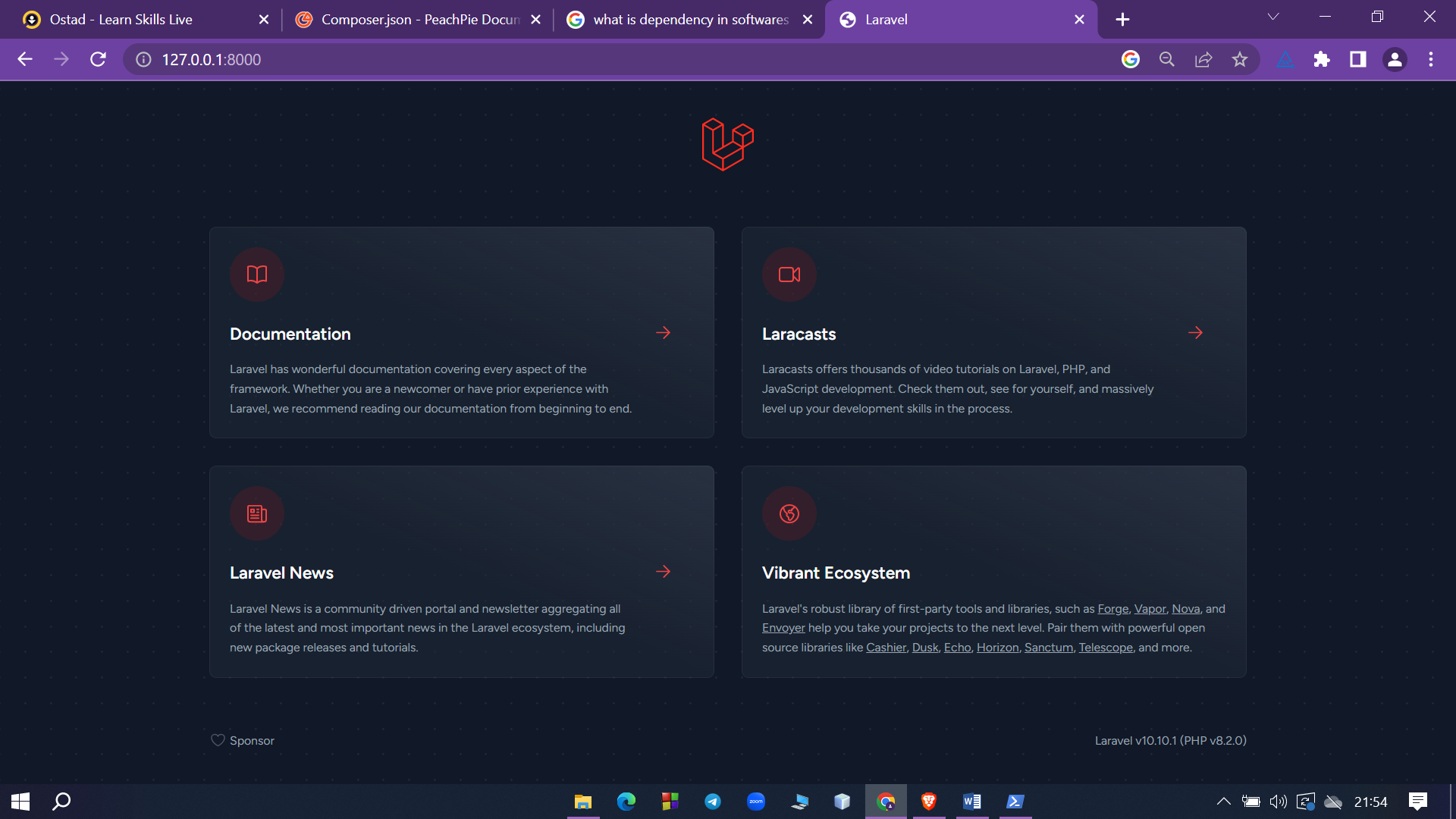
**Step-5**

After executing the above command, I got the following output –



**Step-6**

I copied the URL in the above screenshot and open that URL in the browser. If you see the following screen, it implies Laravel has been installed successfully.



**Part 2:Laravel Folder Structure**

**app**

The majority of our application is housed in the app directory. By default, this directory is namespaced under App and is autoloaded by Composer. The app directory contains a variety of additional directories such as Console, Http,and Providers.Think of the Console and Http directories as providing an API into the core of your application. The HTTP protocol and CLI are both mechanisms to interact with your application, but do not actually contain application logic. In other words, they are two ways of issuing commands to your application. The Console directory contains all of your Artisan commands, while the Http directory contains your controllers, middleware, and requests.

**bootstrap**

The bootstrap directory contains the app.php file which bootstraps the framework. This directory also houses a cache directory which contains framework generated files for performance optimization such as the route and services cache files.

**config**

The config folder includes various configurations and associated parameters required for the smooth functioning of a Laravel application.

**database**

This directory includes various parameters for database functionalities. It includes three sub-directories as given below −

* **Seeds** − This contains the classes used for unit testing database.
* **Migrations** − This folder helps in queries for migrating the database used in the web application.
* **Factories** − This folder is used to generate large number of data records.

**public**

It is the root folder which helps in initializing the Laravel application. It includes the following files and folders −

* **.htaccess** − This file gives the server configuration.
* **javascript and css** − These files are considered as assets.
* **index.php** − This file is required for the initialization of a web application.

**resources**

The resources directory contains your [views](https://laravel.com/docs/10.x/views) as well as your raw, un-compiled assets such as CSS or JavaScript.

**routes**

The routes directory contains all of the route definitions for your 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 your application does not offer a stateless, RESTful API then all your 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](https://laravel.com/docs/10.x/sanctum) and will not have access to session state. The console.php file is where you may define all of your closure based console commands. The channels.php file is where you may register all of the [event broadcasting](https://laravel.com/docs/10.x/broadcasting) channels that your application supports.

**storage**

The storage directory contains your 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 your application. The framework directory is used to store framework generated files and caches. Finally, the logs directory contains your application's log files.

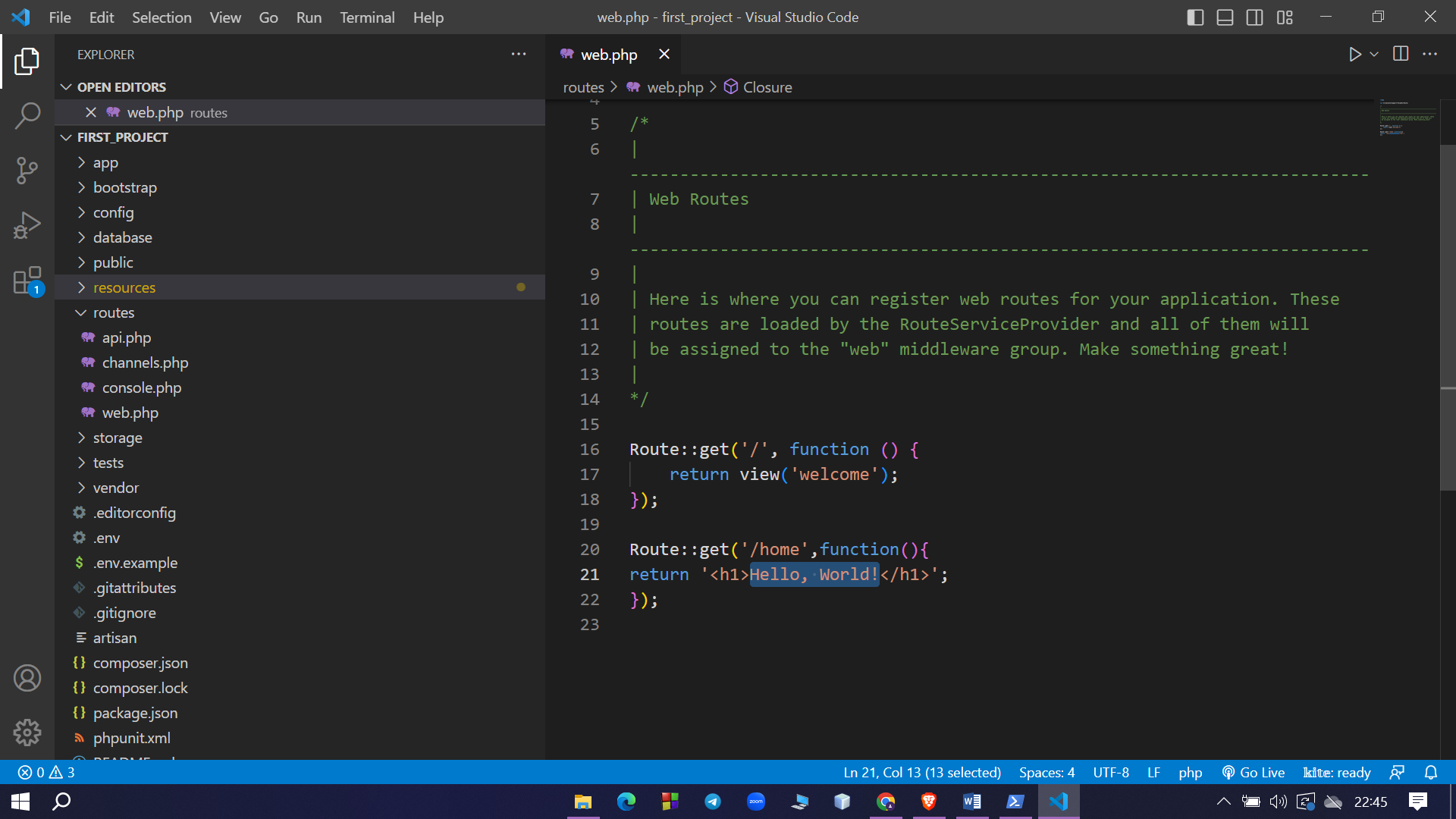
**tests**

The tests directory contains your automated tests. Unit tests and feature tests are provided out of the box. Each test class should be suffixed with the word Test. You may run your tests using the phpunit or ‘php vendor/bin/phpunit’ Or, if you would like a more detailed and beautiful representation of your test results, you may run your tests using the ‘php artisan test’ Artisan command.

**vendor**

The vendor directory contains your Composer dependencies.

**Creating a new route ‘/home’ in the ‘routes/web.php’ file for displaying ‘Hello, World!’**



**After starting the development server and running the route int the browser using ‘localhost:8000/home’ we get the following output**

