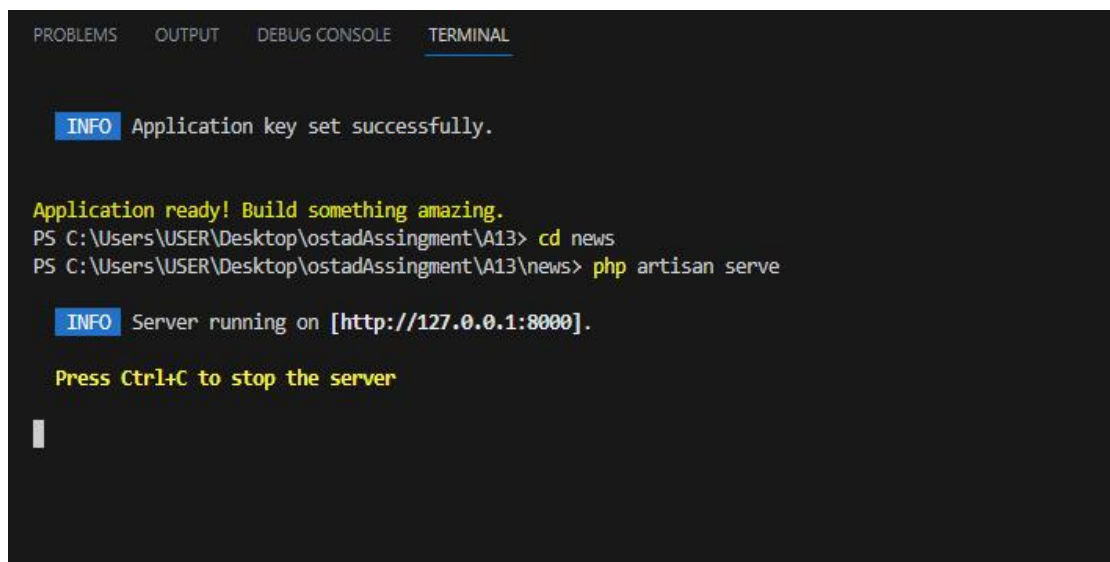


Assignment Part 1:

Steps to install Laravel:

- 1) First I open the vs code terminal
- 2) Then I run the command “laravel new news” .
Here news is my laravel project name.
- 3) Then I start the development server by run the command “php artisan serve”.

Here are the screenshots of running server



The screenshot shows the VS Code terminal interface with the 'TERMINAL' tab selected. It displays the following text:

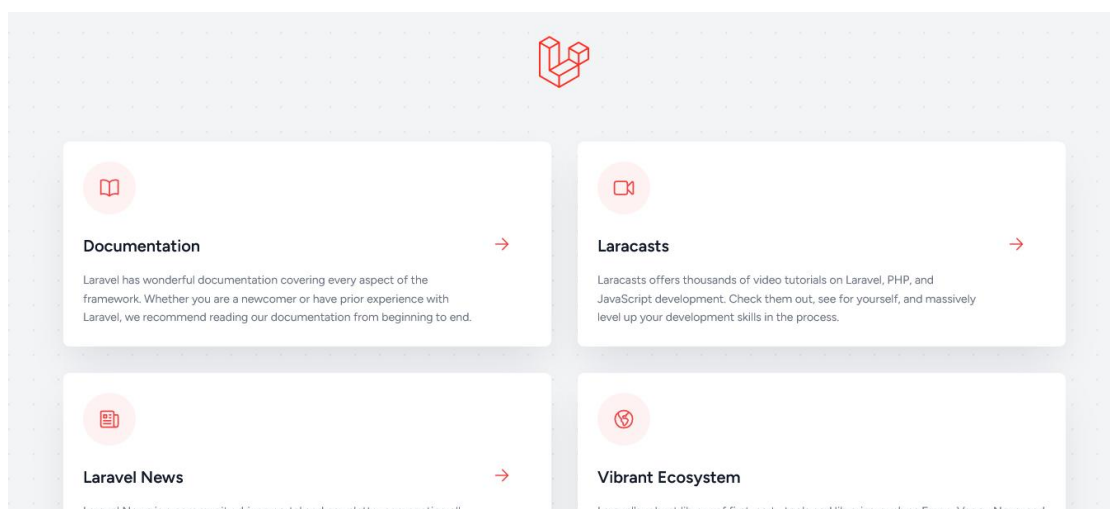
```
INFO Application key set successfully.
```

Application ready! Build something amazing.

```
PS C:\Users\USER\Desktop\ostadAssingment\A13> cd news
PS C:\Users\USER\Desktop\ostadAssingment\A13\news> php artisan serve
```

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

Press Ctrl+C to stop the server



Assignment Part 2:

Here's a short description of the purposes of the major directories in laravel:

App:

The app directory contains the core application code, including the models, controllers, service providers, exceptions and middleware classes.

Here controller classes handle incoming http request , process data and return response.

Model classes are responsible for handling database.

The Providers directory holds service provider classes. Service providers bootstrap various components of application and bind them in the Laravel service container.

The Exceptions directory holds the exception handling classes. It includes the Handler.php file, which allows you to customize how exceptions are handled and format error responses.

The Middleware directory stores middleware classes. Middleware provides a convenient way to filter and modify incoming requests or outgoing responses at various stages of the request lifecycle.

Bootstrap:

The bootstrap directory contains the files responsible for bootstrapping the Laravel application. It includes the app.php file, which sets up the application and registers service providers, and the autoload.php file, which loads the Composer-generated autoloader.

Config:

The config directory contains configuration files for various aspects of the Laravel application. It includes files like app.php (general application configuration), database.php (database connection settings), mail.php (email configuration), and more.

Database:

The database directory holds the database-related files. It includes database migrations for creating and altering tables, database seeders for seeding data in database, and the factories directory for defining data factories used during testing or seeding.

Public:

The public directory is the web server's document root, and it contains the publicly accessible files of application. The index.php file in this directory serves as the entry point for all incoming HTTP requests. Static assets such as CSS, JavaScript, images, and uploaded files are typically stored in this folder.

Resources:

The resources directory contains non-PHP resources used by the application, such as views (in the views subdirectory), language files (in the lang subdirectory), and front-end assets (CSS, JavaScript, etc.) organized in the assets subdirectory.

Routes:

The routes directory contains the route definitions for laravel application. It includes files such as web.php for defining web routes and api.php for defining API routes. This is where we define the endpoints and their corresponding controller methods or closures.

Storage:

The storage directory stores various files generated by laravel application, such as logs, cached files, session files, and uploaded files. It includes subdirectories like app for storing application-specific files, framework for framework-generated files, and logs for log files.

Tests:

The tests directory contains automated tests for application. It includes the TestCase.php file as the base test case class, along with the Feature and Unit directories for organizing different types of tests. We can write PHPUnit tests in these directories to ensure the correctness of our code.

Vendor:

The vendor directory is created and managed by Composer. It contains all the dependencies required by Laravel application. This includes the Laravel framework itself, along with any additional packages or libraries installed by using Composer.

Finally I have created a new Route "hello" in web.php of routes folder which display "Hello, World!".

```
✓ Route::get('hello', function () {  
    |     return "<h1>Hello, World!</h1>";  
    |  
    |});
```

