OSTAD.APP

Web Development with PHP & Laravel - Batch 1

Name of Assignment : Laravel Installation and Folder Structure.

- Part 1: Laravel Installation
- Part 2: Laravel Folder Structure

Full Name : Md. Rafiqul Islam

Module : 13

Rafiqul Islam

Student Sign.

Professor Sign.

Part 1: Laravel Installation

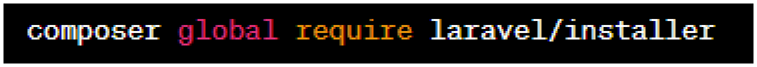
To install Laravel, I followed the steps bellow:

- Step 1: Install Composer

  At first, make sure I have Composer and PHP installed on my system. Composer is a dependency management tool for PHP, and Laravel uses it.

- Step 2: Install Laravel

  Once Composer was installed, I opened my command-line interface (CLI) and ran the following command to install Laravel globally.

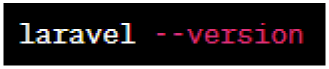
  

  ```
  composer global require laravel/installer
  ```

  This command installs the Laravel installer package globally, allowing me to create new Laravel projects from anywhere on my system.

- Step 3: Verify Laravel Installation

  After the installation was complete, I verified that Laravel was installed correctly by running the following command:

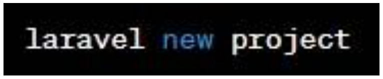
  

  ```
  laravel --version
  ```

  This command will display the installed Laravel version if everything is set up correctly.

- Step 4: Create a New Laravel Project

  To create a new Laravel project, I navigated to the directory where I want to create my project using the CLI and ran the following command:

  ```
  laravel new project
  ```

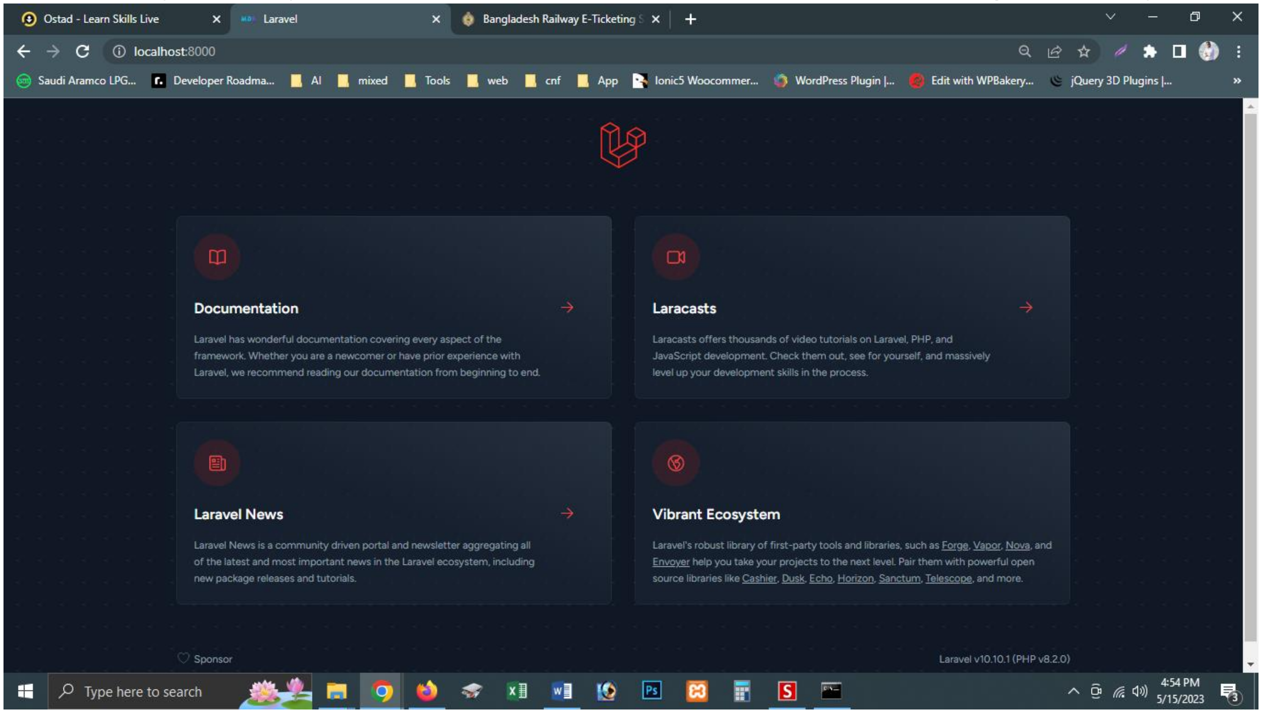
  Where “project” is the name of my new Laravel project.

- Step 5: Run the Development Server

  Once the project was created, navigated to the root directory of my project using the cli and ran the following command:

```
php artisan serve
```

  This command starts the Laravel development server on http://localhost:8000, allowing me to access my Laravel application through my web browser. then I visited the URL http://localhost:8000 in my browser, and I saw the default Laravel welcome page. I took a screenshot of the welcome pate as proof that Laravel was installed and working correctly.



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.

Sponsor

Laravel v10.10.1 (PHP v8.2.0)

Part 2: Laravel Folder Structure

Here is a brief description of each of the folders in a Laravel project that mentions on the assignment:

app: Contains the core application files, including controllers, models, and custom classes.

bootstrap: Contains files responsible for bootstrapping and initializing the Laravel framework.

config: Contains configuration files for various aspects of the Laravel application, such as database settings, application settings, and service providers.

database: Contains database-related files, including migrations, seeds, and database factories.

public: This is the web server's document root and contains the entry point for the application (index.php), as well as publicly accessible assets like CSS, JavaScript, and image files.

resources: Contains the views, language files, and raw assets (such as SCSS or JavaScript files) that are used in the application.

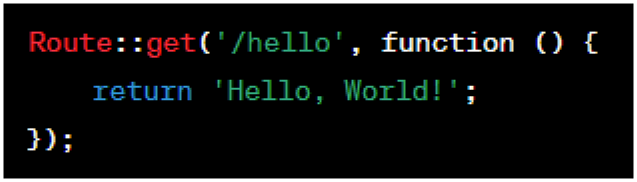
routes: Contains all the route definitions for the application, which determine how the HTTP requests should be handled.

storage: Stores files generated by the application, such as logs, compiled views, and uploaded files.

tests: Contains test cases for the application, which are used for testing and ensuring the application's functionality.

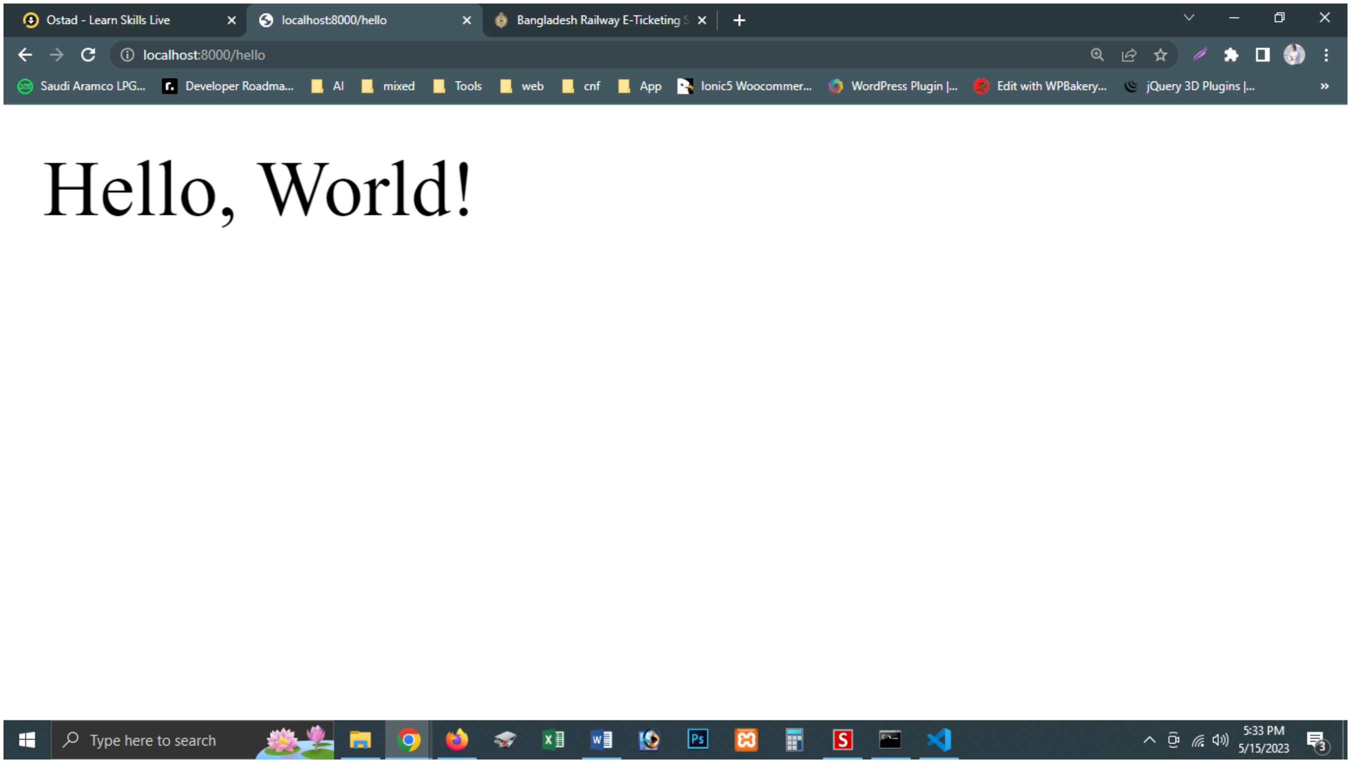
vendor: Contains the dependencies installed via Composer, including Laravel itself and any third-party packages.

To create new route that display a “Hello, World!” message, I added the following code to the “routes/web.php” file:



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

Then I visited the URL http://localhost:8000/hello in my browser, and I saw the “Hello, World!” message. I took a screenshot of the message as proof that the new route was working correctly.



Hello, World!