Assignment-13
Md.Sumon

Part 1:
To install Laravel, I followed these steps:
1. At first i have installed PHP on My system.
2. Then I downloaded the Composer and installed it. Composer is a dependency manager for PHP.
3. Installing Laravel:
✔ Opened my command line interface (CLI) or terminal.
✔ Runned the following command to install Laravel using Composer:
✔ composer -project laravel/laravel project name
4. Creating a New Laravel Project:
✔ Navigated to the directory where i wanted to create my Laravel project.
✔ Runned the following command to create a new Laravel project:
✔ laravel new project-name
5. Served the Laravel Application:
✔ Change into the project directory: using this command
✔ cd project-name
6. Then opened the folder into my VS CODE code editor my using this command:

Code :
7. Finally To start the development server and serve my Laravel application, I runned the following command:
✔ php artisan serve
Below Screen sort shows the Running of my development Server.

Part 2:
Describing the purpose of each folder in a Laravel Project:
app:
The app folder is the core of the Laravel application.
It contains the application's models, controllers, middleware, and other PHP classes.
The app folder is where we define the business logic of your application.

bootstrap:
The bootstrap folder contains files responsible for bootstrapping the Laravel framework.
It includes the app.php file, which sets up the application and loads the necessary components.
The bootstrap folder is primarily used for configuring the application environment and performing any initialization tasks.

config:
The config folder contains configuration files for various aspects of the Laravel application.
It includes files like app.php, database.php, mail.php, etc., which allow us to customize the behavior of our application.
We can modify these configuration files to set database connections, cache settings, mail settings, and more.

database:
The database folder holds the database-related files for our Laravel application.
It includes migration files that define the structure of our database tables.
The seeds directory contains files that populate the database with sample data.
We can also create factories in the factories directory to generate fake data for testing purposes.

public:
The public folder is the web server's document root and the entry point for all HTTP requests.
It contains the index.php file, which serves as the front controller for the application.
Static assets like CSS, JavaScript, and images are typically stored in the public folder, making them accessible to the outside world.

resources:
The resources folder holds non-PHP resources used by the application, such as views, language files, and assets.
The views directory contains the Blade templates that define the structure and layout of our application's HTML pages.
Other directories within resources may include language files for localization, CSS and JavaScript files, and other assets.

routes:
The routes folder contains route definitions for our application.
The web.php file defines routes that handle HTTP requests made by web browsers.
The api.php file defines routes for API endpoints.
Additional route files can be created to organize and manage routes for different parts of our Application.

storage:
The storage folder is used to store files generated by the application.
It includes directories for logs, cache files, session files, and other temporary or dynamically generated content.
Laravel also uses the storage folder to store uploaded files, such as user-generated images or Documents.

tests:
The tests folder contains automated tests for your Laravel application.
It includes test cases and suites that help ensure the functionality and integrity of our code.
Laravel provides a testing framework that allows you to write unit tests, integration tests, and More.

vendor:
The vendor folder contains the dependencies installed via Composer, including the Laravel framework itself.
It includes all the external libraries and packages required by your application.
The vendor folder is generated and managed by Composer, so we should not modify its contents directly.