**Part 1: Laravel Installation:**

1. Install the PHP latest version:

\* Download the php latest version from <https://www.php.net/downloads.php>.

\*After finishing the download, install it.

\* Set the path in environment variable .

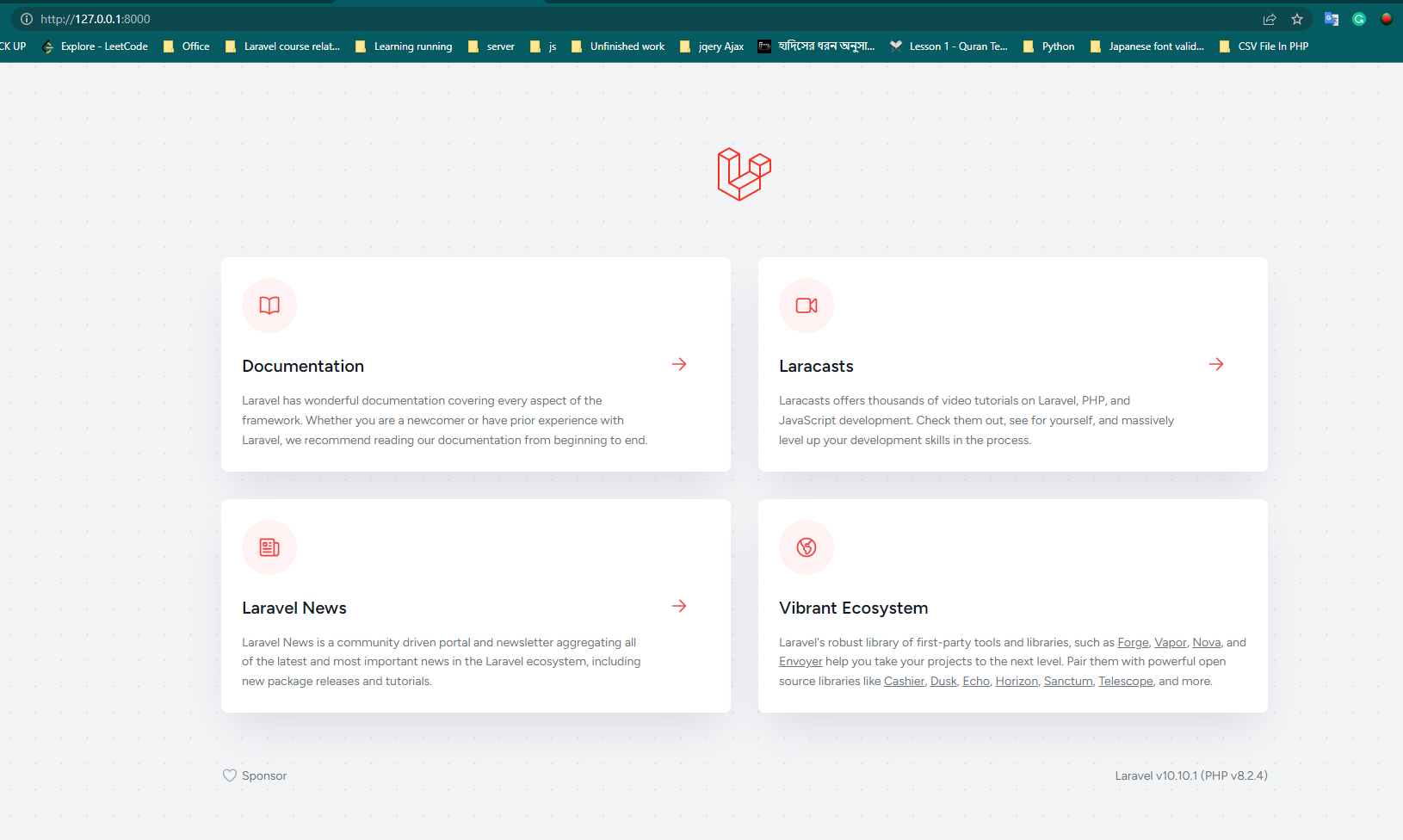
2.Download the xampp from <https://www.apachefriends.org/download.html> and install it.

3.Download the composer from <https://getcomposer.org/download/> and install it by Composer-Setup.exe.

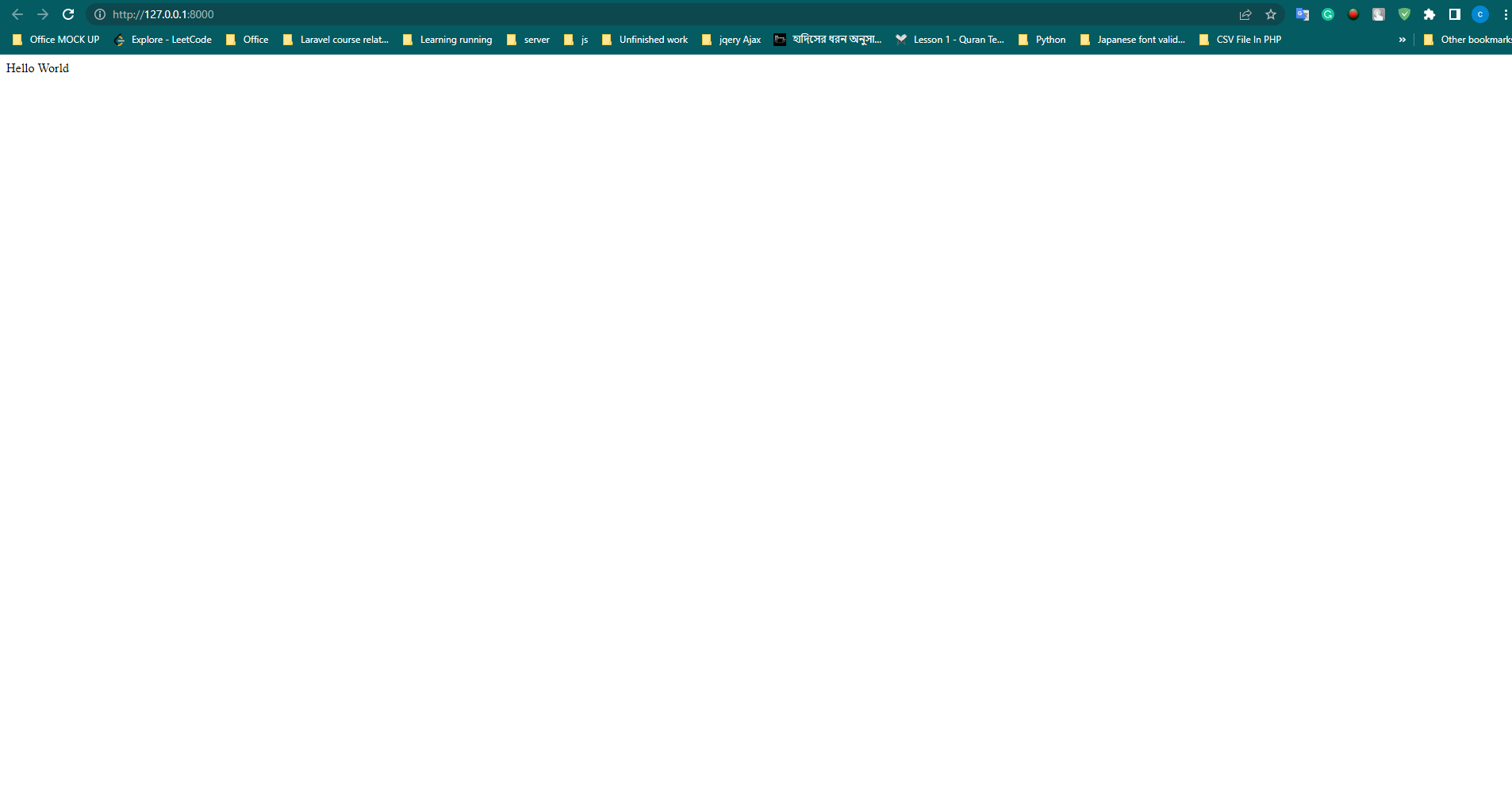
4. Create new Laravel project by composer with the command “composer create-project --prefer-dist laravel/laravel newProject”.

5.Successfully create the project, run the project by the command “php artisan serve” in cmd.

Project image:



**Part 2: Laravel Folder Structure:**



**In a Laravel project, the folders serve different purposes and organize various components of the application. Here's a description of each folder**:

1. app: This folder contains the core application code, including models, controllers, and other classes that define the application's logic. It's where you'll write most of your code.

2. bootstrap: The bootstrap folder contains files that are responsible for bootstrapping the Laravel framework and initializing the application. It includes the app.php file, which sets up the application environment and loads necessary dependencies.

3. config: The config folder contains configuration files for various aspects of the Laravel application, such as database settings, application-specific configurations, and third-party package configurations. You can modify these files to customize the behavior of your application.

4. database: This folder contains database-related files, such as migrations and seeders. Migrations are used to manage database schema changes, while seeders allow you to populate the database with dummy data.

5. public: The public folder is the web server's document root, and it contains the entry point for the application, which is the index.php file. It also includes publicly accessible assets like CSS, JavaScript, and image files.

6. resources: The resources folder holds files that are used during the development process, such as views, language files, and assets like CSS and JavaScript files. Views are typically written in Blade syntax and define the HTML templates for your application's user interface.

7. routes: The routes folder contains files that define the application's routes, which determine how different URLs map to controller actions or closures. Routes can be defined for handling HTTP requests, such as GET, POST, PUT, or DELETE, and they help direct traffic within the application.

8. storage: The storage folder is used to store various files generated by the application, such as log files, cached views, and uploaded files. It has subfolders like app, framework, and logs, where different types of files are stored.

9. tests: The tests folder holds the automated tests for your Laravel application. It provides a convenient location to organize and run tests, ensuring that your code behaves as expected and helps maintain application quality.

10. Vendor: The vendor folder contains the dependencies installed via Composer, the PHP package manager used by Laravel. It includes third-party libraries and packages that your application relies on. It's generally recommended not to modify or add files directly within this folder, as Composer manages it for you.

These folders provide a structured organization for the different components of a Laravel project, making it easier to locate and manage the various files and functionalities within the application.