

Laravel Framework

ROJO, J/ALIWATE, D/EDO, A EDO/UY, ULRICH/ODARVE, C.

- What is a Framework
- Comparison between Framework and Development from scratch
- The Laravel Framework
- Advantages of Laravel Framework
- Installation of Laravel Framework in windows OS
- Run Laravel application in the server

- A framework is a pre-built structure that provides a foundation for developing applications.
- It includes reusable code, predefined functions, libraries, and tools that simplify and accelerate development.
- Frameworks enforce coding standards and best practices, reducing development time and minimizing

Comparison: Framework vs. Development from Scratch

Feature	Framework	Development from Scratch
Speed	Faster development due to pre-built components.	Slower as everything must be built from the ground up.
Code Reusability	High—offers pre-written modules and libraries.	Low—developers must write all functionality manually.
Complexity	Lower—handles many backend tasks automatically.	Higher—developers must implement everything themselves.
Customization	Limited to what the framework allows.	Full control over every aspect of development.
Security	More secure—frameworks have built-in security features.	Must implement security measures manually.
Maintenance	Easier—frameworks provide updates and bug fixes.	Harder—developers must maintain and update all code.
Learning Curve	May require learning framework-specific concepts.	Requires deep knowledge of all technologies used.
Scalability	Easier to scale due to optimized architecture.	Can be optimized but requires more effort.

- Laravel is often considered a **full-stack framework** because it provides both **backend** (server-side) and **frontend** (client-side) capabilities, allowing developers to build complete web applications without relying on additional frameworks.

- Laravel is a **PHP framework** designed for building web applications efficiently.
- It follows the **Model-View-Controller (MVC)** architectural pattern and provides built-in tools to handle common web development tasks, such as authentication, routing, caching, and database management.

Features of Laravel Framework

1. **MVC Architecture** – Separates business logic, user interface, and data handling for better maintainability.
2. **Eloquent ORM** – An advanced database abstraction layer that simplifies database interactions.
3. **Blade Templating Engine** – A lightweight and powerful way to build dynamic web pages.
4. **Routing System** – Simplifies URL handling and request management.
5. **Security Features** – Includes hashed passwords, SQL injection prevention, and CSRF protection.
6. **Authentication & Authorization** – Comes with pre-built authentication functionality.
7. **Artisan Command-Line Tool** – Automates repetitive tasks like migrations, database seeding, and testing.
8. **Queue Management** – Handles background jobs efficiently.
9. **Event & Broadcasting System** – Allows real-time communication (e.g., for chat applications).
10. **RESTful API Support** – Easily build APIs with built-in support for JSON responses.

Laravel vs. Core PHP Development (from Scratch)

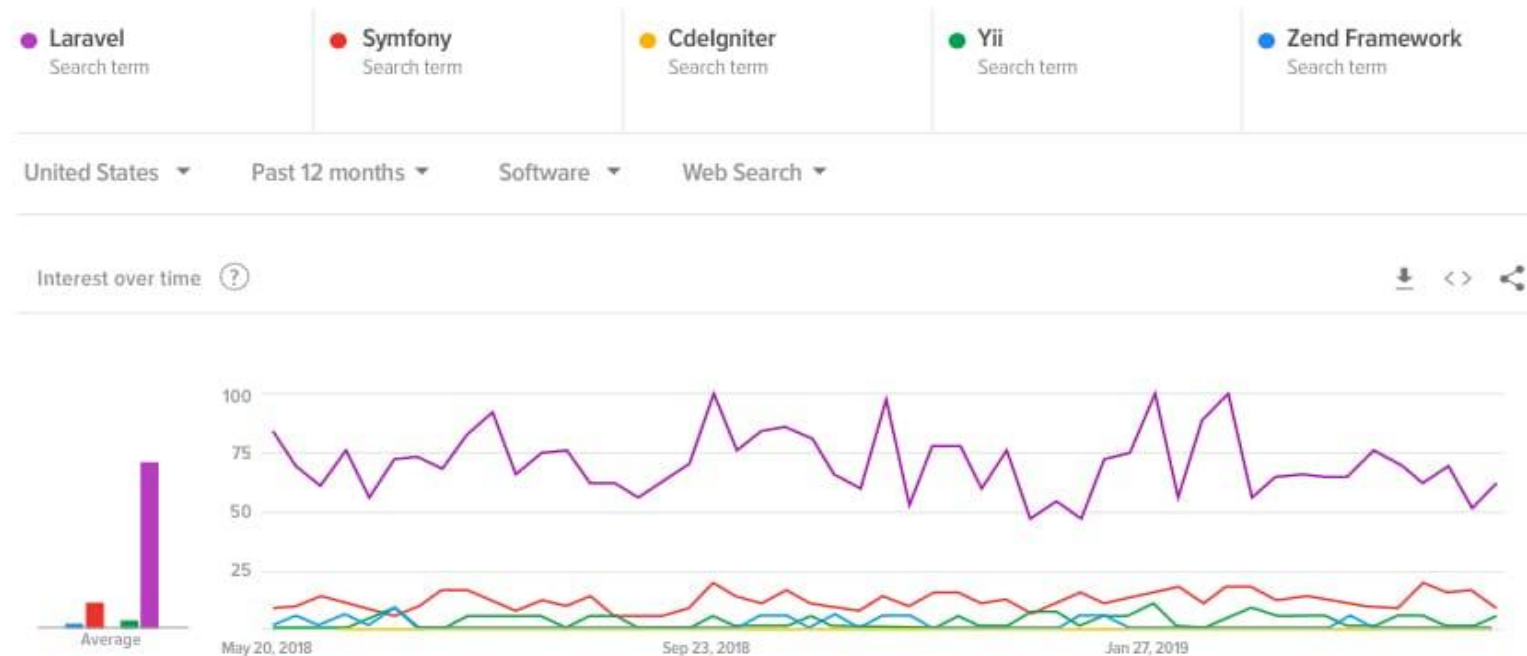
Feature	Laravel Framework	Core PHP (From Scratch)
Development Speed	Faster—pre-built features and tools	Slower—must write everything manually
Code Organization	Follows MVC for better structure	Requires manual structuring
Security	Built-in protections (CSRF, SQL Injection)	Must implement security manually
Database Handling	Eloquent ORM makes it easy	Manual SQL queries required
Built-in Features	Authentication, email, queues, etc.	Everything must be built from scratch
Scalability	Easier to scale with built-in optimizations	Requires additional effort for scaling

ADVANTAGES OF LARAVEL FRAMEWORK

1. Speeds up development process

- Laravel framework works on a modular basis.
- Speeds up the development process of web applications,
- Laravel built in 2011 by Taylor Otwell

Popularity of Laravel Framework



2. Better Authentication and Authorization Option

Laravel framework makes it extremely simple and easy to implement a complete authentication system—

- a. Login,
- b. Registration, and
- c. Password Reset.

3. Less Technical Vulnerabilities

Example of critical web application security vulnerabilities.

1. SQL injection,
2. cross-site request forgery,
3. cross-site scripting

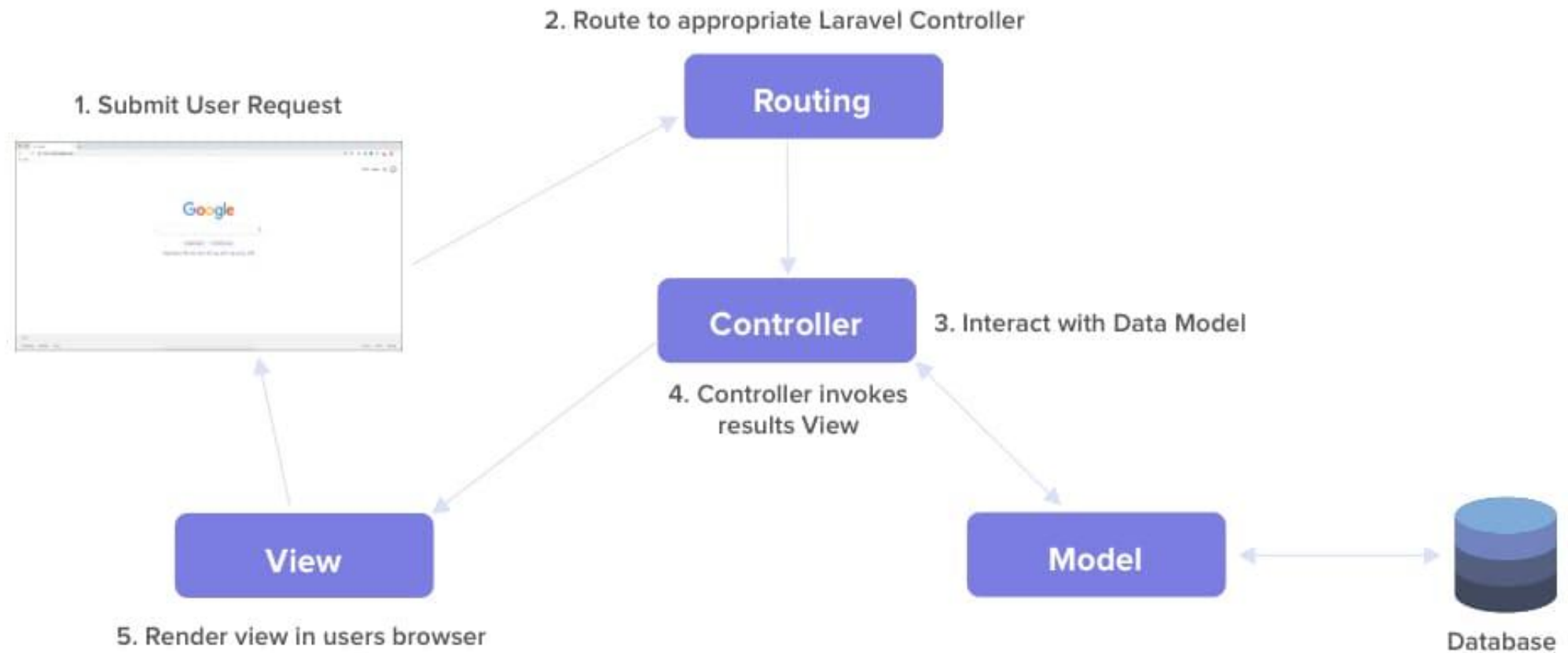
4. MVC Architecture of Laravel Framework

- a. Laravel follows a Model-view-controller architecture
- b. MVC architecture comes up with built-in functionalities that developers can use.

MVC architecture provides :

1. better documentation
2. improved performance,
3. multiple built-in functionalities

Architecture of Laravel MVC



5. Traffic Handling

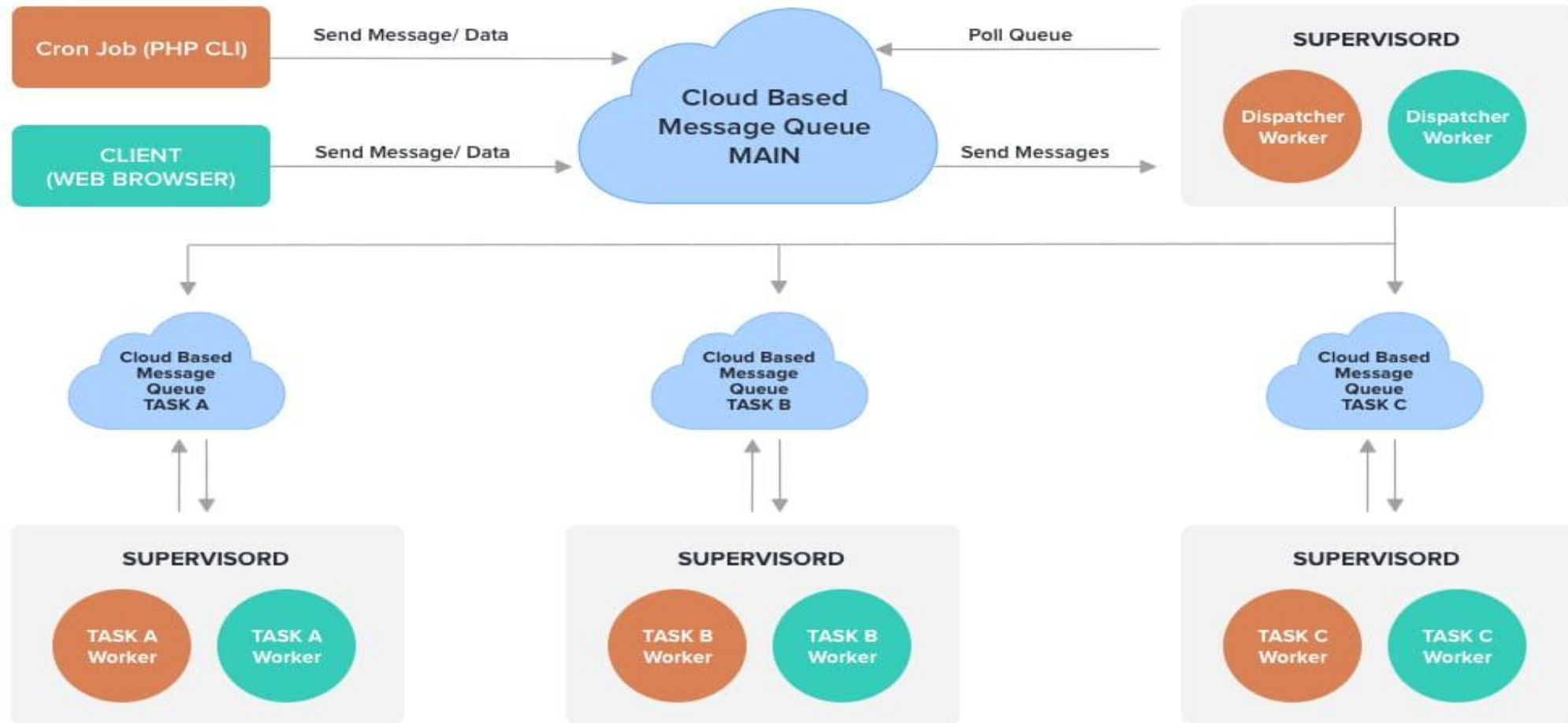
Laravel framework provides :

1. load balancing alongside others.
2. queue service
3. keeps a web server healthy.

Queues allow you to defer the processing of a time-consuming task, such as:

sending an email, until a later time, which drastically speeds up web requests to your application.

Message Queue System of Laravel



Laravel installation in Windows

https://getcomposer.org/download/

[Home](#) | [Getting Started](#) | [Download](#) | [Documentation](#) | [Browse Packages](#)

Download Composer Latest: v1.10.9

Windows Installer

The installer will download composer for you and set up your PATH environment variable so you can simply call `composer` from any directory.

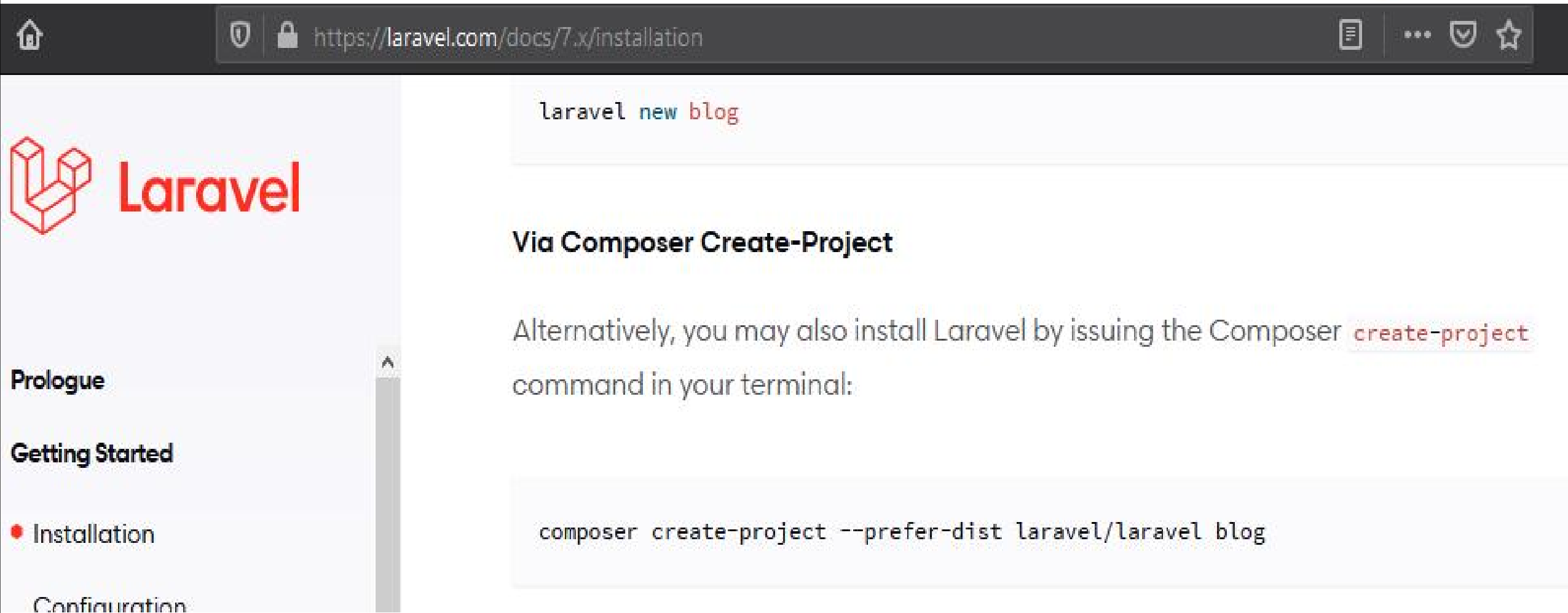
Download and run [Composer-Setup.exe](#) - it will install the latest composer version whenever it is executed.

Command-line installation

To quickly install Composer in the current directory, run the following script in your terminal. To automate the installation, use [the guide on installing Composer programmatically](#).

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') === 'e5325b19b381bfd88ce90a5ddb7823406b2a38cff6b3046719132d71e3941c7e56') { echo 'Composer setup script is not officially signed' . PHP_EOL; }"
php composer-setup.php
php -r "unlink('composer-setup.php');"
```

Creating a project in Laravel



The screenshot shows a web browser window with the URL `https://laravel.com/docs/7.x/installation`. The page features the Laravel logo on the left and a navigation menu with items: Prologue, Getting Started, Installation (highlighted with a red dot), and Configuration. The main content area displays the terminal command `laravel new blog` in a code block. Below this, the section **Via Composer Create-Project** is introduced, followed by the text: "Alternatively, you may also install Laravel by issuing the Composer `create-project` command in your terminal:". A second code block shows the command `composer create-project --prefer-dist laravel/laravel blog`.

laravel new blog

Via Composer Create-Project

Alternatively, you may also install Laravel by issuing the Composer `create-project` command in your terminal:

```
composer create-project --prefer-dist laravel/laravel blog
```

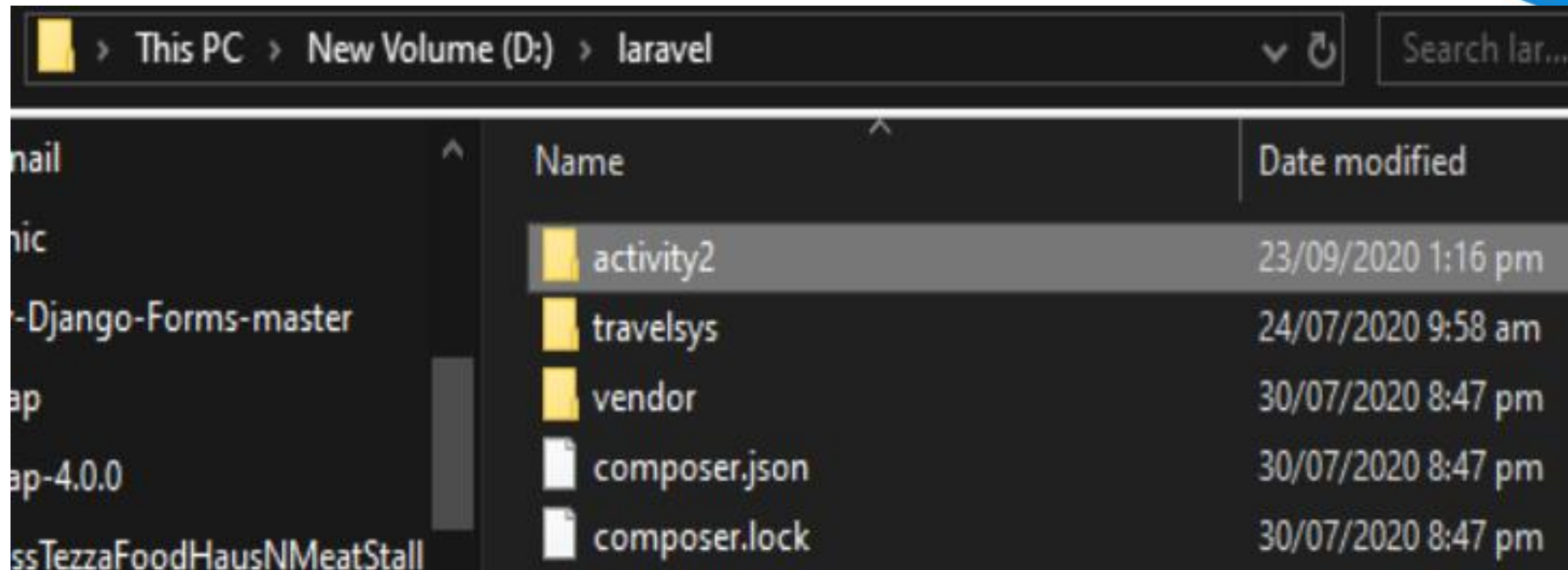
During Installation

```
D:\laravel>composer create-project --prefer-dist laravel/laravel activity2
Creating a "laravel/laravel" project at "./activity2"
Installing laravel/laravel (v7.28.0)
- Installing laravel/laravel (v7.28.0): Downloading (100%)
Created project in D:\laravel\activity2
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 100 installs, 0 updates, 0 removals
- Installing voku/portable-ascii (1.5.3): Loading from cache
- Installing symfony/polyfill-ctype (v1.18.1): Loading from cache
- Installing phpoption/phpoption (1.7.5): Loading from cache
- Installing vlucas/phpdotenv (v4.1.8): Loading from cache
- Installing symfony/css-selector (v5.1.5): Loading from cache
- Installing tijsverkoyen/css-to-inline-styles (2.2.3): Loading from cache
- Installing symfony/polyfill-php80 (v1.18.1): Loading from cache
- Installing symfony/polyfill-mbstring (v1.18.1): Loading from cache
- Installing symfony/var-dumper (v5.1.5): Downloading (100%)
- Installing symfony/deprecation-contracts (v2.2.0): Downloading (100%)
> @php artisan key:generate --ansi
Application key set successfully.

D:\laravel>
```

successfully.
are looking for funding.
command to find out more!

Directory created with laravel packages



The screenshot shows a Windows File Explorer window with the address bar displaying the path: This PC > New Volume (D:) > laravel. The search bar contains the text "Search lar...". The main pane shows a list of files and folders:

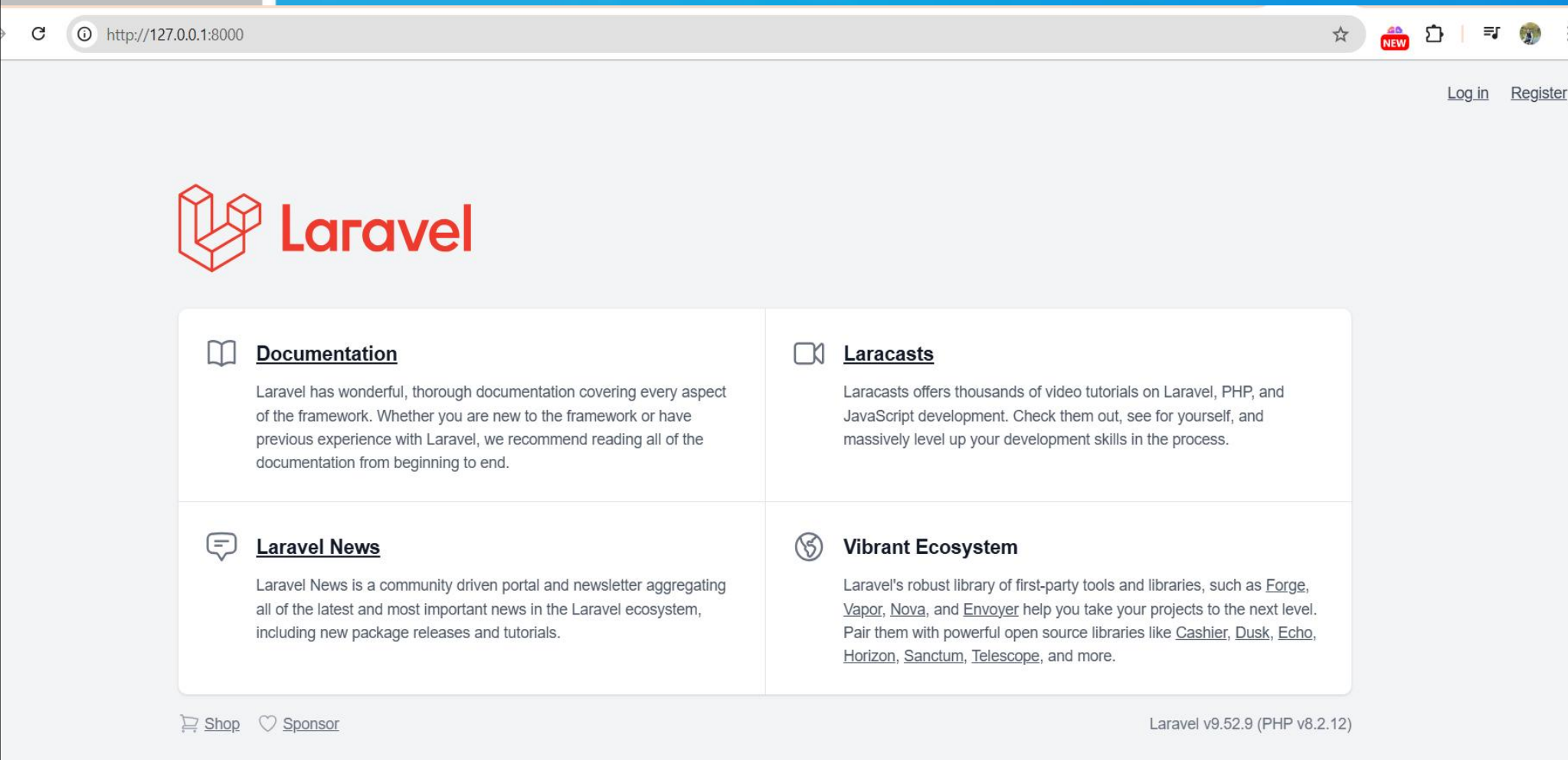
Name	Date modified
activity2	23/09/2020 1:16 pm
travelsys	24/07/2020 9:58 am
vendor	30/07/2020 8:47 pm
composer.json	30/07/2020 8:47 pm
composer.lock	30/07/2020 8:47 pm

Running the application in the server

Local Development Server

If you have PHP installed locally and you would like to use PHP's built-in development server to serve your application, you may use the `serve` Artisan command. This command will start a development server at `http://localhost:8000:`

```
php artisan serve
```



The screenshot shows the Laravel website homepage. At the top right, there are links for "Log in" and "Register". The main content area features the Laravel logo and four featured sections: "Documentation", "Laracasts", "Laravel News", and "Vibrant Ecosystem". Each section contains a brief description of the resource. At the bottom left, there are links for "Shop" and "Sponsor". At the bottom right, the version information "Laravel v9.52.9 (PHP v8.2.12)" is displayed.

http://127.0.0.1:8000

Log in Register

Laravel

Documentation
Laravel has wonderful, thorough documentation covering every aspect of the framework. Whether you are new to the framework or have previous experience with Laravel, we recommend reading all of the 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.

Shop Sponsor

Laravel v9.52.9 (PHP v8.2.12)

```
C:\> Command Prompt - php artisan serve
```

```
D:\laravel\activity2>php artisan serve  
Laravel development server started: http://127.0.0.1:8000
```

Output

The image shows a VS Code editor with the following components:

- EXPLORER:** A file tree on the left showing the project structure. The `.env` file is selected.
- ACTIVITY2:** A list of files in the current directory, including `artisan`, `composer.json`, `package-lock.json`, and `vite.config.js`.
- .env file:** A list of environment variables:

```
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:mZ+kBfHmWja8qQL0sQKYFh
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATED_CHANNEL=null
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=dbsample
15 DB_USERNAME=root
16 DB_PASSWORD=
17
18 BROADCAST_DRIVER=log
19 CACHE_DRIVER=file
20 FILESYSTEM_DISK=local
21 QUEUE_CONNECTION=sync
22 SESSION_DRIVER=file
23 SESSION_LIFETIME=120
24
25 MEMCACHED_HOST=127.0.0.1
26
27 REDIS_HOST=127.0.0.1
28 REDIS_PASSWORD=null
```
- TERMINAL:** A terminal window at the bottom showing the command `php artisan serve` and its output: `PS D:\laravel\activity2> php artisan serve` followed by an `INFO` message: `Server running on [http://127.0.0.1:8000].`

The terminal output shows the command `php artisan serve` being executed. The output is as follows:

```
PS D:\laravel\activity2> php artisan serve
INFO Server running on [http://127.0.0.1:8000].
Press Ctrl+C to stop the server
2025-03-26 13:44:04 .....
```

The terminal output shows the command `npm run build` being executed. The output is as follows:

```
PS D:\laravel\activity2> npm run build
> build
> vite build
vite v4.3.9 building for production...
✓ 104 modules transformed.
public/build/manifest.json 0.26 kB | gzip: 0.14 kB
public/build/assets/app-bbd6a014.css 225.00 kB | gzip: 30.41 kB
public/build/assets/app-66e7f68a.js 110.84 kB | gzip: 36.02 kB
✓ built in 6.04s
PS D:\laravel\activity2> php artisan serve
INFO Server running on [http://127.0.0.1:8000].
```

← → ↻ ⓘ http://127.0.0.1:8000/login ☆ NEW

Laravel Login Register

Login

Email Address

Password

Remember Me

[Login](#) [Forgot Your Password?](#)

Sample Login page.

(notice there is no .php because of the routing, improving security)

Register

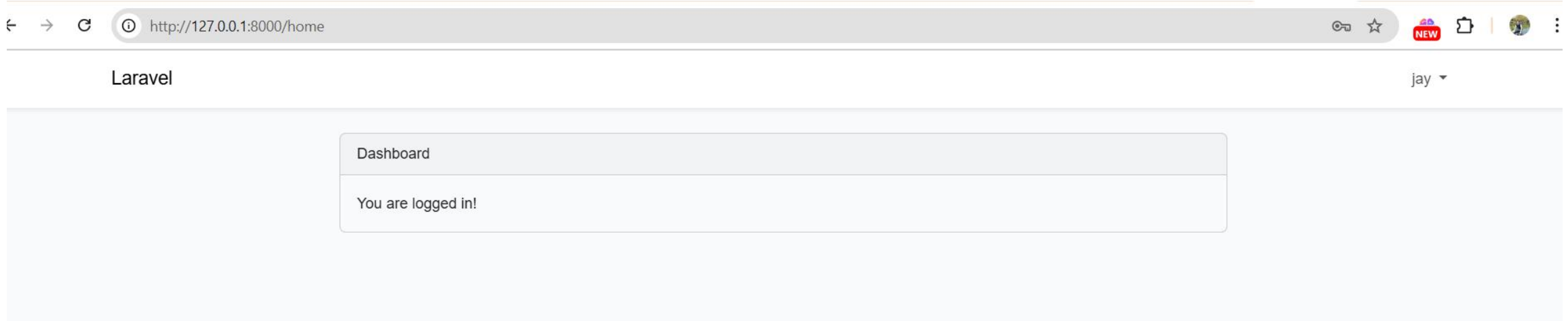
Name

Email Address

Password

Confirm Password

Sample Register page.



Sample/Default dashboard (*successful login*)

End of presentation