**NAME**: Shuvechchha Pun

**LBU ID:** 77261198

**GROUP**: C

**SUBJECT:** ADVANCED WEB ENGINEERING

**COMPONENT 2 –** Laravel: View and Routes

**Laravel**:

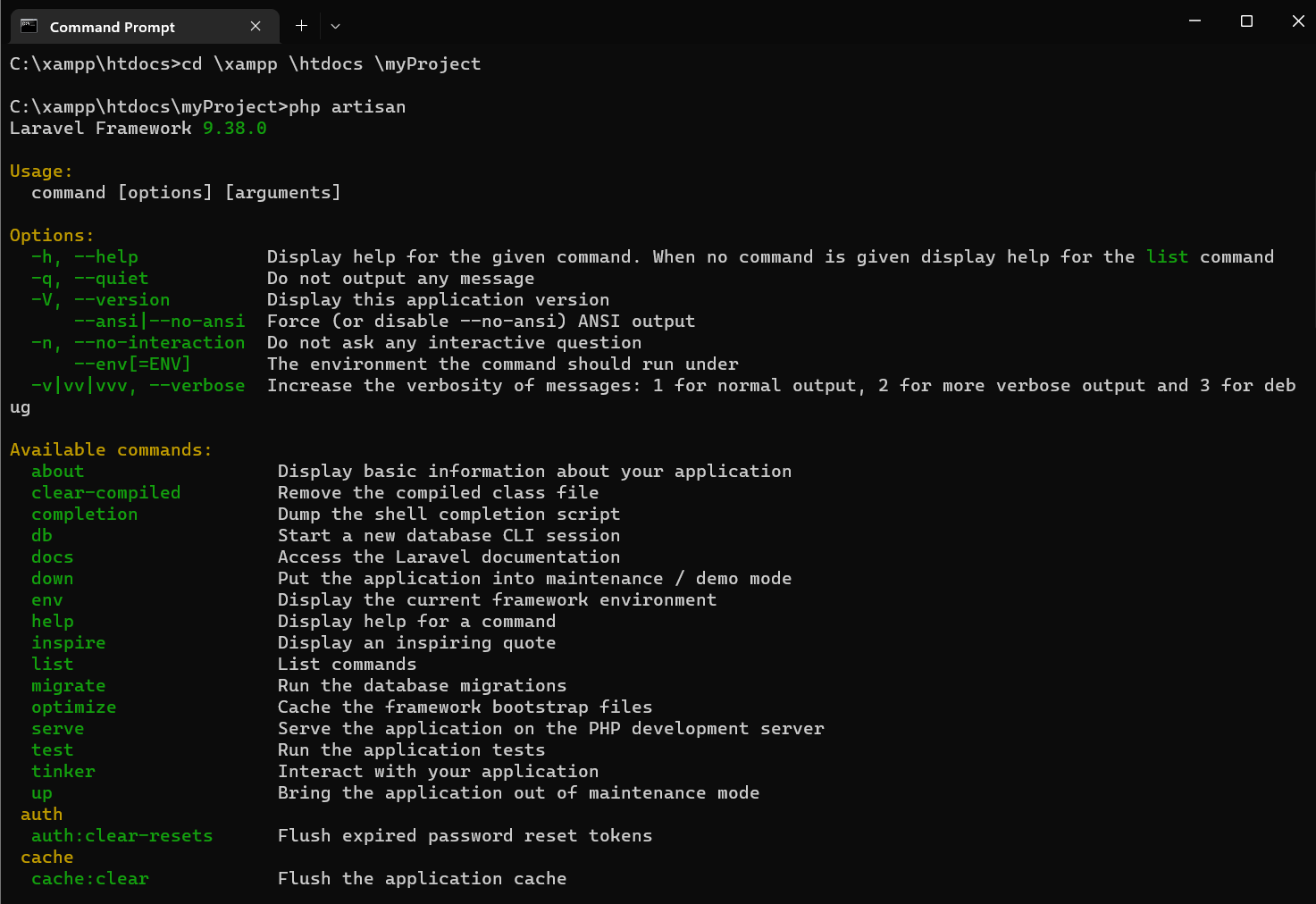
Laravel is a web application framework for Php which follows MVC architecture. Composer is used by Laravel to handle its components. Before using Laravel first we need to make sure to install composer. Composer is a PHP dependency manager.

Installation of Laravel: By issuing this command in terminal.

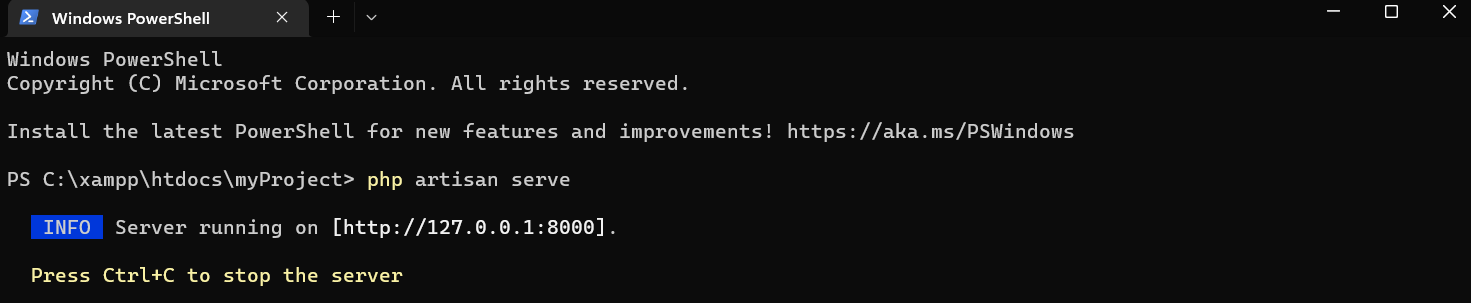
Composer create-project “Laravel/Laravel” myProject

OR

Composer global require ‘Laravel/installer’

Version of Laravel: Php artisan serve command helps running applications on the PHP development server. 

LARAVEL LIVE SERVER:



A screenshot of a computer

Description automatically generated

**MVC:**

The Model-View-Controller (MVC) architectural pattern divides a framework consists of three basic logical components: the model, the view, and the controller. MVC design, divides a database's back-end architecture into logical sections.

Three important MVC component are:

Model: It includes all the data and its related logic.

View: Present data to the user or handles user interaction.

Controller: An interface between Model and View components.

Major benefits of using MVC architecture:

* Simple code that is simple to extend and grow.
* Provides the best test-driven development assistance.
* It works effectively for Web apps that have big teams of web designers and engineers supporting them.

**Routing:**

In Laravel, routing enables one to direct all application requests to the appropriate controller. It means generating a request URL for an application. Routes are created inside the routes folder in Laravel. They are created in the web.php file for websites. APIs are created within api.php.

The structure of the route is very simple. Open the appropriate file (either ‘web.php’ or ‘api.php’) and start the code with ‘Route::’

Laravel offers following route methods: get, post, put, delete, patch

1. STATIC ROUTING:

Route::get ('/about’, function () {

return view('welcome');

};

view: it will navigate to argument view

1. DYNAMIC ROUTING:

ROUTE::GET('post', function () { post=url

return view ('post', [ post=name of the blade file while loading while it goes to URL

'post' => '<h1>Hello world</h1>'

])

post= name of the variable that is call in the URL

Text

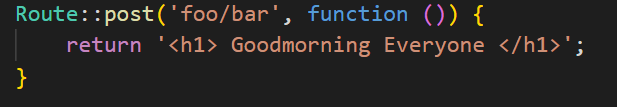
Description automatically generated

GET:

Text

Description automatically generated

POST:

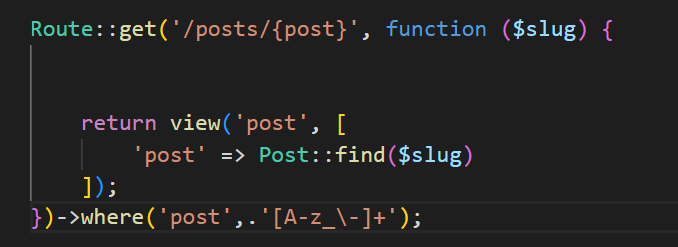


WILDCARD:

Text

Description automatically generated

WILDCARD CONSTRAINTS:



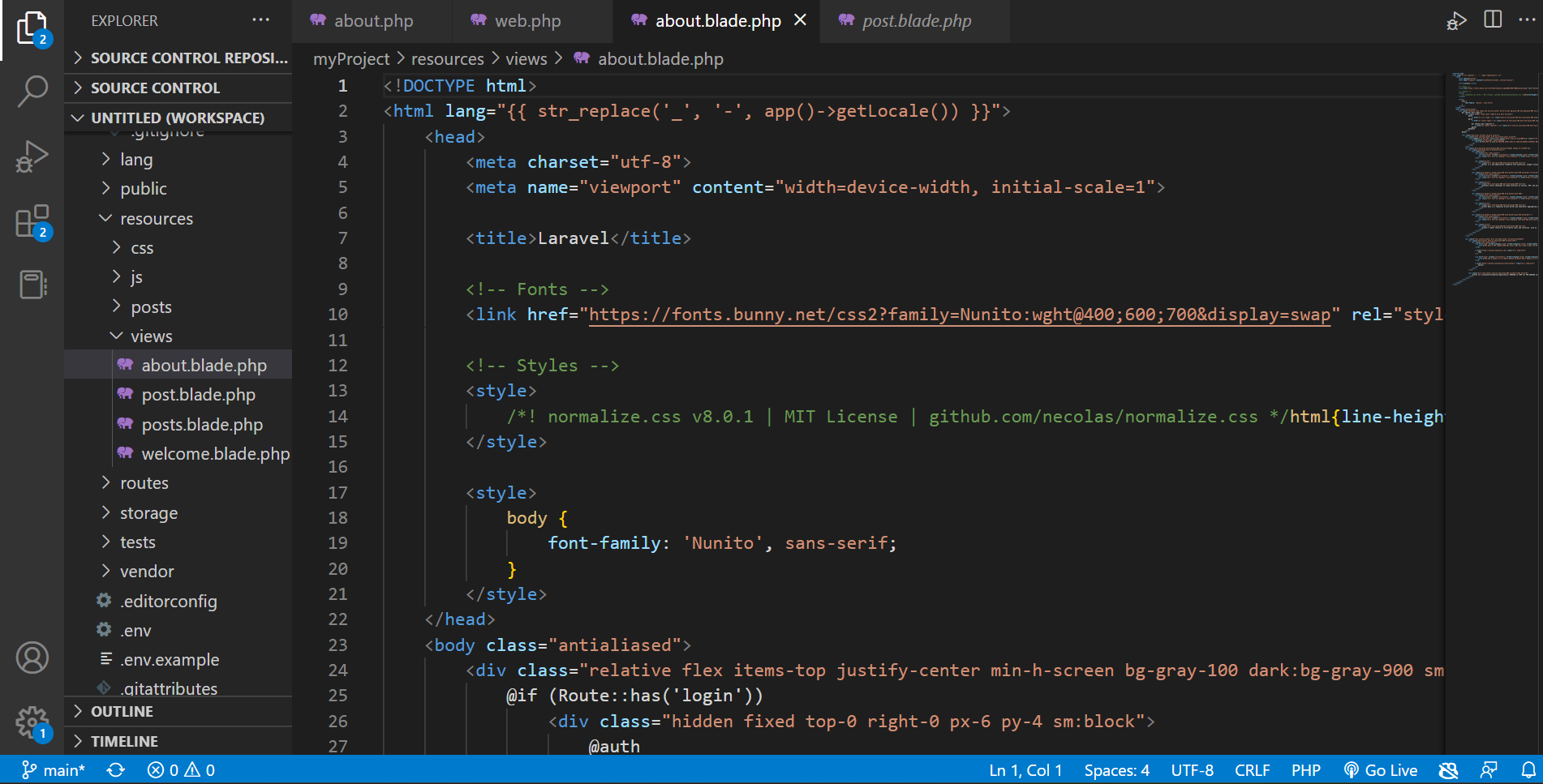
**Version Control**: Version Control is a package that renders database version control for Eloquent Model. Eloquent is a method for performing data queries in Laravel mode. It operates by generating a separate \*\_versions database table that correlates with the model (i.e., users\_versions).

Databases version control is made possible by the foundation provided by version control. It enables you to manage file alterations over time and store the modifications in a database. Additionally, we may quickly get to files anytime we need them. For instance, if we manage user balances and update those balances, we are at a loss as to how to retrieve the previous value of the record. However, the version control package now makes it simple to access previous values.

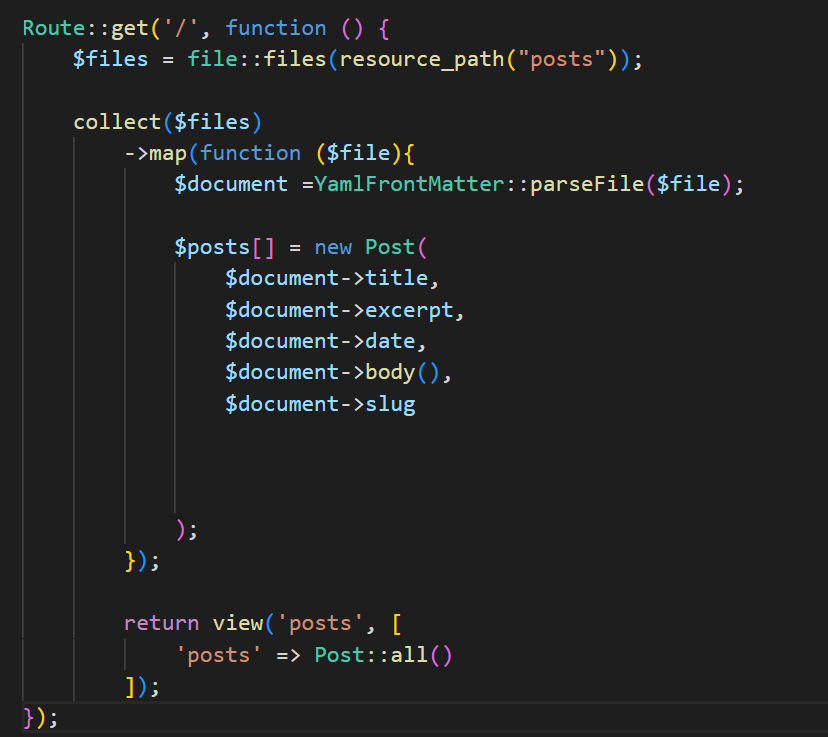
**Blade**:

Numerous website design tasks often begin with layouts. The developer can create elegant HTML-based designs and themes using the Laravel Blade template framework.

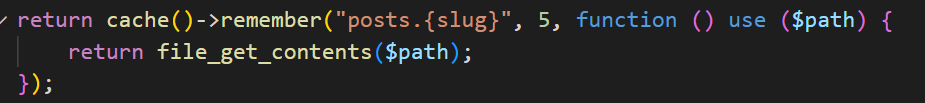
In Laravel, every view is typically constructed using the blade template. Blade engine renders views quickly because it caches them until they are changed. Each file in resources/views ends with.blade.php. Blade is a templating language which Laravel uses by default to create the HTML layout of the website.



COLLECTION:



CACHING:



SORTING:

A screenshot of a computer

Description automatically generated with medium confidence