



**Advanced Web Engineering - Component Two**

**Submitted By : Asmita Thapa Submitted To: Pratik Shrestha**

**University ID : 77261189**

**Laravel**

**Introduction:**

Laravel is a web application framework with expressive, elegant syntax. We believe development must be an enjoyable, creative experience to be truly fulfilling. Laravel attempts to take the pain out of development by easing common tasks used in the majority of web projects, such as:

* [Simple, fast routing engine](https://laravel.com/docs/routing).
* [Powerful dependency injection container](https://laravel.com/docs/container).
* Multiple back-ends for [session](https://laravel.com/docs/session) and [cache](https://laravel.com/docs/cache) storage.
* Database agnostic [schema migrations](https://laravel.com/docs/migrations).
* [Robust background job processing](https://laravel.com/docs/queues).
* [Real-time event broadcasting](https://laravel.com/docs/broadcasting).
* Uses MVC Architecture

Laravel is accessible, yet powerful, providing tools needed for large, robust applications. A superb combination of simplicity, elegance, and innovation gives you a complete toolset required to build any application with which you are tasked.

**MVC Architecture:**

A Model is a representation of a real-life instance or object in our code base. The View represents the interface through which the user interacts with our application. When a user takes an action, the Controller handles the action and updates the Model if necessary. Simply saying, Model; plays with the actual data, View: generate view and UI/UX and Controller; interact with data model

Looking at a simple scenario,

If we open an e-commerce website, you can see the different pages are provided by the View layer. When you click on a particular product to view more, the Controller layer processes the user’s action. This may involve getting data from a data source using the Model layer. The data is then bundled up together and arranged in a View layer and displayed to the user. Rinse and repeat.

CONTROLLER

VIEW

MOEL

Queries

Sends data user actions updates

Fig: MVC Architecture

**Route:**

Route is a way of creating a request URL for your application. It is created inside the routes folder in Laravel. They are created in the web.php file for websites. For APIs, they are created in the api.php.

Additionally, the routes for your web interface are defined in the routes/web.php file.

Route::resource: The Route::resource method can be a Restful Controller that produces all the basic routes required for an application and is dealt via controller class.

When a request matches the specified route URI, the show method on the App\Http\ControllersUserController lesson is invoked, passing the route parameters to the method.

Two routes—one for the web and the other for the API—are included in Laravel's basic installation. The web approach is as follows:

Route::get(‘/’, function () {

return view(‘welcome’);

});

**Blade:**

Blade is a templating tool for creating original layouts. Other views can use the layout created in this way, and it has a unified structure and design.

Blade differs from other templating engines in the following ways:

• It doesn't prevent developers from using standard PHP code in views.

• Until they are updated, the blade views created in this way are compiled and cached.

Laravel defines the child elements using the Blade @extends directive.

The following considerations should be made when expanding a layout. −

• Views specified in the Blade Layout specifically inject the container.

• Different view portions are made as child elements.

• Child elements are saved as child.blade.php files in the layouts folder.

Created aboutus.blade.php page in view folder and coded css file in public folder rather than creating in resources folder. This make it easier for navigation.

