

# Building a Simple Web Application using Laravel

Evita Chandra/13514034<sup>1</sup>

*Computer Science/Informatics*

*School of Electrical Engineering and Informatics*

*Institut Teknologi Bandung, Jl. Ganessa 10 Bandung 40132, Indonesia*

*<sup>1</sup>13514034@std.stei.itb.ac.id*

**Abstract**— Laravel is an open source rapid development web framework, for use in building dynamic website using PHP. It is based on model-view-controller (MVC) development architectural pattern. In this report, we will learn a basic introduction to the Laravel framework and its implementation for building a simple web application. The result proves that using Laravel for building a web application can make things easier because it has a lot of features, including a modular packaging system, easy and many ways for accessing relational databases, utilities that aid in application deployment and maintenance, and many more.

**Keywords**—Laravel, Web Application, framework, MVC

## I. INTRODUCTION

PHP, known as the most popular server-side scripting language in the world with the ability to power everything from personal blogs to hugely popular websites, has evolved a lot since the first inline code snippets appeared in static HTML files. Nowadays, many web developers need to build complex websites and web apps, and sometimes it can take too much time and effort to start from the scratch, that's why many of them choose to use PHP framework as a solution for the problem.

One of the most popular and powerful PHP framework for building a web application is Laravel. Laravel is an open source web development platform under MIT License. What sets it apart is the expressive syntax and its modular packaging systems that has been designed to ease and speed up the web development process. The reason why author decide to learn this framework was because many PHP development companies have adopted Laravel due to its dynamic capabilities to handle big web development projects. According to Google trends, Laravel is the most popular

PHP framework among the developers. Not only that, it provides well-organized, reusable, and maintainable code, supporting products and package, and also promote advance and modern web development practices, such as MVC and object-oriented programming tools.

## II. BASIC LARAVEL FRAMEWORK

First, readers should learn the basic of Laravel framework before starting to build a web application using Laravel framework. The basic of Laravel framework includes content on database migrations, the Eloquent ORM, routing, views.

### 1. Database Migration

Laravel has made processing with database very easy. Laravel's database migrations provide an easy way to define your database table structure and modifications using fluent, expressive PHP code. You can simply run the migrations you push into source control and it will automatically add columns to their local copy of the database. Here is the example command to generate a new database migration [1]:

```
Php artisan make:migration  
create_users_table -create=users
```

The new migration will create users table and will be placed in your `database/migrations` directory. Each migration file name contains a timestamp which allows Laravel to determine the order of the migrations. A migration class contains two methods: `up` and `down`. The `up` method is used to add new tables, columns, or indexes to your database, while the `down` method should simply reverse the operations performed by the `up` method[2].

```
class CreateUsersTable extends Migration  
{  
    public function up()  
    {  
    }  
}
```

```

{
    Schema::create('users', function
(Blueprint $table) {
        $table->increments('id');
        $table->string('name');
        $table->string('airline');
        $table->timestamps();
    });
}

public function down()
{
    Schema::drop('users');
}
}

```

Here is the command to run the immigration.

```
php artisan migrate
```

## 2. Eloquent ORM

Eloquent is Laravel's default ORM (object-relational mapper). Eloquent makes it easy to retrieve and store data in the database using already defined models. Each Eloquent model corresponds directly with a single database table. Here is the command to generate a model.[1]

```
php artisan make:model User
```

The model will be placed in the `app` directory of your application.

## 3. Routing

Routing in Laravel is meant to route your request to an appropriate controller. The routes of the application can be defined in `app/Http/routes.php` file. Here is the general route syntax for each of the possible request.[2]

```

Route::get('user/{id}',
'UserController@show');
Route::post('foo/bar', function () { //});
Route::put('foo/bar', function () { //});
Route::delete('foo/bar', function () {
//});

```

## 4. View

In MVC framework, the letter “V” stands for Views. It separates the application logic and the presentation logic. Views are stored in `resources/views` directory. Generally, the view contains the HTML which will be served by the application.[2]

## 5. Controller

All your request handling is defined in Controller classes. Controllers can group related request handling logic into a single class. Controllers are stored in the `app/Http/Controllers` directory.[2]

```

<?php namespace App\Http\Controllers; use
App\User; use
App\Http\Controllers\Controller; class

```

```

UserController extends Controller{ /**
* Show the profile of the given user. *
* @param int $id * @return Response
*/ public function show($id) {
return view('user.profile', ['user' =>
User::findOrFail($id)]); }
}

```

## III. BUILDING THE WEB APPLICATION

After learning how to use Laravel framework, author started the implementation of it by building a simple web application. The interface of the web application was made by using Bootstrap template and FontAwesome CSS. Both of them are modified as needed.

After creating the front-end, The first thing we need to do is making the database to store all the infos about our site information. In this case we will use a user table, which will store user information. Create and migrate the database into the system. You can defined the database credentials in the `database.php` file located at `config` directory.

After that, we need to migrate all of the database. Here is the example of command line to create migration file for table user.

```

Php artisan make:migration
create_users_table -create=users

```

After that, write code in the migration file and run the migration.

```

1 <?php
2
3 class CreateUsersTable extends Migration
4 {
5     public function up()
6     {
7         Schema::create('users', function (Blueprint $table) {
8             $table->increments('id');
9             $table->string('name');
10            $table->string('email');
11            $table->string('password');
12            $table->string('address');
13            $table->string('airline');
14            $table->timestamps();
15        });
16    }
17
18    public function down()
19    {
20        Schema::drop('users');
21    }
22 }

```

After that, create `user.php`, as a model under the `app` folder. Here is the example of the `user.php`.

```

<?php
use Illuminate\Auth\UserInterface;
use Illuminate\Auth\Reminders\RemindableInterface;

class User extends Eloquent implements UserInterface, RemindableInterface {

    protected $table = 'users';

    |
}

```

After creating a login form and a register form, we need to make a controller to handle request, connect to database

and redirect it to the homepage. We need to code our login and logout functions in the UserController.php file located at app/controllers. The login function should look like the following code.

```
<?php
class UserController extends BaseController {

    public function postLogin()
    {
        $email=Input::get('email');
        $password=Input::get('password');

        if (Auth::attempt(array('email' => $email, 'password' => $password)))
        {
            return Redirect::route('index');
        }else{
            return Redirect::route('index')
                ->with('error','Please check your password & email');
        }
    }

    public function getLogout()
    {
        Auth::logout();
        return Redirect::route('index');
    }
}
```

While the register function should look like this.

```
public function post_new()
{
    // Get the input fields from the form into an array
    $new_user = array(
        'name' => Input::get('name'),
        'password' => Input::get('password')
        'email' => Input::get('email')
        'address' => Input::get('address')
    );

    // Create the array of validation rules
    $rules = array('id' => Input::get('id'),
        'name' => 'required|min:3|max:128|alpha_dash|unique:users',
        'password' => 'required|min:3|max:128'
    );

    // Make the validator
    $validation = Validator::make($new_user, $rules);
    if ( $validation -> fails() )
    {
        return Redirect::back()
            ->with_errors($validation)
            ->with_input();
    }

    // Hash the password
    $new_user['password'] = Hash::make($new_user['password']);

    // create new user and redirect to the login page with a success message
    $user = new User($new_user);
    $user->save();
    return Redirect::to_action('home@login')->with('success_message', true);
}
```

Next, we need to add route to our route file, routes.php, found under apps/http/.

```
Route::get('/', ['middleware' => 'guest', 'uses' => 'BookController@getIndex']);
Route::get('auth/register', 'AuthAuthController@getRegister');
Route::post('auth/register', 'AuthAuthController@postRegister');
Route::get('auth/login', 'AuthAuthController@getLogin');
Route::post('auth/login', 'AuthAuthController@postLogin');
Route::get('auth/logout', 'AuthAuthController@getLogout');
```

So that is it to make a simple web application. All you have to do is use the function in the front-end file, front-end file can be found under apps/views folder.

#### IV. RESULT AND DISCUSSION

These are the results of the web application that we have made from the previous chapter, after implementing

Laravel for building the web application.

#### 1. Register/Login Page

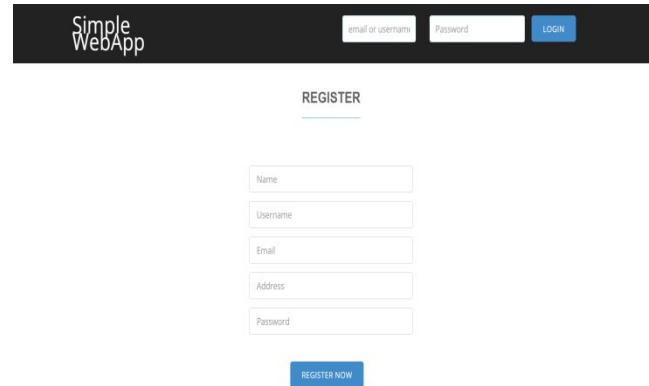


Figure 1 Register/Login Page

The register/login page consists of register form and login bar. The users can login by writing down their email and password in the login bar, the program will check whether they have been registered or not. If they have, it will be redirected to the Home Page. If they haven't been registered or the email/password input are wrong, it will remain on the same page. The user that haven't been registered can write down their information on the register form to sign up. The information that are given are saved in database. The page will be redirected to homepage.

#### 2. Home Page

Logout

### Manajemen User

| Name   | Email          | Address                | Password |
|--------|----------------|------------------------|----------|
| Anto   | anto@gmail.com | Jl. Ir. Djanda Bandung | 0000000  |
| Tono   | tono@gmail.com | Bandung                | 12345678 |
| Hahaha | haha@gmail.com | Jl. Ganesha 15 Bandung | hahaha   |

Figure 2 Home Page

After succeeding at logging in or registering, user will be redirected to the home page where there are information of the users that have been registered in the system. The user can logout by clicking the Logout button at the top.

After doing some exploration about Laravel and learning it, author found that the framework is extremely flexible and has a lot of features that is helpful for building a web application. It also has well-supported and quite a lot documentation, that can be helpful and makes learning it easier.

According to Bull(2012), "If you're looking for a solid,

fresh, new, simple, and well-supported PHP framework then you could do a lot worse than choose Laravel. In fact you couldn't really do a lot better. It's lightweight enough that it doesn't get in the way of your planning and development work. But it does still offer plenty of structure and handy built-in features."

## V. CONCLUSION

Laravel is one of the most popular and powerful PHP framework for building web application. It is based on model-view-controller (MVC) development architectural pattern and has a lot of features, such as a modular packaging system, easy and many ways for accessing relational databases, etc. It also has been designed to ease and speed up the web development process while building a web application.

## VI. ACKNOWLEDGMENT

I would like to express my special thanks of gratitude to the lecturers of IF3280 Socio-Informatics and Professionalism course, Dr.Ir. Rinaldi Munir, MT., Dr. Eng. Ayu Purwarianti, ST.,MT. and Dr. Dessi Puji Lestari, who gave author the opportunity to learn new things while making this project, which also helped author in doing a lot of Research and came to know about so many new things.

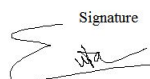
## REFERENCES

- [1] *Basic Task List*. Retrieved May 1, 2017, from <https://laravel.com/docs/5.1/quickstart>
- [2] *Laravel Tutorial* . Retrieved May 1, 2017 from <https://www.tutorialspoint.com/laravel/>
- [3] *How to Craft a Digital e-Shop with Larave*. Retrieved May 3, 2017 from <http://codetutorial.io>
- [4] *Introduction*. Retrieved May 2, 2017 from <https://laravel.com/docs/4.2/introduction/>
- [5] Bull. (2012). *Using Laravel : First Impression*. Retrieved May 4, 2017 from <https://blogs.kent.ac.uk/>

## STATEMENT

This statement shows that I agree that what I wrote is my own writing, not plagiarism or translation from others paper.

Bandung, 5 May 2017

Signature  


Evita Chandra/13514034