Creating a Social Networking Website with Laravel

Scarletta Julia Yapfrine - 13514074
Informatics / Computer Science Major
School of Electrical Engineering and Informatics
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13514074@std.stei.itb.ac.id

***Abstract*—In this era of technology, internet has become an integral part of our life. With the rise of the internet, there are also many kinds of popular websites which emerged. One example of a widely used website is social networking website like Facebook. Laravel is an example of tool or framework used to create a website. With its simple syntax and rich features, Laravel is worth to be learnt. On this project, Laravel is learnt by creating a simple social networking website with it. The conclusion of this project is that Laravel is a suitable framework for creating a website since it has many features that simplify the process of creating a website.**

***Keywords*—laravel, social networking, website**

I. Introduction

Website is one of the core parts of the internet. People use websites to look for information or write the information themselves. There are many kinds of websites which are often used nowadays. Some kinds of websites which are often used by people are personal websites, photo sharing websites such as Instagram, informational websites, e-commerce websites such as Amazon.com, and social networking websites such as Facebook and Twitter.

With the popularity of websites nowadays, it is really important for a programmer or a software engineer to have a basic skill in building a website. There are many frameworks which help in creating a website and simplify the processes. Several back-end frameworks which often used to build a website currently are Ruby on Rails, Laravel, CodeIgniter, and ASP.Net. Each framework is built on different languages. Ruby on Rails is obviously built on Ruby programming language, which is a high-level programming language. Laravel and CodeIgniter are built on PHP, a recursive acronym of PHP: Hypertext Preprocessor. ASP.Net is built on Visual Basic or C# (the users can choose between them). It is really important to be skillful in a web framework which is popular and mainly used right now. But beware, since a framework which is popular and mainly used today, can become obsolete and be left behind in several years due to the rapid development of technology.

On this project, the writer do the exploration of a framework to build a website. The framework that is used is Laravel. It is one of the frameworks that is mainly used right now and it helps to understand about MVC (Model, View, and Controller) logic and OOP (Object-Oriented Programming) in building a website. Bean (2015, p. 10) stated that " …, Laravel will not only make you productive in short term, but it also has the potential to make you a better developer in the long run.". Currently, some examples of websites implementing Laravel framework are OnePlus.net (a technology marketplace website) and Larasocial (a social networking website). The website that is created on this project is social networking website, since it is one of the websites that is popular right now. This paper will discuss about the implementation of Laravel framework in creating a simple social networking website.

II. The Basics of Laravel

Here are some Laravel's basic features from its official documentation. The features listed below are used to create the simple social networking website on this project.

A. Facades

Laravel use facades to provide a "static" interface to classes that are available in the application's service container. It means that Laravel's facades simplify the syntax to call another interface that needs to be implemented in a program. Here is the example of facades

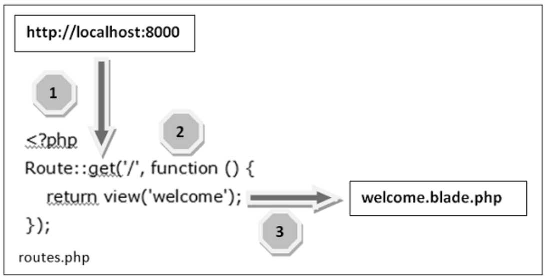
```
use Illuminate\Support\Facades\Cache;

Route::get('/cache', function () {
    return Cache::get('key');
});
```

Above code line 1 is the example of facades. It helps calling the cache interface so that it can be used and implemented in the program.

B. Basic Routing

The most basic Laravel routes simply accept a URI and a Closure. It means that Laravel routing route the request to an appropriate controller to execute the right thing.



```
<?php
Route::get('/', function () {
   return view('welcome');
});
```

Figure II-1. Route Mechanism

On figure II-1, it can be seen that the routing get the method and the root from the executed URL. It then matches with the appropriate method in route.php file. The function in route.php file will then execute its content. Here is the example syntax of routing

```
Route::get('user/{id}', function ($id) {
    return 'User '.$id;
});
```

Above code is an example of routing with parameters. The code inside '{}' in routing parameter is a captured segment of URL.

C. *Middleware*

Middleware is a middle medium between a request and a response. It serves as a filtering mechanism. Middleware is usually used to verify whether the user of the application is authenticated or not. Below is the example of a middleware to handle access control list

```
class RoleMiddleware {
   public function handle($request, Closure $next,
$role) {
      echo "Role: ".$role;
      return $next($request);
   }
}
```

D. *CSRF Protection*

Laravel makes it easy to protect application from cross-site request forgery (CSRF) attacks. Cross-site request forgeries are a type of malicious exploit whereby unauthorized commands are performed on behalf of an authenticated user.

Laravel automatically generates a CSRF "token" for each active user session managed by the application. This token is used to verify that the authenticated user is the one actually making the requests to the application.

Hidden CSRF token field should be included anytime a HTML form is defined in application so that the CSRF protection middleware can validate the request. Below is the example to use CSRF token

```
<form method="POST" action="/profile">
    {{ csrf_field() }}
    ...
</form>
```

E. *Model*

Model is the structure of data. It represents the relations of how the data is stored. Here is the example of model in Laravel

```
namespace App;
use Illuminate\Database\Eloquent\Model;
class Post extends Model
{
    public function user()
    {
        return $this->belongsTo('App\User');
    }
}
```

F. *View*

View is the front-end of the application. It tells how the elements in user interface is organized such as button and form. Here is the example of view in Laravel with blade

```
<html>
   <head>
      <title>@yield('title')</title>
   </head>
   <body>
      @section('sidebar')
      This is the master sidebar.
      @show
   </body>
</html>
```

G. *Controller*

Controller is the middleman between the view and the model.

```
namespace App\Http\Controllers;
use App\Http\Controllers\Controller;
class MyController extends Controller {
   public function index(){
      echo 'index';
   }
   public function create(){
      echo 'create';
   }
}
```

H. *Related Works*

There is currently one example of social networking websites that is built using Laravel. That website is called Larasocial.

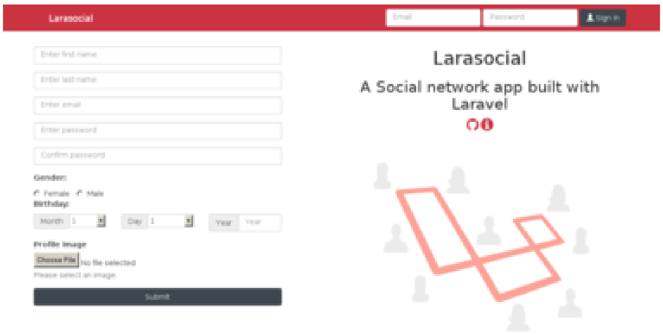


Figure II-2 Larasocial

III. Creating Social Networking Website with Laravel

The website is created gradually from the homepage until the website is finished. Here are the processes in creating simple social networking website with Laravel

A. *Creating the Registration Form*

The registration form on this website consists of email address, name, password, and submit button. There are some validations implemented on this website. The validations are needed because the email that is entered must be unique, all the fields must be filled, password must be at least 4 characters, and name must not be more than 120 characters. The validations are implemented using Laravel validation rules. There are many validation rules provided by Laravel, such as Between, Boolean, Max, Min, Required, Same, and Unique. In the registration form, the validation rules implemented are Required, Min, Max, and Unique. Those validations are implemented in the user controller which connects the user model and the welcome view of the website. The connection among controller, model, and view is implemented in routes.php.

The email, name, and password of the user needs to be uploaded to the database after the form is filled. It is also handled by route and controller. But, the explanation of how the user data is stored in the database exists in the user model and the migration of that model. Before inserting the user's password to the database, the password is hashed first using Bcrypt so that the password is not stored in its original form. After the registration process is completed, the controller will check user's authentication. After that, it will redirect the user to the dashboard page and the user will be logged in.

B. Creating the Login Form

The login form has two fields, namely email and password. There is also a submit button in the form. The function of login form is to be used by a returning user (user who had registered before). There are also some validations implemented in this form. This form implemented Required validation to validate that the email and password fields are filled when the user clicks the submit button to log in.

After the user clicks the login button, the program will check if the email that is entered exists in the database. If itexists, the program will match the entered password with the password in the database. If it is matched, the user will be redirected to the dashboard page. If the email entered does not exist or the password is false, the system will output an error message. The checking of the user's email and password is done with the help of Authentication facade. Same with the registration form, the validation and checking are implemented in the controller. Login form use the same model with registration form, that is user model, since they access the same table in database.

C. Creating the Dashboard

The dashboard page has some elements, which are a textarea for the user to write post, a submit button to submit the post, and a list of other users' posts. If user wants to create a post, the user needs to type the content of the post to the textarea in dashboard page. The post must not be longer than 1000 characters since there is a validation for this in the post controller (with Required and Max). After the user submits the post, the post will appear in the list along with the post from other users.

User can like, dislike, edit, or delete his own posts, but can only like or dislike a post created by another user. That is implemented in the view by using authentication and @if feature in Blade. Like, dislike, edit, and delete is implemented in every post in the form of the link. If the user clicks the link to like, the link will change into "You like this post" and if the user clicks that again, the like will be undone. The same thing goes to dislike link. But, a user cannot like and dislike the same post. If the user does both, then the previous one will be undone. If the user wants to edit his or her own post, a modal will appear and the user can retype the post then saves the change. If the user wants to delete a post previously created by him or her, the user just needs to click the delete link. These features are implemented with the help of AJAX. Dashboard page access likes, users, and posts database tables.

D. Creating the User Page

At the user page, the user see his or her image profile and change it. The user also can change the name of his or her account. The change of the user's name will be saved in the database. But, the user's image is not saved on the database. The user's image can only be seen by the user itself on the user page and is implemented using Storage facade. This page accesses table users in the database. User page also uses user controller,

IV. Result and Discussion

The website is consisted of welcome page, dashboard page, and account page. Below, from point A to C, are the screenshots of the pages

A. Welcome Page

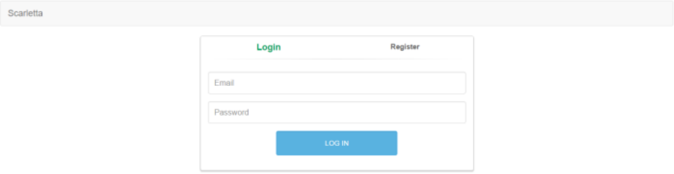


Figure IV-1 Login Form

Figure IV-1 above is the view of the login form. As already stated before, the login form has two fields to fill email and password. If the user finishes filling the fields, he or she can click on the login button. If the entered name and password is correct, user will be redirected to the dashboard page, else the error message will appear.

Below is the view of the registration form. It has three fields, namely email address, name, and password. In order to register, user must fill those three fields, or the error message will appear. The email must not be registered before.

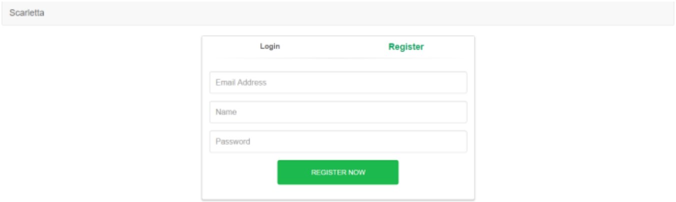
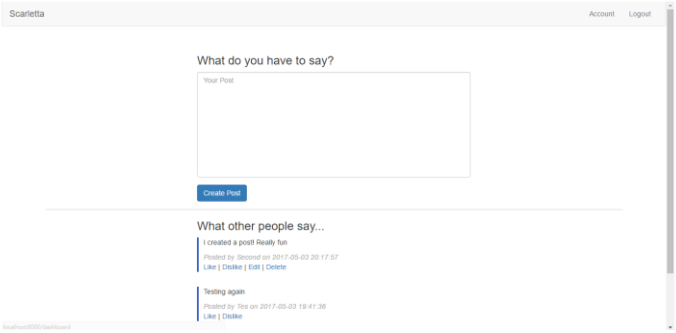


Figure IV-2 Registration Form

B. Dashboard Page



I created a post! Really fun

Posted by Second on 2017-05-03 20:17:57

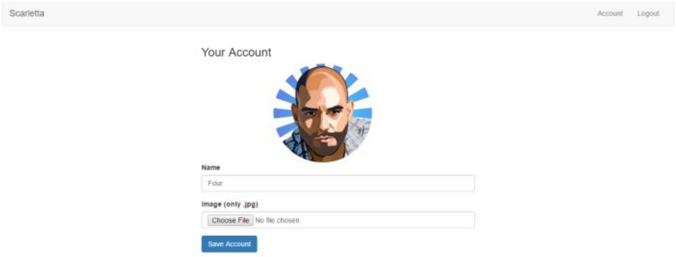
Testing again

Posted by Tes on 2017-05-03 19:41:36

Figure IV-3 Dashboard Page

Above is the view of the dashboard page. As it can be seen, there is a textarea and submit button for posting a new post. Below the submit button, there is a list of posts from every user in this social networking website. User can like, dislike, edit, or delete his or her own posts, but the user is only able to like or dislike the posts from other users.

C. Account Page



Name: Four

Image (only .jpg): No file chosen

Figure IV-4 Account Page

On figure IV-4 above, the account page can be seen. In this page users can set or change their profile pictures. User can also change his or her name.

D. Discussion

Using Laravel framework in building a website is really helping. It has many kinds of facades that can be used to make the program a lot simpler. While using vanilla framework, there are many things need to be done, such as making the JavaScript code to do a validation and making a database queries to retrieve the data and insert it. With Laravel, doing the validation is a lot easier. The database queries are not needed to be made here, since there are models and migrations.

V. Conclusion

Laravel as a framework that is built on PHP programming language is a good framework to build a web application. It uses MVC design pattern which is a popular design pattern to make a website right now. It also uses object-oriented paradigm, a paradigm that is widely used currently in many applications. Laravel has many features such as Facade that simplify the process of programming a web. The processes of validation and authentication is also easy to be implemented here.

VI. Acknowledgment

First of all, the writer wants to thank the Mighty God for the blessing so that the writer can finish this paper. The writer also wants to thank the lecturers of Socio-Informatics and Professionalism course, Dr. Eng. Ayu Purwarianti, ST.,MT., Dr. Dessi Puji Lestari, and Dr. Ir. Rinaldi Munir, M.T. for their teachings and assistances this semester.

References

[1] Bean, Martin. (2015). *Laravel 5 Essentials.* Birmingham: Packt Publishing Ltd

[2] Maximilian. (2016). *Laravel PHP - Create a Social Network (Full App).* Retrieved May 1, 2017, from https://learninglaravel.net/laravel-php-create-a-social-network-full-app

[3] Otwell, Taylor. (2015). *Laravel Documentation.* Retrieved May 1, 2017, from https://laravel.com/docs/5.2

Statement

With this, I declare that this paper is my own creation, not taken, not translated from another person's paper, and not a plagiarism.

Bandung, May 5, 2017

Scarletta Julia Yapfrine – 13514074