

Linh Tang

BUILDING WEBSITES WITH LARAVEL AND VUEJS

Thesis

CENTRIA UNIVERSITY OF APPLIED SCIENCES

Information Technology

October 2019

ABSTRACT

Centria University of Applied Sciences	Date September 2019	Author Linh Tang
Degree programme Information Technology		
Name of thesis BUILDING WEBSITES WITH LARAVEL AND VUEJS		
Instructor Kauko Kolehmainen	Pages 42+2	
Supervisor Kauko Kolehmainen		
<p>The aim of this thesis was to create a website for a virtual coffee shop. Although this shop is not real, its purpose is based on the demand of many companies in the world. The website contributes to managing the work of the owners effectively.</p> <p>The thesis discussed two contents: the first part was to review knowledge about technical requirements such as XAMPP, Laravel, Vue.js. Outstanding features of Laravel and Vue.js are indicated in this part. Laravel is also compared with other PHP frameworks to explain why Laravel was chosen as the main framework for this thesis. The second part was about creating the website. This website was built on Visual Studio Code using Laravel as a backend framework and Vue.js as a frontend framework. Moreover, Vuetify is used to support Vue.js in decorating the web pages.</p> <p>As a result of the thesis work, the website is an example of a website supporting user and order management system of a company.</p>		

<p>Key words HTML, CSS, PHP, Laravel, JavaScript, Vue.js, Vuetify, Website, MySQL, XAMPP.</p>
--

ABSTRACT

CONTENTS

1 INTRODUCTION.....	1
2 REQUIREMENTS FOR BUILDING LOCALHOST.....	2
2.1 Client and Server.....	2
2.2 XAMPP	3
2.3 Installing and configuring MySQL.....	5
3 LARAVEL OVERVIEW	7
3.1 History of PHP frameworks and Laravel	7
3.2 Features of Laravel	8
3.3 Laravel's frontend.....	15
3.4 Compare Laravel with other frameworks	17
3.5 Setting Laravel project	18
4 ADDING VUEJS INTO LARAVEL PROJECT	22
4.1 Introduction and installation	22
4.2 Vue.js overview.....	23
4.3 Vuetify	29
5 MY PROJECT	31
5.1 Idea	31
5.2 Planning	31
5.3 Obstacles	31
5.4 Implementation and Construction.....	32
5.5 The result	37
6 CONCLUSION	42
REFERENCES.....	43

FIGURES

Figure 1. Step 1: Instructions for installing XAMPP on MacOS (Julie 2012).	4
Figure 2. Step 2: Instructions for installing XAMPP on MacOS (Julie 2012).	4
Figure 3. Installing MySQL (Julie 2012).	5
Figure 4. The request/response life cycle (Stauffer 2017).	13
Figure 5. The devDependencies property (Gore 2017).	16
Figure 6. The searching trend of PHP Framework (Screenshot from Google Trend).	17
Figure 7. The command to check PHP version (Saunier 2014).	19
Figure 8. The command to install Composer on MacOS (Saunier 2014).	19
Figure 9. The command to check PHP version and Composer on Windows (Saunier 2014).	19
Figure 10. The command to create a Laravel project (Saunier 2014).	20
Figure 11. The sekeleton of a Laravel application (Stauffer 2017).	20
Figure 12. The npm commands (Gore 2017).	22
Figure 13. The example for template of Vue (Nelson 2018).	23
Figure 14. The error caused by the name of properties of the data (Nelson 2018).	24
Figure 15. An example for Watcher (Macrae 2018).	25
Figure 16. The example for a Vue component (Nelson 2018).	26
Figure 17. An example for a prop (Nelson 2018).	27
Figure 18. Importing Vuex (Nelson 2018).	27
Figure 19. An example for a scoped CSS in Vue (Macrae 2018).	28
Figure 20. An example for scoped CSS in Vue (Macrae 2018).	29
Figure 21. The code of Vuetify is added into JavaScript file (Wanyoike, 2019)	30
Figure 22 The command to create Laravel project	32
Figure 23 Modifying .env file (Screenshot from Visual Studio Code)	32
Figure 24. The picture of Sequel Pro	33
Figure 25. The login form (Screenshot from Visual Studio Code)	33
Figure 26. The content of web.php (Screenshot from Visual Studio Code)	34
Figure 27. The contents of app.js (Screenshot from Visual Studio Code)	35
Figure 28. The web.php was modified (Screenshot from Visual Studio Code).	36
Figure 29. The list of Chat Controller files (Screenshot from Visual Studio Code)	36
Figure 30. The result of the home page	37
Figure 31. The picture of login form.	38
Figure 32. The result of admin page.	38

Figure 33. The result of user management page.	39
Figure 34. The result of branch page	39
Figure 35. The order form of customers.	40
Figure 36. The forum page.....	40
Figure 37 a. The messages page of Branch2. b. The messages page of admin	41

TABLES

Table 1. The methods of resource controller (Stauffer 2017).....	10
--	----

1 INTRODUCTION

Nowadays, the Internet has become an important part of modern life because of its large benefits. A well-maintained website with an interesting user interface can help an enterprise get a competitive advantage in its business and improve its reputation. Besides, a website can save the budget of the enterprise on marketing campaigns when its brochure and catalog can be updated or changed easily. The Internet allows businesses to expand their market to the world without geographical barriers and get access to new customers.

There are many methods to build a website. PHP or hypertext preprocessor is one of the popular languages in supplying a basic structure for the development of websites. PHP frameworks support developers to speed up the development process by well organized, reusable and maintaining code. Modern PHP frameworks containing Laravel, Symphony, Zend Framework, CodeIgniter, Cake2 and Phalcon follow the Model View Control (MVC) pattern that separates presentation, logic, and advanced web development practices. Laravel is a new PHP framework; however, it is the most popular in building rapid website or application development. Laravel has expressive and elegant syntaxes with full document for each released version, which helps beginners and developers learn and apply them into real projects easily. Laravel also provides the Artisan command-line interface which includes a number of useful commands when developing applications.

Moreover, Laravel allows developers to use Bootstrap and JavaScript as frontend packages which are installed by using NPM command to design User interface. Laravel has some basic JavaScript scaffolding to build websites by using Vue.js library. Laravel and Vue.js is a great combination so that developers can build a completed full-stack website. Vue.js contains some useful libraries such as vue-router, vuex and axios. On the other hand, developers can choose one kind of Vue.js framework to save their time in designing. This thesis selected Vuetify which has sufficient documentation and free interesting themes on its websites.

This thesis aims to discuss useful knowledge and skills of Laravel as well as Vue.js in order to build a full-stack website. The project focused on two web pages designed for a coffee shop. These web pages are user management website for the admin role and an order management website for another role. Through Laravel and Vue.js, users can access web pages through an interactive and modern user interface.

2 REQUIREMENTS FOR BUILDING LOCALHOST

Each machine on the Internet has a certain responsibility such as server or client. A server is a machine supplying services to others while a client is a machine connecting to the server to use data or services. Client and server process can be designed in the same machine or different machines communicated by a network (Yadav 2009.). A Web page is a part of the Internet when it is one of the ways to share information on the Internet as same as email, File Transfer, Protocol (FTP), and instant messaging services (Gil 2019). To understand how Websites work, section 2.1 introduced the basic knowledge of client and server; then section 2.2 provided necessary steps to install a local server and to use it.

2.1 Client and server

User's computer or browser is seen as the client. Clients have specific features such as being controlled by a user individually. That means the user can choose any connection or machine depending on user's demand without any control. Clients have temporary connections with temporarily assigned addresses which can be altered. Diversity resources including multimedia capabilities, a mouse, and real-time bi-lateralism with the user are designed for clients. One of the client's important characters is that limits changes of the local file system when a program gets to access the Internet. Most client programs are protected in a sort of "sandbox" against malicious coding. Moreover, clients are permitted to turn off or move without penalty. When building on the client, developers need to work with XHTML, CSS, and JavaScript by using Web browsers, browser extensions, text editor and integrated development environment as crucial tools. Web browsers are used for showing the result of the programs (Harris & McCulloh 2008, 803). Chrome is one kind of prominent Web browser because of its various extensions; hence, it was chosen for the project of this thesis. Browser extensions of Chrome assemble bundles of HTML, CSS, JavaScript, images, and other files of the web platform. With a wide range of functional possibilities of these extensions, developers can modify the content and design of the user interface (Mehta, 2016.). Developers choose a certain text editor to write plain text files depending on their purpose and their operating system (Harris & McCulloh 2008, 804-805.). This project was created by Visual Studio Code with fulfilling relevant syntax, code coloring, and other factors in order to work with HTML, CSS, JavaScript and PHP.

On the other hand, servers, which are the equipment hosting the Web pages, are different from clients at some points. Firstly, there are several server administrators to control servers so that all data on the servers is secure. Next, the connection of servers is permanent with stable IP addresses because servers

take responsibility in receiving a request from clients. Server administrators have the burden of dubbing server's name which is called domain name to support users in finding a website easily. Web servers usually communicate with other programs or machines such as data servers. When a server stops working, an error alert is shown on the screen to notify users who try to get access to web pages. This means that servers are reliable. Unix and Linux, which are stable operating systems, are preferred to run Web servers. The last feature of servers is having specialized software which is a fundamental factor in converting a computer into a server. There are two options, which were born early, Apache and Microsoft IIS (Harris & McCulloh 2008, 804.). Moreover, server-side needs complex items to begin working. These items are a web server, a server-side language, a data server, a mail server, an FTP server and phpMyAdmin. In detail, a web server has a mission that allows users to request Web pages; developers can choose signing on to a hosting service to use its server or install their own. Various languages called server-side languages are provided to support server functionality (Harris & McCulloh 2008, 805.). PHP language is preferred to build websites in this thesis work because it has enough vital functions, speed, and free price. Almost websites have to manage data; hence, the data server has survived to handle them. MySQL, which has many advantages such as free, powerful, flexible, is the best choice in the open-source world. In addition, phpMyAdmin is a program to get access to MySQL databases. In some cases, websites need the mail server to deal with sending and receiving email and FTP server to allow to send files remotely. (Harris & McCulloh 2008, 805-806.)

2.2 XAMPP

XAMPP is one kind of Apache distribution with several web development technologies in a single package. Its advantage is small size, free price, and portability; therefore, XAMPP is an excellent tool for students in practicing and testing applications in PHP, MySQL. There are 4 different packages and 2 add-ons including XAMPP Basic Package, XAMPP Development Package, XAMPP Upgrade Package, XAMPP Lite, Perl add-ons, and Tomcat add-ons downloaded through Installer, ZIP Archive and Self-extracting ZIP archive. The packages can be downloaded in multiple ways, through Installer ZIP Archive Self-extracting ZIP archive (Murphy 2008.). XAMPP has some features containing Apache, PHP, MySQL, phpAdmin, Mercury Mail, File Zilla FTP server, php libraries, additional languages which are Perl and SQLite, control and configuration tool which are known as a control panel to alter components on or off. (Harris & McCulloh 2008, 806.)

XAMPP accounts for only 34MB with an available version for Window, Linux, Mac OS and the Solaris operating system (Harris & McCulloh 2008, 806). Firstly, developers need to access to

<https://www.apachefriends.org/index.html>. This page has a link to go to the XAMPP version following their operating system. This thesis discusses how to download XAMPP for MacOS. After selecting the Installer version for macOS, Installer link leads to SourceForge where the download file is (Julie 2012.). The file begins downloading automatically and causes the result like Figure 1.



Figure 1. Step 1: Instructions for installing XAMPP on MacOS (Julie 2012).

The newest XAMPP version for macOS is 7.3.9 displayed in the name of XAMPP. Users can choose an older version; however, the process is similar. After XAMPP version is dragged into the applications folder, all folder and files of XAMPP are copied into /Application/XAMPP as shown as Figure 2. Developers can find a path to open the XAMPP control panel. XAMPP control panel allows developers to start and stop Apache and MySQL server running on macOS quickly. Due to the support of these servers for development purposes only, they are turned on only when users need. The <http://localhost/xampp/index.php> link is utilized to test whether the web server is running. (Julie 2012.)

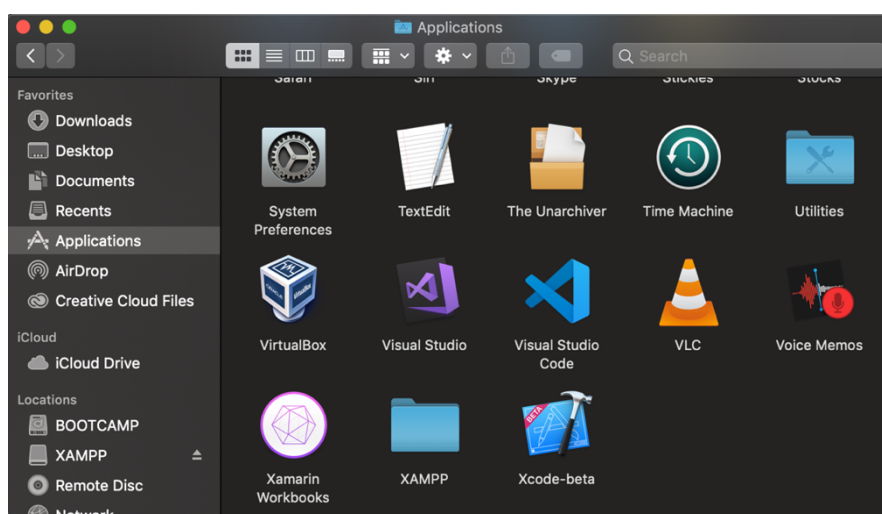


Figure 2. Step 2: Instructions for installing XAMPP on MacOS (Julie 2012).

Although XAMPP provides a quick and easy method for Apache, MySQL, and PHP, it has a big disadvantage about security if users do not config to reduce risks. For example, MySQL administrator user has no password in default; MySQL and Apache are the same users and especially, some services can access to the network unless the user's machine opens personal firewall. In order to overcome this harm, XAMPP adds a utility for each operating system. This utility is active in different ways depending on the kind of operating system. Linux/UNIX uses "# /opt/lampp/lampp security" command while in Windows, users need to open the Security Console by navigating to <http://localhost/xampp/index.php> and select Security from the menu. It is easy for Mac to start a terminal window and type "# /Applications/XAMPP/xamppfiles/xampp security" command. (Julie 2012.)

2.3 Installing and configuring MySQL

The next important step is downloading MySQL after installing XAMPP. The newest version of MySQL for macOS is 8.0.17 with 301.2M from <https://dev.mysql.com/downloads/mysql/>. MySQL is released by MySQL AB company that develop, maintain and distribute MySQL database server. This thesis focuses on only MySQL version for macOS. After clicking the suitable link to download the file for macOS, users can open the DMG archive and see a folder with a few files in it, as shown in Figure 3. The next step is doubling click the *.pkg and following the installation steps to complete the process. (Julie 2012.)

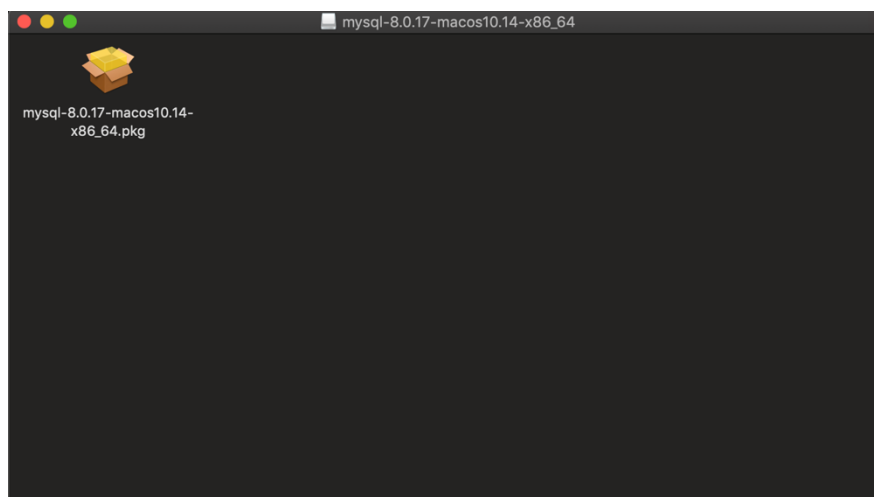


Figure 3. Installing MySQL (Julie 2012)

Security of MySQL is established when a server is started and only the administrator has permission to change this rule; however, users can check it and report all errors to the Internet service provider. A

gravity problem needs to be concerned. It is that the owner of MySQL daemon is non-root so that the malicious access is limited individually to gain connect to the server and destroy files. There are several ways to control the database by MySQL such as a command-line interface or MySQL applications. Each solution has different considerations because of its security risks. As regards the MySQL installation in the workstation of users, there are some problems in a public area unless they improve their vigilance by using screensaver or lock screen mechanism with a password. The reason is that any stranger can change their database directly when the users are absented. On the other hand, MySQL on a server outside the user's network has other hidden risks. They work on the Internet; therefore, it is hard to avoid the case of unencrypted and intercepted data by thieves who steal pieces of data and then match them together. That problem is solved by communicating MySQL through a secure connection such as Secure Shell (SSH) which takes responsibility in encrypting all data transmission to and from the remote machine, and phpMyAdmin which is used for web-based administration interface. (Julie 2012.)

3 LARAVEL OVERVIEW

Nowadays, web developers have more responsibility in building a more different web application than in the past because their code is mandated to be the unique business logic and contain common components such as user authentication, input validation, database access and templating. Therefore, a range of application development frameworks with various available components and libraries have been established. Laravel is one kind of these frameworks of PHP language. There are several benefits Laravel brings to programmers including configuration files, service providers, prescribed directory structures, and application bootstraps. This is the reason for using a framework in general and Laravel in particular. Before researching about Laravel and its useful features in section 3.2 and 3.3, section 3.1 exploited the history of PHP and Laravel.

3.1 History of PHP frameworks and Laravel

PHP is one of the programming languages which has the weakest entry barriers because it is easy to set up on all kinds of desktop operating systems and easy to access for new programmers who have some knowledge about HTML, inline conditions and statement. PHP contains diverse functions for building a website. However, in some case, it usually causes a dissatisfied feeling for newcomers when coding becomes confused and complex after they add more features to their project. This is a strong motivation to create more modern web application frameworks (Saunier 2014.). Web application frameworks are developing dramatically to catch up with the lightning pace of Information Technology in general and the Internet in specific. PHP framework is no exception, it started to release several new frameworks to adapt to the programmer's demand. The first PHP framework is CakePHP which appeared in 2005 and was followed by Symfony, CodeIgniter, Zend Framework and Kohana. Yii came in 2008 and the next years were the presentation of Aura, Slim and especially FuelPHP and Laravel was born in 2011. Each framework focuses on useful functions, some of them concentrate on database object-relational mappers (ORMs), Model View Control (MVC) while other concerns about rapid development and enterprise design patterns and e-commerce such as Symfony and Zend. (Stauffer 2017.)

CakePHP and CodeIgniter, which are the earliest PHP frameworks, were designed from Rails. CodeIgniter was simple to use with excellent detailed documentation and became the best choice in the independent PHP frameworks in 2010. However, its weakness was that its modern technology and patterns developed slowly while other frameworks were constructed. Therefore, they are advancing effectively

and surpassing CodeIgniter in technology and outstanding features. In 2010, Taylor Otwell decided to write another framework to limit all disadvantages of CodeIgniter. This framework is Laravel. Laravel 1 based on scratch was allowed to utility in June 2011. Its characters include a custom ORM (Eloquent), closure-based routing, a module system for extension and supporters for form, validation and authentication. Other features such as controllers, unit testing, command-line tool, inversion of control container, Eloquent relationships and migrations are added into Laravel 2 and 3 which were built in November 2011 and February 2012, respectively. After these versions, Taylor realized that a package manager known as component became valuable in modern industry; therefore, he decided to rewrite the framework with Composer which distributes and bundles components. Before releasing Laravel 4, Taylor created Illuminate components in May 2013. Most components of Laravel 4, which were provided by Composer, are a combination of Symfony framework and Illuminate components. This leads to the dependence of Laravel 4 on releasing Symfony every six months (Stauffer 2017.). Based on Symfony and its various libraries, Laravel 4 advanced and extended its components. For example, SwiftMailer was applied for straightforward emailing, Carbon was added to expressive data and time handling, Doctrine was used to change its inflector and database abstraction tools. Instead of working itself, Laravel has strong support from Symfony (Saunier 2014). Besides, Laravel 4 also brings queues, mail components, facades, and database seeding. The newest version of Laravel is Laravel 5 released in February 2015. With Laravel 5, the directory structure is refurbished, the form and HTML helpers are removed. Moreover, the contract interfaces, new views, Socialite for social media, authentication, Elixir for asset compilation, Scheduler for environment management, form requests and REPL (read-evaluate-print loop) are added into this version. (Bean 2015.)

3.2 Features of Laravel

The main features and sources have contributed in forming a Laravel project are modularity, testability, routing, configuration management, query builder and Object-relational mapper (ORM), schema builder, template engine, email, authentication, Redis, and queues. Laravel is separated into individual modules and cooperated with Composer Dependency Manager in order to update quickly. Several helpers survive in Laravel support programmers in getting access routes from their test, collecting HTML, guaranteeing for method and even disguising users. Routing is an important factor of Laravel giving developers flexible functions such as GET, POST, PUT, DELETE and filter functions. Each program is operated in a different environment. The differences are about database or email server credentials setting and error forms. Programmers can have freedom in deciding the setting following the demands. Query builder helps PHP syntax handle database queries without writing SQL. The database scheme is

defined with a PHP code by the assist of schema builder and it is also monitored with database migrations. A specific point in Laravel is a template which is shipped with Blade. SwiftMailer library brings several benefits for Laravel in sending an email with content and attachments. Besides, Laravel arranges the registration, authentication and password reminder tools to supply for users. Users also connect to Redis, an in-memory key-value store, then Redis becomes a session or general-purpose cache. In the situation that programs need to send emails to an enormous number of users, the queues allow them to run these emails in the background instead of keeping users waiting until the task is completed. (Saunier 2014.)

In detail, Laravel programs have no ability to connect with the end user without routes which take responsibility in taking requests from users and delivering responses via HTTPs. Laravel divides routes into two kinds that are web routes in `routes/web.php` and API in `routes/api.php`. Web routes are visited by end users while API routes are developer's API. The way to define a route is to link a path (/) with a closure. Closures are defined as a function that is used to assign a variable and to pass as a parameter to other functions and methods. When customer visits / (the root of the application's domain), Laravel's router runs this closure and responses the result. Some route verbs are accepted in Laravel containing `Route::get`, `Route::put`, `Route::post`, `Route::delete` which allow programs to only match when HTTP request uses the GET, PUT, POST, DELETE action respectively. Beside closures, Laravel also handles routes with the clumsier. The clumsier demonstrates its abilities in larger applications, especially in applications which have to solve hundreds of milliseconds off of each request because it can put all routes in a file. The syntax to define parameters for routes and pass them to the closure are extremely simple. A question mark can be added after the parameter name to arbitrary choice route parameters; however, programs need to set a default value for the route's corresponding variable. Regular expressions are also a way to define a route with a parameter placed on particular requirements. The convenient function which Laravel provides for users is of naming routes. It is useful for users to call complex routes by simple nicknames. Moreover, they no longer spend more time on rewriting the links after changing the paths. The next character of the route in Laravel is the route group. All routes in a group can share several features such as certain authentication, path prefix, controller namespace, and configuration settings. Thanks to this connection, the project can be reduced to some duplications. The route of Laravel in a file is shown as `view()`. Views are not similar to Model view control pattern; however, they are files which have tasks to perform the output result such as HTML in the case of website. Plain PHP and Blade templates are two formats of Laravel's views. It is no doubt that controllers have important missions in organizing routes in one place by grouping similar routes together. An application formatted following CRUD is a good example to research the key role of controllers because

they handle all actions in this application. Controllers ensure the traffic of HTTP requests of the application. In addition, controllers can get user input and work on it by the Input facade or the Request object. In order to save time in naming the methods, Laravel has a traditional REST/CRUD controller called a resource controller with some effective rules for routes. The methods of Laravel's resource controllers are shown in Table 1. (Stauffer 2017.)

Table 1. The methods of resource controller (Stauffer 2017)

Verb	URL	Controller method	Name	Description
GET	tasks	index()	tasks.index	Show all tasks
GET	tasks/create	create()	tasks.create	Show the create task form
POST	tasks	store()	tasks.store	Accept form submission from the create task form
GET	tasks/{task}	show()	tasks.show	Show one task
GET	tasks/{task}/edit	edit()	tasks.edit	Edit one task
PUT/PATCH	tasks/{task}	update()	tasks.update	Accept form submission from the edit task form
DELETE	tasks/{task}	destroy()	tasks.destroy	Delete one task

In the case of route model binding, a particular parameter name is used to support the route resolver to search an Eloquent record corresponded with its ID; and pass it in a parameter as an alternative of passing only the ID. There are two options: implicit and custom. Except for views, Laravel also provides `redirect` to return from a controller method or route definition. Developers can generate a redirect by using the `redirect` global helper or the facade. Some methods of `redirect` are `redirect() → to()`, `redirect() → route()`, `redirect() → back()`, `redirect() → with()`, `home()`, `refresh()`, `away()`, `secure()` and `action()`. Laravel supports several abort methods to exit a route. They are `abort()`, `abort_if()` and `abort_unless()`. They get HTTP status codes, a message and a headers array. After views, redirects and aborts, programmers also have to understand about responses. There are `response() → make()`, `response() → json()` and `→ jsonp()`, `response() → download()` and `→ file()`. (Stauffer 2017.)

As mentioned above, Blade is a kind of template of Laravel. It is developed from .NET's Razor engine. Blade's syntax is clear, concise, and powerful. Especially, Blade promotes its strength when the developer's programs become complex caused by many nested inheritances and difficult conditionals. After compiling Blade syntax into PHP code and caching, programmers can use native PHP in these Blade files. It is easy to use Blade because its control structures are similar to PHP's. In addition, a Blade

structure allows views to extend, modify and contain other views as template inheritance. In Blade, `@if` and `@unless` are called directives matched between a pattern and a PHP output. (Stauffer 2017.)

Laravel focuses on some tools such as Artisan, Tinker and the installer for the command-line interaction. Artisan is a combination of command-line actions and adding ability. The list of Artisan commands of an application depends on a package and specific code of the application. Except Artisan commands running from the command-line, developers can use them from other code via Artisan facade, for example, `Artisan::call()` and `Artisan::queue()`. The second tool of Laravel is Tinker which is also called REPL (read-eval-print loop). It mimics the command-line prompt so that the results are printed out of the screen after programmers type their command into the REPL. It is an excellent tool for database interaction, for testing new ideas and for running code snippets. (Stauffer 2017.)

The database of Laravel has some special functions for instance migrations, seeders and the query builder. A migration is a single file which determines the modifications when this migration runs up and this migration runs down. There are some basic commands for migrations that are `migrate:install`, `migrate:reset`, `migrate:refresh`, `migrate:rollback` and `migrate:status` with different missions. With respect to seeder, seeding is a simple action with `DatabaseSeeder` class which is called the seeder through `run()` method in `database/seeds` folder. The seeders can be run with migration or separately. After connecting, migrating and seeding the tables, developers need a fluent interface to interact with their database. This interface is the query builder. The query builder attends to change the raw SQL queries as a basic escape to ensure the application's security (Stauffer 2017.). A great feature of Laravel which needs to be discussed in this thesis is Eloquent. Eloquent is the main factor that composes of the difference among Laravel and other PHP frameworks because of its simplicity in the complexity of DataMapper ORMs world. Eloquent handles the database through some actions such as retrieving, representing, and persisting in its table through minimal code. Moreover, it can communicate with all the data in general, each table row in particular. (Malatesta 2015.)

Another function Laravel brings to programmers is a basic user authentication containing registration, login, sessions, password resets, and access permissions. Hence, they can save more time when building an application. This system is easy to use and apply in several different settings. To work with the authenticated user of an application, developers can interact with the `auth()` global helper, or insert an instance of `Illuminate\Auth\AuthManager`, or deal with the Auth facade. Some checks are utilized such as `auth() → check()`, `auth() → guest()`, `auth() → user()`, and `auth() → id()`. In the new project of Laravel,

they includes a `create_users_table` migration, a `User` model and available controllers such as a `RegisterController`, a `LoginController`, a `ForgotPasswordController`, and a `ResetPasswordController`. Thanks to these controllers, developers no longer take more time in the code which creates the completed application. The `RegisterController` show a registration form with validated input. After successful registration, users are directed to the web page the developers want through `$redirectTo` property and it is no doubt that new users are added into the application's database. The functions of `LoginController` are exactly as its name, that means allowing users to log in. The `LoginController` contains `AuthenticatesUsers` trait which is divided into the `RedirectsUsers` and `ThrottlesLogins` traits. The same point between the `LoginController` and the `RegisterController` is owning `$redirectTo` property. Especially, the `AuthenticatesUsers` trait supplies the login form, input validation, error notification, logout function and redirection for users. Developers are free in designing the User interface for login form by overriding the default `showLoginForm()` method, modifying the validation in the `validateLogin()` method, and performing some notifications for a successful login in the empty `authenticated()` method. The `ThrottlesLogins` trait has different responsibilities from the `AuthenticatesUsers` trait. The function of `ThrottlesLogins` trait is to reduce the login ability of users when there are too many failed logins within a short time. In fact, the `ThrottlesLogins` allows failed login 5 times per 60 seconds. It counts the fail login time of username and IP address combination by the cache. After that, it directs users back to the login page with notification. However, this function is not available in Laravel 5.1. The login form cannot lack reset password function therefore, `ResetPasswordController` exists in `ResetsPasswords` trait to supply validation and access to the basic password reset views. This trait also contains `PasswordBroker` class, which manages password reset emails and resets the passwords by `resetPassword()`, `broker()`, and `guard()` methods. Moreover, `ForgotPasswordController` uses `SendsPasswordResetEmails` trait. This trait deals with `showLinkRequestForm()` method, `sendResetLinkEmail()` method. Besides the above controllers, Laravel application needs a tool called as `Auth::routes()` to package all routes to its routes file. Laravel 5.2 was added a new tool with more skeleton code so that the authentication of the application running more quickly. That is the auth scaffold. The auth scaffold adds `Auth::routes()` to the routes file, designs a view for each route, and includes a `HomeController`. Moreover, `Auth` also provides route middleware to check each route in the application and reduces routes to guest or users. They contain `auth`, `auth.basic`, `guest` which restricting access to authenticated users with and without HTTP Basic authentication, to unauthenticated users respectively. A combination between a driver determining the authentication state and provider allowing to create a user by certain criteria is known as a guard. `Web` and `api.web` are two kinds of traditional authentication styles. Both the driver and the provider can be changed when programmers want to alter users' identification and persistence; and alter the storage type or retrieval methods. Websites often show some error such as 403, and 404 error. These alerts were started from the access control list (ACL)

system in order to determine the authorization of the users. This means that they notify to users whether they are able to access these websites. Through Gate facade with some verbs such as can, cannot, allows, and denies, users can check their authorization. Excepts Gate facade, the user mode, middleware, and Blade directives also own some convenience helpers to execute this task. (Stauffer 2017.)

The knowledge about the request and response object of a Laravel application is crucial. A request which is produced by an HTTP request or a command-line interaction is transmitted to Illuminate Request object. After passing the request through many layers, the Illuminate Response object receives and handles it to create a result. This result is back out to the above layers to the end-user. The request/response lifecycle is shown in Figure 4. Every web request is set configuration in a .htaccess file or an Nginx configuration setting at the web server level and routed to public/index.php in the Laravel application directory. In all files of the Laravel application, the kernel file plays the key role because it receives a user request, processes it with middleware, solves exceptions, then passes it to the page router and gives the response result. (Stauffer 2017.)

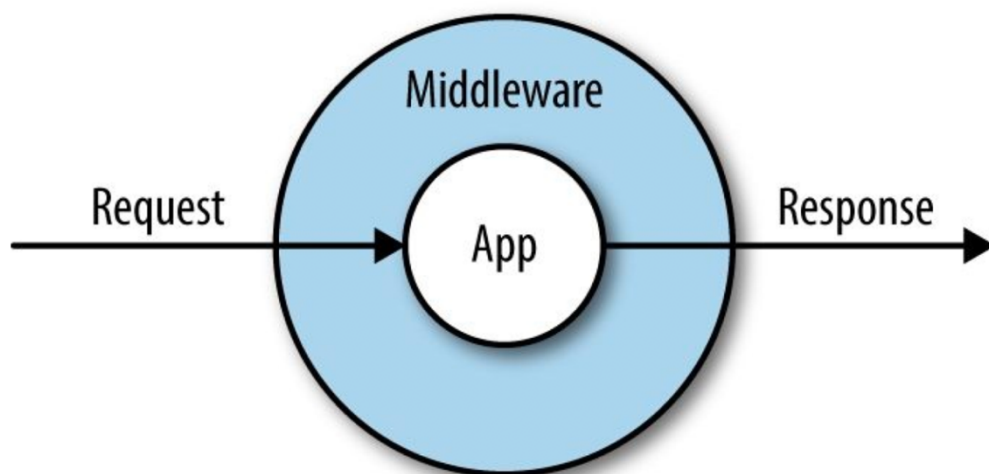


Figure 4. The request/response life cycle (Stauffer 2017).

Additionally, a kernel is divided into two types. They are the HTTP kernel working with web requests and the console kernel working with console, cron, and artisan requests. The Illuminate Request class of Laravel is a special extension from Symfony. The Request object has the main purpose which is to collect information about the current request. It uses methods such as the basic user input method through a form or an Ajax component explicitly, the user and request state methods with non-explicit input through a form too. In case the input is either explicit or defined by the browser, the file method can handle effectively. The Response object is the same as the Request object when it derived from Symfony. An Illuminate Response object replies to the end user with headers, cookies, content, some specialized

response types such as views, download, files, and JSON on the final response. In Figure 4, an important part of request/response lifecycle which cannot be ignored is middleware. The definition of middleware is that it is a list of layers wrapping around an application. It is a place where every request passes through it and the response is returned back again before going to the end user; therefore, middleware takes responsibility for decorating requests and responses. Middleware can receive or reject a request with some condition about its accessed time. When having too many requests, middleware sends back a 429 status after a threshold is passed. (Stauffer 2017.)

Another tool of Laravel that plays the key role at the core of all features by binding and resolving concrete classes and interfaces of an application in the container. The container is a powerful management tool for a network. It is usually called a service container or dependency injection container. The reason why dependency injection name appeared is that each class's dependencies are injected in from the outside instead of being instantiated within a class. It especially happens with construction injection which injects an object's dependencies when it is created, setter injection in which the class supplies a method to inject a given dependency and method injection where methods are called to predict their dependencies to be injected. (Stauffer 2017.)

For the reason of dealing with a database, Laravel supplies the Storage facade and some helper functions to handle a package of file manipulation tools. These tools can connect the Laravel application to the local filesystem, S3, Rackspace and FTP. Each connection is known as a disk. Laravel's file is managed in `config/filesystems.php` file. The local disk handles the local storage system through `storage/app`. The public disk - one kind of the local disk - is used for serving any file in `storage/app/public` directory. On the opposite side, the S3 disk works with cloud-based file storage systems. The special point of storage is the survival of the session. Session storage is known as a primary tool of a web application to manage the state between the page requests. It controls the session drivers by files, cookies, a database, Memcached or Redis, in-memory arrays. Session setting and drivers are configured in `config/session.php`. Developers can encrypt their session data as well as select a type of driver, and specify the length of session storage. There is a similar structure with a session that is cache configured at `config/cache.php`. The difference between the cache and the session is that the cache stores the data cached per application while the session stores the data cached per user. This is a strength of the cache compared to the session in order to store large database results, API calls, and slow queries. However, when solving the requests and responses, programmers prefer to connect to another tool than the session and the cache. This tool is the cookie which has the same points as the session and the cache such as structure, an available facade, a global helper, and mental models. Due to the fact that cookies are attached in the requests and

response, the requests contain them and provide them when the users visit the page; the responses send them out with the instructions for users to save them; especially, cookies can be queued through CookieJar queue. (Stauffer 2017.)

Other aspects of Laravel's features are mail and notification which are complex systems supplying APIs. SwiftMailer supports Laravel's mail functionality cooperating with drivers for Mailgun, Mandrill, Sparkpost, SES, SMTP, PHP Mail, and Sendmail. Nowadays, users' communication is developing to be more complex and quicker. Therefore, a new function known as notifications is added into Laravel 5.3. Notification is a PHP class that contains all the information of communications which the developers send to the users. These communications can be an email, SMS via Nexmo, WebSockets ping, a record to a database, or a message to a Slack. In order to send a notification, Laravel allows to use the Notification facade and insert the Notifiable trait to an Eloquent class. (Stauffer 2017.)

There are some architectural patterns and structures which are lesser common than above features in Laravel that are queues, queued jobs, events, WebSocket event publishing and scheduler. The working of a queue worker is arranging jobs which are added into an application out of the queue one at a time and showing the suitable behavior such as removing, returning or marking these jobs. Laravel applies Redis, beanstalkd, Amazon's SQS or database tables to solve all queues. Queues can handle a costly or a slow process easily, even heavy processes quickly compared to the ability of an application server. UserSubscribed, UserSignedUp, and ContactWasAdded are events which notify that something has taken place. In events which can be fired out of the application, some of them can be listened by several event listeners and then replied in response; however, some of them are never received. About WebSocket, it is a protocol. Instead of passing information via HTTP requests, WebSockets own a large library supplying communication between the client and the server directly. They are applied in Gmail and Facebook. Laravel supports developers maximally because it provides a scheduler tool. It is a convenient tool to manage whether tasks run or not. (Stauffer 2017.)

3.3 Laravel's frontend

Although Laravel is a backend framework, a Laravel project is equipped with enough frontend tools to build a complete website with interactive UI. These tools are JavaScript, Sass asset files and package.json file. Since Laravel 5.3, Vue.js has become a default frontend framework for Laravel. This means that some Vue.js files appear in Laravel's skeleton after the Laravel application is installed. Some features of Vue.js were mentioned in section 4. JavaScript assets are located in the resources/assets.js folder.

This folder contains some .js file and a sub-directory component of a .vue file. app.js is the main JavaScript file where an adjacent file is imported on the first line to load some libraries such as jQuery, and Lodash. Laravel use Sass, a CSS extension, to decorate its code in the resources/assets/sass directory. One of Sass files in Laravel is app.scss that imports other Sass files such as Bootstrap CSS framework. The most important file in the root of the Laravel project is the package.json file because it contributes to the configuration and the dependency management for the Node modules. The property specifies that jQuery, Vue and Lodash module were the devDependencies as shown in Figure 5. There is only a default view file in resources/views/ folder. That is welcome.blade.php file used as a home page of the Laravel project. However, developers can modify it or add other blade files in this folder. (Gore 2017.)

```

package.json:

{
  ...
  "devDependencies": {
    "axios": "^0.17",
    "bootstrap-sass": "^3.3.7",
    "cross-env": "^5.1",
    "jquery": "^3.2",
    "laravel-mix": "^1.4",
    "lodash": "^4.17.4",
    "vue": "^2.5.3"
  }
}

```

Figure 5. The devDependencies property (Gore 2017).

There are several dependencies of third-party JavaScript libraries and these dependencies need to be managed effectively in a browser. In some applications, a module system is used to handle dependency management problems such as CommonJS and native ES modules. However, CommonJS cannot act in a browser and native ES modules receive only browser support. That is the reason why the webpack was added in Laravel 5.5. Thanks to the webpack, a module system can be processed in a project. The webpack can analyze the entry file to find bootstrap, vue, Example.vue and other dependencies. This action is called bundling process. All results of its analyzation are bundled into a single file called bundle.js. During the bundling process, the webpack also transforms a module with some webpack loaders so that some verbs which are not allowed in older browsers can be understood in any browser. Besides webpack's advantage, it also has one drawback: its configuration is hard. Therefore, the Mix was created to reduce this disadvantage. Laravel Mix owns the most popular options of the webpack and hides them by a simple API. The Mix file is located in the root of the project directory. (Gore 2017.)

3.4 Comparing Laravel with other frameworks

Section 3.1 introduced some PHP frameworks. Except for Codeigniter's drawbacks shown in this section, other frameworks also have both advantages and disadvantages. Therefore, choosing the PHP framework depends on the developers' requirements. According to Google trends, searching times of these frameworks in the world are different. In Figure 6, the result leads to the conclusion that Laravel is the most popular in the PHP world with the highest searching time. Thanks to the features mentioned in section 3.2, the Laravel framework has become the best PHP framework. Beginners can learn and build applications quickly basing on good documentation of each release. Laravel inherited Symfony's advantages; however, it no longer takes as more time as Symfony. Although Symfony documentation is immense, it is hard for newcomers to learn and apply to a web project quickly. Except for this disadvantage, Symfony has been still popular for large projects relying on modularity. (Dudkin, 2019.)

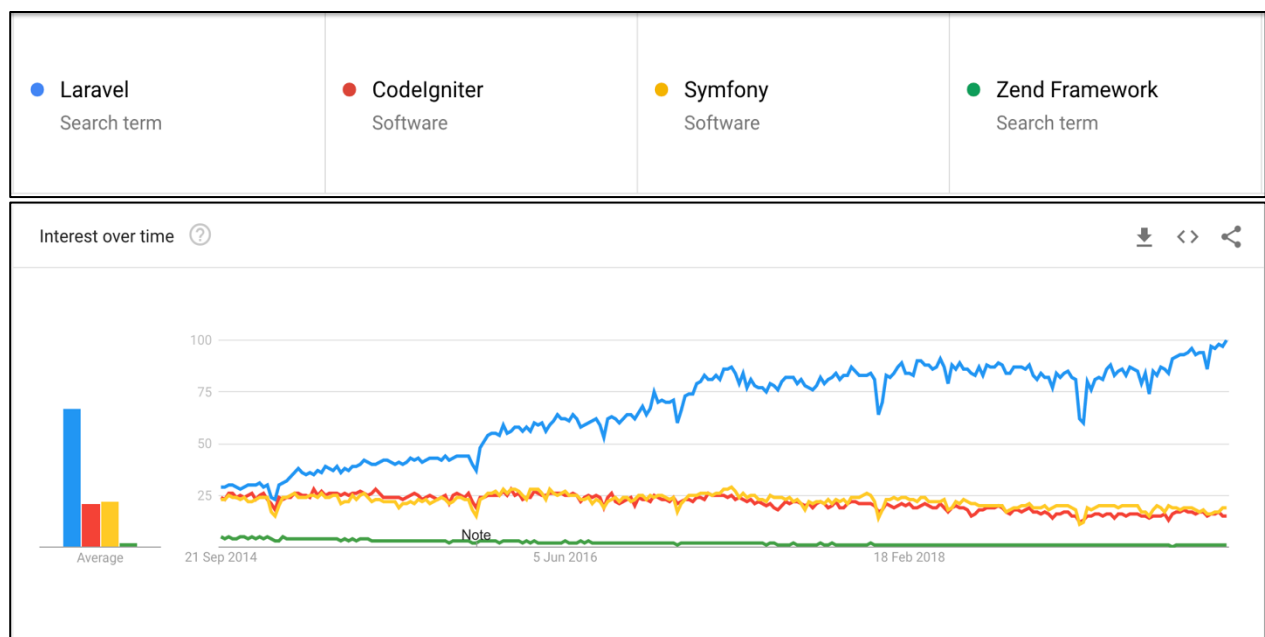


Figure 6. The searching trend of PHP Framework (Screenshot from Google Trend)

Besides Laravel, CakePHP is also one of the favored PHP frameworks of many developers because it permits them to write less code. CakePHP uses three different formats including PHP, XML, and YAML while Laravel has only PHP. This means that people utilizing Laravel need to understand only one format. On the other side, thanks to these formats, CakePHP's configuration is more flexible than Laravel. Kernel events are used in CakePHP for the same function as middleware of Laravel. They both process HTTP requests. CakePHP owns the ORM doctrine to handle the database and reduce the size of the code

while Laravel owns Eloquent. Eloquent is a part of ORM; therefore, working with Laravel is simpler than CakePHP. Blade is a template engine of Laravel; however, CakePHP uses Twig to process data. These templates are similar in structure. There is only a difference between them: Twig is stricter than Blade in handling data. In conclusion, Laravel is popular in small and less complex projects because CakePHP has greater reliability, flexibility to deal with larger, more complex projects. (Scott 2019.)

Another competitor of Laravel is Zend Framework. Zend is object oriented-based and open source. It is also free for students and developers. It is a combination of Model View Controller and Front Controller design pattern. Zend was born with the agile methodology of the brain. This architecture helps developers concentrate on only necessary components (Dudkin 2019.). Several tools are added in Zend such as debugging, drag and drop editor, and cryptographic coding. There are some differences between Laravel and Zend. Firstly, Laravel supplies both authentication and authorization in its application while Zend has only an authentication feature. Secondly, a developer can learn Laravel from beginner to master level quickly; however, it is a hard study process when developers choose Zend Framework. Next, Laravel is good at handling SQL queries while Zend does not support any tools to work with SQL queries. These frameworks are used for building web applications, and desktop applications. Although Zend has poor community support compared to Laravel, it has high the performance than Laravel. Therefore, Zend is suitable for complicated projects. (educba 2019.)

3.5 Setting the Laravel project

Laravel can be set up in Windows, Unix, Linux or macOS. Laravel project has some requirements to serve Laravel sites irrespective of processing the development environment from a virtual machine via Vagrant or depending on MAMP/WAMP/XAMPP. These requirements are for the PHP version which needs to be larger than 5.6.4 for Laravel 5.3 or larger than 5.5.9 for Laravel 5.1 and 5.2, OpenSSL PHP extension, PDO PHP extension, Mbstring PHP extension, and Tokenizer PHP extension (Stauffer 2017.). Laravel is downloaded and updated through Composer, a third-party package. Composer is an effective dependency manager, the same as Bundler from the Ruby community, npm from node.js and other JavaScript frameworks. Composer is installed on a per-project basic without packages globally; therefore, it needs PEAR to add PHP dependencies into the system. With Composer, Laravel project is downloaded quickly without searching, downloading and unzipping files. Besides, it is easy to upgrade the dependencies without feeling worried about version conflicts. The Composer also has a central repository to find and select packages effectively. Developers can access <http://getcomposer.org> to download the newest version of the Composer. After downloading, the Composer commands are added into

the list of commands which are executed in the command line from any location of a filesystem. There is a `composer.json` file that is read and collaborated with Packagist from <https://packagist.org> in order to download some dependencies to a local directory and their state to `composer.lock`. The first step of installing the Composer is checking the PHP version on the operating system following the command shown in Figure 7. (Saunier 2014.)

```
$ php -v
```

Figure 7. The command to check PHP version (Saunier 2014).

The PHP version affects on the installation of Composer. In case this command is not found, or PHP version is smaller than the requirement for Laravel version, developers using MacOS can install PHP again through MAMP or Homebrew. The next step is installing Composer through different commands for each operating system. Figure 8 is the command in macOS. The command with `sudo` is to move `composer.phar` to a directory that is contained in the application's `PATH`. The last command in Figure 8 is to check if the Composer is installed successfully or not. (Saunier 2014.)

```
$ curl -sS https://getcomposer.org/installer | php
$ sudo mv composer.phar /usr/local/bin/composer
$ composer
```

Figure 8. The command to install Composer on MacOS (Saunier 2014).

In Windows, Composer is downloaded directly from <http://getcomposer.org/download/> with `Composer-Setup.exe` file. The location of the Composer file needs to be in `C:\PHP5\` or `C:\PHP\` as a default; or `C:\xampp\php\` in case of XAMPP; or `C:\wamp\bin\php\php5.x.x\` in case of WAMP. Programmers check the PHP version and the list of available command of composer following the commands in Figure 9. (Saunier 2014.)

```
> php -v
> composer
```

Figure 9. The command to check PHP version and Composer on Windows (Saunier 2014).

When the Composer exists in the operating system, programmers can create a Laravel application. The command, which is displayed in Figure 10, brings the newest Laravel version and its dependencies to the developers' folder. It takes a few minutes to complete this installation depending on their CPU capability and connection speed. The root of the directory has a .gitignore file and a .gitkeep file for each placeholder directory; hence, developers can run git init for the project. (Saunier 2014.)

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

Figure 10. The command to create a Laravel project (Saunier 2014).

After a Laravel project is created, its main folders consist of some contents. The first folder is the app that includes models, controllers, route definitions, commands, and PHP domain code. Secondly, the bootstrap is used to boots the Laravel framework, and the config takes responsibility in the configuration. Next, the database has database migrations and seeds while the public is pointed through the server. Additionally, the resource is a place where non-PHP files survive such as Views, language files, Sass/LESS, and source JavaScript files. The contents of the routes are HTTP routes and console routes. The storage folder owns caches, logs, and compiled system files; however, the tests one organizes unit and integration tests live. The last folder is the vendor saving the Composer's dependencies. A Laravel application's skeleton contains several files and directories as shown in Figure 11. (Stauffer 2017.)

```
app/  
bootstrap/  
config/  
database/  
public/  
resources/  
routes/  
storage/  
tests/  
vendor/  
.env  
.env.example  
.gitattributes  
.gitignore  
artisan  
composer.json  
composer.lock  
gulpfile.js  
package.json  
phpunit.xml  
readme.md  
server.php
```

Figure 11. The sekeleton of a Laravel application (Stauffer 2017).

Moreover, some loose files are added in the root directory. They are `.env` and `.env example`, `artisan`, `.gitignore` and `.gitattributes`, `composer.json` and `composer.lock`, `gulpfile.js`, `package.json`, `phpunit.xml`, `readme.md`, and `server.php`. One group of them takes responsibility in regulating the environment variables. That is a group of `.env` and `.env example` in which `.env` is duplication of `.env example` and is modified to be suitable for each project. `Artisan` is an indispensable file in running Artisan commands from the command line. Laravel project has a backup server through `server.php` so that less-capable servers can see the Laravel application. (Stauffer 2017.)

4 ADDING VUEJS INTO LARAVEL PROJECT

Nowadays, websites are becoming more complex and more interactive with effective user interfaces. Therefore, frontend development has to be improved day by day to adapt to users' demands. Because of various benefits of the JavaScript frameworks, frontend has been added complicated functionality and powerful tools in order to solve complex state, execute client-side routing, and control a complicated code. Vue.js is the latest JavaScript framework. It is a popular tool in building interactive websites with several useful functions (Macrae 2018.). All features of Vue.js were mentioned in section 4.2 after the history of Vue.js was described in section 4.1. Section 4.3 displayed some strength of Vue.js compared to other JavaScript frameworks. Then, section 4.4 showed the steps to practice with Vue.js.

4.1 Introduction and installation

The creator of Vue.js is Evan You, a Chinese developer. He had an interesting idea about Vue.js when he worked in a project at Google Creative Labs. This project required him to build a complex UI (user interface) with a large size of repeated HTML. At that time, there were some frameworks such as Angular, React.js, and Backbone.js. However, none of them contained enough tools and libraries to adapt to this project. He needed a flexible and lightweight framework. That was the reason why Vue.js was born and released the first version in 2014. The purpose of Vue.js is that it supports rapid prototyping through data binding and components. It is easy for beginners to study it and create a complete project from simple to complex project by accessing its documentation. (Filipova 2016.)

The way to install Vue into a Laravel project is not complex. Programmers need to add an external script in the body of HTML document and install all the development dependencies by npm install command that is a third-party script. After downloading successfully, the webpack is added into this application. The role of webpack was discussed in section 3.3. npm allows running some commands such as running dev script, and watch script and to download available packages such as vue-router, vuex, and font packages. Figure 12 contains some examples of npm commands. (Gore 2017.)

```
$ npm install
$ npm run dev
$ npm i --save-dev vuex vue-router axios
$ npm i --save-dev font-awesome open-sans-all
```

Figure 12. The npm commands (Gore 2017).

Today, Vue can run on the recent versions of any browser. This project's browser is Google Chrome because its features are suitable for Vue Devtools. However, Firefox is also a good choice. Vue Devtools supports Vue in debugging effectively. All operating systems are permitted to download and build a Vue application. (Gore 2017.)

4.2 Vue.js overview

When creating an instance of Vue.js, some basic options play key roles in a Vue.js application. One of them is a template. The template is known as the heart of Vue. In fact, a template is a normal HTML attached with special attributes or directives. The template is used for generating the DOM which replaces the element that was chosen with the `el` option on the web page. The `el` option is a property specifying the place of Vue instance on the web page. A template allows using CSS selectors for HTML element when it has an ID. CSS selector is began by a hashtag (#) or put in `<script></script>`. In some cases, components create a template by rendering instead of using normal HTML or a string template because the render function completes this job more sequentially (Nelson 2018.). Figure 13 is a basic example of a template in Vue.js.

```
<!-- Script Element for our App -->
<script>
  var app = new Vue({
    el: '#app',
    data: {
      propertyName: 'Hello from Getting to Know Vue.js!'
    }
  });
</script>
```

Figure 13. The example for template of Vue (Nelson 2018).

In Figure 13, data property is used to store the database. It can store strings, arrays, and objects in an object. Developers need to add new data before creating a new instance of Vue because all the properties of data are added to a reactive system at once and this system controls all their changes. Therefore, in case some data cannot be defined the value, they are assigned as null or undefined values. The empty object `{}` is not accepted. There is a rule in dubbing the name of the data's properties. To avoid conflict with Vue's internal properties and method, `$` or `_` is not a suitable character to start the name. Figure 14

is an example when the name of properties of the data causes errors and Vue cannot find this data. (Nelson 2018.)

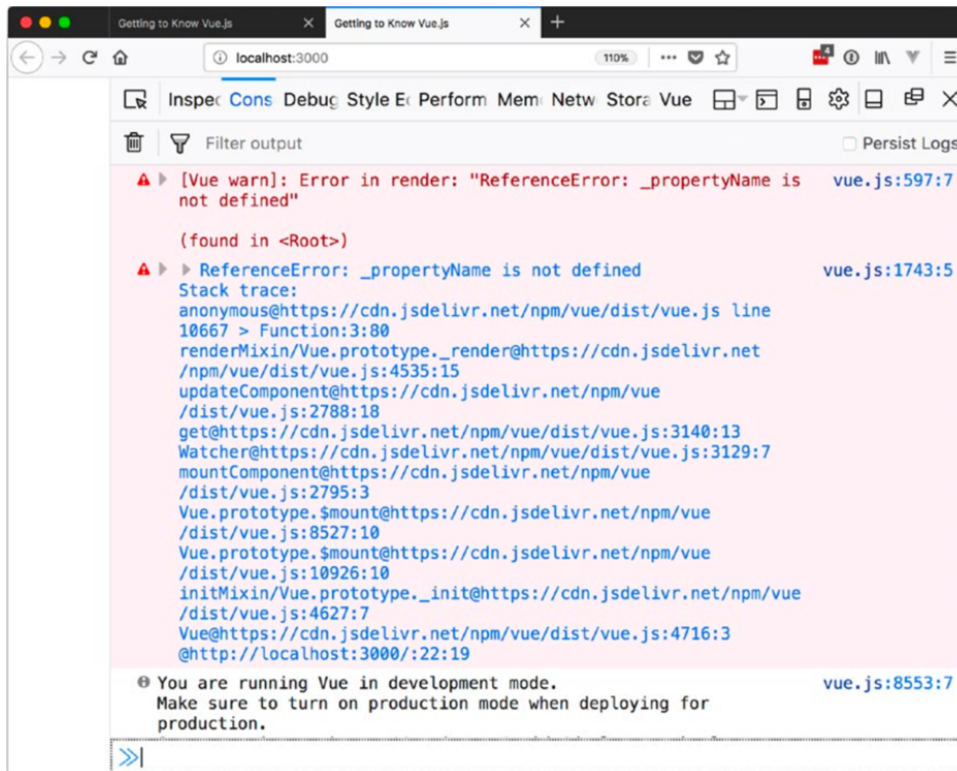


Figure 14. The error caused by the name of properties of the data (Nelson 2018).

Vue is one kind of JavaScript; therefore, it also uses methods in order to execute actions between users and an application. Methods are accessed from a reference to the Vue instance. The arrow function `() => {}` is able to prevent Vue from accessing the proper context of methods; therefore, it needs to be limited using in them. Beside of the methods, Vue supplies computed properties. The way to use computed properties is similar to methods; however, their result is cached. The values are updated in only a situation. That is when the values based on the computed properties change (Nelson 2018.). An exciting property in Vue is a watcher. Watchers have the responsibility in watching a property of the data object or a computed property for changes. Watchers perform asynchronous operations well. When a property needs to be watched, it is simply to use the watcher by specifying the name of this property as shown in Figure 15. (Macrae 2018.)

```

<script>
  new Vue({
    el: '#app',
    data: {
      count: 0
    },
    watchers: {
      count() {
        // this.count has been changed!
      }
    }
  });
</script>

```

Figure 15. An example for Watcher (Macrae 2018).

In Vue, there is the best way of manipulating data in the template. This property is called filter, a popular choice in other template languages. It contributes to reducing the duplication and simplifying the code of the application. An expression can include multiple filters by connecting them. Filter functions can pass given data as an argument. V-bind is often used to bind values to arguments. In addition, `Vue.filter()` is preferred to use for registering filters on over the application. However, the filter has both pros and cons. Due to the reason that the functions of filters handle input and return output without concerning external data, filters are placed at the position where the developers are not allowed to add methods and data. Therefore, programmers only pass data of the filters in an argument in order to get access to them. The other drawback of a filter is that filter is utilized in interpolations and v-bind directives. Although programmers could put filter anywhere the expressions appear in Vue 1, a filter has to be put in a method or computed property in Vue2. (Macrae 2018.)

Events in Vue have a responsibility in displaying data interactively. They are the listener which waits for events, event handlers which solve events and modifiers which change events (Nelson 2018). The v-on directive helps an event listener bind to an element. It uses the name of this event and the event listener as the argument and the passed value respectively (Macrae 2018). A handler method is usually called when programs need to achieve a cumbersome amount of variable assignments in an expression. Programmers change the behavior of an event through event modifiers declaratively. That means declaring it in the markup instead of assigning the modification in the JavaScript. Programmers can see this modification when the event handlers are registered in the markup. Some popular event modifiers of Vue are stop, prevent, capture, self, once, and passive. In order to stop the propagation of the current

event, `event.stopPropagation()` is called for stop event. Prevent event does not allow the user agent to handle the event with `event.preventDefault()` while capture permits to call the handler before the target of the event accesses it by capture mode. Besides, self has a special task in calling the handling in only a condition if the event starts on the element. That leads to saving more time on checking the `event.target` in order to handle right events that begin on the registered element. On the other hand, once the event means calling handler once without removing the handler from the element. The last event is passive that alters event handler option of passive to true. Therefore, the handler is allowed to call `event.preventDefault()` and is ignored by the browser. Of course, these modifiers can be chained easily. In case of input, the modifiers are shown by some keys such as Enter, Tab, Delete, Esc, Space, Up, Down, Left, Right. (Nelson 2018.)

The main feature in Vue.js being help developers separate their application into small portions instead of organizing them in one JavaScript file is component. Component contributes to maintaining and re-using their code effectively. Component is seen as a miniature copy of a Vue instance with enough characters of a Vue instance except for an `el` property. All components in a Vue application cooperate together in order to deal with data and perform UI. `Vue.component` is the way to register the component. `Vue.component` contains two parameters. One of them is a name of the component and another is a JavaScript object that includes all contents for this component. The name can be the same as kebab-case or PascalCase. Figure 16 is an example of a component. (Nelson 2018.)

```
Vue.component('OurHeader', {
  template: `
    <h1>App Header</h1>
  `
});
```

Figure 16. The example for a Vue component (Nelson 2018).

In components, data is performed as a function that returns an object; hence, each instance of the component owns its data individually without being affected by the changes of other instances of the component. In addition, a component can communicate with the parent component to share data through props. This means that it can get access and work with data of the parent component. Prop's function is also the specification of types and default values when assigning an object replacing an array. The only five accepted types are String, Number, Boolean, Array, and Object. An error can appear when the type of a value is not in this group. Figure 17 shows how to use a prop of a component. In addition to types, props can be used with other options such as default value, required, and the validator. (Nelson 2018.)

```

Vue.component('OurFourthHeader', {
  props: {
    text: String
  },
  template: `
    <h1>{{text}}</h1>
  `
});

```

Figure 17. An example for a prop (Nelson 2018).

In the opposite direction, a parent component can get access to data of the child component; however, it uses events instead of props. With events, the parent component listens to the change of event on the element's declaration and shows it through `v-on:event-name`. However, the child component has to send the event through `$emit` method attaching two values. They are the name of an event and a value to pass. In order to wrap other content, a component needs a `<slot></slot>`. (Nelson 2018.)

Due to the fact that all application own data from small to large, simple to complicate, developers need to be supported to manage data effectively. Vue provides some solutions consisting of a simple data object, a do-it-yourself data store, and a state management library in order to adapt to this demand. A simple data object is the basic and simple way to manage data and share access to the same values among Vue instances. However, it is not enough flexible in case of complicate projects and replaced by DIY (do-it-yourself) data store method. Excepts for its function which is as same as the function of the simple data object, methods are added to DIY to change date indirectly. This leads to managing data easily when programmers can follow the changes in data. DIY data store has more advantages than the simple data object; however, it has less useful features than Vuex in the managing state. Vuex is a library that supplies state management by several new treats and features. Vuex is known as a suitable application state for all Vue components. Programmers have to install Vuex by npm or CDN (content delivery network) before applying it in their code. Figure 18 is the way to import Vuex to the application. (Nelson 2018.)

```

import Vue from 'vue';
import Vuex from 'vuex';

Vue.use(Vuex);

```

Figure 18. Importing Vuex (Nelson 2018).

Vuex has some important properties which are discussed in this paragraph. The first property is called state that includes all the shared data. The functions of this state is similar to the functions of the state of DIY data store. The second one is getter which incorporates all results of data in the store. A getter is also seen as a computed property that allows reusing in several Vue components; therefore, its results are often cached. The getter function can merge the results of other getters in order to handle more complex results in small portions. The next way to update the value of the store is using a mutation which owns a string type and a handler function. The handler function requires at least one parameter called state, and another parameter known as a payload is added to this function. In the case of passing multiple values to the mutation, an object is utilized. Synchronizing all mutations when logging them to follow changes is compulsory. However, when using a callback or promise or other asynchronous task, the mutation is disabled and replaced by action. After an action completes these tasks, commit can be called, and the results are saved in the transaction history of Vuex. Actions and mutations have a common format such as a string type, name, a handler function except a dispatch called at the store when committing an action. In fact, the larger data is becoming in the technology world, the more complex data management is. Therefore, in recent releases, another option is added to Vuex. That is a module where a store is declared. Each module contains its state, getters, mutations, and actions. (Nelson 2018.)

There are various ways to design style for a Vue application such as using v-bind:class, v-bind:style, and vue-loader. Especially, v-bind:class and v-bind:style are special class binding to set a class attribute and to decorate it to be more beautiful. v-bind:style is also known as an inline style binding. Through vue-loader, programmers can write scoped CSS with <style></style> tag in code as shown in Figure 17. The difference between Vue and JavaScript is that the CSS of a component influences on all the HTML. Hence, developers need to write the <style scoped> tag instead of only <style> tag so that the CSS designs for only that component. In Figure 19, lang="scss" is added in order to apply SCSS as an alternative to CSS for that component. However, that application needs to be installed the sass-loader and node-scss by npm. (Macrae 2018.)

```
<style lang="scss" scoped>
  $color: red;

  .number {
    font-weight: bold;
    color: $color;
  }
</style>
```

Figure 19. An example for a scoped CSS in Vue (Macrae 2018).

Vue has a special library that handles the routing of an application. Thanks to this library, all websites can be navigated from any other page on client-side instead of working with the server. This is called vue-router downloaded by npm or added by CDN. It is not hard to create the router because it requires only an array of paths and the related components such as the example in Figure 20. The content of a component is displayed following the path. After setting the router, it needs to be passed in to Vue and <router-view/> component needs to be added in the template. The position of <router-view/> is where the router is displayed. (Macrae 2018.)

```
import PageHome from './components/pages/Home';
import PageAbout from './components/pages/About';

const router = new VueRouter({
  routes: [
    {
      path: '/',
      component: PageHome
    },
    {
      path: '/about',
      component: PageAbout
    }
  ]
});
```

Figure 20. An example for scoped CSS in Vue (Macrae 2018).

Axios is a familiar tool in Vue.js because it is used for displaying AJAX requests to the web service. In order to use axios, the project needs to add the import line after installing it through npm package. Axios allows to create requests to a GET, PUT, POST, DELETE URL as a simple API. Axios has a promise-based library to recover the response with then callback. In the callback, the data is returned for the component instance. (Gore 2017.)

4.3 Vuetify

Thanks to several useful features of Vue.js, it is a preferred choice to build many websites and applications. However, Vue.js concentrates on the application logic more than on its appearance. Many creators created various frontend UI frameworks for Vue.js to design UI professionally. Vuetify is one of these frameworks. It was built by John Leider who has worked in Vue community since 2014 and is supported by a vibrant Discord community forum. Vuetify is confident about working on all browsers including old browsers such as IE11 and Safari 9 with the babel-polyfill requirement. Vuetify provides several

free or premium price and pre-made layouts on its website. Therefore, developers can access them easily and apply them to their projects quickly. The installation of Vuetify is similar to vue-router or vue through npm. Besides, developers need to modify JavaScript file such as Figure 21 to import it. (Wanyoike 2019.)

```
import Vue from "vue";
import Vuetify from "vuetify";
import "vuetify/dist/vuetify.min.css";

Vue.use(Vuetify);
```

Figure 21. The code of Vuetify is added into JavaScript file (Wanyoike, 2019)

Vuetify has various components which are used simply to decorate web pages. The v-app is one of its components replaced inside the <body> tag and usually is a parent of other Vuetify components. It defines grid breakpoints for the layout. The lower level of the v-app is the v-content component. Next, the v-toolbar component shows the primary toolbar of the application containing icons, menus, and search bar. Some properties are added inside the tag of components to customize their CSS. For example, the dark property is used to change the background of a component to black and its text to white. Another property is color which allows the background of the component to alter depending on the developer's idea. Furthermore, other components which are applied regularly in diverse projects are v-container, v-layout, and v-flex. (Wanyoike 2019.)

5 THE PROJECT

When the author experienced the internship at a company in Vietnam, the author was assigned in a Frontend project and the tasks were using Vue.js to design the basic UI for this project. The researcher had an opportunity to learn and apply a new framework in a real project. Then, the researcher realized that Vue.js has several great features, therefore, it was brought into this project. However, the author would like to promote the project from a frontend to a full-stack website. Laravel which has an excellent combination between a PHP framework and a JavaScript framework was researched. This section discussed the project in deep.

5.1 Idea

The aim of this project is building a website for The Coffee House. The main task is that focusing on authentication of login form as well as some permissions of admin and user role. Admin, logged in by the owner of this company, can manage all users by reading, editing and deleting users who are known as its small branches. Each role has different UIs to work. There is a website so that customers can access and order foods and drinks on it. All orders are sent to branches and they can see all of them on their UI. In addition, a forum page and a message page are built to create a friendly environment for all staff in the coffee shop because they can communicate and share all information through these pages privately and collectively. The project is built by Laravel framework and Vue.js.

5.2 Planning

When having the idea, the author wrote UI of websites in the paper to determine what the author needed to create step by step. This paper helped controlling the process of this project which were done, and which were not completed. Firstly, the author built a welcome page and a login form. Then some other pages were created to perform different UI for admin and other users. Thanks to Laravel's features, the project was supported well. Before beginning to write code, the author prepared enough tools such as XAMPP, Visual Studio Code and Squel Pro. These tools are technical requirements for building websites, testing websites, managing database.

5.3 Obstacles

When beginning this project, the author was a newcomer with new knowledge and skill in PHP framework in general and Laravel in particular. Therefore, there are more challenges during the process of this

thesis. The amount of new knowledge which the author had to learn and improve is much. This project is based on various references from professional developers. The researcher read Laravel documentation, Vue.js documentation as well as some Laravel projects on GitHub. It takes around 2 weeks to collect this information. Then the researcher spent time on reading, selecting and memorizing the necessary knowledge to apply them on the project.

5.4 Implementation and Construction

The first step of this project was setting XAMPP, downloading Composer, MySQL or MariaDB. After downloading XAMPP, the project needed to be created at `/Applications/XAMPP/htdocs/` folder. At this folder, the project was started by the command in Figure 22. The project's name is `my-project`. Especially, `npm` and `PHP` was installed to allow `npm` and `artisan` command to work, respectively. The last command in Figure 22 was to check the version of `PHP`.

```
composer global require "laravel/installer"  
laravel new my-project  
npm install  
php artisan --version
```

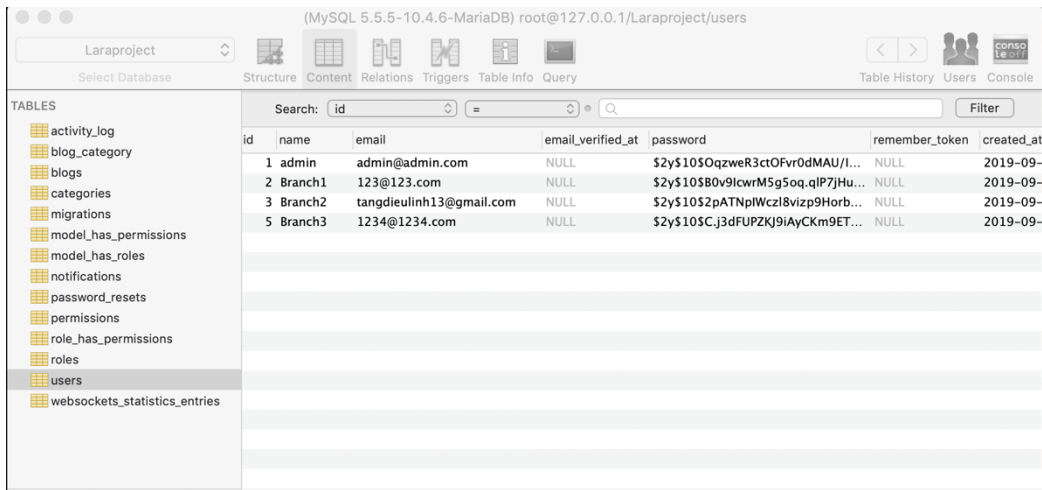
Figure 22 The command to create Laravel project

The second step was setting the database. `MySQL` was installed by `homebrew`. Instead of accessing the database on `phpMyAdmin`, the author used `Sequel Pro` that was a software designed for database management on `macOS`. It was suitable for the author's laptop when the database was created and edited fast and easily. `Sequel Pro` is free for downloading. In order to connect the project to the database, a `.env` file was modified at some lines as shown in Figure 23. In this figure, the host was `127.0.0.1`, the database name was `Laraproject` at port `3307`.

```
DB_CONNECTION=mysql  
DB_HOST=127.0.0.1  
DB_PORT=3307  
DB_DATABASE=Laraproject  
DB_USERNAME=root  
DB_PASSWORD=
```

Figure 23 Modifying `.env` file (Screenshot from Visual Studio Code)

A crucial migration added into the project is `create_users_table` —create=users via php artisan. The database had a table containing `id`, `name`, `email`, `email_verified_at`, and `password`. All migrations were created in `/app/database/migrations/` folder. Then, “php artisan migrate” command was ran. The command was successful; hence, the table was created and displayed on Sequel Pro as shown in Figure 24. Installing `vue-router`, `vuetify`, and `axios` were via “`npm install vue-router vuetify - -save`” command and adding their import into the project were the required step.



id	name	email	email_verified_at	password	remember_token	created_at
1	admin	admin@admin.com	NULL	\$2y\$10\$QzweR3ctOFvr0dMAU/1...	NULL	2019-09-
2	Branch1	123@123.com	NULL	\$2y\$10\$80v9lcvrM5g5oq.qIP7jHu...	NULL	2019-09-
3	Branch2	tangdieulinh13@gmail.com	NULL	\$2y\$10\$2pATNplWczl8vizzp9Horb...	NULL	2019-09-
5	Branch3	1234@1234.com	NULL	\$2y\$10\$Cj3dFUPZKJ9lAyCKm9ET...	NULL	2019-09-

Figure 24. The picture of Sequel Pro

Laravel provides a good authentication system. The author applied it after downloading it via php artisan. Some files appeared in the project as shown in Figure 25 in order to build login form, register form, login button, register button and logout button. Some lines in `welcome.blade.php` were edited to show the login button, the register button, and the logout button in different cases. If users logged in, the logout button appears.

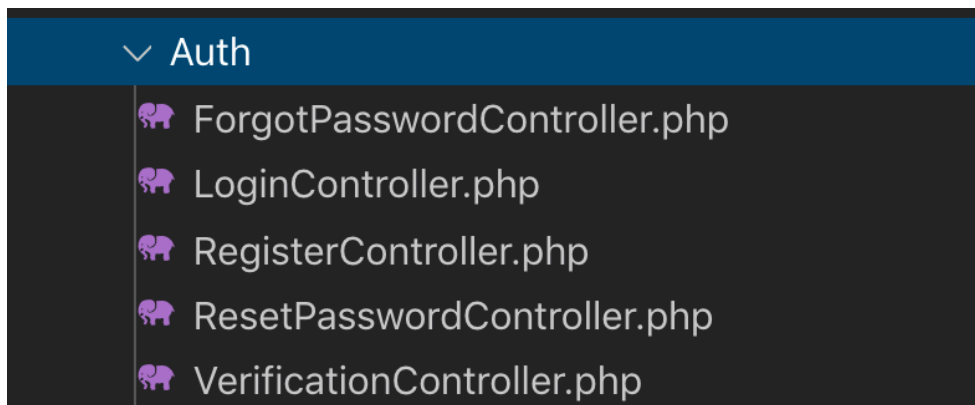


Figure 25. The login form (Screenshot from Visual Studio Code)

Then, `route/` folder was modified. In this folder, `web.php` contains all important routes, each route directed to different pages. Figure 26 was the content of `web.php`. Some routes direct to `/admin/` and some

routes direct to /company/ depending on the role of the users. Authentication is shown through ['auth', 'company'] and ['auth', 'admin'] by using middleware. There were some controllers in Figure 25. Therefore, the next step was to create controllers. The project's controllers consist of HomeController, companyController, adminController, and orderController in app/Http/Controllers/ folder. Each controller was changed to have suitable content.

```

Route::get('/', function () {
    return view('welcome');
});

Auth::routes();

Route::get('/home', 'HomeController@index')->name('home');

Route::group(['company', 'middleware' => ['auth', 'company']], function() {
    Route::get('/company/{name}', 'companyController@index');
    Route::get('/company', 'companyController@index');
    Route::get('/company/{any?}', 'companyController@index')->where('any', '.*');
});

Route::group(['company', 'middleware' => ['auth', 'company']], function() {
    Route::get('/order/{name}', 'orderController@index');
});

Route::group(['admin', 'middleware' => ['auth', 'admin']], function() {
    Route::get('/admin', 'adminController@index');
    Route::get('/admin/{name}', 'adminController@index');
    Route::get('/admin/{any?}', 'adminController@index')->where('any', '.*');
});

Route::any('/admin/{any?}', 'adminController@index')->where('any', '.*')->middleware('auth');
Route::any('/company/{any?}', 'companyController@index')->where('any', '.*')->middleware('auth');

```

Figure 26. The content of web.php (Screenshot from Visual Studio Code)

With these controllers, several blade templates were created in /resources/views/ folder. The added files are company.blade.php, admin.blade.php, order.blade.php while home.blade.php and welcome.blade.php are default files which were included in the project after its installation. Besides web.php, /route folder also contains api.php and amin.php. api.php allows the application to access data through api while amin.php contains Route::apiResource of users and roles so that admin add, edit, delete, and user list easily. Similarly to web.api, AdminApi/RoleController.php and AdminApi/UserController.php were added to the project.

The project has a resource/js/components folder where some Vue.js file exists. The author also set up another folder called pages. Components folder includes _barofcom.vue, _sidebar.vue, and _navbar.vue to display the sidebar of company role, the sidebar of the admin role, the navigation bar of both roles, respectively. Admin.vue and Company.vue used these files as separated components. All files in pages folder were to show the content of the website. The user interface of the admin role is a combination of _sidebar.vue, _navbar.vue, Admin.vue, and Users.vue. All Vue.js files were applied Vuetify framework

in the code. Some of the tags of Vuetify the author used in the project are v-app, v-list, v-container, v-icon, ...

Another file the researcher focused on is app.js because this file was used to import important libraries. A routes constant was assigned an array of paths as shown in Figure 27. These paths direct to related components when it is called. Before that, all components had to be imported in this file so that the process can find them. If there is any necessary library or any problem in the location of the components, error alerts and the application is built unsuccessfully. app.js is found in /resources/js/ folder.

```
const routes = [
  {
    path: "/admin/welcome",
    component: Welcome
  },
  {
    path: "/admin/roles",
    component: Roles
  },
  {
    path: "/admin/users",
    component: Users
  },
  {
    path: "/admin/menu",
    component: Menu
  },
  {
    path: "/company/dashboard",
    component: Dashboard
  }
];
```

Figure 27. The contents of app.js (Screenshot from Visual Studio Code)

Besides, the author also created a page called Messages page where admin and other users can contact closely. Thanks to this page, all staff of the coffee shop can communicate without using other application messages. A ChatMessage.php file was added in /app/events/ folder. This file contains some necessary functions such as broadcastOn() that return the messages of the current user. An important library supporting to build this page is a pusher. Pusher was installed by Composer. The website of pusher supplies different app_key, and app_id for each project. After updating the .env file with the author's app_key, app_id, ChatController with sendMessage() function were built to send messages from the current user

to a receiver. Next, chat.blade.php was set in /resource/views to design UI of the chat room. In order to route to chat page, web.php was appended with two lines as shown in Figure 28.

```
Route::post('/chat','ChatController@sendMessage');  
Route::get('/chat','ChatController@chatPage');
```

Figure 28. The web.php was modified (Screenshot from Visual Studio Code).

Another page the author created is a forum page so that all users can attend an online chat meeting. This forum allows users to post a new discussion, to see other discussions and to comment. A valuable library that supplied enough tools to build this forum is Chatter. Chatter was installed through Composer similarly to other libraries. After installing completely, DevDojo\Chatter\ChatterServiceProvider::class, a service provider, needed to be added into the app.php. A list of tables was created after running “php artisan migrate”. Figure 29 is a group of new Controller files in this project. The User interface of this forum page was modified through discussion.blade.php and home.blade.php which are located in /vendor/devdojo/chatter/src/views/ folder.

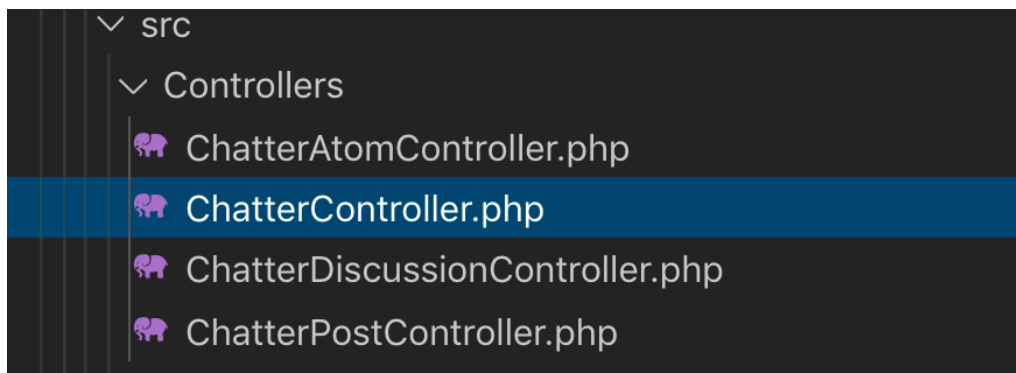


Figure 29. The list of Chat Controller files (Screenshot from Visual Studio Code).

When completing all code in my project, “npm run watch” was ran to build Laravel Mix. Then “php artisan serve” was turned on, the project was live on my localhost, <http://127.0.0.1:8000> was accessed to test the user interface of the project. On Google Chrome, the author checked all errors in the console panel, network panel.

5.5 The result

When going to `http://127.0.0.1:8000`, the first page that appeared was the home page which contents were decided by `welcome.blade.php` in `/resources/views`. The User interface of the home page was displayed as same as Figure 30. Home page contained the name of the coffee shop and login button, register button. These contents replaced the primitive content of Laravel's home page. It was decorated by scoped CSS added in its component.

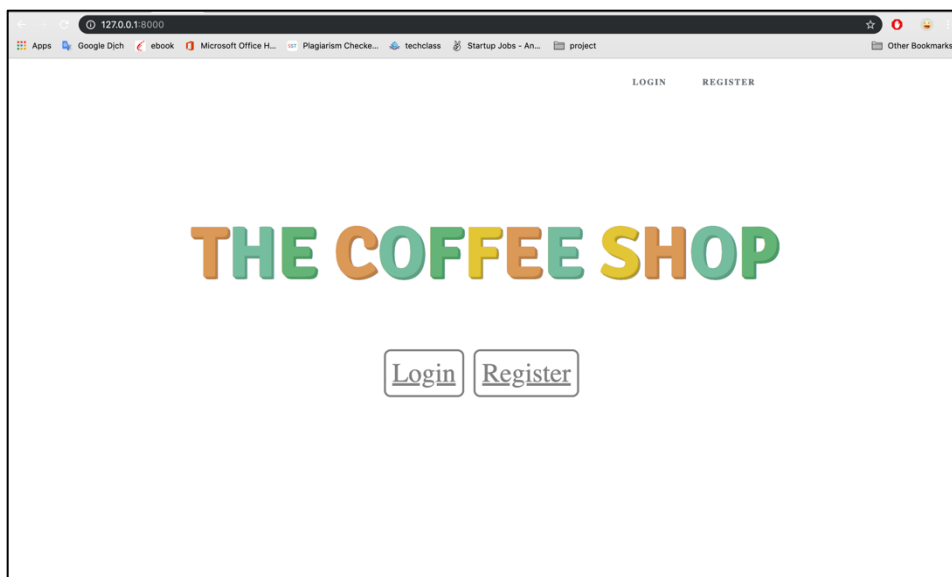
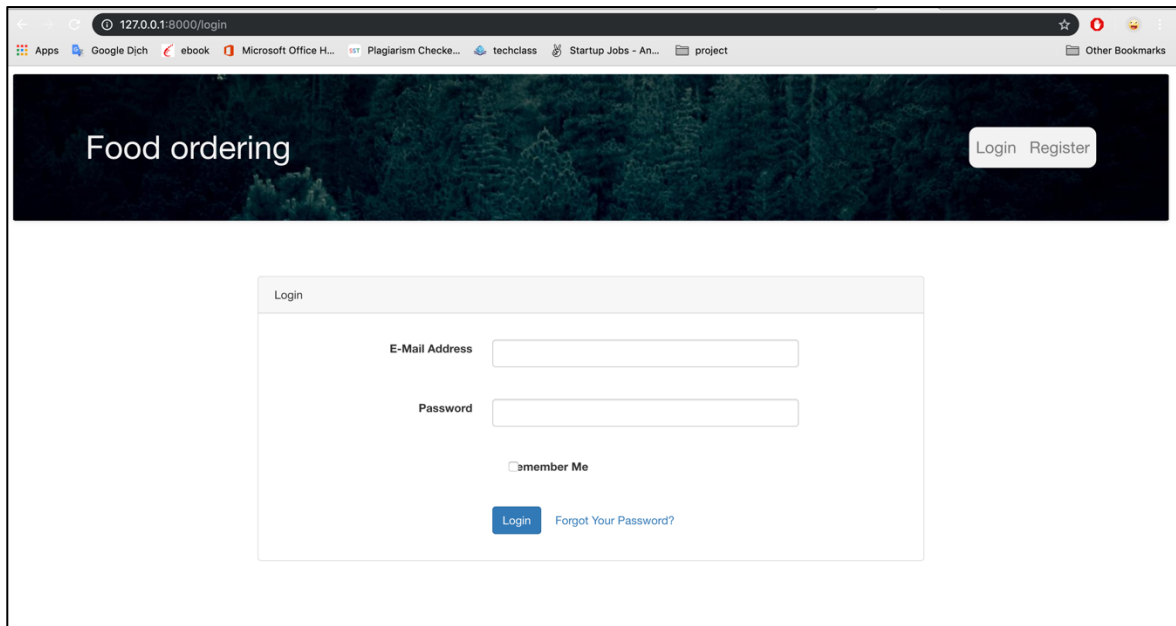


Figure 30. The result of the home page

Login button directs to login form with `http://127.0.0.1:8000/login` route. Login form includes e-mail address input, and password input. Unless they are filled the right characters, the error notification can be shown. Moreover, “remember me” checkbox was added to memorize the email address and password of users. Figure 31 is the login form. When clicking the register button, a registration form can be displayed instead of the login form. All information which is entered into the registration form is saved in the database table of MySQL.



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/login'. The browser's tab bar includes 'Apps', 'Google Digh', 'ebook', 'Microsoft Office H...', 'Plagiarism Checke...', 'techclass', 'Startup Jobs - An...', and 'project'. The page features a dark green header with the text 'Food ordering' and a 'Login Register' button. Below the header is a white login form titled 'Login'. The form contains two input fields: 'E-Mail Address' and 'Password'. Below these fields is a checkbox labeled 'Remember Me'. At the bottom of the form are two buttons: 'Login' (in blue) and 'Forgot Your Password?' (in grey).

Figure 31. The picture of login form.

After entering the email address and password of admin, the homepage's contents are changed following Figure 32. The name of the user is shown on the screen. Login and register are replaced by log out button in the menu of the navigation bar. The route name was also altered with <http://127.0.0.1:8000/home>.

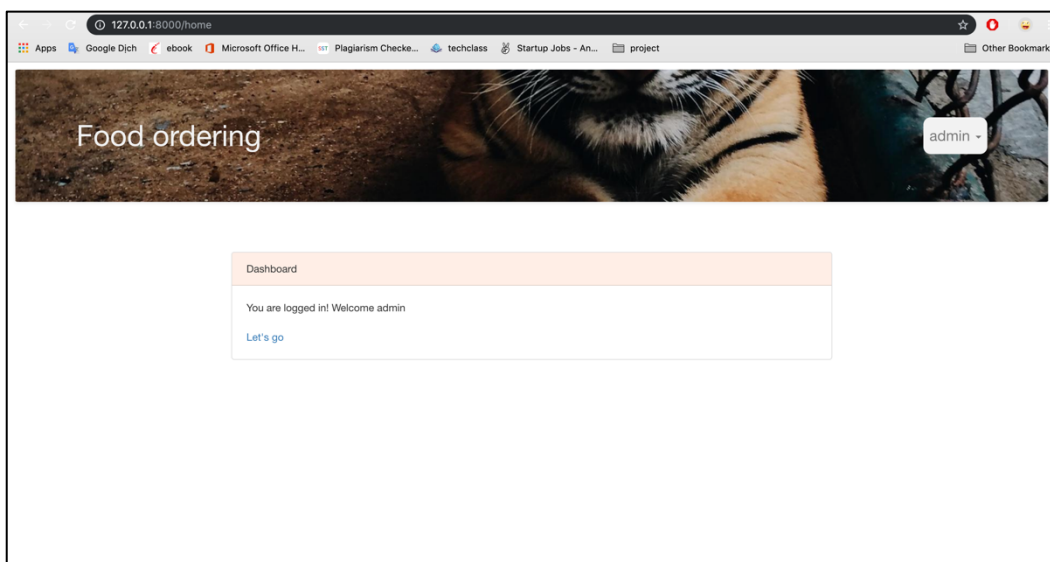


Figure 32. The result of admin page.

In Figure 32, there is a link directing to the admin page in case of the admin role. UI of admin page is similar to Figure 33. The sidebar has three options: Menu, User management, Role management. At user management choice, the list of users is displayed at the center of this page. Admin has permission on editing, deleting, and adding new items.

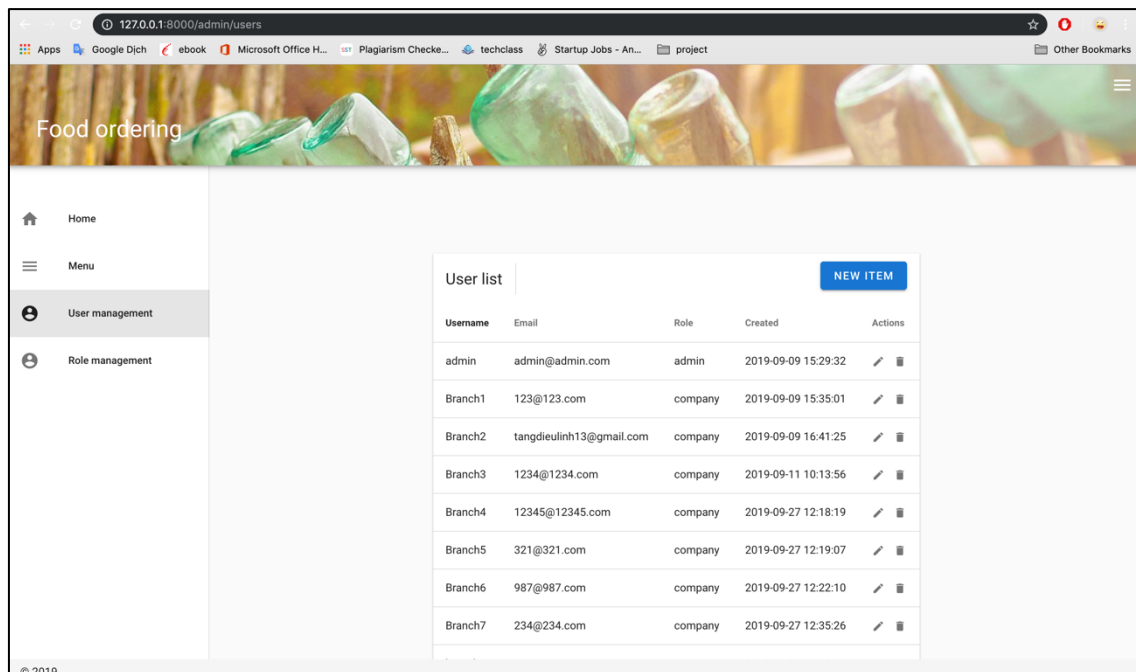


Figure 33. The result of user management page.

However, the UI of other users is different from admin. Other users do not have the permission to update and change the list of users. They only observe all orders which were sent from customers. According to the table of order list, each branch can follow and analyze their sales easily and quickly. The result of the branch page was shown in Figure 34.

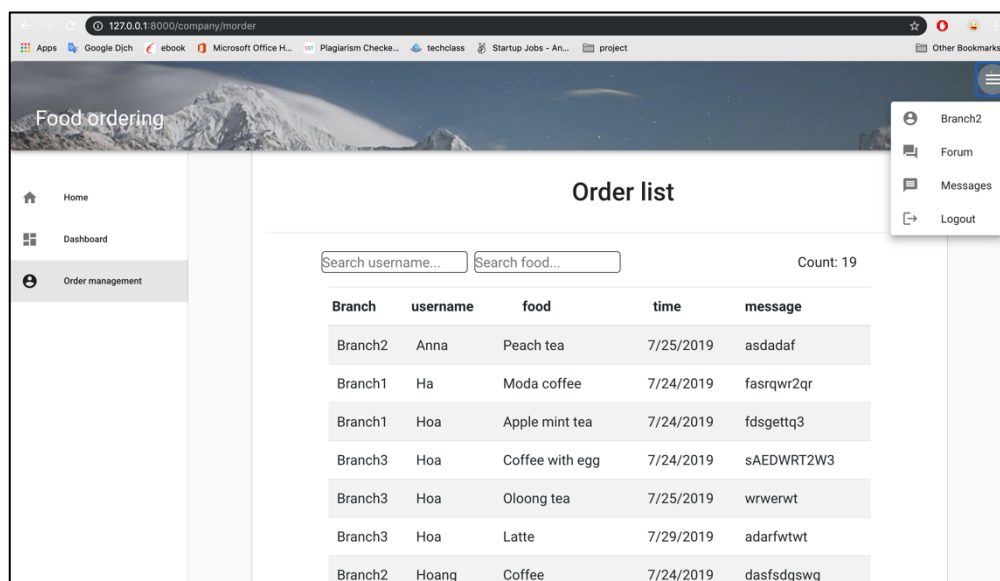


Figure 34. The result of branch page

Figure 34 is the result after some customers ordered food. A webpage was created for customers which are shown in Figure 35. At this page, the customers chose foods or drinks and filled their information sent to each branch. The route of this page is 127.0.0.1:8000/order/Branch2. The route name depends on the name of the Branch. Each branch manages different orders.

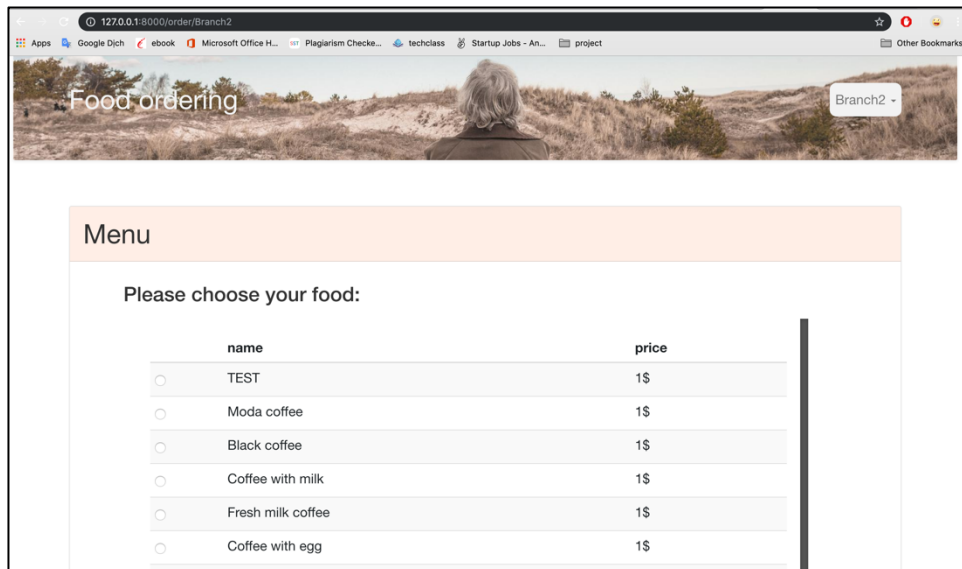


Figure 35. The order form of customers.

Furthermore, in Figure 34, there is a menu in the navigation. This menu has four elements directing to different pages including the home page, the forum page, the message page, and logout function. The forum page and the message page were decorated through dissimilar styles compared to other pages as shown in Figure 36 and 37, respectively. At the forum page, users are allowed to write new discussion and comment to other member's discussion.

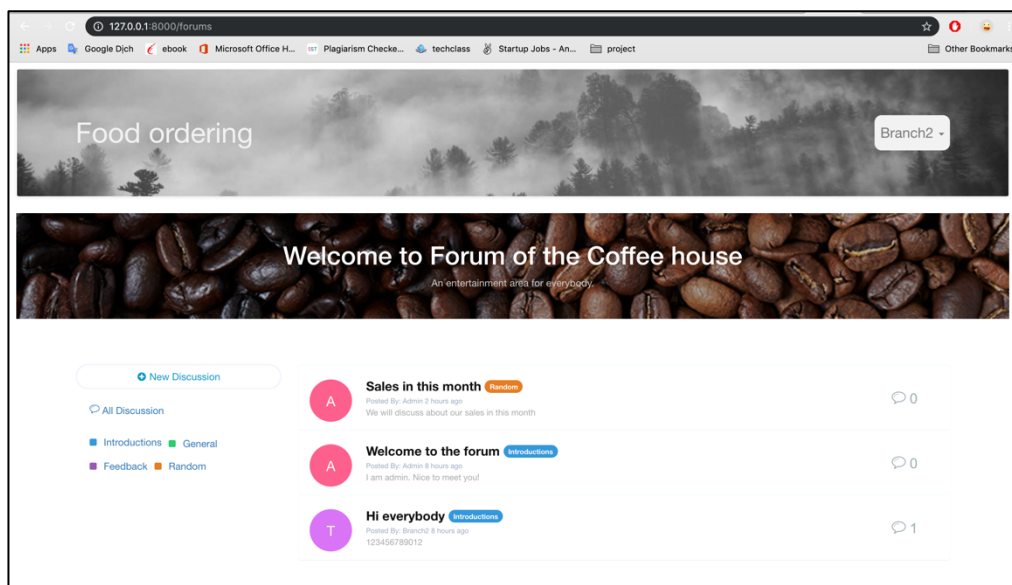
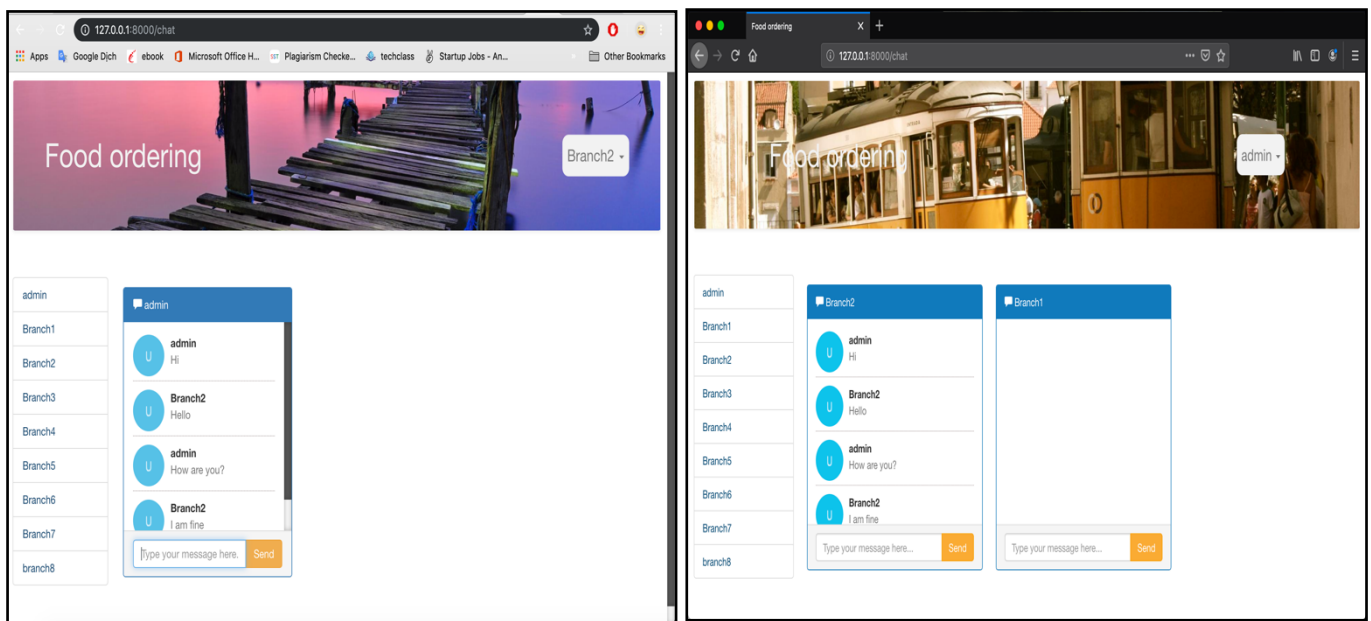


Figure 36. The forum page.

Figure 37 is the conversation between the admin and Branch2. Figure 37a is the UI of Branch2 running on Google Chrome while Figure 37b is the UI of admin running on Firefox. The messages page also allows opening many chat windows at the same time. The list of users is shown on the left side of the screen to choose the person who needs to be contacted. <http://127.0.0.1:8000/chat> is the route of the messages page.



a

b

Figure 37 a. The messages page of Branch2. b. The messages page of admin

In addition, when a branch logs in, he cannot access to admin page and vice versa. If he tries to access all the pages of admin, the result always returns to the home page automatically. This is one of the useful functions provided by the Laravel framework. Logout is the last button in the menu of the navigation bar. After logging out, the webpage returns to the welcome page as shown in Figure 30.

6 CONCLUSION

The primary goal of the project was to build the website for a coffee shop based on Laravel and Vue.js. The main tasks were creating different UI for admin and other users. Laravel and Vue.js are suitable for designing a full-stack website in general and this project in particular. Authentication of Laravel supported this project to build a login function and a register function. Besides, its database was managed by Sequel Pro effectively. The technical requirements such as XAMPP, MySQL, Laravel, and Vue.js are explained in this thesis. Vuetify is used in its code as a decorating tool to make the website nicer.

The website can bring some advantages to the coffee shop when its admin can control users and users can follow all orders from customers. This coffee shop is only in the author's imagination. However, it is a good example of a Laravel project and can apply for other management systems of many companies that need an effective management system when the database major is developing quickly.

The coffee shop has a website containing a menu and allows customers to select one kind of food and send information to branch admins. However, it lacks a shopping cart to make payment. For further development, the web developer can construct a payment system connected to the bank and ensure high security. This order page can be decorated with several images of food and drinks and add more information about foods and drinks such as their ingredients, and the ways how to cook so that this page can attract more customers. About the admin page, programmers can add more tags except for user management tag and role management tag, for example, permission tag which can be used in determining what kinds of action users are allowed or not allowed when they have a branch role or another role. Adding tag depends on the demand of the web developers.

REFERENCES

- Badenas, S. T. 2018. Step by Step tutorial to add Vuetify to Laravel with Auth scaffolding and Vuex. Available at: <https://medium.com/@sergiturbadenas/step-by-step-tutorial-to-add-vuetify-to-laravel-with-scaffolding-auth-and-vuex-ac3f5a875bc5>. Accessed 11th September 2019.
- Barnes, E. L. 2012. Building a User Management System in Laravel. Available at: <https://gist.github.com/anchetaWern/4223764>. Accessed 11th September 2019.
- Bean, M. 2015. Laravel 5 Essentials. United Kingdom: Packt Publishing.
- Dudkin, I. 2019. PHP FRAMEWORKS: BEST FRAMEWORK COMPARISON LIST. Available at: <https://skywell.software/blog/php-frameworks-comparison-list/>. Accessed 20th September 2019.
- Dvorski, D. D. 2007. INSTALLING, CONFIGURING, AND DEVELOPING WITH XAMPP. Ontario: Skills Canada.
- educba, 2019. Laravel vs Zend. Available at: <https://www.educba.com/laravel-vs-zend/>. Accessed 21st September 2019.
- Filipova, O. 2016. Learning Vue.js 2. Birmingham, UK: Packt Publishing.
- Filipova, O. 2016. Learning Vue.js 2. Mumbai: Packt Publishing.
- Gil, P. 2019. Internet vs. Web: What's the Difference?. Available at: <https://www.lifewire.com/difference-between-the-internet-and-the-web-2483335>. Accessed 14 September 2019.
- Gore, A. 2017. Full-Stack Vue.js 2 and Laravel 5. 1st ed. Mumbai: Packt Publishing.
- Harris, A. & McCulloh, C. 2008. Moving from Web pages to Web sites. In: HTML, XHTML, and CSS All-in-One Desk Reference for Dummies. Indiana: John Wiley & Sons, Inc., pp. 803-811.

Julie, C. M. 2012. Sams Teach Yourself PHP, MySQL® and Apache All in One. USA: Pearson Education, Inc..

Macrae, C. 2018. Vue.js: Up and Running - Building Accessible and Performant Web Apps. s.l.:O'Reilly Media.

Malatesta, F. 2015. Learning Laravel's Eloquent. Birmingham: Packt Publishing.

Mehta, P. 2016. Creating Google Chrome Extensions. USA: Apress Berkely.

Murphy, L. 2008. Building Websites with ExpressionEngine 1.6. 1st ed. s.l.:Packt Publishing, Limited.

Nelson, B. 2018. Getting to Know Vue.js. Minnesota, USA: s.n.

Rauschmayer, A. 2014. Speaking JavaScript: An In-Depth Guide for Programmers. s.l.:O'Reilly Media.

Ripoche, B. 2018. Create a SPA with role-based authentication with Laravel and Vue.js. Available at: <https://medium.com/@ripoche.b/create-a-spa-with-role-based-authentication-with-laravel-and-vue-js-ac4b260b882f>. Accessed 11th September 2019.

Saunier, R. 2014. Getting Started with Laravel 4. s.l.:Packt Publishing, Limited.

Scott, M. 2019. CAKEPHP VS LARAVEL: WHICH ONE TO CHOOSE FOR WEBSITE FRAMEWORK. Available at: <https://www.ingeniumweb.com/blog/post/cakephp-vs-laravel-which-one-to-choose-for-website-framework/4198/>. Accessed 20th September 2019.

Stauffer, M. 2017. Laravel: Up and Running. Sebastopol, CA 95472: O'Reilly Media.

Team, K. B. 2019. Laravel 5 Users and Roles Management. s.l.:s.n.

Wanyoike, M. 2019. How to Get Started with Vuetify. Available at: <https://www.sitepoint.com/get-started-vuetify/> Accessed 23th September 2019.

Yadav, S. C. 2009. Introduction to Client Server Computing. Varanasi: New Age International Ltd.

