OAMK
OULU UNIVERSITY OF
APPLIED SCIENCES

Richard Awale

# DESIGNING AND DEVELOPING A DYNAMIC WEBSITE USING LARAVEL

# DESIGNING AND DEVELOPING A WEBSITE USING LARAVEL

Richard Awale
Bachelor’s Thesis
Autumn 2018
Information technology
Oulu University of Applied Sciences

# ABSTRACT

## OULU UNIVERSITY OF APPLIED SCIENCES

Degree Programme in Information Technology. Option of Internet services.

---

Author: Richard Awale
Title of the bachelor's thesis: Designing and Developing a Website using Laravel

Term and year of completion: 2018 Number of pages: 44
Supervisor: Janne Kumpuoja

---

The main purpose of this thesis was to design and develop a user-friendly, responsive website with an admin panel for easy maintenance of the website. The project was given to me by Mr. Arindam Mallick, owner of the Indian restaurant "Indian Cuisine".

The urge to learn a new technology for me and with the consultation of the client Laravel framework was chosen along with other web development tools such as HTML, CSS, JS, and Bootstrap. The MAMP localhost was the localhost server used for the development of this project. Laravel was then set up to the MAMP localhost in a local computer. A free to use bootstrap template was chosen after discussing with the client. The necessary update was made to the template to get the desire front-end although it could be improved. The database and tables were created according to the need and were migrated to the server.

The website has been created and is handed over to the client. Although all the project objectives were completed, there is some room for improvement which is currently on the process, Thus the website is not online.

---

Keywords: Laravel, MVC, MAMP localhost, PHP, PHP framework

# PREFACE

This project was ordered by “INDIAN CUISINE”, an Indian restaurant located in Kajaaninkatu 38,90100. The client representative is the owner of the restaurant Mr. Arindam Mallick. The work was done for the Oulu University of Applied Science. The supervisor of the thesis was Mr. Janne Kumpuoja. The role of the thesis supervisor is to guide the author of the thesis about the thesis document and developing the webpage itself.

I would like to thank Mr. Arindam Mallick for trusting me and believing in me to develop a website for his restaurant. I would also like to thank my supervisor for helping me in all the thesis related work and guiding me to complete my thesis. I would also like to thank my project partner Mr. Bishal Shah without whom I could not have completed some part of my thesis especially things related to the stripe payment and some Laravel back-end.

Oulu, 26.11.2018

Richard Awale

## CONTENTS

ABSTRACT 3
PREFACE 4
TABLE OF CONTENTS 5
VOCABULARY 7
1 INTRODUCTION 8
2 WEB DEVELOPMENT TOOLS AND TECHNOLOGIES USED 10
2.1 Front-End 10
2.1.1 HTML 10
2.1.2 CSS 10
2.1.3 JavaScript (JS) 11
2.1.4 Bootstrap 12
2.2 Backend 13
2.2.1 MAMP localhost 13
2.2.2 PHP 15
2.2.3 PHP Framework 16
2.2.4 PHP(framework) security and threats 17
2.2.5 Stripe payment 18
3 LARAVEL 20
3.1 Brief History of Laravel 20
3.2 Installations 21
3.2.1 Composer 22
3.2.2 Server Requirement 22
3.3 Configurations 23
3.4 Routes 24
3.5 Controllers 25
3.6 View 26
4 IMPLEMENTATION (CREATING WEBSITE FOR AN INDIAN RESTAURANT INDIANCUISINE) 27
4.1 Front-end implementation 28
4.1.1 Usage of Bootstrap in Project 28

4.1.2 Blade files. 30
4.2 Backend implementation 34
4.2.1 Controllers 34
4.2.2 Model 38
4.2.3 Database 39
5 CONCLUSION 42

## VOCABULARY

Bootstrap - Bootstrap is a free and open-source front-end library for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation, and other interface components, as well as optional JavaScript extensions.

CSS – Cascading Style Sheets

HTML – Hyper Text Markup Language

JS – JavaScript

PHP – Hypertext Pre-processor

# 1 INTRODUCTION

I have studied Information Technology at the Oulu University of Applied Science for over 4 years and I am hoping to implement the things I have learned during that period and learn a new PHP framework “Laravel” in process. I have always been fond of Front-end development of a website but to work on both Front and Back-end of this project will be a challenge and source for my personal growth. In this project, a website for “IndianCuisne” has been designed and developed.

## 1.1 IndianCuisine

“IndianCuisine” is an average size Indian eatery which serves quality Indian food. The restaurant is in Kajaaninkatu 38, 90100, Oulu. During the weekdays from 10.30 am till 3 pm lunch is served at a special price. There is also a buffet during Lunch time where you can get salads, rice, Butter naan different vegetable, and meat curries and hot & cold beverages for just 10.40 €. Apart from lunchtime, there is also a dinner with the selection of food ranging from vegan, vegetarian and non-vegetarian Indian curries. They also serve varieties of Naan bread, starters, and desserts.

## 1.2 Motivation

For any start-up company or even for an established and reputed company such as “IndianCuisine” a well-informed and easy to use website is the must for future development of the company. Nowadays there are websites for everything from small stores to big supermarkets and hotels. But having a website is not enough currently, having a competitively attractive and easy to use website is a must for the better success of a company. The previous website has been outdated and in need of change. So, to be updated with the current trend the project was offered.

### 1.3 Objective

The major aim of this thesis was to develop a dynamic and responsive website for the client. The major task could be divided into several individual small parts as:

I) Selecting a Front-end template according to the client
II) Adding and Modifying the layout
III) Ensuring the website is responsive
IV) Creating necessary database tables
V) Following the MVC model
VI) Creating an admin panel for an easy use. (CRUD operation)
VII) Securing Authentication

# 2 WEB DEVELOPMENT TOOLS AND TECHNOLOGIES USED

Web development is the process of creating either a static or dynamic webpage or web-application with the different technologies for the internet(WWW) or intranet (private network). The web development process can be broadly divided into two parts. The first part, Front-End consists of a User Interface(UI) of the web page i.e. how the page will look for the user. The second part is Back-end. It consists of all the databases and table and basically concerns all the server-side programming. The technologies used in the project are briefly described below:

## 2.1 Front-End

Front-end is a part of website development where the view or design of a website is created using Front-end Development tools. The tools used in this project are explained below.

### 2.1.1 HTML

HTML stands for a Hyper Text Markup Language. It is a Markup language for creating web pages and websites. The elements of HTML are the building block of a website and are represented by tags. Tags do not render themselves on a webpage, but they help to render other content. Example of HTML tags is “body”, “head”, “title”, “table”. The tags are enclosed in <>.

Syntax: < body> </body>,  .

Most of the tags have closing tags and they are represented as above.

### 2.1.2 CSS

CSS stands for a Cascading Style Sheet. It is used to add a style to HTML elements. It can be written in an HTML file or separately as a “.CSS” extension.

Syntax:

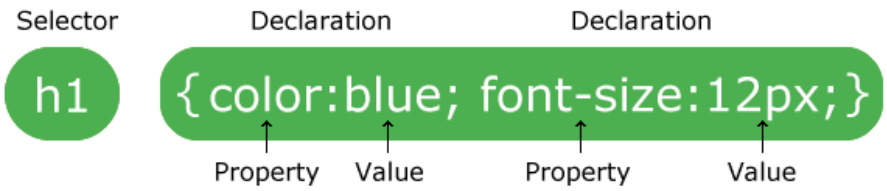


*FIGURE 1. Standard CSS syntax* (W3Schools CSS, Date of retrieval 19th May 2018)

A standard CSS syntax consists of a selector and a declaration block as shown in figure 1. The selector points to the HTML element which one wishes to style. The property and value are separated by a colon (:). Each declaration block is surrounded by curly braces and each declaration ends with a semicolon (;).

### 2.1.3 JavaScript (JS)

JS is a high level, interpreted programming language. It is a scripting language, primarily used on the Web. Being an interpreted language, it does not need to be compiled. It renders web pages in an interactive and dynamic fashion. It is one of the three core technologies of the World Wide Web (WWW) along with HTML and CSS. (W3schools, Date of retrieval 19th May 2018).

It is case sensitive and generally, the name of the function and variable is written with a lower case. Every statement is ended with a semicolon (;). For example, a variable is defined as: “var a = 10;”. Here ‘var’ is a datatype of the variable ‘a’, ‘=’ is an operator which assigns a value ‘10’ to the variable ‘a’. <script> </script> is used in an HTML file to insert JS script into HTML. If separate JS file is created its source must be pointed using an src attribute of <script>. Commonly JS is used for image manipulation, form validation and dynamic changes of content. To select an HTML element, JS often uses a **document. getElementById ()** method.

```
<!DOCTYPE html>
<html>
<body>


<p id="demo"></p>

<script>
document.getElementById("demo").inner
HTML = "Hello JavaScript!";
</script>

</body>
</html>
```

Hello JavaScript!

*FIGURE 2. An example of the use of JS in HTML* (W3schools JS, Date of retrieval 19th May 2018)

This JavaScript example in figure 2 writes "Hello JavaScript!" into an HTML element with id="demo".

### 2.1.4 Bootstrap

Bootstrap is a free and open-source front-end library for designing websites and web applications. It contains HTML and CSS-based design templates for typography, forms, buttons, navigation, and other interface components, as well as optional JavaScript extensions. Unlike many web frameworks, it concerns itself with front-end development only. (W3schools Bootstrap, Date of retrieval 20th May 2018).

Bootstrap can be downloaded from “getbootstrap.com” and then the link can be added to the source folder in the project as shown below:

<link href =” bootstrap.css (path)” type=”text/css” rel = “stylesheet”>

or It can be directly introduced to code as CDN (Content Delivery Network) without downloading as shown below:

<link rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

## 2.2 Backend

In the client-server model, the client is considered as a front-end and the server is considered as a backend, even some presentation work is done on the server. The technology used for the backend part of the project is explained below.

### 2.2.1 MAMP localhost

MAMP is an acronym of macOS (also on Windows), the operating system; Apache, the web server; MySQL, the database management system; and PHP, Perl, or Python, all programming languages used for web development. MAMP installs a local server environment in the firm's computer. MAMP was chosen as it was one of the most popular and easy to operate localhost server and it has PHPMyAdmin. (MAMP, Date of retrieval 16th May 2018).

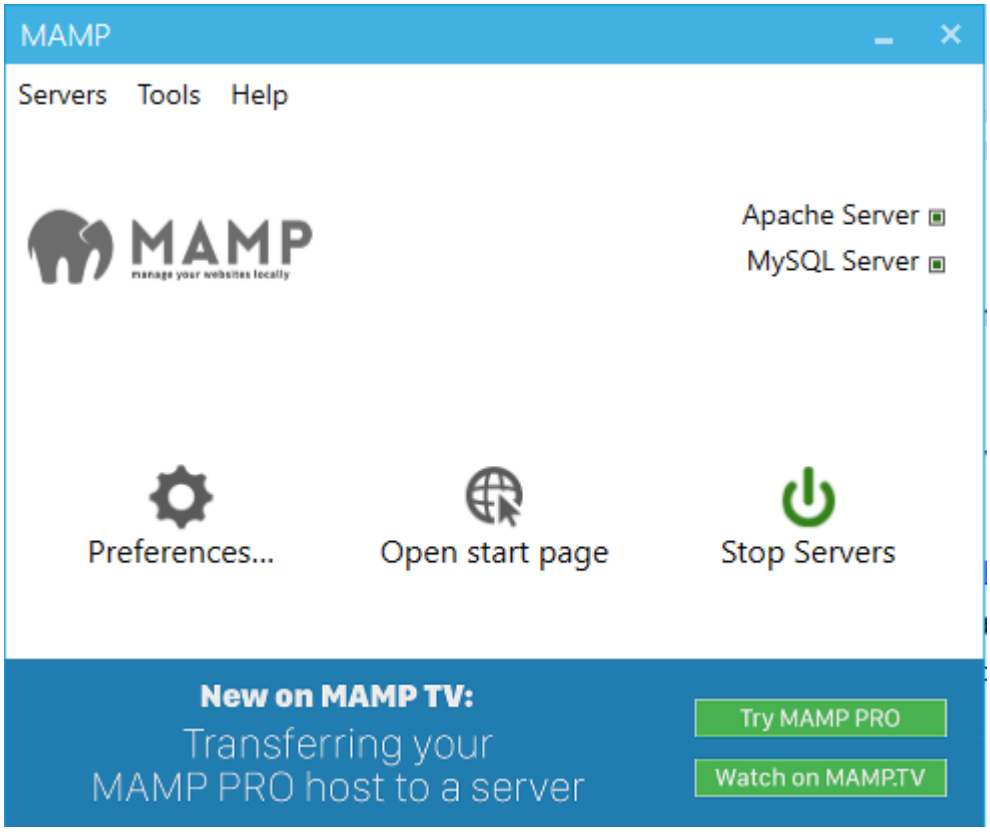


*FIGURE 3. The appearance of MAMP localhost window*

So after the localhost was set up, the default MAMP homepage could be run by clicking Open start page as shown in figure 3 and figure 4 shows its web homepage.

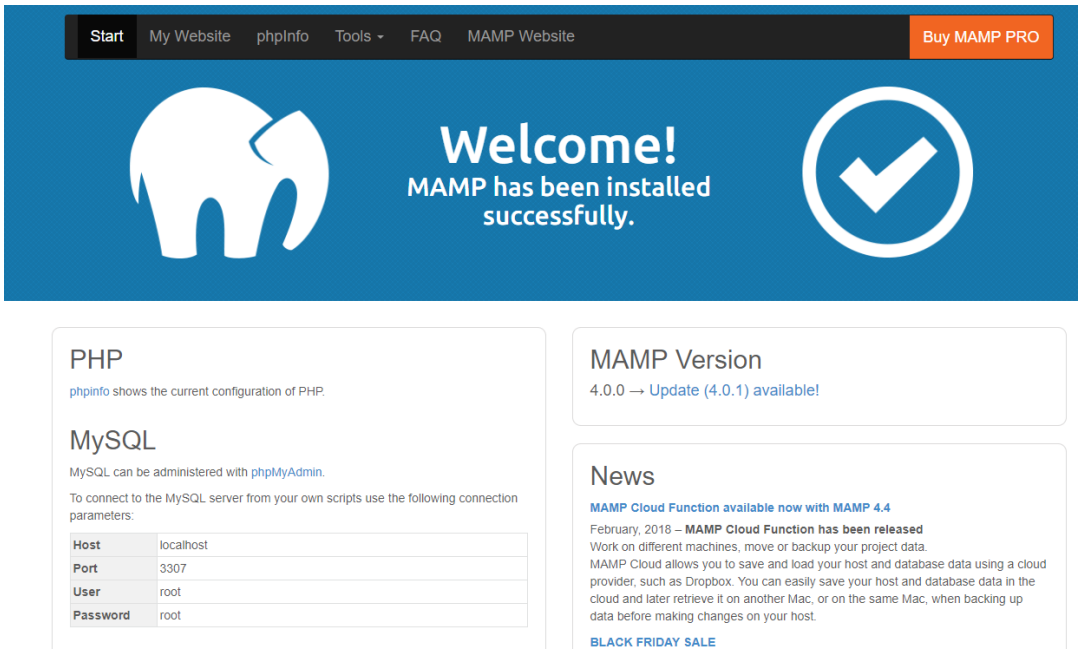


*FIGURE 4. The homepage of MAMP localhost*

phpMyAdmin is a graphical interface for the management of a database. Because of graphical interface, many MySQL queries could be operated directly into server easily. Figure 5 shows the layout of the phpMyAdmin homepage.

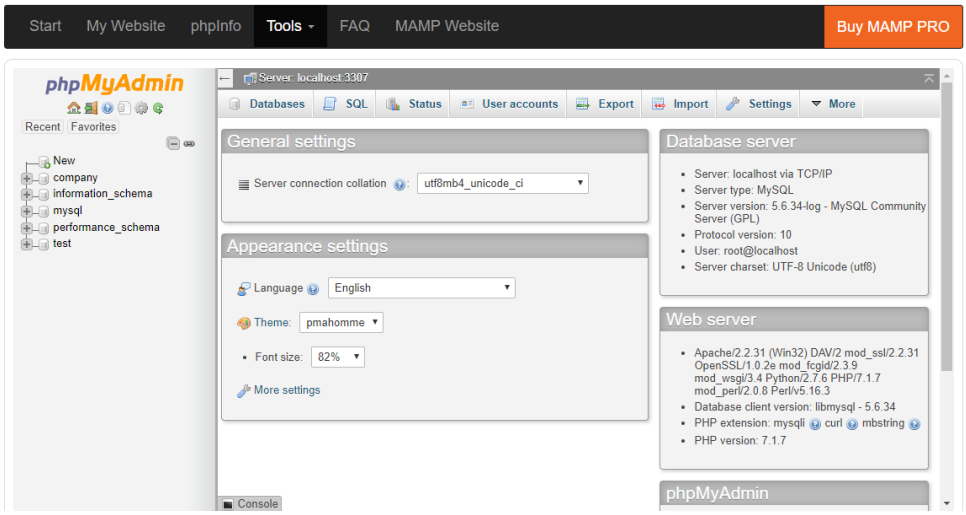


*FIGURE 5. Homepage of phpMyAdmin*

It is a combination of free to use software such as Apache, MySQL, PHP, so it is offered free of charge. The software used in MAMP are:

- **Apache**: It is an open source HTTP web server and a vital part of MAMP. Its modular structure helps it to be easily enhanced with the use of add-ons.
- **NGINX**: “Nginx is a web server. It can act as a reverse proxy server for HTTP, HTTPS, SMTP, POP3, and IMAP protocols, as well as a load balancer and an HTTP cache.” (https://www.mamp.info/en/, cited:07.05.2018).
- **MySQL:** It is used to develop a dynamic website. It is the most used relational database system and MAMP provides an easy to use MySQL interface in phpMyAdmin.
- **PHP:** It is the most commonly used server-side programming language for creating websites.

### 2.2.2 PHP

PHP stands for “Hypertext Preprocessor”. It is a general-purpose open-source scripting language especially suited for web development and is executed on the server. The extension for a PHP file is “. php”. A PHP file can include normal texts, HTML tags, CSS styles, JavaScript scripts, and PHP codes. The PHP codes are executed on the server and the result is passed to a browser as a plain HTML.

Everything happening in the back-end of a web-development could be done with PHP. It can generate dynamic contents. It can collect data from HTML forms and pass it to databases. It also could perform a CRUD (“Create Read Update and Delete”) operation in a database. Admin privileges, normal user and guest privileges could be set with the help of PHP. (PHP, Date of retrieval 24th May 2018)

- It is compatible with various OS platforms (Linux, Windows, Unix, Mac OS)
- It could be run in many servers (Apache, IIS)
- It can be paired with different databases.

**Syntax:** A PHP script can be placed anywhere in a document enclosed inside <? PHP and?> as shown in figure 6. PHP statements end with a semicolon (;).

“<? PHP

// PHP code goes here

?>”

```
<!DOCTYPE html>
<html>
<body>

<h1>My first PHP page</h1>

<?php
echo "Hello World!";
?>

</body>
</html>
```

**My first PHP page**

Hello World!

*FIGURE 6. Use of PHP in HTML*

### 2.2.3 PHP Framework

PHP is an open source, easy to use and compatible with different platforms many PHP frameworks have been created for more ease of use. A framework in a web development can be defined as a software which is designed to support the development of web resources, web APIs and web applications including web services. It provides many predefined functions such as libraries for database access, templates and session management. Code reusability is one of the most prominent features of any framework.

So, like any other Framework in web development, the PHP framework is a platform which provides structure and allows the user to develop web applications. It is possible to save plenty of time by using the PHP framework as applications can be faster with PHP frameworks. Some of them are listed below: (Sitepoint best PHP frameworks, Date of retrieval 27th May 2018)

i) Laravel
ii) CodeIgniter
iii) Yii
iv) Symfony
v) CakePHP

### 2.2.4 PHP(framework) security and threats

In this age of technology, anything and everything which are connected to the Internet(WWW) are prone to a cyber-attack, As PHP being one of the most used server-side programming languages it has got its own drawbacks and hacking incidents. But when any Framework is used, there is a lot of code which is generated by the framework itself and people generally do not notice some little details which could lead to a possible cyber-attack to the web application in the future. Thus, using a Framework makes the language easy but also increases the chance of hacking if not all the security measures are taken to neutralize the threat. Some of the most common threats and ways to prevent them in PHP are described below.

i) **SQL Injection:** Here the attacker enters an SQL fragment as a value in the URL or web form. The attacker can delete(drop) the entire database of the application or can modify or destruct the database so the application malfunctions.
This can be avoided by carefully monitoring any input from a user and using a PDO prepared statement instead of a MySQL statement. PDO separated the data from instructions so it prevents data being treated other than data.

ii) **XSS (Cross Site Scripting):** Here the attacker injects code (commonly JS code) into the output of the PHP script. This attack is possible when an input that was to be sent to a site is displayed on a web page. (Example: News article posting). The attacker can post JS code in the message which damages the web application severely.
Like SQL injection, the main way to prevent this attack is never trusting data coming from a user or any third-party sources. This attack can be prevented by the use of

Data Validation. Here we provide a strict guideline is provided for the data inputted by a user in a form. (Example: If an email is required, set the rule so that only a valid and formatted email ID could be entered).

iii) **Remote File Inclusion:** Adding a remote file into the application is a Remote File Inclusion. So here an attacker attacks a web application with the help of remote files. A file which is included in this web application tampers with malicious code is sent to the application via a remote file.
This can be prevented by modifying “php.ini” file. The “php.ini” file is a default configuration file. So, it can be turned “allow_url_fopen” off, which sets that external files could not be included. By default, this option is turned on.

iv) **Session Hijacking:** Session IDs are created when you are using the application and are they like a key to a safe deposit box. Stealing a session ID is called session hijacking.
“When a session is set up between a client and a web server, PHP will store the session ID in a cookie on the client side probably called PHPSESSID. Sending the ID with the page request gives the user access to the session info persisted on the server (which populates the super global $_SESSION array).”
(Sitepoint PHP security vulnerabilities, Date of retrieval 27th May 2018)
The attacker generally gets the session ID with an XSS attack so preventing it could also prevent session hijacking. Also modifying a “php.ini.” setting will prevent JS from being given access to the session ID.

### 2.2.5 Stripe payment

To inject a Stripe payment into the application, the following four steps were followed.

i) Setting up Stripe elements: The elements are directly loaded from the main server of stripe into the payment page. The instance is created with the help of a stripe element and a personal API-key.

ii) Creating a payment form: A simple form is created which is later injected in the stripe elements and is later mounted to every element container.

iii) Creating a token to secure card information: After the element is mounted on the card an event handler is created which handles the submit event on the form. This handler later sends the field to stripe for tokenization and prevents from resubmission of the data.

iv) Submission of token and handling the server: This is the last step where the token is sent to the stripe backend which handles all the server-side information.

# 3 LARAVEL

Laravel is a free open source Symfony based PHP web framework. It was created by Taylor Otwell. It follows an MVC (model-view-controller) pattern for the development of web applications. The source code of Laravel is hosted on GitHub and licensed under the terms of MIT license. (Laravel, Date of retrieval 27th May 2018)

To quote Laravel introduction page its philosophy is “Laravel is a web application framework with an expressive, elegant syntax. We believe development must be an enjoyable, creative experience to be truly fulfilling. Laravel attempts to take the pain out of development by easing common tasks used in the majority of web projects, such as authentication, routing, sessions, and caching.” (Laravel Introduction, Date of retrieval 27th May 2018)

## 3.1 Brief History of Laravel

Laravel was created to provide an alternative to the CodeIgniter framework (one of the most used PHP framework) which lacks certain features such as built-in support for user authentication and authorization. Taylor Otwell released the first beta of Laravel on June 9, 2011, and later in the same month came up with Laravel 1. Unlike CodeIgniter, Laravel included a built-in support for authentication, localization, models, views, sessions and another mechanism, but it lacked support for controllers that prevented it from being a true MVC framework. (Maxoffsky code-blog, Date of retrieval 27th May 2018).

Laravel 2 did not take much time to release as it was released on September 2011, mere months after its original beta release. This time they improved their shortcoming from Laravel 1 by creating the support for controller making it a fully MVC-compliant framework. Laravel 3 came just over a year later in February 2012. This was when Laravel’s user base and popularity began to grow.

In May 2013, Laravel 4 was released as a complete rewrite of the framework, incorporating a package manager called Composer. The composer is an application-level package manager for PHP that allowed the people to collaborate instead of competing.

(Medium, Date of retrieval 28th May 2018). Laravel 5 was released after a couple of years in February 2015.

In March 2015, a SitePoint survey listed Laravel as the most popular PHP framework. (Sitepoint best PHP framework, Date of retrieval 27th May 2018). A brief history of Laravel version releases is shown in figure 7.

| Version | Release date | PHP version | Notes |
|---|---|---|---|
| 1.0 | June 2011 | | NA |
| 2.0 | September 2011 | | NA |
| 3.0 | February 22, 2012 | | NA |
| 3.1 | March 27, 2012 | | NA |
| 3.2 | May 22, 2012 | | NA |
| 4.0 | May 28, 2013 | ≥ 5.3.0 | NA |
| 4.1 | December 12, 2013 | ≥ 5.3.0 | NA |
| 4.2 | June 1, 2014 | ≥ 5.4.0 | NA |
| 5.0 | February 4, 2015 | ≥ 5.4.0 | NA |
| 5.1 LTS | June 9, 2015 | ≥ 5.5.9 | NA |
| 5.2 | December 21, 2015 | ≥ 5.5.9 | NA |
| 5.3 | August 23, 2016 | ≥ 5.6.4 | NA |
| 5.4 | January 24, 2017 | ≥ 5.6.4 | NA |
| 5.5 LTS | August 30, 2017 | ≥ 7.0.0 | NA |
| 5.6 | February 7, 2018 | ≥ 7.1.3 | NA |

*FIGURE 7. History of Laravel* (Github Laravel releases, Date of retrieval 27th May 2018*).* Legends: Red- Old versions, Yellow – Older version, Still Supported, Green – Current stable version.

### 3.2 Installations

Laravel installation is easier than the other PHP frameworks because of the detailed documentation is given in the Laravel homepage. There is a separate instruction for installing Laravel according to the system of a developer.

### 3.2.1 Composer

A composer is a tool for a dependency management in PHP. It allows a user to declare the libraries this project depends on and it will manage them too. Composer requires a PHP version above 5.3.2 to run. The installer can be directly executed in the command line.

The composer is installed to a specific directory by using the –install-dir option and additionally (re)name it as well as using the --filename option.

```
php composer-setup.php --install-dir=bin --filename=composer
```

Or,

Composer PHAR can be placed anywhere. If it is placed in a directory of the project, it can be accessed globally. (Composer Installation, Date of retrieval 28th May 2018).

### 3.2.2 Server Requirement

The Laravel framework has a certain server requirement for its installation. Laravel Homestead satisfies all these requirements but is not a compulsory tool for using Laravel. So, if not using Laravel Homestead, following requirement for the server must be met.

- PHP >= 7.1.3
- OpenSSL PHP Extension
- PDO PHP Extension
- Mbstring PHP Extension
- Tokenizer PHP Extension
- XML PHP Extension
- Ctype PHP Extension
- JSON PHP Extension

FIGURE 8. Server Requirement for installation of Laravel *(Laravel 5.6 documentation Date of retrieval 28th May 2018)*

If all the requirements shown above in figure 8 are fulfilled, basically Laravel is installed by downloading Composer, then using Composer to download the Laravel installer followed by running the create-project command in Command line(Terminal). The Figure 9 shows Laravel installations process.

# Install Laravel

## Via Laravel Installer

First, download the Laravel installer using Composer.

```
composer global require "laravel/installer=~1.1"
```

Make sure to place the `~/.composer/vendor/bin` directory in your PATH so the `laravel` executable is found when you run the `laravel` command in your terminal.

Once installed, the simple `laravel new` command will create a fresh Laravel installation in the directory you specify. For instance, `laravel new blog` would create a directory named `blog` containing a fresh Laravel installation with all dependencies installed. This method of installation is much faster than installing via Composer.

## Via Composer Create-Project

You may also install Laravel by issuing the Composer `create-project` command in your terminal:

```
composer create-project laravel/laravel {directory} 4.2 --prefer-dist
```

FIGURE 9. Laravel Installations *(Laravel installations, Date of retrieval 27th May 2018).*

### 3.3 Configurations

Here are some of the basic configuration required once Laravel has been installed or when the application has been created:

- After the installation of Laravel, web server's document is configured to be the public directory. By default, the index.php serves as the front controller for all HTTP requests entering the application.
- All the configuration files for the Laravel framework are stored in the "config" directory. Every new configuration can be added by a user in this folder according to his/her needs.

- Some of the permissions for directories may need to be configured. The “storage” and the “bootstrap/cache” directory should be given “755” right if not using homestead. There is no need to configure the directory permissions if using homestead.
- The application key is needed to be configured once the Laravel application has been created. But if the application has been created using composer the application key is automatically generated by “PHP artisan key: generate”. If not using composer, then the application key can be configured from “. env” file.

Other configuration can be managed from the “config” directory according to the need of the application or project.

### 3.4 Routes

Routes provide the link between path and links. The path of the file in the project and the web links are linked by the help of routing. The basic method for defining routes is shown in figure 10.

```
Route::get('foo', function () {
    return 'Hello World';
});
```

FIGURE 10. Defining routes *(Laravel routing, Date of retrieval 8th September 2018)*

All the routes are stored in the routes directory. The routes/web.php file stores and defines routes for a web interface. The available router methods are shown in the figure 11.

```
Route::get($uri, $callback);
Route::post($uri, $callback);
Route::put($uri, $callback);
Route::patch($uri, $callback);
Route::delete($uri, $callback);
Route::options($uri, $callback);
```

FIGURE 11. Routes Method *(Laravel routing, Date of retrieval 8th September 2018)*

### 3.5 Controllers

Controllers provide options to define logic behind the handling of requests using HTTP Post and HTTP GET. Related request handling logic can be grouped into a single class using a controller. The root directory for Controllers is “app/Http/Controllers”.

The Figure 12 shows an example of a basic Controller class. The user-defined Controller class is extended to the “Controller” class included in Laravel. The base controller contains middleware method which may be used to attach the middleware to controller actions.

```php
<?php

namespace App\Http\Controllers;

use App\User;
use App\Http\Controllers\Controller;

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

You can define a route to this controller action like so:

```php
Route::get('user/{id}', 'UserController@show');
```

*FIGURE 12. Example of a basic controller class*
*(Laravel Controllers, Date of retrieval 8th September 2018)*

### 3.6 View

Views contain HTML codes of this application and separate the controller/application logic from the presentation logic. The views are stored in “resources/views” directory. An example of a view file is shown in figure 13.

```
<html>
    <body>
        <h1>Hello, {{ $name }}</h1>
    </body>
</html>
```

*FIGURE 13. Example of a view file* (Laravel views, Date of retrieval 8th September 2018)

# 4 IMPLEMENTATION (CREATING WEBSITE FOR AN INDIAN RESTAURANT INDIANCUISINE)

The project was developed using the technology described in the Web development and technology used a section of the thesis. The project was to develop a website for Indian restaurant “IndianCuisine”.

At first, an environment and a server were needed for the development of the website so by discussing with another member of the project, it was decided to choose MAMP as the local environment and it was set up. MAMP could be easily downloaded from https://www.mamp.info/en/, an official website for MAMP. The MAMP localhost directory structure is shown in the figure 14.

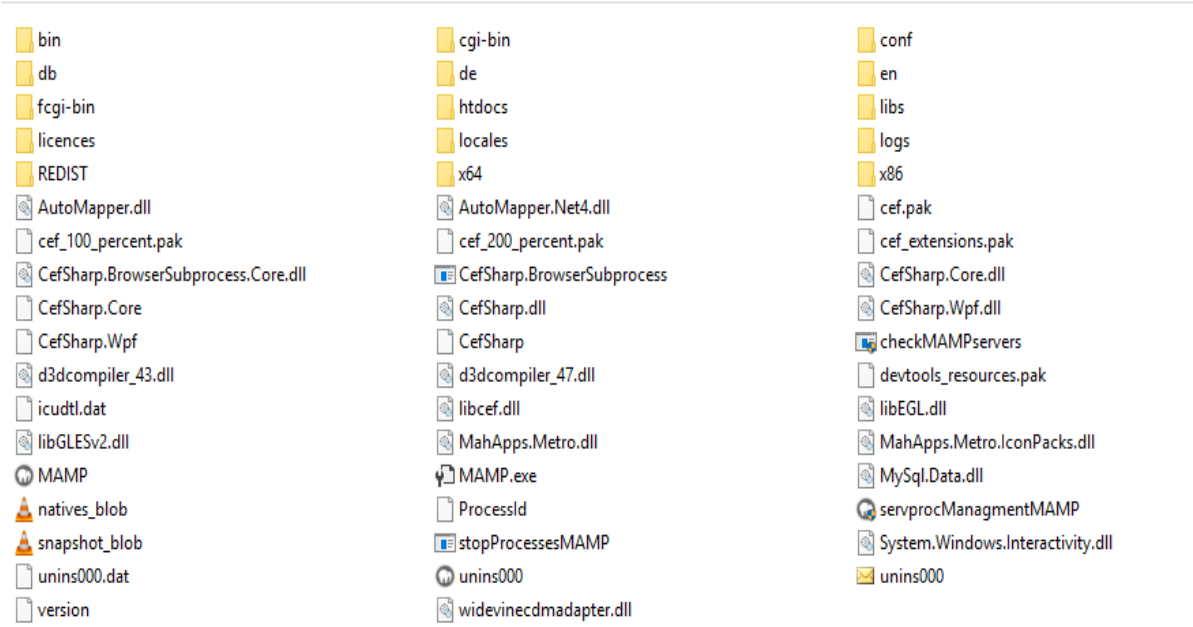


*FIGURE 14. MAMP localhost home directory*

Then a Laravel project was created inside “htdocs” with : php composer-setup.php --install-dir=bin --filename=composer.

After the creation of the Laravel project the home directory structure is shown in figure 15.

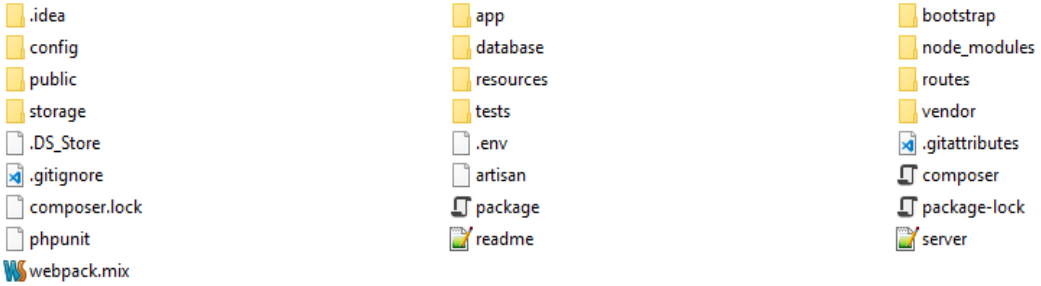


*FIGURE 15. Laravel home directory.*

All the controller files are stored inside an “app” directory. The assets of the project like CSS file, JS file, fonts, images are stored in public folder of the project And the UI or HTML view files are stored in the resource folder of the project.

As mentioned above in the theory section of the thesis a project could be divided into Front-end and backend parts. So also with the implementation, there were two parts, i.e. “Front-end implementation” and “Backend implementation”.

## 4.1 Front-end implementation

The author and his team member Bishal Shah discussed with the owner of “IndianCuisine” to decide the UI of the page. After going through some Bootstrap theme, the theme used in the project was decided. In addition, some components were extracted from other templates later on the project.

### 4.1.1 Usage of Bootstrap in Project

Bootstrap was downloaded from https://getbootstrap.com/docs/3.3/getting-started/ and a bootstrap.css file was placed in the public folder (“MAMP\htdocs\thesis\public\dist\css”).

Bootstrap was injected into the “main.blade.php” file with the line of code shown in the figure 16 which contains the header and footer of the project.

```
<link href="{{asset('dist/css/bootstrap.css')}}" type="text/css" rel="stylesheet" media="all">
```

*FIGURE 16. Inserting Bootstrap into the project.*

Once the Bootstrap had been injected into the project, the Bootstrap template was chosen according to the restaurant owner needs. The template is free to use and link to the template is “https://p.w3layouts.com/demos_new/template_demo/28-11-2017/haute_cuisine-demo_Free/651491422/web/index.html “.

The next task with Bootstrap was to filter out whatever necessary in the demo template and remove what was unnecessary. Also, some features were added according to need. So, to format the layout to be dynamic and responsive CSS files were saved in a CSS folder of the public directory. The main CSS files are listed below:

i) Bootstrap.css: Bootstrap.css is a core Bootstrap css file which includes styling of different Element, Classes, and IDs.

ii) Admin.css: Admin.css contains CSS related to an admin panel of the project. The admin panel consists of CRUD operation which a normal user cannot perform.

iii) Menu.css: The Menu.css file consists of styling related to menu page of the website. The menu page consists of the menu of a restaurant with the price of item and possibility of adding the item to the cart.

iv) Style.css: Style.css consists of general CSS which include CSS related to the common part of the website such as a header, a footer, a navigation bar, padding and the margin of the website.

### 4.1.2 Blade files.

The homepage consists of a Navigation bar, a menu, staffs, a gallery, a navigation, and a footer section. The Figure 17 shows the layout of the homepage followed by brief a description of major blade files.

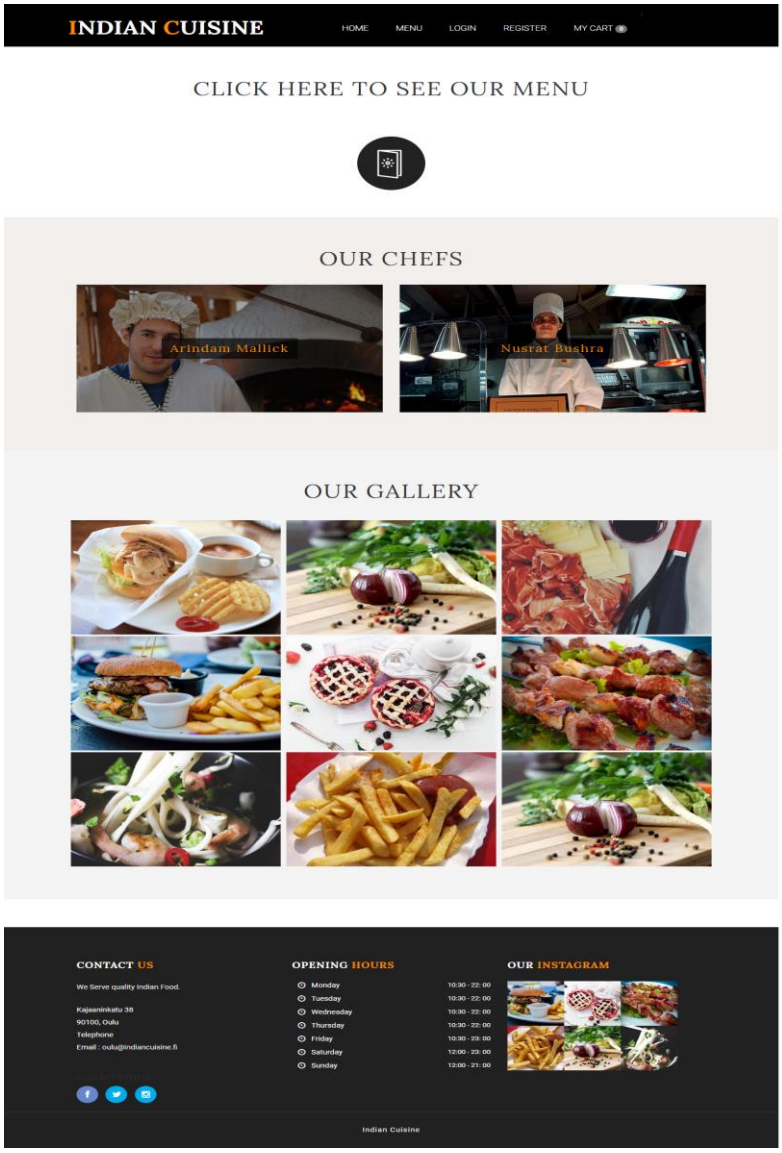


*FIGURE 17. Homepage of Website*

i) Login.blade.php and Register.blade.php

The Login blade has a simple login setup where an admin or a user can sign in with their email address and password. There is a link to reset the password if there is

some problem when logging in. Another link is to Register.blade.php where a new user can register their account. The form consists of a Name, a Phone number, an Email address, a Password and at last a confirmation password field. Once every field is filled with a valid value their account is created.

ii) Main.blade.php

Main.blade.php contains all the Header and Footer of the website. The linking of CSS files, JS files, fonts, and all other links are mentioned in the header section of this file, which is shown in figure 18.

```
<head>
    <title>Indian Cuisine</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

    <script type="application/x-javascript"> addEventListener("load", function() { setTimeout(hideURLbar, 0); }, false);
    function hideURLbar(){ window.scrollTo(0,1); } </script>
    <!-- Custom Theme files -->
    <link href="{{asset('dist/css/bootstrap.css')}}" type="text/css" rel="stylesheet" media="all">
    <link href="{{asset('dist/css/style.css')}}" type="text/css" rel="stylesheet" media="all">
    <link href="{{asset('dist/css/menu.css')}}" type="text/css" rel="stylesheet" media="all">
    <link rel="stylesheet" href="{{asset('dist/css/swipebox.css')}}">
    <link href="{{asset('dist/css/menu.css')}}" type="text/css" rel="stylesheet" media="all">
    <link rel="stylesheet" href="{{asset('//code.jquery.com/ui/1.11.2/themes/smoothness/jquery-ui.css')}}">


    <link href="{{asset('dist/css/font-awesome.css')}}" rel="stylesheet">


    <link href='//fonts.googleapis.com/css?family=Roboto:400,100,100italic,300,300italic,400italic,500,500italic,700,700italic,900,900italic'
    rel='stylesheet' type='text/css'>
    <link href="//fonts.googleapis.com/css?family=Lora:400,400i,700,700i" rel="stylesheet">


</head>
```

*FIGURE 18. Linking of assets in the project file "main.blade.php".*

The “Header” and “Footer” is same on all pages. It is written once in Main.blade.php which is later injected in all other pages wherever necessary. This is one of the exciting features of the Laravel blade template.

The Header contains a simple responsive collapsible navigation bar as shown below in the figure 19.

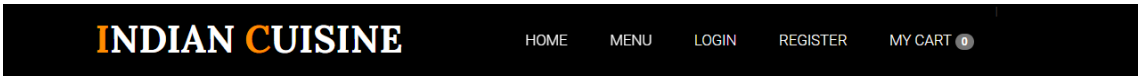


*FIGURE 19. Header Navigation*

The Footer contains Contact Info, Opening Hours and social media platform links as shown below in the figure 20.

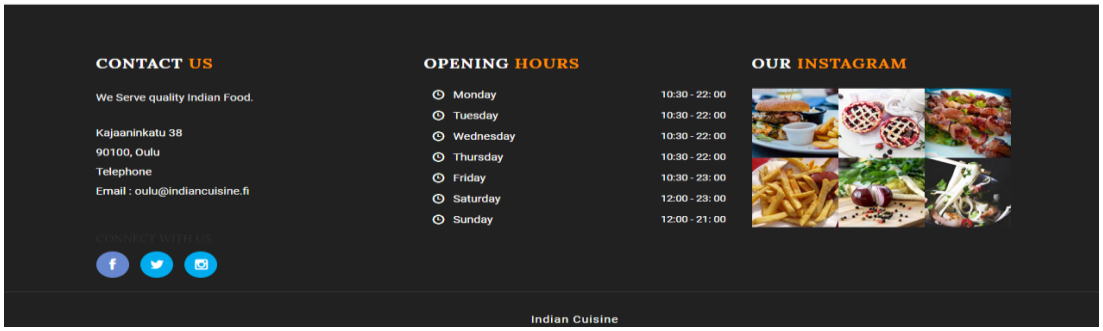


*FIGURE 20. Footer*

iii) Home.blade.php

Home.blade.php consists of the body of the homepage or the main content of the home page. There are 3 sections in the body part. The first section is the Link to the menu page which on clicking redirects to the menu page of the website. The second section contains the info about the owner of the restaurant. And finally, the third part has the images of delicacies served in the restaurant.

iv) Menus.blade.php

Menus.blade.php consists of a menu page which has categories such as chicken, lamb, beef and vegetable. And under each category, there is food related to it as shown below in figure 21.

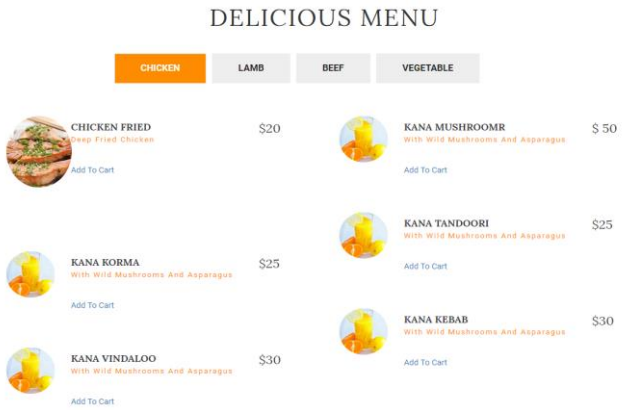


*FIGURE 21. Menus.blade.php page*

v) Cart (index.blade.php)

Once the item has been added to the cart from menus.blade.php, The Cart page is populated with the food item and its details. The details include the name of the item, the price of individual food items, the quantity of the item, the spiciness of the item, the option to remove or edit the selected item and an action to proceed with the selected item and go to checkout. The figure 22 shows the layout of the cart page.

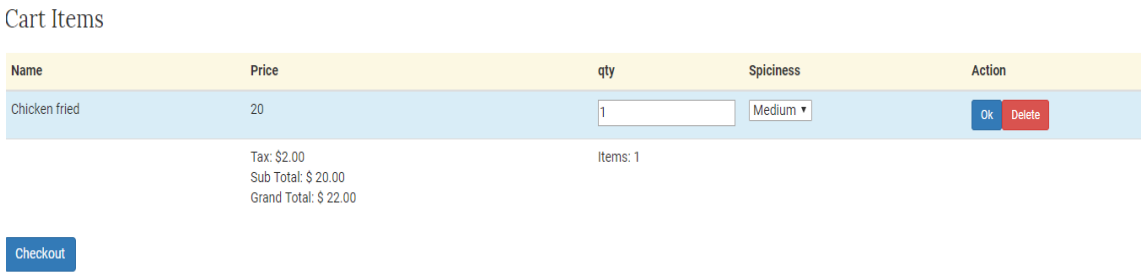


*FIGURE 22. The layout of cart page*

vi) Food-info.blade.php

When the checkout button on the cart page has been clicked it redirects the user to a food-info page where there is a simple form as shown in the figure 23 where the user can enter their details about date and time of pickup.

*FIGURE 23. Food-info form*

vii) Payment.blade.php

After “Proceed to payment” button has been clicked on the food-info.blade.php, the page redirects the user to the Payment page where the user can enter their card details and pay for the items.

## 4.2 Backend implementation

Backend refers to all the activities relating to server-side. It normally comprises servers and databases. Some components of the backend part of the thesis are briefly explained below.

### 4.2.1 Controllers

The Controllers used are listed and explained briefly below:

i) Authentication Controllers: This is a group of controllers which consist of Login, Register, Reset Password and Forget Password Controller. The above-mentioned Controllers deals with signing in, signing up and resetting password respectively. A middleware controller called Admin was created to design user-admin interface to block the user from getting access to the admin panel. The Admin class is shown in the figure 24.

```php
class Admin
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request $request
     * @param  \Closure $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        if (Auth::check() && Auth::user()->isAdmin()) {


            return $next($request);
        }
        return back();
    }
}
```

*FIGURE 24. Admin class*

ii) Address Controller: This controller creates, stores and validates address, time and date. The store function is shown below in the figure 25.

```php
public function store(Request $request)
{
    $this->validate($request,[
        'address'=>'required',
        'time'=>'required',
        'date'=>'required',


    ]);
    Auth::user()->address()->create($request->all());

    return redirect()->route('checkout.payment');

}
```

*FIGURE 25. Address Controller's store function*

iii) Cart Controller: The application uses the Gloudemans package for handling the cart controller. It adds various methods which accept the variety of parameter. For instance, an example is shown below in the figure 26 where the given function adds

a name, a price, and an ID of the product. Also, an optional key for spiciness is included. The default value for spiciness is medium.

```
public function addItem($id)
{
    $product=Product::find($id);
    Cart::add($id,$product->name,1,$product->price,['spiciness'=>'medium']);
    session()->flash('notif',' 1 Food added to your cart.');

    return back();
}
```

*FIGURE 26. Add Item function of Cart Controller*

iv) Categories Controller: This controller deals with all the activities related to the category of food from the menu. When a category is deleted first all the products of the category are deleted and then only the entire category is deleted as shown below in the figure 27.

```
public function destroy($id)
{
    $category=Category::find($id);
    $category->products()->delete();
    $category->delete();
    session()->flash('notif','Category successfully deleted.');

    return back();
}
```

*FIGURE 27. Destroy function of Category Controller.*

v) Checkout Controller: This controller deals with receiving an order by communicating with an order controller, notifying the user about the placement of their order via mail and also the payment of the order. This is shown below in figure 28.

```php
public function storePayment(Request $request)
{
    try {
        $charge = \Stripe\Charge::create(array(
            "amount" => Cart::total()*100, // Amount in cents
            "currency" => "usd",
            "source" => $request->stripeToken,
            "description" => "Indian Cuisine food charge"
        ));

    } catch (\Stripe\Error\Card $e) {
        // The card has been declined
    }
    //Create the order
    Order::createOrder();

    //redirect user to some page
    session()->flash('notif',' Thankyou for the purchase.You will be notified when the food is ready');
```

*FIGURE 28. Store Payment function of Checkout Controller.*

vi) Front Controller: This controller deals with rendering the Front end view of the web application.

vii) Home Controller: This controller checks whether the user is logged in or not and renders the homepage accordingly.

viii) Products Controller: This controller deals with the CRUD function of the products. Since the product has images as well, so it stores and specifies the image requirement(validation) and only if the requirement is met the images are uploaded and the new product is created as shown below in the figure 29.

```php
public function store(Request $request)
{
    $formInput=$request->except('image');

    //for validation
    $this->validate($request,[
        'name'=>'required',
        'spiciness'=>'required',
        'price'=>'required',
        'image'=>'image|mimes:png,jpg,jpeg|max:1000'

    ]);

    $image=$request->image;
    if($image){
        $imageName=$image->getClientOriginalName();
        $image->move('images',$imageName);
        $formInput['image']=$imageName;
    }
    Product::create($formInput);
    session()->flash('notif','New Product Added.');
    return redirect()->route('product.index');
}
```

*FIGURE 29. Image validation and creation of Product*

### 4.2.2 Model

A Model corresponds to the record in the database. It is extended from the eloquent master class and is presents a table in a database. The migration tool is used for creating, modifying and sharing the database scheme. The figure 30 illustrates an example of migration along with the model.

```php
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Product extends Model
{
    protected $fillable=['name','description','category_id','price','spiciness','image'];
    public  function category()
    {
        return  $this->belongsTo(Category::class);
    }


}
```

*FIGURE 30. Product Model*

After the migration file and table have created the model needs to be adjusted according to the database table. The model shown in the figure 30 illustrates the model with all its

attributes and relationships. The field from the database is injected into the $fillable class to prevent from the possible server-side error. The relationship is also established here. For instance, the model category is linked with the model product.

### 4.2.3 Database

A Database is an organized collection of data. In this project, a MySQL database management system is used. The website has a database name “company”. The tables and their functionality in the project are explained below briefly. The list of tables is shown in figure 31.

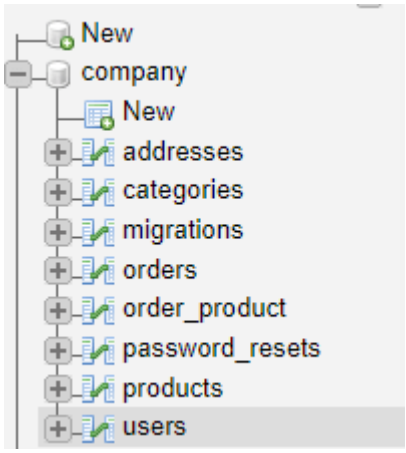


*FIGURE 31. Database and its tables*

i) Addresses table

Addresses table stores the address of the user, date and time of the pickup of order, ID, address, date and time are the database fields in this table. The figure 23 shows where the data inputted by the user is stored in this table.

ii) Categories table

Categories table stores the category of food which can only be appended by the user with admin privileges. This table has just two fields: an ID field and a name field. The ID is an autoincrement field and a name contains its respective category name. In the figure 21, the categories like Chicken, beef, vegetable have been generated from this table.

iii) Orders table

The Orders table has 4 fields: an ID field, a user_id field, a total field, and a ready field. Once the order has been made by a user, the orders are stored in this table. The ID is an auto increasing positive integer value, a user_id refers to the user who has ordered the item, a total refers to the total price of the item ordered and ready is a Boolean field which indicates whether the food is ready or not.

iv) Orders_product table

The orders_product table has 5 fields: an ID field, a product_id field, an order_id field, a qty field, and a total field. Here an ID field is autoincremented, product_id refers to the product/item in the menu, order_id refers to ID from Orders table, qty field refers to several the specific item and total here refers to the price of the ordered number of the item. The difference between this table and an order table is that this table lists the individual items ordered by the user and the Orders table stores all the item ordered by a user.

For example: If a user orders 2 fried chicken and 1 vegetable tempura, this table stores it as two entries: one for fried chicken and one for vegetable tempura whereas the Orders table stores it as a single order as it is ordered by the same user at once.

v) Password_resets table

As the name suggests the Password_resets table is used for resetting of the password. It has two fields: an email field and a token field.

vi) Products table

The Products table has 7 fields: ID, name, price, description, spiciness, image, and category_id. As always ID is auto increasing, the name stores the name of the dish, the price stores the price of the dish, the description has a brief description of the dish, the spiciness stores the default spiciness of the dish, the image stores the name of the image file stored in the server and category_id stores the category of the dish.

vii) Users table

The Users table stores the information of the user. It fields are ID, name, email, pnumber, password, confirmed, token, admin and remember token. ID is auto increment, the name stores the name of the user, the email stores the email, the pnumber stores the phone number, the password stores the hashed password of the user, the confirmed stores the Boolean value whether the user has confirmed the email or not while creating the account, the admin is another Boolean where 1 refers to the admin and 0 refers to the normal user.

# 5 CONCLUSION

At the beginning when I chose this topic to develop a website using the Laravel framework, I knew I was fighting an uphill battle. I have no previous experience with this framework and to be frank I was a little hesitant. I decided that this is a challenge I want to take as it has already taken me for five years to complete my bachelor’s degree and I wanted to prove myself that I can still learn new things. The first couple of months when learning and designing the website was challenging but I was not aware that the real challenge for me was writing about it in this thesis document.

The main objective of this project was to develop a responsive, client-friendly website with the Laravel framework for an Indian restaurant. Also, other objectives are listed in Chapter 1.3. I managed to complete all the objectives which I wanted to complete in this project. I completed the website and have given all the rights of use and further development of the project to the owner of the restaurant. Although it is not being used as the main website for the restaurant, the owner was happy with the work of a student who has never done any project with the Laravel framework.

Finally, to sum it all up although the website is not online right now the project process itself taught a lot of valuable life lessons as well as some new technology. The project has helped me to learn the Laravel framework, Bootstrap integration and using version control. It taught there needs to be constant communication between all parties to have a good and successful project.

## REFERENCES


Colorlib, Date of retrieval 15th May 2018, https://colorlib.com/wp/template/redcayenne/

Composer Installation, Date of retrieval 28th May 2018, https://getcomposer.org/doc/00-intro.md

Github, Date of retrieval 27th May 2018, https://github.com/laravel/laravel/releases

Laravel Introduction, Date of retrieval 27th May 2018, https://laravel.com/docs/4.2/introduction

Laravel, Date of retrieval 27th May 2018, https://laravel-news.com/voten-open-source

Laravel documentation, Date of retrieval 28th May 2018, https://laravel.com/docs/5.6

Laravel views, Date of retrieval 8th September 2018, https://laravel.com/docs/5.6/views

Laravel controllers, Date of retrieval 8th September 2018, https://laravel.com/docs/5.6/controllers#introduction

Laravel routing, Date of retrieval 8th September 2018, https://laravel.com/docs/5.6/routing

MAMP, Date of retrieval 16th May 2018, https://www.mamp.info/en/

Maxoffsky, Maxoffsky Code blog, Date of retrieval 27th May 2018, https://maxoffsky.com/code-blog/history-of-laravel-php-framework-eloquence-emerging/

Medium, Date of retrieval 28th May 2018, https://medium.com/vehikl-news/a-brief-history-of-laravel-5d55970885bc

PHP, Date of retrieval 24th May 2018, www.php.net

Sitepoint php security vulnerabilities, Date of retrieval 27th May 2018, https://www.sitepoint.com/top-10-php-security-vulnerabilities/

Sitepoint best php frameworks, Date of retrieval 27th May 2018, https://www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results/

W3schools CSS, Date of retrieval 19th May 2018 https://www.w3schools.com/css/default.asp

W3schools JS, Date of retrieval 19th May 2018, https://www.w3schools.com/html/html_scripts.asp

W3schools JS, Date of retrieval 19th May 2018, https://www.w3schools.com/js/default.asp

W3schools Bootstrap, Date of retrieval 20th May 2018, https://www.w3schools.com/bootstrap/default.asp