

Jamal Armel

Web application development with Laravel PHP Framework version 4

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Media Engineering

Thesis

11 April 2014



Author(s) Title Number of Pages Date	Jamal Armel Web application development with Laravel PHP Framework version 4 53 pages 11 April 2014
Degree	Bachelor of Engineering
Degree Programme	Media Engineering
Specialisation option	.NET application development and Hybrid Media
Instructor(s)	Aarne Klemetti, Senior Lecturer
<p>The purpose of this thesis work was to learn a new PHP framework and use it efficiently to build an eCommerce web application for a small start-up freelancing company that will let potential customers check products by category and pass orders securely. To fulfil this set of requirements, a system consisting of a web application with a backend was designed and implemented using built in Laravel features such as Composer, Eloquent, Blade and Artisan and a WAMP stack.</p> <p>The web application was built using the Laravel framework version 4, a modern PHP framework that aims at making PHP development easier, faster and more intuitive. The web application was built following the MVC architecture pattern. Admin panels were created for easily updating and managing the categories and products and uploading product images as well. A public interface was made available also to let registered users to log in and add orders to their carts and proceed to check out using <i>PayPal</i>.</p> <p>The application is easily expandable and features can be added or removed effortlessly thanks to the Laravel's ability to manage packages through Composer's Packagist online repository.</p> <p>The results proved that Laravel 4 is effectively a premium choice for a PHP framework that helps developers rapidly build secure, upgradable web applications.</p>	
Keywords	PHP, Laravel 4, MVC, Database, eCommerce

Contents

List of Abbreviations

1	Introduction	1
2	Laravel's main features	2
2.1	Architecture	2
2.2	MVC	4
2.2.1	Model	4
2.2.2	Views	4
2.2.3	Control	5
2.2.4	Database	5
2.3	Composer	7
2.4	Artisan	10
3	Creating the workflow and configuring our environment	11
3.1	Operating system	11
3.2	Terminal	11
3.3	Text editor	11
3.4	Bootstrap as the HTML5/CSS3/Javascript framework	12
3.5	Apache–MySQL–PHP package	13
3.6	Installing Composer	14
3.7	Installing Laravel 4	15
3.8	Database	16
4	Building the application with Laravel 4	20
4.1	Designing our application	20
4.1.1	The Idea	20
4.1.2	Entities, relationships and attributes	20

4.1.3	Map of the application	21
4.2	Creating a new application	23
4.2.1	Creating a main view	23
4.2.2	Creating the Eloquent models and their respective schemas	25
4.2.3	Image managing as an example of dependency management	30
4.2.4	Creating the Controllers and their respective Routes	32
4.2.5	Creating the views	38
4.3	Authentication and security	44
4.3.1	Authenticating users	44
4.3.2	Securing the application	48
5	Conclusion	50
	References	52

List of abbreviations

MVC	Model, View and Control
WAMP	Windows, Apache, MySQL, and PHP
PHP	Personal Home Page
DBMS	Database Management System
SQL	Structured Query Language
MySQL	My Structured Query Language
ORM	Object Relational Mapper
Apache	Apache HTTP Server
HTTP	HyperText Transfer Protocol
CRUD	Create, Read, Update and Delete
CSRF	Cross-Site Request Forgery
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
JSON	JavaScript Object Notation
URL	Uniform Resource Locator
XML	Extensible Markup Language
API	Application Programming Interface
UI	User Interface

1 Introduction

The purpose of this thesis work is to learn a new PHP framework and use it efficiently to build an eCommerce web application for a small start-up freelancing company that will let potential customers check products by category and pass orders securely. To fulfil this set of requirements, a system consisting of a web application with a backend will be designed and implemented using a modern MVC framework.

It is worthwhile considering the use of a PHP framework when time is a limitation and the developer's PHP coding skills do not match the high level demanded to build a complex application. Frameworks handle all the repetitive basic tasks of a PHP project, letting the developer concentrate her/his efforts on the business logic and the general structure of the project as a whole, in doing so, frameworks are becoming an ideal tool used by said developers to rapidly build complex operational prototypes in a matter of hours with minimal time spent on coding. Frameworks offer also whole range of ready-made utilities and libraries.

The use of a robust framework is recommended when the security of the web application is an essential requirement. It even becomes a necessity when the developer lacks the necessary know-how to prevent security breaches from happening. Most of the modern frameworks have built-in security features that range from input sanitising to automatic cookie encryption.

Organised structure of the project as a whole, clear and clean code are required when working in an organisation or co-developing an application in a team of developers. Frameworks permit the organisation of said code into a logical architecture, thus facilitating its maintainability and expandability. To achieve this, modern PHP frameworks follow the Model-View-Controller (MVC) architecture pattern.

Among the highly popular PHP frameworks, Laravel stands out with its claim in its ability to produce a development process that is agreeable for the developer without losing the application's functionality. That is one of the many reasons it was chosen as the framework of choice for building an eCommerce web application for Armel Solutions freelance start-up. This thesis work will study if Laravel lives up to its claim by evaluating its ability in building an up and running secure eCommerce web application in minimal time.

This is a sample, click download link to get the full Tutorial

CLICK BELOW

