

พัฒนาเว็บแอปพลิเคชันฐานข้อมูลด้วย Laravel 5

เรียนรู้การทำเว็บแอปพลิเคชันด้วย Laravel โดยใช้ฐานข้อมูล MySQL
พร้อมเวิร์คชอป

โดย เอกสิทธิ์ ศรีสุขะ

<http://www.codeexcellent.com>

<http://www.facebook.com/codeexcellent>

ชื่อผู้แต่ง : นายเอกสิทธิ์ ศรีสุขะ
ชื่อหนังสือ : พัฒนาเว็บแอปพลิเคชันฐานข้อมูลด้วย Laravel 5
จำนวนหน้า : 319 หน้า
พิมพ์ครั้งที่ : 1
จำนวน : 20 เล่ม (Print on demand)
เดือนปีที่พิมพ์ : มิถุนายน 2561
ISBN : 978-616-468-676-2
พิมพ์ที่ : บริษัท เอเชีย ดิจิตอลการพิมพ์ จำกัด
21/19-20 ถ.งามวงศ์วาน แขวงลาดยาว
เขตจตุจักร กรุงเทพมหานคร 10900
ออกแบบปก : นายเอกสิทธิ์ ศรีสุขะ
ราคา : 450 บาท

สงวนลิขสิทธิ์ตาม พ.ร.บ. ลิขสิทธิ์ พ.ศ. 2537
ห้ามลอกเลียนแบบไม่ว่าส่วนหนึ่งส่วนใดของหนังสือ/เอกสารเล่มนี้
นอกจากจะได้รับอนุญาตเป็นลายลักษณ์อักษร

แต่ พ่อ แม่ ผู้ให้กำเนิด

ครู อาจารย์ ผู้ประสิทธิ์ประสาทวิชา และผู้มีพระคุณทุกท่าน

คำนำ

หากผู้อ่านที่มีการใช้งานภาษา PHP และต้องการจะเลือกหาเฟรมเวิร์คสักตัว สำหรับใช้เป็นเครื่องมือเพื่อช่วยให้เราทำงานเร็วขึ้น หรือเพื่อต้องการโครงสร้างโค้ดโปรแกรมที่เป็นมาตรฐานตามหลักสากล ที่สามารถต่อเติมแก้ไข และสามารถนำไลบรารีสำเร็จรูปจากที่อื่นเข้ามาใช้งานได้ หนึ่งในเฟรมเวิร์คที่มีความน่าสนใจมากที่สุด คือ Laravel

Laravel ช่วยให้เราเริ่มสร้างแอปพลิเคชันได้อย่างรวดเร็ว โดยที่เครื่องมือที่จำเป็นหลายอย่าง ได้ถูกเตรียมพร้อมไว้ให้เราค่อนข้างครบแล้ว ทุกอย่างจบในตัวเอง ทำให้เราไม่จำเป็นต้องไปหาโปรแกรมอื่นมาติดตั้งเพิ่มเติมให้ยุ่งยาก ดังนั้นช่วยประหยัดเวลาของเราลงไปได้เป็นอย่างมาก

หนังสือเล่มนี้จัดทำขึ้น เพื่อใช้เป็นคู่มือภาษาไทยสำหรับท่านที่สนใจศึกษา Laravel 5 และต้องการที่จะนำมาใช้ในการทำงาน ซึ่งเนื้อหาจะเริ่มอธิบายจากขั้นพื้นฐานก่อน ตั้งแต่การติดตั้ง ไปจนถึงวิธีการนำขึ้น server เพื่อใช้งานจริง ซึ่งในหนังสือเล่มนี้ ท่านจะได้เรียนรู้เครื่องมือต่างๆที่ Laravel เตรียมไว้ให้เรา อาทิเช่น Artisans, Eloquent, Blade รวมไปถึงเทคนิคขั้นสูงอย่าง Email, Facebook Login, Service Provider และการทำงานร่วมกับฟรอนเอนด์ด้วย Ajax และ AngularJS เป็นต้น สุดท้ายนี้ผู้เขียนหวังว่าหนังสือเล่มนี้จะเป็นประโยชน์ ช่วยลดเวลาเรียนรู้ และช่วยให้ผู้อ่านได้สนุกกับการใช้งาน Laravel ครับ

ขอบคุณครับ

เอกสิทธิ์

ekkazit@gmail.com

สารบัญ

	หน้า
บทที่ 1 รู้จักกับ Laravel	1
แนะนำ Laravel	1
จุดเด่นของ Laravel	2
Text Editor	2
Composer	4
การสร้างโปรเจกต์	6
การทดสอบโปรเจกต์	7
โครงสร้างของโปรเจกต์	8
การตั้งค่าฐานข้อมูล	10
 บทที่ 2 Routing, View, Template และ Controller	 13
Routing	13
เริ่มต้นกับ Route	15
การ Route ด้วย Parameter	16
การตั้งชื่อ Route	18
การกำหนดกลุ่มให้กับ Route	19
การแสดง Route ทั้งหมดในระบบ	21
View	22
การส่งข้อมูลไปแสดงผลในวิว	23
การส่งข้อมูลในรูปแบบอาเรย์	25
การใช้คำสั่ง Compact	26
Template	27
การแสดงผลข้อมูล	28
การวนลูปแสดงผลข้อมูลในอาเรย์	29
คำสั่งเงื่อนไข	31
การทำ Master Page	32
การรวม Template	34

สารบัญ (ต่อ)

	หน้า
Controllers	36
การสร้าง Controller โดยใช้ Artisan	37
Resource Controller	40
Controller JSON Response	42
บทที่ 3 ออกแบบฐานข้อมูลด้วย Migration และ Seeding	45
ภาพรวมโปรเจกต์	45
Relationship	48
สร้าง Migration Scripts	49
Migration Data Type	50
การ Migrate และ Rollback	53
Seeding	56
การเพิ่ม Seed Data	59
การ Migrate Seed Data	61
บทที่ 4 การทำงานกับข้อมูลด้วย Models	63
รู้จักกับ Model	63
Eloquent ORM	64
การสร้าง Model	64
Model Relationship	66
การใช้งาน Tinker Shell	68
การทำงานกับข้อมูลด้วย Model	70
การแสดงผลข้อมูล	70
การจัดเรียงผลลัพธ์	74
การแสดงผลข้อมูลเป็นออบเจกต์	75
กลุ่มคำสั่ง Aggregation	78
การเพิ่มข้อมูล	80

สารบัญ (ต่อ)

	หน้า
การแก้ไขข้อมูล	81
การลบข้อมูล	82
บทที่ 5 การออกแบบหน้าจอด้วย Bootstrap	85
Assets	85
Bootstrap	86
Container	91
Grid System	92
Grid Offset	94
Navbar	96
Panel	98
Table	99
Form Components	102
Alert	106
Font-Awesome	107
Toastr	109
บทที่ 6 การแสดงรายการและแบ่งหน้าด้วย Pagination	113
Product Controller	113
การแสดงรายการสินค้า	117
หยาอดรวมตัวเลข	119
การกำหนดรูปแบบตัวเลข	120
การกำหนด Style ให้กับข้อมูล	121
การค้นหาข้อมูล	124
Pagination	129

สารบัญ (ต่อ)

	หน้า
บทที่ 7 ฟอร์มรับข้อมูล และการตรวจสอบความถูกต้อง	135
ติดตั้ง Form Plugin	135
การแสดงผลบนฟอร์ม	137
การสร้าง Dropdown List	140
การตรวจสอบความถูกต้องของฟอร์ม	143
การเขียนกฎการตรวจสอบ	144
การอัปโหลดรูปภาพ	150
การเพิ่มข้อมูลสินค้า	154
การลบข้อมูลสินค้า	157
 บทที่ 8 AngularJS และ Restful API	 163
รู้จักกับ AngularJS	163
การติดตั้ง	163
โปรแกรมแรกของเรา	167
2-Way Data Binding	169
ไคเร็กทีฟ	170
การกำหนด Format ข้อมูล	174
Filters	175
Restful API	177
AngularJS Service	179
AngularJS UI Rendering	180
การกรองด้วยประเภทสินค้า	184
การค้นหาสินค้า	189

สารบัญ (ต่อ)

	หน้า
บทที่ 9 Service Provider และ Middleware	195
Cart Controller	195
การเพิ่มสินค้าลงตะกร้า	197
การแสดงสินค้าในตะกร้า	200
การลบสินค้าในตะกร้า	203
การปรับปรุงจำนวนสินค้า	204
Service Provider	206
Facade	211
Middleware	214
บทที่ 10 Authentication	219
การสร้างระบบ Authentication	219
Database Migrations	221
หน้าจอล็อกอิน	222
การลงทะเบียนผู้ใช้ในระบบ	222
การปรับปรุงเลย์เอาต์หน้าจอ	223
สิทธิ์ในการใช้งานเมนู	225
การตรวจสอบสิทธิ์ด้วย Middleware	228
การล็อกอินด้วยชื่อผู้ใช้	230
ลิ้มรสผ่าน	231
Facebook Login	234
บทที่ 11 Reporting และ Chart	241
หน้าจอการชำระเงิน	241
ใบสั่งซื้อ	245
พิมพ์ใบสั่งซื้อ	250
จบการสั่งซื้อ	253

สารบัญ (ต่อ)

	หน้า
การติดตั้ง Chart.js	255
การใช้งาน Chart.js เบื้องต้น	258
การใช้งาน Chart.js กราฟวงกลม	261
การแสดงผล Chart.js โดยใช้ Ajax	262
บทที่ 12 Localization, Captcha และ PayPal	269
Localization	269
สร้างไฟล์แปลภาษา	270
การเปลี่ยนภาษาจากเมนู	273
Captcha	274
PayPal	277
บทที่ 13 Styling Your App และ Git	281
พื้นฐานของ CSS	281
CSS Selector	282
CSS Property และ Value	283
การเปลี่ยนตัวอักษร	284
การเปลี่ยนสีพื้นหลัง	285
การเปลี่ยนสี Navbar	286
การเปลี่ยนสี Panel และ List Group	287
คำสั่ง Git เบื้องต้น	289
บทที่ 14 การติดตั้งลงสู่ Server	295
ภาพรวมการติดตั้ง	295
เริ่มการติดตั้ง	296

รู้จักกับ Laravel

ในบทนี้จะเป็นการเกริ่นนำให้ผู้อ่านได้รู้จักกับ Laravel รวมไปถึงเครื่องมือที่ใช้ เพื่อให้ผู้อ่านที่กำลังเริ่มต้นศึกษา สามารถที่จะจัดเตรียมสิ่งต่างๆ ที่จำเป็นสำหรับการพัฒนาได้ นอกจากนี้ยังได้กล่าวถึงการติดตั้ง การสร้างโปรเจกต์ การทดสอบการทำงาน รวมไปถึงแนะนำโครงสร้างหน้าที่ของไฟล์และโฟลเดอร์ต่างๆ ที่อยู่ในโปรเจกต์อีกด้วย

แนะนำ Laravel

Laravel คือ เฟรมเวิร์คภาษา PHP ที่พัฒนาขึ้นเพื่อช่วยให้การสร้างแอปพลิเคชันทำได้ง่าย และใช้เวลาที่รวดเร็วมากยิ่งขึ้น โดยได้จัดเตรียมชุดคำสั่งและไลบรารีต่างๆ ให้เราได้ใช้งานอย่างมากมาย เรียกได้ว่าครบและจบในที่เดียว นอกจากนี้ยังมีการจัดวางโครงสร้างของโปรเจกต์ตามหลักของ **PSR-2** และ **PSR-4** ซึ่งเป็นมาตรฐานของการเขียนโปรแกรมภาษา PHP ยุคใหม่ จึงทำให้เราสามารถที่จะติดตั้งไลบรารี หรือส่วนเสริมต่างๆ ที่มีผู้พัฒนาไว้ดีแล้ว สามารถนำมาใช้ได้ทันที

Laravel เหมาะกับงานทุกประเภท เราสามารถนำมาใช้สร้างระบบต่างๆ ได้อย่างหลากหลาย ยกตัวอย่างเช่น ระบบ E-Commerce, งานระดับองค์กร, งานด้านฐานข้อมูล, โปรเจกต์ของนักศึกษา หรือนำมาใช้ทำระบบออนไลน์สำหรับธุรกิจสตาร์ทอัพ ก็มีความเหมาะสมเป็นอย่างยิ่ง ซึ่งอาจจะกล่าวได้ว่า หากเราค้นเคยกับการเขียน PHP อยู่แล้ว แต่กำลังจะมองหาเฟรมเวิร์คดีๆ ไว้ใช้งานสักตัว เพื่อใช้ผลิตผลงานออกมาให้รวดเร็วกว่าเดิม Laravel ถือเป็นตัวเลือกที่เหมาะสมที่สุด ที่นักพัฒนาทั่วโลกต่างเลือกใช้

จุดเด่นของ Laravel

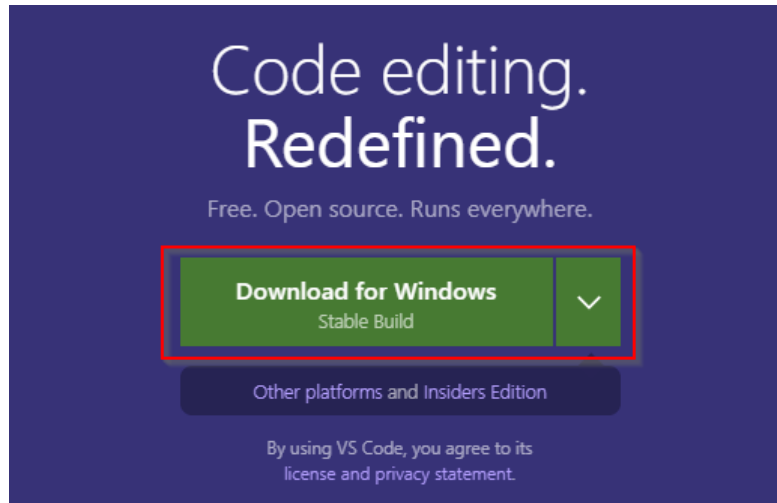
ปัจจุบันมี PHP เฟรมเวิร์คเกิดขึ้นอย่างมากมาย ซึ่งแต่ละตัวก็ล้วนแล้วแต่มีจุดเด่นของตัวเองที่แตกต่างกัน สำหรับ Laravel ของเรานั้น ก็มีสิ่งที่เป็นจุดเด่นที่เหนือกว่าเฟรมเวิร์คค่ายอื่นๆ อยู่มากมายหลายด้าน อาทิเช่น

- เตรียมเครื่องมือไว้ให้พร้อมแล้ว เช่น Route, Controller, Template, Models, Database และ Email เป็นต้น
- เตรียมคำสั่งไว้ช่วยลดงานที่ซ้ำซ้อน เพื่อเพิ่มความเร็วในการพัฒนา ได้แก่ Artisan
- เตรียมโมดูลที่ทุกระบบจำเป็นจะต้องมีไว้ให้เราแล้ว เช่น Login, Logout, Register, Forgot Password และการตรวจสอบสิทธิ์เมนูในหน้าจอ เป็นต้น
- รองรับไลบรารีจากภายนอก สามารถติดตั้ง และนำมาใช้งานได้อย่างง่ายดาย
- มีผู้สร้างเครื่องมือสนับสนุนเป็นจำนวนมาก เช่น Laravel Blade Snippets ซึ่งเป็นปลั๊กอินของ Visual Studio Code เป็นต้น
- มีผู้ใช้งานเป็นจำนวนมาก
- มีเว็บไซต์ เอกสาร หนังสือ บทความ และแหล่งความรู้เป็นจำนวนมาก ทั้งภาษาอังกฤษและภาษาไทย ซึ่งเราสามารถหาอ่านได้ง่าย

Text Editor

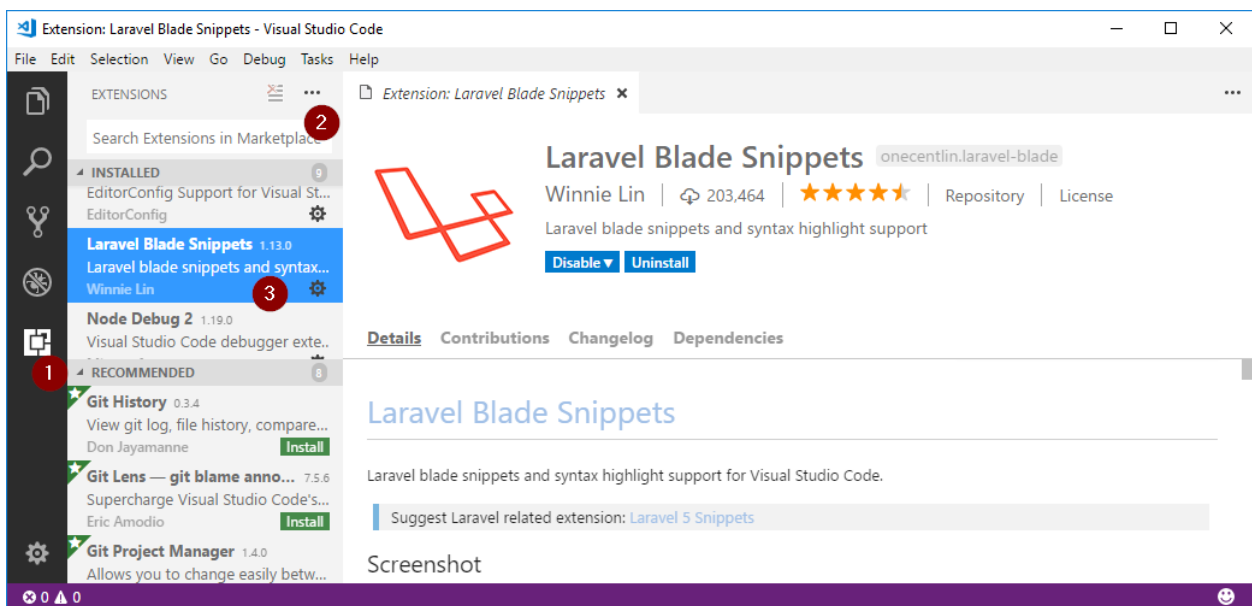
โปรแกรมอิดิเตอร์ส่วนใหญ่จะรองรับการเขียน PHP อยู่แล้ว เช่น Notepad++, Sublime Text, Atom หรือจะเป็นอิดิเตอร์ที่มีความสามารถสูง (แต่มีค่าใช้จ่าย) อย่าง PHPStorm เป็นต้น ท่านสามารถเลือกใช้ตัวใดก็ได้ตามความถนัด แต่ในหนังสือเล่มนี้จะเลือกใช้ Visual Studio Code เนื่องจากพัฒนาโดย Microsoft, ทำงานค่อนข้างรวดเร็ว, มีการอัปเดตอย่างต่อเนื่อง, รองรับการพัฒนาภาษาไทย และเราสามารถติดตั้งปลั๊กอินรองรับ Laravel ลงไปได้ ที่สำคัญคือเราสามารถดาวน์โหลดมาใช้งานได้ฟรี โดยเข้าไปดาวน์โหลดได้ที่

<https://code.visualstudio.com>



ภาพที่ 1-1 เว็บไซต์สำหรับดาวน์โหลด Visual Studio Code

หลังจากดาวน์โหลดเรียบร้อยแล้ว ให้ทำการติดตั้งโดยคลิก "Next" ไปจนจบกระบวนการ จากนั้นให้เปิดโปรแกรม Visual Studio Code ขึ้นมา แล้วทำการติดตั้งปลั๊กอินสำหรับพัฒนา Laravel ดังต่อไปนี้



ภาพที่ 1-2 ขั้นตอนการติดตั้งปลั๊กอินบน Visual Studio Code

จากภาพที่ 1-2 ขั้นตอนการติดตั้งจะเริ่มจาก คลิกที่ไอคอนหมายเลข (1) จากนั้นจะปรากฏหน้าจอให้ค้นหาปลั๊กอิน ให้พิมพ์ชื่อปลั๊กอินที่ต้องการ ลงในช่องหมายเลข (2) เมื่อพบแล้ว จึงเลือกปลั๊กอินตามหมายเลข (3) แล้วคลิกที่ปุ่ม "Install" เพื่อดำเนินการติดตั้ง

สำหรับรายชื่อของปลั๊กอินที่เราจะต้องติดตั้ง จะเป็นตามตารางที่ 1-1

ตารางที่ 1-1 ปลั๊กอินที่ติดตั้งใน Visual Studio Code

ปลั๊กอิน	ผู้พัฒนา	หน้าที่การทำงาน
PHP Intellisense	Felix Becker	จัดระเบียบโค้ดภาษา PHP และช่วยในการพิมพ์คำสั่ง
Laravel Blade Snippets	Winnie Lie	ช่วยในการแสดง Syntax ของ Blade Template รวมไปถึงคำสั่งลัด เพื่อช่วยให้พิมพ์ง่ายขึ้น

เมื่อเราติดตั้งปลั๊กอินครบทุกตัวแล้ว ให้เราแก้ไขคอนฟิกของ Visual Studio Code เล็กน้อย โดยเลือกที่เมนู **Preferences > Settings** หรือกดปุ่ม **Ctrl + Comma** จากนั้นเพิ่มโค้ดด้านล่างนี้ลงไป เพื่อให้ปลั๊กอินรองรับไฟล์ที่เป็นนามสกุล blade.php เป็นอันเสร็จสิ้นขั้นตอนในการปรับแต่งเครื่องมือของเรา

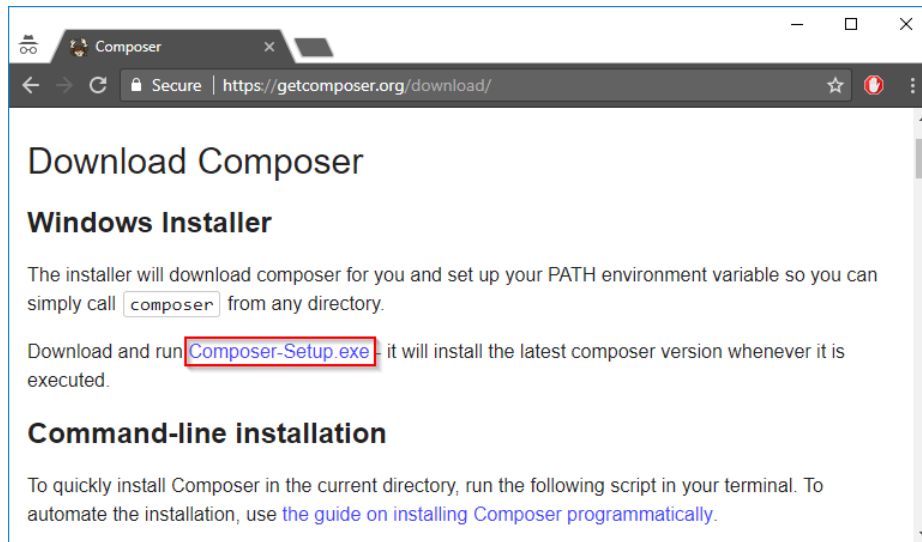
settings.json

```
"emmet.includeLanguages": {  
  "blade": "html"  
},
```

Composer

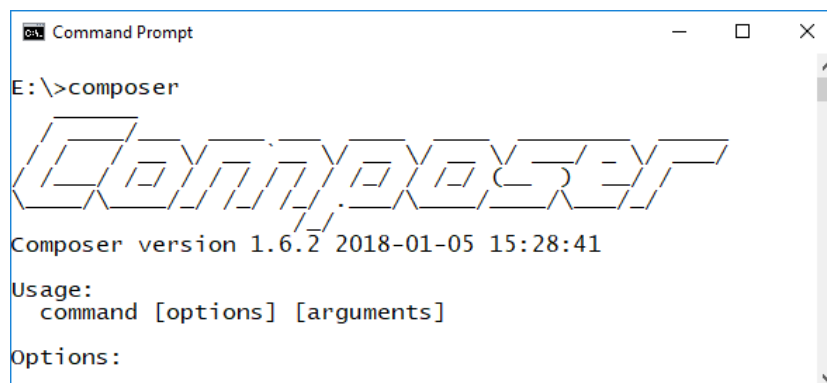
Composer เป็นตัวช่วยในการจัดการไลบรารี โดยทำหน้าที่ค้นหาไลบรารีที่เราต้องการ ทำการติดตั้ง อัปเดตเวอร์ชัน รวมถึงการถอนออกจากระบบของเรา ซึ่งเราจะใช้คำสั่ง **composer** สำหรับการสร้างโปรเจกต์ Laravel ของเรานี้ด้วย ดังนั้นเราจึงต้องทำการติดตั้ง composer ก่อนเป็นอันดับแรก

สำหรับการติดตั้งให้เข้าไปดาวน์โหลดที่เว็บไซต์ <https://getcomposer.org/download> จากนั้นคลิกดาวน์โหลดไฟล์ **Composer-Setup.exe** ตามภาพที่ 1-3



ภาพที่ 1-3 ขั้นตอนการดาวน์โหลด Composer

หลังจากที่ดาวน์โหลดเป็นที่เรียบร้อย ให้ดำเนินการติดตั้งไปตามปกติ แต่จะมีอยู่ขั้นตอนหนึ่งที่จะต้องเลือก PHP ดังนั้นเมื่อพบหน้าจอนี้ ก็ให้เรา Browse ไปยังโฟลเดอร์ที่อยู่ PHP ของเรา จากนั้นคลิก "Next" ไปจนเสร็จสิ้นการติดตั้ง คราวนี้เราจะมาทดสอบว่าใช้งานได้หรือไม่ ซึ่งจะทดสอบโดยพิมพ์คำสั่ง composer บน Command Prompt ดังต่อไปนี้



ภาพที่ 1-4 ทดสอบการใช้คำสั่ง composer

เมื่อทดสอบแล้วได้ผลลัพธ์ตามภาพ 1-4 แสดงว่าใช้งานได้ พร้อมแล้วที่จะใช้งานต่อไป แต่ถ้าไม่ได้ผลตามนี้ อันดับแรกให้เราตรวจสอบดูว่า PHP ถูกติดตั้งอย่างถูกต้องหรือไม่ หรือตรวจสอบ **PATH** ของ Environment Variable ขึ้นมายังโฟลเดอร์ PHP หรือเปล่า และถ้าหากตรวจสอบทั้งหมดแล้วพบว่าถูกต้อง อันดับที่สองให้เราเปิดไฟล์ **php.ini** แล้วเข้าไปดูในส่วนของ extension ชื่อว่า **php_openssl.dll**, **php_mbstring.dll** และ **php_curl.dll** ควรถูกเปิดใช้งาน ซึ่งปัญหานี้มักพบบ่อยในขั้นตอนเหล่านี้

การสร้างโปรเจกต์

เราจะมาเริ่มสร้างโปรเจกต์ Laravel โดยใช้คำสั่ง **composer** ดังต่อไปนี้

```
composer create-project laravel/laravel <project name> --prefer-dist
```

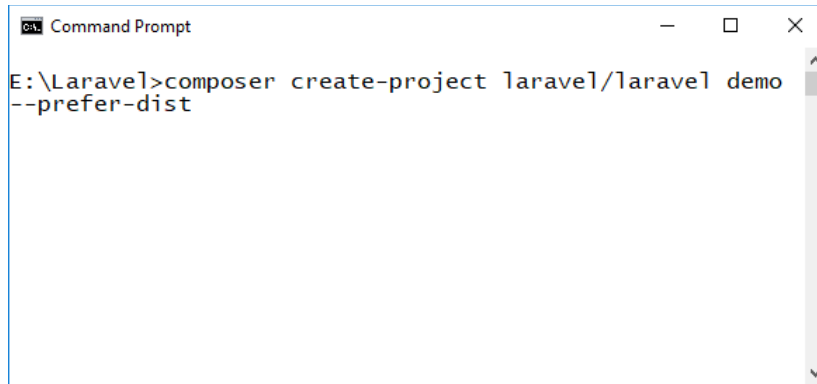
โดยที่	create-project	คือ คำสั่งในการสร้างโปรเจกต์
	laravel/laravel	คือ ชื่อแวนเดอร์ / แพ็กเกจ ที่จะไปดาวน์โหลดมาสร้างโปรเจกต์ให้เรา
	project name	คือ ชื่อโปรเจกต์ของเรา

ส่วนออฟชั่น

--prefer-dist คือ การสั่งให้ **composer** เลือกดาวน์โหลดไลบรารีที่เกี่ยวข้องกับโปรเจกต์เรา โดยให้เลือกเฉพาะตัวไลบรารีเท่านั้น โดยไม่ต้องนำซอร์สโค้ดมาด้วย

คราวนี้เรามาลองสร้างโปรเจกต์ใหม่ชื่อว่า **"demo"** โดยใช้คำสั่งข้างต้น

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



ภาพที่ 1-5 การสร้างโปรเจกต์ใหม่โดยใช้ composer

หลังจากนั้น composer ก็สร้างโปรเจกต์ให้กับเรา ซึ่งในครั้งแรกอาจจะใช้เวลา 4-5 นาที ในการดาวน์โหลดและติดตั้ง ซึ่งก็ขึ้นอยู่กับความเร็วอินเทอร์เน็ตของเราด้วย แต่ในการสร้างโปรเจกต์ครั้งถัดไป จะใช้เวลาเร็วขึ้นมาก เพราะมันใช้สิ่งที่เคยดาวน์โหลดในครั้งที่แล้ว ซึ่งเก็บอยู่ในเครื่องมาติดตั้งให้กับเรา

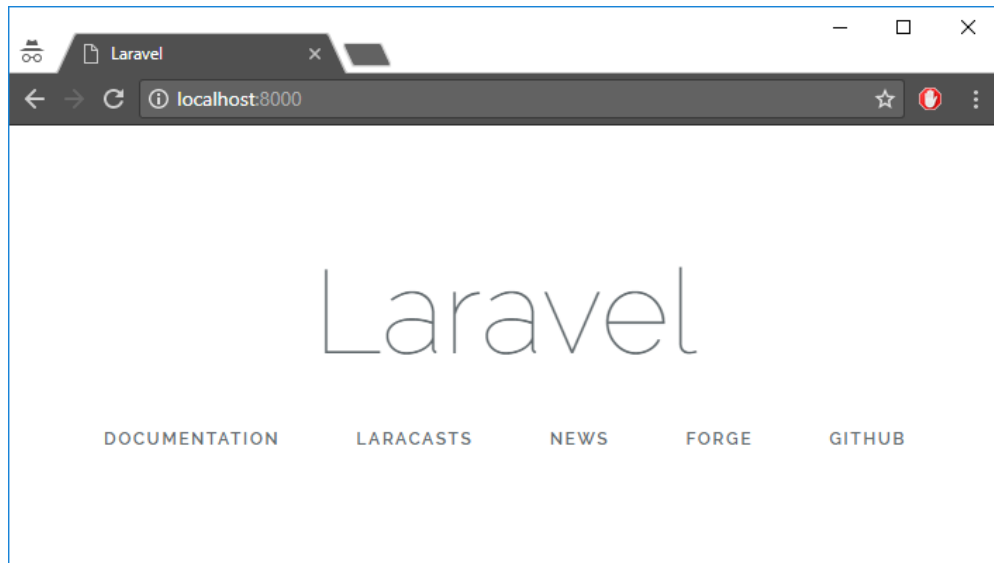
การทดสอบโปรเจกต์

ให้เข้ามายังโฟลเดอร์ demo ของเรา จากนั้นใช้คำสั่ง **php artisan serve** เพื่อสั่งให้เว็บเซิร์ฟเวอร์ทำงาน ซึ่งโดยปกติจะทำงานที่พอร์ต 8000 ตามภาพที่ 1-6



ภาพที่ 1-6 การสั่งให้เว็บเซิร์ฟเวอร์ทำงาน

หลังจากเซิร์ฟเวอร์เริ่มทำงานแล้ว ให้เราเปิดเบราว์เซอร์ขึ้นมา แล้วเข้าไปที่ยูอาร์แอล <http://localhost:8000> จะเห็นว่าโปรเจกต์ของเราสามารถที่จะใช้งานได้แล้ว โดยแสดงเป็นหน้าแรกขึ้นมาตามภาพ 1-7



ภาพที่ 1-7 หน้าแรกของเรา

โครงสร้างของโปรเจกต์

ให้เราเปิด Visual Studio Code ขึ้นมา จากนั้นเลือก **File > Open Folder** แล้วเข้ามายังโฟลเดอร์ demo ซึ่งเป็นโปรเจกต์ของเรา หรืออีกวิธีหนึ่งให้ใช้ command prompt โดยใช้คำสั่ง `cd` เพื่อเข้ามายังโปรเจกต์เรา แล้วใช้คำสั่ง `code .` (คำสั่ง `code` แล้วตามด้วยจุด) ก็จะเป็นการเปิดโปรเจกต์จากตำแหน่งที่เราอยู่เช่นเดียวกัน เราจะเห็นโครงสร้างของไฟล์และโฟลเดอร์ต่างๆ ที่ Laravel สร้างไว้ให้ สรุปโดยย่อตามตารางที่ 1-2 ดังนี้

ตารางที่ 1-2 โครงสร้างไฟล์ของโปรเจกต์

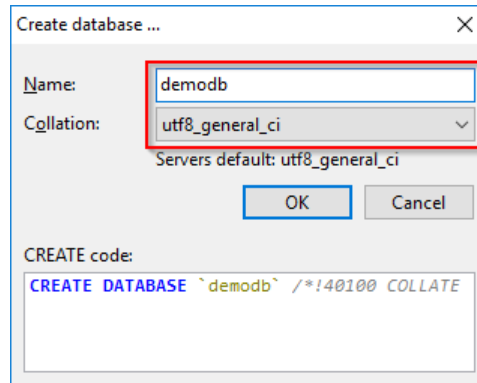
โฟลเดอร์/ไฟล์	หน้าที่
app	โฟลเดอร์ที่เก็บโค้ดโปรแกรมของเรา
bootstrap	โฟลเดอร์สำหรับเก็บโค้ดในช่วงที่ระบบเริ่มทำงาน
config	โฟลเดอร์สำหรับโค้ดในส่วนของคอนฟิก
database	โฟลเดอร์สำหรับไฟล์ฐานข้อมูล และไมเกรชั่น
public	โฟลเดอร์ที่เก็บ Assets เช่น CSS, JavaScript และ Images เป็นต้น
resources	โฟลเดอร์ที่เก็บ Assets ที่ยังไม่ถูกคอมไพล์ (SASS หรือ LESS) และเก็บไฟล์เกี่ยวกับการแปลภาษา รวมไปถึง HTML เทมเพลตของเรา
storage	โฟลเดอร์สำหรับสโตร์ข้อมูลเก็บไว้ เช่น การทำแคช (cache) และยังเก็บไฟล์ที่ได้จากการคอมไพล์ของ Laravel
routes	โฟลเดอร์ที่เก็บไฟล์สำหรับ routing ของระบบ
tests	โฟลเดอร์ที่เก็บไฟล์สคริปต์เพื่อทดสอบระบบ
vendor	โฟลเดอร์ที่เก็บไลบรารีจากภายนอกที่ composer โหลดมาไว้ให้เรา
.env	ไฟล์คอนฟิกของระบบ
artisan	ไฟล์คำสั่ง command line ของ Laravel
server.php	ไฟล์สำหรับสั่งให้ server ของ Laravel ทำงาน

ภายในโฟลเดอร์ **app** ยังมีโฟลเดอร์ย่อยที่สำคัญที่เราจะต้องทราบไว้ ซึ่งได้แก่

Console	โฟลเดอร์ที่ใช้เก็บคำสั่ง Artisan
Exceptions	โฟลเดอร์ที่เก็บไฟล์สำหรับการจัดการ Exception
Http	โฟลเดอร์ที่เก็บไฟล์ Controller และ Middleware
Providers	โฟลเดอร์ที่เก็บไฟล์สำหรับทำ Middleware ซึ่งเราจะกล่าวถึงในบทที่ 9 ต่อไป

การตั้งค่าฐานข้อมูล

หัวข้อสุดท้ายของบทนี้จะเป็นการตั้งค่าการเชื่อมต่อฐานข้อมูล MySQL โดยอันดับแรก ให้เราสร้างฐานข้อมูลชื่อ "demodb" ดังต่อไปนี้



ภาพที่ 1-8 การสร้างฐานข้อมูล MySQL ชื่อ demodb

จากภาพที่ 1-8 มีรายละเอียดอยู่บางส่วนที่ควรทราบ นั่นคือ Collation หรือ Charset ควรกำหนดให้เป็น **UTF-8** เพื่อให้ฐานข้อมูลรองรับการจัดเก็บภาษาไทยได้ หลังจากสร้าง database เป็นที่เรียบร้อยแล้ว ให้เปิดไฟล์ **.env** ในโปรเจกต์ของเรา จากนั้นแก้ไขค่าการเชื่อมต่อ ดังตัวอย่าง

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=demodb
DB_USERNAME=<username>
DB_PASSWORD=<password>
```

โดยที่

DB_DATABASE	คือ ชื่อฐานข้อมูล
DB_USERNAME	คือ username ของฐานข้อมูล
DB_PASSWORD	คือ password ของฐานข้อมูล

เมื่อสร้างฐานข้อมูลและแก้ไขไฟล์ .env เป็นที่เรียบร้อยแล้ว ก็เป็นอันจบขั้นตอนการติดตั้งในบทนี้ และพร้อมที่จะเข้าสู่เนื้อหาของ Laravel แบบเจาะลึกในบทถัดไป

สรุปท้ายบท

ในบทนี้เราได้กล่าวถึงเรื่องที่เราควรรู้ก่อนเริ่มพัฒนาเว็บแอปพลิเคชันด้วย Laravel ซึ่งได้กล่าวถึงในเรื่องของการติดตั้งเครื่องมือที่ใช้ในการพัฒนา, การติดตั้งปลั๊กอิน, การสร้างโปรเจกต์ และการทดสอบการทำงาน รวมไปถึงได้อธิบายโครงสร้างหน้าที่ของไฟล์ และโฟลเดอร์ต่างๆ อีกด้วย ในบทต่อไปเราจะเข้าสู่เนื้อหาพื้นฐานของ Laravel ได้แก่ Routing, Controller และ Views

Localization, Captcha และ PayPal

ในบทนี้จะกล่าวถึงในส่วนเสริมต่างๆ ซึ่งในบางระบบก็อาจจะไม่จำเป็นต้องใช้ก็ได้ โดยเราจะพูดถึงเรื่องการทำเว็บไซต์ให้รองรับหลายๆ ภาษา เพื่อให้ชาวต่างประเทศสามารถที่จะเข้าใจ และใช้งานฟีเจอร์ต่างๆ ได้อย่างเต็มความสามารถ นอกจากนี้เราจะกล่าวถึง Captcha ซึ่งเป็นวิธีการรักษาความปลอดภัยในเว็บไซต์แบบหนึ่ง และสุดท้ายเราจะมาสมัครใช้งาน PayPal และนำโค้ดมาติดตั้ง เพื่อรองรับการชำระเงินของโปรเจกต์เรา อีกด้วย

Localization

Laravel รองรับการทำ Localization (L10N) หรือการทำเว็บไซต์หลายภาษา ซึ่งได้วางโครงสร้างของโค้ดเพื่อให้เราสามารถใช้งานได้ทันที โดยองค์ประกอบของระบบแปลภาษานั้น จะต้องประกอบไปด้วยไฟล์ translation ซึ่งเป็นไฟล์ที่แปลเป็นภาษาต่างๆ ไฟล์ดังกล่าวจะอยู่ที่ **resources/lang** โดยจัดเก็บเป็นโครงสร้างตามภาพ 12-1

/resources
/lang
/en
messages.php
/th
messages.php

ภาพที่ 12-1 โครงสร้างไฟล์สำหรับแปลภาษาของ Laravel

จากภาพ 12-1 เราจะเห็นว่ามีไฟล์เดอร์ `en/messages.php` ซึ่งจะเป็นไฟล์ที่เก็บภาษาอังกฤษ และเราจะเห็นไฟล์เดอร์ `th/messages.php` ซึ่งก็จะเป็นไฟล์ที่แปลมาเป็นภาษาไทย เช่นเดียวกัน หากเรามีภาษาใหม่ๆ เช่น ภาษาจีน ก็ให้สร้างไฟล์เดอร์ และไฟล์ชื่อว่า `zh/messages.php` ก็สามารถรองรับภาษาจีนได้

สร้างไฟล์แปลภาษา

เรามาสร้างไฟล์สำหรับแปลภาษาไทยและอังกฤษชื่อ `messages.php` โดยบันทึกไว้ในไฟล์เดอร์ `resources/lang` ดังต่อไปนี้

resources/lang/en/messages.php

```
<?php

return [
    'product' => 'Product',
    'placeholder' => 'Type product code or name',
    'addToCart' => 'Add to Cart',
    'all' => 'All',
    'menu.home' => 'Home',
    'menu.cart' => 'Cart',
    'menu.login' => 'Login',
    'menu.register' => 'Register',
    'menu.product' => 'Product Mgt',
    'menu.report' => 'Report',
    'menu.logout' => 'Logout',
];
```

resources/lang/th/messages.php

```
<?php
return [
    'product' => 'สินค้าในร้าน',
    'placeholder' => 'พิมพ์ชื่อสินค้าเพื่อค้นหา',
    'addToCart' => 'หยิบใส่ตะกร้า',
    'all' => 'ทั้งหมด',
    'menu.home' => 'หน้าแรก',
    'menu.cart' => 'ตะกร้า',
    'menu.login' => 'ล็อกอิน',
    'menu.register' => 'ลงทะเบียน',
    'menu.product' => 'จัดการข้อมูลสินค้า',
    'menu.report' => 'รายงาน',
    'menu.logout' => 'ออกจากระบบ',
];
```

หลังจากที่เพิ่มไฟล์แปลภาษาเรียบร้อยแล้ว เมื่อเราใช้งานที่หน้าเว็บเพจจะใช้คำสั่ง `{{ __(messages.<key>) }}` หรือ `@lang(messages.<key>)` โดย messages คือ messages.php ของเรา ส่วน key ก็คือ key ของอาเรย์ที่แปลภาษาของเรา ยกตัวอย่างเช่น

`{{ __("messages.product") }}` จะได้ข้อความว่า “สินค้าในร้าน”

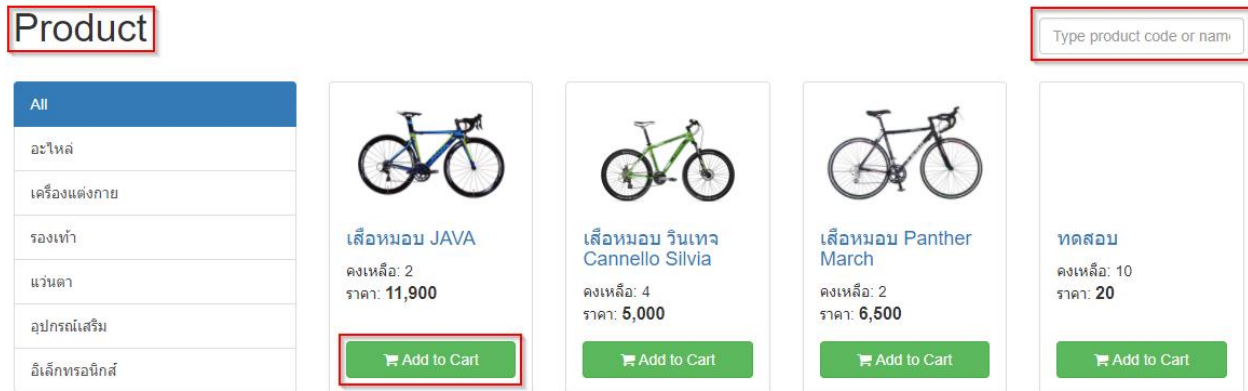
`{{ __("messages.placeholder") }}` จะได้ข้อความว่า “พิมพ์ชื่อสินค้าเพื่อค้นหา” เป็นต้น

ในขั้นตอนถัดมาเราจะมาแก้ไข home.blade.php และ master.blade.php โดยเพิ่มภาษาเข้าไปดังนี้

resources/views/layouts/master.blade.php

```
<ul class="nav navbar-nav">
    <li><a href="{ { URL::to('home') } }">{{__('messages.menu.home')}}</a></li>
    <li><a href="{ { URL::to('product') } }">{{__('messages.menu.product')}}</a></li>
    <li><a href="{ { URL::to('chart') } }">{{__('messages.menu.report')}}</a></li>
</ul>
```

จากนั้นทดสอบบนเว็บเบราว์เซอร์ จะเห็นว่าหน้าจอของเราเมนูเป็นภาษาอังกฤษเรียบร้อยแล้ว



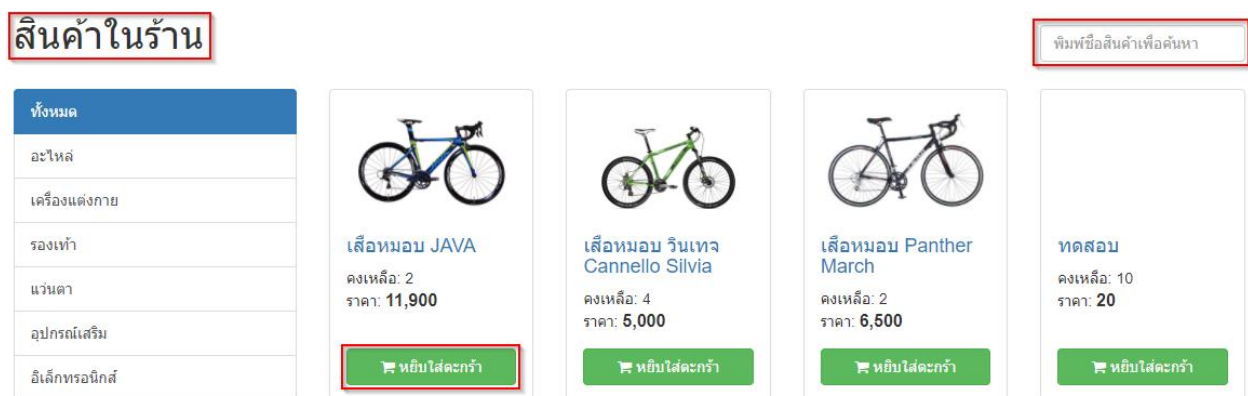
ภาพที่ 12-2 หน้าจอเมื่อใส่ไฟล์แปลภาษาแล้ว

หากเราต้องการให้เว็บของเราแสดงภาษาไทย ให้แก้ไขไฟล์ `config/app.php` ที่ locale ดังต่อไปนี้

`app/config.php`

```
'locale' => 'th',
```

หลังจากที่เราแก้ไขแล้ว ลอง refresh หน้าจออีกครั้ง จะเห็นว่าเว็บเพจของเราเปลี่ยนมาเป็นภาษาไทย ตามที่กำหนดไว้อย่างถูกต้อง



ภาพที่ 12-3 หน้าจอเมื่อเปลี่ยน Locale เป็นภาษาไทย

การเปลี่ยนภาษาจากเมนู

เราจะมาสร้างเมนูเพื่อให้ผู้ใช้คลิกเพื่อเปลี่ยนภาษาได้ง่าย โดยเปิดไฟล์ `master.blade.php` แล้วเพิ่มเมนูเพื่อการเปลี่ยนภาษาดังต่อไปนี้

resources/views/layouts/master.blade.php

```
<ul class="nav navbar-nav navbar-right">
  <li><a href="{{ URL::to('th') }}">TH</a></li>
  <li><a href="{{ URL::to('en') }}">EN</a></li>
  ...
</ul>
```

จากนั้นให้เพิ่ม route เพื่อรองรับการเปลี่ยนภาษาตามตัวอย่าง

routes/web.php

```
Route::get('/{locale?}', 'HomeController@index')->name('home');
```

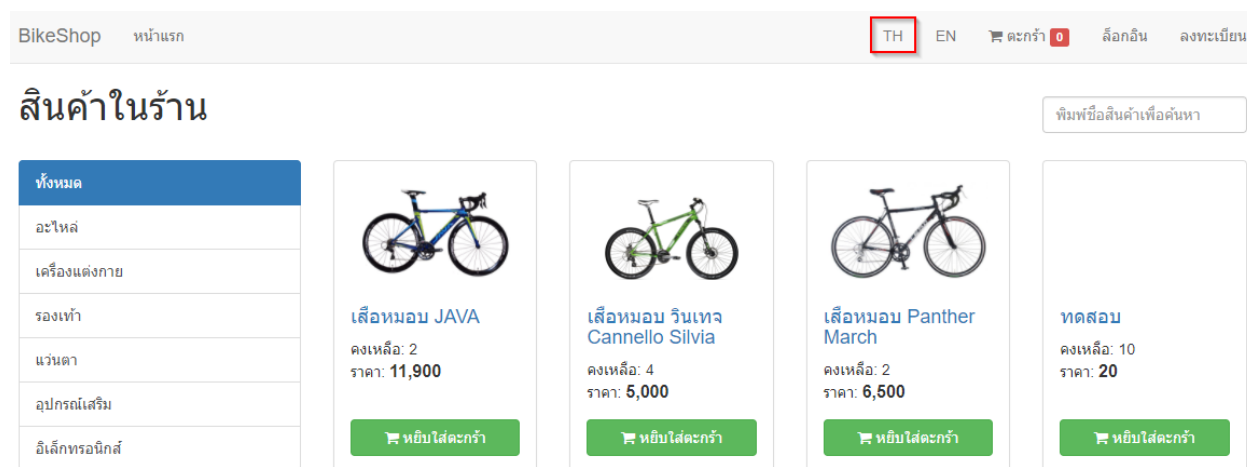
ขั้นตอนสุดท้าย ที่ HomeController ในฟังก์ชัน `index()` ให้เพิ่ม `$locale` เพื่อรองรับคลิกเปลี่ยนภาษาจากเมนูดังต่อไปนี้

app/Http/Controllers/HomeController.php

```
public function index($locale = null)
{
    if($locale) {
        app()->setLocale($locale);
    }
    return view('home');
}
```

จากโค้ดด้านบน จะเห็นว่าเราเพิ่มตัวแปร \$locale ลงในฟังก์ชัน index() ดังนั้นเมื่อผู้ใช้เปลี่ยนภาษา ก็จะทำให้ตัวแปร \$locale มีค่าตามภาษานั้นๆ เมื่อได้ค่าแล้วเราก็จะใช้คำสั่ง `app()->setLocale()` เพื่อทำการเปลี่ยนภาษาต่อไป

ทดสอบการเปลี่ยนภาษาโดยเปิดเบราว์เซอร์ขึ้นมาและลองคลิกที่ “TH” หรือ “EN” จะเห็นว่าภาษาเปลี่ยนตามที่เราต้องการ



ภาพที่ 12-4 การเปลี่ยนภาษาโดยคลิกที่เมนูบาร์

Captcha

เชื่อว่าผู้อ่านหลายท่านที่ใช้เว็บบ่อยๆ คงเคยเห็นขั้นตอนอยู่อันหนึ่ง ที่ให้เรากรอกตัวเลขตามภาพที่เห็น หรือมีป้ายที่เป็นรูปภาพให้เราเลือก ตามโจทย์ที่ต้องการ สิ่งเหล่านี้เรียกว่า “Captcha”

Captcha เป็นขั้นตอนการยืนยันตัวตนอย่างหนึ่ง เพื่อให้ระบบรู้ว่าสิ่งที่กำลังใช้ระบบอยู่นั้น คือ “คน” ไม่ใช่โรบอท ซึ่งโรบอทก็คือ “เว็บคลอเลอร์” (Web Clawler) ซึ่งเป็นโปรแกรมตัวหนึ่งสำหรับการเปิดดูเว็บไซต์ และทำการดาวน์โหลดเว็บไซต์เหล่านั้น โดยจะไต่ไปตามลิงค์ของเว็บไซต์ไปเรื่อยๆ ในบางหน้าจอก็อาจต้องการให้โรบอทเข้ามาทำงาน ซึ่งอาจจะหน้าจอก็มีความสำคัญสูง หรือเป็นความลับ เราก็จะต้องติดตั้ง Captcha เพื่อกันไว้ไม่ให้โรบอทเข้ามาทำงานในส่วนนั้นนั่นเอง

การติดตั้ง Captcha สำหรับ Laravel เราจะใช้ไลบรารี **zablose/captcha** ซึ่งติดตั้งโดยใช้คำสั่ง composer ดังนี้

```
composer require zablose/captcha
```

หลังจากติดตั้งเป็นที่เรียบร้อยแล้ว เรามาเพิ่มโค้ดในหน้าล็อกอินที่ไฟล์ login.blade.php เพื่อให้แสดง captcha ออกมาดังต่อไปนี้

resources/views/auth/login.blade.php

```
<div class="form-group">
    <div class="col-md-6 col-md-offset-4">
        <label></label>
    </div>
</div>
<div class="form-group{{ $errors->has('captcha') ? ' has-error' : '' }}">
    <label class="col-md-4 control-label">Captcha</label>
    <div class="col-md-6">
        <input type="text" class="form-control" name="captcha" autocomplete="off">
        @if ($errors->has('captcha'))
            <span class="help-block"><strong>{{ $errors->first('captcha') }}</strong></span>
        @endif
    </div>
</div>
```

จากนั้นในหน้าล็อกอิน เมื่อเราคลิกปุ่ม “เข้าสู่ระบบ” มันจะเข้ามาทำงานที่ LoginController ดังนั้นให้เราเพิ่มฟังก์ชัน **validateLogin()** สำหรับตรวจสอบความถูกต้องของ Captcha ดังต่อไปนี้

app/Http/Controller/Auth/LoginController.php

```
use Illuminate\Http\Request;

class LoginController extends Controller{
    ...
    protected function validateLogin(Request $request){
        $this->validate($request, [
            $this->username() => 'required',
            'password'        => 'required',
            'captcha'         => 'required|captcha',
        ]);
    }
}
```

เรามาทดสอบการทำงาน โดยเปิดเบราว์เซอร์แล้วเข้ามายังหน้า Login ผู้อ่านจะเห็น Captcha ปรากฏขึ้น และหากเราใส่ข้อมูลถูกต้องตามรูปภาพ ก็จะสามารถล็อกอินเพื่อเข้าระบบได้ ในทำนองเดียวกัน หากใส่ข้อมูลไม่ตรงกับรูปภาพ หรือ ไม่ได้ใส่ข้อมูลใดๆ ระบบจะแสดงข้อผิดพลาด และไม่สามารถเข้าสู่ระบบได้ ตามภาพที่ 12-5

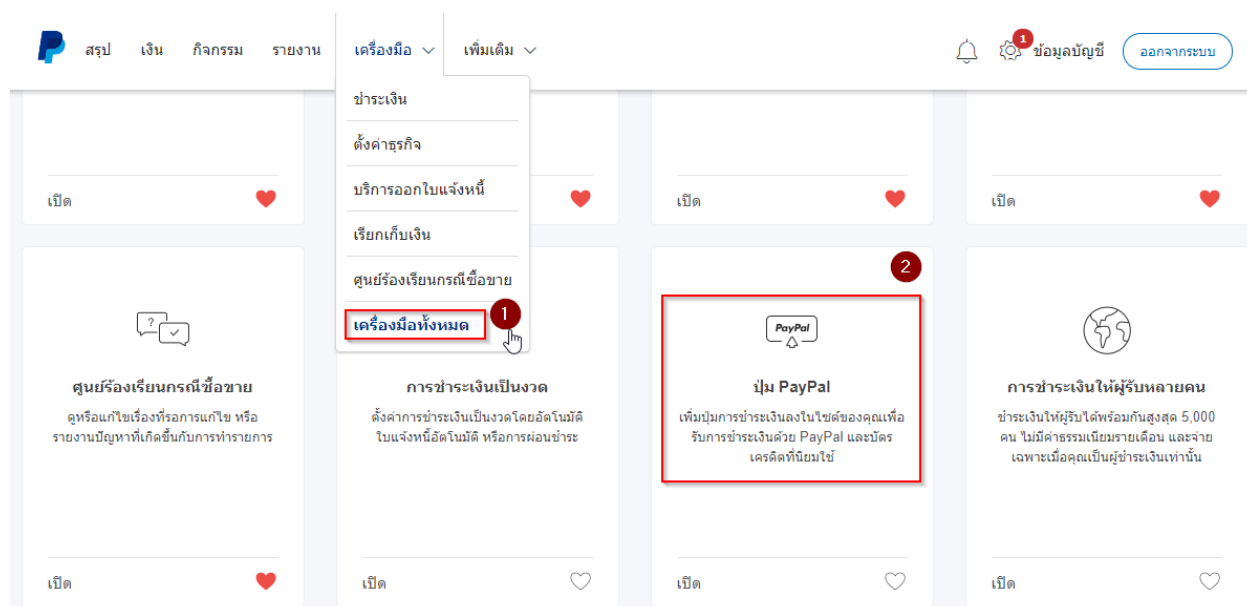
ภาพที่ 12-5 ทดสอบการทำงานของ Captcha กรณีกรอกข้อมูลไม่ถูกต้อง

PayPal

ในหัวข้อสุดท้ายของบทนี้ เราจะมาทำระบบชำระเงินด้วย PayPal เพื่อเพิ่มช่องทางการชำระเงินในเว็บไซต์ของเรา ให้ง่ายต่อการใช้งานของลูกค้าอีกด้วย

วิธีติดตั้ง PayPal ในเว็บไซต์เรานั้น ผู้เขียนขอแบ่งตามแนวทางการใช้ ซึ่งจะมี 2 วิธีคือ วิธีแรกใช้ PayPal สคริปต์ ซึ่งเราจะเข้าไปสร้างปุ่มชำระเงินที่เว็บไซต์ของ PayPal แล้ว copy สคริปต์นั้นมาใช้ในเว็บเรา (คล้ายกับปุ่ม Like บน Facebook) และวิธีที่สอง ให้ติดตั้งไลบรารี Laravel ที่เชื่อมต่อกับ PayPal Gateway จากนั้นเขียนโค้ดเพื่อรองรับการทำงานเอาเอง

เพื่อความง่าย ในหนังสือเล่มนี้จะเลือกใช้วิธีแรก โดยเริ่มต้นให้เราเข้าไปสมัครสมาชิกในเว็บ www.paypal.com หลังจากสมัครสมาชิกแล้ว ให้คลิกที่เมนู **เครื่องมือ > เครื่องมือทั้งหมด** จากนั้นคลิกที่ **ปุ่ม PayPal**



ภาพที่ 12-6 การสมัครใช้บริการปุ่มรับชำระเงินด้วย PayPal

จากนั้นเลือกการสร้างปุ่ม “ซื้อทันที” และกำหนดค่าต่างๆ เช่นอัตราภาษี เช่นต้น ตามภาพที่ 12-7

ชื่อสินค้า ID รายการสินค้า (ไม่จำเป็นต้องระบุ) [นี่คืออะไร](#)


ราคา สกุลเงิน
 THB ▼ [ต้องการหลายๆราคา?](#)

ปุ่มที่ปรับแต่งเองได้

- ☐ เพิ่มเมนูรอปดาวน์ที่มีราคา/ตัวเลือก [ตัวอย่าง](#)
- ☐ เพิ่มเมนูรอปดาวน์ [ตัวอย่าง](#)
- ☐ เพิ่มฟิลด์ข้อความ [ตัวอย่าง](#)
- ▶ [ปรับแต่งข้อความหรือลักษณะที่ปรากฏ](#) (ระบุหรือไม่ก็ได้)

มุมมองลูกค้าของคุณ

ซื้อทันที



การส่งสินค้า

ใช้จำนวนเงินเฉพาะ: THB [ความช่วยเหลือ](#)

ภาษี

ใช้ส่วนลดภาษี %

ID บัญชีผู้ค้า [ดูข้อมูลเพิ่มเติม](#)

☒ ใช้ ID บัญชีผู้ค้าแบบปลอดภัยของจีน

ภาพที่ 12-7 การตั้งค่าเพื่อสร้างปุ่ม “ซื้อทันที” ของ PayPal

จากนั้นคลิกที่ “บันทึกการเปลี่ยนแปลง” ระบบจะแสดงสคริปต์ให้เรา copy ไปใส่ในเว็บไซต์ของเรา


เว็บไซต์ อีเมล

```
<form action="https://www.paypal.com/cgi-bin/webscr" method="post" target="_top">
<input type="hidden" name="cmd" value="_s-xclick">
<input type="hidden" name="hosted_button_id" value="DW2KQ62M8FK9Q">
<input type="image" src="https://www.paypalobjects.com/th_TH/TH/i/btn/btn_buynowCC_LG.gif" border="0" name="submit" alt="PayPal - วิธีชำระเงินแบบออนไลน์ที่
```

[เลือกรหัส](#) [กลับไปเพื่อแก้ไขปุ่มนี้](#)

มุมมองของผู้ซื้อ

ซื้อทันที



ภาพที่ 12-8 สคริปต์ที่ PayPal สร้างให้ สำหรับนำไปใช้ในเว็บไซต์ของเรา

ให้เราเปิดไฟล์ checkout.blade.php จากนั้นนำสคริปต์ PayPal ที่ได้ไปใส่ไว้ในหน้าดังกล่าว ดังต่อไปนี้

resources/views/cart/checkout.blade.php

```
<div class="panel panel-primary">
  <div class="panel-heading">
    <div class="panel-title"><strong>ข้อมูลลูกค้า</strong></div>
  </div>
  ...
  <div class="panel-footer">
    <form action="https://www.paypal.com/cgi-bin/webscr"
      method="post" target="_top">
      <input type="hidden" name="cmd" value="_s-xclick">
      <input type="hidden" name="hosted_button_id" value="DW2KAD2M8FK9Q">
      <input type="image" src="..." border="0" name="submit" alt="...">
      
    </form>
  </div>
</div>
```

จากนั้นเมื่อดูในหน้าเว็บไซต์ของเรา จะเห็นว่าไอคอน “ซื้อทันที” ได้แสดงในหน้าจอแล้ว

ชำระเงิน




🏠 หน้าร้าน / สินค้าในตะกร้า / ชำระเงิน

รายการสินค้า			
รหัส	ชื่อสินค้า	จำนวน	ราคา
🚲	P003 เสื้อหมอบ Panther March	1	6,500
รวม		1	6,500

ข้อมูลลูกค้า
ชื่อ-นามสกุล

อีเมล

ซื้อทันที

ยืนยันการสั่งซื้อ

ชำระเงิน

ภาพที่ 12-9 การวางไอคอนซื้อทันทีในเว็บไซต์เรา

แม้ว่าเราจะวางไอคอนลงไปแล้ว แต่ก็ยังใช้งานได้ไม่สะดวกมากนัก เพราะผู้ใช้งานยังต้องกรอกจำนวนเงิน และจำนวนที่ซื้อเข้าไปเอง ดังนั้นให้เราเพิ่ม **quantity** และ **amount** ในไอคอนของ PayPal เพื่อที่ผู้ใช้คลิกปุ่ม จะไม่ได้ไม่ต้องกรอก จำนวนเงินลงไปอีกแล้ว

```
<form action="https://www.paypal.com/cgi-bin/webscr" method="post" target="_top">
  <input type="hidden" name="cmd" value="_s-xclick">
  <input type="hidden" name="hosted_button_id" value="DW2KAD2M8FK9Q">
  <input TYPE="hidden" name="charset" value="utf-8">
  <input type="hidden" name="currency_code" value="THB">
  <input type="hidden" name="business" value="อีเมลของคุณ@gmail.com">
  <input type="hidden" name="item_name" value="ชื่อสินค้าของคุณ">
  <input type="hidden" name="quantity" value="จำนวนที่ซื้อรวม">
  <input type="hidden" name="amount" value="จำนวนเงินที่ต้องชำระ">
  <input type="image" src="..." border="0" name="submit" alt="...">
  
</form>
```

ทดสอบการทำงานเมื่อลูกค้าคลิกที่ “ซื้อทันที” ระบบจะ redirect ไปยังหน้า PayPal ซึ่งถ้าหากลูกค้าคนนั้น มี account ของ PayPal ก็จะสามารถที่จะล็อกอินเพื่อชำระเงินตามมูลค่าสินค้าของเราได้ โดยลูกค้าสามารถชำระเงินได้ทันที ไม่จำเป็นต้องกรอกอะไรเพิ่มเติมอีกแล้ว

สรุปท้ายบท

ในบทนี้เราได้เรียนในเรื่องของส่วนเสริมต่างๆ ที่ช่วยอำนวยความสะดวก เพิ่มยอดการเข้าชม และสร้างความปลอดภัยให้ระบบได้ดียิ่งขึ้น เราได้เรียนในเรื่องการทำเว็บไซต์หลายภาษา เริ่มจากการสร้างไฟล์แปลภาษา จากนั้นนำมาใช้ในเทมเพลต และทำการเลือก Locale ในการแสดงภาษา นอกจากนี้เราได้เรียนเรื่องของ Captcha เพื่อช่วยป้องกันไม่ให้โรบอทเข้ามาทำงาน ในหน้าเว็บที่เราต้องการปกป้อง และเรื่องสุดท้าย เราได้เรียนการใช้ PayPal เพื่อช่วยให้ลูกค้าสามารถชำระเงินได้ง่ายขึ้น เพิ่มช่องทางการชำระเงินในเว็บไซต์ของเรา โดยเริ่มจากการสมัครใช้บริการ การสร้างไอคอน “ซื้อทันที” และการนำสคริปต์มาใช้ในเว็บไซต์ของเรา พร้อมกับปรับแต่งสิ่งต่างๆ เพิ่มเติม

การติดตั้งลงสู่ Server

ในบทนี้เราจะมาทำการติดตั้งโปรเจกต์ของเรา ลงในเซิร์ฟเวอร์เพื่อใช้งานจริง โดยเราจะใช้ Ubuntu Server เป็นแพลตฟอร์มหลักในการทำงาน ผู้อ่านจะได้เรียนรู้ขั้นตอนการติดตั้งโดยละเอียด และยังสามารถเรียนรู้การใช้คำสั่งอื่นๆอีกด้วย รายละเอียดขอเชิญติดตามได้เลยครับ

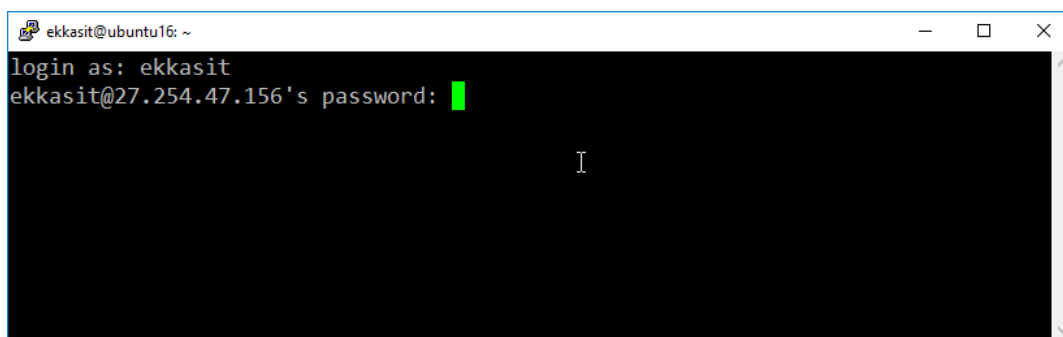
ภาพรวมการติดตั้ง

หนังสือเล่มนี้จะสาธิตวิธีการติดตั้งบน **Ubuntu** เวอร์ชัน **16.04** ซึ่งสาเหตุที่เลือกระบบนี้เพราะว่า Hosting ในประเทศไทยส่วนใหญ่ มักจะมี PHP + MySQL และ Linux ให้เราแล้ว และค่าใช้จ่ายของ Linux ส่วนใหญ่จะมีราคาถูกกว่า Hosting ที่เป็น Windows และอีกเหตุผลหนึ่งคือประสิทธิภาพที่ได้ค่อนข้างดี นั่นเป็นเพราะความเข้ากันได้ของ Apache + mod_php และ PHP7 ที่ทำงานบน Ubuntu นั่นเอง

หากผู้อ่านต้องการติดตั้งบน Windows ก็สามารถทำได้โดยใช้ **XAMPP** (<https://www.apachefriends.org>) โดยเข้าไปดาวน์โหลดโปรแกรมในเว็บดังกล่าว จากนั้นทำการติดตั้งโปรแกรม หลังจากติดตั้ง XAMPP เรียบร้อยแล้ว ให้ย้ายไฟล์เดอร์โปรเจกต์ของเรา ให้ไปไว้ยังไฟล์เดอร์ **htdocs** จากนั้นปรับแต่งค่าเพียงเล็กน้อยก็สามารถใช้งานได้ทันที รายละเอียดต่างๆ สามารถอ่านได้จากเว็บไซต์ หรือบทความที่มีเป็นจำนวนมาก

เริ่มการติดตั้ง

ในขั้นตอนแรกให้เราติดตั้งระบบปฏิบัติการ Ubuntu เวอร์ชัน 16.04 เสียก่อน หรือหากไม่อยากจะติดตั้งเอง ผู้อ่านจะต้องทำการสมัครใช้บริการ Web Hosting ที่มีให้บริการอยู่มากมาย เมื่อสมัครแล้ว ท่านจะได้รับ IP Address พร้อมกับ username และ password สำหรับล็อกอินเพื่อใช้บริการ จากนั้นให้เปิดโปรแกรม Terminal (หนังสือเล่มนี้ใช้ PuTTY) และทำการล็อกอินเข้าระบบ ดังนี้



ภาพที่ 14-1 การล็อกอินเข้าระบบด้วย PuTTY

1. เมื่อเข้าสู่ระบบได้แล้ว ให้ใช้คำสั่ง **sudo apt-get update** เพื่อเป็นการปรับปรุง repository ซึ่งคำสั่ง sudo จะเป็นคำสั่งเพื่อใช้สิทธิ์ของ root ดังนั้นจะมีการถาม password จากเรา

```
sudo apt-get update
```

2. จากนั้นทำการติดตั้ง php

```
sudo apt-get install php
```

3. หลังจากติดตั้งแล้วให้ใช้คำสั่ง **php -v** เพื่อตรวจสอบเวอร์ชัน

```
php -v
```

4. จากนั้นทำการติดตั้ง mbstring, xml, zip, curl และ gd ซึ่งเป็นสิ่งที่ Laravel จะต้องใช้ในการทำงาน โดยใช้คำสั่งดังต่อไปนี้

```
sudo apt-get install php7.0-mbstring
sudo apt-get install php-xml
sudo apt-get install php7.0-zip
sudo apt-get install curl
sudo apt-get install php7.0-gd
```

5. ถัดมาเราจะมาติดตั้ง **composer**

```
curl -sS https://getcomposer.org/installer | sudo php -- --install-dir=/usr/local/bin --
filename=composer
```

6. ทดสอบว่า composer ติดตั้งเรียบร้อยแล้วหรือไม่ โดยพิมพ์ composer ลงใน Terminal

```
composer
```

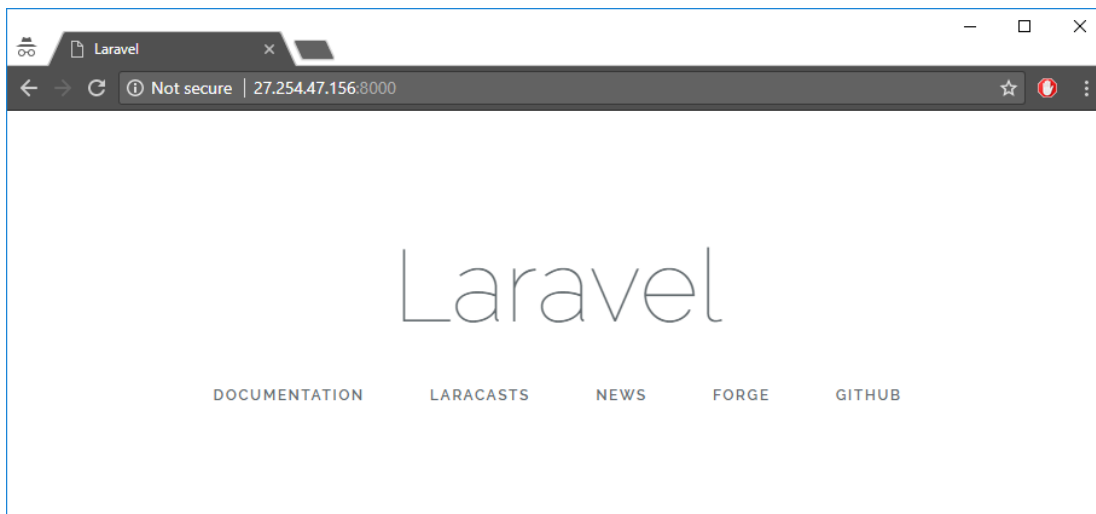
7. จากนั้นทดสอบสร้าง Laravel โปรเจกต์ชื่อว่า “laraveltest” เพื่อใช้ทดสอบว่าสิ่งต่างๆ ที่เราได้ทำมานั้นยังใช้งานได้ดียังหรือไม่ ด้วยการพิมพ์คำสั่ง composer เพื่อสร้างดังต่อไปนี้

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

8. ให้เข้าไปยังไดเรกทอรีของ laraveltest จากนั้นสั่งให้เว็บเซิร์ฟเวอร์ทำงาน โดยระบุ host และ port ดังต่อไปนี้

```
php artisan serve --host=0.0.0.0 --port=8000
```

9. ทดสอบโดยเปิดเบราว์เซอร์และเข้าไปที่ `http://<server IP address>:8000` จะเห็นว่า โปรเจกต์ของเราใช้งานได้เมื่อผ่านอินเทอร์เน็ตแล้ว



ภาพที่ 14-2 ทดสอบโปรเจกต์ของเราผ่านอินเทอร์เน็ต

10. จากนั้นติดตั้ง Git บน Ubuntu เพื่อให้รองรับการใช้คำสั่ง git สำหรับดึงโค้ดโปรแกรมมาติดตั้ง ซึ่งหากผู้อ่านได้ติดตั้งแล้ว (ทดสอบโดยพิมพ์คำสั่ง git บน Terminal) ก็ให้ข้ามขั้นตอนในข้อนี้ไปได้

```
sudo apt-get update  
sudo apt-get install git
```

11. จากนั้นให้เข้าไปที่ `/var/www` ซึ่งเราจะเก็บโปรเจกต์ไว้ที่นี่ และสร้างไดเรกทอรีชื่อ “bikeshop”

```
cd /var/www  
sudo mkdir bikeshop
```

12. จากนั้นกำหนดสิทธิ์ไดเรกทอรีนี้ให้กับผู้ใช้ที่ล็อกอินอยู่ ด้วยคำสั่ง `chown -R <user>:<group> bikeshop` ดังต่อไปนี้ (-R ย่อมาจาก recursive หมายถึงทุกๆ ไฟล์และไดเรกทอรีภายในทั้งหมดจะถูกกระทำด้วย)

```
sudo chown -R ekkasit:ekkasit bikeshop
```

13. จากนั้นเข้ามายังไดเรกทอรี bikeshop และเริ่มกำหนดค่าของ git เพื่อที่เราจะดึงโปรเจกต์ที่อยู่ใน Git server มาติดตั้งที่นี่

```
git init
git remote add origin https://github.com/ekkazit/laravel-bikeshop.git
git pull origin master
```

14. จะเห็นว่าโปรเจกต์ bikeshop ของเราถูกดึงมาจาก Git และมาติดตั้งให้เราเรียบร้อยแล้ว ให้เราสร้างไฟล์ .env และ copy .env ต้นฉบับของเรามาใส่ดังนี้

```
sudo nano .env
```

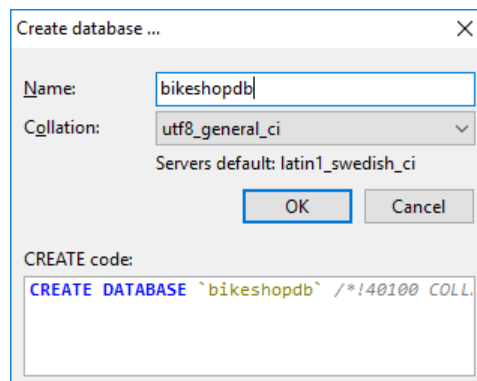
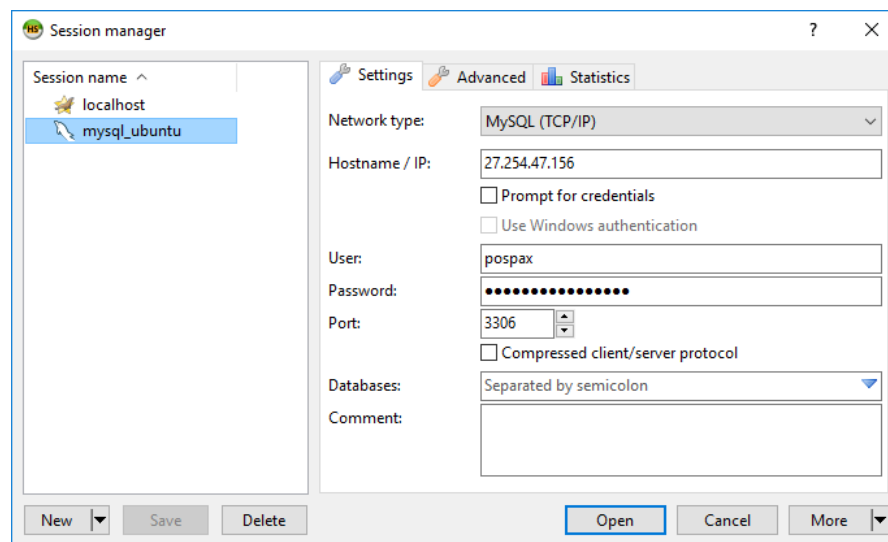
```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:2FjpZMflW46y78AJ9IUu1V3TFv/x/fv+vCoDYHpat0=
APP_DEBUG=true
APP_LOG_LEVEL=debug
APP_URL=http://localhost

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=bikeshopdb
DB_USERNAME=<mysql>
DB_PASSWORD=<mysql>
```

15. จากนั้นติดตั้ง MySQL บน Ubuntu โดยใช้คำสั่งต่อไปนี้ (กรณีที่ผู้อ่านติดตั้งแล้วให้ข้ามขั้นตอนนี้ไปได้)


```
sudo apt-get update  
sudo apt-get install mysql-server  
mysql_secure_installation
```

16. จากนั้นใช้คำสั่ง MySQL เพื่อสร้าง username / password และสิทธิ์ต่างๆ เมื่อเสร็จเรียบร้อยแล้ว ให้ทำการล็อกอินเข้าไปด้วย user ดังกล่าว และสร้างฐานข้อมูลชื่อ “bikeshopdb” ดังต่อไปนี้



ภาพที่ 14-3 การสร้างฐานข้อมูล MySQL ด้วย GUI

17. กลับมายังไฟล์ **.env** ของเรา ให้แก้ไขการเชื่อมต่อตาม MySQL ที่เราสร้างไว้ดังต่อไปนี้

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=bikeshopdb
DB_USERNAME=<mysql user>
DB_PASSWORD=<mysql password>
```

18. จากนั้นใช้ **composer update** เพื่อติดตั้งไลบรารีที่กำหนดไว้ใน composer.json ของ Laravel

```
composer update
```

19. หากติดตั้งไม่สำเร็จเนื่องจากไลบรารีบางตัวต้องการใช้ PHP เวอร์ชันที่สูงกว่า เราจะต้องทำการติดตั้ง PHP ใหม่อีกครั้งในเวอร์ชันที่สูงขึ้นไป (เดิม PHP 7.0 ให้เปลี่ยนเป็น PHP 7.1)

```
sudo add-apt-repository ppa:ondrej/php
sudo apt-get update
sudo apt-get install php7.1 php7.1-common
sudo apt-get install php7.1-curl php7.1-xml php7.1-zip php7.1-gd php7.1-mysql php7.1-
mbstring
```

```
php -v
```

```
ekkasit@ubuntu16: /var/www/bikeshop
ekkasit@ubuntu16:/var/www/bikeshop$ php -v
PHP 7.1.14-1+ubuntu16.04.1+deb.sury.org+1 (cli) (built: Feb  9 2018 09:33:27) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.1.0, Copyright (c) 1998-2018 Zend Technologies
    with Zend OPcache v7.1.14-1+ubuntu16.04.1+deb.sury.org+1, Copyright (c) 1999-2018, by Zend Technologies
ekkasit@ubuntu16:/var/www/bikeshop$
```

ภาพที่ 14-4 การทดสอบเวอร์ชัน PHP (7.1)

20. จากนั้นใช้คำสั่ง **composer update** อีกครั้งก็สามารถที่จะติดตั้งได้

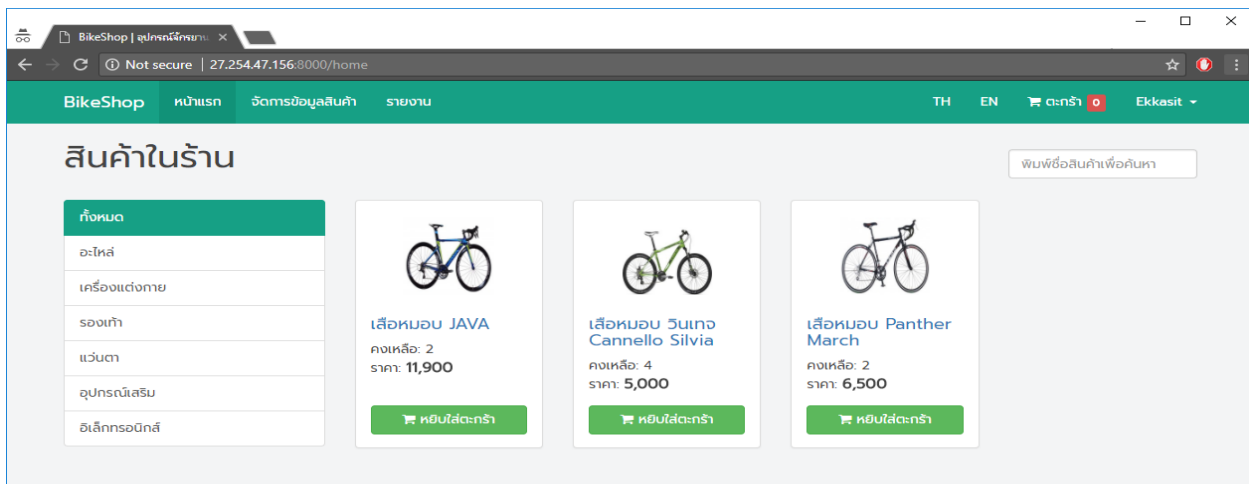
```
composer update
```

21. ถัดมาเราจะทำการ migrate ตารางต่างๆ และข้อมูลที่กำหนดไว้ใน seed ดังต่อไปนี้

```
php artisan migrate
php artisan db:seed
```

22. ทดสอบโดยการให้คำสั่ง **php artisan serve** อีกครั้ง จะเห็นว่าโปรเจกต์ของเราสามารถทำงานผ่านอินเทอร์เน็ตได้ มีข้อมูลต่างๆ ครบถ้วน ถึงตรงนี้การติดตั้งของเราใกล้ที่จะเสร็จแล้ว (ประมาณ 80%)

```
php artisan serve --host=0.0.0.0 --port=8000
```



ภาพที่ 14-5 ทดสอบโปรเจกต์ bikeshop สามารถใช้งานผ่านอินเทอร์เน็ตได้

23. ในข้อที่แล้วเซิร์ฟเวอร์ที่ใช้งานยังเป็น Built-in Server อยู่ เราจะต้องเปลี่ยนมาใช้ Apache เนื่องจากความสามารถในการทำงานบนระบบ Production การรับผู้ใช้เป็นจำนวนมาก และเป็น server ตัวจริงที่เราไม่ต้องพิมพ์ port ลงไปในยูอาร์แอลแล้ว เริ่มต้นให้เปิดไฟล์ `/etc/apache2/sites-available/000-default.conf`

```
nano /etc/apache2/sites-available/000-default.conf
```

24. จากนั้นเพิ่ม DocumentRoot และ Directory ลงในไฟล์ดังกล่าว (โดยให้คอมเมนต์ DocumentRoot อันเดิม ไม่ควรลบทิ้ง)

```
DocumentRoot /var/www/bikeshop/public
```

```
<Directory "/var/www/bikeshop">
```

```
    AllowOverride All
```

```
</Directory>
```

25. กำหนดสิทธิ์ไดเรกทอรีให้ user ของ Apache ได้มีสิทธิ์ใช้งาน (ยูสเซอร์ชื่อว่า **www-data**) เพื่อที่จะสามารถเขียนไฟล์และอัปโหลดรูปภาพได้

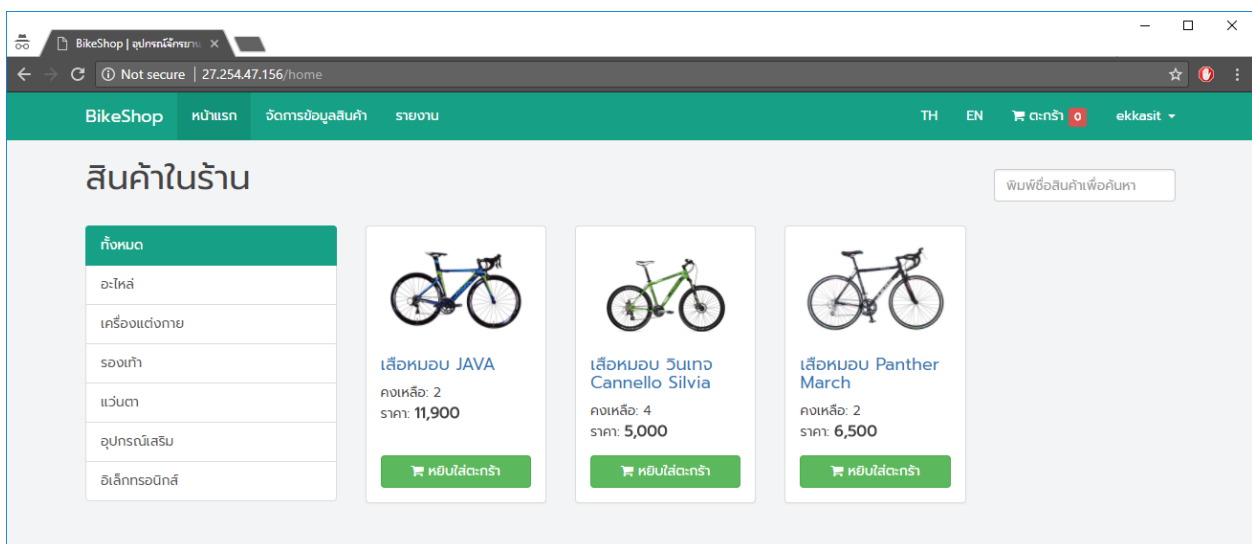
```
sudo chown -R www-data:www-data /var/www/bikeshop/  
sudo chmod -R 755 /var/www/bikeshop/storage
```

26. จากนั้นใช้คำสั่ง **disstite** เพื่อถอดเว็บออกไปจากระบบ แล้วใช้ **ensite** เพื่อนำระบบใส่เข้าไปในเว็บไซต์ใหม่ให้ Apache รู้จัก

```
sudo a2disstite 000-default.conf  
sudo a2ensite 000-default.conf
```

27. ทำการ reload ตัว service ของ Apache อีกครั้ง และให้ลองเข้าระบบด้วย IP Address โดยไม่ต้องพิมพ์พอร์ต ก็จะทำให้ระบบของเราพร้อมใช้งานได้เรียบร้อยแล้ว เป็นอันจบขั้นตอนของการติดตั้งลงใน Production เพื่อใช้งานจริง หากคุณจดโดเมนก็ให้ทำการแมพ DNS เข้ากับ IP ดังกล่าว ก็จะสามารถเข้าใช้งานด้วยชื่อโดเมนได้

```
sudo service apache2 restart
```



ภาพที่ 14-6 โปรเจกต์ของเราติดตั้งเสร็จสมบูรณ์แล้ว

สรุปท้ายบท

ในบทนี้ผู้อ่านจะได้เรียนรู้การติดตั้งโปรเจกต์ของเราลงในเซิร์ฟเวอร์ Ubuntu โดยละเอียด ผู้อ่านยังได้เรียนรู้การใช้งานคำสั่งลินุกซ์เบื้องต้น ในกรณีที่ต้องการติดตั้งระบบใช้งานด้วยตนเอง สำหรับเนื้อหาต่างๆ ของหนังสือเล่มนี้ก็ขอจบลง在此 หากผู้อ่านมีข้อสงสัย สามารถติดต่อผู้เขียนได้โดยตรงทางเว็บไซต์ www.codeexcellent.com หรือสามารถติดต่อทาง LINE ที่ได้ระบุไว้ ผู้เขียนยินดีที่จะช่วยให้ผู้อ่านทุกท่าน สามารถที่จะนำ Laravel ไปใช้งาน และได้ผลงานออกมาตามสิ่งที่มุ่งหวังครับ

เกี่ยวกับผู้เขียน

นายเอกสิทธิ์ ศรีสุขะ

การศึกษา

- วิศวกรรมศาสตรมหาบัณฑิต (วศ.ม.), วิศวกรรมคอมพิวเตอร์, มหาวิทยาลัยเกษตรศาสตร์
- อุตสาหกรรมศาสตรบัณฑิต (อส.บ.), เทคโนโลยีสารสนเทศเพื่ออุตสาหกรรม, มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

การทำงาน

- **Software Engineer**, บริษัท FICO จำกัด: ให้บริการด้านการผลิตซอฟต์แวร์ระบบ Commerce Marketing Platform
- **Technical Consultant**, บริษัท Comm-IT จำกัด: ให้บริการด้าน Asset Management Software (Maximo) สำหรับโรงงานอุตสาหกรรม
- **Technical Consultant**, บริษัท Convergence Team จำกัด: ให้บริการด้าน Asset Management Software (Maximo) สำหรับโรงงานอุตสาหกรรม
- **System Analyst**, บริษัท Insource Asia จำกัด: ให้บริการด้านโซลูชันด้านการจองโรงแรม จองที่พัก ทั้งในประเทศไทยและต่างประเทศ
- **System Analyst**, บริษัท Toptier Solutions จำกัด: ให้บริการด้านซอฟต์แวร์สำหรับองค์กร, ซอฟต์แวร์กระบวนการผลิตในงานอุตสาหกรรม, และเออาร์ชอร์ส
- **Software Developer**, บริษัท พันธวิษ จำกัด: ให้บริการด้าน E-Procurement / E-Auction ภาครัฐและภาคเอกชน รวมไปถึงการขายสินค้าออนไลน์

ผลงาน

- Code Excellent ผู้ก่อตั้งเว็บไซต์ ให้บริการด้านฝึกอบรมทั้งภายในและนอกองค์กร
- PosPax ระบบ Point of Sale ที่ทำงานบนคลาวด์ สำหรับธุรกิจ retail ขนาดเล็ก
- Maximo Customization งาน customize ซอฟต์แวร์ Maximo ให้องค์กร อาทิเช่น Glow, Gulf, Eternal Rasin, Pimai Salt และ Bestlin
- PTVN Auction ระบบประมูลออนไลน์ ทั้งภาครัฐและเอกชน
- PTVN EBD ระบบ e-Procurement สำหรับองค์กร
- True Web Ordering ระบบ shopping cart ซื้อขายอุปกรณ์โทรศัพท์ระหว่างดีลเลอร์
- Thai Language Specific Web Crawler (Link) งานวิจัยเว็บไซต์สำหรับเก็บรวบรวมเว็บไซต์ภาษาไทย
- Extract Transform Load:ETL (ซอฟต์แวร์ภายใน) สำหรับย้ายและแปลงรูปแบบข้อมูลขนาดใหญ่ สำหรับธุรกิจประกันภัย