S1

Laravel เป็น PHP Framework ทำให้สามารถเขียนโค๊ดที่ดูสะอาดตาและสามารถอ่านและทำความเข้าใจได้โดยง่าย ยังสามารถดาวน์โหลดมาใช้งานได้ฟรี ออกแบบมาเพื่อพัฒนาเว็บแอพพลิเคชั่นในรูปแบบ MVC พัฒนาโดยมีผู้นำทีมคือนาย Taylor Otwell ภายใต้ลิขสิทธิ์ของ MIT และ source code ได้ถูกเก็บไว้บน host ของ Github

Sเทคโนโลยีที่ทำการศึกษา

เทคโนโลยีเว็บ

มี 4 ยุค 1.0 2.0 3.0 4.0

เว็บ 1.0 เป็นเว็บในยุคเริ่มต้นและยังคงมีให้เห็นอยู่ในปัจจุบัน มักมีรูปแบบของไฟล์เป็นนามสกุลเป็น .htm

เว็บ 2.0 เป็นเว็บในปัจจุบันที่มีการใช้อินเตอร์เน็ตเพื่อเขียนบล็อก (Blog), แชร์รูปภาพ, ร่วมเขียนวีกี (Wiki), แสดงความคิดเห็น (Post Comment)

เว็บ 3.0 เป็นเว็บในยุคอนาคตอันใกล้ คือ เว็บที่มีการพัฒนาการต่อจากเว็บ 2.0 ความแตกต่างคือสร้างความฉลาดมากกว่า

นอกจากการกล่าวถึง Web 3.0 แล้ว ยังมีการคาดการณ์เทคโนโลยีเว็บไปถึง WEB 4.0 ซึ่งมีการเรียกกันว่า “A Symbiotic web” คือ เว็บที่ทำงานแบบ Artificial Intelligence (AI) ที่ฉลาดมากยิ่งขึ้น

**Visual Studio**

* Visual Studio เป็น โปรแกรมตัวหนึ่งที่เป็นเครื่องมือที่ช่วยพัฒนาซอฟต์แวร์และระบบต่างๆ ซึ่งสามารถติดต่อสื่อสารพูดคุยกับคอมพิวเตอร์ได้ในระดับหนึ่งแล้ว แต่ยังไม่สามารถพัฒนาเป็นระบบเองได้ ซึ่งนักโปรแกรมเมอร์จะนำเครื่องมือมาใช้ในการพัฒนาต่อยอดให้เกิดเป็นระบบต่างๆ หรือเป็นเว็บไซต์ และแอพพลิเคชั่นต่างๆ
* **Laravel**

Laravel เป็น PHP Framework ประเภทหนึ่ง ที่มีความสามารถในการเขียน PHP ในลักษณะของ Model-View-Controller(MVC) ในปัจจุบันนี้ PHP Framework Laravel framework ได้รับความนิยมมากเป็นอันดับที่ 1 เนื่องจากความง่าย ไม่ซับซ้อนเกินไป ทำงานได้รวดเร็ว และ มีฟังก์ชันการทำงานต่าง ๆ ที่ Framework จำเป็นต้องมีครบครัน ทำให้พัฒนาระบบได้สะดวก รวดเร็วมากยิ่งขึ้น

จุดเด่น

* แนวคิดการพัฒนาแอพพลิเคชั่นที่แพร่หลายในปัจจุบัน
* ส่วนขยายของ Laravel ที่ชื่อว่า Bundle ซึ่งช่วยให้ประหยัดเวลาในการเขียน Code ลงเป็นอย่างมากโดยใช้คำสั่งผ่าน Command Line ในการติดตั้งผ่านคำสั่ง “php artisan” แทน
* การเรียกใช้งานคลาสต่าง ๆ ที่ง่ายขึ้นเพราะ Laravel เรียกใช้งานคลาสโดย Name Space โดยคำสั่งที่สั้นและเข้าใจง่าย
* Unit testing สามารถสร้าง Unit test ขึ้นมาเพื่อทดสอบงานของตัวเองได้ โดยสร้างผ่านชุดคำสั่ง “artisan” สามารถดูคำสั่งได้จาก Testing: Getting Started
* Eloquent ORM ชุดคำสั่งหรือเครื่องมือในการ Query ข้อมูลต่าง ๆ ในฐานข้อมูลสามารถดูคำสั่งได้จาก DatabasesQuery
* Routing สามารถกำหนดชื่อของ Url เพื่อชี้ไปยังส่วนต่าง ๆ เช่น View หรือ Controller ตามที่ต้องการได้คำสั่งอ่านและเข้าใจง่ายมากสามารถดูตัวอย่างการใช้งานได้จาก Routing
* Restful Controller สามารถกรองชนิดการส่งคำร้องขอจากฟอร์มทั้งแบบ Post, Get, Put/Patch, Delete
* View Composer ส่วนของ Code HTML ที่นำมาเรียงติดต่อกัน และจะทำงาน หลังจากประกอบกันเสร็จเรียบร้อยแล้ว เช่นเราแบ่งส่วน header, container, footer และนำมาเรียกใช้ต่อกันภายหลังเป็นต้น

**สรุป**

จากที่กล่าวมาข้างต้นก็พอจะมองเห็นว่าเจ้าตัว Laravel Framework ก็เป็น Framework ทางเลือกหนึ่งสำหรับมือใหม่ เพราะสามารถขึ้นโครงสร้างได้ง่าย สุดท้ายนี้การเลือกใช้ Framework นั้นก็ขึ้นอยู่กับตัวบุคคล องค์กร หรือความเหมาะสมของงานนั่นเอง

รูปMVC

**Route**

จะเห็นได้ว่า Route จะทำหน้าที่เป็นอย่างแรก ในการกำหนดทิศทางว่า จะสั่งให้ระบบทำงานในส่วนไหน เพราะฉะนั้นสิ่งแรกที่เราควรจะต้องทำความเข้าใจเพื่อที่จะศึกษา Laravel Framework ก็คือส่วนของ Route เพราะ Route จะเป็นตัวกำหนดว่า จะสั่งให้ระบบไปทำการ Process ที่ Controller หรือว่า จะเรียกใช้งาน Model แล้วนำข้อมูลมา Process แล้วส่งไปที่ View เลย

**Controller**

ส่วนต่อไปที่เราควรจะศึกษาทำความเข้าใจก็คือ Controller ใน Laravel Framework นั้น Controller จะทำหน้าที่เรียกใช้ข้อมูลต่าง ๆ เพื่อนำมาประมวลผล และส่งค่าไปแสดงผลในส่วนของ View จากภาพจะเห็นได้ว่า Route จะทำหน้าที่ในการสั่ง ให้ Controller ทำงาน และ Controller จะทำการเรียกใช้ Model เพื่อทำการ Query ข้อมูลมาใช้งาน เมื่อได้รับข้อมูล และประมวลผลเรียบร้อยแล้ว Controller จะส่งค่าไปในส่วนของ View เพื่อทำการแสดงผลลัพธ์จากการ Process ออกมา

**Model**

ในส่วนของ Model ใน Laravel Framework จะอยู่ในหมวด Eloquent ORM คือการนำ Table มาเรียกใช้งานในรูปแบบของ Object เพื่อที่จะให้เราสามารถใช้งานได้ง่ายขึ้น โดยจะรวมรวมคำสั่งInsert,Update,Delete ให้เราสามารถเรียกใช้อย่างง่ายได้ รวมถึงการ Query ข้อมูลต่าง ๆ ด้วย

**view**

ในส่วนของ View นั้น คือส่วนที่ทำหน้าที่ไว้แสดงผลข้อมูลต่าง ๆ ที่ส่งมาจาก Controller หรือModel ใน Laravel Framework เราสามารถที่จะทำการขอเรียกใช้งานข้อมูลจาก Model

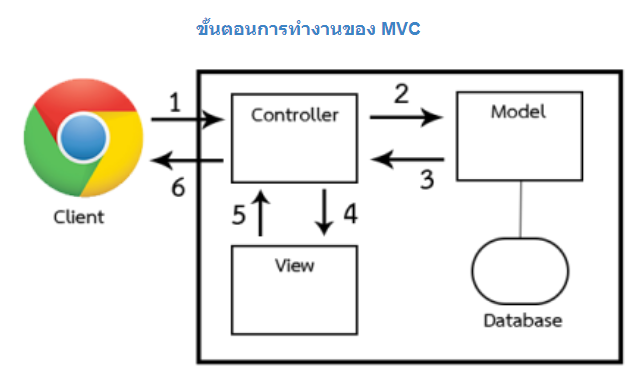
เครื่องมือที่ใช้

Atom

อีคลิป

สับลาม

วิชั่วสตูดิโอ โค๊ด



1. เริ่มจาก Client ส่ง Request ไปที่ Web App ซึ่งจะถูกส่งต่อให้ Controller ทำการตรวจสอบข้อมูลที่มาให้ (Request Method, Request Parameters)

2. แล้ว Controller จะเรียก Method ให้ทำงานเพื่อจัดการ Request นั้น

3. Model จะทำการคำนวณและอาจติดต่อกับ Database เพื่อจัดการกับ Request นั้น แล้วส่งผลลัพธ์กลับไปที่ Controller

4. เมื่อ Controller ได้ผลลัพธ์จาก Model แล้วก็ใช้ผลลัพธ์นั้นส่งต่อให้ View ทำงาน

5. View จะสร้าง Page สำหรับแสดงผลลัพธ์นั้น แล้วส่ง page กลับไปที่ Controller

6. Controller ส่ง Page นั้น (เป็น Response) กลับไปยัง Client

OOP

แปลตรงๆ ก็คือการเขียนโปรแกรมเชิงวัตถุ เป็นกระบวนทัศน์หรือรูปแบบหนึ่งของการเขียนโปรแกรม โดยจะมองทุกอย่างในโปรแกรมหรือแอพพลิเคชันเป็น “วัตถุ” ซึ่งก็เหมือนวัตถุในชีวิตจริงของเรา อย่างเช่น สุนัก, แมว, โทรทัศน์, จักรยาน , โต๊ะ หรือ เก้าอี้

โดย**วัตถุ** (Object) จะเก็บสถานะใน **ฟิล์ด** (field) (ตัวแปลในบางภาษา) และเปิดเผยพฤติกรรมผ่าน **เมธอด** (method) (ฟังก์ชันในบางภาษา) โดยที่ method จะดำเนินการกับสถานะภายในของ object นั้น และเป็นกลไกหลักในการสื่อสารระหว่าง object กับ object ซึ่งจะมีการซ้อนสถานะภายใน และการติดต่อกับ object ทั้งหมดจำเป็นต้องกระทำผ่าน method ของ object นั้น

*ซึ่ง OOP จะมีหลักการที่เป็นเสาหลักอยู่ 4 ข้อ ดังนี้*

### Abstraction

เป็นหลักการนามธรรมของ object ซึ่งเป็นกระบวนการในการเอาส่วน concrete ของ class หรือส่วนที่มีการ implementation ของ object ออกมา โดยรักษาลักษณะร่วมกัน หรือคุณลักษณะให้กับ concreate object ต่างๆ ซึ่งจะเป็นการใช้หลักการอื่นๆ ของ OOP ในการทำให้เกิด abstaction โดยเฉพาะอย่างยิ่ง Polymorphism และในบางภาษาจะมี abstract/interface keyword มาให้เพื่อระบุ class ที่ต้องการให้มีคุณสมบัติ abstraction ตัวอย่างดังนี้

### Encapsulation

เป็นหลักการการห่อหุ้มสถานะหรือข้อมูลของ object จากภายนอกให้ไม่สามารถเข้าถึงสถานะของ object ได้ ดั้งนั้นการเข้าถึงจะต้องกระทำผ่านทาง method เท่านั้น (เรียกว่า **setter/getter**) ซึ่งการซ้อนข้อมูลก็มีหลายระดับ **private** จำกัดการเข้าถึงอย่างสิ้นเชิง, **protected** จำกัดการเข้าถึงจากภายนอกแต่ class ลูกสามารถเข้าถึงได้ หรือ **public** ไม่จำกัดดารเข้าถึงเลยภายนอกสามารถเข้าถึงได้ เป็นต้น บางภาษาอาจจะมีมากกว่านี้ ตัวอย่างเช่น

### Inheritance

เป็นหลักการสืบทอดคุณสมบัติของ object ซึ่ง class (**subclass**) ที่สืบทอดจาก class อื่น (**superclass**) จะได้รับสถานะและพฤติกรรมของ class นั้นทั้งหมด และยังสามารถเปลี่ยนพฤติกรรมของบาง method (**overriding**) ที่ได้สืบทอดมาได้ อีกทั้งสามารถขยายความสามารถเพิ่มเติมได้อีก ทำให้เราสามารถนำโค้ดที่มีอยู่แล้วมาใช้ใหม่ได้ (code reuse) ตัวอย่างเช่น

### Polymorphism

เป็นหลักการพ้องรูปหรือการมีหลายรูปของ object คล้ายๆ กับหลักการของชีววิทยาที่สิ่งมีชีวิตสามารถมีหลายรูปแบบ หรือหลายสายพันธุ์ (เช่น มนุษย์ มี มองโกลลอยด์, คอเคซอยด์ หรือนิกรอยด์ เป็นตั้น ซึ่งทั้งหมดก็เป็น มนุษย์ เหมือนกัน) โดย subclass ของ class สามารถกำหนดพฤติกรรมที่เป็นเอกลักษณ์ของตนเองได้ และยังแชร์บางส่วนของฟังก์ชันการทำงานเดียวกันกับ superclass ตัวอย่างเช่น