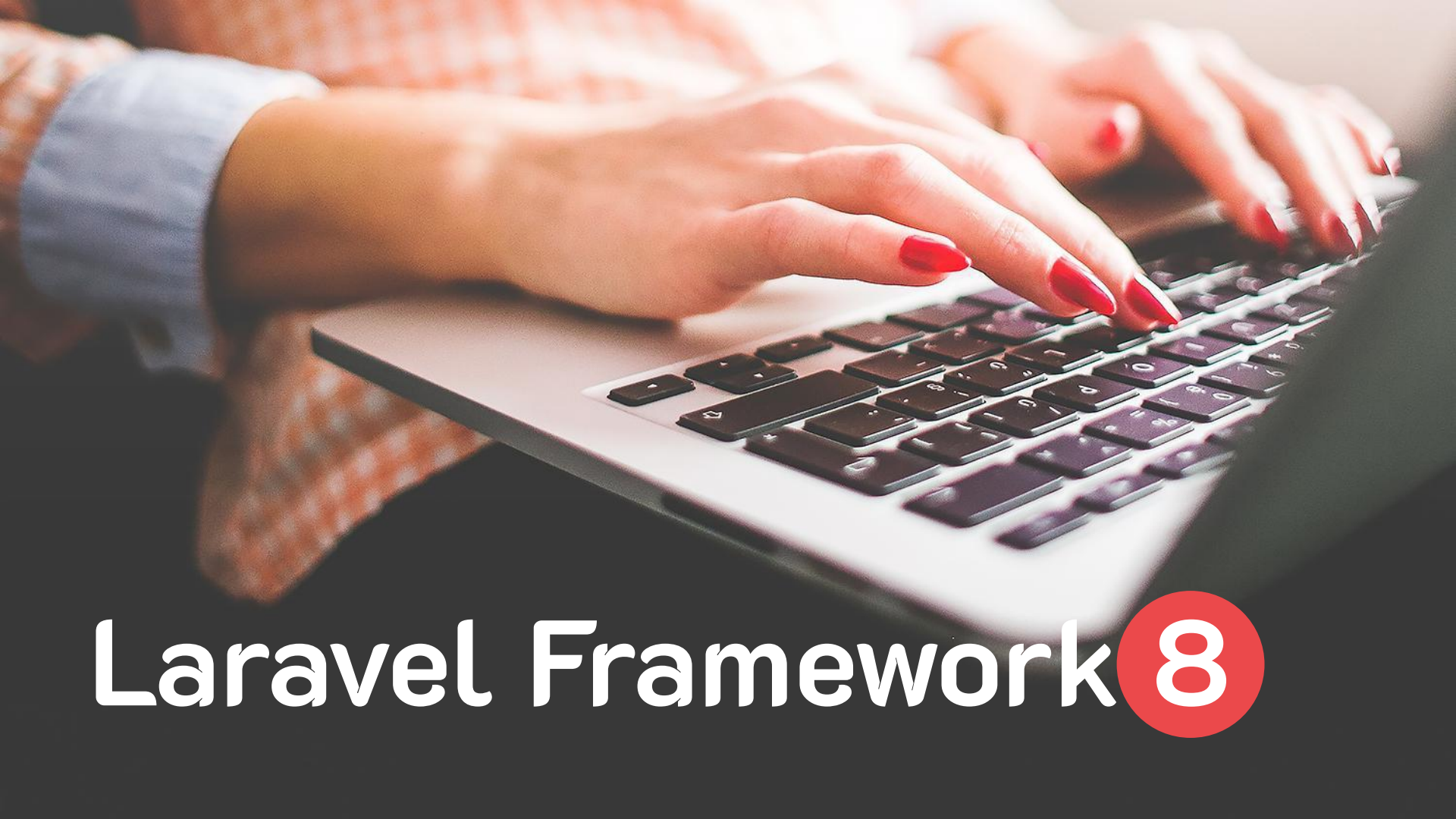

Laravel Framework 8

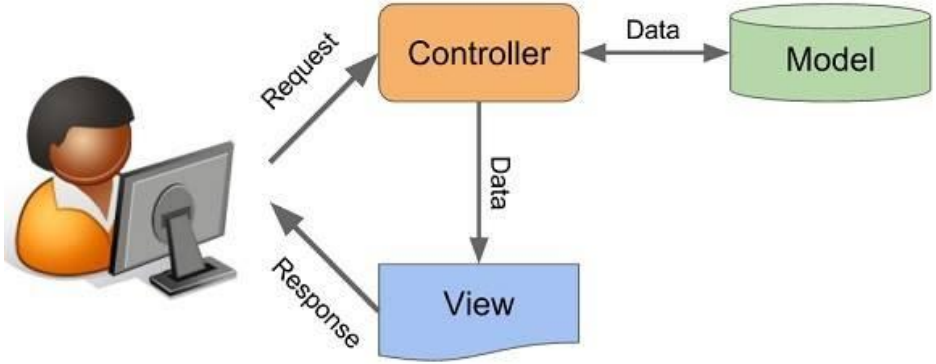
พื้นฐานที่ต้องมี

- PHP เบื้องต้น
- SQL เบื้องต้น

เครื่องมือที่ต้องติดตั้ง

- XAMPP
- Visual Studio Code
- Nodejs
- Composer

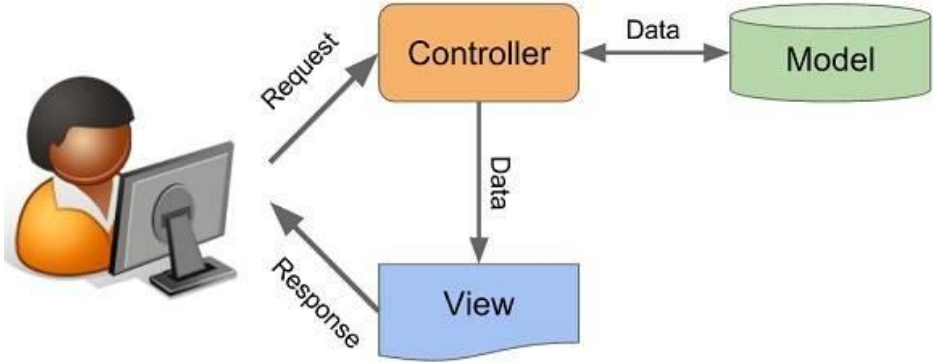
สถาปัตยกรรม MVC



Controller

- เป็นส่วนที่ทำงานเป็นอันดับแรกเมื่อมีโปรแกรมถูกเรียกจาก Web Browser
- เป็นส่วนที่ติดต่อการทำงานระหว่างผู้ใช้และโปรแกรม
- มีการติดต่อกับ Database(ฐานข้อมูล) ด้วย Model และแสดงผลข้อมูลผ่านทาง View
- เป็นส่วนที่มีการประมวลผลหลักของโปรแกรม

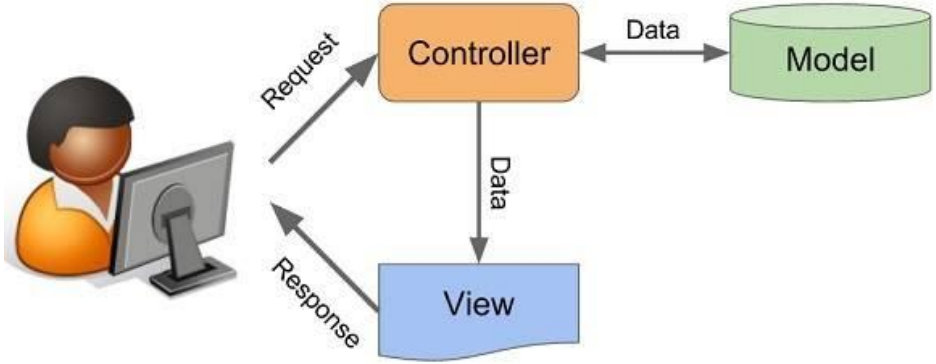
สถาปัตยกรรม MVC



Model

- ดูแลในเรื่องของการติดต่อสื่อสารระหว่าง Object และ Database โดยที่ผู้พัฒนาไม่ต้องยุ่งยากกับการใช้ SQL command
- Handles validation(ตรวจสอบความถูกต้อง), association(ความสัมพันธ์ของข้อมูล), transactions และอื่นๆ

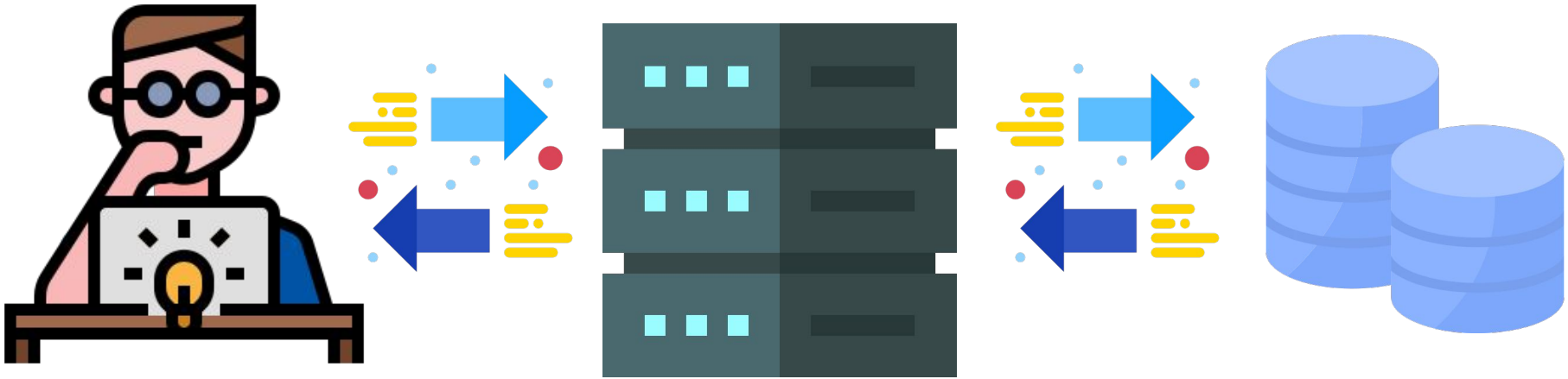
สถาปัตยกรรม MVC



View

- เป็นส่วนที่ต้องแสดงผลผ่าน Web Browserเขียนด้วยพื้นฐานของ HTML แทรกด้วย Script PHP หรือ JavaScript ตาม Syntax ของแต่ละภาษาที่ใช้
- การทำงานสัมพันธ์อยู่กับ Controller
- View เป็นการแสดงผลนำข้อมูลจาก Controller มาแสดงผลสามารถกำหนด Style Sheet และ Template เพื่อให้งานเว็บแอพลิเคชั่นนั้นมีมาตรฐานเดียวกันทั้งหมดได้

สถาปัตยกรรม MVC

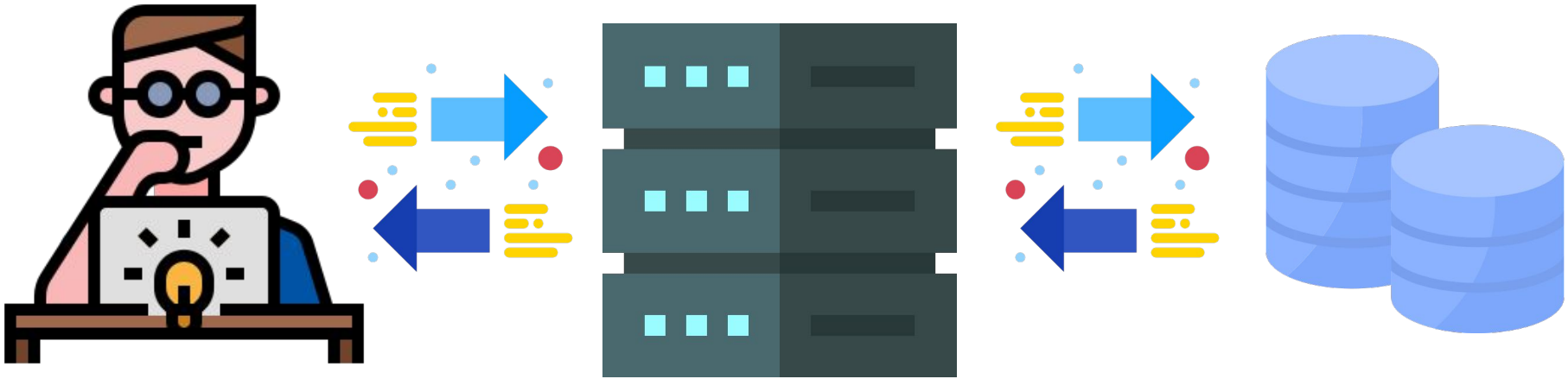


User (Browser)

Server

Database

สถาปัตยกรรม MVC



View Controller Model

หน้าที่ของโฟลเดอร์และไฟล์โปรเจค

- **app** เป็นโฟลเดอร์ที่ใช้เก็บ
  ไฟล์จำพวก **Model** หรือ **Controller** ที่ใช้ในการประมวลผลและติดต่อกับฐานข้อมูล

- **database** เป็นโฟลเดอร์ที่ใช้เก็บ
  ไฟล์จำพวก **Migrations** และ **Seeding** เพื่อใช้ในการสร้าง **Table** หรือใส่ข้อมูลในฐานข้อมูล

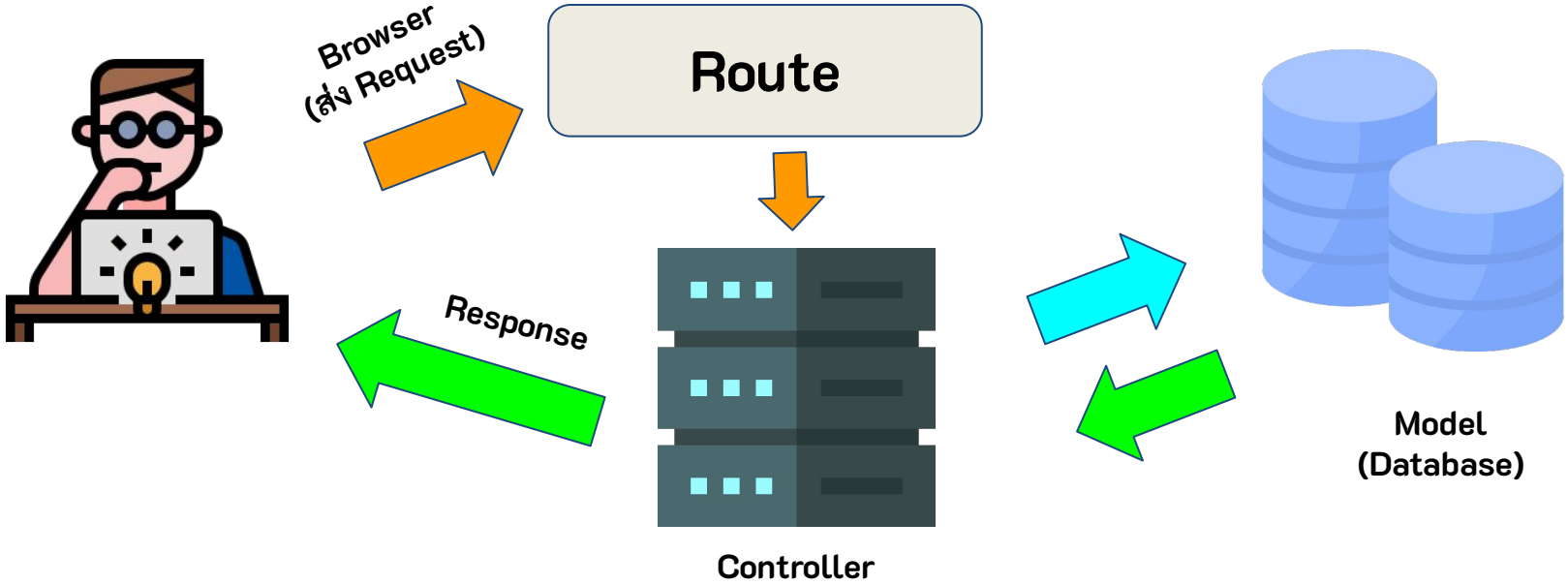
- **public** ใช้เก็บพวก Javascript, CSS รวมไปถึง Assets ,
  File index และ .htaccess ทุกส่วนสามารถเข้าถึงได้เป็นตัวจัดการไฟล์แบบ

หน้าที่ของโฟลเดอร์และไฟล์โปรเจค

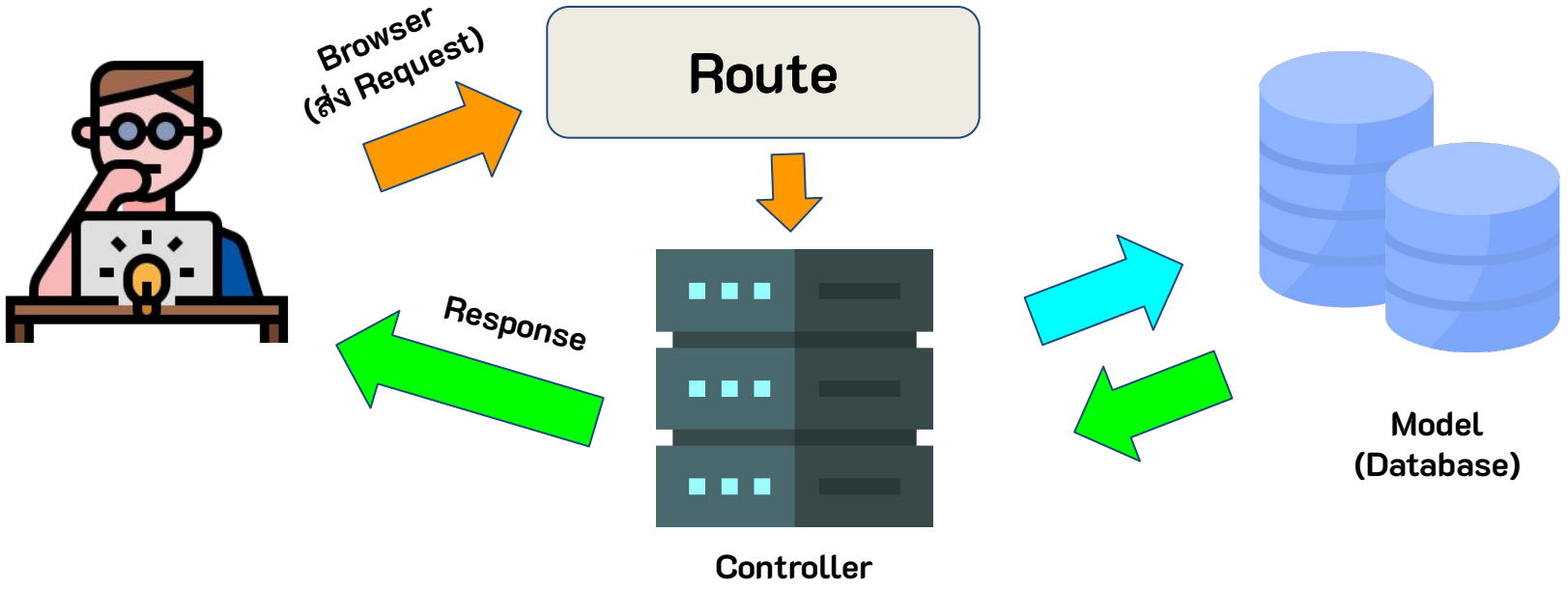
- **resources** ใช้เก็บโฟลเดอร์ที่ใช้ในส่วนของการแสดงผลต่าง ๆ **(Views)**
- **routes** เป็นส่วนที่ใช้เก็บไฟล์ในการกำหนดเส้นทาง **(Url)** ของ **Web**
- **storage** คลังพื้นที่จัดเก็บข้อมูล **Session, caches** , รูปภาพหรือไฟล์ที่ถูกทาง **blade engine** ทำการ **compiled** มาแล้ว
- **tests** เป็นส่วนที่ใช้จัดการพวก **automated tests** เช่น **unit test**
- ไฟล์ **.env** เป็นไฟล์ที่ใช้ **config laravel** กับฐานข้อมูล

Routing

การกำหนดเส้นทางหรือ URL ในการอนุญาตให้เข้าถึงข้อมูลรวมไปถึงตรวจสอบ URL Request เพื่อจะได้กำหนดรูปแบบการทำงาน เช่น ทำงานอะไรที่ Controller ไหนและเกิด Action อะไรบ้าง



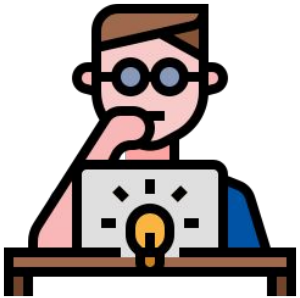
Controller


Browser
(ส่ง Request)
Route
Response
Controller
Model
(Database)

การนิยาม Route

routes/web.php

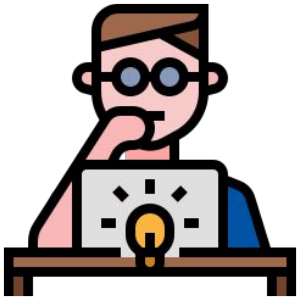
```
Route::get('path',function(){
    คำสั่งต่างๆ
})
```



GET / POST

วิธีการส่งข้อมูล

- **GET** ส่งค่าผ่านทาง URL ทำให้ผู้ใช้งานสามารถมองเห็นข้อมูลที่ส่งไปได้
- **POST** ส่งค่าผู้ใช้งานไม่สามารถมองเห็นข้อมูลที่ส่งไปได้



Dynamic Route

ส่ง Parameter ไปพร้อมกับ Route

```
Route::get('/users/{name}',function($name){
        echo "<h1>Hello $name</h1>";
});
```

HTTP Status Code

เป็นรหัสที่บ่งบอกสถานะของ Request ตัวอย่างเช่น

- **200 Ok** (ดำเนินการเสร็จสมบูรณ์)
- **201 Create (สร้างข้อมูลใหม่เรียบร้อย)**
- **400 Bad Request** (Server ไม่เข้าใจว่า Request นี้เกี่ยวกับอะไร)
- **404 Not Found** (หาข้อมูลที่เรียกไม่เจอหรือไม่สามารถใช้งานได้)
- **500 Internal Server Error** (Request ถูกต้องแต่มีข้อผิดพลาดที่ฝั่ง Server)

การสร้าง View

resource/views/ชื่อview.blade.php

```
Route::get('url',function(){
        return view('ชื่อview');
})
```

Blade Template

Template คือหน้าตา Application เป็นส่วนที่ไว้ใช้แสดงผลข้อมูล ผลลัพธ์จากการประมวลผลใน View มาแสดงผลในหน้าเว็บร่วมกับ HTML

Blade Template คือชุดคำสั่งมาตรฐานในการแสดงผลและประมวลผลข้อมูลใน Laravel Framework สามารถแทรกชุดคำสั่งของภาษา PHP ได้

Blade Syntax

แสดงผลข้อมูลใน PHP ธรรมดา	แสดงผลข้อมูลใน Blade
<? php echo $ชื่อตัวแปร ?>	{{ $ชื่อตัวแปร }}

กำหนดเงื่อนไขใน PHP ธรรมดา	กำหนดเงื่อนไขใน Blade
if(condition){	@if(condition)
}else{	@else
}	@endif

Blade Syntax

foreach ใน PHP ธรรมดา	foreach ใน Blade
foreach(){	@foeach()
}	@endforeach

Laravel JetStream

คือ ไลบรารี่ที่ให้บริการด้าน UI และจัดการเกี่ยวกับระบบ Login และยืนยันตัวตนผู้ใช้ (Authentication) ออกแบบด้วย Tailwind CSS

- ระบบโปรไฟล์และเปลี่ยนรหัสผ่าน
- ยืนยันตัวตนผ่านอีเมล (Email Verification)
- เพิ่มความปลอดภัยด้วยการยืนยันตัวตน 2 ชั้นด้วยระบบ Two-Factor Authentication
- มีการ Generate รหัสเฉพาะสำหรับการกู้คืนบัญชีผู้ใช้
- Browser Session บอกสถานะการ Login ว่า Login กี่เครื่อง

LiveWire

การใช้งาน JetStream ที่ออกแบบด้วย Tailwind CSS สามารถเลือกใช้งาน Package 2 ตัว คือ LiveWire , Inertia มาจัดการส่วนของ Frontend

LiveWire เป็น Full Stack Framework ของ Laravel Framework ใช้ในการจัดการส่วนของ User Interface (UI) เป็นรูปแบบ Dynamic Front-end ใช้ร่วมกับ Blade ได้