

พัฒนาเว็บแอปพลิเคชันด้วย  
Laravel Framework 10.x

# ต้องมีพื้นฐานอะไรบ้าง



<https://www.youtube.com/c/KongRuksiamTutorial>



<https://www.facebook.com/KongRuksiamTutorial/>

# ต้องมีพื้นฐานอะไรบ้าง



- มีพื้นฐานการเขียนโปรแกรมด้วยภาษา PHP
- มีพื้นฐานการเขียนโปรแกรมเชิงวัตถุด้วยภาษา PHP  
(Class , Method , Object)
- มีพื้นฐาน SQL สำหรับจัดการฐานข้อมูล
- มีพื้นฐาน HTML5 , CSS3 , JavaScript , Bootstrap 5  
สำหรับนำมาสร้างเว็บแอปพลิเคชัน

เครื่องมือพื้นฐาน

# เครื่องมือพื้นฐาน

- XAMPP
- Composer
- Nodejs
- Visual Studio Code



# รู้จักกับ MVC (Model-View-Controller)



<https://www.youtube.com/c/KongRuksiamTutorial>

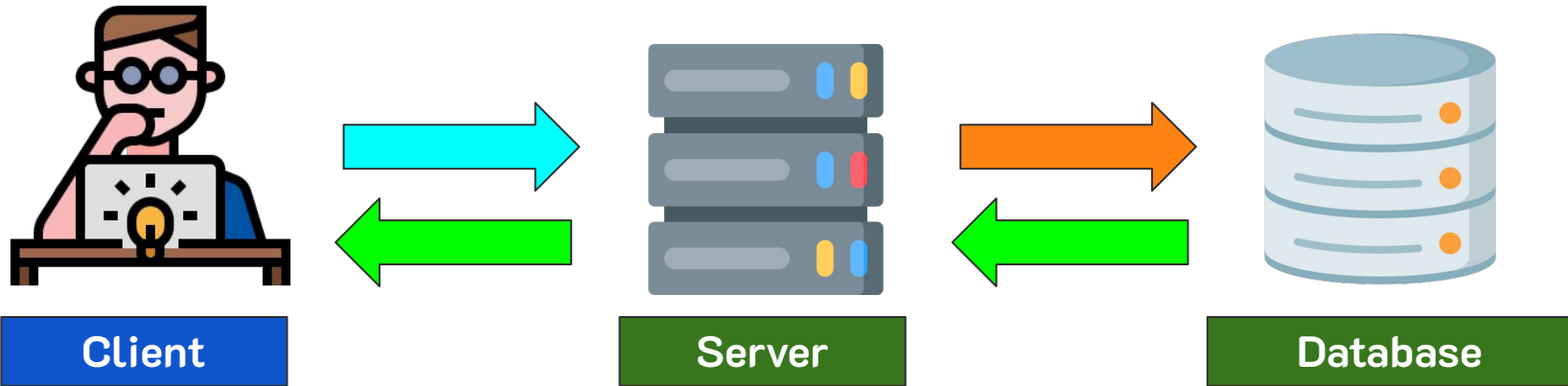


<https://www.facebook.com/KongRuksiamTutorial/>

# รู้จักกับ MVC

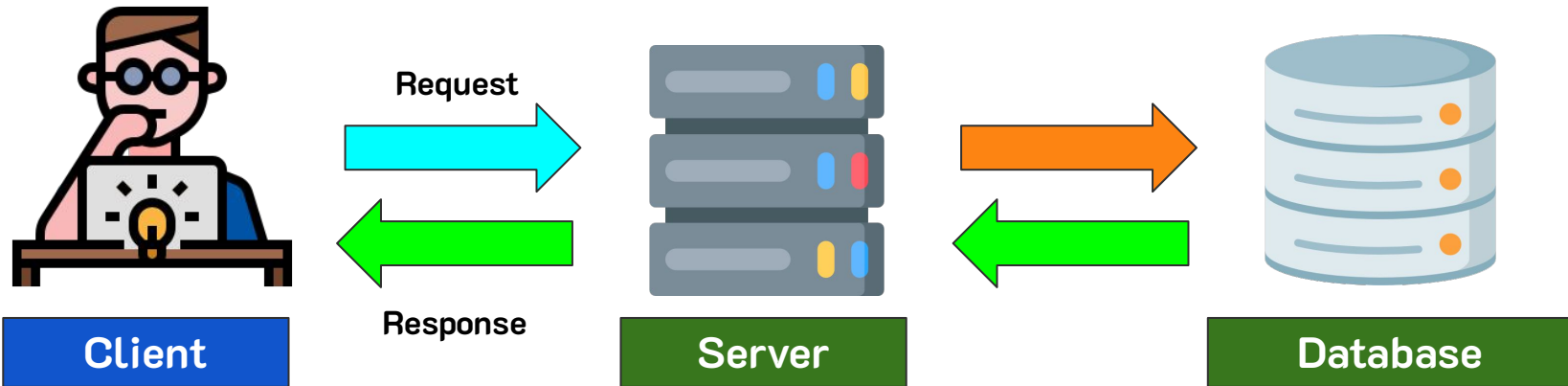
- **Model (M)** คือ ส่วนที่เก็บข้อมูลของ Application (Back-End)
- **View (V)** คือ ส่วนที่ทำงานฝั่ง Front-End หรือหน้าต่างแอปพลิเคชัน เป็นส่วนที่ไว้ใช้แสดงผลข้อมูลด้วย HTML (Blade Template)
- **Controller (C)** คือ ส่วนประมวลผลคำสั่งต่างๆ ในแอปพลิเคชัน โดยควบคุมการทำงานระหว่าง Model และ View (Back-End)

# รู้จักกับ MVC

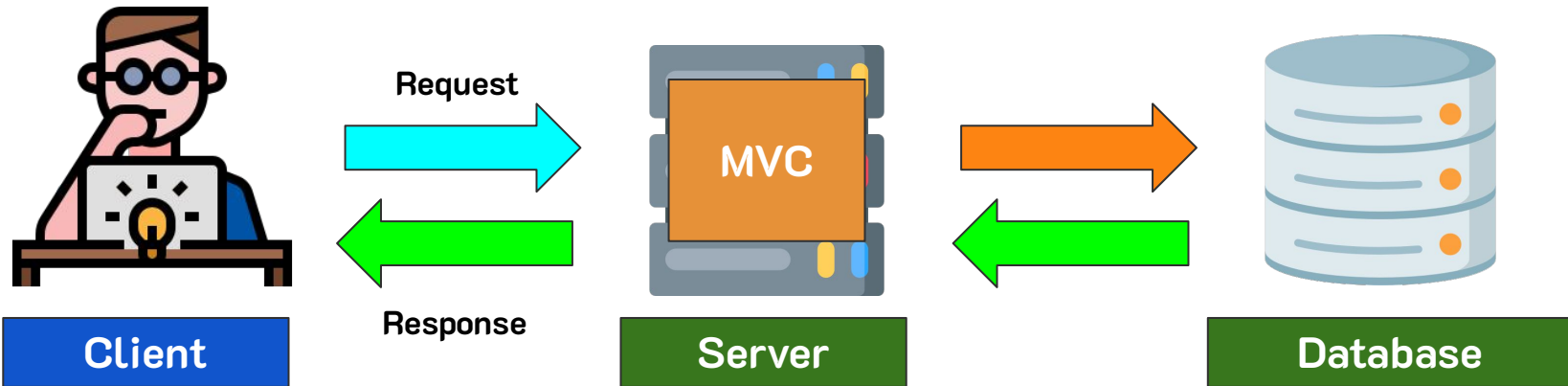




# รู้จักกับ MVC



# รู้จักกับ MVC



# รู้จักกับ MVC



Client

Routing

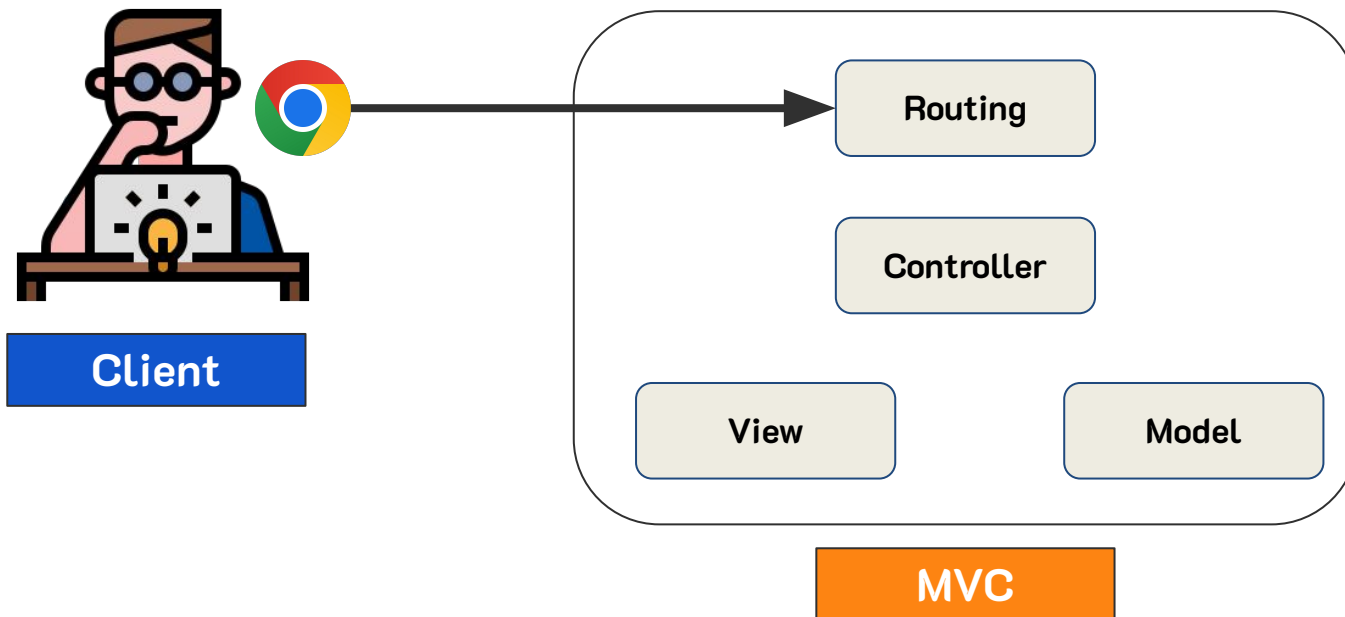
Controller

View

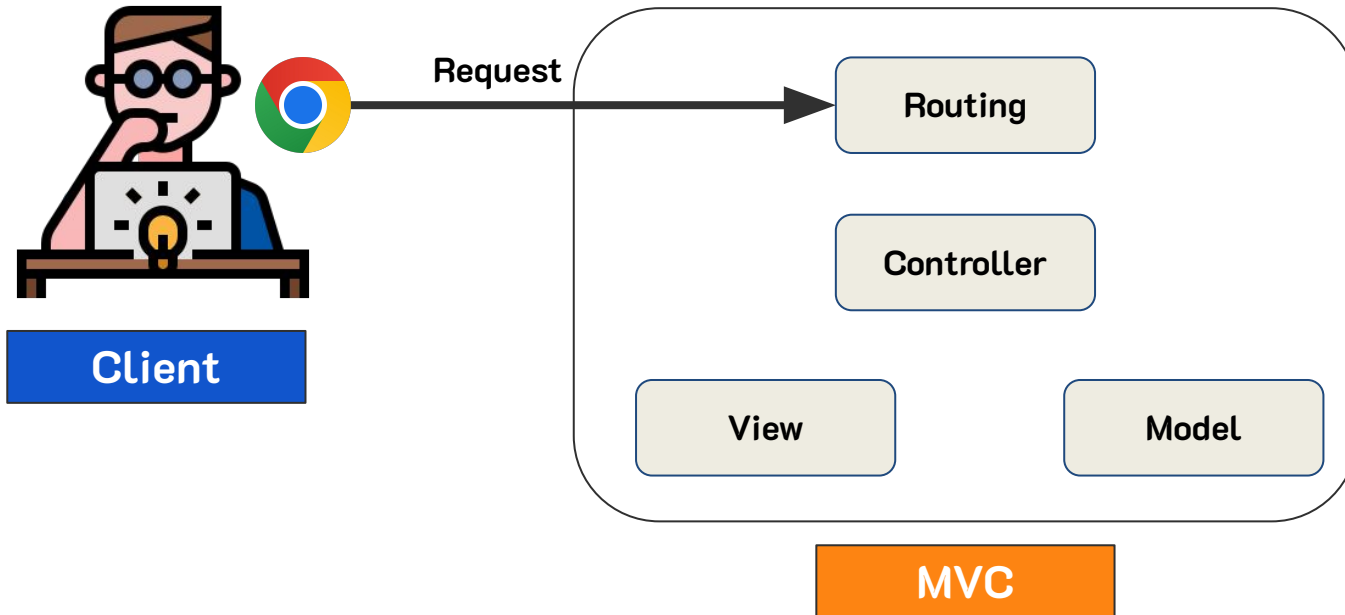
Model

MVC

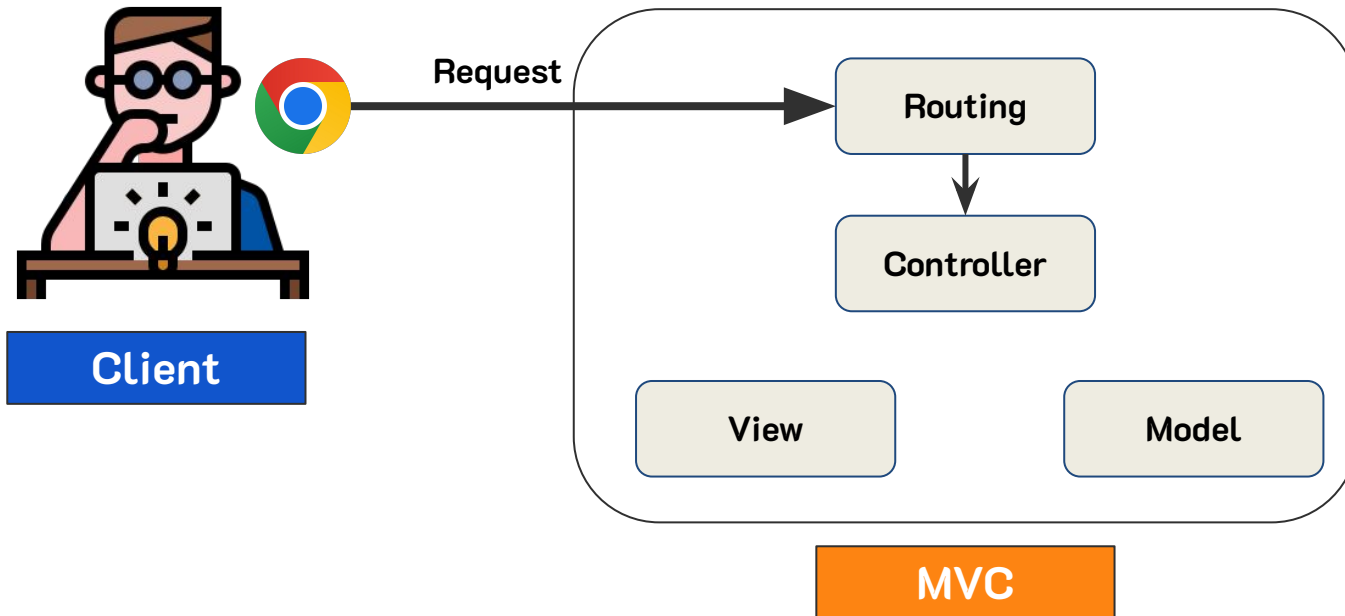
# รู้จักกับ MVC



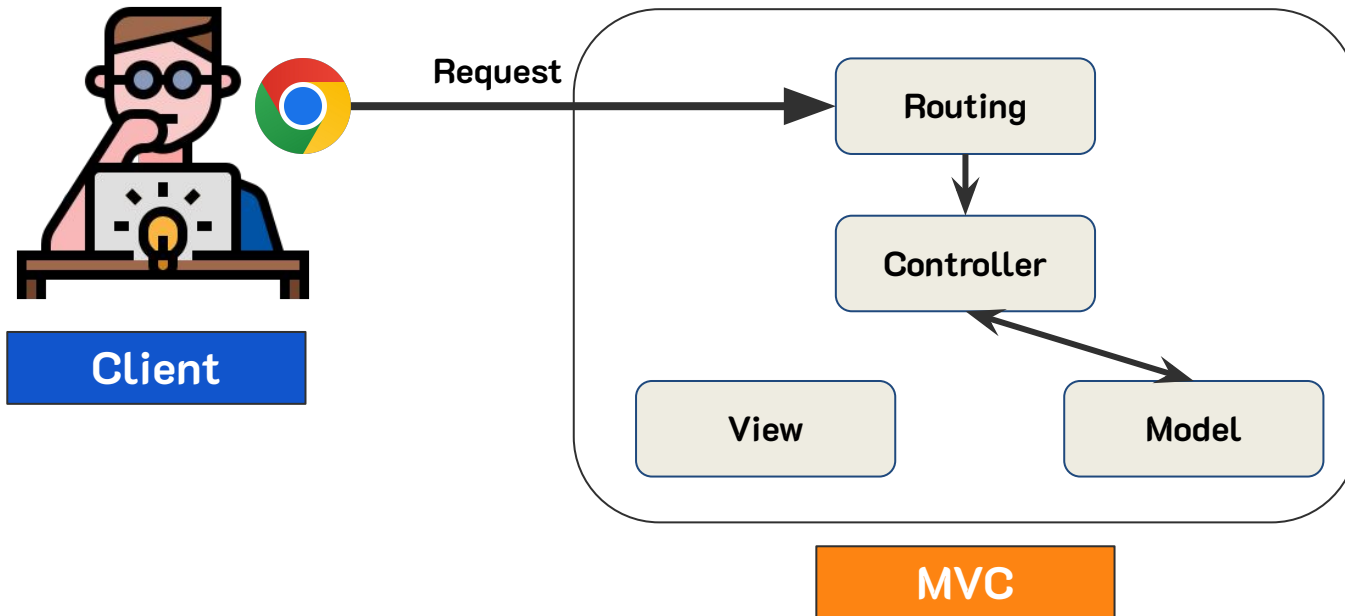
# รู้จักกับ MVC



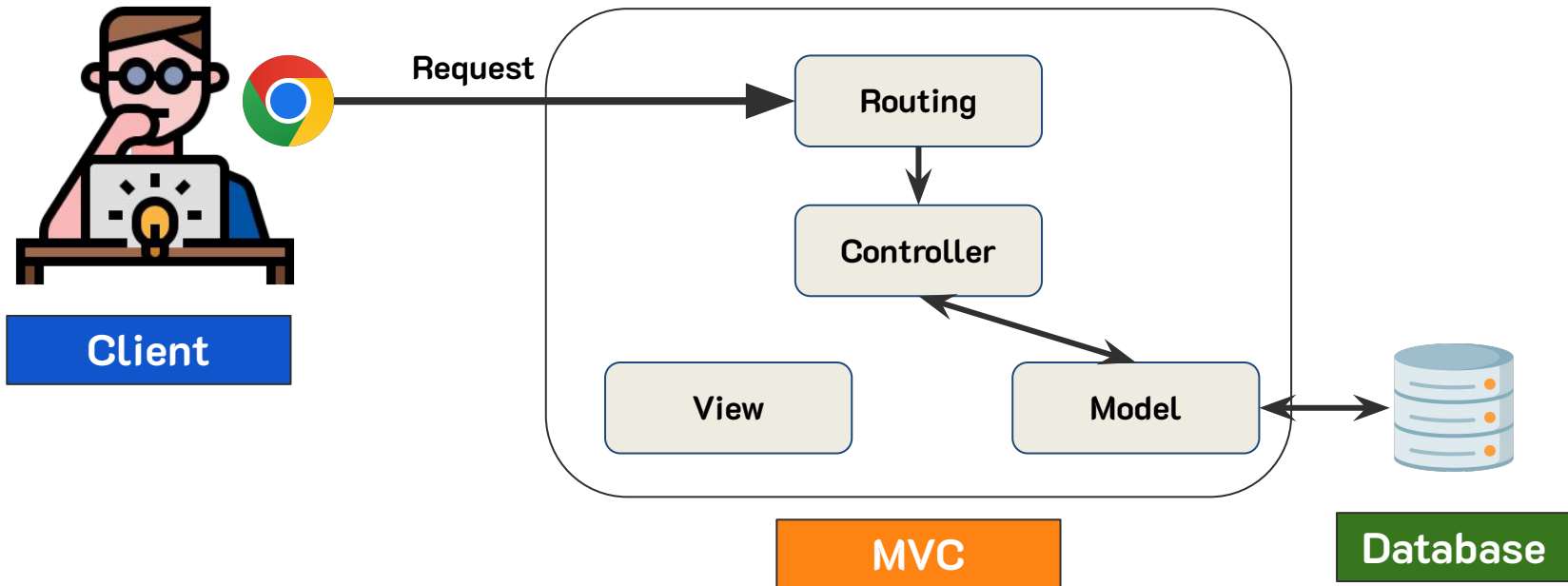
# รู้จักกับ MVC



# รู้จักกับ MVC

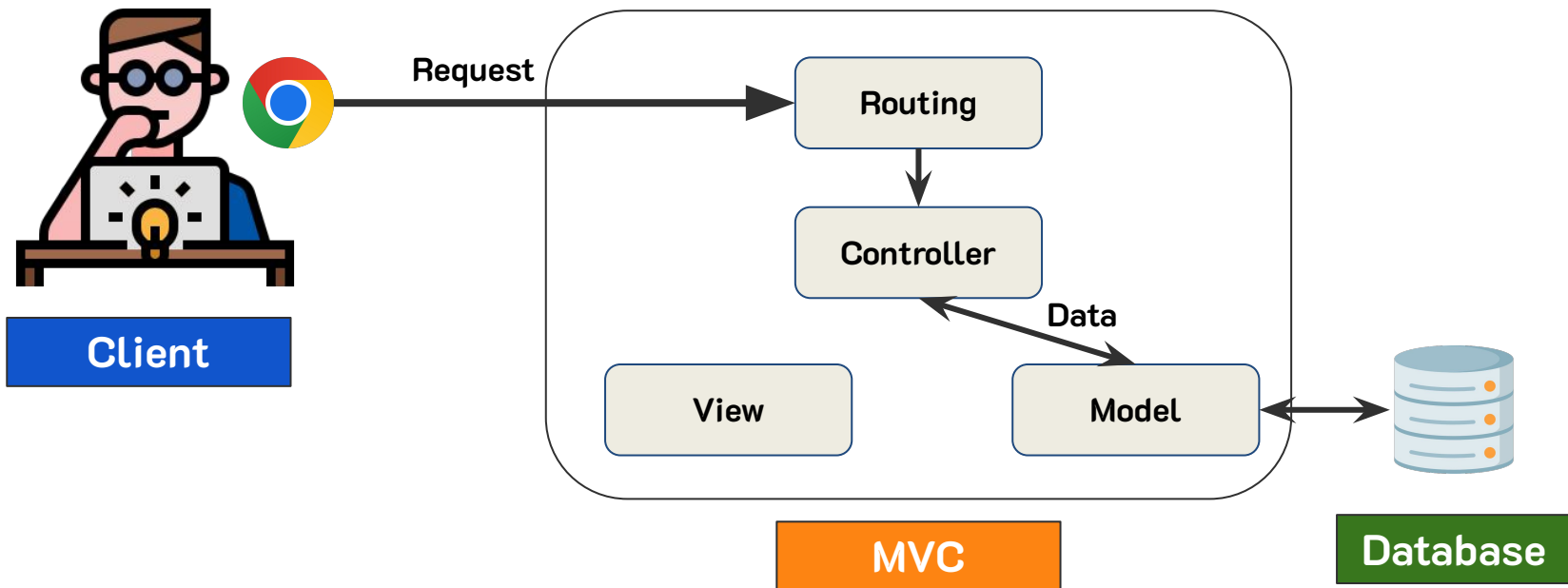


# รู้จักกับ MVC

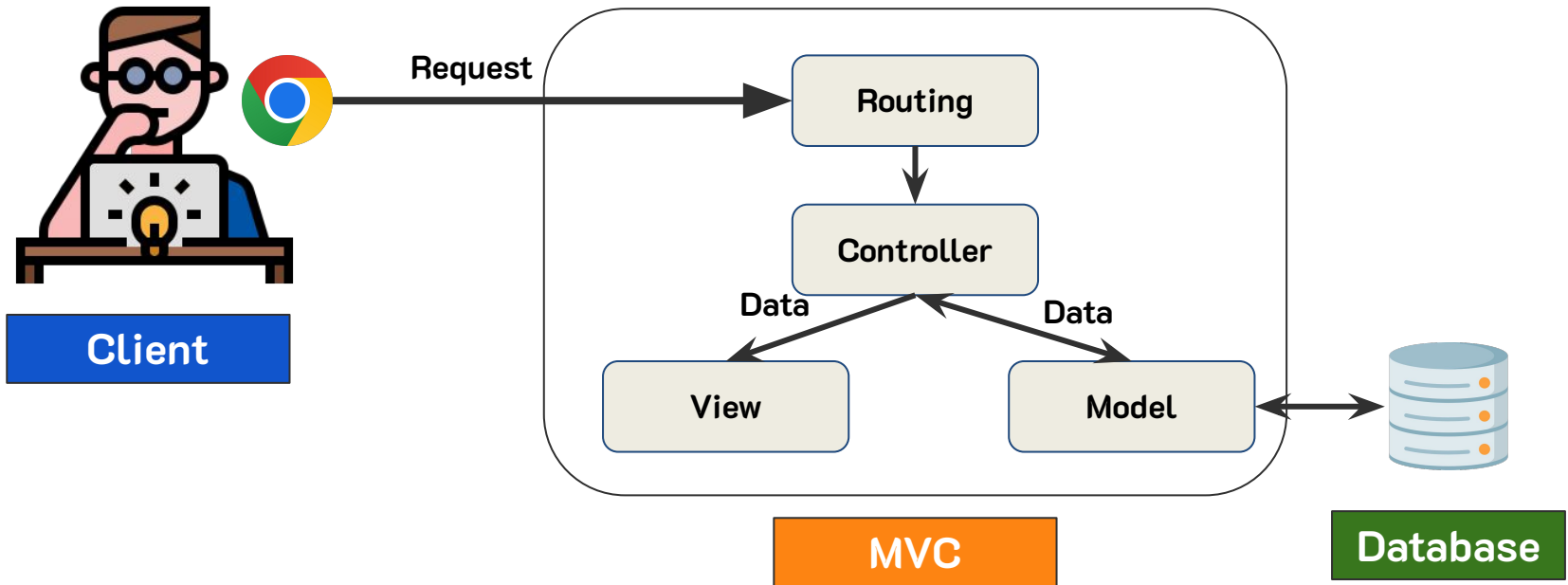




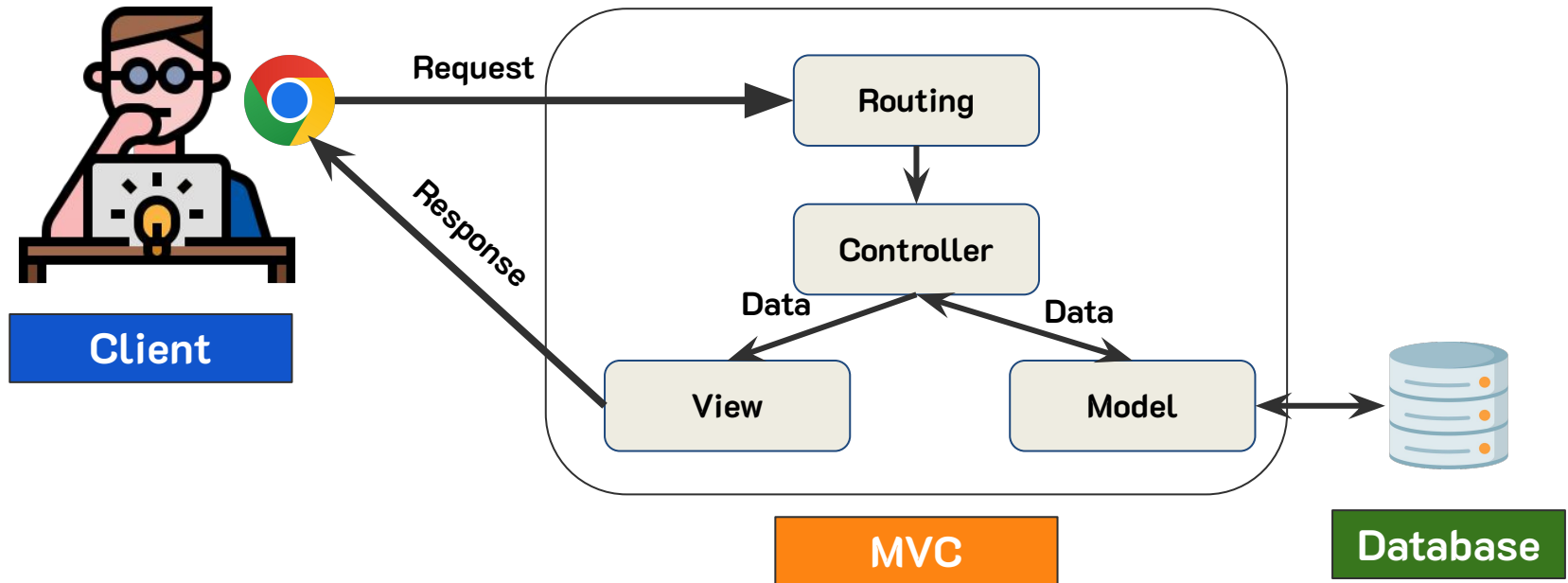
# รู้จักกับ MVC



# รู้จักกับ MVC

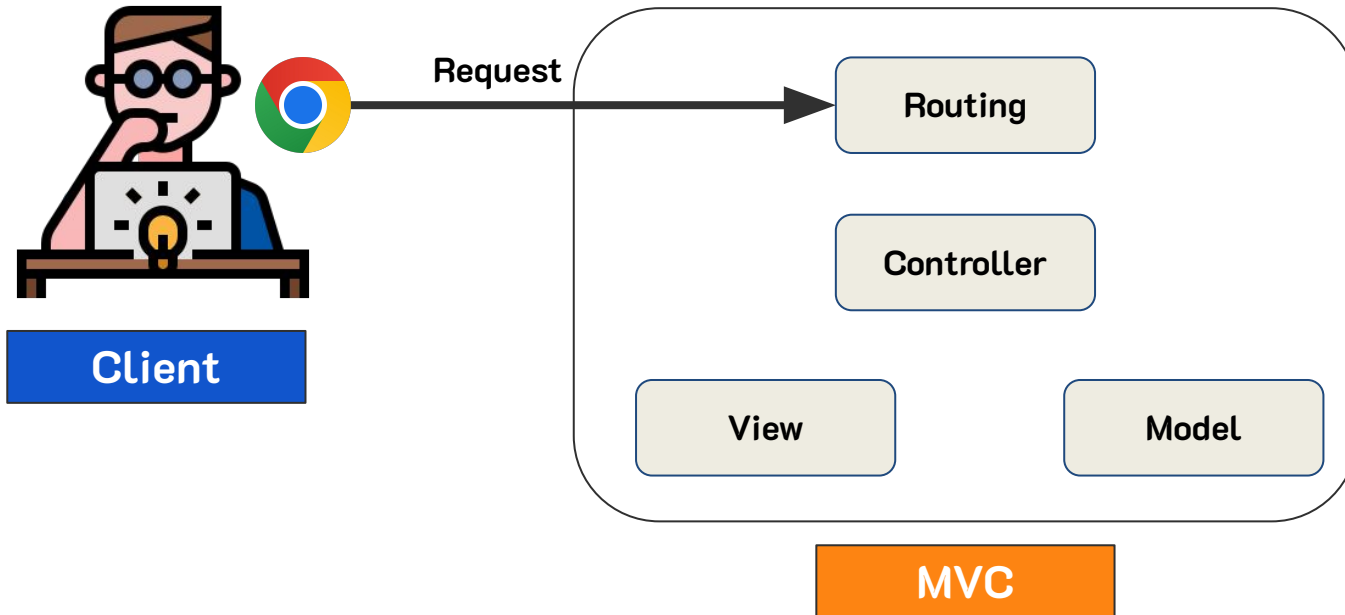


# รู้จักกับ MVC

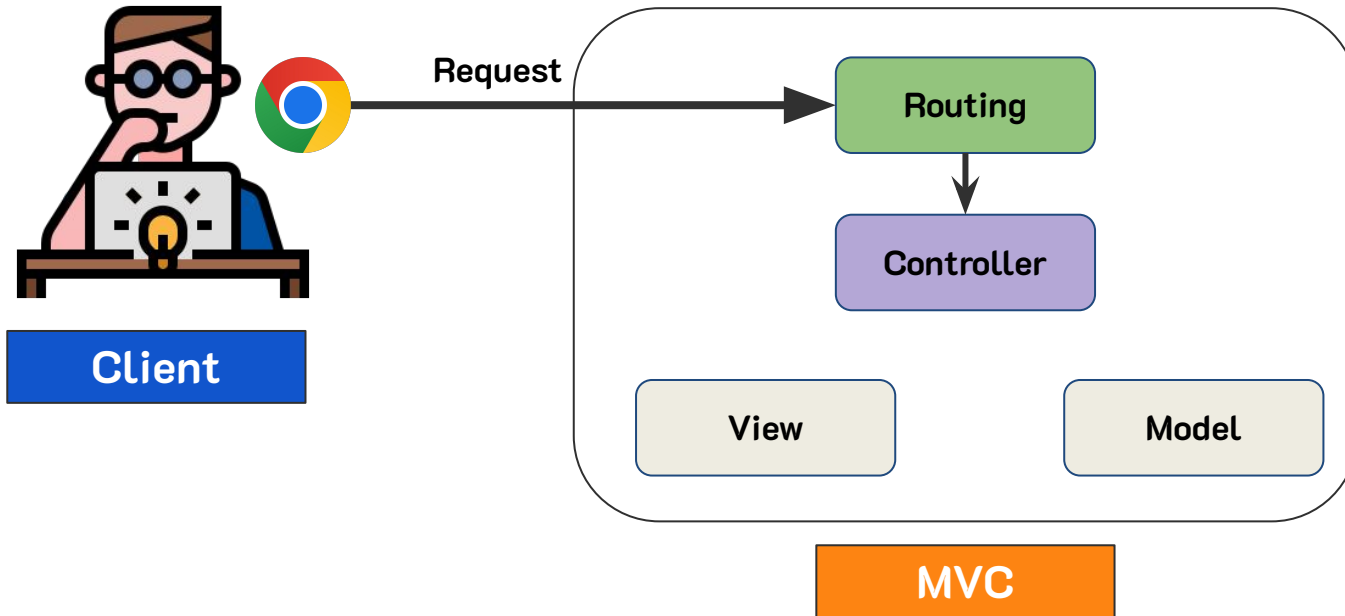


# การสร้าง Controller

# Routing



# Routing



# Routing

- **Routing** คือ ส่วนที่ใช้ระบุเส้นทางในการรับส่งข้อมูล
- **Controller** คือ ศูนย์กลางสำหรับรับส่งข้อมูลและการประมวลผลโดยเชื่อมโยงการทำงานระหว่าง Model และ View

โครงสร้าง URL

<https://www.example.com/product/computer>

# Routing

- **Routing** คือ ส่วนที่ใช้ระบุเส้นทางในการรับส่งข้อมูล
- **Controller** คือ ศูนย์กลางสำหรับรับส่งข้อมูลและการประมวลผลโดยเชื่อมโยงการทำงานระหว่าง Model และ View

## โครงสร้าง URL

Protocol

Domain

Path

<https://www.example.com/product/computer>



# Routing & Controller

## Controller

Method 1

Method 2

Method 3

**Controller (C)** คือ ส่วนประมวลผลคำสั่งต่างๆ ในแอปพลิเคชันโดยควบคุมการทำงานระหว่าง Model และ View

ภายใน Controller จะกำหนดกระบวนการทำงานผ่านส่วนที่เรียกว่าเมธอด (**Method**)

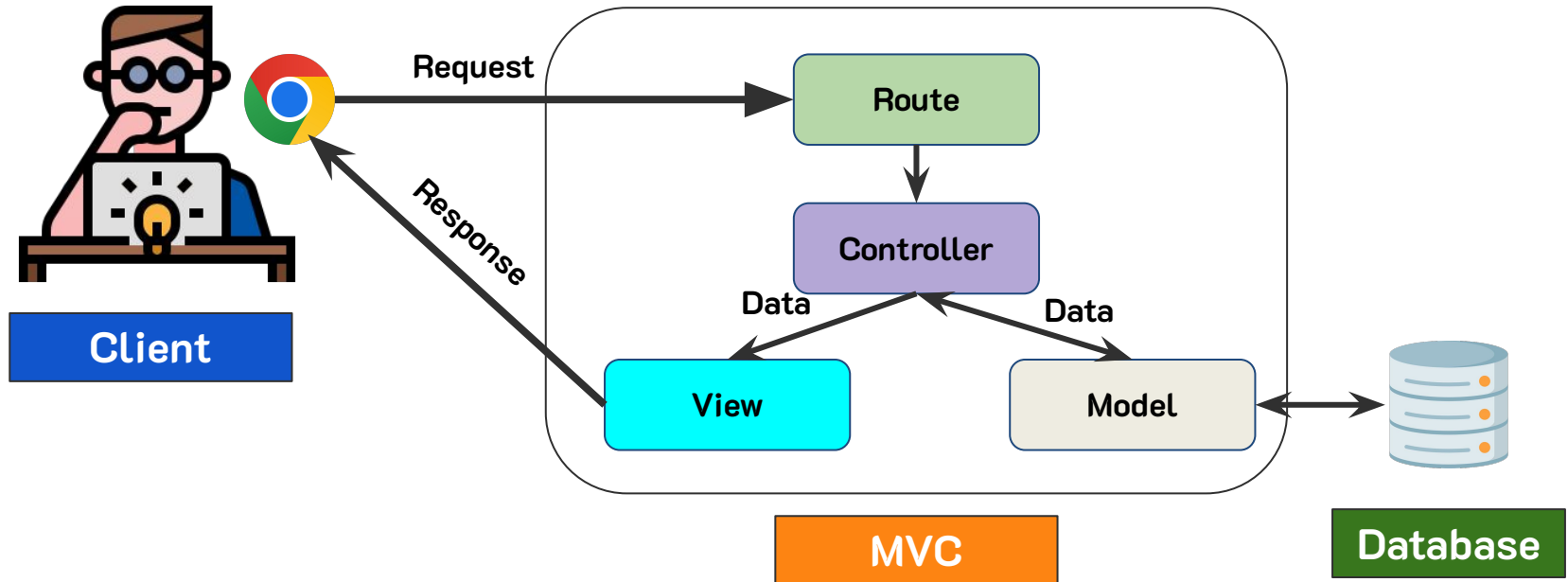
รู้จักกับ View

# รู้จักกับ View

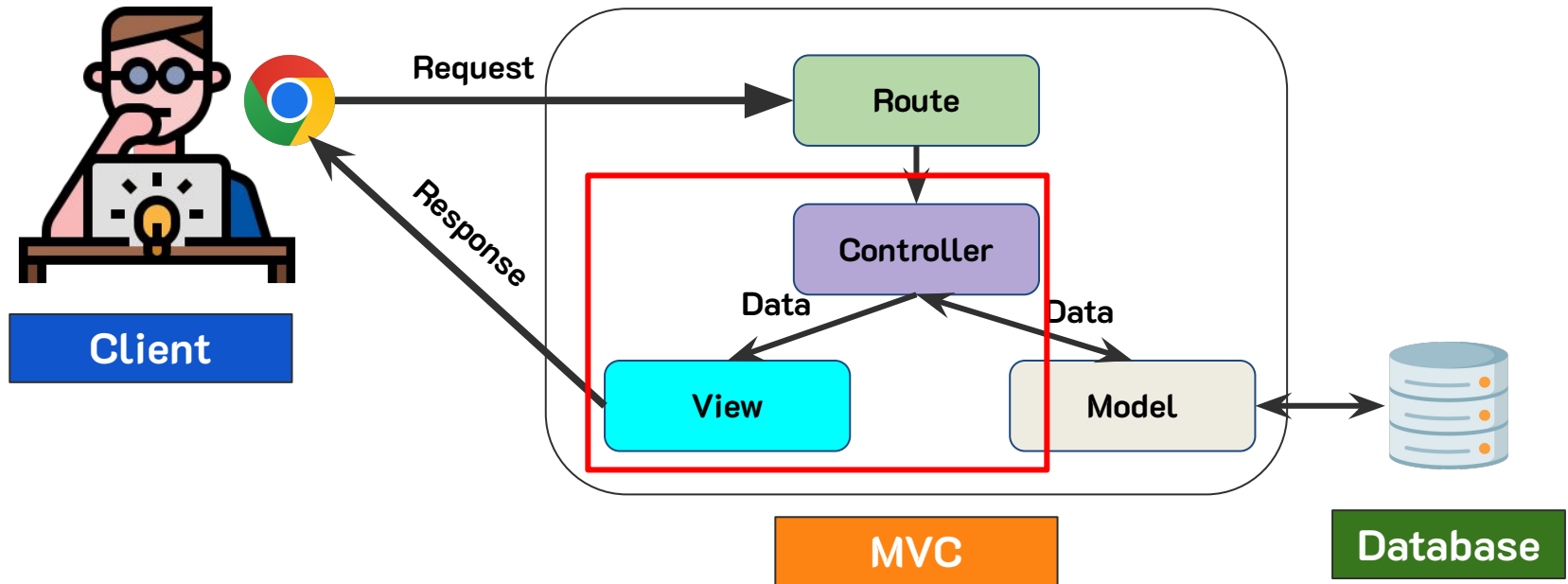


**View** คือหน้าต่างแอปพลิเคชันเป็นส่วน  
ที่ไว้ใช้แสดงผลข้อมูลหรือผลลัพธ์จากการ  
ประมวลผลในหน้าเว็บเพจโดยทำงานร่วม  
กับ HTML (Blade Template)

# รู้จักกับ View



# รู้จักกับ View



# Bootstrap คืออะไร



Frontend Framework ที่รวม HTML, CSS และ Javascript เข้าด้วยกัน เป็นชุดคำสั่งที่ถูกพัฒนาขึ้นมาเพื่อกำหนดมาตรฐานหรือรูปแบบการพัฒนาเว็บไซต์ในส่วนของ User Interface (UI) และทำให้เว็บรองรับการแสดงผลขนาดหน้าจอที่แตกต่างกันอีกด้วย



<https://www.youtube.com/c/KongRuksiamTutorial>



<https://www.facebook.com/KongRuksiamTutorial/>

# จุดเด่นของ Bootstrap



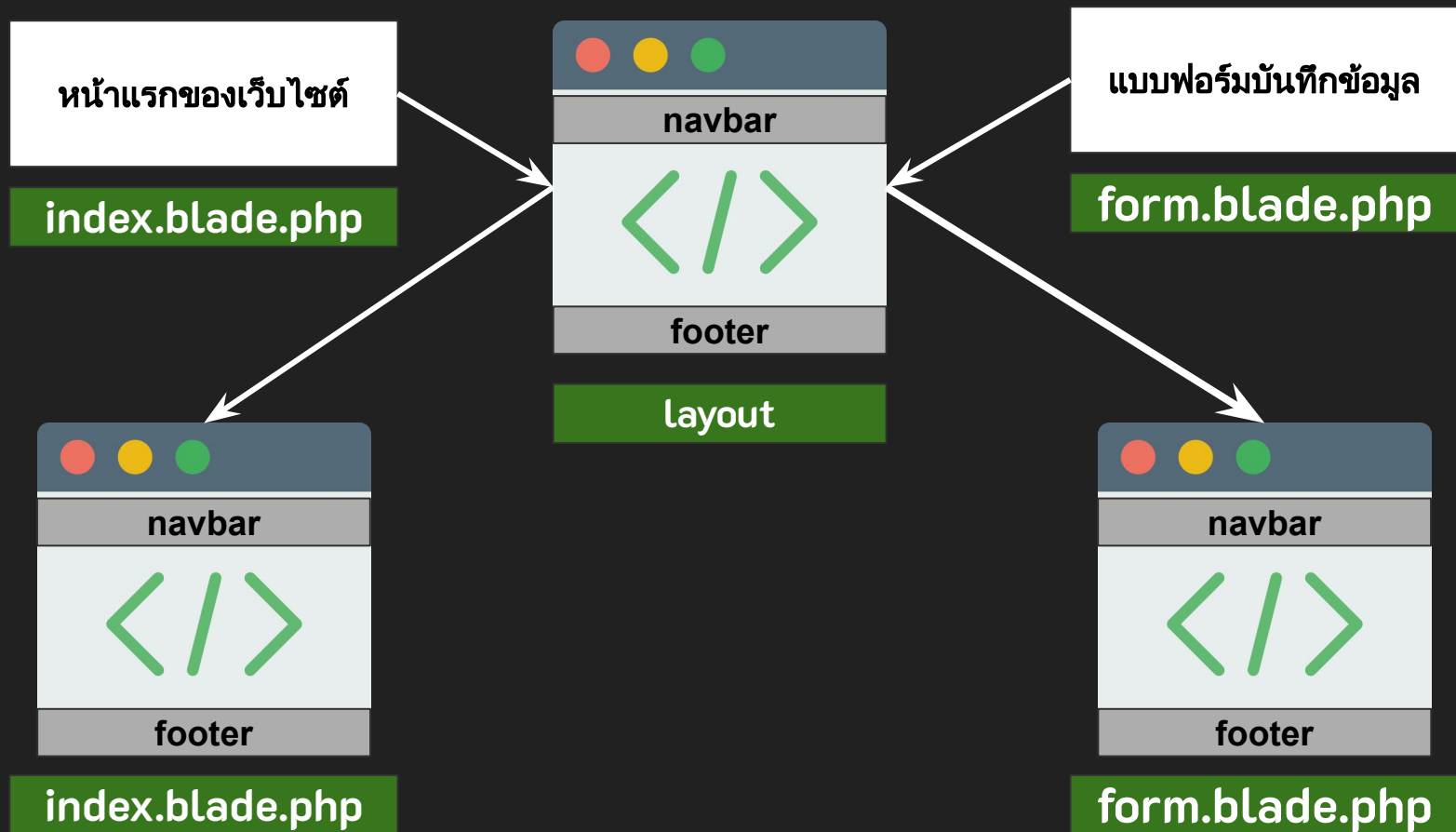
- มี UI ที่สวยงาม
- เรียนรู้ง่ายและเป็นที่นิยมใช้ทั่วโลกมีการพัฒนาและปรับปรุงต่อเนื่องจนถึงเวอร์ชัน 5
- ประหยัดเวลาในการพัฒนาเว็บไซต์
- รองรับการแสดงผลในอุปกรณ์ที่มีหน้าจอต่างขนาดกัน

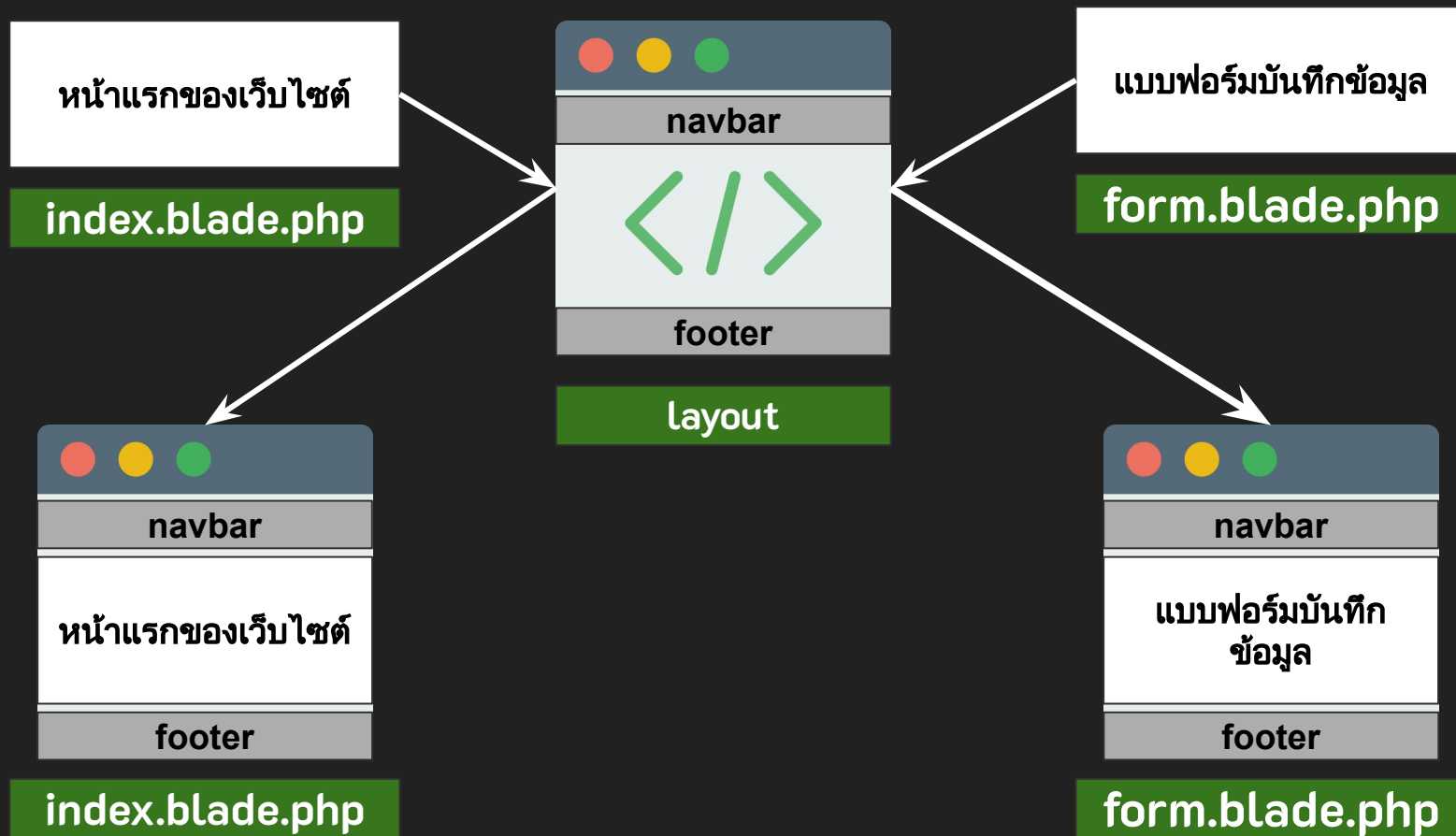
# การสร้าง Layout



# การสร้าง Layout

คือ การกำหนดโครงสร้างหลักในหน้าเว็บเพจ  
ที่ทุกๆหน้าจะใช้งานร่วมกันเพื่อลดความซ้ำซ้อน  
ของโค้ด เช่น แท็บเมนู (Navbar) , Footer เป็นต้น





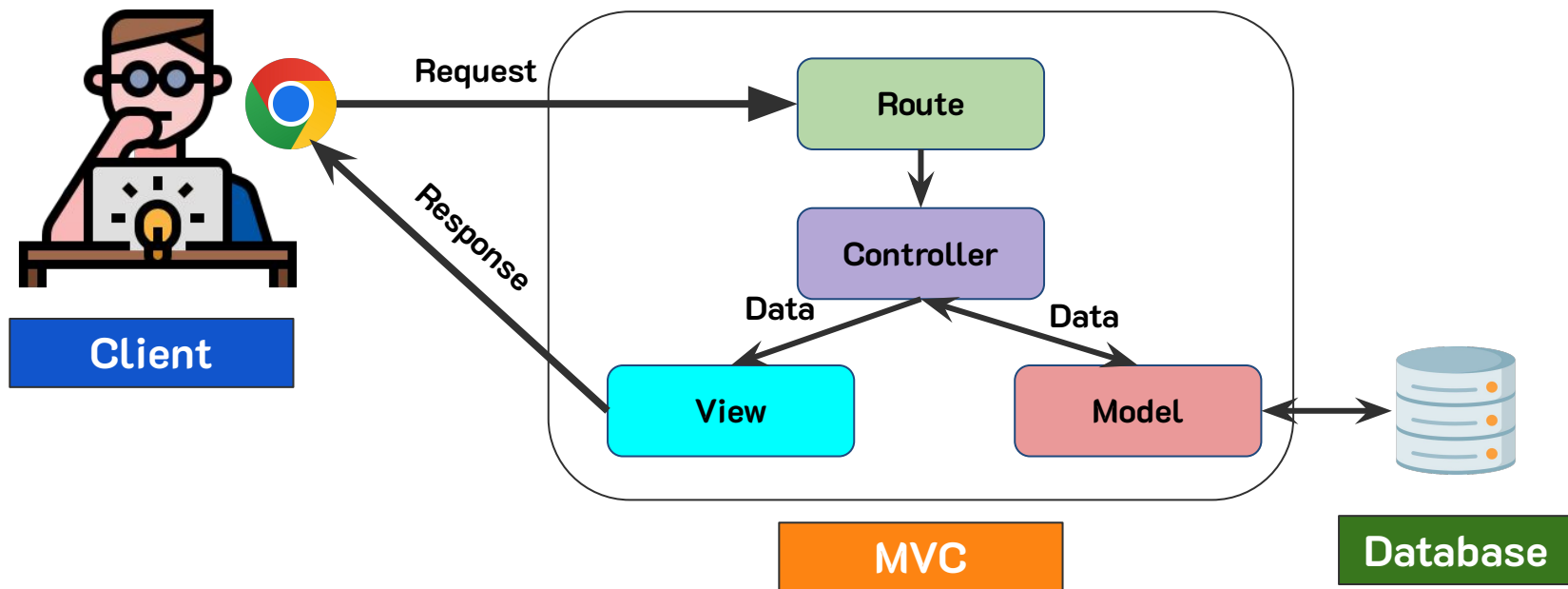
# การสร้างโมเดล (Model)

# โมเดลคืออะไร

**Model (M)** คือ ส่วนที่เก็บข้อมูลของ Application โดยจะทำหน้าที่จัดการฐานข้อมูลแทนการใช้คำสั่ง SQL โดยตรง ผ่านการกำหนด Class และ Object ซึ่งจะเรียกส่วนนี้ว่า **ORM (Object Relational Mapping)**



# รู้จักกับโมเดล



# โมเดลบทความ (Blog)

ชื่อคอลัมน์	ชนิดข้อมูล	ความหมาย
title	string	ชื่อบทความ
content	text	เนื้อหาบทความ
status	boolean	สถานะบทความ <ul style="list-style-type: none"><li>• True (1) = เผยแพร่</li><li>• False (0) = ฉบับร่าง</li></ul>

# ภาพรวมโปรเจกต์

กลุ่มผู้ใช้งานระบบ	ความสามารถ
นักเขียน (Author)	จัดการข้อมูลบทความได้ <ul style="list-style-type: none"><li>● เพิ่ม , ลบ , แก้ไขบทความ</li><li>● เผยแพร่ / ฉบับร่าง บทความ</li></ul>
ผู้ใช้งาน (User)	อ่านบทความที่เผยแพร่ได้