



# **CODING** **STANDARD**

**WELFARE REQUEST SYSTEM**

**Cluster 6**

**SOFTWARE ENGINEERING**  
**BURAPHA UNIVERSITY**

---

**CREATED BY CLUTER 6 | 2023**



### การควบคุมการเปลี่ยนแปลงเอกสาร

ตารางที่ 0 - 1 การควบคุมการเปลี่ยนแปลงเอกสาร

ชื่อเอกสาร		Coding Standard (CSD)		
เวอร์ชันปัจจุบัน		2.0.1		
วันที่แก้ไขล่าสุด		27 กุมภาพันธ์ 2566		
ผู้รับผิดชอบหลัก		Development Team		
เวอร์ชัน	วันที่	รายการที่แก้ไข	ผู้แก้ไข	หมายเหตุ
1.0.1	15 กุมภาพันธ์ 2566	จัดทำการควบคุมการเปลี่ยนแปลงเอกสาร	ยุคนธร	-
		จัดทำเนื้อหา	พงศ์เพชร, ภูริช	-
1.0.2	16 กุมภาพันธ์ 2566	จัดทำสารบัญ	ยุคนธร	-
		จัดหน้าเนื้อหา	ยุคนธร	-
2.0.1	27 กุมภาพันธ์ 2566	แก้ไขข้อมูล	ภูริช, วิษยุตม์	-
2.0.2	7 มีนาคม 2566	แก้ไขข้อมูล	ยุคนธร, วิษยุตม์, เบญญทิพย์	



# CODING STANDARD

## WELFARE REQUEST SYSTEM : WRS



หน้า | ข

### สารบัญ

หน้า

การควบคุมการเปลี่ยนแปลงเอกสาร.....	ก
สารบัญ.....	ข
สารบัญตาราง.....	ค
บทนำ.....	1
ส่วนที่ 1 มาตรฐานการเขียนโปรแกรม (Coding Standards).....	2
การตั้งชื่อไฟล์และคลาส.....	2
การจัดทำมาตรฐานเกี่ยวกับฐานข้อมูล.....	5
การเขียนคอมเมนต์.....	7



# CODING STANDARD

## WELFARE REQUEST SYSTEM : WRS



หน้า | ค

### สารบัญตาราง

ตารางที่

หน้า

0 - 1 การควบคุมการเปลี่ยนแปลงเอกสาร.....ก



# CODING STANDARD

## WELFARE REQUEST SYSTEM : WRS



หน้า | 1

### บทนำ

ในการจัดทำผลิตภัณฑ์เพื่อให้มั่นใจว่าผลิตภัณฑ์นั้นได้มีการปฏิบัติตามข้อกำหนดระเบียบหลักการเกี่ยวกับการตรวจสอบที่กำหนดไว้อย่างสม่ำเสมอและมีประสิทธิภาพสอดคล้องกับวัตถุประสงค์ของทีมการตั้งมาตรฐานให้กับผลิตภัณฑ์ทำให้ผลิตภัณฑ์มีคุณภาพตามข้อตกลงภายในทีมระบบใบเบิกสวัสดิการพนักงาน (Welfare Request System : WRS) จึงได้มีการจัดทำเอกสารสำหรับมาตรฐานของทีมโดยประกอบด้วยมาตรฐานการตรวจสอบภายในเอกสาร แผนภาพยูสเคส (Use Case Diagram) คำอธิบายยูสเคส (Use Case Description) แผนภาพกิจกรรม (Activity Diagram) แผนภาพคลาส (Class Diagram) แผนภาพลำดับการทำงาน (Sequence Diagram) แผนภาพความสัมพันธ์ของข้อมูล (Entity Relationship Diagram) แผนภาพสถานะ (State Machine Diagram) และ Specification เพื่อจะทำให้ผลิตภัณฑ์ที่ได้มีคุณภาพ และตรงกับความต้องการของลูกค้า รวมถึงเกิดข้อผิดพลาดน้อยที่สุด ก่อนผ่านการตรวจสอบโดยฝ่ายคุณภาพ (Quality)

ดังนั้น ระบบใบเบิกสวัสดิการพนักงาน (Welfare Request System : WRS) เป็นส่วนหนึ่งของเอกสารที่นำไปใช้ประกอบสำหรับการตรวจสอบคุณภาพของเอกสารก่อนนำไปเป็นตัวชี้วัดสำหรับการประเมินเอกสารของสมาชิกในทีมและประเมินทีม



## ส่วนที่ 1 มาตรฐานการเขียนโปรแกรม (Coding Standards)

มาตรฐานการเขียนโปรแกรมนี้เป็นมาตรฐานที่กำหนดขึ้นในบริบทของการพัฒนาระบบโดยใช้ Laravel 8 ซึ่งเป็นสถาปัตยกรรมซอฟต์แวร์แบบ MVC (Model View และ Controller) โดยส่วนที่มีการกำหนดมาตรฐาน ประกอบด้วยไฟล์ในโฟลเดอร์ Controllers, Models, Views, Config และ Helpers ดังนั้นจึงมีการกำหนดมาตรฐานการเขียนโปรแกรมแบ่งตามหัวเรื่อง และ MVC รวมถึง Config Helpers และมาตรฐานเกี่ยวกับฐานข้อมูล ดังนี้

### 1. การตั้งชื่อไฟล์และคลาส

เกี่ยวกับชื่อไฟล์ Controller Model และ View รวมทั้งชื่อคลาสของ Controller และ Model

#### 1.1 การตั้งชื่อไฟล์และคลาสของ Controller

##### หลักการตั้งชื่อไฟล์และคลาส

1.1.1 ตั้งชื่อไฟล์ด้วยตัวอักษรพิมพ์ใหญ่เป็นตัวขึ้นต้นเท่านั้น ถ้ามากกว่า 2 คำ ให้คั่นคำด้วยเครื่องหมายขีดล่าง ( ) และต่อด้วยอักษรพิมพ์เล็ก เช่น IT\_add\_item.php

1.1.2 การตั้งชื่อคลาสต้องเป็นชื่อเดียวกันกับชื่อไฟล์ และขึ้นต้นด้วยตัวอักษรพิมพ์ใหญ่เหมือนกัน เช่น IT\_add\_item

##### ข้อห้าม

1.1.3 ห้ามตั้งชื่อขึ้นต้นด้วยตัวอักษร C เช่น c\_project\_con\_project และ controller\_project

#### 1.2 การตั้งชื่อไฟล์และคลาสของ Model

1.2.1 โมเดลต้องประกอบด้วย 2 ไฟล์ คือ Data สำหรับเพิ่ม ลบ และแก้ไขข้อมูล และ M สำหรับรับข้อมูลมาแสดงผล



1.2.2 ตั้งชื่อไฟล์ต้องขึ้นต้นด้วยอักขรพิมพ์ใหญ่เท่านั้น จากนั้นค้นคำด้วยเครื่องหมายขีดล่าง ( \_ ) และต่อด้วยชื่อตารางของระบบเป็นตัวอักขรพิมพ์เล็กเท่านั้น เช่น Da\_ams\_it.php

### 1.3 การตั้งชื่อไฟล์ View

#### หลักการตั้งชื่อไฟล์

1.3.1 ตั้งชื่อไฟล์ด้วยตัวอักขรพิมพ์เล็กเท่านั้น และค้นคำด้วยเครื่องหมายขีดล่าง ( \_ ) เช่น v\_item\_detail.php

1.3.2 ชื่อไฟล์ต้องขึ้นต้นด้วย v\_ เช่น v\_category\_list.php

1.3.3 ชื่อ view ต้องสอดคล้องกับชื่อฟังก์ชันหรือคลาส เช่น HR\_add\_user ควรตั้งชื่อไฟล์ว่า v\_HR\_add\_user.blade.php

## 2. การตั้งชื่อฟังก์ชัน

เกี่ยวกับฟังก์ชันของ Controller และ Model

### 2.1 การตั้งชื่อฟังก์ชัน Controller

#### หลักการตั้งชื่อฟังก์ชัน

2.1.1 ตั้งชื่อฟังก์ชันด้วยตัวอักขรพิมพ์เล็กเป็นตัวขึ้นต้นของทุกคำเสมอ คำถัดไปใช้เครื่องหมายขีดล่าง ( \_ ) ในการค้น และตามด้วยตัวอักขรพิมพ์เล็ก เช่น email\_send() และ set\_session() เป็นต้น

#### หมวดของฟังก์ชัน

2.1.2 ฟังก์ชันสำหรับแสดงหน้าจอการบันทึกข้อมูลหรือแก้ไขข้อมูล

1) ฟังก์ชันการบันทึกข้อมูล ตั้งชื่อขึ้นต้นด้วย insert คำถัดไปใช้เครื่องหมายขีดล่าง ( \_ ) ในการค้น และตามด้วยตัวอักขรพิมพ์เล็ก เช่น insert\_login\_form() เป็นต้น



2) ฟังก์ชันการแก้ไขข้อมูล ตั้งชื่อขึ้นต้นด้วย update คำถัดไปใช้เครื่องหมายขีดล่าง ( ) ในการค้น และตามด้วยตัวอักษรพิมพ์เล็ก เช่น update\_item() และ update\_role\_by\_name() เป็นต้น

### 2.1.3 ฟังก์ชันสำหรับการบันทึก หรือแก้ไขลงฐานข้อมูล

1) ฟังก์ชันสำหรับการบันทึกหรือเพิ่มข้อมูล ตั้งชื่อขึ้นต้นด้วย insert คำถัดไปใช้เครื่องหมายขีดล่าง ( ) ในการค้น และตามด้วยตัวอักษรพิมพ์เล็ก เช่น insert\_item()

2) ฟังก์ชันสำหรับการแก้ไขข้อมูล ตั้งชื่อขึ้นต้นด้วย update คำถัดไปใช้เครื่องหมายขีดล่าง ( ) ในการค้น และตามด้วยตัวอักษรพิมพ์เล็ก เช่น update item() และ update\_password() เป็นต้น

2.1.4 ฟังก์ชันสำหรับการลบข้อมูลในฐานข้อมูลตั้งชื่อขึ้นต้นด้วย delete คำถัดไปใช้เครื่องหมายขีดล่าง ( ) ในการค้น และตามด้วยตัวอักษรพิมพ์เล็ก เช่น delete\_item() และ delete\_account() เป็นต้น

### 2.1.5 ฟังก์ชันสำหรับการแสดง

1) ฟังก์ชันสำหรับแสดงข้อมูล ตั้งชื่อขึ้นต้นด้วย show คำถัดไปใช้เครื่องหมายขีดล่าง ( ) ในการค้น และตามด้วยตัวอักษรพิมพ์เล็ก เช่น show\_list\_item() และ show\_detail\_item(Sitem\_id) เป็นต้น

2.1.6 ฟังก์ชันสำหรับการนำเข้าข้อมูล ตั้งชื่อขึ้นต้นด้วย upload คำถัดไปใช้เครื่องหมายขีดล่าง ( ) ในการค้น และตามด้วยตัวอักษรพิมพ์เล็ก เช่น upload\_image() เป็นต้น

2.1.7 ฟังก์ชันอื่น ๆ ตั้งชื่อให้สอดคล้องกับการทำงาน เช่น set\_session() หรือ logout() เป็นต้น





## 2.2 การตั้งชื่อฟังก์ชัน Model

### หลักการตั้งชื่อฟังก์ชัน

2.2.1 ตั้งชื่อฟังก์ชันด้วยตัวอักษรพิมพ์เล็กเป็นตัวขึ้นต้นของทุกคำเสมอคำถัดไปใช้เครื่องหมายขีดล่าง ( \_ ) ในการคั่น และตามด้วยตัวอักษรพิมพ์เล็ก เช่น `check_item()` และ `get_all()` เป็นต้น

2.2.2 ฟังก์ชันสำหรับเลือกข้อมูลทั้งหมด ตั้งชื่อขึ้นต้นด้วย `get_all` คำถัดไปใช้เครื่องหมายขีดล่าง ( \_ ) ในการคั่น และตามด้วยตัวอักษรพิมพ์เล็ก เช่น `get_all_user`

2.2.3 ฟังก์ชันสำหรับเลือกข้อมูลบางส่วน ตั้งชื่อขึ้นต้นด้วย `get` คำถัดไปใช้เครื่องหมายขีดล่าง ( \_ ) ในการคั่น และตามด้วยเครื่องหมายพิมพ์เล็ก เช่น `get_user_by_name` เป็นต้น

2.2.4 ฟังก์ชันอื่น ๆ ตั้งชื่อให้สอดคล้องกับการทำงาน เช่น `search_with_filter()`

## 3. การตั้งชื่อตัวแปร

### 3.1 ตัวแปรทั่วไปหรือ Array

3.1.1 ให้ตั้งชื่อตัวแปรด้วยรูปแบบ Snake Case

3.1.2 ใช้ชื่อตัวแปรให้สอดคล้องกับข้อมูล เช่น `$item` และ `$item_id` เป็นต้น

### 3.2 ตัวแปรในรอบของลูป

3.2.1 ใช้ตัวแปร `$i` เพื่อนับบรรทัดของลูป

3.2.2 ใช้ตัวแปร `$i` `$j` และ `$k` หรือ `$x` `$y` และ `$z` ร่วมกัน กรณีมีลูปมากกว่า 1 ลูป ได้ตามความเหมาะสม

## 4. การจัดทำมาตรฐานเกี่ยวกับฐานข้อมูล

### 4.1 การตั้งชื่อฐานข้อมูล

ให้ตั้งชื่อฐานข้อมูลเป็นอักษรพิมพ์เล็กทั้งหมดตามด้วยเครื่องหมายขีดล่าง



(\_) ตัวอย่างเช่น welfare\_request\_system เป็นต้น

### 4.2 การตั้งชื่อตาราง

ให้ตั้งชื่อตารางโดยใช้ชื่อข้อมูลแบบเป็นพหูพจน์หรือคำนามที่แสดงถึงสิ่งที่มีจำนวนตั้งแต่สองหน่วยขึ้นไป ตัวอย่างเช่น admin และ it\_support เป็นต้น

### 4.3 การตั้งชื่อฟิลด์

ให้ตั้งชื่อฟิลด์โดยตัวอักษรภาษาอังกฤษพิมพ์เล็กทั้งหมดเป็นชื่อของข้อมูล คำถัดไปคั่นด้วยเครื่องหมายขีดล่าง (\_) ตัวอย่างเช่น adm\_id adm\_name และ adm\_email เป็นต้น

### 4.4 การเขียนคอมเมนต์ (Comment) ของตารางและฟิลด์

ทุกตาราง และทุกฟิลด์ต้องมีการคอมเมนต์ หรือนิยามความหมายกำกับไว้ให้ครบถ้วน ไม่มีข้อยกเว้น

#### หลักการเขียนคอมเมนต์

4.4.1 ตารางให้นิยามความหมายว่า ตารางเก็บข้อมูลใด หรือใช้สำหรับทำอะไร ตัวอย่างเช่นตาราง ams\_admin คือ ตารางเก็บข้อมูลของผู้ดูแลระบบ

4.4.2 ฟิลด์ให้นิยามความหมายว่าใช้เก็บข้อมูลใด ตัวอย่างเช่น

- adm\_id คือ รหัสผู้ดูแลระบบ
- adm\_name คือ ชื่อผู้ดูแลระบบ
- adm\_email คือ อีเมลผู้ดูแลระบบ

4.4.3 การระบุตัวอย่างของข้อมูล หากฟิลด์นั้นมีตัวอย่างของข้อมูลชัดเจน ให้ใส่ตัวอย่างในเครื่องหมายวงเล็บด้วย ตัวอย่างเช่น item\_status คือ สถานะของอุปกรณ์ (Y = ใช้งาน N = ไม่ใช้งาน)

4.4.4 ฟิลด์ที่อ้างอิงจากฟิลด์อื่น (FK) ให้อธิบายความหมายเดียวกันกับตารางตั้งต้น (PK) และต่อท้ายด้วยว่ามาจากตารางใดหรือฐานข้อมูลใด (ระบุชื่อฐานข้อมูลด้วยหากอยู่คนละฐานข้อมูล) ตัวอย่างเช่น agd\_personId คือ รหัสบุคลากร (ตาราง hr\_persons)



## 5. การเขียนคอมเมนต์

### 5.1 คอมเมนต์คลาสของ Controller และ Model

ในคลาสของ Controller และ Model ให้เขียนคอมเมนต์รูปแบบเดียวกัน

#### ข้อบังคับ

5.1.1 ให้เขียนคอมเมนต์ทุกไฟล์ ไม่มีข้อยกเว้น

5.1.2 ให้เขียนคอมเมนต์ฟังก์ชันกำกับทุกฟังก์ชัน ไม่มีข้อยกเว้น

5.1.3 เขียนคอมเมนต์คลาสไว้บรรทัดแรกของไฟล์

5.1.4 เขียนคอมเมนต์คลาสด้วยภาษาอังกฤษหรือภาษาไทยเท่านั้น

#### หลักการเขียนคอมเมนต์ส่วนของ Class

5.1.5 บรรทัดที่ 1 ใช้เครื่องหมายเปิดคอมเมนต์ คือ /\*

5.1.6 บรรทัดที่ 2 ระบุชื่อคลาส เช่น Login

5.1.7 บรรทัดที่ 3 ระบุชื่อคลาสและข้อความเกี่ยวกับการทำงานคร่าว ๆ  
เช่น Member Details of Member

5.1.8 บรรทัดที่ 4 ระบุชื่อและรหัสนิสิตผู้สร้างไฟล์คลาสหลังหัวข้อ @author  
เช่น @author : Wichayut Loarod 64160172

5.1.9 บรรทัดที่ 5 ระบุวันที่สร้างไฟล์คลาสหลังหัวข้อ @Create Date  
เช่น @Create Date : 2022-07-09

5.1.10 บรรทัดที่ 6 ใช้เครื่องหมายปิดคอมเมนต์ คือ \*/

#### หมายเหตุ

แต่ละบรรทัดให้ใส่เครื่องหมาย \* และเว้นวรรค 1 ครั้งก่อนเขียนข้อความ  
เสมอ (ยกเว้นบรรทัดที่ 1 และบรรทัดที่ 6)



ตัวอย่างการคอมเมนต์คลาส

/\*

\* Member

\* Member Details of Member

\* @author : Wichayut Loarod 64160172

\* @Create Date : 2022-07-09

\*/

## 5.2 คอมเมนต์ฟังก์ชันใน Controller, Model Helper และ View

ในฟังก์ชันของ Controller ModelHelper และ View (ส่วนที่เป็น JavaScript)  
ให้เขียนคอมเมนต์รูปแบบเดียวกัน

ข้อบังคับ

5.2.1 เขียนคอมเมนต์กำกับทุกฟังก์ชัน ไม่มีข้อยกเว้น

5.2.2 เขียนคอมเมนต์ฟังก์ชันไว้ด้านบน ก่อนประกาศฟังก์ชันนั้น ๆ

5.2.3 เขียนคอมเมนต์ฟังก์ชันด้วยภาษาอังกฤษ หรือภาษาไทยเท่านั้น  
หลักการเขียนคอมเมนต์ส่วนของฟังก์ชัน

5.2.4 บรรทัดที่ 1 ใช้เครื่องหมายเปิดคอมเมนต์ คือ /\*

5.2.5 บรรทัดที่ 2 ระบุฟังก์ชัน เช่น check\_email\_admin()

5.2.6 บรรทัดที่ 3 ระบุชื่อข้อความเกี่ยวกับการทำงานคร่าว ๆ เช่น check  
email in database

5.2.7 บรรทัดที่ 4 ระบุข้อมูลที่ระบบต้องรับเข้ามาหลังหัวข้อ @input  
เช่น @input : user\_email



# CODING STANDARD

## WELFARE REQUEST SYSTEM : WRS



หน้า | 9

5.2.8 บรรทัดที่ 5 ระบุข้อมูลที่ระบบแสดงหรือข้อมูลส่งออกมา หลังหัวข้อ @output เช่น @output : user\_email

5.2.9 บรรทัดที่ 6 ระบุชื่อผู้สร้างไฟล์คลาสหลังหัวข้อ @author เช่น @author : Wichayut Loarod

5.2.10 บรรทัดที่ 7 ระบุวันที่สร้างไฟล์คลาสหลังหัวข้อ @Create Date เช่น @Create Date : 2565-07-09

5.2.11 บรรทัดที่ 8 ใช้เครื่องหมายปิดคอมเมนต์ คือ \*/

5.2.12 แต่ละบรรทัดให้ใส่เครื่องหมาย \* และเว้นวรรค 1 ครั้งก่อนเขียนข้อความเสมอ (ยกเว้นบรรทัดที่ 1 และบรรทัดที่ 8)

5.2.13 กรณีไม่มีข้อมูลรับเข้าหรือส่งออกให้ใส่เครื่องหมายขีด (-)

ตัวอย่างการคอมเมนต์ส่วนของฟังก์ชัน

/\*

\* check\_email\_admin()

\* check\_email\_in\_database

\* @input : user\_email

\* @output : user\_email

\* @author : Wichayut Loarod 64160172

\* @Create Date : 2022-07-09

\*/



### 5.3 คอมเมนต์ส่วนของ View

#### ข้อบังคับ

- 5.3.1 ให้เขียนคอมเมนต์ทุกไฟล์ View ไม่มีข้อยกเว้น
- 5.3.2 หากมีฟังก์ชันในส่วนของ View ต้องเขียนคอมเมนต์ฟังก์ชันกำกับทุกฟังก์ชันไม่มีข้อยกเว้น
- 5.3.3 เขียนคอมเมนต์ส่วนของ View ไว้บรรทัดแรกของไฟล์
- 5.3.4 เขียนคอมเมนต์ส่วนของ View ด้วยภาษาอังกฤษ หรือภาษาไทยเท่านั้น  
หลักการเขียนคอมเมนต์ส่วนของ View
- 5.3.5 บรรทัดที่ 1 ใช้เครื่องหมายเปิดคอมเมนต์ คือ /\*
- 5.3.6 บรรทัดที่ 2 ระบุชื่อไฟล์ View เช่น v\_edit\_company.blade.php
- 5.3.7 บรรทัดที่ 3 ระบุชื่อข้อความเกี่ยวกับการทำงานคร่าว ๆ เช่น Display form edit company by entrepreneur
- 5.3.8 บรรทัดที่ 4 ระบุข้อมูลที่ระบบต้องรับเข้ามาหลังหัวข้อ @input เช่น @input : com\_name com\_description com\_tel com\_file[] com\_lat และ com\_lon
- 5.3.9 บรรทัดที่ 5 ระบุข้อมูลที่ระบบแสดงหรือข้อมูลส่งออกมา หลังหัวข้อ @output เช่น @output : form edit company
- 5.3.10 บรรทัดที่ 6 ระบุชื่อผู้สร้างไฟล์คลาสหลังหัวข้อ @author เช่น @author : Wichayut Laorod 64160172
- 5.3.11 บรรทัดที่ 7 ระบุวันที่สร้างไฟล์คลาสหลังหัวข้อ @Create Date เช่น @Create Date : 2022-07-09
- 5.3.12 บรรทัดที่ 8 ใช้เครื่องหมายปิดคอมเมนต์ คือ \*/หมายเหตุ
- 5.3.13 แต่ละบรรทัดให้ใส่เครื่องหมาย \* และเว้นวรรค 1 ครั้งก่อนเขียนข้อความเสมอ (ยกเว้นบรรทัดที่ 1 และบรรทัดที่ 8)



5.3.14 กรณีไม่มีข้อมูลรับหรือส่งออกเข้าให้ใส่เครื่องหมายขีด (-)

ตัวอย่างการคอมเมนต์ส่วนของ View

/\*

\* v\_edit\_company.blade.php

\* Display form edit company by entrepreneur

\*@input : com\_namecom\_descriptioncom\_telcom\_file[] com\_la

และ com\_lon

\* @output : form edit company

\* @author : Wichayut Loarod 64160172

\* @Create Date : 2022-07-09

\*/

#### 5.4 คอมเมนต์บรรทัดเดียวหรือตัวแปรต่าง ๆ

กรณีต้องการคอมเมนต์เพื่อบรรยายความหมายของตัวแปร หรือส่วนการ

ทำงานบรรทัดนั้น ๆ หลักการเขียนคอมเมนต์

5.4.1 ขึ้นต้นด้วยเครื่องหมายคอมเมนต์ก่อนที่ประกาศตัวแปร 1 บรรทัด เช่น

ตัวอย่างการคอมเมนต์

//Set default value of the variable i

\$i=0;

// Input Password

<input type="password" name="password">



### 5.5 คอมเมนต์สำหรับการแก้ไขไฟล์หรือฟังก์ชัน Controller Model และ View

กรณีต้องการคอมเมนต์เพื่อแสดงการแก้ไขไฟล์หรือฟังก์ชัน

#### หลักการเขียนคอมเมนต์

5.5.1 ให้เขียนต่อจากคอมเมนต์ในส่วนของหัวไฟล์หรือฟังก์ชันนั้น โดยระบุครั้งที่แก้ไข วันที่ทำการแก้ไข ชื่อผู้แก้ไข (กรณีไม่ใช่ผู้สร้างไฟล์ หรือฟังก์ชันนั้น) และอธิบายสิ่งที่ทำการแก้ไข เช่น @Update Date2: 2022-07-09 Wichayut Loarod 64160172 แก้ไขส่วนของปุ่มลึ้มรหัสผ่าน

#### ตัวอย่างการคอมเมนต์สำหรับการแก้ไขไฟล์หรือฟังก์ชัน

/\*

\* v\_forgot\_password\_tourist.blade.php

\* Displays a screen for entering your email address to recover your password or reset it.

\* @input : user\_email

\* @output : -

\* @author : Wichayut Loarod 64160172

\* @Create Date : 2022-07-09

\* @Update Date 1: 2022-07-10 Yukonthorn Jantranuwatkul  
64160298 edit button confirm

\* @Update Date 2: 2022-07-11 Sirinya Sondirok 64160074 edit  
button cancel

\*/