

ร้านค้าสำหรับแอพพลิเคชันการประมวลผลภาพซึ่งจัดการงานบนระบบ

ประมวลผลแบบกลุ่ม

Marketplace for Image Processing Application using Task

Management on Cluster Computing System

พศิน จันทร์ทัน

สุชี สาระพันธ์

ปริญญาอิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2565

ปริญญาบัณฑีปการศึกษา 2565

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ร้านค้าสำหรับแอพพลิเคชั่นการประมวลผลภาพซึ่งจัดการงานบนระบบประมวลผลแบบกลุ่ม

Marketplace for Image Processing Application using Task Management on Cluster Computing

System

**ผู้จัดทำ**

1. นายพศิน จันทร์หัน รหัสนักศึกษา 63015121

2. นายสุรี สาระพันธ์ รหัสนักศึกษา 63015190

อาจารย์ที่ปรึกษา

(รศ. ดร.อรุณัตร จิตต์ไสวภัตร)

# ร้านค้าสำหรับแอพพลิเคชั่นการประมวลผลภาพซึ่งจัดการงานบน

## ระบบประมวลผลแบบกลุ่ม

นายพศิน	จันทร์ทัน	63015121
นายสุธี	สาระพันธ์	63015190
รศ.ดร.อรนัตร์	จิตต์โสภกตร์	อาจารย์ที่ปรึกษา
ปีการศึกษา 2565		

### บทคัดย่อ

ในปัจจุบันการประมวลผลภาพเป็นสิ่งสำคัญที่ใช้ทรัพยากรที่มากขึ้นเนื่องจากมีการใช้งานอย่างแพร่หลายในหลากหลายกลุ่มอุตสาหกรรม อย่างไรก็ตาม การใช้บริการงานประมวลผลบนอินเทอร์เน็ต (Cloud Processing) ที่มีค่าใช้จ่ายสูงอาจไม่เป็นทางเลือกที่เหมาะสมสำหรับบางบริษัทหรือบุคคลที่ต้องการประมวลผลภาพจำนวนมาก ไม่สูงมาก ดังนั้น โครงการนี้จึงได้ถูกจัดทำขึ้นมาเพื่อให้บริการการประมวลผลภาพให้สะดวกสบายและเหมาะสมกับทุกกลุ่มลูกค้า

โครงการร้านค้าสำหรับแอพพลิเคชั่นการประมวลผลภาพซึ่งจัดการงานบนระบบประมวลผลแบบกลุ่ม มีจุดประสงค์เพื่อสร้างระบบให้บริการใช้งานการประมวลผลภาพและแสดงผลงานการงานที่ได้จัดทำขึ้น โดยผู้ใช้สามารถเข้าถึงแอพพลิเคชั่นและเลือกใช้งานตามความต้องการ ได้หลากหลายเช่น การปรับแต่งภาพ, การจัดการกับรูปแบบไฟล์ภาพ และ ซื้อขายน้ำหนักของโมเดล นอกจากนี้ ผู้ใช้งานยังสามารถแสดงผลงานการประมวลผลที่ได้จัดทำขึ้นมา เพื่อให้ผู้ใช้งานอื่นๆ ได้เข้าถึง และ ซื้อขายกันได้ในระบบ โดยทั้งนี้ การใช้งานร้านค้านี้จะเหมาะสมกับกลุ่มผู้ใช้งานที่ต้องการประมวลผลภาพจำนวนมาก ไม่สูงมาก และ ไม่ต้องการเสียค่าใช้จ่ายที่สูง

ร้านค้าสำหรับแอพพลิเคชั่นการประมวลผลภาพซึ่งจัดการงานบนระบบประมวลผลแบบกลุ่ม เป็นโครงการที่มุ่งเน้นการพัฒนาระบบให้มีความสามารถในการจัดการและประมวลผลภาพอย่างมีประสิทธิภาพและเร็วขึ้น โดยใช้ระบบประมวลผลแบบกลุ่มที่มีประสิทธิภาพสูงเป็นพื้นฐาน ซึ่งจะช่วยลดต้นทุนและเพิ่มประสิทธิภาพในการทำงานของระบบให้สูงขึ้น

# **Marketplace for Image Processing Application using Task Management on Cluster Computing System**

Mr. Pasin Chantharathan 63015121

Mr. Sutee Saraphan 63015190

Assoc.Prof.Dr. Orachat Chitsobhuk Advisor

Academic Year 2022

## **ABSTRACT**

Currently, image processing is becoming increasingly important as it is widely used in various industries. However, using cloud processing services, which are expensive, may not be a suitable option for some companies or individuals who do not need to process a large amount of images. Therefore, this project was created to provide image processing services that are convenient and suitable for all customers.

The project Marketplace for Image Processing Application using Task Management on Cluster Computing System. The objective is to create a system that provides image processing services and displays the work performed. Users can access the application and choose the services they need, such as image customization, file format management, and buying and selling of model weights. Additionally, users can showcase their processed images for others to view and buy within the system. This shop is suitable for users who do not need to process a large number of images and do not want to pay high costs.

The Marketplace for Image Processing Application using Task Management on Cluster Computing System is a project that focuses on developing a system that efficiently manages and processes images using a high-performance group-based processing system as the foundation. This system will help reduce costs and increase the efficiency of the system's performance. Users will benefit from the system's fast and accurate image processing capabilities.

## กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สามารถดำเนินการงานประสบความสำเร็จ เนื่องจากได้รับความอนุเคราะห์ จาก รศ.ดร.อรรถตร์ จิตต์ไสภัคตร์ อาจารย์ที่ปรึกษาปริญญานิพนธ์ที่กรุณากำหนด คำปรึกษาเพื่อ ปรับปรุง แก้ไขในข้อผิดพลาด และ เสนอแนวทางการทำงาน รวมไปถึงการให้อุปกรณ์สิ่งของต่างๆ และ ความรู้ แนวทางการศึกษา ที่จำเป็นต่อการดำเนินการศึกษาและค้นคว้าเพื่อจัดทำปริญญานิพนธ์ฉบับนี้ คณะผู้จัดทำขอทราบขอบพระคุณเป็นอย่างไร ณ ที่นี่

ขอขอบพระคุณอาจารย์สรยุทธ กลมกล่อม อาจารย์ผู้ประสานงานที่ให้ข้อมูลและคำปรึกษาแก่ คณะผู้จัดทำปริญญานิพนธ์จนสามารถสำเร็จลงด้วยดี

คณะผู้จัดทำมีความซาบซึ้งในความกรุณาของทุกท่านที่กล่าวถึงและผู้ไม่ได้อ่านมาในที่นี้ที่มี ส่วนช่วยให้ความช่วยเหลือและสนับสนุนด้วยดีตลอดมา จึงขอทราบขอบพระคุณด้วยความจริงใจ

พศิน จันทร์ทัน

สุนิ สาระพันธ์

## สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ .....	IV
สารบัญรูปภาพ.....	VII
สารบัญตาราง.....	XIII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาของปัจจุบัน.....	1
1.2 วัตถุประสงค์ของโครงการ.....	2
1.3 ประโยชน์ที่คาดว่าจะได้รับ .....	2
1.4 ขอบเขตการวิจัย .....	3
บทที่ 2 เอกสารและทฤษฎีที่เกี่ยวข้อง .....	2
2.1 ทฤษฎีที่เกี่ยวข้อง .....	2
2.2 งานที่เกี่ยวข้อง .....	7
2.3 เครื่องมือที่ใช้งานในการพัฒนาระบบ .....	9
2.4 งานวิจัยที่เกี่ยวข้อง.....	12
บทที่ 3 การออกแบบและการพัฒนา.....	5
3.1 รายละเอียดของระบบ.....	5
3.2 ภาพรวมของระบบ .....	17
3.3 แผนภาพผังงาน (Flowchart) .....	19

## สารบัญ(ต่อ)

	หน้า
3.4 แผนภาพยูสเคส (Use Case Diagram) .....	23
3.5 แผนภาพแสดงองค์ประกอบ (Component Diagram) .....	28
3.6 แผนภาพแสดงความสัมพันธ์ของข้อมูล (E-R Diagram) .....	29
3.7 แผนภาพการทำงานแบบลำดับปฏิสัมพันธ์ (Sequence Diagram) .....	33
3.8 การออกแบบส่วนต่อประสานกับผู้ใช้ (User Interface) .....	37
บทที่ 4 ผลการทดลอง / ความก้าวหน้า .....	17
4.1 ส่วนของหน้าลงทะเบียนสมัครเข้าใช้งาน .....	17
4.2 ส่วนของหน้าลงทะเบียนเข้าสู่ระบบใช้งาน .....	48
4.3 ส่วนของหน้าแก้ไขข้อมูลผู้ใช้งาน .....	51
4.4 ส่วนของหน้าจัดการไฟล์ .....	53
4.5 ส่วนของหน้าตั้งงานประมวลผล .....	57
4.6 ส่วนของหน้าตลาดซื้อขาย .....	60
4.7 ส่วนของหน้าประวัติการใช้งานของผู้ใช้ .....	64
4.8 ส่วนของหน้าประวัติการใช้งานของสินค้า .....	65
4.9 ส่วนของหน้าร้านค้าแสดงสินค้า .....	67
4.10 ส่วนของหน้าแสดงข้อมูลเฉพาะสินค้า .....	69
4.11 ส่วนของการแบ่งภาระงาน .....	70
4.12 ส่วนของ Application Image Processing .....	74
บทที่ 5 สรุปผลและข้อเสนอแนะ .....	44
5.1 สรุปผลที่จากโครงการ .....	44

## สารบัญ(ต่อ)

	หน้า
5.2 ปัญหาและอุปสรรค.....	80
5.3 แนวทางการพัฒนาต่อไปในอนาคต .....	80
บรรณานุกรม .....	59
ภาคผนวก ก .....	82
รายละเอียดของ diagram.....	82
ก.1 Component diagram(Detail) .....	82
ภาคผนวก ข .....	89
Test case .....	89
ঃ.1 Frontend test .....	89
ঃ.2 Backend test .....	93

## สารบัญตาราง

ตาราง	หน้า
3.1 รายละเอียดของ Use case Register.....	24
3.2 รายละเอียดของ Use case Login.....	24
3.3 รายละเอียดของ Use case Upload image.....	25
3.4 รายละเอียดของ Use case enchanted image.....	25
3.5 รายละเอียดของ Use case Load balance config.....	25
3.6 รายละเอียดของ Monitoring hardware.....	26
3.7 รายละเอียดของ Using statistics report.....	26
3.8 รายละเอียดของ Subscribe VIP plan.....	27
3.9 รายละเอียดของ Buy weight for model.....	27
3.10 รายละเอียดของ Payment log.....	27
3.11 แสดงรายละเอียดตารางข้อมูลผู้ใช้ (user).....	30
3.12 แสดงรายละเอียดตารางข้อมูลไฟล์รูปภาพ (Image_file).....	30
3.13 แสดงรายละเอียดตารางข้อมูลของไฟล์เดอร์เก็บรูปภาพ (folder_img).....	30
3.14 แสดงรายละเอียดตารางเก็บข้อมูลการเข้าสู่ระบบ (login_log).....	31
3.15 แสดงรายละเอียดตารางข้อมูลการชำระเงิน (payment).....	31
3.16 แสดงรายละเอียดตารางข้อมูลของสินค้า (product).....	31
3.17 แสดงรายละเอียดตารางข้อมูลการสั่งงานประมวลผล (job).....	32
ก.1 UserManagement component.....	83
ก.2 DriveManagement component.....	83
ก.3 Marketplace component.....	83

## สารบัญตาราง(ต่อ)

ตาราง	หน้า
ก.4 History component.....	84
ก.5 ImageProcessingApplicationPage component.....	84
ก.6 user_authentication component.....	85
ก.7 drive_management component.....	85
ก.8 product_management component.....	86
ก.9 user_management component.....	86
ก.10 image_processing_application component.....	86
ก.11 payment_management component.....	87
ก.12 preview_process component.....	87
ก.13 image_processing_process component.....	88
ก.14 load balancing component.....	88
ก.15 Database component.....	89
ก.16 File storage(NAS)component.....	89
ภ.1 Frontend testcases.....	90
ภ.2 Backend testcase.....	94

# สารบัญรูปภาพ

รูป

หน้า

2.1 Cluster Computing.....	5
2.2 การประมวลผลแบบคลุ่มเมฆ (Cloud Computing).....	5
2.3 แผนผังรูปแบบการใช้งานการแบ่งภาระงาน(Load Balance).....	6
2.4 ตัวอย่างของงานประมวลผลภาพบน AnimeFilter.....	7
2.5 ตัวอย่างของ Film Emulator.....	8
2.6 ตัวอย่างของ aws marketplace.....	8
2.7 Component system ของ Vue.js.....	9
2.8 รูปแบบการทำงานของ Kubernetes.....	10
2.9 แผนผังแสดงระบบที่ใช้ในการทดสอบIngress controller.....	12
2.10 รูปคลัพธ์จากการปรับตั้งความคมชัดด้วยวิธีที่แตกต่างกัน.....	13
2.11 แผนผังรูปแบบการกระจายงานในช่วงที่สองที่มีการกระจายภาระระหว่าง CPU และ GPU....	14
2.12 แผนผัง System Architecture ของ ProCon.....	15
3.1 ภาพรวมของระบบ Overview.....	17
3.2 ภาพรวมของระบบ Level 0.....	18
3.3 ภาพรวมของระบบ Level 1.....	18
3.4 แผนภาพผังงานของการสั่งงานประมวลผลภาพ (Web Application).....	19
3.5 แผนภาพผังงานของการสั่งงานประมวลผลภาพบน Marketplace (Web Application).....	20
3.6 แผนภาพผังงานของการจัดเก็บไฟล์ภาพไว้ใน Drive (Web Application).....	21
3.7 แผนภาพผังงานของการใช้งาน Marketplace สำหรับขาย Weight ที่ผู้ใช้ได้ทำการ Train.....	22
3.8 แผนภาพยูสเคส.....	23

## สารบัญรูปภาพ(ต่อ)

รูป	หน้า
3.9 แผนภาพแสดงองค์ประกอบ.....	28
3.10 แผนภาพแสดงความสัมพันธ์ของข้อมูล.....	29
3.11 Sequence diagram ในส่วนของการสั่งงานประมวลผลภาพ.....	33
3.12 Sequence diagram ในส่วนของการสั่งงานประมวลผลภาพบน Marketplace.....	34
3.13 Sequence diagram ในส่วนของการจัดเก็บไฟล์ภาพไว้ใน Drive.....	35
3.14 รูปภาพแสดงการใช้งาน Marketplace สำหรับขาย Weight ที่ผู้ใช้ได้ทำการ Train.....	36
3.15 การออกแบบส่วนลงชื่อเข้าใช้.....	37
3.16 การออกแบบส่วนลงทะเบียนสมัครสมาชิก.....	37
3.17 การออกแบบส่วนหน้าป่าวสาร.....	38
3.18 การออกแบบส่วนพื้นที่เก็บบันทึกรูปภาพ.....	38
3.19 การออกแบบส่วนพื้นที่แสดงรูปภาพที่เก็บบันทึกไว้.....	39
3.20 การออกแบบส่วนแอพพลิเคชั่นประมวลผลภาพ.....	39
3.21 การออกแบบส่วนพื้นที่ซื้อขาย.....	40
3.22 การออกแบบส่วนตั้งค่าบัญชี.....	40
3.23 การออกแบบส่วนตั้งค่าและดูบันทึกการใช้ทรัพยากร.....	41
3.24 การออกแบบส่วนดูบันทึกการใช้งานของผู้ใช้.....	41
3.25 การออกแบบส่วนดูบันทึกการขายของสินค้า.....	42
3.26 การออกแบบส่วนของการแสดงผลตัวอย่างงานประมวลผล Basic App.....	43
3.27 การออกแบบส่วนของการแสดงผลตัวอย่างงานประมวลผล Yolo Model.....	44
3.28 การออกแบบส่วนของการแสดงผลตัวอย่างงานประมวลผล GAN Model.....	45

## สารบัญรูปภาพ(ต่อ)

หัวข้อ	หน้า
3.29 การออกแบบส่วนของการบีบอัดไฟล์ Zip API.....	45
4.1 ข้อมูลที่ส่งจาก Web application เทียบกับข้อมูลที่ส่วน Backend ได้รับ.....	47
4.2 ข้อมูลผู้ใช้ทำการสมัครสำหรับเพิ่มฐานข้อมูล.....	47
4.3 ข้อความที่แสดงหากอีเมลที่ใช้สมัครนั้นมีอยู่แล้วในฐานข้อมูล.....	48
4.4 ข้อมูลที่ส่งจาก Web application เทียบกับข้อมูลที่ส่วน Backend ได้รับ.....	49
4.5 ข้อมูล Cookie ของผู้ใช้หากทำการเข้าสู่ระบบสำเร็จ.....	50
4.6 ข้อความที่แสดงหากหัสผ่านที่กรอกเข้าสู่ระบบนั้นไม่ถูกต้อง.....	50
4.7 ข้อความที่แสดงหากสามารถเปลี่ยนข้อมูลผู้ใช้ได้สำเร็จ.....	51
4.8 ข้อความที่แสดงหากเกิดข้อผิดพลาดขึ้นขณะเปลี่ยนแปลงข้อมูลผู้ใช้.....	52
4.9 ข้อความที่แสดงหากสามารถเปลี่ยนข้อมูลผู้ใช้ได้สำเร็จ.....	53
4.10 ข้อความที่แสดงหากเกิดข้อผิดพลาดขึ้นขณะเปลี่ยนแปลงข้อมูลผู้ใช้.....	53
4.11 ข้อความที่แสดงหากสามารถเพิ่ม Folder สำหรับเก็บรูปภาพได้.....	54
4.12 ข้อความที่แสดงหากเกิดข้อผิดพลาดขณะเพิ่ม Folder สำหรับเก็บรูปภาพ.....	55
4.13 ข้อความที่แสดงหากสามารถเพิ่มรูปภาพสำเร็จ.....	56
4.14 ข้อความที่แสดงหากเกิดข้อผิดพลาดขณะเพิ่มรูปภาพ.....	56
4.15 รูปที่ผ่านการประมวลผลที่แสดงหากประมวลผลสำเร็จ.....	57
4.16 ข้อความที่แสดงหากเกิดข้อผิดพลาดขณะสร้างไฟล์งานประมวลผล.....	58
4.17 ข้อความที่แสดงหากสามารถสร้างไฟล์งานประมวลผลสำเร็จ.....	59
4.18 ข้อความที่แสดงหากเกิดข้อผิดพลาดขณะสร้างไฟล์งานประมวลผล.....	59
4.19 ข้อความที่แสดงหากสามารถเพิ่มลินค์ได้สำเร็จ.....	60

## สารบัญรูปภาพ(ต่อ)

รูป	หน้า
4.20 ข้อความที่แสดงหากไม่สามารถเพิ่มสินค้าได้.....	61
4.21 ข้อความที่แสดงหากสามารถแก้ไขข้อมูลสินค้าได้สำเร็จ.....	62
4.22 ข้อความที่แสดงหากไม่สามารถแก้ไขสินค้าได้.....	62
4.23 ข้อมูลสินค้าที่สามารถส่งมาทดลองใช้งานที่หน้าสั่งงานประมวลผล.....	63
4.24 ข้อความที่แสดงหากไม่สามารถส่งข้อมูลสินค้ามาที่หน้าสั่งงานประมวลผลได้.....	63
4.25 ประวัติรายการของผู้ใช้ที่มีการเรียงลำดับโดยการแสดงรายการล่าสุดก่อน.....	64
4.26 ประวัติรายการของผู้ใช้ที่มีการเรียงลำดับโดยการแสดงรายการเก่าที่สุดก่อน.....	65
4.27 ประวัติรายการของผู้ใช้ที่มีค้นหาโดยใช้วันที่ที่กำหนดเอาไว้.....	65
4.28 ประวัติรายการของสินค้าที่มีการเรียงลำดับโดยการแสดงรายการล่าสุดก่อน.....	66
4.29 ประวัติรายการของสินค้าที่มีการเรียงลำดับโดยการแสดงรายการเก่าที่สุดก่อน.....	66
4.30 ประวัติรายการของสินค้าที่มีค้นหาโดยใช้วันที่ที่กำหนดเอาไว้.....	67
4.31 รายการของสินค้าที่มีการเรียงลำดับโดยการแสดงสินค้าที่มีการแก้ไขล่าสุด.....	68
4.32 รายการของสินค้าที่มีการเรียงลำดับโดยการแสดงสินค้าที่มีการแก้ไขเก่าสุด.....	68
4.33 รายการของสินค้าที่มีการกรองเลือกเฉพาะประเภท Object detections.....	68
4.34 แสดงรายละเอียดของสินค้าที่ถูกบันทึกเอาไว้ในฐานข้อมูล.....	69
4.35 แสดงข้อความแจ้งเตือนหากทำหางานสินค้าที่ไม่มีอยู่ในฐานข้อมูล.....	70
4.36 ผลการทดลองสั่ง Request !ข้ามไปชั่งน.....	71
4.37 กราฟของผลการทดลองเมื่อสั่งงานประมวลผลด้วยการประมวลผลด้วยเครื่องเดียว.....	72
4.38 กราฟของผลการทดลองเมื่อสั่งงานประมวลผลด้วยระบบคลัสเตอร์ด้วย Round Robin.....	72
4.39 กราฟของผลการทดลองเมื่อสั่งงานประมวลผลด้วยระบบคลัสเตอร์ด้วย Least Connection.....	73

## สารบัญรูปภาพ(ต่อ)

รูป	หน้า
4.40 Repositories ของ DockerHub ที่ทำการเก็บ Application ไว้.....	74
4.41 รูปภาพต้นฉบับ และ รูปภาพที่ผ่านการประมวลผลด้วย Application ASCII.....	75
4.42 รูปภาพต้นฉบับ และ รูปภาพที่ผ่านการประมวลผลด้วย Application BlackWhite.....	75
4.43 รูปภาพต้นฉบับ และ รูปภาพที่ผ่านการประมวลผลด้วย Application PixelArt.....	76
4.44 รูปภาพต้นฉบับ และ รูปภาพที่ผ่านการประมวลผลด้วย Application Mosaic.....	76
ก.1 แผนภาพองค์ประกอบส่วน Frontend.....	82
ก.2 แผนภาพองค์ประกอบส่วน Backend.....	85
ก.3 แผนภาพองค์ประกอบส่วน Server.....	87
ก.4 แผนภาพองค์ประกอบส่วน Storage.....	88

## บทที่ 1

### บทนำ

#### 1.1 ความเป็นมาของปัญหา

ในปัจจุบันผู้ให้บริการแอพพลิเคชั่นการประมวลผลภาพนั้นมีข้อจำกัดเรื่องงานการประมวลผล ใช้ทรัพยากรของเครื่องสูง ไม่มีประสิทธิภาพ หรือ ยังไม่มีการกำหนด จำกัดหน่วยประมวลผลของงานในแต่ละงานประมวลผล และ ไม่รองรับการประมวลผลภาพจำนวนมากพร้อม ๆ กัน พร้อมด้วยแอพพลิเคชั่นการประมวลผลภาพนั้นมีตัวเลือกค่อนข้างน้อยในแอพพลิเคชั่นเดียวให้ใช้งาน และเนื่องจากงานวิจัยด้าน Image Processing ต่าง ๆ นั้นมีการ Train ข้อมูลของ Weight Model เข้ามาช่วยในงานประมวลผลดั้งตัวอย่างในวิชา Image Processing ภายในการ Train ที่มีการให้นักศึกษา ศึกษาโครงงานเรื่องการ Train Weight Model ภายในวิชาขึ้นมาเพื่อมาทดลองภายนอกห้องทดลอง แต่ยังขาดพื้นที่สำหรับการให้บริการด้าน Model ของ Image Processing ที่นักศึกษาได้ทำการ Train Weight ข้อมูลไว้เพื่อให้ได้ทดสอบภายนอกห้องทดลองเนื่องจากใช้ทรัพยากรของเครื่องสูงในการประมวลผลของ Model และ ให้บุคคลภายนอกได้เข้าถึงด้วย

ดังนั้น โครงงาน ร้านค้าสำหรับแอพพลิเคชั่นการประมวลผลภาพซึ่งจัดการงานบนระบบประมวลผลแบบกลุ่ม มีจุดประสงค์เพื่อสร้างระบบการแบ่งการจัดลำดับงานประมวลผลของผู้ใช้บริการแอพพลิเคชั่น เพื่อเป็นอีกทาง เลือกหนึ่งในการใช้งานการประมวลผลภาพในรูปแบบต่าง ๆ อย่างมีประสิทธิภาพ และรองรับการประมวลผลภาพจำนวนมาก พร้อมด้วยรองรับแอพพลิเคชั่นที่สามารถนำเข้ามาเพิ่มในอนาคตได้อีกด้วย และ ในส่วนของ Weight Model ที่นักศึกษาได้ทำการ Train มาแล้วสามารถนำเข้าภายนอก แอพพลิเคชั่นเพื่อเปิดให้บริการในแอพพลิเคชั่นของเราระหว่างทำการซื้อขาย หรือ เป็นที่แสดงผลงานของนักศึกษา และ เป็นอีกทางเลือกให้นักศึกษามีรายได้

โดยสรุปได้ว่า การตกแต่งรูปภาพ หรือ การปรับแต่งภาพ นั้นมีการใช้อยู่แพร่หลาย และ คนส่วนใหญ่มีการใช้งานเรื่องการประมวลผลภาพกันเป็นปกติ การนำ Weight Model มาใช้งาน หรือ จัดแสดงผลงาน และ การซื้อขาย ซึ่งทั้งหมดที่กล่าวมาคือส่วนมากใช้งานบนแอพพลิเคชั่นต่างๆ ผ่านอินเทอร์เน็ตเราจึงพัฒนาโครงงานบนพื้นฐานของเว็บแอพพลิเคชั่นซึ่งสอดคล้อง และ รองรับกับปัญหาดังกล่าว โครงงาน “ร้านค้าสำหรับแอพพลิเคชั่นการประมวลผลภาพซึ่งจัดการงานบนระบบประมวลผลแบบกลุ่ม” นี้ถูกจัดทำขึ้นมาเพื่อแก้ไขในส่วนของกระบวนการนี้ โดยมีการแบ่งเป็น 3 ส่วนคือ

- 1) ส่วนการประมวลผลงานแบบการจัดลำดับงาน (Server) Task Scheduling การแบ่งจัดลำดับการทำงานประมวลผลแต่ละงานไปแต่ละเครื่อง โดยมีการทำหน้าที่รับคำขอที่ใช้ในแต่ละงานประมวลผลเพื่อให้ได้ประสิทธิภาพเพียงพอต่อความต้องการของงานประมวลผล และ ใช้ทรัพยากรของหน่วยประมวลผลให้คุ้มค่ากับทรัพยากรที่มีอยู่ของหน่วยประมวลผล
- 2) ส่วนแอปพลิเคชันการประมวลผลภาพ (User) Image Processing Application ส่วนระบบสำหรับประมวลผลภาพที่ระบบได้มีการเตรียมรูปแบบประมวลผลภาพต่าง ๆ ที่ หลากหลายให้ผู้ใช้ได้เข้ามาใช้งานการประมวลผลภาพจำนวนมากที่ทางเราได้เตรียมจัดทำไว้
- 3) ส่วนแอปพลิเคชันให้บริการซื้อขาย Weight ไมเดล (User) Marketplace ส่วนระบบสำหรับให้ผู้ใช้มีการซื้อ Weight Model และ นำเข้า Weight Model ที่ได้ทำการเทรนนิ่งไว้แล้วมาเปิดให้ใช้บริการบนแอปพลิเคชันเพื่อทำการ ขาย หรือ เพื่อทดลองงานต่าง ๆ โดยเป็นจุดแสดงผลงานของตัว Weight Model ที่ได้ทำการเทรนนิ่งไว้ได้อีกด้วย

## 1.2 วัตถุประสงค์ของโครงการ

- 1) เพื่อจัดสรรการใช้งานประมวลผลของแต่ละงานประมวลผลให้มีประสิทธิภาพเพียงพอ ต่อความต้องการของงาน
- 2) เพื่อให้งานในการประมวลผลภาพนั้นมีการจัดลำดับการประมวลผล และ มีประสิทธิภาพมากยิ่งขึ้นต่อทรัพยากรของเครื่องประมวลผล
- 3) เพื่อนำไปประยุกต์ต่อขอบทางธุรกิจ

## 1.3 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ได้รับประสบการณ์ในการพัฒนาเว็บด้วย Vue.js และ DJANGO
- 2) ได้รับประสบการณ์ในการออกแบบฐานข้อมูลด้วย MongoDB (NOSQL)
- 3) ได้รับประสบการณ์ในการพัฒนาระบบ Cluster Computing ด้วย Kubernetes
- 4) ได้รับประสบการณ์ในการพัฒนาระบบ Server สำหรับงาน Cluster Computing
- 5) ได้รับประสบการณ์ในการพัฒนาแอปพลิเคชันการประมวลผลภาพที่นิยมในปัจจุบัน

## 1.4 ขอบเขตการวิจัย

- 1) ระบบ Cluster Computing เชื่อมต่อผ่านระบบเครือข่ายท้องถิ่น (LAN)
- 2) ระบบ Cluster Computing ในแอพพลิเคชันการประมวลผลภาพด้วยการจัดการลำดับงานบนระบบเครือข่ายนี้นั้นเป็นการแบ่งงานในแต่ละเครื่องเพื่อให้ใช้ทรัพยากรที่จำกัด และ จำกัด ทรัพยากรในแต่ละงานประมวลผล
- 3) การจัดการลำดับงานบนระบบขึ้นอยู่กับประสิทธิภาพชาร์ดแวร์ของ Node ที่พร้อมใช้งานในช่วงเวลาหนึ่ง
- 4) แอพพลิเคชันในส่วนของฝากรายลักษณะ Structures ของ Directory และ User นั้นการแตกระดับขั้นของ Directory ได้ 1 ขั้น
- 5) แอพพลิเคชันในส่วนของการประมวลผลนั้นสามารถทำการประมวลผลภาพจำนวนมาก ซึ่งปรับค่า Parameter ได้ 1 ค่าในงานประมวลผลภาพจำนวนมาก
- 6) ส่วนของ Marketplace ในการนำเข้า Weight นั้นจะต้องรองรับกับตัว Model ที่มีให้บริการภายใต้แอพพลิเคชันเท่านั้น

## บทที่ 2

# เอกสารและทฤษฎีที่เกี่ยวข้อง

ในการจัดทำโครงการ ผู้จัดทำได้ทำการศึกษาทฤษฎีที่เกี่ยวข้อง, งานที่เกี่ยวข้อง, เครื่องมือที่ใช้ งานในการพัฒนาระบบ และ งานวิจัยที่เกี่ยวข้อง เพื่อนำมาปรับใช้ และ เป็นแนวทางในการทำโครงการ ดังต่อไปนี้

## 2.1 ทฤษฎีที่เกี่ยวข้อง

### 2.1.1 Digital Image Processing

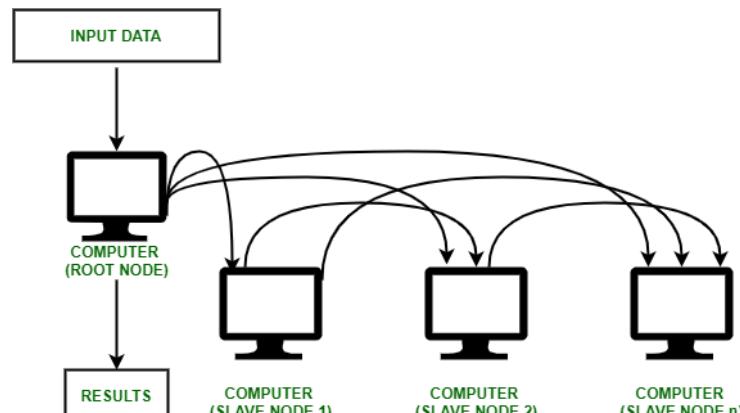
การประมวลผลภาพดิจิทัล หรือ Digital Image Processing คือ การใช้คอมพิวเตอร์ดิจิทัลในการประมวลผลภาพดิจิทัลผ่านอัลกอริทึม ภาพในที่นี่รวมความหมายถึงสัญญาณในระบบดิจิทัล 2 มิติ ภาพนิ่ง และ ภาพเคลื่อนไหว หรือเป็นชุดของภาพนิ่ง ที่เรียกว่า เฟรม (Frame) ซึ่งนับเป็นภาพดิจิทัล 3 มิติ ได้ เช่น ภาพ ทางการแพทย์ หรือ ภาพ 3 มิติหลายชนิด (Multimodal image)

วัตถุประสงค์ของการประมวลผลภาพแบ่งได้ออกเป็น 5 กลุ่มหลักดังนี้

- 1) การสร้างภาพ (Visualization) จากวัตถุที่มองด้วยตาไม่เห็น สามารถทำให้มองเห็นภาพได้
- 2) การปรับความชัด และ การฟื้นฟูภาพ (Image sharpening and Restoration) ใช้สำหรับการปรับความละเอียดของภาพให้ดีขึ้น
- 3) การค้นภาพ (Image retrieval) ทำให้จุดที่น่าสนใจของภาพให้เด่นชัด
- 4) การวัดรูปแบบของภาพ (Measurement of pattern) วัดในภาพทั้งหมดนั้นจะถูกวัดค่ามาเพื่อหารูปแบบของวัตถุภายในภาพ
- 5) การรับจดจำภาพ (Image Recognition) สามารถแยกแยะวัตถุในภาพได้

### 2.1.2 Cluster Computing

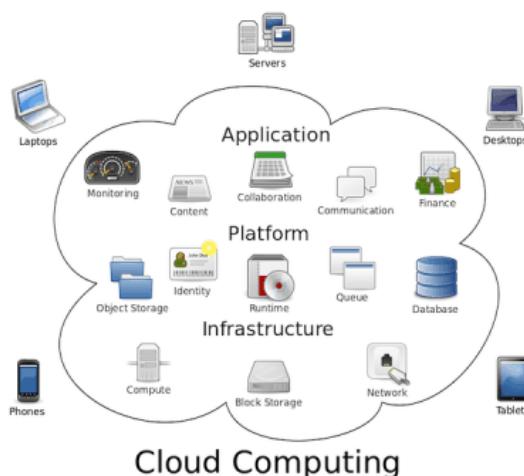
ระบบคลัสเตอร์ หรือ คลัสเตอร์инг เป็นการเชื่อมต่อระบบการทำงานของกลุ่มคอมพิวเตอร์เข้าด้วยกันภายใต้ระบบเครือข่าย ทำให้สามารถกระจายงานประมวลผลที่ทำไปยังเครื่องอื่นๆ ภายในระบบเพื่อให้การประมวลผลใช้ทรัพยากรที่มีอยู่ได้อย่างคุ้มค่าและมีประสิทธิภาพการประมวลผลที่สูงขึ้น โดยถูกคิดขึ้นขึ้นมาเพื่อใช้ในการประมวลผลงานที่มีความซับซ้อนสูงโดยเฉพาะงานด้านวิทยาศาสตร์ เช่น การจำลองโครงสร้างของโมเลกุลทางเคมี, การวิเคราะห์เกี่ยวกับตำแหน่งการเกิดพายุสุริยะ, การประมวลผลภาพ เป็นต้น ดังรูป 2.1



รูป 2.1 Cluster Computing

### 2.1.3 การประมวลผลแบบคลุ่มเมฆ (Cloud Computing)

การประมวลผลแบบคลุ่มเมฆ เป็นการที่ ผู้ให้บริการนั้นมีทรัพยากรที่ใช้สำหรับการประมวลผลอยู่ โดยเมื่อมีผู้ใช้บริการเข้ามาร้องขอต้องการใช้บริการก็จะทำการเปิดให้ผู้เข้าใช้บริการนั้นเข้าถึงทรัพยากรของผู้ให้บริการผ่านระบบอินเตอร์เน็ตทำให้ผู้ใช้บริการไม่จำเป็นต้องเป็นเจ้าของทรัพยากรจริงซึ่งทำให้ผู้ใช้บริการไม่จำเป็นต้องติดตั้งหรือตั้งค่า และ รักษาทำงานเบื้องหลัง โดยผู้ใช้ที่จะสามารถเลือกได้ว่าต้องการใช้ทรัพยากรมากน้อยเท่าไหร่ เพื่อให้เหมาะสมกับงานที่ต้องการประมวลผลมากที่สุด และสามารถเข้าถึงทรัพยากรจากที่ไหนก็ได้ และ ซึ่งโดยส่วนใหญ่จะมีการคิดค่าใช้จ่ายในลักษณะเข้าเป็นรายเดือน หรือ รายปี ขึ้นอยู่กับแต่ละผู้ให้บริการ โดยผู้ให้บริการก็มีอยู่หลากหลายให้เลือกใช้บริการ เช่น Google Cloud Platform, Microsoft azure เป็นต้น ซึ่งรูปแบบหรือแผนผังด้านล่างการประมวลผลแบบคลุ่มเมฆจะแสดงในรูปที่ 2.2



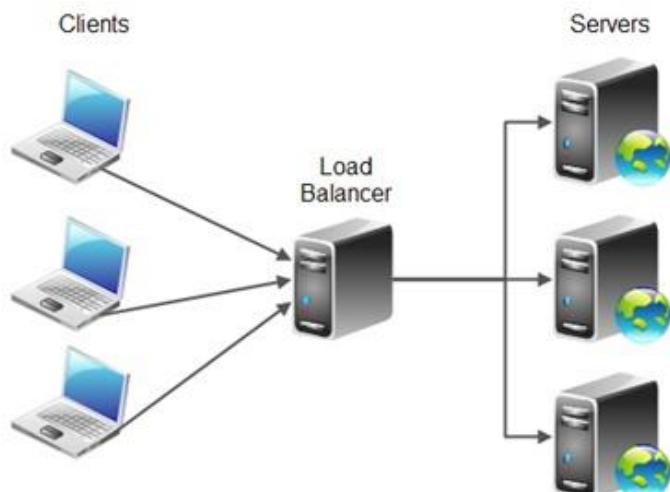
รูป 2.2 การประมวลผลแบบคลุ่มเมฆ (Cloud Computing)

#### 2.1.4 การแบ่งภาระงาน (Load Balance)

การแบ่งภาระงาน หรือ Load Balance คือเทคนิควิธีการทำงานด้าน Network ที่ลูกคิดกันขึ้นมาเพื่อช่วยลดการเกิดปัญหา Server ไม่สามารถรับงานประมวลผลจำนวนมากได้จน Overload และหยุดการทำงานไป โดยวิธีการคือการนำเครื่อง server ที่สามารถทำงานได้ในระดับเดียวกัน และ สามารถทำงานแบบเดียวกัน มาทำงานร่วมกันเพื่อช่วยกระจายปริมาณงานที่เข้ามา โดยจะแบ่งงานไปตามวิธีการที่ผู้ใช้ได้กำหนดเอาไว้ ซึ่งวิธีการแบ่งงานที่นิยมใช้ก็มีต่อไป ดังนี้

1. Round Robin เป็นการแบ่งงานให้กับ Server ในกลุ่มแบบเรียงลำดับ เช่น 1,2,3,1,2,...
2. Sticky เป็นการแบ่งงานให้กับ Server โดยอิงจาก Sessions ที่ผู้ใช้ เคยเข้าไปใช้ก่อนหน้า
3. Workload เป็นการแบ่งงานให้กับ Server โดยการตัดสินจากประสิทธิภาพของ Server ในกลุ่ม

ทั้งนี้การเลือกรูปแบบการแบ่งภาระงานนั้นขึ้นอยู่กับความต้องการของผู้ใช้ที่มีต่อระบบหรืองานนั้นๆ โดยรูปที่ 2.3 จะแสดงถึงแผนผังรูปแบบการใช้งานการแบ่งภาระงาน



รูป 2.3 แผนผังรูปแบบการใช้งานการแบ่งภาระงาน(Load Balance)

## 2.2 งานที่เกี่ยวข้อง

### 2.2.1 Image Processing Application

เป็นบริการประมวลผลภาพที่อยู่ภายใน Application ไม่ว่าจะเป็นการตกแต่งรูปภาพ การเปลี่ยนสีภาพ หรือ การปรับแต่งขนาดรูปภาพ ก็ถือว่าเป็นการทำงานของ Image Processing ที่อยู่ภายในผู้ให้บริการ Application นั้น ๆ ว่ามีการให้บริการอย่างไร

### 2.2.2 AnimeFilter

AnimeFilter หรือ AnimeFilter.com เป็นผู้ให้บริการ Application การประมวลผลภาพโดยการให้ผู้ใช้งานนำเข้าภาพของตนเองที่ต้องการเพื่อให้ Application นำภาพนั้นไปประมวลผลโดยมีการประมวลผลดังนี้ จับวัตถุของภาพที่ต้องการนำมาแปลงภาพให้อยู่ในรูปแบบ Model ที่ทางผู้ให้บริการได้เปิดให้ผู้ใช้งานเข้ามาใช้บริการแสดงดังรูปที่ 2.4



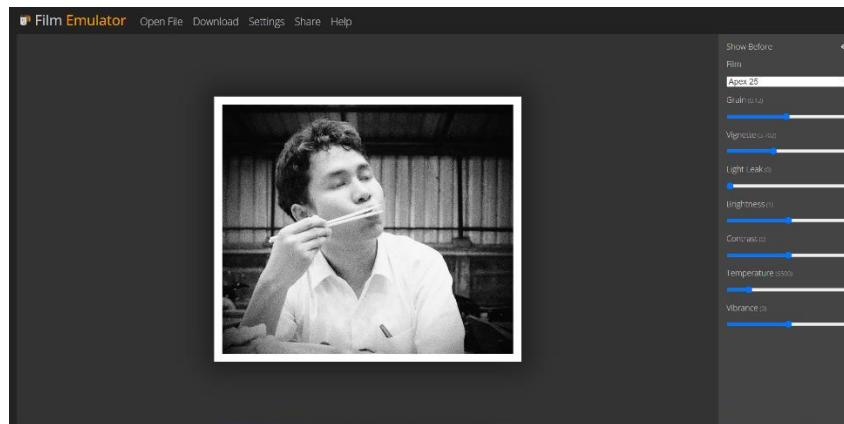
รูป 2.4 ตัวอย่างของงานประมวลผลภาพบน AnimeFilter

สิ่งที่ AnimeFilter ยังไม่มี หรือ นำมาปรับปรุง มาพัฒนาต่อในแอพพลิเคชันการประมวลผลภาพด้วยการจัดการลำดับงานบนระบบเครือข่าย

1. การนำเข้า Weight Model ที่ผู้ใช้ได้ Train มาเพื่อเปลี่ยน Filter เป็นแบบฉบับที่ผู้ใช้ได้ทำการ Train Weight ที่ต้องการมาใช้งาน
2. ให้บริการ Filter ที่หลากหลายแบบจากที่ผู้ใช้นำเข้ามาให้บริการใน Marketplace

### 2.2.2 Film Emulator

Film Emulator หรือ <https://29a.ch/film-emulator> เป็นผู้ให้บริการ Application การประมวลผลภาพโดยการให้ผู้ใช้งานนำเข้าภาพของตนเองที่ต้องการเพื่อให้ Application นำภาพนั้นไปประมวลผลโดยมีการประมวลผลดังนี้ ปรับแต่งสีของภาพตามที่ผู้ใช้ต้องการ โดยมี parameter ที่กำหนดมาให้โดยมี Preset จากกล้องฟิล์มให้ผู้ใช้งานเข้ามาใช้บริการ ตัวอย่างแสดงดังรูปที่ 2.5



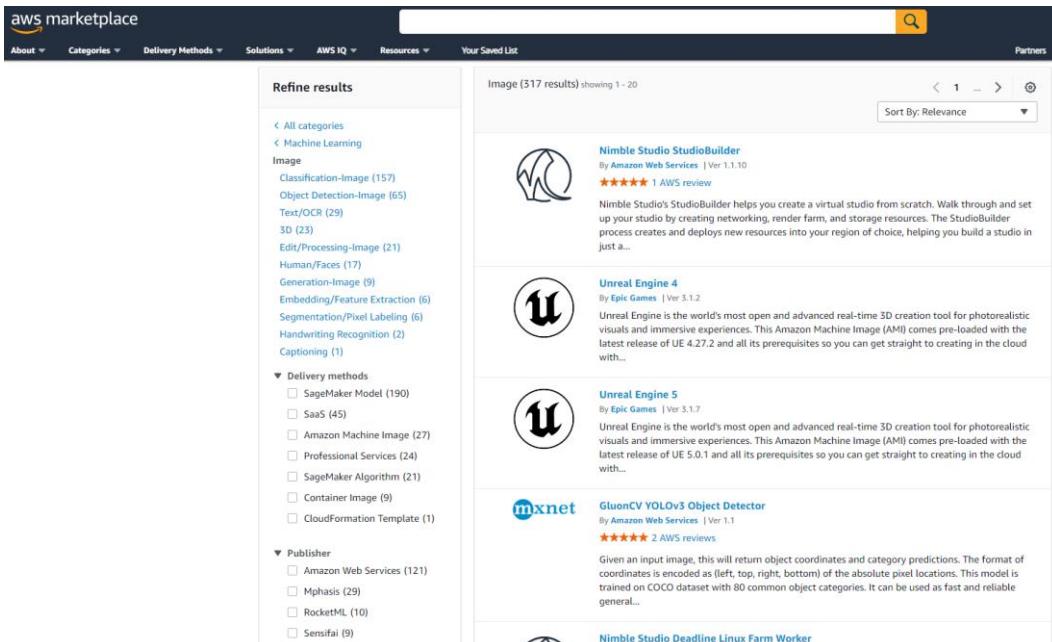
รูป 2.5 ตัวอย่างของ Film Emulator

สิ่งที่ Film Emulator ยังไม่มี หรือ นำมาปรับปรุง มาพัฒนาต่อในแอพพลิเคชั่นการประมวลผลภาพด้วยการจัดการลำดับงานบนระบบเครือข่าย

1. การรองรับการสั่งงานประมวลผลภาพจำนวนนัก
2. การรองรับการประมวลผลภาพจำนวนมากในการปรับค่า Parameter เดียว กัน
3. มี Application ใน การปรับแต่งภาพหลากหลายแบบมากกว่า

### 2.2.3 aws marketplace

aws marketplace หรือ [aws.amazon.com/marketplace/](https://aws.amazon.com/marketplace/) เป็นผู้ให้บริการระบบ Cloud ที่ให้ผู้ใช้เลือกใช้บริการ Model ต่าง ๆ ที่ทาง aws รองรับโดยให้ผู้ใช้จ่ายค่าบริการเป็นรายชั่วโมงเพื่อเข้าใช้บริการงานต่าง ๆ เช่น YOLO, Unreal Engine 4 และ อื่น ๆ อีกมากมาย ดังรูป 2.6



รูป 2.6 ตัวอย่างของ aws marketplace

สิ่งที่ aws marketplace ยังไม่มี หรือ นำมาปรับปรุง มาพัฒนาต่อในแอพพลิเคชั่นการประมวลผลภาพด้วยการจัดการลำดับงานบนระบบเครือข่าย

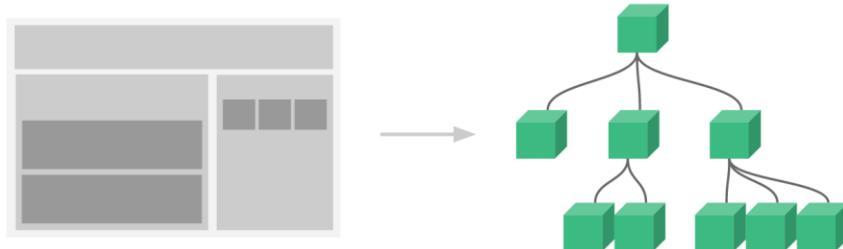
1. การให้บริการเป็นจำนวนภาพที่ประมวลผลแทนการใช้บริการรายชั่วโมง
2. ให้ผู้ใช้รายย่อยอื่น ๆ นำเข้า Weight เพื่อมาซื้อขายได้จาก Model ที่ Application มีให้เหมือนกัน

## 2.3 เครื่องมือที่ใช้งานในการพัฒนาระบบ

### 2.3.1 Vue.js

Vue.js เป็น JavaScript Framework ที่ใช้สำหรับการพัฒนาหน้าจอแสดงผลผู้ใช้ (User Interface) เป็นหลัก โดยที่ library หลักของ Vue.js นั้นมุ่งเน้นไปที่การสร้าง View layer ของ Web application และ Vue.js ยังสามารถสร้าง Web application แบบ Single-Page Application ที่ไม่จำเป็นต้องมีการโหลดเมื่อสลับเปลี่ยนหน้าของ Web application

Vue.js มีส่วนสำคัญอย่างหนึ่งนั้นคือ component system ที่ช่วยให้เราสามารถสร้าง Web application ขนาดใหญ่ที่ประกอบไปด้วย components ขนาดเล็ก และสามารถนำ component เหล่านี้มาใช้งานซ้ำได้อีกด้วย โดยตัวอย่างการทำงานของ Component stem ของ Vue.js แสดงดังรูปที่ 2.7



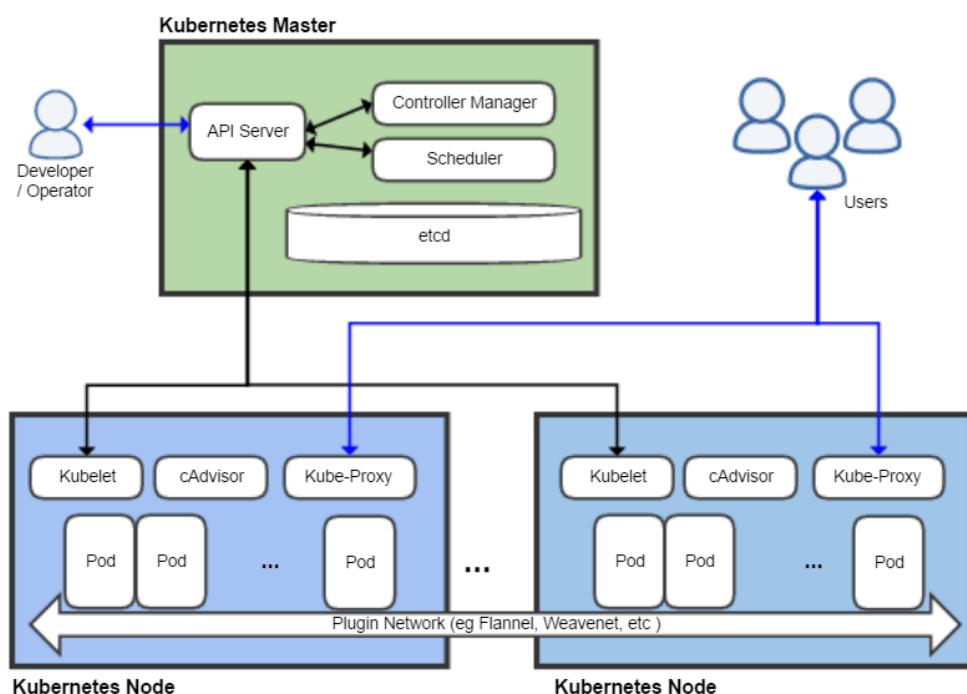
รูป 2.7 Component system ของ Vue.js

### 2.3.2 Docker

Docker คือชุด Platform ที่ใช้การจำลองเสมือนระดับ OS เพื่อคำนึงการทำงานส่วนแพ็คเกจที่เรียกว่าคอนเทนเนอร์ (Container) ซึ่งในแต่ละคอนเทนเนอร์แยกจากกันภายในแต่ละ คอนเทนเนอร์จะประกอบด้วยซอฟต์แวร์ ไลบรารี และ ไฟล์กำหนดค่าของตนเอง (Environment) อิกทึ้งในแต่ละ Container ก็สามารถสร้างช่องทางสื่อสารกันภายใน Container เพื่อให้ช่วยในการประมวลผลโดยมีการใช้ทรัพยากริบบิ้นอย เมื่อเทียบเท่ากับการใช้งานบน OS ปกติทั่วไป และ ลดปัญหาในการเกิดข้อผิดพลาดของ Environment ในการใช้งาน

### 2.3.3 Kubernetes

คือ ถูกพัฒนาโดย Google เพื่อนำมา จัดการระบบคลัสเตอร์ (Cluster management software) สำหรับ Docker Container โดยตัว Kubernetes นั้นใช้สำหรับการ จัดการปล่อยซอฟต์แวร์อัตโนมัติ (automating software deployment) และ ทำการ Scaling ตรวจสอบความถูกต้องของ Container บนระบบ Cluster ได้ตลอดเวลา ตัว Kubernetes นั้นเนื้องจากเป็น Open-Source จึงมีการนำไปดัดแปลงแก้ไขเป็น Kubernetes เวอร์ชันอื่น ๆ มาอีกมากmany ไม่ว่าจะเป็น MicroK8S, K3D, MiniKube โดยทั้งหมดที่กล่าวมา มีเงื่อนไข และ การทำงาน การใช้งานอยู่ในรูปแบบเดียวกัน โดยมีหลักการทำงานพื้นฐานเหมือนกันดัง



รูป 2.8 รูปแบบการทำงานของ Kubernetes

### 2.3.4 Python

Python เป็นภาษาในการเขียนโปรแกรมที่ใช้อย่างแพร่หลายในการพัฒนาซอฟต์แวร์ เว็บ และแอ��เพล็กชัน วิทยาศาสตร์ข้อมูล การประมวลผลภาพ และ แมชชีนเลร์นинг (ML) มีการเรียนรู้ง่าย และสามารถทำงานบนแพลตฟอร์มได้มากมาย ไม่ว่าจะเป็นบน Window, MacOS, Linux ทั้งนี้ Python เป็นซอฟต์แวร์ที่สามารถดาวน์โหลด และ ใช้งานได้ฟรี

### 2.3.5 Django

Django เป็น Framework ที่ถูกเขียนด้วยภาษา Python สำหรับพัฒนาเว็บแอปพลิเคชัน โดยที่สามารถพัฒนาได้อย่างรวดเร็ว และ มีประสิทธิภาพ เหมาะกับการพัฒนาเว็บที่มีการใช้งานบ่อย เช่น การตรวจสอบการดึงข้อมูลจากฐานข้อมูล การจัดการคุกคัก และ การทำงานควบคู่ไปกับ Python ซอฟต์แวร์ อื่น ๆ มาปรับให้ใช้งานภายในเว็บแอปพลิเคชันได้ ตัว Django เป็น open-source ที่สามารถนำมาใช้งานได้ฟรี และ พัฒนาได้ฟรี โดยมี Community ที่ช่วยดูแลรักษาซอฟต์แวร์ตัวนี้ และ มีองค์กรไม่แสวงหาผลกำไรซึ่งว่า Django Software Foundation ที่ทำการพัฒนา ปรับปรุง และ บำรุงรักษา ตัว Django ให้มีคุณภาพสูง และ ฟีเจอร์อีกมากมายให้ใช้งานโดยไม่มีค่าใช้จ่ายตลอดมา

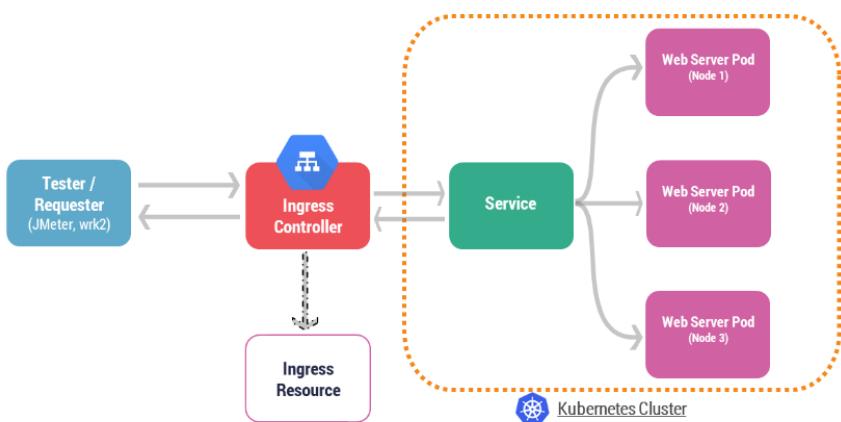
### 2.3.6 MongoDB

MongoDB เป็น NoSQL Database ที่สามารถใช้งานข้าม Platform ต่าง ๆ ได้โดยตัว MongoDB ได้ใช้ NoSQL เป็นการเข้าถึงข้อมูลโดยใช้เทคนิคของการดึงข้อมูลผ่าน Key Pair Value โดยหมายเหตุกับงานที่ต้องการเข้าถึงข้อมูลได้อย่างรวดเร็ว และ เหมาะการทำงานงาน Big Data ตัวของ MongoDB สามารถที่จะสร้างเป็น Cluster เพื่อตอบสนองความต้องการเป็น High Availability (HA) ได้ และ สามารถทำการ Auto Scale ไม่ว่าจะมีการใช้งานมาก หรือ น้อยแค่ไหนก็สามารถปรับแต่ง Environment นั้น ๆ ได้ให้เข้ากับการใช้งาน

## 2.4 งานวิจัยที่เกี่ยวข้อง

### 2.4.1 Performance evaluation and comparison of ingress controllers on Kubernetes cluster

งานวิจัยนี้จัดทำในปี พ.ศ. 2561 โดย คุณอาชิป พวงลำไย และ คุณชัยพร เบนະภาตะพันธ์ กล่าวถึงเรื่องการทดลองเพื่อประเมินสมรรถนะ และเปรียบเทียบประสิทธิภาพของระบบ Load balance แบบ Layer – 7 ที่ถูกใช้งานในสภาพแวดล้อมการทำงานที่อยู่ในรูปแบบของ Container cluster ด้วย Kubernetes โดยแผนผังการวางระบบในการวิจัยจะเป็นดังรูป 2.9



รูป 2.9 แผนผังแสดงระบบที่ใช้ในการทดสอบIngress controller

โดยโครงงานนี้ได้นำเอกสารนี้มาศึกษาเพื่อใช้ในการพัฒนาส่วน Cluster Computing เพื่อให้การจัดลำดับงานและการประมวลผลใน Cluster มีประสิทธิภาพมากขึ้น

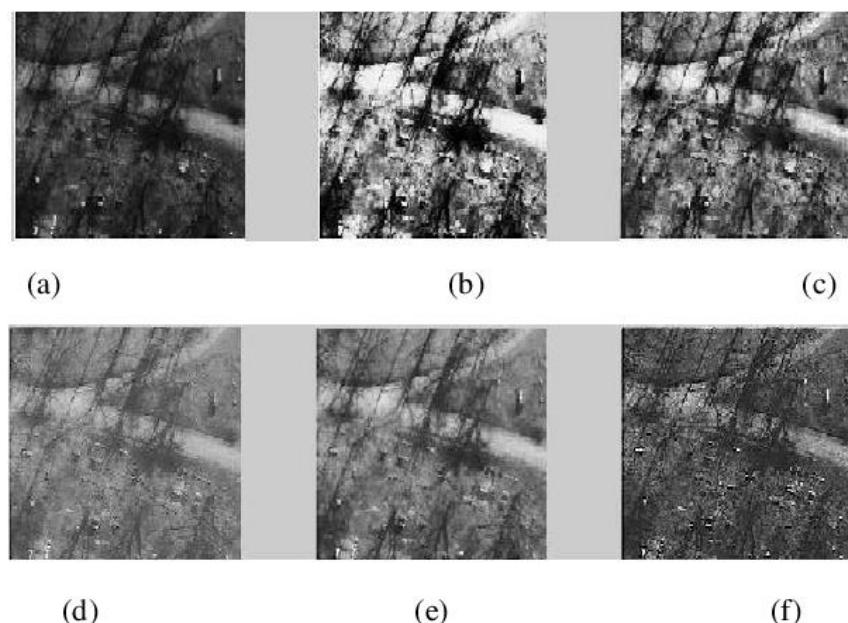
### 2.4.2 A modified gray-level difference algorithm for analyzing Gaussian Blurred texture images

งานวิจัยนี้จัดทำขึ้นในปี 2011 โดย Rui Zhang, Xiang Qian และ Datian Ye กล่าวถึงเรื่องการกระบวนการแยกแยะพื้นผิวของรูปภาพโดยการใช้อัลกอริทึม gray-level difference (GLD) โดยจะมี 4 ขั้นตอนคือ 1. ประเมินความแตกต่างของพื้นผิวภาพโดยเปรียบเทียบกับข้อมูลที่เตรียมเอาไว้ก่อนหน้า 2. ใช้ตัวกรองแบบ Wiener เพื่อลบ Gaussian blur noise ที่ที่อาจเกิดขึ้นหากค่าความแตกต่างมีน้อย 3. ทำซ้ำวิธีการที่ 2. ไปจนกระทั่งค่าความแตกต่างสูงกว่าข้อมูลที่เตรียมเอาไว้ 4. ใช้อัลกอริทึม gray-level difference แบบดั้งเดิม เพื่อเก็บข้อมูลพื้นผิวจากรูปภาพที่ผ่านการประมวลผลแล้ว เพื่อนำมาวัดประสิทธิภาพเทียบกับอัลกอริทึม gray-level difference แบบใหม่

โดยโครงงานนี้ได้นำเอกสารนี้มาเพื่อศึกษาใช้ในการพัฒนาส่วนของการพัฒนา Image processing application เพื่อสร้าง Remove background หรือ ลบพื้นหลัง เพื่อมาเป็น 1 ใน Image processing application ที่จะอยู่ใน Web application

#### 2.4.3 Contrast Stretching Enhancement in Remote Sensing Image

งานวิจัยนี้จัดทำขึ้นในปี 2011 โดย Salem Saleh Al-amri กล่าวถึงเรื่องการปรับแต่งรูปภาพที่ถูกถ่ายมาจากการที่ยมในส่วนของความคมชัดของภาพด้วยเทคนิค Local contrast stretching (LCS) โดยจะทำการ sliding window ผ่านรูปภาพแล้วปรับความคมชัดของภาพไปทีละส่วน โดยสามารถปรับให้ความคมชัดของตัวรูปภาพนั้นเพิ่มขึ้นและลดลงได้ตามความต้องการ และนอกจากนี้ก็ยังนำเสนอวิธีการอื่นที่ใช้ในการปรับแต่งความคมชัด เช่น the bi-histogram equalization (BHE) และ Global histogram equalization เป็นต้น โดยรูปที่ 2.10 จะเป็นรูปผลลัพธ์จากการปรับต่างด้วยวิธีที่แตกต่างกัน



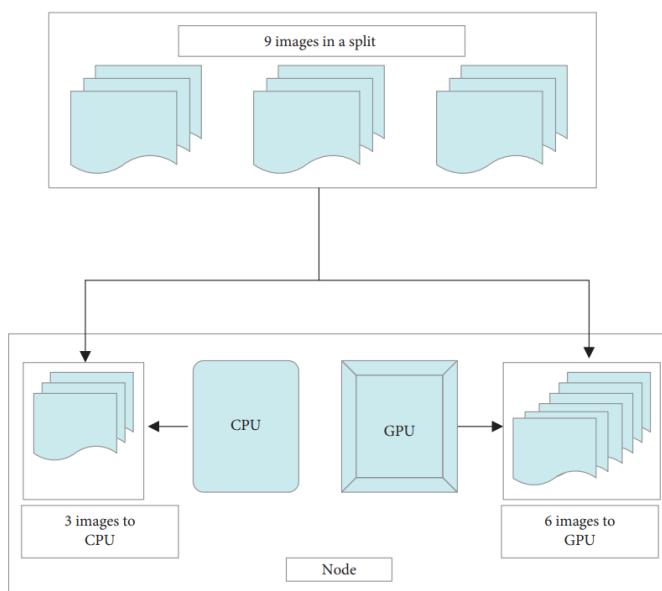
รูป 2.10 รูปผลลัพธ์จากการปรับต่างความคมชัดด้วยวิธีที่แตกต่างกัน โดย (a) Original Image Contrast , (b) Histogram Equalization Contrast Enhancement , (c) Adaptive Histogram Equalization , (d) LPF Homomorphic Contrast Enhancement , (e) HPF Homomorphic Contrast Enhancement และ (f) Unsharp Mask Contrast Enhancement

โดยโครงงานนี้ได้นำเอกสารนี้มาศึกษาเพื่อใช้ในการพัฒนาส่วนของ Image processing application เพื่อสร้าง Filtering และการปรับความคมชัดของรูปภาพที่จะถูกปรับแต่งภายใน Image processing application

#### 2.4.4 Efficient Processing of Image Processing Applications on CPU/GPU

งานวิจัยนี้จัดทำขึ้นในปี 2020 โดย Najia Naz, Abdul Haseeb Malik, Abu Bakar Khurshid, Furqan Aziz, Bader Alouffi, M. Irfan Uddin และ Ahmed AlGhamdi กล่าวถึงเรื่องการกระบวนการ การแบ่งงานที่มีประสิทธิภาพสำหรับ Image processing application จากปกติการที่ประมวลผลงานด้าน

image processing ต้องที่สามารถพบเจอได้เป็นปกติคือ การกระจายภาระงานที่ไม่สมดุล ซึ่งทำให้ประสิทธิภาพของการทำงานนั้นลดลง โดยการแบ่งนั้นจะแบ่งออกเป็น 2 ช่วง ช่วงแรกจะเป็นการแบ่งและแยกแยะข้อมูลของรูปภาพเป็นส่วนๆ ที่เหมาะสมต่อการกระจายไปยังหน่วยประมวลผลต่างๆ จากนั้นในช่วงที่สอง ทำการกระจายเพิ่มเติมไปยัง CPU และ GPU ขึ้นอยู่กับความเร็วในการประมวลผล ดังรูป 2.11

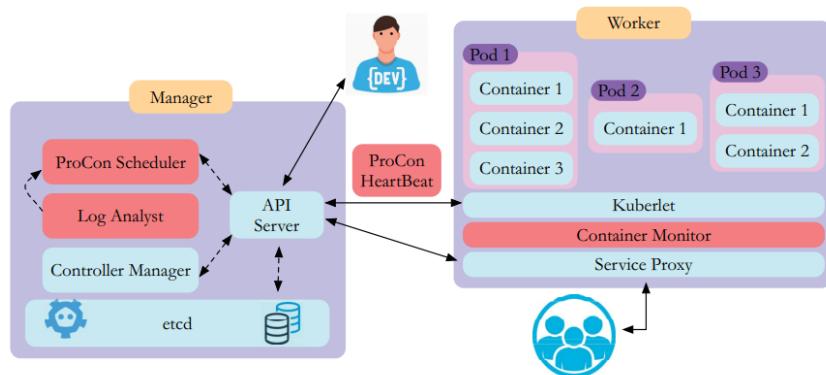


รูป 2.11 แผนผังรูปแบบการกระจายงานในช่วงที่สองที่มีการกระจายภาระงานระหว่าง CPU และ GPU

โดยโครงงานนี้ได้นำเอกสารนี้มาศึกษาเพื่อใช้ในการพัฒนาส่วน Load balance ที่จะทำหน้าที่แบ่งงานภาระงาน Image processing ไปยัง CPU และ GPU

#### 2.4.5 Progress-based Container Scheduling for Short-lived Applications in a Kubernetes Cluster

งานวิจัยนี้จัดทำขึ้นในปี 2019 โดย Yuqi Fu, Shaolun Zhang, Jose Terrero, Ying Mao, Guangya Liu, Sheng Li และ Dingwen Tao กล่าวถึงเรื่องการจัดวางรูปแบบ คอนเทนเนอร์ ที่ตั้งชื่อไว้ว่า ProCon โดยปกติแล้วจะมี Manager node เป็นตัวที่เลือกใช้งาน Worker node แต่ละตัวตามอัลกอริทึมที่ตั้งค่าเอาไว้ โดยที่ ProCon จะทำหน้าที่ในการกำหนด input คอนเทนเนอร์แล้ว พิจารณาการใช้ทรัพยากรทันที รวมไปถึงการประเมินทรัพยากรในอนาคตด้วย และขณะที่ประมวลผลนั้น ProCon ก็จะทำการจัดสมดุลของทรัพยากรภายใน Cluster ไปด้วย โดยรูปแบบแผนผังของ System Architecture ของ ProCon จะเป็นดังรูปที่ 2.12



รูป 2.12 แผนผัง System Architecture ของ ProCon

โดยโครงงานนี้ได้นำงานวิจัยมาศึกษาเพื่อสร้างรูปแบบการจัดการคอนเทนเนอร์ที่มีการจัดทรัพยากรให้แต่ละ Worker อย่างมีประสิทธิภาพ

## บทที่ 3

### การออกแบบและการพัฒนา

การออกแบบและการพัฒนาระบบ แอปพลิเคชันการประมวลผลภาพด้วยการจัดการลำดับงานบนระบบเครือข่าย แบ่งออกเป็น 3 ส่วน คือ Web Application ส่วนของ Image Processing Application และส่วนของ Server โดยมีรายละเอียดส่วนต่าง ๆ ดังนี้

#### 3.1 รายละเอียดของระบบ

##### 1) Web Application

- 1) ผู้ใช้สามารถสมัครสมาชิก เข้าสู่ระบบได้
- 2) สามารถเปลี่ยนรหัสผ่านได้
- 3) สามารถจัดเก็บไฟล์รูปภาพได้
- 4) สามารถสั่งงานประมวลผลภาพได้
- 5) สามารถทดลองดูตัวอย่างงานประมวลผลภาพได้
- 6) สามารถจัดเก็บ Weight Model ได้
- 7) สามารถซื้อขาย Weight Model ได้
- 8) สามารถสั่งงานประมวลผลภาพบน Marker Place ได้
- 9) สามารถดูสถิติการประมวลผลภาพได้

##### 2) Image Processing Application

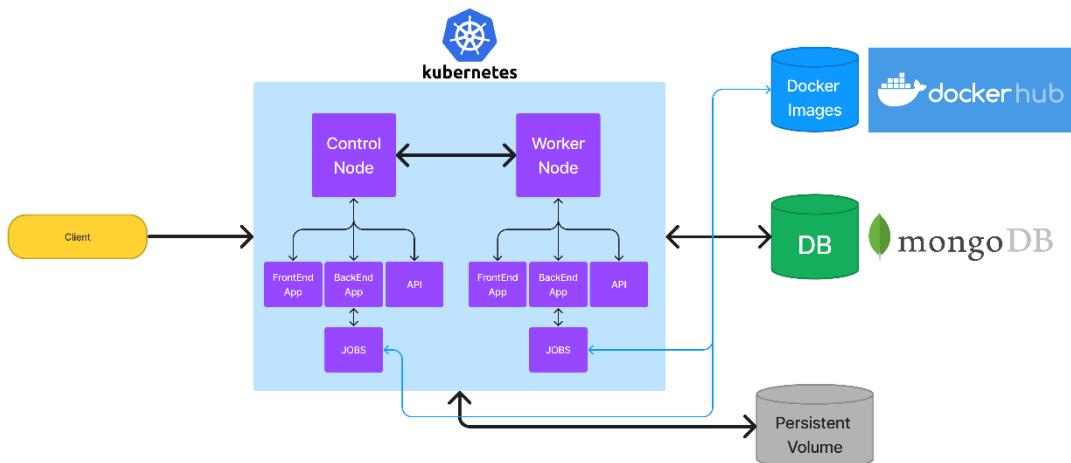
- 1) สามารถประมวลผลภาพเป็นภาพ ASCII ได้
- 2) สามารถประมวลผลภาพเป็นภาพ PixelArt ได้
- 3) สามารถประมวลผลภาพเป็นภาพ Mosaic ได้
- 4) สามารถประมวลผลภาพเป็นภาพขาวดำได้
- 5) สามารถประมวลผลภาพเป็นภาพเพื่อคอมพิวเตอร์ทั้งภาพได้
- 6) สามารถประมวลผลภาพเป็นภาพจาก Model และ Weight ได้

### 3) Server

- 1) สามารถ Task Management งานประมวลผล ได้
- 2) สามารถ Load Balance การ Request เข้าถึง Service ได้
- 3) สามารถจัดเก็บข้อมูลไว้แยกจากตัว Server ได้
- 4) สามารถ High Availability Server ได้
- 5) สามารถ Scale Service บน Server ได้

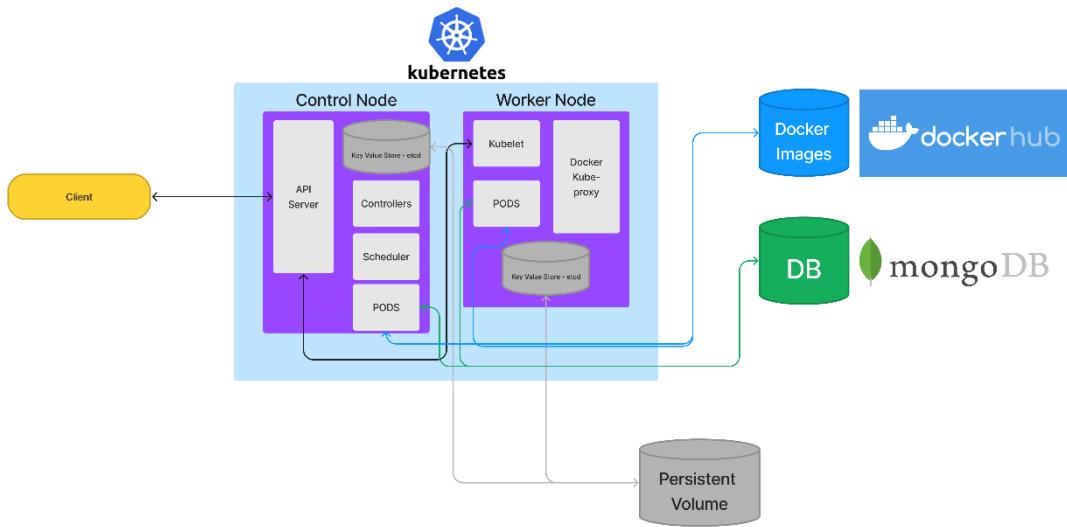
## 3.2 ภาพรวมของระบบ

ภาพรวมในการพัฒนาระบบที่แสดงให้เห็นส่วนต่าง ๆ ที่อยู่ในแผนการพัฒนาโครงการที่ประกอบไปด้วยส่วนประกอบหลัก ๆ คือ Kubernetes ที่ครอบระบบทุกอย่างอยู่โดยจะมี Node เป็น Control ที่ใช้ควบคุมการทำงานต่าง ๆ และ รับคำสั่งต่าง ๆ ให้กับ Worker โดยที่มีการเชื่อมต่อกับ DockerHub MongoDB และ สร้าง Persistent Volume แยกขึ้นมา จะแสดงดังรูปที่ 3.1



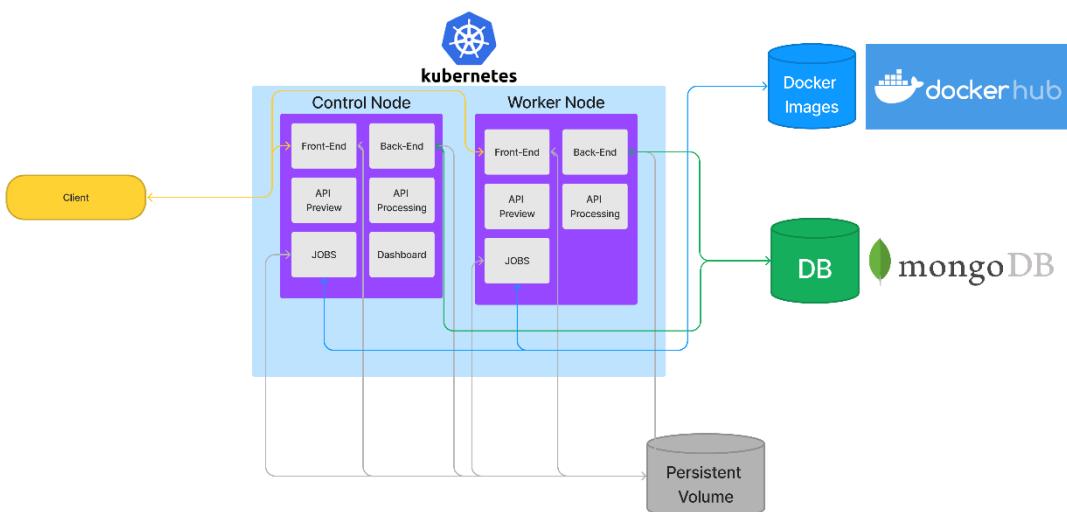
รูป 3.1 ภาพรวมของระบบ Overview

โดยภาพรวมของระบบใน Level 0 นั้นจะกล่าวถึงการเชื่อมต่อของ Control Node และ Worker Node ที่มีการ Scheduler ผ่าน API-server เพื่อทำการ Load Balance งานที่ Control ได้รับไปยัง Worker แต่ละเครื่องตามที่ได้ตั้งค่าไว้ใน Load Balancer ของ Kubernetes ดังรูปที่ 3.2



รูป 3.2 ภาพรวมของระบบ Level 0

โดยภาพรวมของระบบใน Level 1 นั้นจะกล่าวถึงการเชื่อมต่อของ User ไปยัง Web Service นั้นโดยมี Port การเชื่อมต่อไป ในส่วนของ Front-End และ เข้าใช้งานส่วนของการเรียกใช้อิน ๆ ผ่านด้วย Back-End ซึ่งจะมีการเปิดรับ API สำหรับทำการประมวลผลภาพเป็น Preview และ Processing สำหรับการประมวลผลงานส่วนอื่น ๆ มีตัว JOBS ที่เก็บงานประมวลผลที่ได้รับส่งเข้ามา และ กำลังทำงานอยู่ รวมถึงหน้า Dashboard สำหรับจัดการ เฝ้าดูการทำงานของระบบโดยรวม ได้ ดังรูปที่ 3.3

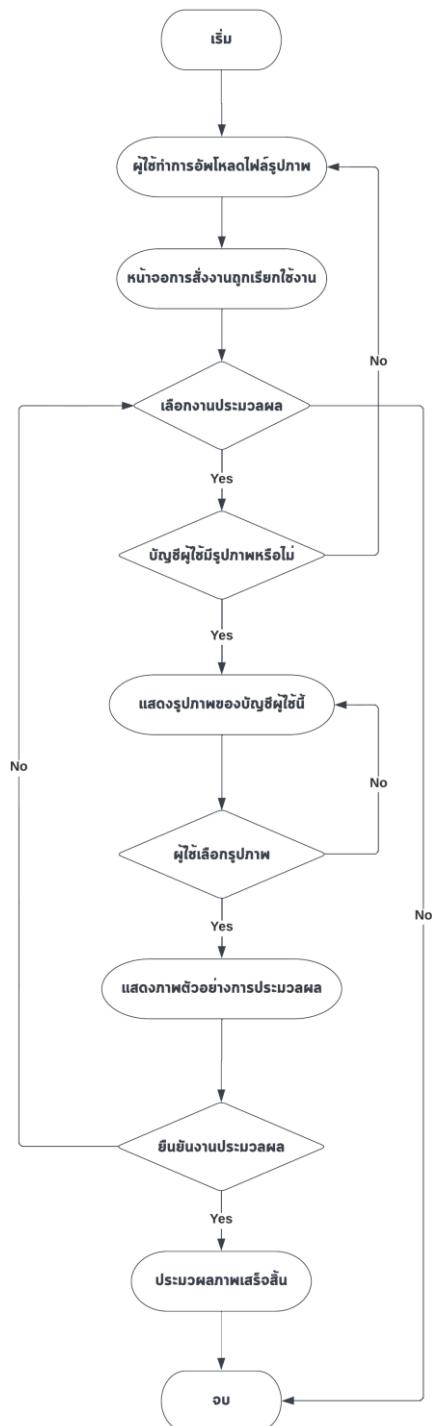


รูป 3.3 ภาพรวมของระบบ Level 1

### 3.3 แผนภาพผังงาน (Flowchart)

#### 3.3.1 แผนภาพผังงาน (Flowchart) ของการสั่งงานประมวลผลภาพ (Web Application)

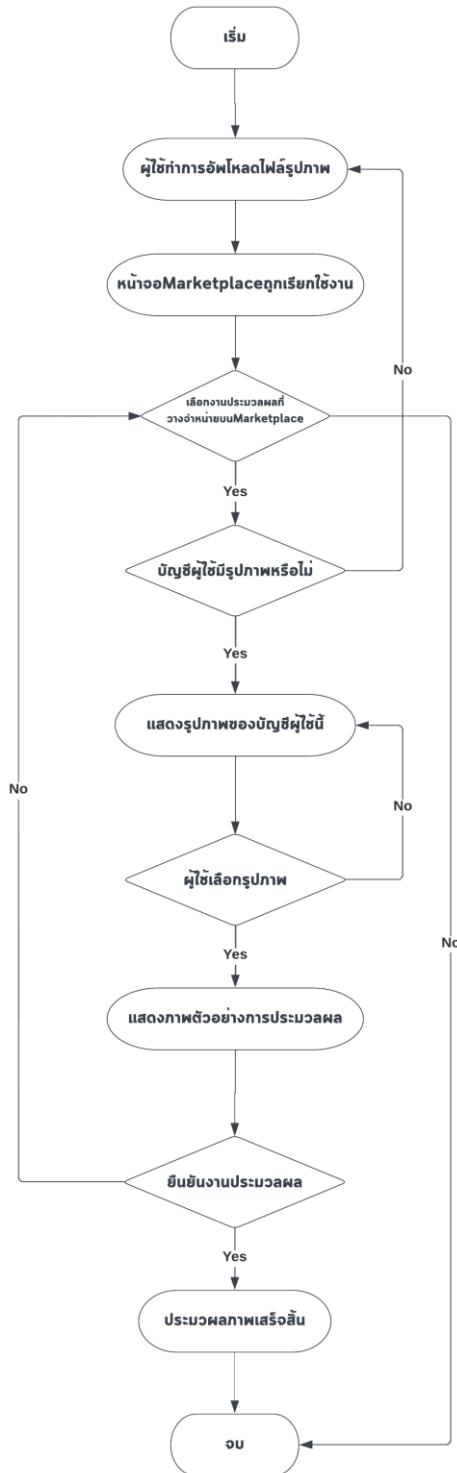
แผนภาพผังงานในส่วนของ Web Application บนเว็บที่ผู้ใช้งานเริ่มทำการใช้งานการสั่งงานประมวลผลภาพจะถูกจัดการโดยแผนภาพผังงานในส่วนของการสั่งงานประมวลผลภาพ จะแสดงดังรูปที่ 3.4



รูป 3.4 แผนภาพผังงานของการสั่งงานประมวลผลภาพ (Web Application)

### 3.3.2 แผนภาพผังงาน (Flowchart) การสั่งงานประมวลผลภาพบน Marketplace (Web Application)

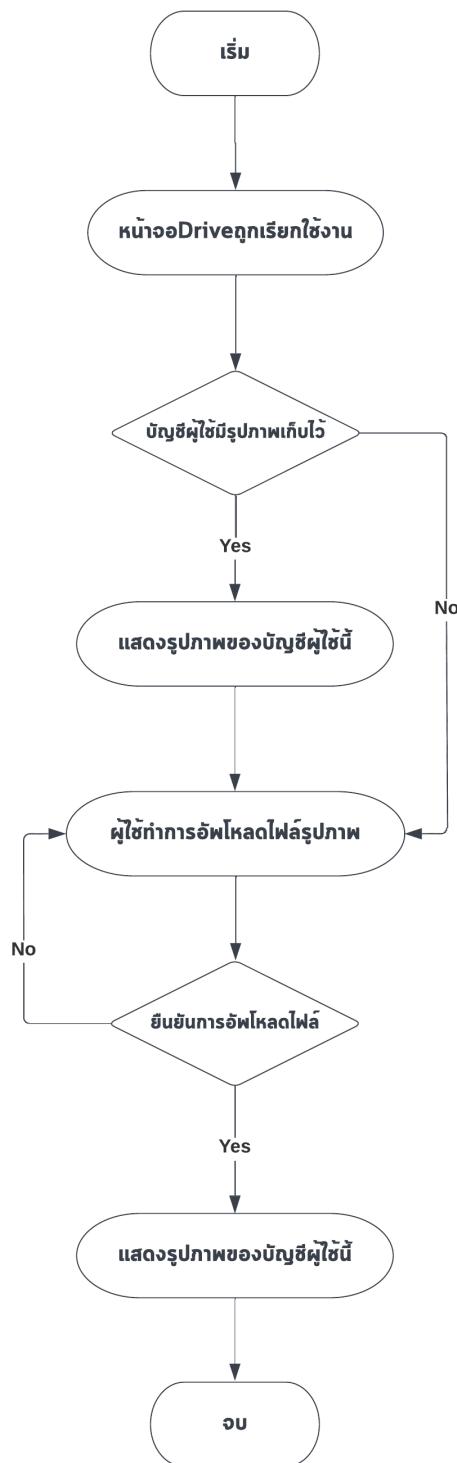
แผนภาพผังงานในส่วนของ Web Application บนเว็บที่ผู้ใช้งานเริ่มทำการใช้งานการสั่งงานประมวลผลภาพบน Marketplace จนถึงการประมวลผลภาพเสร็จสิ้น โดยแผนภาพผังงานในส่วนของการสั่งงานประมวลผลภาพ จะแสดงดังรูปที่ 3.5



รูป 3.5 แผนภาพผังงานของการสั่งงานประมวลผลภาพบน Marketplace (Web Application)

### 3.3.3 แผนภาพผังงาน (Flowchart) การจัดเก็บไฟล์ภาพไว้ใน Drive (Web Application)

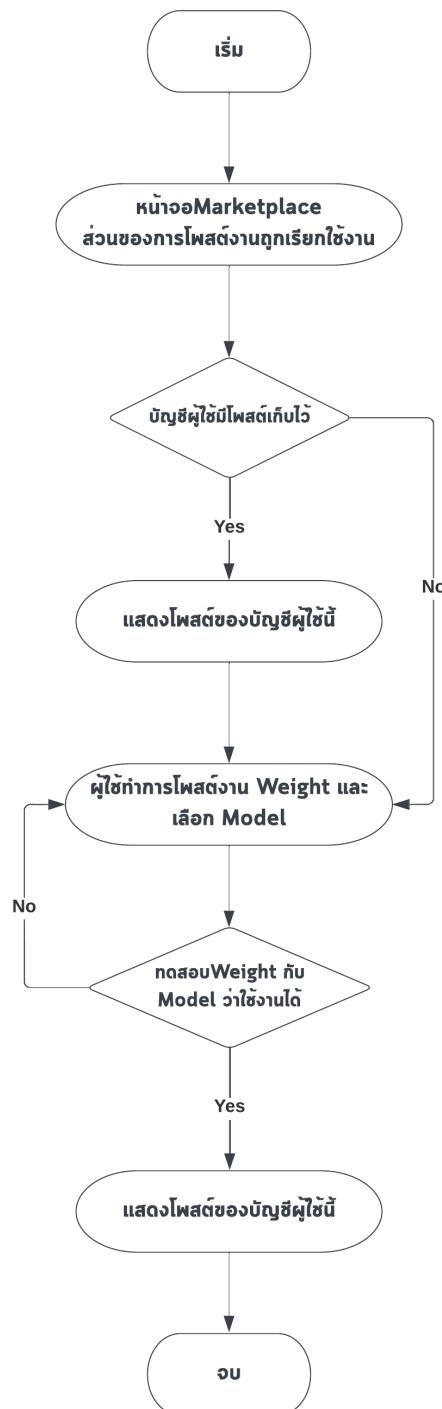
แผนภาพผังงานในส่วนของ Web Application บนเว็บที่ผู้ใช้งานเริ่มทำการจัดเก็บไฟล์ภาพไว้ใน Drive จะถูกจัดการโดยการจัดเก็บไฟล์ภาพเสร็จสิ้น โดยแผนภาพผังงานในส่วนของการจัดเก็บไฟล์ภาพไว้ใน Drive จะแสดงดังรูปที่ 3.6



ຮູບ 3.6 แผนภาพผังงานของการจัดเก็บไฟล์ภาพไว้ใน Drive (Web Application)

### 3.3.4 แผนภาพผังงาน (Flowchart) การใช้งาน Marketplace สำหรับขาย Weight ที่ผู้ใช้ได้ทำการ Train (Web Application)

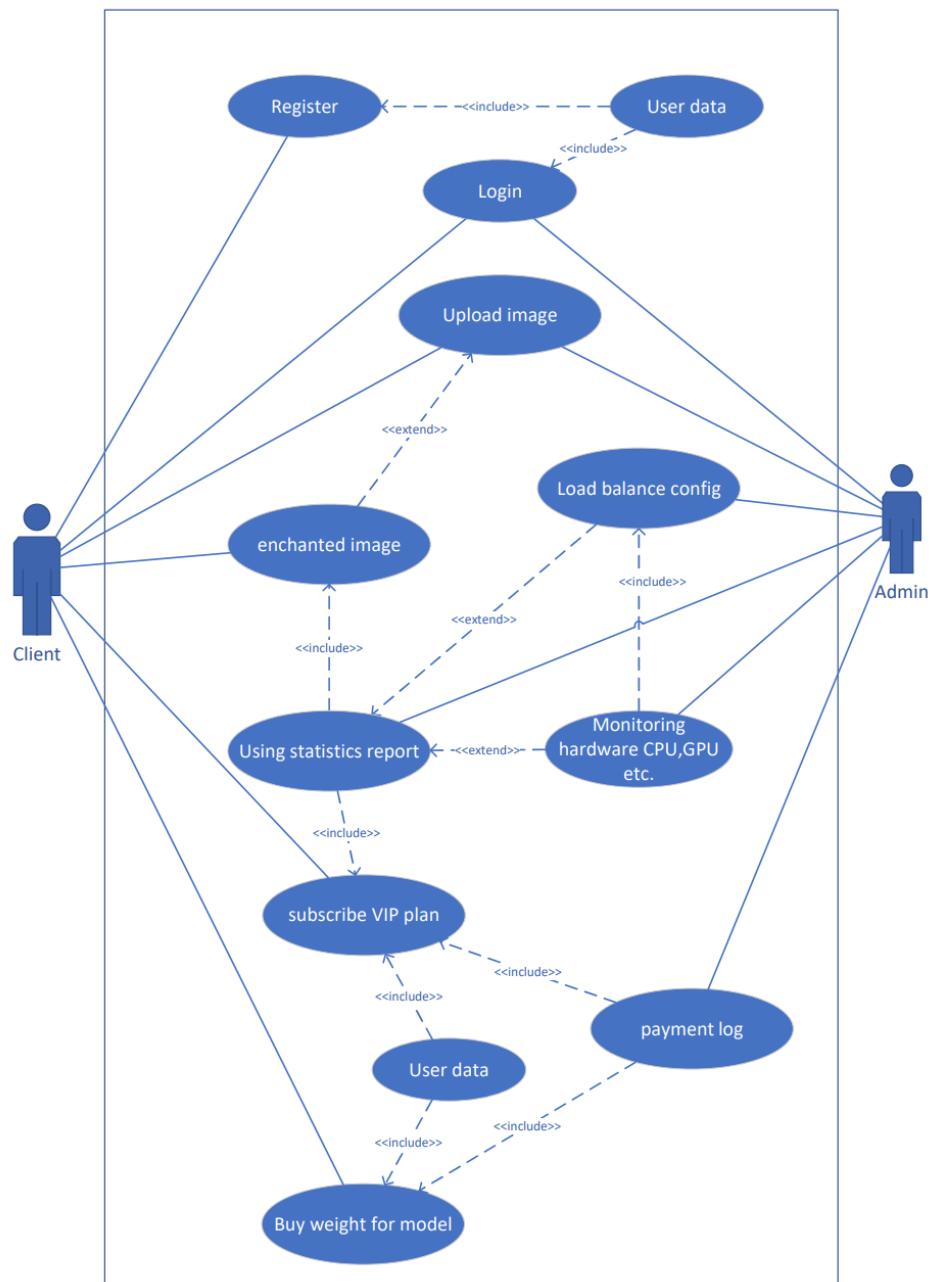
แผนภาพผังงานในส่วนของ Web Application บนเว็บที่ผู้ใช้งานเริ่มทำการใช้งาน Marketplace สำหรับขาย Weight ที่ผู้ใช้ได้ทำการ Train จนถึงการโพสต์งานเสร็จสิ้น โดยแผนภาพผังงานในส่วนของการขาย Weight ที่ผู้ใช้ได้ทำการ Train จะแสดงดังรูปที่ 3.7



รูป 3.7 แผนภาพผังงานของการใช้งาน Marketplace สำหรับขาย Weight ที่ผู้ใช้ได้ทำการ Train

### 3.4 แผนภาพยูสเคส (Use Case Diagram)

แผนภาพยูสเคสจะแสดงให้เห็นถึงผู้ใช้ที่ถูกแบ่งเป็น 2 ประเภท คือ Client คือ ผู้ใช้ที่เข้ามาในรูปแบบลูกค้าที่ต้องการใช้บริการงานประมวลผลภาพต่างๆ และ Administer คือผู้ใช้ที่เป็นผู้ดูแลในส่วนของข้อมูลในระบบ และ ตั้งค่า ในส่วนของการประมวลผลภาพ รวมไปถึงการตรวจสอบการ payment ต่างๆ โดยแผนภาพ ยูสเคส จะแสดงดังรูปที่ 3.8



รูป 3.8 แผนภาพยูสเคส

โดย ส่วนรายละเอียดของ Use case จะแสดงดังตารางที่ 3.1 ถึง 3.10

ตาราง 3.1 รายละเอียดของ Use case Register

<b>Use Case Title :</b> Register	<b>Use case ID:</b> 1
<b>Description :</b> ลงทะเบียนสมัครเข้าใช้งาน	
<b>Actor :</b> Client	
<b>Precondition :</b> เข้าใช้งานเว็บไซต์	
<b>Postcondition :</b> ได้รับผลการยืนยันการสมัคร	
<b>Fail End Condition :</b> แจ้งเตือนข้อความ “การสมัครบัญชีล้มเหลว เนื่องจาก (สาเหตุ) กรุณาลองใหม่”	
<b>Trigger :</b> กดปุ่มคลิกยืนยันการสมัครใช้งาน	

ตาราง 3.2 รายละเอียดของ Use case Login

<b>Use Case Title :</b> Login	<b>Use case ID:</b> 2
<b>Description :</b> ลงชื่อเข้าใช้งานเว็บไซต์	
<b>Actor :</b> Client, Administer	
<b>Precondition :</b> เข้าใช้งานเว็บไซต์และสมัครเข้าใช้งานแล้ว	
<b>Postcondition :</b> ได้รับผลการยืนยันตัวตน	
<b>Fail End Condition :</b> แจ้งเตือนข้อความ “การลงชื่อเข้าใช้ล้มเหลว เนื่องจาก (สาเหตุ) กรุณาลองใหม่”	
<b>Trigger :</b> กดปุ่มคลิกลงชื่อเข้าใช้	

**ตาราง 3.3 รายละเอียดของ Use case Upload image**

<b>Use Case Title :</b> Upload image	<b>Use case ID:</b> 3
<b>Description :</b> อัปโหลดรูปภาพเพื่อเก็บไว้ในฐานข้อมูล	
<b>Actor :</b> Client, Administer	
<b>Precondition :</b> ยืนยันตัวตนลงชื่อเข้าใช้	
<b>Postcondition :</b> ได้รับผลอัปโหลดรูปภาพไว้ในฐานข้อมูล	
<b>Fail End Condition :</b> แจ้งเตือนข้อความ “การอัปโหลดรูปภาพล้มเหลว เนื่องจาก (สาเหตุ) กรุณาลองใหม่”	
<b>Trigger :</b> เลือกรูปภาพที่จะอัปโหลดและกดปุ่มอัปโหลด	

**ตาราง 3.4 รายละเอียดของ Use case enchanted image**

<b>Use Case Title :</b> Enchanted image	<b>Use case ID:</b> 4
<b>Description :</b> ปรับแต่งรูปภาพ	
<b>Actor :</b> Client	
<b>Precondition :</b> ยืนยันตัวตนลงชื่อเข้าใช้	
<b>Postcondition :</b> ได้รับรูปภาพที่ผ่านการปรับแต่ง	
<b>Fail End Condition :</b> แจ้งเตือนข้อความ “การปรับแต่งรูปภาพล้มเหลว เนื่องจาก (สาเหตุ) กรุณาลองใหม่”	
<b>Trigger :</b> เลือกรูปภาพที่จะปรับแต่ง เลือกฟังก์ชันการปรับแต่ง และกดปุ่มปรับแต่ง	

**ตาราง 3.5 รายละเอียดของ Use case Load balance config**

<b>Use Case Title :</b> Load balance config	<b>Use case ID:</b> 5
<b>Description :</b> ปรับแต่งการทำงานของ Load balance	
<b>Actor :</b> Administer	
<b>Precondition :</b> ยืนยันตัวตนลงชื่อเข้าใช้	
<b>Postcondition :</b> ได้ผลยืนยันการปรับแต่ง Load balance	
<b>Fail End Condition :</b> แจ้งเตือนข้อความ “การปรับแต่งรูปภาพล้มเหลว เนื่องจาก (สาเหตุ) กรุณาลองใหม่”	
<b>Trigger :</b> เลือกฟังก์ชันการปรับแต่ง และกดปุ่มปรับแต่ง	

### ตาราง 3.6 รายละเอียดของ Monitoring hardware

<b>Use Case Title :</b> Monitoring hardware	<b>Use case ID:</b> 6
<b>Description :</b> ดูระดับการทำงานของฮาร์ดแวร์ต่างๆ เช่น CPU ,GPU เป็นต้น	
<b>Actor :</b> Administer	
<b>Fail End Condition :</b>	
<b>Precondition :</b> ยืนยันตัวตนลงชื่อเข้าใช้	
<b>Postcondition :</b> ได้ผลการทำงานของฮาร์ดแวร์	
<b>Fail End Condition :</b> แจ้งเตือนข้อความ “ไม่สามารถเข้าถึงระดับการทำงานของฮาร์ดแวร์ได้เนื่องจาก (สาเหตุ) กรุณาลองใหม่”	
<b>Trigger :</b> เลือกฟังก์ชันดูระดับการทำงาน	

### ตาราง 3.7 รายละเอียดของ Using statistics report

<b>Use Case Title :</b> Using statistics report	<b>Use case ID:</b> 7
<b>Description :</b> ดูสถิติการใช้งานของผู้ใช้ที่อยู่ในระดับ Client	
<b>Actor :</b> Administer	
<b>Precondition :</b> ยืนยันตัวตนลงชื่อเข้าใช้	
<b>Postcondition :</b> ได้สถิติการใช้งาน	
<b>Fail End Condition :</b> แจ้งเตือนข้อความ “ไม่สามารถเข้าถึงสถิติผู้ใช้ได้ เนื่องจาก (สาเหตุ) กรุณาลองใหม่”	
<b>Trigger :</b> เลือกฟังก์ชันดูสถิติการใช้งาน	

ตาราง 3.8 รายละเอียดของ Subscribe VIP plan

<b>Use Case Title :</b> Subscribe VIP plan	<b>Use case ID:</b> 8
<b>Description :</b> สมัครเข้าเป็นผู้ใช้ Client ระดับ VIP	
<b>Actor :</b> Client	
<b>Precondition :</b> ยืนยันตัวตนลงชื่อเข้าใช้	
<b>Postcondition :</b> ได้สถานะเป็น Client ระดับ VIP	
<b>Fail End Condition :</b> แจ้งเตือนข้อความ “การสมัครล้มเหลว เนื่องจาก (สาเหตุ) กรุณาลองใหม่”	
<b>Trigger :</b> เลือกสมัครเข้าเป็นผู้ใช้ Client ระดับ VIP และชำระเงิน	

ตาราง 3.9 รายละเอียดของ Buy weight for model

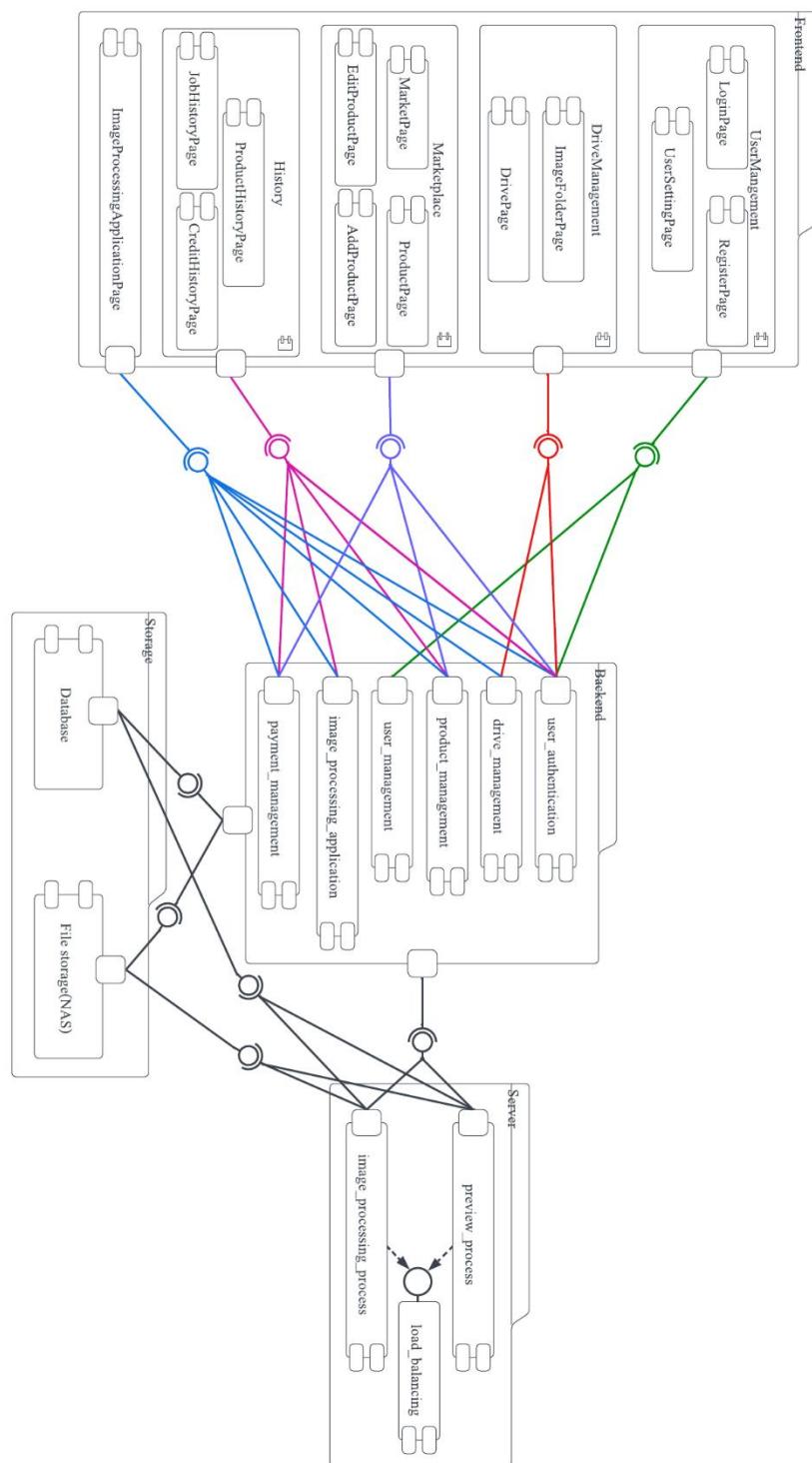
<b>Use Case Title :</b> Buy weight for model	<b>Use case ID:</b> 9
<b>Description :</b> ซื้อ weight สำหรับ Model เพื่อนำไปทำการประมวลผลภาพ	
<b>Actor :</b> Client	
<b>Precondition :</b> ยืนยันตัวตนลงชื่อเข้าใช้	
<b>Postcondition :</b> ได้รับ weight สำหรับ Model ไว้ในฐานข้อมูล	
<b>Fail End Condition :</b> แจ้งเตือนข้อความ “การซื้อขายล้มเหลว เนื่องจาก (สาเหตุ) กรุณาลองใหม่”	
<b>Trigger :</b> เลือก weight สำหรับ Model และชำระเงิน	

ตาราง 3.10 รายละเอียดของ Payment log

<b>Use Case Title :</b> Payment log	<b>Use case ID:</b> 10
<b>Description :</b> คุณบันทึกการชำระเงินต่างๆ	
<b>Actor :</b> Administer	
<b>Precondition :</b> ยืนยันตัวตนลงชื่อเข้าใช้	
<b>Postcondition :</b> ได้ผลบันทึกการชำระเงิน	
<b>Fail End Condition :</b> แจ้งเตือนข้อความ “ไม่สามารถเข้าถึงบันทึกการชำระเงินได้ เนื่องจาก (สาเหตุ) กรุณาลองใหม่”	
<b>Trigger :</b> เลือกฟังก์ชันคุณบันทึกการชำระเงิน	

### 3.5 แผนภาพแสดงองค์ประกอบ (Component Diagram)

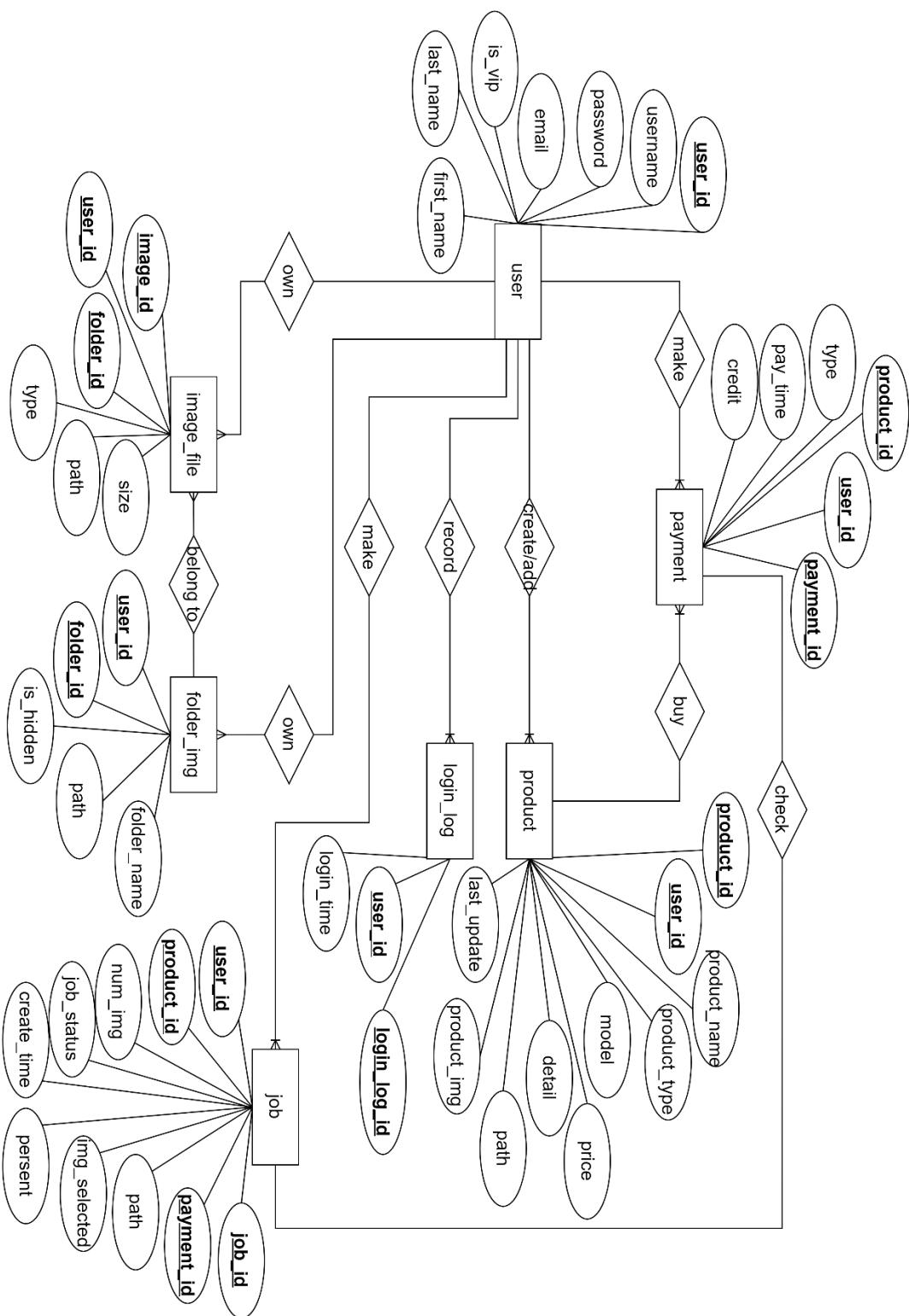
แผนภาพแสดงโครงสร้างและความสัมพันธ์ระหว่างองค์ประกอบภายในโครงงานเพื่อให้เกิดความเข้าใจและสามารถสื่อสารได้เข้าใจตรงกันภายในกลุ่มโครงงาน และยังช่วยลดความซับซ้อนของระบบลงไปอีกด้วย โดยแผนภาพแสดงองค์ประกอบจะแสดงดังรูปที่ 3.9



รูป 3.9 แผนภาพแสดงองค์ประกอบ

### 3.6 แผนภาพแสดงความสัมพันธ์ของข้อมูล (E-R Diagram)

แผนภาพแสดงความสัมพันธ์ของข้อมูลที่ใช้ในการออกแบบระบบฐานข้อมูล ประกอบด้วย 9 tables คือ user, image\_file , folder\_img , img\_app , job , login\_log , product , payment และ feed\_post โดย แผนภาพแสดงความสัมพันธ์ของข้อมูลจะแสดงดังรูปที่ 3.10



รูป 3.10 แผนภาพแสดงความสัมพันธ์ของข้อมูล

โดยรายละเอียดของแผนภาพแสดงความสัมพันธ์ของข้อมูลแสดงดังตารางที่ 3.11 ถึง 3.17

ตาราง 3.11 แสดงรายละเอียดตารางข้อมูลผู้ใช้ (user)

คีย์	ชื่อแอ็อดทริบิวต์	ชนิดตัวแปร	คำอธิบาย
PK	user_id	Varchar(50)	รหัสประจำตัวของผู้ใช้งาน
-	username	Varchar(50)	ชื่อสำหรับแสดงผลของผู้ใช้
-	password	Varchar(50)	รหัสผ่านของผู้ใช้งาน
-	email	Varchar(50)	อีเมลของผู้ใช้งาน
-	is_vip	Boolean	สถานะระดับของผู้ใช้
-	first_name	Varchar(50)	ชื่อของผู้ใช้งาน
-	last_name	Varchar(50)	นามสกุลของผู้ใช้งาน

ตาราง 3.12 แสดงรายละเอียดตารางข้อมูลไฟล์รูปภาพ (Image\_file)

คีย์	ชื่อแอ็อดทริบิวต์	ชนิดตัวแปร	คำอธิบาย
PK	image_id	Varchar(50)	รหัสข้อมูลรูปภาพ
FK	user_id	Varchar(50)	รหัสประจำตัวของผู้ใช้งานเจ้าของรูปภาพ
FK	folder_id	Varchar(50)	รหัสข้อมูลโฟลเดอร์ที่เก็บรูปภาพ
-	type	Varchar(50)	ประเภทของรูปภาพ
-	path	Varchar(50)	เส้นทางการเข้าถึงไฟล์รูปภาพ
-	size	Integer	ขนาดของไฟล์รูปภาพ

ตาราง 3.13 แสดงรายละเอียดตารางข้อมูลของโฟลเดอร์เก็บรูปภาพ (folder\_img)

คีย์	ชื่อแอ็อดทริบิวต์	ชนิดตัวแปร	คำอธิบาย
PK	folder_id	Varchar(50)	รหัสข้อมูลโฟลเดอร์
FK	user_id	Varchar(50)	รหัสประจำตัวของผู้ใช้งานเจ้าของโฟลเดอร์
-	is_hidden	Varchar(50)	สถานะการแสดงผล
-	folder_name	Varchar(50)	ชื่อของโฟลเดอร์
-	path	Varchar(50)	เส้นทางการเข้าถึงโฟลเดอร์

ตาราง 3.14 แสดงรายละเอียดตารางเก็บข้อมูลการเข้าสู่ระบบ (login\_log)

คีย์	ชื่อแอ็อดทริบิวต์	ชนิดตัวแปร	คำอธิบาย
PK	login_log_id	Varchar(50)	รหัสข้อมูล Log การเข้าสู่ระบบ
FK	user_id	Varchar(50)	รหัสประจำตัวของผู้ใช้งานที่เข้าสู่ระบบ
-	login_time	Datetime	เวลาที่ทำการเข้าสู่ระบบ

ตาราง 3.15 แสดงรายละเอียดตารางข้อมูลการชำระเงิน (payment)

คีย์	ชื่อแอ็อดทริบิวต์	ชนิดตัวแปร	คำอธิบาย
PK	payment_id	Varchar(50)	รหัสข้อมูลการชำระเงิน
FK	user_id	Varchar(50)	รหัสประจำตัวผู้ใช้ที่ชำระเงิน
FK	product_id	Varchar(50)	รหัสสินค้าที่ทำการซื้อ
-	type	Varchar(50)	ประเภทของการชำระเงิน
-	pay_time	Datetime	เวลาที่ทำการชำระเงิน
-	credit	Integer	จำนวนเครดิตที่ใช้

ตาราง 3.16 แสดงรายละเอียดตารางข้อมูลของสินค้า (product)

คีย์	ชื่อแอ็อดทริบิวต์	ชนิดตัวแปร	คำอธิบาย
PK	product_id	Varchar(50)	รหัสประจำสินค้า
FK	user_id	Varchar(50)	รหัสประจำตัวผู้ใช้ที่เป็นเจ้าของสินค้า
-	product_name	Varchar(50)	ชื่อสินค้า
-	product_type	Varchar(50)	ประเภทสินค้า
-	model	Varchar(50)	โมเดลที่สินค้าใช้ในการประมวลผล
-	price	Integer	ราคาของสินค้า
-	detail	Vachar(1000)	รายละเอียดของสินค้า
-	path	Vachar(50)	เส้นทางเก็บไฟล์สินค้า
-	product_img	Vachar(50)	เส้นทางเก็บไฟล์รูปของสินค้า
-	last_update	Datetime	วันเวลาที่มีการเปลี่ยนแปลงรายละเอียดสินค้าล่าสุด

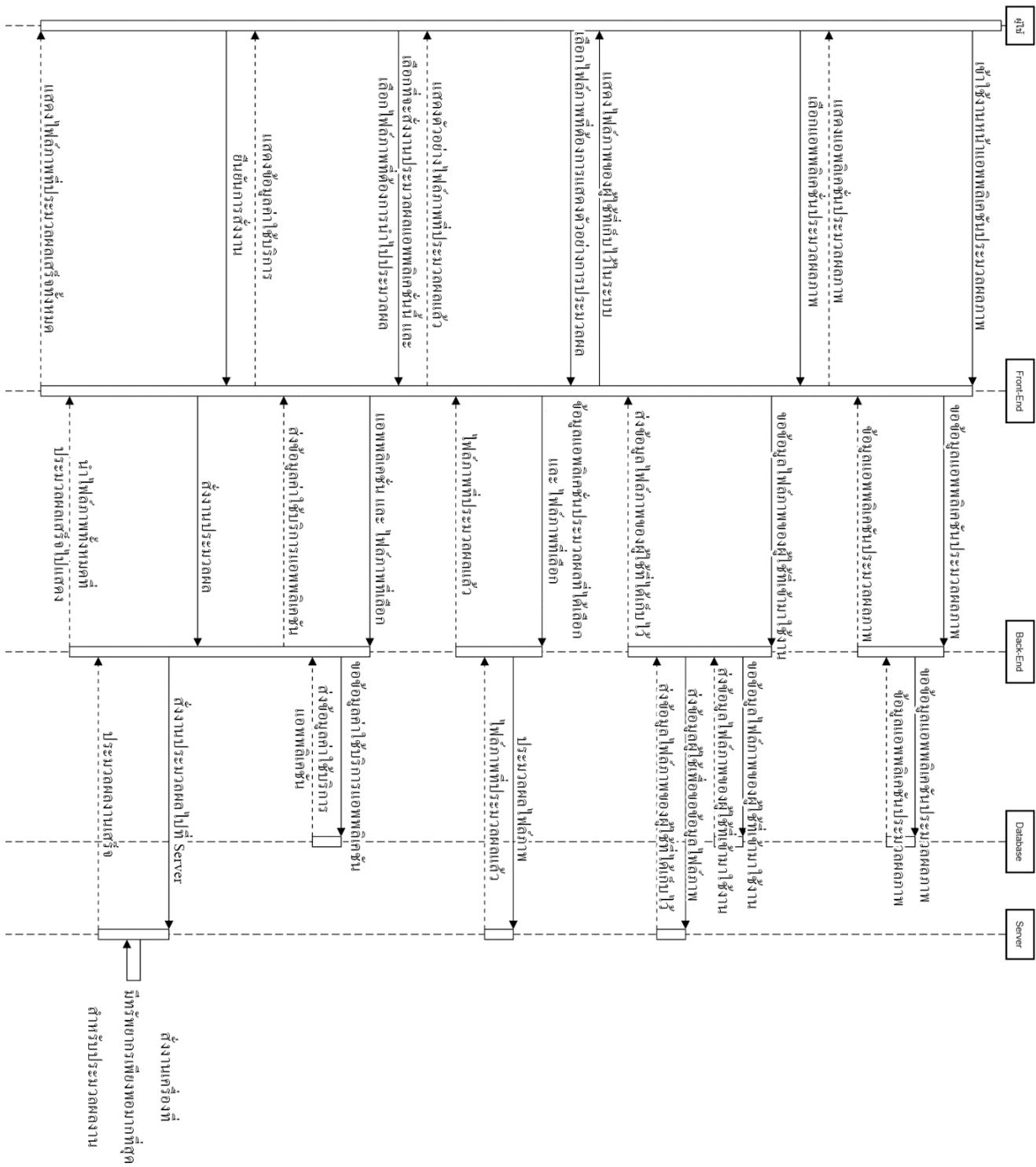
ตาราง 3.17 แสดงรายละเอียดตารางข้อมูลการสั่งงานประมวลผล (job)

คีย์	ชื่อแอ็อดทริบิวต์	ชนิดตัวแปร	คำอธิบาย
PK	job_id	Varchar(50)	รหัสระบุงานที่สั่ง
FK	user_id	Varchar(50)	รหัสประจำตัวของผู้ใช้งานที่สั่งงาน
FK	product_id	Varchar(50)	รหัสข้อมูล สินค้า ที่ต้องการนำมาใช้งานร่วม
FK	Payment_id	Varchar(50)	รหัสข้อมูลการชำระเงินที่เกิดขึ้นตอนสั่งงาน ประมวลผล
-	img_selected	Varchar(50)	ข้อมูลรูปที่สั่งงาน
-	path	Varchar(50)	เส้นทางของไฟล์เดอร์ที่เก็บไฟล์รูปที่จะสั่งงาน
-	num_img	Integer	จำนวนรูปภาพที่สั่งงาน
-	persent	Integer	เปอร์เซ็นต์แสดงผลการประมวลผล
-	job_status	Integer	สถานะของงานที่สั่ง
-	create_time	Datetime	เวลาที่สั่งงาน

### 3.7 แผนภาพการทำงานแบบลำดับปฏิสัมพันธ์ (Sequence Diagram)

### 3.7.1 แผนภาพการทำงานในส่วนของการสั่งงานประมวลผลภาพ (Web Application)

โดยรูปแบบคำดับปฏิสัมพันธ์นี้จะเริ่มตั้งแต่เหตุการณ์ที่ ผู้ใช้งานเริ่มทำการใช้งานการสั่ง งาน ประมวลผลภาพจนถึงการประมวลผลภาพเสร็จสิ้น โดยรูปภาพแสดงการทำงานแบบคำดับปฏิสัมพันธ์ โดยรวมในส่วนของการสั่งงานประมวลผลภาพ จะแสดงดังรูปที่ 3.11

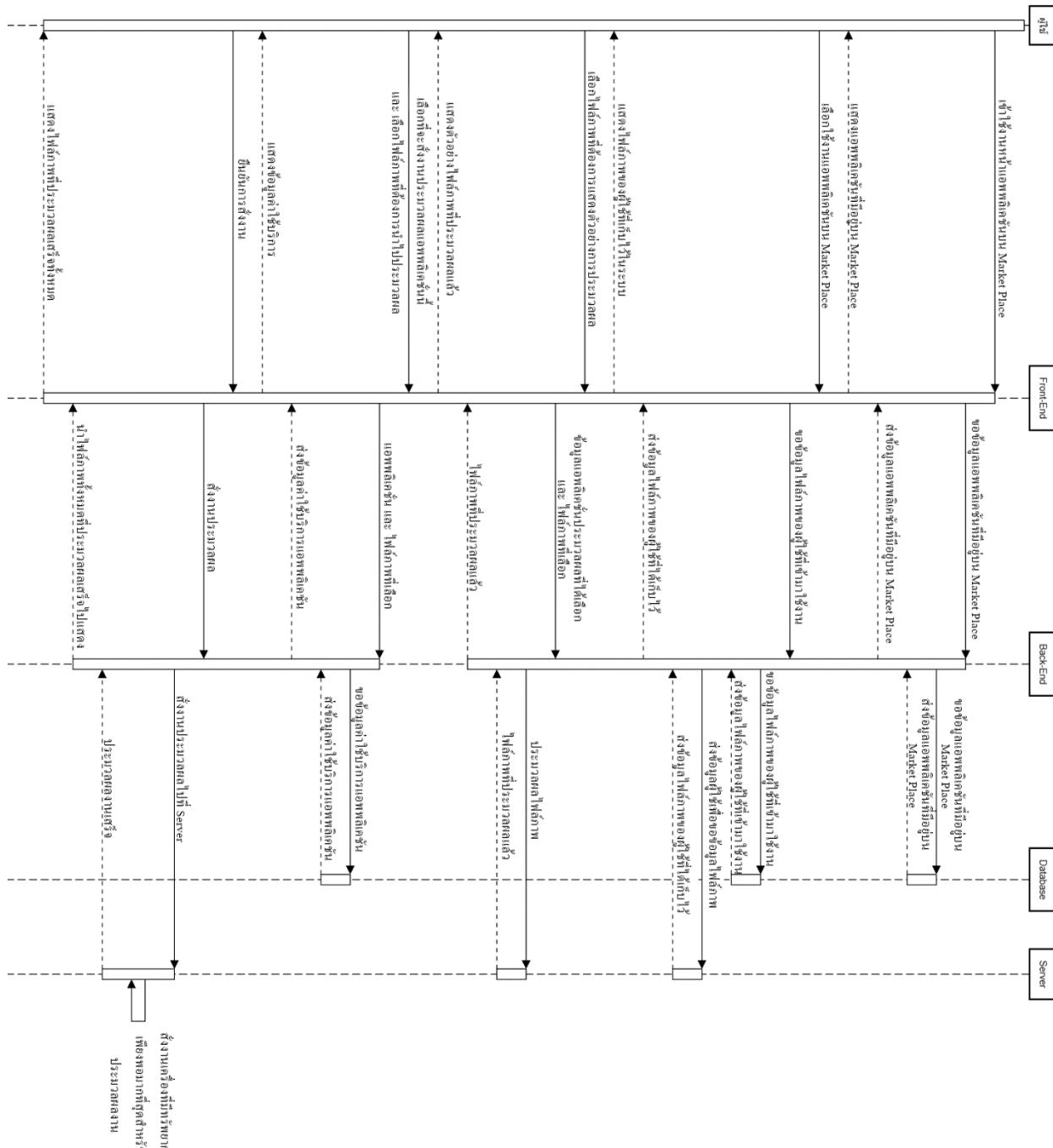


รูป 3.11 รูปภาพแสดงการทำางานแบบลำดับปฏิสัมพันธ์โดยรวมในส่วนของการสั่งงานประมวลผลภาพ

### 3.7.2 แผนภาพการทำงานในส่วนของการสั่งงานประมวลผลภาพบน Marketplace (Web)

### Application)

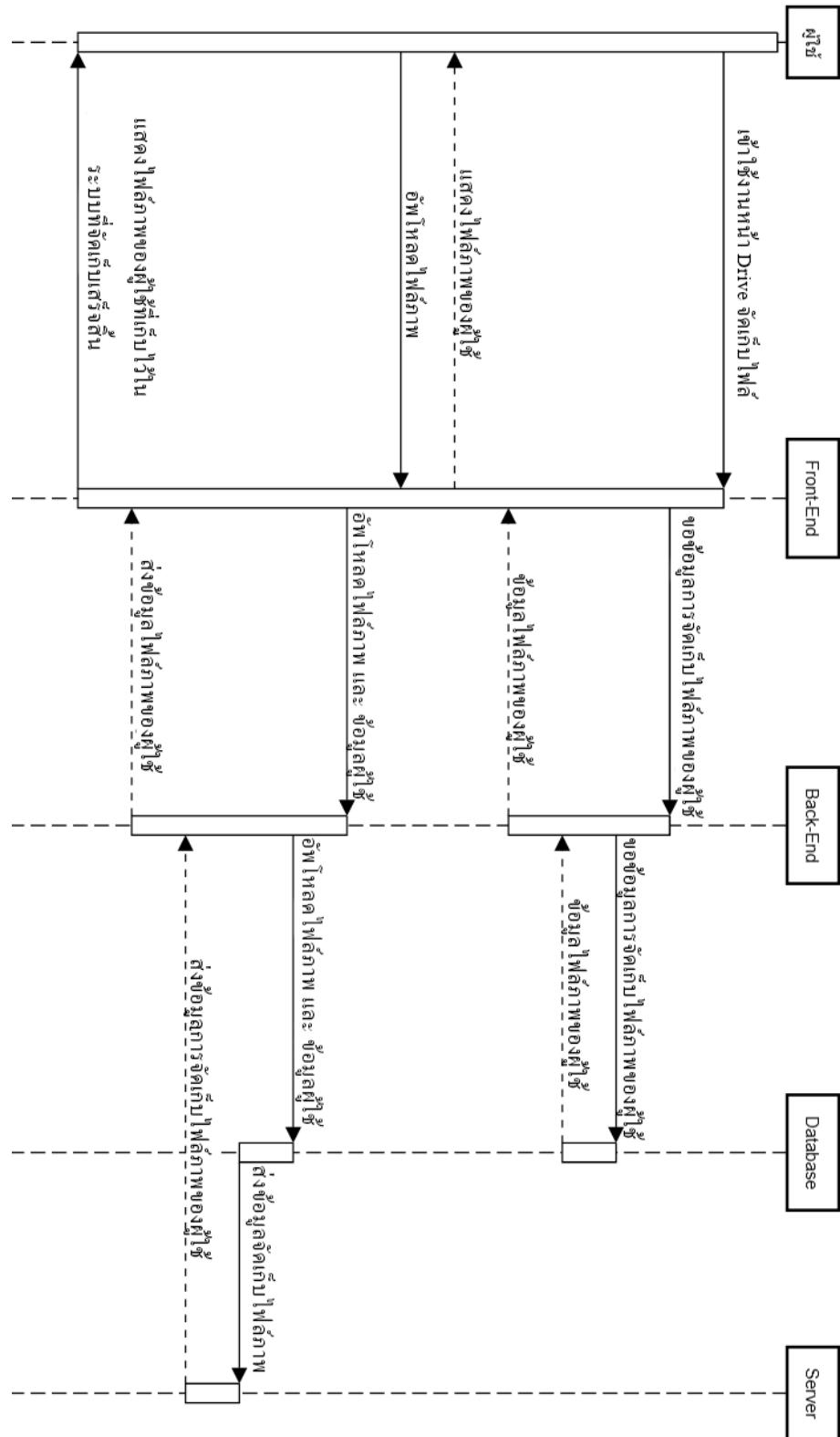
โดยรูปแบบลำดับปฏิสัมพันธ์นี้จะเริ่มตั้งแต่เหตุการณ์ที่ ผู้ใช้งานเริ่มทำการใช้งานการสั่ง งาน ประมวลผลภาพบน Marketplace จนถึงการประมวลผลภาพเสร็จสิ้น โดยรูปภาพแสดงการทำงานแบบ ลำดับปฏิสัมพันธ์โดยรวมในส่วนของการสั่งงานประมวลผลภาพ จะแสดงดังรูปที่ 3.12



รูป 3.12 รูปภาพแสดงการทำงานแบบลำดับปกิสัมพันธ์โดยรวมในส่วนของการสั่งงานประมวลผลภาพพุ่น

### 3.7.3 แผนภาพการทำงานในส่วนของการจัดเก็บไฟล์ภาพไว้ใน Drive (Web Application)

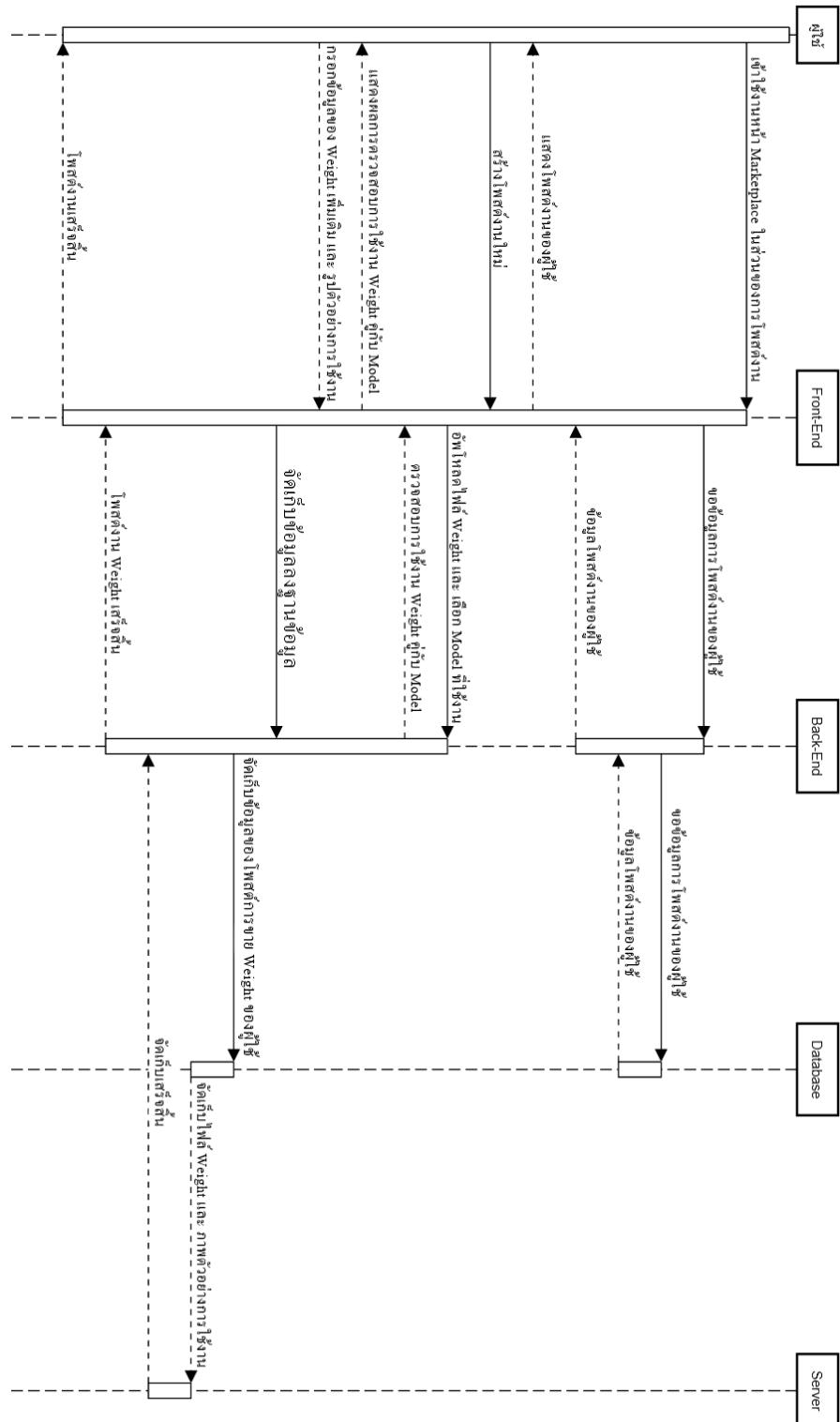
โดยรูปแบบลำดับปฏิสัมพันธ์นี้จะเริ่มต้นแต่เหตุการณ์ที่ ผู้ใช้งานเริ่มทำการจัดเก็บไฟล์ภาพไว้ใน Drive จนถึงการการจัดเก็บไฟล์ภาพเสร็จสิ้น โดยรูปภาพแสดงการทำงานแบบลำดับปฏิสัมพันธ์โดยรวมในส่วนของการจัดเก็บไฟล์ภาพไว้ใน Drive จะแสดงดังรูปที่ 3.13



รูป 3.13 รูปภาพแสดงการทำงานแบบลำดับปฏิสัมพันธ์โดยรวมในส่วนของการจัดเก็บไฟล์ภาพไว้ใน Drive

### 3.7.4 แผนภาพการทำงานในส่วนของการใช้งาน Marketplace สำหรับขาย Weight ที่ผู้ใช้ได้ทำการ Train (Web Application)

โดยรูปแบบลำดับปฏิสัมพันธ์นี้จะเริ่มต้นแต่เหตุการณ์ที่ การใช้งาน Marketplace สำหรับขาย Weight ที่ผู้ใช้ได้ทำการ Train จนถึงการโพสต์งานเสร็จสิ้น โดยรูปภาพแสดงการทำงานแบบลำดับปฏิสัมพันธ์โดยรวมในส่วนของการ Weight ที่ผู้ใช้ได้ทำการ Train จะแสดงดังรูปที่ 3.14



รูป 3.14 รูปภาพแสดงการใช้งาน Marketplace สำหรับขาย Weight ที่ผู้ใช้ได้ทำการ Train

### 3.8 การออกแบบส่วนต่อประสานกับผู้ใช้ (User Interface)

#### 3.8.1 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนลงชื่อเข้าใช้ (Login)

จะแสดงช่องข้อความสำหรับกรอก Email และ รหัสผ่านเพื่อที่จะนำไปยืนยันตัวตนสำหรับเข้าใช้งาน web application โดยการออกแบบในส่วนลงชื่อเข้าใช้ จะแสดงดังรูปที่ 3.15

รูป 3.15 การออกแบบส่วนลงชื่อเข้าใช้

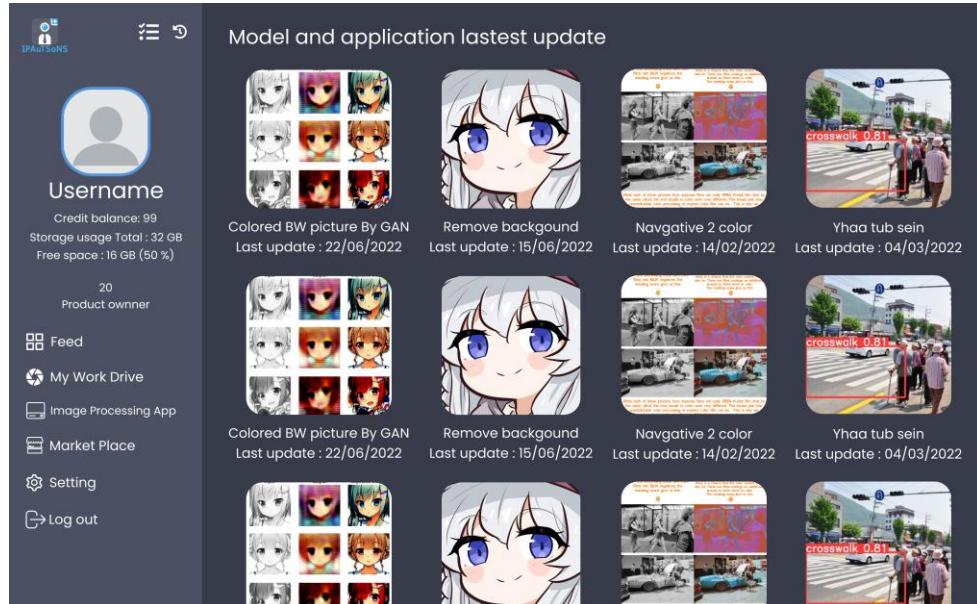
#### 3.8.2 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนลงทะเบียนสมัครสมาชิก (Register)

จะแสดงช่องข้อความสำหรับกรอก ชื่อ Username, Email, รหัสผ่าน, รหัสผ่านอีกครั้ง, ชื่อจริง และ นามสกุล เพื่อที่จะนำไปยืนยันตัวตนสำหรับเข้าใช้งาน web application โดยการออกแบบในส่วนลงทะเบียนสมัครสมาชิก จะแสดงดังรูปที่ 3.16

รูป 3.16 การออกแบบส่วนลงทะเบียนสมัครสมาชิก

### 3.8.3 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนหน้าข่าวสาร (Feed page)

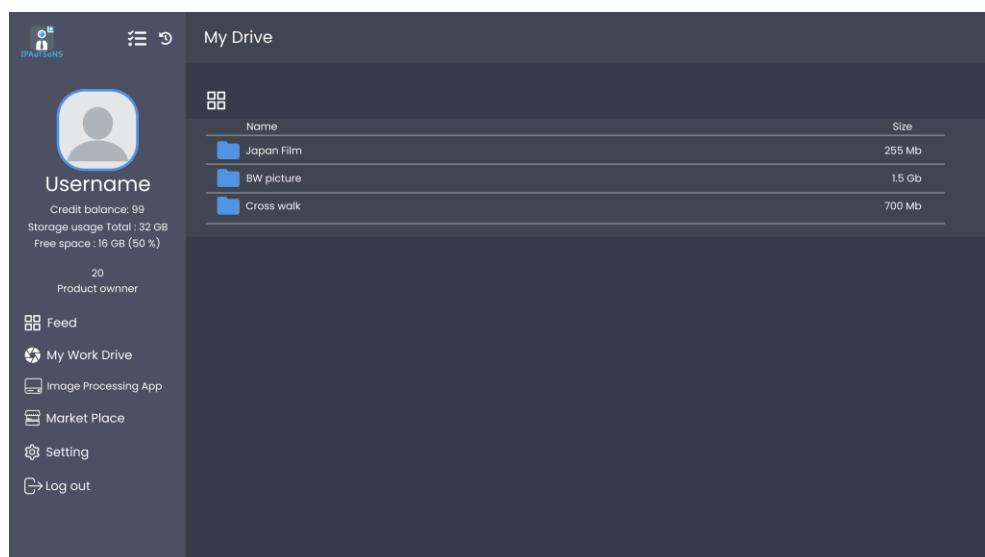
จะแสดงกล่องของข้อความที่ประกอบด้วยสินค้า Weight model ที่มีการ Update ล่าสุดแต่จะแสดงเพียงจำนวนหนึ่งเท่านั้น และจะมีการแนะนำให้ไปสามารถไปดูสินค้าอื่นๆเพิ่มเติมได้ที่หน้าตลาดซื้อขาย โดยการออกแบบในส่วนหน้าข่าวสาร จะแสดงดังรูปที่ 3.17



รูป 3.17 การออกแบบส่วนหน้าข่าวสาร

### 3.8.4 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนหน้าพื้นที่เก็บบันทึกรูปภาพ (Drive page)

จะแสดงลิสต์ของโฟลเดอร์ที่เก็บรูปภาพเพื่อให้ผู้ใช้เข้าถึงรูปภาพที่เก็บไว้อยู่ได้ โดยการออกแบบในส่วนพื้นที่เก็บบันทึกรูปภาพ จะแสดงดังรูปที่ 3.18



รูป 3.18 การออกแบบส่วนพื้นที่เก็บบันทึกรูปภาพ

### 3.8.5 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนพื้นที่แสดงรูปภาพที่เก็บบันทึกไว้ (Show Drive page)

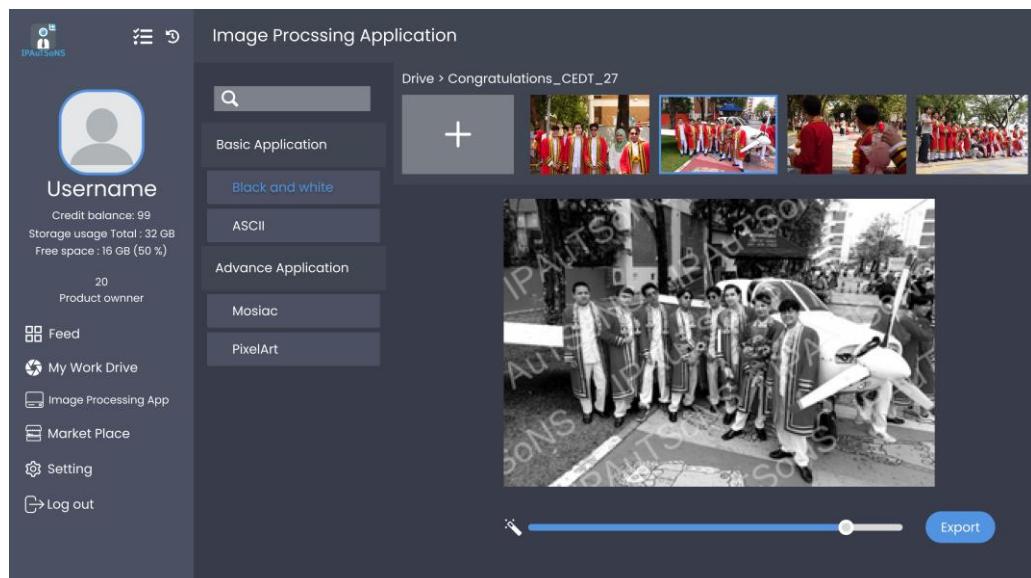
จะแสดงรูปภาพพร้อมรายละเอียดของรูปที่อยู่ในไฟล์เดอร์ที่เก็บรูปภาพนั้นๆ โดยการออกแบบในส่วนพื้นที่แสดงรูปภาพที่เก็บบันทึกไว้ จะแสดงดังรูปที่ 3.19



รูป 3.19 การออกแบบส่วนพื้นที่แสดงรูปภาพที่เก็บบันทึกไว้

### 3.8.6 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนแอพพลิเคชันประมวลผลภาพ (Image application page)

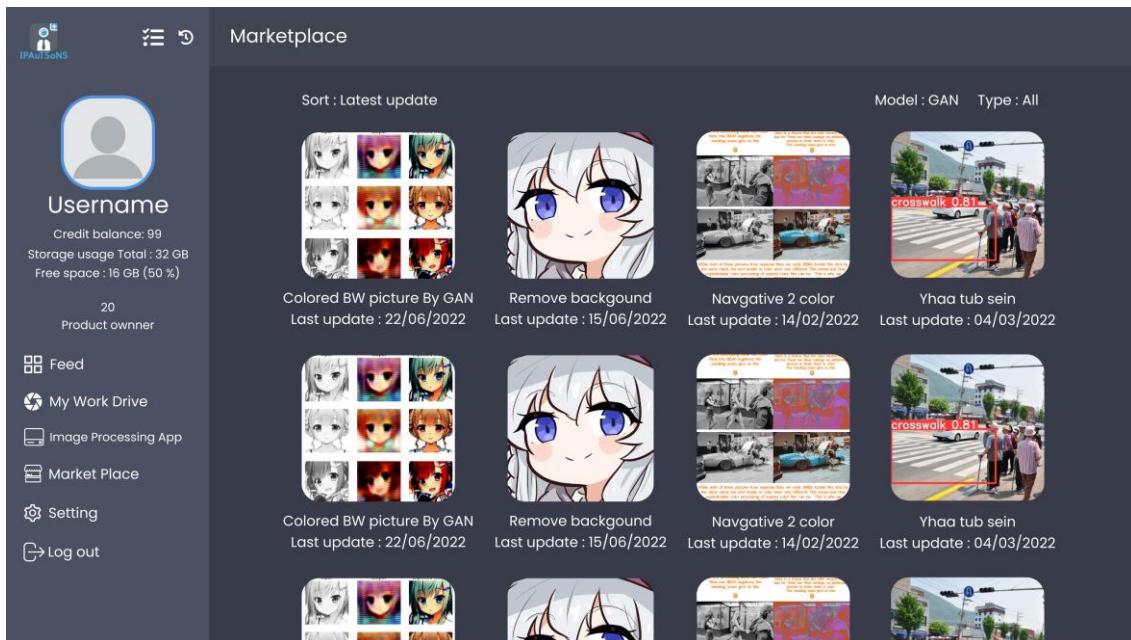
จะแสดงແນບເລືອກ Application การປ່ຽນແຕ່ງຮູບທີ່ດ້ານຊ້າຍ ແລະ ແສດງຜົດກາຣປ່ຽນແຕ່ງທີ່ທາງດ້ານພື້ນທີ່ຕ່າງກົາ ໂດຍທີ່ດ້ານນັນຈະເປັນພື້ນທີ່ເລືອກຮູບປາກທີ່ຕ້ອງກາຣ ໂດຍກາຣອອກແນບໃນສ່ວນ ແພພລິເຄີ່ນປະມວລຜົດກາພ ຈະແສດງດັງຮູບທີ່ 3.20



รูป 3.20 การออกแบบส่วนແພພລິເຄີ່ນປະມວລຜົດກາພ

### 3.8.7 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนพื้นที่ชี้ขาย (Marketplace page)

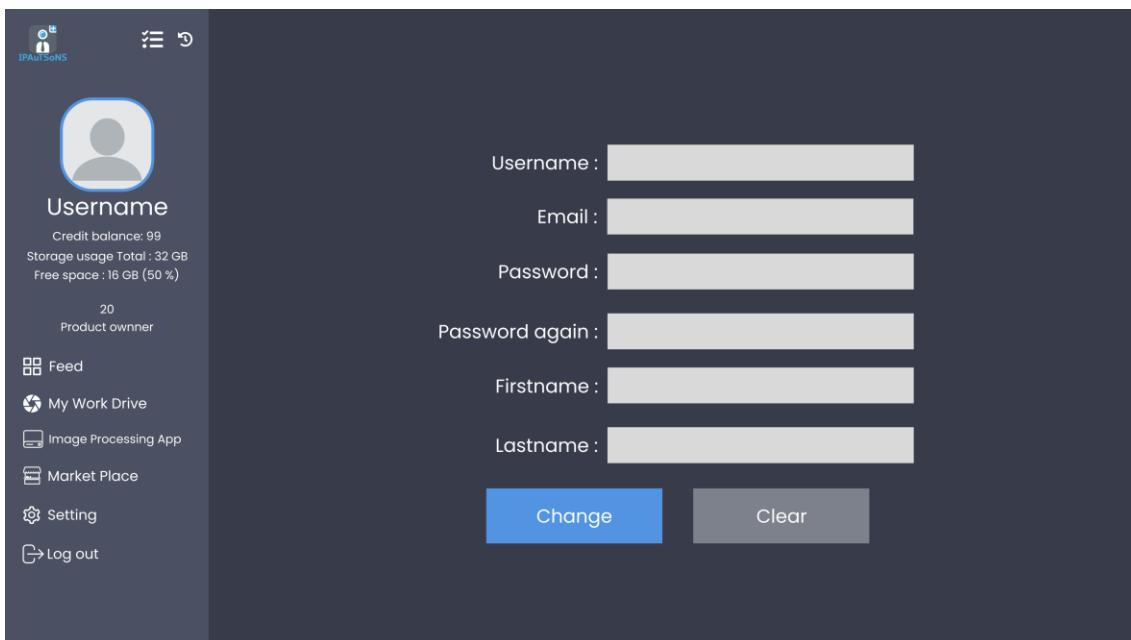
จะแสดงແຄບເລືອກ Application ການປ່ຽນແຕ່ງຮູບທີ່ດ້ານຊ້າຍ ໂດຍຈະເປັນການກວດເລືອກ Application ທີ່ສັນໃຈ ແລະ ໂດຍທີ່ຕຽດລາງຈະເປັນສິນຄ້າທີ່ນໍາມາຂາຍເຊັ່ນ Weight ທີ່ຝານກາຣ train ມາແລ້ວເປັນດັ່ນ ໂດຍການອອກແບນໃນສ່ວນພື້ນທີ່ເຊື້ອຂາຍ ຈະແສດງດັ່ງຮູບທີ່ 3.21



ຮູບ 3.21 ການອອກແບນສ່ວນພື້ນທີ່ເຊື້ອຂາຍ

### 3.8.8 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนตั้งค่าบัญชี (Setting page)

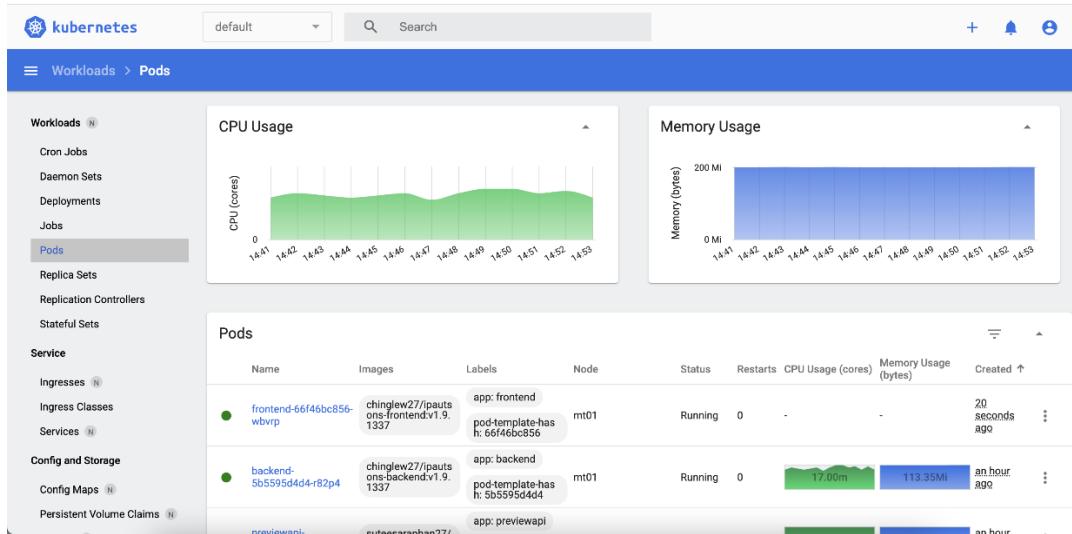
จะແສດງກລ່ອງຂໍຂວາມທີ່ມີຂໍ້ມູນຂອງຜູ້ໃຊ້ຍຸ່ງ ໂດຍສາມາດແກ່ໄໄ ໄດ້ຮັມໄປຄຶງຮູບໂປຣໄຟລ໌ ເອງທີ່ສາມາດເປີ່ຍິນ ໄດ້ ໂດຍການອອກແບນໃນສ່ວນຕັ້ງຄ່າບັນຫຼື ຈະແສດງດັ່ງຮູບທີ່ 3.22



ຮູບ 3.22 ການອອກແບນສ່ວນຕັ້ງຄ່າບັນຫຼື

### 3.8.9 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนตั้งค่าและคุณลักษณะของการใช้ทรัพยากร (Monitoring and Config page)

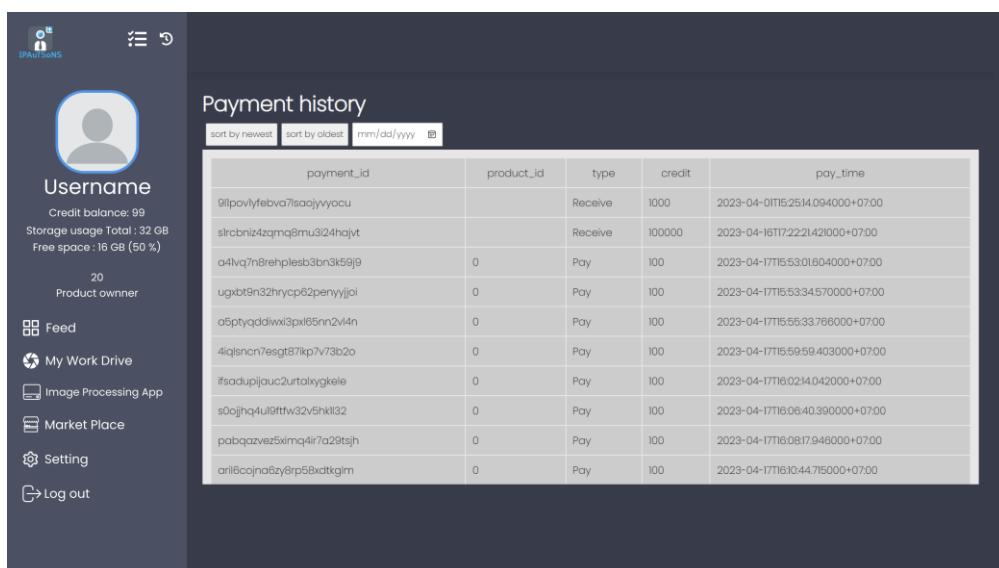
จะแสดงกราฟสถิติการใช้งาน hardware ต่างๆ เช่น CPU, GPU, RAM เป็นต้น โดยสามารถปรับแต่งการสั่งงานได้ภายในหน้านี้ โดยการออกแบบในส่วนตั้งค่าและคุณลักษณะของการใช้ทรัพยากร จะแสดงดังรูปที่ 3.23



รูป 3.23 การออกแบบส่วนตั้งค่าและคุณลักษณะของการใช้ทรัพยากร

### 3.8.10 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนคุณลักษณะของการใช้งานของผู้ใช้ (User history page)

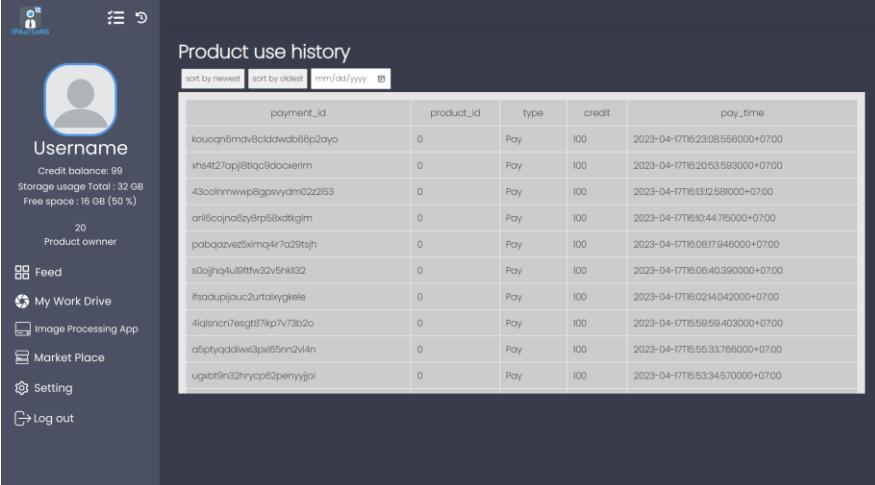
จะแสดงบันทึกการใช้งานสินค้าที่ใช้งานและเครดิตที่ใช้หรือได้รับของผู้ใช้ ซึ่งจะมีรูปแบบการเรียงลำดับและค้นหาตามวันที่ทำการให้ผู้ใช้เลือกเพื่อความสะดวกในเดือกดูข้อมูล โดยการออกแบบในส่วนคุณลักษณะของการใช้งานของผู้ใช้ จะแสดงดังรูปที่ 3.24



รูป 3.24 การออกแบบส่วนคุณลักษณะของการใช้งานของผู้ใช้

### 3.8.11 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนดูบันทึกการขายของสินค้า (Product history page)

จะแสดงบันทึกการที่สินค้าถูกนำไปส่งงานประมวลผล โดยผู้ใช้อื่นๆ ซึ่งจะมีรูปแบบการเรียงลำดับและค้นหาตามวันที่ทำการให้ผู้ใช้เลือกเพื่อความสะดวกในการเลือกดูข้อมูล โดยการออกแบบในส่วนดูบันทึกการขายของสินค้า จะแสดงดังรูปที่ 3.25



The screenshot shows a user interface for managing product usage history. On the left, there's a sidebar with a user profile picture, 'Username' (kouoqnn6mdv@cliddwdl66p2oy), 'Credit balance: 99', 'Storage usage Total: 32 GB', 'Free space: 18 GB (50%)', and a 'Product owner' section with a count of 20. Below these are links for 'Feed', 'My Work Drive', 'Image Processing App', 'Market Place', 'Setting', and 'Log out'. The main area is titled 'Product use history' and contains a table with the following data:

payment_id	product_id	type	credit	pay_time
kouoqnn6mdv@cliddwdl66p2oy	0	Pay	100	2023-04-17T16:23:08.566000+07:00
xhs4t27cap1tgc93dcowerm	0	Pay	100	2023-04-17T16:20:53.593000+07:00
43colnmwwp8gpsydm02z2i53	0	Pay	100	2023-04-17T16:13:12.580000+07:00
anil6cojnna6zy8rp56zatkglm	0	Pay	100	2023-04-17T16:04:47.750000+07:00
pabqazvez5imq4ir7o2itsjh	0	Pay	100	2023-04-17T16:08:17.946000+07:00
s0jijng4u0ltffw32z59kl32	0	Pay	100	2023-04-17T16:08:40.390000+07:00
ifadupijauc2urtoiyglele	0	Pay	100	2023-04-17T16:02:14.042000+07:00
4iqjsncn7esgt87kp7v73b2o	0	Pay	100	2023-04-17T15:59:59.403000+07:00
afptyyqddiwix0pxi5m5n2v14n	0	Pay	100	2023-04-17T15:55:33.768000+07:00
ugxbt9n32hrycp62penyyjoi	0	Pay	100	2023-04-17T15:53:34.570000+07:00

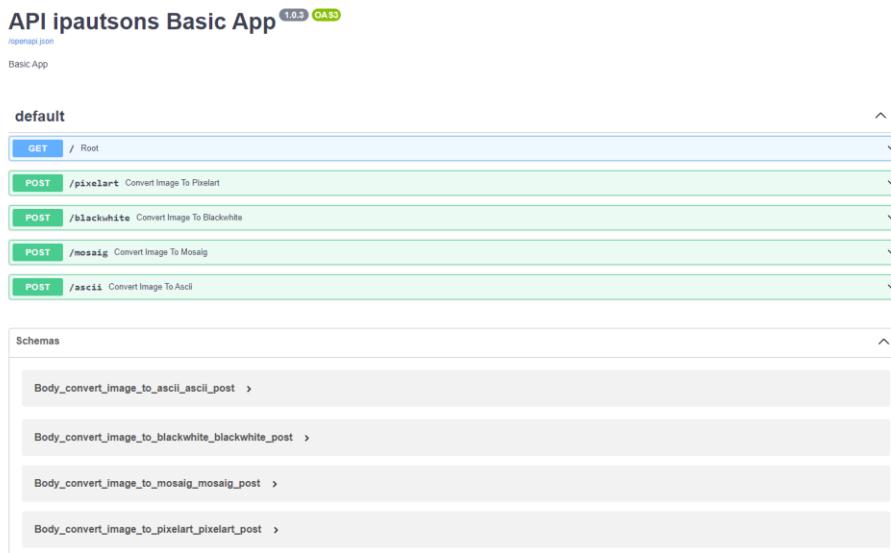
รูป 3.25 การออกแบบส่วนดูบันทึกการขายของสินค้า

### 3.8.12 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนของการแสดงผลตัวอย่างงานประมวลผลแอพพลิเคชันเบื้องต้น (API Previews Basic Application)

API สำหรับการประมวลผลแสดงตัวอย่างงานประมวลผลเบื้องต้น โดยจะมีแอพพลิเคชันดังต่อไปนี้ ASCII, BlackWhite, Pixelart, Mosaic ให้ผู้ใช้ทำการส่งไฟล์รูปภาพเข้ามาในระบบ และ จะส่งกลับคืนไปเป็นรูปภาพตามงานประมวลผลที่ได้เลือก เป็นการเปิดรับคำสั่งจากฝั่งของ Back-End ระบบเพื่อแสดงผลภาพจากงานประมวลผลที่จัดทำไว้โดยจะมีการเตรียม Model หรือ Library ต่างๆ เปิดใช้งานพร้อมรับคำสั่งประมวลผล โดยจะมี API Routing ดังนี้

1. /ascii ประมวลผลภาพเป็นรูปแบบ ASCII
2. /blackwhite ประมวลผลภาพเป็นรูปแบบ BlackWhite หรือ ขาวดำ
3. /pixelart ประมวลผลภาพเป็นรูปแบบ PixelArt โดยสามารถปรับแต่งขนาดของ Pixel ในภาพได้
4. /mosiac ประมวลผลภาพเป็นรูปแบบ Mosaic

โดย API Documents จะแสดงดังรูปที่ 3.26



รูป 3.26 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนของการแสดงผลตัวอย่างงานประมวลผล Basic App

### 3.8.13 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนของการแสดงผลตัวอย่างงานประมวลผล และงานประมวลผลด้วย Yolo Model (API Yolo Model Preview)

API ในการประมวลผลตรวจจับวัตถุในภาพ โดยใช้ YoloV5 Model ที่จะมี Weight พื้นฐานที่รองรับวัตถุ 80 ชนิด และสามารถให้ผู้ใช้ทำการอัปโหลด และเลือก Weight ที่มีอยู่ในระบบ เพื่อนำมาประมวลผลแทน Weight พื้นฐานได้ โดยสามารถตอบกลับเป็นในรูปแบบ Json และรูปภาพ เป็นการเปิดรับคำสั่งจากฝั่งของ Back-End ระบบเพื่อแสดงผลภาพจากการประมวลผลที่จัดทำไว้โดยจะ มีการเตรียม Model หรือ Library ต่าง ๆ เปิดใช้งานพร้อมรับคำสั่งประมวลผลโดยจะมี API Routing

1. /detect เป็นการ detect ภาพด้วย Weight Model ตัวเริ่มต้นที่ตั้งค่าไว้

2. /detect-to-json เป็นการ detect ภาพจาก Weight Model ที่เลือก และคืนค่ามาด้วย จำนวนของวัตถุ และชื่อวัตถุ ที่จับได้ในรูปแบบ Json

3. /detect-to-image เป็นการ detect ภาพจาก Weight Model ที่เลือก และ ส่งรูปภาพ กลับมาพร้อมกรอบวัตถุที่จับได้พร้อมด้วยความแม่นยำของวัตถุนั้น ๆ

โดย API Documents จะแสดงดังรูปที่ 3.27

Ipautsons API YoloModel Preview 1.0.0 OAS3

/openapi.json

Ipautsons API YoloModel Preview

**default**

**GET** / Root

**POST** /detect-to-json Detect Garbage Return Json Result

**POST** /detect-to-img Detect Garbage Return Base64 Img

**POST** /detect Detect Garbage

**Schemas**

- Body\_detect\_garbage\_detect\_post >
- Body\_detect\_garbage\_return\_base64\_img\_detect\_to\_img\_post >
- Body\_detect\_garbage\_return\_json\_result\_detect\_to\_json\_post >
- HTTPValidationError >
- ValidationError >

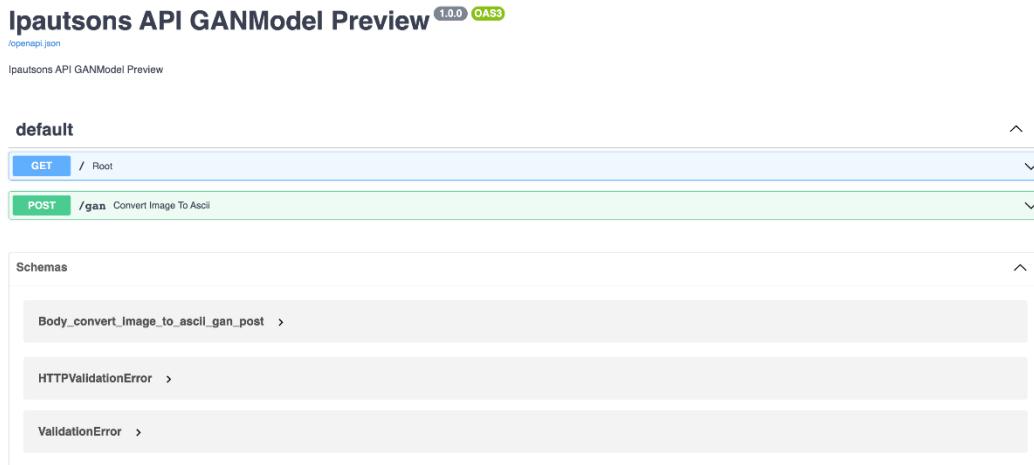
รูป 3.27 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนของการแสดงผลตัวอย่างงานประมวลผล Yolo Model

### 3.8.14 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนของการแสดงผลตัวอย่างงานประมวลผล และงานประมวลผล ด้วย GAN Model (API GAN Model Preview)

API ในการประมวลผลตรวจจับวัตถุในภาพโดยใช้ GAN Model ที่จะมี Weight พื้นฐานที่เปลี่ยนชนิดสี หรือ ลายเส้นของภาพ และ สามารถให้ผู้ใช้ทำการอัปโหลด และ เลือก Weight ที่มีอยู่ในระบบเพื่อนำมาประมวลผลแทน Weight พื้นฐานได้ โดยสามารถตอบกลับเป็นในรูปแบบภาพ เป็นการเปิดรับคำสั่งจากฝั่งของ Back-End ระบบเพื่อแสดงผลภาพจากการประมวลผลที่จัดทำไว้โดยจะมีการเตรียม Model หรือ Library ต่าง ๆ เปิดใช้งานพร้อมรองรับคำสั่งประมวลผลเปิดใช้งานพร้อมรองรับคำสั่งประมวลผลโดยจะมี API Routing

1. /gan เป็นการโหลดภาพขึ้นมาใหม่ด้วยภาพเดิม พร้อม Weight ที่ได้ทำการเลือกไว้ หรือ ตัวตั้งต้นที่ได้ตั้งค่าไว้

โดย API Documents จะแสดงดังรูปที่ 3.28



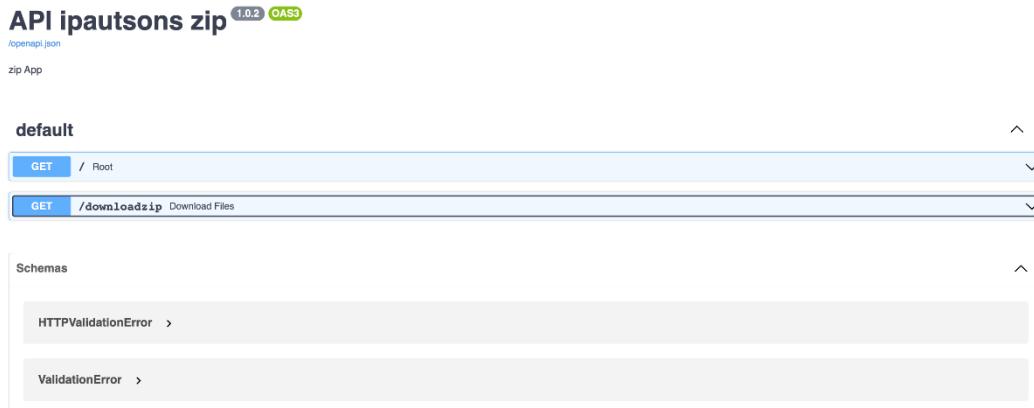
รูป 3.28 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนของการแสดงผลตัวอย่างงานประมวลผล GAN Model

### 3.8.15 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนของการบีบอัดไฟล์ของผู้ใช้ (API Zip file)

API ในการบีบอัดไฟล์ในโฟลเดอร์ของผู้ใช้ที่ให้บริการอยู่ในระบบผ่าน Persistent Volume โดยจะให้ผู้ใช้ทำการเลือกโฟลเดอร์ที่จะนำเข้ามา และ ส่งคืนกลับไปเป็นไฟล์ Zip เพื่อให้ผู้ใช้สามารถนำไฟล์ในโฟลเดอร์ไปใช้งานในงานอื่น ๆ ต่อได้ โดยจะมี API Routing

1. ./downloadzip เป็นการใส่ค่าตำแหน่งของโฟลเดอร์ที่ต้องการจะทำการบีบอัดไฟล์ในรูปแบบ Zip เพื่อประมวลผลบีบอัดส่งกลับมาเป็นไฟล์ Zip

จะแสดงดังรูปที่ 3.29



รูป 3.29 การออกแบบส่วนต่อประสานกับผู้ใช้ในส่วนของการบีบอัดไฟล์ Zip API

## บทที่ 4

### ผลการทดลอง / ความก้าวหน้า

#### 4.1 ส่วนของหน้างานทะเบียนสมัครเข้าใช้งาน

##### 4.1.1 การทดลองระบบการสมัครเข้าใช้งานของผู้ใช้งานระดับ Client

###### 4.1.1.1 จุดประสงค์การทดลอง

เพื่อทดสอบการสมัครเข้าใช้งานจากส่วนหน้า Register page หรือ หน้างานทะเบียนสมัครเข้าใช้งาน ว่าสามารถทำงานส่งข้อมูลผู้ใช้ที่กรอกเข้ามา แล้วส่งต่อไปยังส่วน Backend เพื่อทำการตรวจสอบ ข้อมูลผู้ใช้ และสถานะการมีอยู่ของผู้ใช้ ว่าเคยสมัครมาก่อนหน้านี้หรือไม่ รวมไปถึงการตรวจสอบ รหัสผ่านเบื้องต้น หากไม่มีข้อผิดพลาดใดๆ ให้ทำการเพิ่มผู้ใช้ไปยังฐานข้อมูล

###### 4.1.1.2 วิธีการทดลอง

ทดสอบผ่าน Web Application ที่พัฒนาขึ้นมา โดยการนำข้อมูลผู้ใช้จำนวนหนึ่งไปทดลอง กรอกลงในหน้างานทะเบียนสมัครเข้าใช้งาน และ กดสมัครการใช้งาน หากการทดลองสำเร็จ ข้อมูลจาก หน้า Web Application จะถูกส่งไปตรวจสอบที่ส่วน Backend ได้ และ ถ้าข้อมูลถูกต้องสามารถ ลงทะเบียนสมัครเข้าใช้งานได้ และมีข้อมูลของผู้ใช้ปรากฏขึ้นในฐานข้อมูล แต่ถ้าหากข้อมูลผู้ใช้ไม่ ถูกต้อง เช่น รหัสผ่านที่กรอกมาเพื่อยืนยันอีกครั้งไม่เหมือนกัน หรือ หากชื่อผู้ใช้หรืออีเมลที่สมัครเข้ามา มีอยู่แล้ว ในระบบ ก็จะแสดงข้อความตอบกลับว่าเกิดข้อผิดพลาดขึ้น

###### 4.1.1.3 ผลการทดลอง

ส่วนของการลงทะเบียนสมัครเข้าใช้งาน สามารถส่งข้อมูลจากหน้า Web Application ไปยังส่วน Backend ได้ถูกต้องและครบถ้วน และส่วน Backend สามารถตรวจสอบข้อมูลที่ทำการส่งเข้ามาได้ และ เพิ่มข้อมูลผู้ใช้เข้าในฐานข้อมูล ได้หากข้อมูลที่กรอกเข้ามา มีความสมบูรณ์ และ สามารถส่งข้อความแจ้ง เตือนกลับไปในกรณีที่ข้อมูลมีข้อผิดพลาด โดยการแสดงผลในแต่ละส่วนจะเป็นดังรูปที่ 4.1 - 4.3

Email :  
63015121@kmit.ac.th

Password :  
.....

Password confirm :  
.....

Firstname :  
Pasin

Lastname :  
Chantharathan

System check identified no issues (0 silenced).  
December 08, 2022 - 14:42:24  
Django version 4.0.6, using settings 'IPAutSoNsAPI.settings'  
Starting development server at http://127.0.0.1:8000/  
Quit the server with CTRL-BREAK.  
{'user\_id': 'lcpcls12cbqd', 'email': '63015121@kmit.ac.th', 'password': '123456789', 'is\_vip': 'false', 'first\_name': 'Pasin', 'last\_name': 'Chantharathan'}  
[08/Dec/2022 14:43:16] "POST /api/register HTTP/1.1" 200 120

รูป 4.1 ข้อมูลที่ส่งจาก Web application (บน) เทียบกับข้อมูลที่ส่วน Backend (ล่าง) ได้รับ

Connect View Collection Help

localhost:27017 ...

Documents ipautsons.api\_user +

My Queries Databases +

Search

- admin
- config
- ipautsons
  - \_\_schema\_\_
  - api\_folder\_img
  - api\_image\_file
  - api\_job
  - api\_user** ...
    - auth\_group
    - auth\_group\_permissions
    - auth\_permission
    - auth\_user
    - auth\_user\_groups
    - auth\_user\_user\_permissions
    - django\_admin\_log
    - django\_content\_type
    - django\_migrations

ipautsons.api\_user 2 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

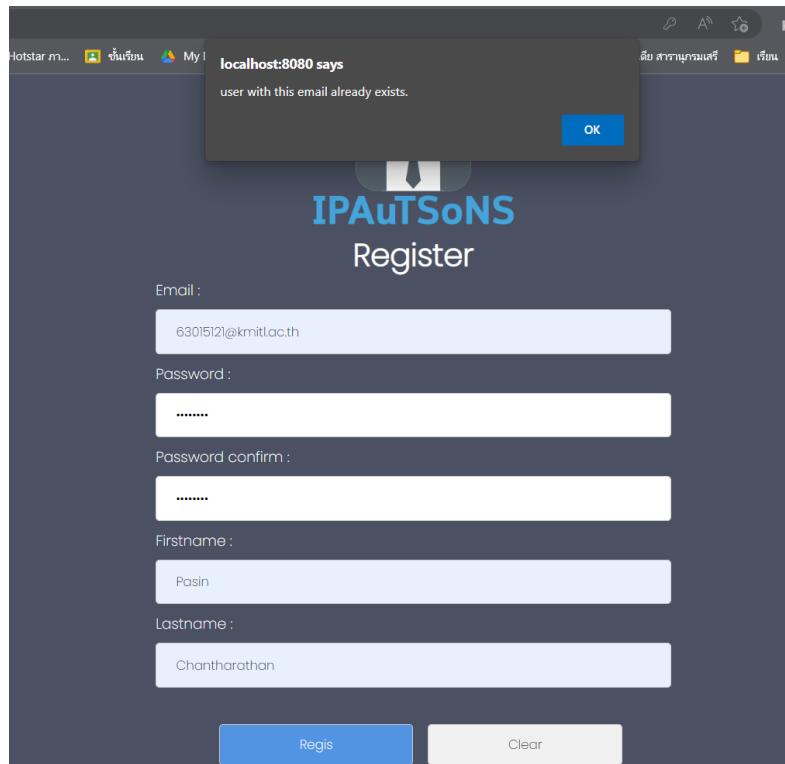
Filter Type a query: { field: 'value' } Reset Find More Options ▾

ADD DATA EXPORT COLLECTION 1 - 2 of 2 < > ⌂ ⌂ ⌂

```
_id: ObjectId('633e73a16b6268563235a9f3')
last_login: null
user_id: "i6r6nvxzco"
password: "pbkdf2_sha256$320000$YAx8sCTPZYcwRYtLzdWxCR$Gw39nf2ypQtvEL6IEa2pAUprj..."
email: "pasin98@gmail.com"
is_vip: false
first_name: "Pasin"
last_name: "Chantharathan"
```

```
_id: ObjectId('63919594c3c8722b008001dc')
last_login: null
user_id: "lcpcls12cbqd"
password: "pbkdf2_sha256$320000$0ZKKqP7FlxLgC4CTtty34$V/NvMPuqTHCl43NsNrBB8Yy0q9..."
email: "63015121@kmit.ac.th"
is_vip: false
first_name: "Pasin"
last_name: "Chantharathan"
```

รูป 4.2 ข้อมูลที่ทำการสมัครสำเร็จถูกเพิ่มในฐานข้อมูล



รูป 4.3 ข้อความที่แสดงหากอีเมลที่ใช้สมัครนั้นมีอยู่แล้วในฐานข้อมูล

## 4.2 ส่วนของหน้าลงทะเบียนเข้าสู่ระบบใช้งาน

### 4.2.1 การทดลองระบบลงทะเบียนเข้าสู่ระบบเพื่อผู้ใช้ระดับ Client เข้าใช้งาน Web application

#### 4.2.1.1 จุดประสงค์การทดลอง

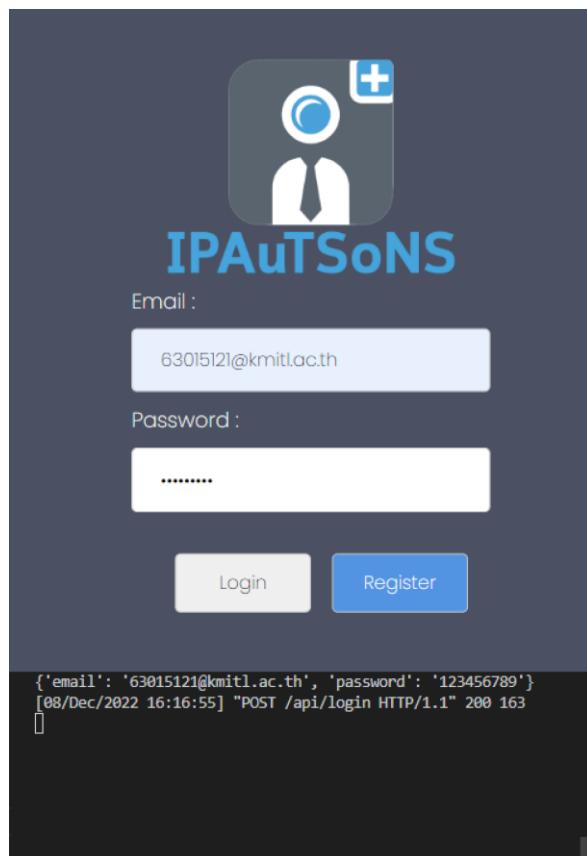
เพื่อทดสอบการทำงานของ Web application ส่วนหน้าลงทะเบียนเข้าสู่ระบบใช้งานว่าสามารถทำงานถูกต้อง สามารถส่งข้อมูลจากหน้าลงทะเบียนเข้าสู่ระบบไปยังส่วน Backend ได้อย่างครบถ้วน และนำข้อมูลที่ไปตรวจสอบเทียบกับข้อมูลผู้ใช้ในฐานข้อมูลเพื่อทำการยืนยันตัวตนผู้ใช้ได้ และสามารถแจ้งเตือนให้ผู้ใช้ได้หากมีข้อมูลส่วน个体พิเศษ และ สามารถส่ง Cookie กลับมายัง Web application เพื่อใช้ในการยืนยันตัวตนได้

#### 4.2.1.2 วิธีการทดลอง

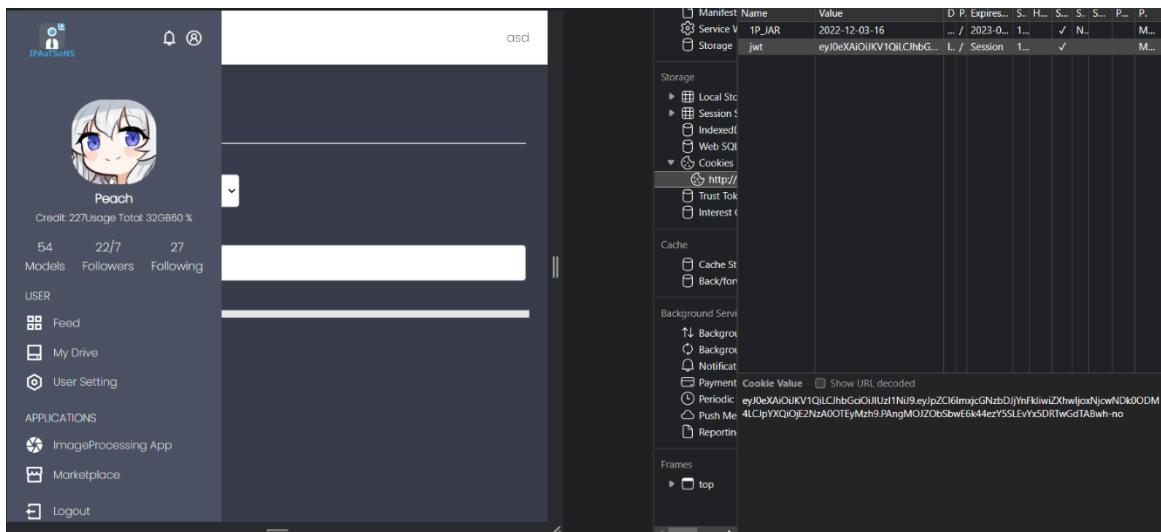
ทดสอบผ่าน Web Application ที่พัฒนาขึ้นมา โดยการนำข้อมูลผู้ใช้ที่มีและไม่มีในฐานข้อมูลจำนวนหนึ่งไปทดลองกรอกลงในหน้าลงทะเบียนเข้าสู่ระบบเข้าใช้งาน และ เข้าสู่ระบบ หากการทดลองสำเร็จ ข้อมูลผู้ใช้ที่ทำการลงทะเบียนเข้าสู่ระบบที่หน้า Web Application จะถูกส่งไปตรวจสอบที่ส่วน Backend ได้ และ ถ้าข้อมูลถูกต้องผู้ใช้จะเข้าใช้งาน Web applications ได้ แต่ถ้าหากข้อมูลผู้ใช้ไม่ถูกต้อง เช่น รหัสผ่านที่กรอกมาไม่เหมือนกับในฐานข้อมูล หรือ หากอีเมลที่กรอกเข้ามาเข้าสู่ระบบไม่มีภายในฐานข้อมูล ก็จะแสดงข้อความตอบกลับว่าเกิดข้อผิดพลาดขึ้น

#### 4.2.1.3 ผลการทดสอบ

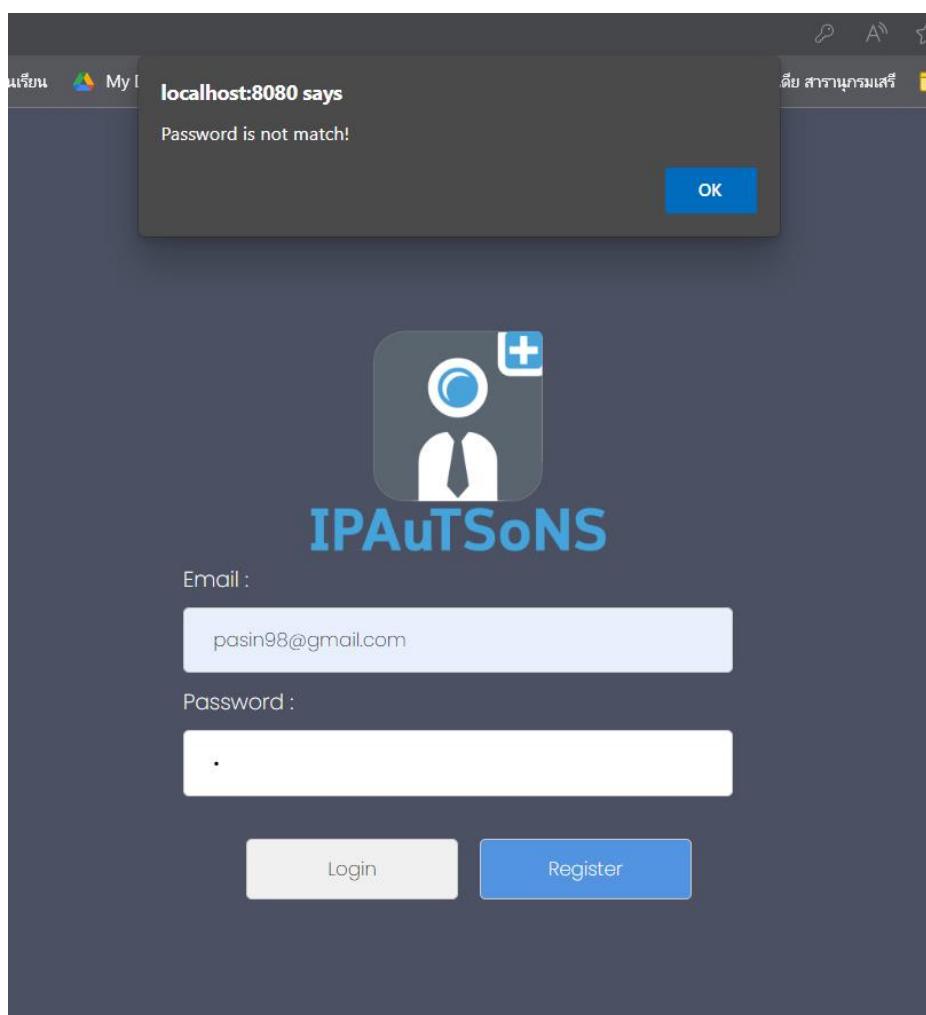
ส่วนของการลงทะเบียนเข้าสู่ระบบสามารถส่งข้อมูลจากหน้าเว็บ Web application ไปยังส่วน Back end ได้ถูกต้องและครบถ้วนและส่วน Backend สามารถตรวจสอบข้อมูลที่ได้ทำการส่งเข้ามาและยืนยันตัวตนโดยเบริบเนทีบกับข้อมูลของผู้ใช้ในฐานข้อมูล ได้หากข้อมูลของผู้ใช้ที่กรอกเข้ามานั้นมีความถูกต้องจะส่ง Cookie กลับไปยังเว็บแอปพลิเคชันเพื่อใช้ยืนยันตัวตนผู้ใช้ และ ในกรณีที่เกิดข้อผิดพลาดขึ้น เช่น ไม่พบ Email ของผู้ใช้ในระบบ หรือ รหัสผ่านผิดพลาดก็จะทำการส่งข้อความแจ้งเตือนกลับไปแสดงให้ผู้ใช้ทราบ โดยการแสดงผลในแต่ละส่วนจะเป็นดังรูปที่ 4.4 - 4.6



รูป 4.4 ข้อมูลที่ส่งจาก Web application (บน) เทียบกับข้อมูลที่ส่วน Backend (ล่าง) ได้รับ



รูป 4.5 ข้อมูล Cookie ของผู้ใช้หากทำการเข้าสู่ระบบสำเร็จ



รูป 4.6 ข้อความที่แสดงหากรหัสผ่านที่กรอกเข้าสู่ระบบนั้นไม่ถูกต้อง

## 4.3 ส่วนของหน้าแก้ไขข้อมูลผู้ใช้งาน

### 4.3.1 การทดลองระบบแก้ไขข้อมูลผู้ใช้ระดับ Client

#### 4.3.1.1 จุดประสงค์การทดลอง

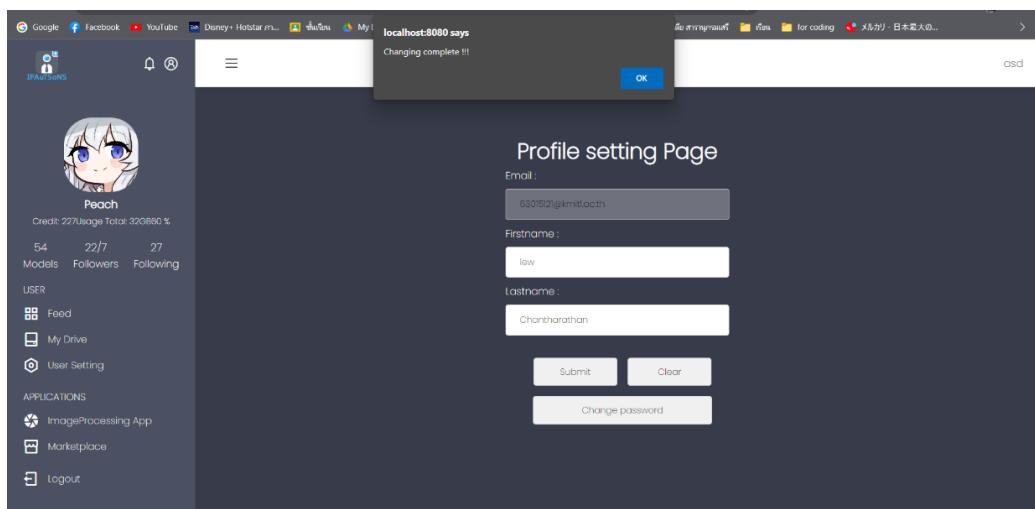
เพื่อทดสอบการทำงานของ Web applications ส่วนแก้ไขข้อมูลผู้ใช้ว่าสามารถทำงานถูกต้อง สามารถส่งข้อมูลจากหน้าแก้ไขข้อมูลผู้ใช้และ Cookie ยืนยันตัวตนไปยังส่วน Backend ได้อย่าง ครบถ้วนและนำข้อมูลที่ทำการแก้ไขไปเปลี่ยนแปลงข้อมูลในฐานข้อมูลได้ โดยสามารถแจ้งเตือนผู้ใช้ ได้หากมีข้อผิดพลาดเกิดขึ้น

#### 4.3.1.2 วิธีการทดลอง

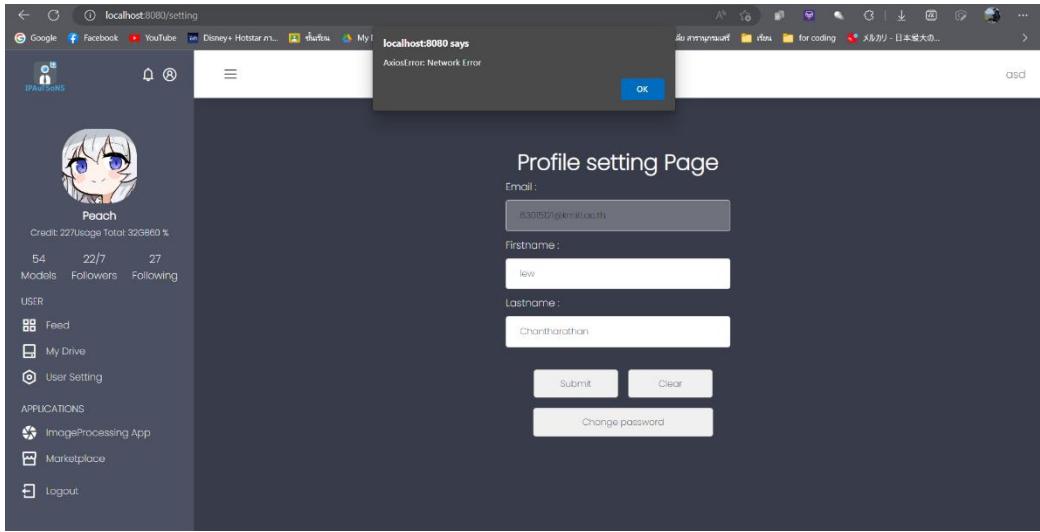
ทดสอบผ่าน Web Application ที่พัฒนาขึ้นมา โดยการนำชุดข้อมูลที่คล้ายกับข้อมูลผู้ใช้นำไป ทดลองกรอกลงในหน้าแก้ไขข้อมูลผู้ใช้และทำการกดเปลี่ยนแปลงแปลง หากสำเร็จข้อมูลที่ผู้ใช้กรอก เข้าไปเพื่อทำการเปลี่ยนแปลงและ Cookie ยืนยันตัวตนจะถูกส่งไปส่วนตรวจสอบที่ Backend ซึ่งส่วน Backend จะทำการยืนยันตัวตนของผู้ใช้หากสำเร็จจะทำการเปลี่ยนแปลงข้อมูลของผู้ใช้ในฐานข้อมูลแต่ ถ้าหากยืนยันตัวตนไม่สำเร็จหรือเกิดข้อผิดพลาดขึ้นก็จะแสดงข้อความตอบกลับ และทดลองปิดการ ทำงานของฐานข้อมูลเพื่อทดสอบการแจ้งเตือนหากมีข้อผิดพลาดเกิดขึ้น

#### 4.3.1.3 ผลการทดลอง

ส่วนของการแก้ไขข้อมูลผู้ใช้สามารถส่งข้อมูลจากหน้าเว็บ Web application ไปยังส่วน Backend ได้ถูกต้องและครบถ้วนและส่วน Backend สามารถตรวจสอบข้อมูลที่ได้ทำการส่งเข้ามา และ ยืนยันตัวตนโดยเปรียบเทียบ Cookie กับข้อมูลของผู้ใช้ในฐานข้อมูล และ เมื่อทำการยืนตัวตนเสร็จก็ทำการเปลี่ยนแปลงข้อมูลของผู้ใช้ในฐานข้อมูล และ เมื่อมีข้อผิดพลาดเกิดขึ้นก็จะทำการแจ้งเตือนให้ผู้ใช้ทราบโดยการแสดงผลในแต่ละส่วนจะเป็นดังรูปที่ 4.7 - 4.8



รูป 4.7 ข้อความที่แสดงหากสามารถเปลี่ยนข้อมูลผู้ใช้ได้สำเร็จ



รูป 4.8 ข้อความที่แสดงหากเกิดข้อผิดพลาดขึ้นจะเปลี่ยนแปลงข้อมูลผู้ใช้

#### 4.3.2 การทดลองระบบแก้ไขรหัสผ่านผู้ใช้ระดับ Client

##### 4.3.2.1 จุดประสงค์การทดลอง

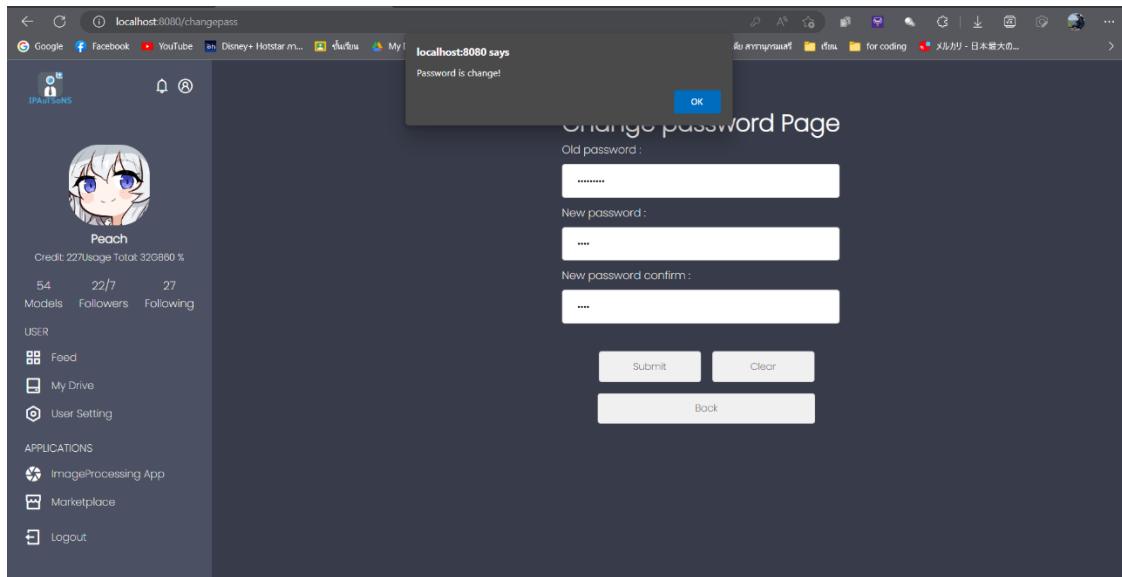
เพื่อทดสอบการทำงานของ Web applications ส่วนแก้ไขรหัสผ่านผู้ใช้ว่าสามารถทำงานถูกต้องสามารถส่งข้อมูลจากหน้าแก้ไขรหัสผ่านผู้ใช้และ Cookie ยืนยันตัวตนไปยังส่วน Backend ได้อย่างครบถ้วนและนำรหัสผ่านที่ทำการแก้ไขไปเปลี่ยนแปลงข้อมูลในฐานข้อมูลได้ โดยสามารถแจ้งเตือนผู้ใช้ได้หากมีข้อผิดพลาดเกิดขึ้น

##### 4.3.2.2 วิธีการทดลอง

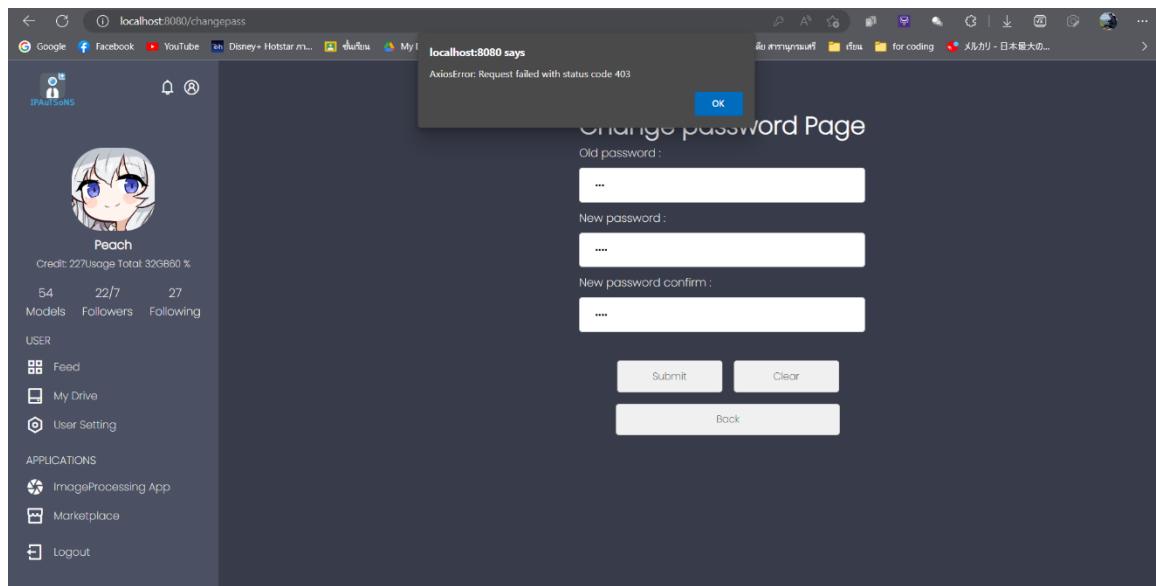
ทดสอบผ่าน Web Application ที่พัฒนาขึ้นมา โดยการนำชุดข้อมูลรหัสผ่านผู้ใช้เข้าไปทดลองกรอกลงในหน้าแก้ไขรหัสผ่านผู้ใช้และทำการกดเปลี่ยนแปลงแปลง หากสำเร็จรหัสผ่านที่ผู้ใช้กรอกเข้าไปเพื่อทำการเปลี่ยนแปลง และ Cookie ยืนยันตัวตนจะถูกส่งไปส่วนตรวจสอบที่ Backend ซึ่งส่วน Backend จะทำการยืนยันตัวตนของผู้ใช้หากสำเร็จจะทำการเปลี่ยนแปลงรหัสผ่านของผู้ใช้ในฐานข้อมูล แต่ถ้าหากยืนยันตัวตนไม่สำเร็จหรือเกิดข้อผิดพลาดขึ้นก็จะแสดงข้อความตอบกลับ และทดลองปิดการทำงานของฐานข้อมูลเพื่อทดสอบการแจ้งเตือนหากมีข้อผิดพลาดเกิดขึ้น

##### 4.3.2.3 ผลการทดลอง

ส่วนของการแก้ไขรหัสผ่านผู้ใช้สามารถส่งข้อมูลจากหน้าเว็บ Web application ไปยังส่วน Backend ได้ถูกต้องและครบถ้วนและส่วน Backend สามารถตรวจสอบรหัสผ่านที่ได้ทำการส่งเข้ามา และยืนยันตัวตนโดยเปรียบเทียบ Cookie กับรหัสผ่านของผู้ใช้ในฐานข้อมูล และเมื่อทำการยืนยันเสร็จก็ทำการเปลี่ยนแปลงรหัสผ่านของผู้ใช้ในฐานข้อมูล และเมื่อมีข้อผิดพลาดเกิดขึ้นก็จะทำการแจ้งเตือนให้ผู้ใช้ทราบโดยการแสดงผลในแต่ละส่วนจะเป็นดังรูปที่ 4.9 - 4.10



รูป 4.9 ข้อความที่แสดงหากสามารถเปลี่ยนข้อมูลผู้ใช้ได้สำเร็จ



รูป 4.10 ข้อความที่แสดงหากเกิดข้อผิดพลาดขึ้นขณะเปลี่ยนแปลงข้อมูลผู้ใช้

## 4.4 ส่วนของหน้าจัดการไฟล์

### 4.4.1 การทดลองระบบจัดการ Folder สำหรับเก็บไฟล์รูปภาพของผู้ใช้

#### 4.4.1.1 จุดประสงค์การทดลอง

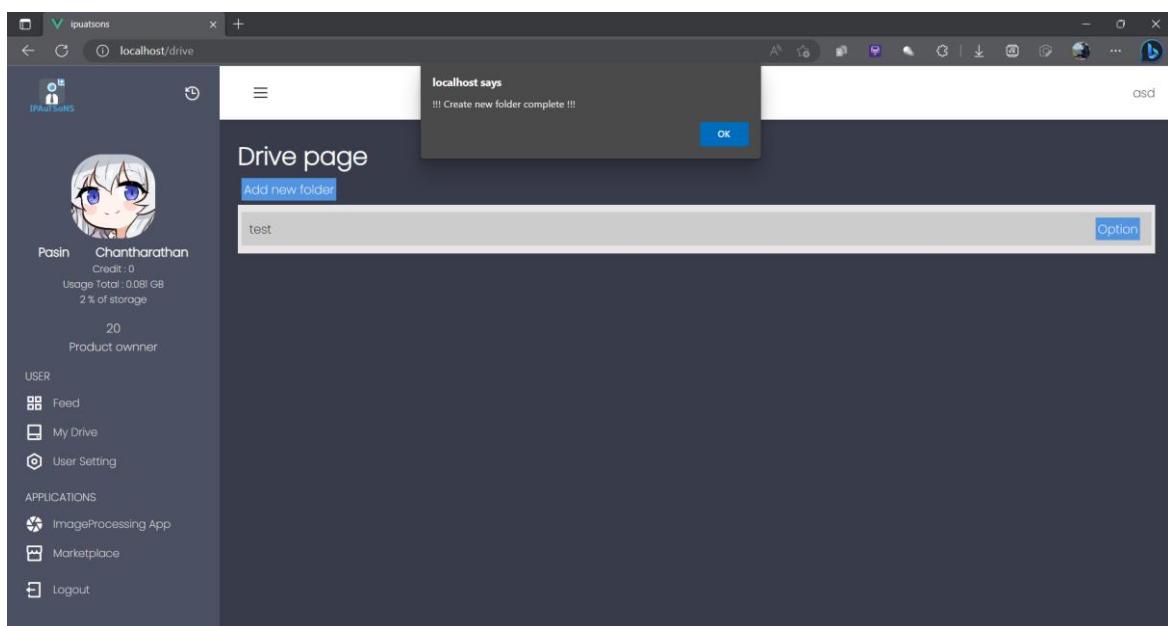
เพื่อทดสอบการทำงานของ Web applications หน้าจัดการไฟล์ในส่วนจัดการ Folder สำหรับเก็บไฟล์รูปภาพของผู้ใช้ ว่าสามารถทำงานถูกต้อง สามารถเพิ่ม Folder และ ลบ Folder ไปได้โดยมีการยืนยันตัวตนจาก Cookie ที่เก็บเอาไว้ โดยจะเพิ่ม-ลบข้อมูลของ Folder ที่ฐานข้อมูลด้วย และ มีการตรวจสอบชื่อ Folder โดยถ้าหากมีข้อมูลพลาดก็จะแสดงข้อความแจ้งเตือนข้อมูลพลาดบีบีนา

#### 4.4.1.2 วิธีการทดลอง

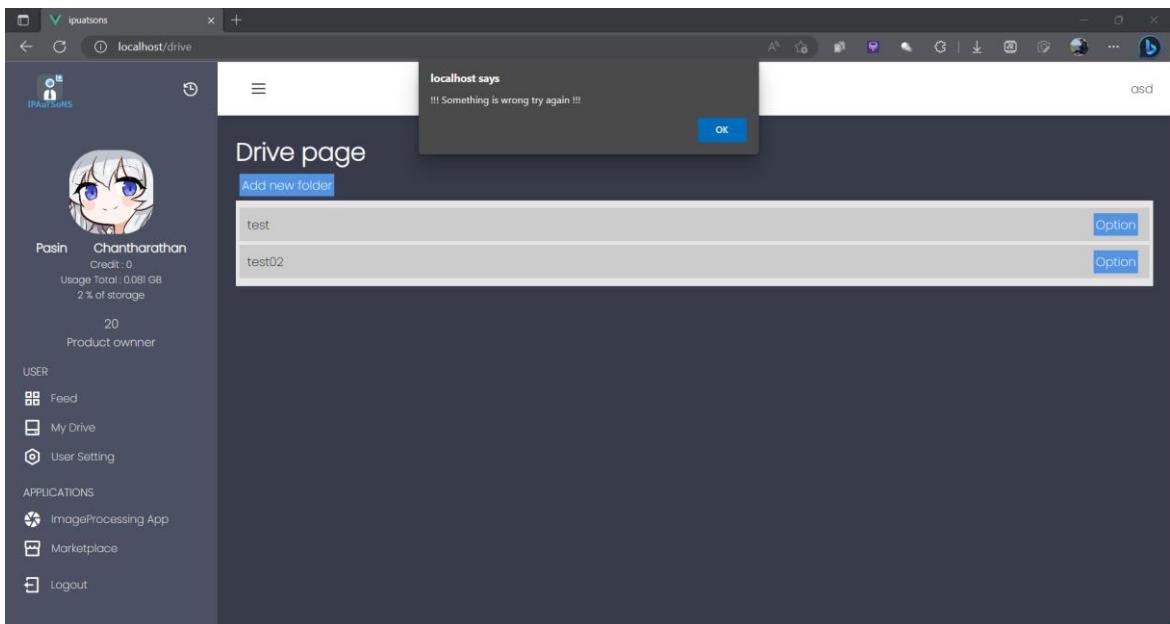
ทดสอบผ่าน Web Application ที่พัฒนาขึ้นมา โดยทำการเพิ่ม-ลบ Folder เก็บไฟล์ในหลายๆ รูปแบบแบบที่ต่างกันและซ้ำกัน หากทำงานได้สำเร็จจะมีข้อความแสดงผลบอกว่าทำการเพิ่ม-ลบ Folder สำเร็จ และทำการเพิ่ม-ลบข้อมูลของ Folder ในฐานข้อมูล โดยที่การทำงานหากมีข้อผิดพลาดก็จะส่งแสดงผลข้อความว่าการทำงานนั้นไม่สำเร็จ

#### 4.4.1.3 ผลการทดลอง

ส่วนของระบบจัดการ Folder สำหรับเก็บไฟล์รูปภาพสามารถส่งข้อมูลจากหน้าเว็บ Web application ไปยังส่วน Backend ได้ถูกต้องและครบถ้วนและส่วน Backend สามารถยืนยันตัวตนผู้ใช้ที่ได้ทำการส่งเข้ามา และทำการ เพิ่ม-ลบ Folder เก็บรูปภาพได้ และสามารถตรวจสอบชื่อ Folder ที่ซ้ำกันได้รวมไปถึงทำการเพิ่ม-ลบข้อมูลของ Folder ในฐานข้อมูลได้ และสามารถส่งข้อความแจ้งเตือนได้หากเกิดการทำงานที่ผิดพลาดขึ้น โดยการแสดงผลในแต่ละส่วนจะเป็นดังรูปที่ 4.11 - 4.12



รูป 4.11 ข้อความที่แสดงหากสามารถเพิ่ม Folder สำหรับเก็บรูปภาพได้



รูป 4.12 ข้อความที่แสดงหากเกิดข้อผิดพลาดขณะเพิ่ม Folder สำหรับเก็บรูปภาพ

#### 4.4.2 การทดลองระบบจัดการไฟล์รูปภาพของผู้ใช้

##### 4.4.2.1 จุดประสงค์การทดลอง

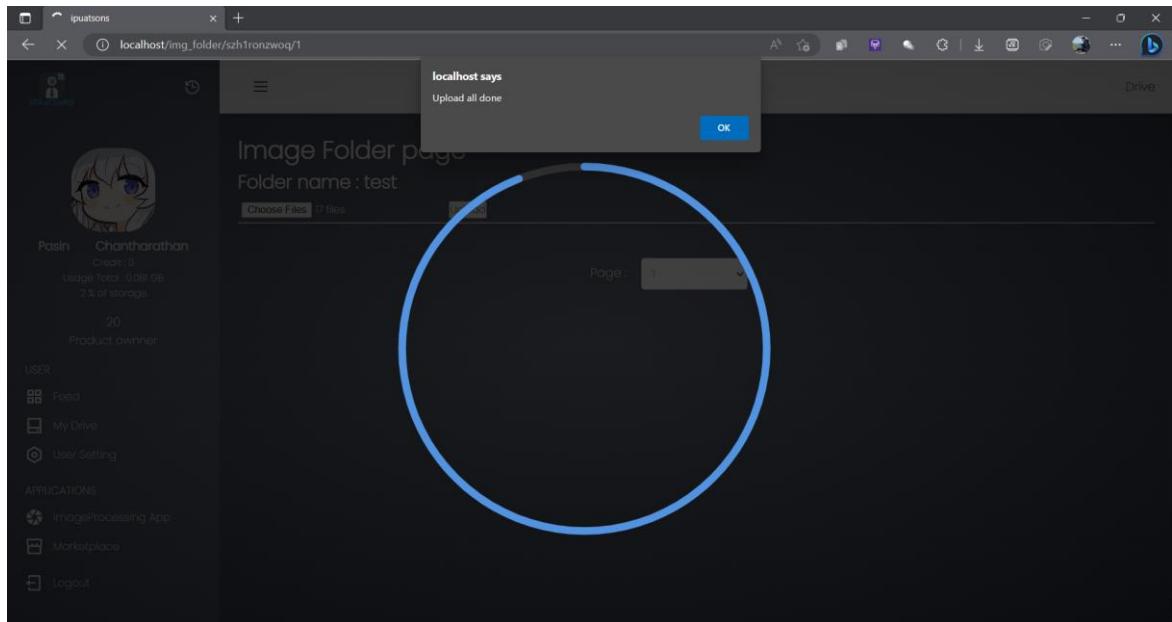
เพื่อทดสอบการทำงานของ Web applications หน้าจัดการไฟล์ในส่วนจัดการไฟล์รูปภาพ ของผู้ใช้ว่าสามารถทำงานถูกต้อง สามารถเพิ่ม รูปภาพ และ ลบ รูปภาพ ไปได้โดยมีการส่ง Cookie ไปยัง Backend เพื่อยืนยันตัวตน โดยจะเพิ่ม-ลบข้อมูลของ รูปภาพ ที่ฐานข้อมูลด้วย โดยถ้าหากมีข้อผิดพลาดก็ จะแสดงข้อความแจ้งเตือนข้อผิดพลาดขึ้นมา

##### 4.4.2.2 วิธีการทดลอง

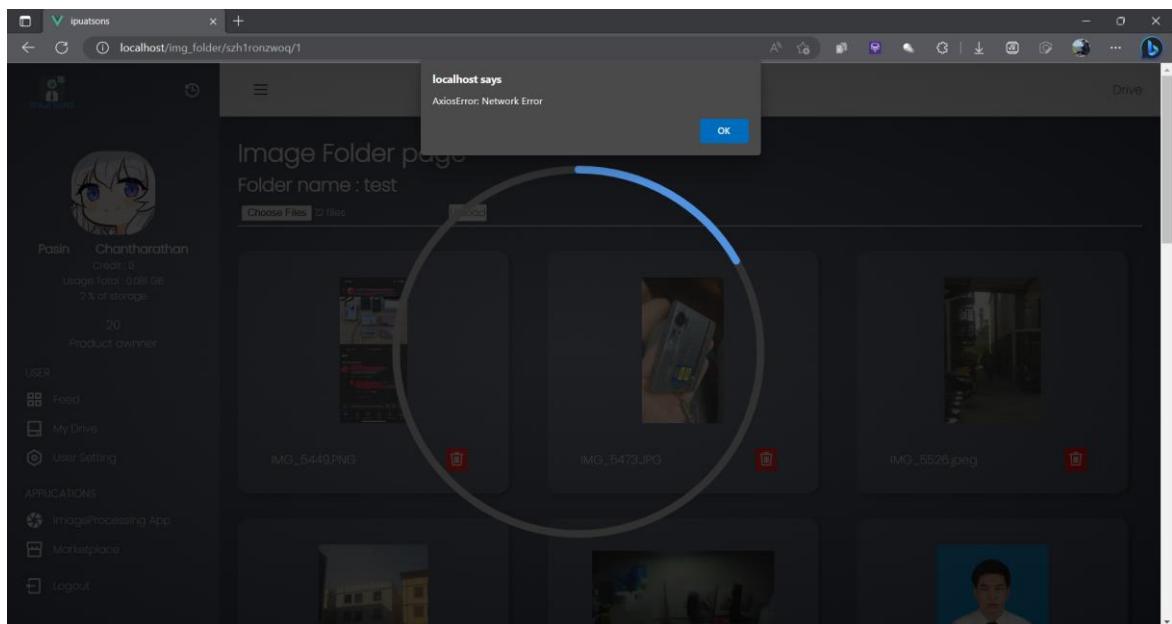
ทดสอบผ่าน Web Application ที่พัฒนาขึ้นมา โดยทำการเพิ่ม-ลบ Folder เก็บไฟล์ในหลายๆ รูปแบบแบบที่ต่างกันและซ้ำกัน หากทำงานได้สำเร็จจะมีข้อความแสดงผลบวกว่าทำการเพิ่ม-ลบ Folder สำเร็จ และ ทำการเพิ่ม-ลบข้อมูลของ Folder ในฐานข้อมูล โดยที่การทำงานหากมีข้อผิดพลาดก็จะส่ง แสดงผลข้อความว่าการทำงานนั้นไม่สำเร็จ

##### 4.4.2.3 ผลการทดลอง

ส่วนของระบบจัดการไฟล์รูปภาพสามารถส่งข้อมูลจากหน้าเว็บ Web application ไปยังส่วน Backend ได้ถูกต้องและครบถ้วนและส่วน Backend สามารถยืนยันตัวตนผู้ใช้ที่ได้ทำการส่งเข้ามา และ ทำการ เพิ่ม-ลบรูปภาพ ได้ รวมไปถึงทำการเพิ่ม-ลบข้อมูลของรูปภาพในฐานข้อมูล ได้ และสามารถส่ง ข้อความแจ้งเตือนได้หากเกิดการทำงานที่ผิดพลาดขึ้น โดยการแสดงผลในแต่ละส่วนจะเป็นดังรูปที่ 4.13



รูป 4.13 ข้อความที่แสดงหากสามารถเพิ่มรูปภาพสำเร็จ



รูป 4.14 ข้อความที่แสดงหากเกิดข้อผิดพลาดขณะเพิ่มรูปภาพ

## 4.5 ส่วนของหน้าสั่งงานประมวลผล

### 4.5.1 การทดลองระบบการสร้างไฟล์รูปภาพที่ผ่านการประมวลผลเพื่อทดสอบการใช้งานฟิลเตอร์

#### 4.5.2.1 จุดประสงค์การทดลอง

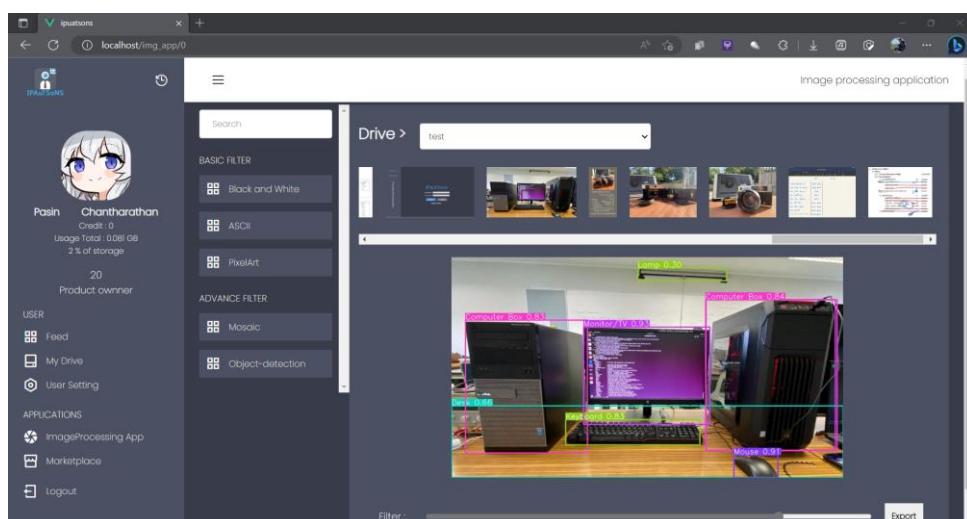
เพื่อทดสอบการทำงานของ Web applications หน้าสั่งงานประมวลผลในส่วนระบบการสร้างไฟล์รูปภาพที่ผ่านการประมวลผลเพื่อทดสอบการใช้งานฟิลเตอร์ เพื่อให้ผู้ใช้ได้เห็นผลลัพธ์ก่อนสั่งงานจริง และสามารถทำงานถูกต้อง สามารถสั่งงานประมวลผลภาพได้โดยที่ Web application ส่งข้อมูลไปยังส่วน Backend และมีการส่ง Cookie ไปยังส่วน Backend เพื่อยืนยันตัวตน และ ข้อมูลรูปภาพกับข้อมูลฟิลเตอร์ที่ใช้ และส่วน Backend จะส่งต่อไปยัง Pod ของ Docker ที่ได้มีการเปิดการทำงานเอาไว้ผ่าน API ซึ่งจะส่งรูปภาพที่ประมวลผลเสร็จแล้วกลับมาเพื่อให้ Web application นำไปแสดงผลต่อไป โดยถ้าหากมีข้อผิดพลาดก็จะแสดงข้อความแจ้งเตือนข้อผิดพลาดขึ้นมา

#### 4.5.2.1 วิธีการทดลอง

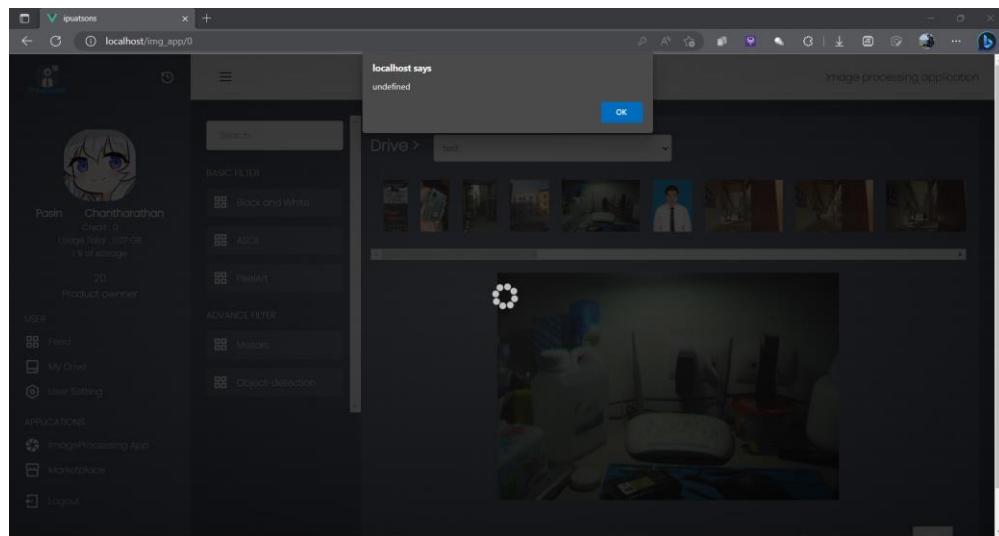
ทดสอบผ่าน Web Application ที่พัฒนาขึ้นมา โดยทำการใช้งานฟิลเตอร์หลากหลายรูปแบบ กับหลากหลายรูป หากทำงานได้สำเร็จจะมีรูปภาพที่ผ่านการประมวลผลแสดงขึ้นมา แต่ถ้าหากมีข้อผิดพลาด ก็จะมีการแสดงข้อความเตือนปรากฏขึ้นมา

#### 4.5.2.1 ผลการทดลอง

ส่วนของระบบการสร้างไฟล์รูปภาพที่ผ่านการประมวลผลเพื่อทดสอบการใช้งานฟิลเตอร์ สามารถส่งข้อมูลจากหน้าเว็บ Web application ไปยังส่วน Backend ได้ถูกต้องและครบถ้วนและส่วน Backend สามารถยืนยันตัวตนผู้ใช้ที่ได้ทำการสั่งเข้ามา และ ทำการส่งต่อข้อมูลรูปภาพและฟิลเตอร์ที่ใช้งานไปยัง Pod เพื่อทำการประมวลผลได้สำเร็จ และ ส่งรูปภาพกลับมาให้ Web application นำมาแสดงผล ได้ และ หากการทำงานผิดพลาดก็มีการแสดงข้อความเตือนปรากฏขึ้นมา โดยการแสดงผลในแต่ละส่วน จะเป็นดังรูปที่ 4.15 - 4.16



รูป 4.15 รูปที่ผ่านการประมวลผลที่แสดงหากประมวลผลสำเร็จ



รูป 4.16 ข้อความที่แสดงหากเกิดข้อผิดพลาดขณะสร้างไฟล์งานประมวลผล

#### 4.5.2 การทดลองระบบการสร้างไฟล์การสั่งงานประมวลผลภาพ

##### 4.5.2.1 จุดประสงค์การทดลอง

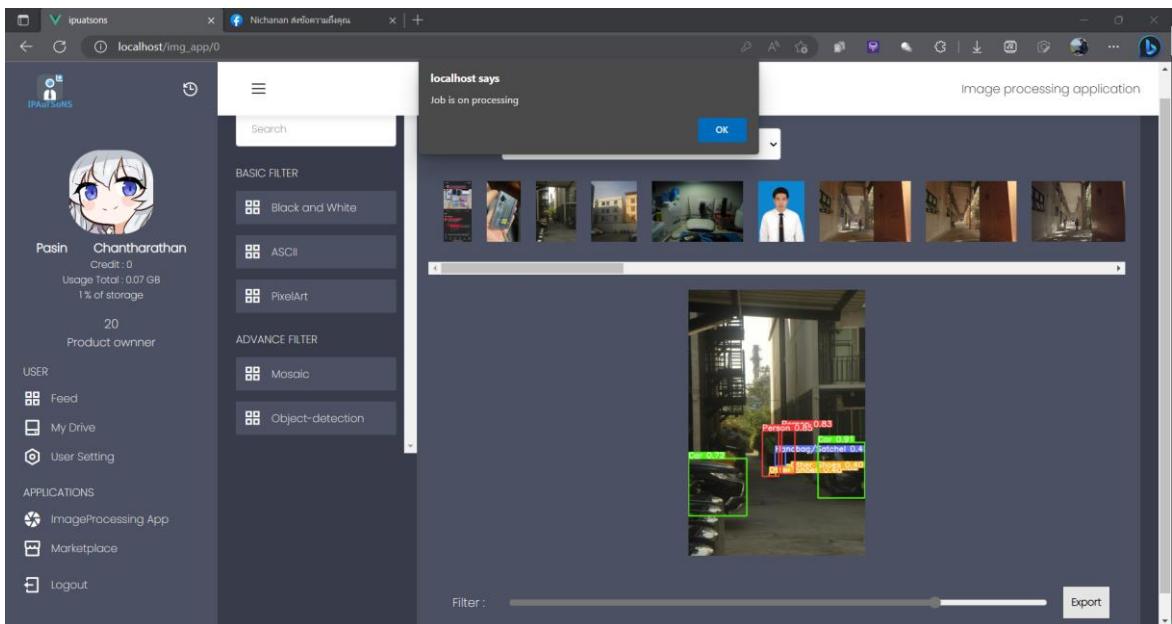
เพื่อทดสอบการทำงานของ Web applications หน้าสั่งงานประมวลผลในส่วนระบบการสร้างไฟล์ YAML เพื่อสั่งงานประมวลผลภาพของผู้ใช้ว่าสามารถทำงานถูกต้อง สามารถสั่งงานประมวลผลภาพได้โดยที่ Web application ลั่งข้อมูลไปยังส่วน Backend และมีการส่ง Cookie ไปยังส่วน Backend เพื่อยืนยันตัวตน โดยจะเพิ่มข้อมูลของ งานประมวลผล ที่ฐานข้อมูลด้วย โดยถ้าหากมีข้อผิดพลาดก็จะแสดงข้อความแจ้งเตือนข้อผิดพลาดขึ้นมา

##### 4.5.2.2 วิธีการทดลอง

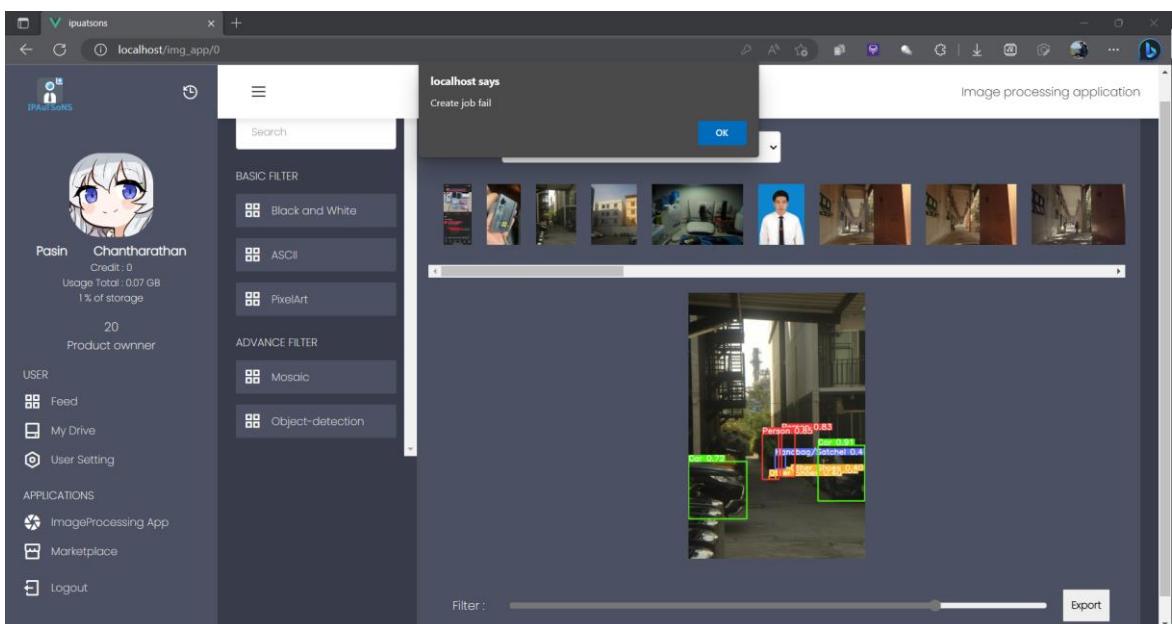
ทดสอบผ่าน Web Application ที่พัฒนาขึ้นมา โดยทำการเพิ่มงานประมวลผลภาพ ในหลายๆ รูปแบบที่ต่างกัน หากทำงานได้สำเร็จจะมีข้อความแสดงผลบอกว่าทำการสร้างงานประมวลผลสำเร็จ และ ทำการเพิ่มข้อมูลของ งานประมวลผล ในฐานข้อมูล โดยที่การทำงานหากมีข้อผิดพลาดก็จะสั่งแสดงผลข้อความว่าการทำงานนั้นไม่สำเร็จ

##### 4.5.2.3 ผลการทดลอง

ส่วนของระบบจัดการไฟล์รูปภาพสามารถส่งข้อมูลจากหน้าเว็บ Web application ไปยังส่วน Backend ได้ถูกต้องและครบถ้วนและส่วน Backend สามารถยืนยันตัวตนผู้ใช้ที่ได้ทำการส่งเข้ามา และทำการ เพิ่ม-ลบรูปภาพได้ รวมไปถึงทำการเพิ่ม-ลบข้อมูลของรูปภาพในฐานข้อมูลได้ และสามารถสั่งข้อความแจ้งเตือนได้หากเกิดการทำงานที่ผิดพลาดขึ้น โดยการแสดงผลในแต่ละส่วนจะเป็นดังรูปที่ 4.17



รูป 4.17 ข้อความที่แสดงหากสามารถสร้างไฟล์งานประมวลผลสำเร็จ



รูป 4.18 ข้อความที่แสดงหากเกิดข้อผิดพลาดขณะสร้างไฟล์งานประมวลผล

## 4.6 ส่วนของหน้าต่างด้วยชื่อขาย

### 4.6.1 การทดลองระบบการเพิ่มสินค้าในตลาดด้วยชื่อขาย

#### 4.6.1.1 จุดประสงค์การทดลอง

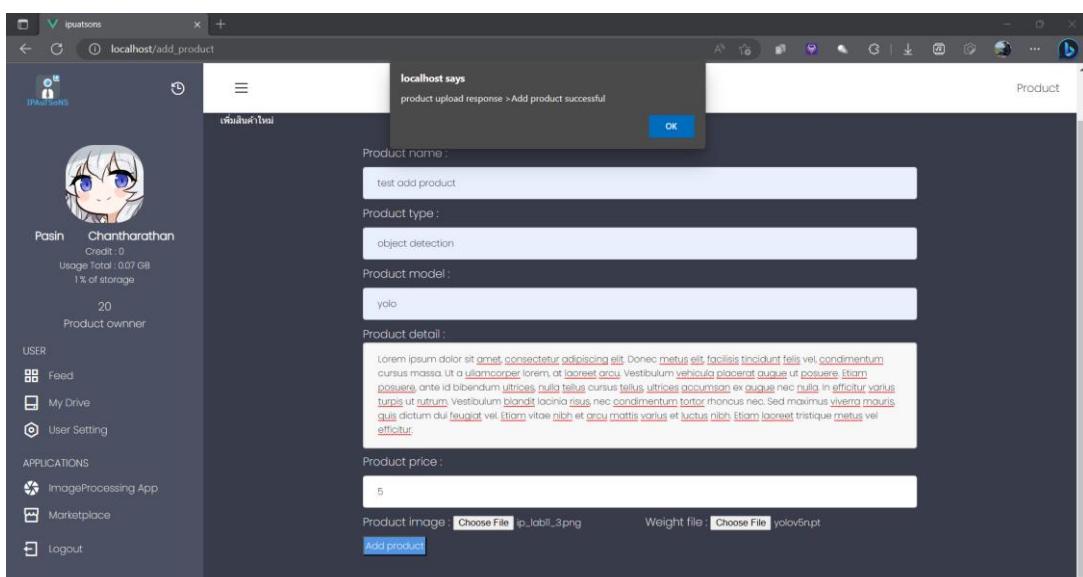
เพื่อทดสอบการใช้งานระบบการเพิ่มสินค้าว่ามีการทำงานที่ถูกต้อง โดยจะสามารถส่งข้อมูลสินค้า และ ตัว Weight ของ model และรูปภาพของการใช้สินค้า ที่ผู้ใช้กรอกเข้ามาจากหน้า Frontend ไปยัง Backend ได้และทดสอบการทำงานของตัว Weight ได้ โดยที่หากการทำงานสำเร็จ ข้อมูลของสินค้าถูกเพิ่มไปที่ฐานข้อมูลด้วย และ หากการทำงานไม่สำเร็จก็จะมีการแสดงข้อความแจ้งเตือนผู้ใช้

#### 4.6.1.2 วิธีการทดลอง

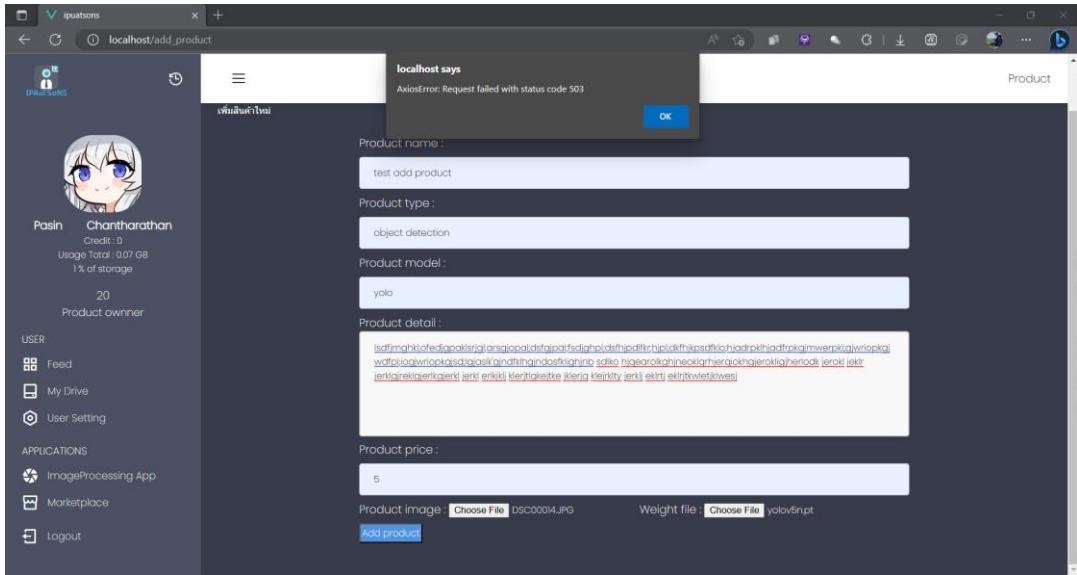
ทดสอบโดยการเพิ่มสินค้าด้วยข้อมูลที่แตกต่างกันหลายๆ ครั้ง โดยจะตรวจสอบข้อมูลที่ได้รับมาและถูกเพิ่มในฐานข้อมูลว่ามีความครบถ้วนและถูกต้องตรงตามรูปแบบที่ได้มีการออกแบบเอาไว้ และถ้าหากการทำงานเกิดข้อผิดพลาดขึ้น ก็จะมีการแสดงข้อความแจ้งเตือนผู้ใช้ว่าการเพิ่มสินค้านั้นไม่สำเร็จ

#### 4.6.1.3 ผลการทดลอง

จากการการทดลองระบบการเพิ่มสินค้าในตลาดด้วยของ Web application สามารถเพิ่มสินค้า และสามารถตรวจสอบ Weight ของ Model ที่ Upload เข้ามาโดยหลังการทำงานเสร็จสิ้นจะแสดงข้อความว่าผู้ใช้นี้เพิ่มสินค้าสำเร็จและแสดงหน้าต่างหน้า Marketplace ขึ้นมา และในการทำงานที่ไม่สำเร็จนั้นก็สามารถแสดงข้อความกลับมาให้ผู้ใช้ได้รับทราบ โดยการแสดงผลแต่ละส่วนสามารถดูได้จากรูปที่ 4.19 และ 4.20



รูป 4.19 ข้อความที่แสดงหากสามารถเพิ่มสินค้าได้สำเร็จ



รูป 4.20 ข้อความที่แสดงหากไม่สามารถเพิ่มสินค้าได้

#### 4.6.2 การทดลองระบบการแก้ไขข้อมูลสินค้าในตลาดซื้อขาย

##### 4.6.2.1 จุดประสงค์การทดลอง

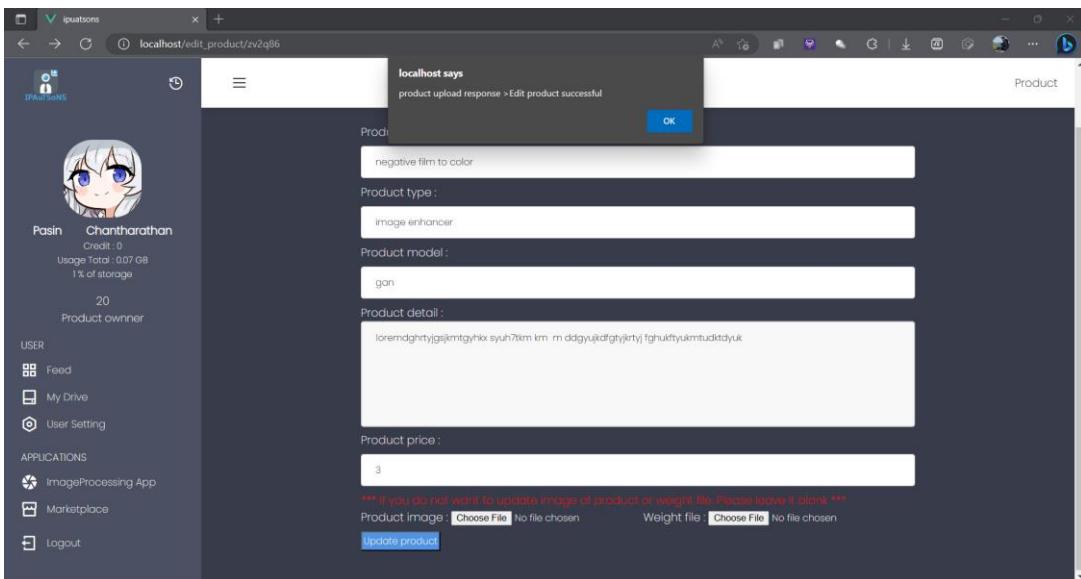
เพื่อทดสอบการใช้งานระบบการแก้ไขข้อมูลสินค้าในตลาดซื้อขายว่ามีการทำงานที่ถูกต้อง โดยจะสามารถส่งข้อมูลสินค้า และ ตัว Weight ของ model และรูปภาพของการใช้สินค้าที่ผู้ใช้ได้ทำการแก้ไขจากหน้า Frontend ไปยัง Backend ได้และทดสอบการทำงานของตัว Weight ได้ โดยที่หากการทำงานสำเร็จ ข้อมูลของสินค้าที่อยู่ในฐานข้อมูลจะมีการแก้ไข และ หากการทำงานไม่สำเร็จก็จะมีการแสดงข้อความแจ้งเตือนผู้ใช้

##### 4.6.2.2 วิธีการทดลอง

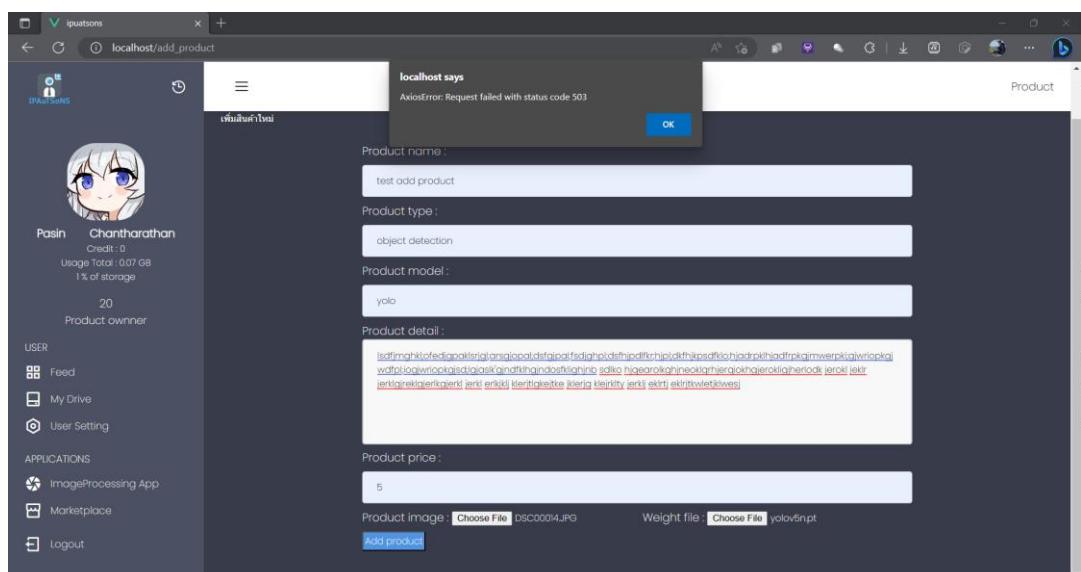
ทดสอบโดยการแก้ไขข้อมูลสินค้าด้วยข้อมูลที่แตกต่างกันหลายๆ ครั้ง โดยจะตรวจสอบข้อมูลที่ได้รับมาและถูกแก้ไขในฐานข้อมูลว่ามีความครบถ้วนและถูกต้องตามรูปแบบที่ได้มีการออกแบบเอาไว้และถ้าหากการทำงานเกิดข้อผิดพลาดขึ้น ก็จะมีการแสดงข้อความแจ้งเตือนผู้ใช้ว่าการแก้ไขข้อมูลสินค้านั้นไม่สำเร็จ

##### 4.6.2.3 ผลการทดลอง

จากการทดลองระบบการเพิ่มสินค้าในตลาดซื้อขายของ Web application สามารถแก้ไขข้อมูลสินค้าและสามารถตรวจสอบ Weight ของ Model ที่ Upload เข้ามาโดยหลังการทำงานเสร็จสิ้นจะแสดงข้อความว่าผู้ใช้นั้นแก้ไขข้อมูลสินค้าสำเร็จและแสดงหน้าต่างหน้า Marketplace ขึ้นมา และในการทำงานที่ไม่สำเร็จนั้นก็สามารถแสดงข้อความกลับมาให้ผู้ใช้ได้รับทราบ โดยการแสดงผลแต่ละส่วนสามารถดูได้จากรูปที่ 4.21 และ 4.22



รูป 4.21 ข้อความที่แสดงหากสามารถแก้ไขข้อมูลสินค้าได้สำเร็จ



รูป 4.22 ข้อความที่แสดงหากไม่สามารถแก้ไขสินค้าได้

#### 4.6.3 การทดลองระบบการส่งข้อมูลสินค้าเพื่อทดลองใช้ในส่วนหน้าสั่งงานประมวลผล

##### 4.6.3.1 ชุดประสิทธิภาพทดลอง

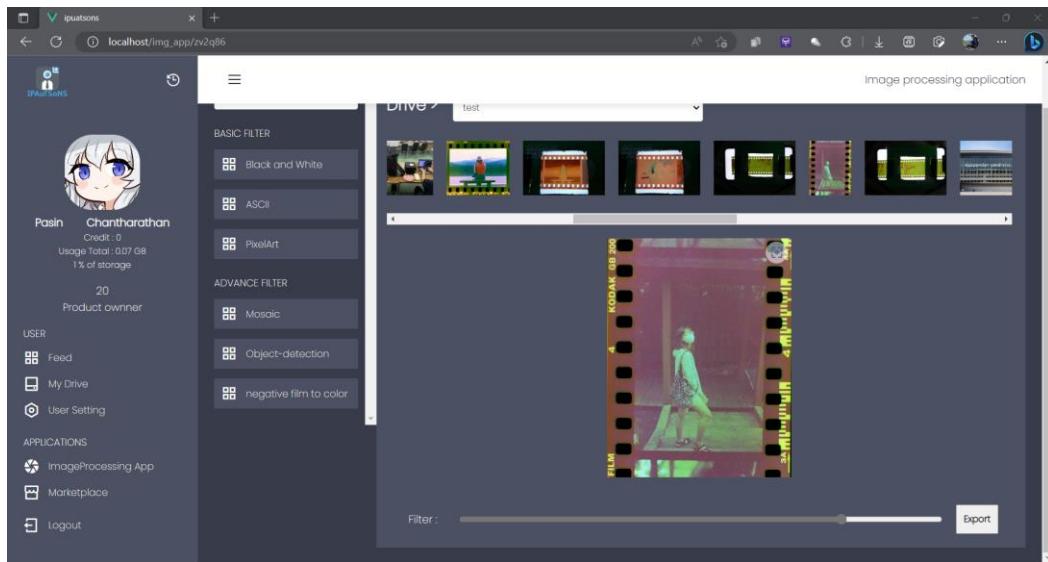
เพื่อทดสอบการใช้งานระบบการส่งข้อมูลสินค้าเพื่อทดลองใช้ในส่วนหน้าสั่งงานประมวลผลว่ามีการทำงานที่ถูกต้อง โดยจะสามารถส่งข้อมูลสินค้าจากหน้าส่วนต่อไปนี้สั่งงานประมวลผลได้ การทำงานที่ถูกต้อง โดยจะสามารถส่งข้อมูลสินค้าจากหน้าสั่งงานประมวลผลได้ โดยที่หากการทำงานสำเร็จ ข้อมูลของสินค้าที่อยู่ในฐานข้อมูลจะถูกส่งไปหน้าสั่งงานประมวลผลและสามารถทดลองใช้งานได้ และ หากการทำงานไม่สำเร็จก็จะมีการแสดงข้อความแจ้งเตือนผู้ใช้

#### 4.6.3.2 วิธีการทดลอง

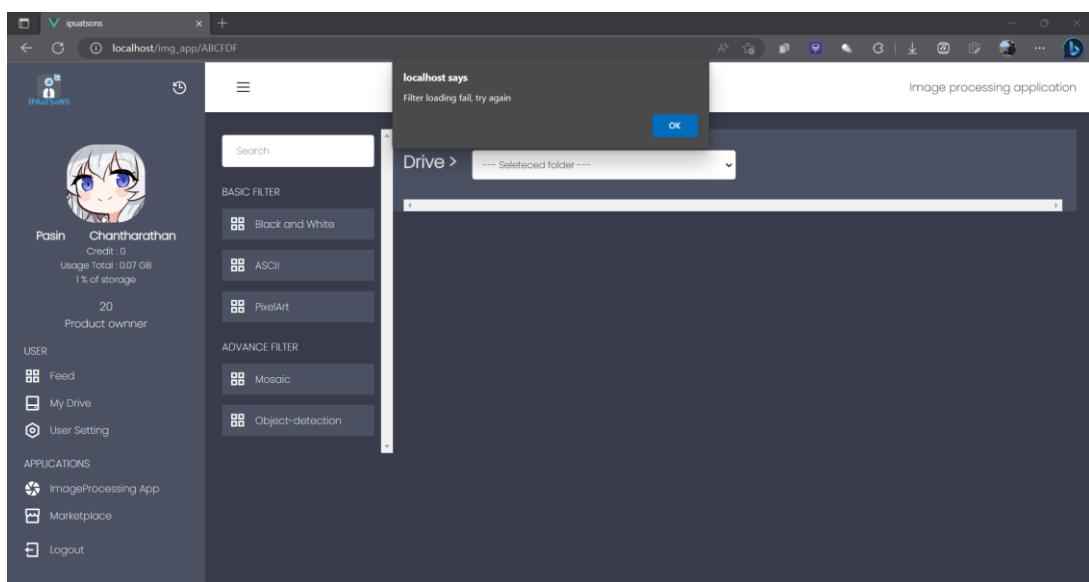
ทดสอบโดยการเพิ่มสินค้าจำนวนหนึ่งมาแล้วทำการเลือกทดลองใช้สินค้านั้นในหน้าสั่งงานประมวลผล โดยจะมีการส่งข้อมูลแบบที่ผิดพลาดไปด้วยเพื่อทดสอบการตรวจสอบการตรวจจับข้อผิดพลาด

#### 4.6.3.3 ผลการทดลอง

จากการการทดลองระบบการส่งข้อมูลสินค้าเพื่อทดลองใช้ในส่วนหน้าสั่งงานประมวลผล ของ Web application สามารถส่งข้อมูลสินค้าเพื่อไปแสดงผลในส่วนหน้าสั่งงานประมวลผลได้โดยมีข้อมูลครบถ้วนและถูกต้องตามที่ได้ทำการออกแบบไว้ และเมื่อส่งข้อมูลในรูปแบบที่ผิดพลาดก็สามารถแสดงข้อความแจ้งเตือนผู้ใช้ได้ โดยแต่ละส่วนสามารถดูได้จากรูปที่ 4.23 และ 4.24



รูป 4.23 ข้อมูลสินค้าที่สามารถส่งมาทดลองใช้งานที่หน้าสั่งงานประมวลผล จะแสดงในแบบทางค้านซ้าย



รูป 4.24 ข้อความที่แสดงหากไม่สามารถส่งข้อมูลสินค้ามาที่หน้าสั่งงานประมวลผลได้

## 4.7 ส่วนของหน้าประวัติการใช้งานของผู้ใช้

### 4.6.1 การทดลองระบบการเรียงลำดับและค้นหาการทำรายการด้วยวันที่

#### 4.6.1.1 จุดประสงค์การทดลอง

เพื่อทดสอบการแสดงผลและประเมินผลของการทำรายการต่าง ๆ ของผู้ใช้ เช่นการจ่ายเครดิตเพื่อสั่งงานประมวลผล หรือ การได้รับเครดิตจากการที่มีผู้ใช้นำสินค้าเราไปสั่งงานประมวลผล ซึ่งจะมีการเรียงลำดับรายการเพื่อให้เกิดความสะดวกต่อผู้ใช้ในดูประวัติการใช้งานของตนเอง

#### 4.6.1.2 วิธีการทดลอง

ทดลองโดยการเพิ่มรายการเป็นข้อมูลที่เตรียมเอาไว้โดยไม่เรียงลำดับการเพิ่มข้อมูลแต่ละวันที่ในข้อมูลนั้นเป็นเวลาที่ใกล้เคียงกันโดยห่างกันตั้งแต่ 1 นาที, 5 นาที จนไปถึง 2 วัน เมื่อมีการเรียกดูข้อมูลค่าเริ่มต้นจะเรียกดูข้อมูลที่มีวันที่ล่าสุดขึ้นมาก่อน และเรียงลำดับจนไปถึงต่ำสุด และเมื่อมีการเรียกดูข้อมูลเก่าสุด ก็จะแสดงผลเป็นข้อมูลที่มีวันที่เก่าที่สุด และเรียงลำดับไปจนถึงใหม่สุด และถ้าหากเลือกค้างไว้เป็นวันที่ ก็จะแสดงผลรายการที่เก่าที่สุดนับตั้งแต่วันที่นั้นและเรียงลำดับตามจำนวนรายการที่มีวันที่ล่าสุด

#### 4.6.1.3 ผลการทดลอง

จากการทดลองระบบการเรียงลำดับและค้นหาการทำรายการด้วยวันที่ของหน้าประวัติการใช้งานของผู้ใช้ ผลคือสามารถเรียงลำดับและแสดงผลรายการ ได้อย่างถูกต้องและครบถ้วน โดยสามารถดูผลการเรียงลำดับและค้นหา ได้รูปที่ 4.25 , 4.26 และ 4.27

Payment history				
sort by newest	sort by oldest	mm/dd/yyyy		
kouoqn6mdv8clddwdb66p2ayo	0	Pay	100	2023-04-17T16:23:08.556000+07:00
xhs4t27apji8tlqc9docxerim	0	Pay	100	2023-04-17T16:20:53.593000+07:00
43colnwwp8gpsvydlm02z2l53	0	Pay	100	2023-04-17T16:13:12.581000+07:00
aril6cojna6zy8rp58xdtkglm	0	Pay	100	2023-04-17T16:10:44.715000+07:00
pabqazvez5ximq4ir7a29tsjh	0	Pay	100	2023-04-17T16:08:17.946000+07:00
s0ojjhq4ui9ftfw32v5hkll32	0	Pay	100	2023-04-17T16:06:40.390000+07:00
ifsadupijauc2urtalxygkela	0	Pay	100	2023-04-17T16:02:14.042000+07:00
4iqlsncn7esgt87ikp7v73b2o	0	Pay	100	2023-04-17T15:59:59.403000+07:00
a5ptyqddiwxl3pxl65nn2vl4n	0	Pay	100	2023-04-17T15:55:33.766000+07:00

รูป 4.25 ประวัติรายการของผู้ใช้ที่มีการเรียงลำดับโดยการแสดงรายการล่าสุดก่อน

Payment history				
<a href="#">sort by newest</a> <a href="#">sort by oldest</a> <a href="#">mm/dd/yyyy</a> <a href="#">grid</a>				
payment_id	product_id	type	credit	pay_time
9llpovlyfebva7lsaojyvyocu		Receive	1000	2023-04-01T15:25:14.094000+07:00
s1rcbniz4zqmq8mu3l24hajvt		Receive	100000	2023-04-16T17:22:21.421000+07:00
a4lvq7n8rehplesb3bn3k59j9	0	Pay	100	2023-04-17T15:53:01.604000+07:00
ugxbt9n32hrycp62penyyjjo	0	Pay	100	2023-04-17T15:53:34.570000+07:00
a5ptyqddiwx3pxl65nn2vl4n	0	Pay	100	2023-04-17T15:55:33.766000+07:00
4iqlsncn7esgt87ikp7v73b2o	0	Pay	100	2023-04-17T15:59:59.403000+07:00
ifsadupijauc2urtalxygkele	0	Pay	100	2023-04-17T16:02:14.042000+07:00
s0ojjhq4ul9ftfw32v5hkl32	0	Pay	100	2023-04-17T16:06:40.390000+07:00
pabqazvez5ximq4ir7a29tsjh	0	Pay	100	2023-04-17T16:08:17.946000+07:00
aril6cojna6zy8rp58dtkglm	0	Pay	100	2023-04-17T16:10:44.715000+07:00

รูป 4.26 ประวัติรายการของผู้ใช้ที่มีการเรียกลำดับโดยการแสดงรายการเรียกที่สุดก่อน

Payment history				
<a href="#">sort by newest</a> <a href="#">sort by oldest</a> <a href="#">mm/dd/yyyy</a> <a href="#">grid</a>				
payment_id	product_id	type	credit	pay_time
s1rcbniz4zqmq8mu3l24hajvt		Receive	100000	2023-04-16T17:22:21.421000+07:00
a4lvq7n8rehplesb3bn3k59j9	0	Pay	100	2023-04-17T15:53:01.604000+07:00
ugxbt9n32hrycp62penyyjjo	0	Pay	100	2023-04-17T15:53:34.570000+07:00
a5ptyqddiwx3pxl65nn2vl4n	0	Pay	100	2023-04-17T15:55:33.766000+07:00
4iqlsncn7esgt87ikp7v73b2o	0	Pay	100	2023-04-17T15:59:59.403000+07:00
ifsadupijauc2urtalxygkele	0	Pay	100	2023-04-17T16:02:14.042000+07:00
s0ojjhq4ul9ftfw32v5hkl32	0	Pay	100	2023-04-17T16:06:40.390000+07:00
pabqazvez5ximq4ir7a29tsjh	0	Pay	100	2023-04-17T16:08:17.946000+07:00
aril6cojna6zy8rp58dtkglm	0	Pay	100	2023-04-17T16:10:44.715000+07:00

รูป 4.27 ประวัติรายการของผู้ใช้ที่มีคืนหาโดยไขว้วนที่ที่กำหนดเอาไว้

## 4.8 ส่วนของหน้าประวัติการใช้งานของสินค้า

### 4.6.1 การทดลองระบบการเรียกลำดับและคืนหาการทำรายการตัววันที่

#### 4.6.1.1 จุดประสงค์การทดลอง

เพื่อทดสอบการแสดงผลและประเมินผลของการทำรายการการถูกใช้งานของตัวสินค้า ซึ่งจะสามารถถูกได้เฉพาะเจ้าของของสินค้าคนนั้นๆ เท่านั้น ซึ่งจะมีการเรียกลำดับรายการเพื่อให้เกิดความสะดวกต่อผู้ใช้ในดุประวัติการใช้งานของสินค้าตนเอง

#### 4.6.1.2 วิธีการทดลอง

ทดลองโดยการเพิ่มรายการเป็นข้อมูลที่เตรียมเอาไว้โดยไม่เรียงลำดับการเพิ่มข้อมูลแต่วันที่ในข้อมูลนั้นเป็นเวลาที่ใกล้เคียงกัน โดยห่างกันตั้งแต่ 1 นาที, 5 นาที จนไปถึง 2 วัน เมื่อมีการเรียกคุ้นชื่อ ข้อมูลค่าเริ่มต้นจะเรียกคุ้นชื่อข้อมูลที่มีวันที่ล่าสุดขึ้นมาก่อน และเรียงลำดับจนไปถึงต่ำสุด และเมื่อมีการเรียกคุ้นชื่อ ข้อมูลเก่าสุด ก็จะแสดงผลเป็นข้อมูลที่มีวันที่เก่าที่สุด และเรียงลำดับไปจนถึงใหม่สุด และถ้าหากเลือกค้างเป็นวันที่ ก็จะแสดงผลรายการที่เก่าที่สุดนับตั้งแต่วันที่นั้นและเรียงลำดับมาจนถึงรายการที่มีวันที่ล่าสุด

#### 4.6.1.3 ผลการทดลอง

จากการทดลองระบบการเรียงลำดับและค้นหาการทำรายการด้วยวันที่ของหน้าประวัติการใช้งานของผู้ใช้ ผลคือสามารถเรียงลำดับและแสดงผลรายการ ได้อย่างถูกต้องและครบถ้วน โดยสามารถดูผลการเรียงลำดับและค้นหา ได้รูปที่ 4.28 , 4.29 และ 4.30

Product use history				
sort by newest	sort by oldest	mm/dd/yyyy		
payment_id	user_id	credit	type	pay_time
rj7pgtuphmoicwtnbup4cnuhb	p6mitrb9ar	5	1	2023-04-02T16:25:06.507000+07:00
sok07vrucfm3y4kfr8ggnd3f	p6mitrb9ar	5	1	2023-04-01T16:26:25.380000+07:00
vn2bz4m8ze9cd29mwbnbwf42w	p6mitrb9ar	5	1	2023-04-01T16:26:30.540000+07:00
a9028icdla5pb6njat6qwhzy	p6mitrb9ar	5	1	2023-04-01T16:26:39.846000+07:00
9lipovlyfebva7isojyyyocu	p6mitrb9ar	5	1	2023-04-01T16:26:34.094000+07:00

รูป 4.28 ประวัติรายการของสินค้าที่มีการเรียงลำดับโดยการแสดงรายการล่าสุดก่อน

Product use history				
sort by newest	sort by oldest	mm/dd/yyyy		
payment_id	user_id	credit	type	pay_time
9lipovlyfebva7isojyyyocu	p6mitrb9ar	5	1	2023-04-01T16:26:34.094000+07:00
a9028icdla5pb6njat6qwhzy	p6mitrb9ar	5	1	2023-04-01T16:26:39.846000+07:00
vn2bz4m8ze9cd29mwbnbwf42w	p6mitrb9ar	5	1	2023-04-01T16:26:30.540000+07:00
sok07vrucfm3y4kfr8ggnd3f	p6mitrb9ar	5	1	2023-04-01T16:26:25.380000+07:00
rj7pgtuphmoicwtnbup4cnuhb	p6mitrb9ar	5	1	2023-04-02T16:25:06.507000+07:00

รูป 4.29 ประวัติรายการของสินค้าที่มีการเรียงลำดับโดยการแสดงรายการเก่าที่สุดก่อน

Product use history				
sort by newest	sort by oldest	mm/dd/yyyy		
payment_id	user_id	credit	type	pay_time
rj7pgtuphmoicwlnbup4cnuhb	p5mitrb9or	5	1	2023-04-02T16:25:06.507000+07:00

รูป 4.30 ประวัติรายการของสินค้าที่มีคืนหาโดยใช้วันที่ที่กำหนดเอาไว้

## 4.9 ส่วนของหน้าร้านค้าแสดงสินค้า

### 4.9.1 การทดลองระบบการเรียงลำดับสินค้าด้วยวันที่ที่มีการแก้ไขข้อมูลสินค้า

#### 4.9.1.1 จุดประสงค์การทดลอง

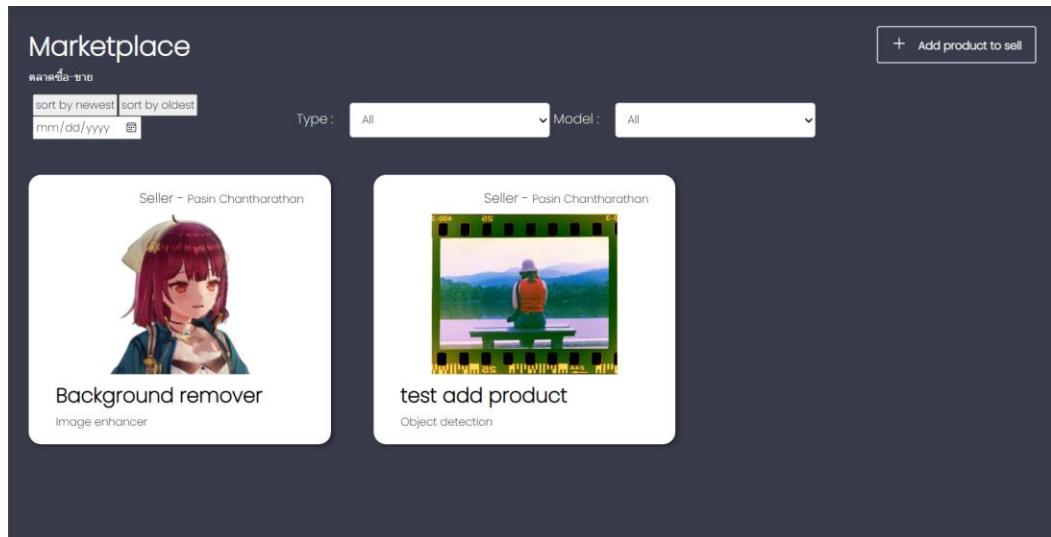
เพื่อทดสอบการแสดงผลและประมวลผลของสินค้าซึ่งจะมีการเรียงลำดับการแสดงสินค้าแต่ละรายการเพื่อให้เกิดความสะดวกต่อผู้ใช้ในการเลือกชมสินค้า

#### 4.9.1.2 วิธีการทดลอง

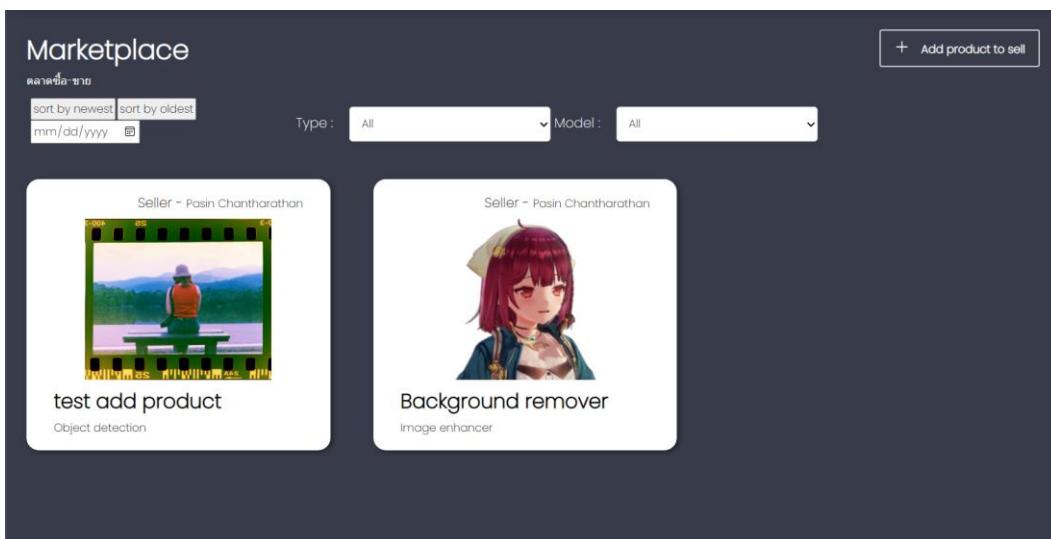
ทดลองโดยการเพิ่มสินค้าที่เป็นข้อมูลที่เตรียมเอาไว้โดยไม่เรียงลำดับการเพิ่มข้อมูลแต่วันที่ที่แก้ไขในข้อมูลนั้นเป็นเวลาที่ใกล้เคียงกันโดยห่างกันด้วย 1 นาที, 5 นาที จนไปถึง 2 วัน เมื่อมีการเรียกดูข้อมูลค่าเริ่มต้นจะเรียกดูสินค้าที่มีวันที่การแก้ไขล่าสุดขึ้นมาก่อน และเรียงลำดับตามไปถึงสินค้าวันที่การแก้ไขเก่าสุด และเมื่อมีการเรียกดูส่วนค้าที่มีวันที่แก้ไขเก่าสุด ก็จะแสดงผลเป็นข้อมูลที่มีวันที่ที่แก้ไขข้อมูลสินค้าเก่าที่สุด และเรียงลำดับไปจนถึงใหม่สุด

#### 4.9.1.3 ผลการทดลอง

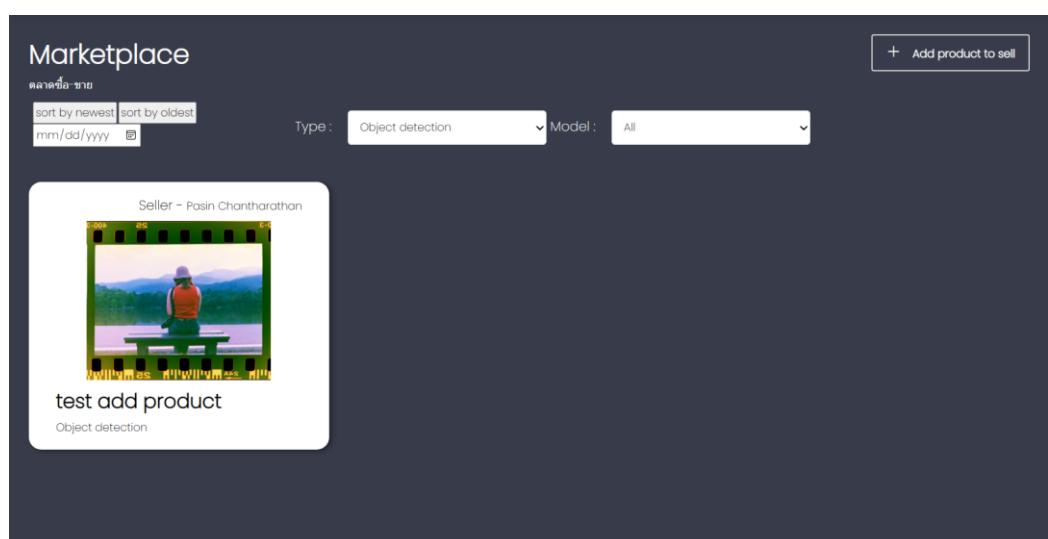
จากการทดลองระบบการเรียงลำดับและคืนหาการทำรายการด้วยวันที่ของหน้าประวัติการใช้งานของผู้ใช้ ผลคือสามารถเรียงลำดับและแสดงผลรายการได้อย่างถูกต้องและครบถ้วน โดยสามารถคุณภาพการเรียงลำดับและคืนหา ได้รูปที่ 4.31 , 4.32 และ 4.33



รูป 4.31 รายการของสินค้าที่มีการเรียงลำดับโดยการแสดงสินค้าที่มีการแก้ไขค่าสุด



รูป 4.32 รายการของสินค้าที่มีการเรียงลำดับโดยการแสดงสินค้าที่มีการแก้ไขมากที่สุด



รูป 4.33 รายการของสินค้าที่มีการกรองเลือกเฉพาะประเภท Object detections

## 4.10 ส่วนของหน้าแสดงข้อมูลเฉพาะสินค้า

### 4.10.1 การทดลองระบบแสดงผลข้อมูลสินค้า

#### 4.9.1.1 จุดประสงค์การทดลอง

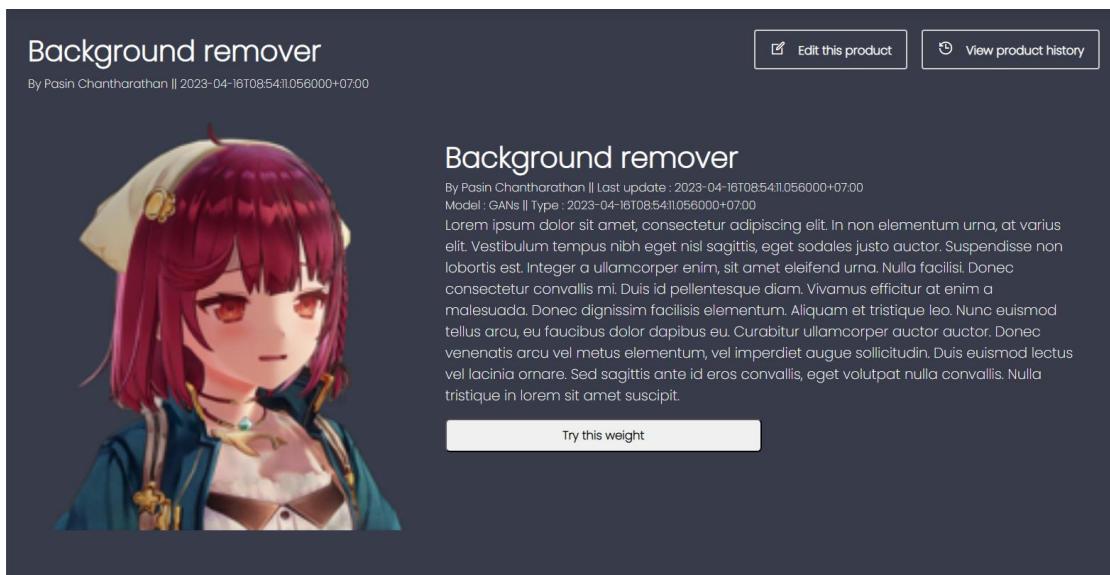
เพื่อทดสอบการแสดงผลข้อมูลของสินค้าซึ่งจะมีรายละเอียดของตัวสินค้าเพื่อให้เกิดความสะดวกต่อผู้ใช้ในการเลือกซื้อสินค้า

#### 4.10.1.2 วิธีการทดลอง

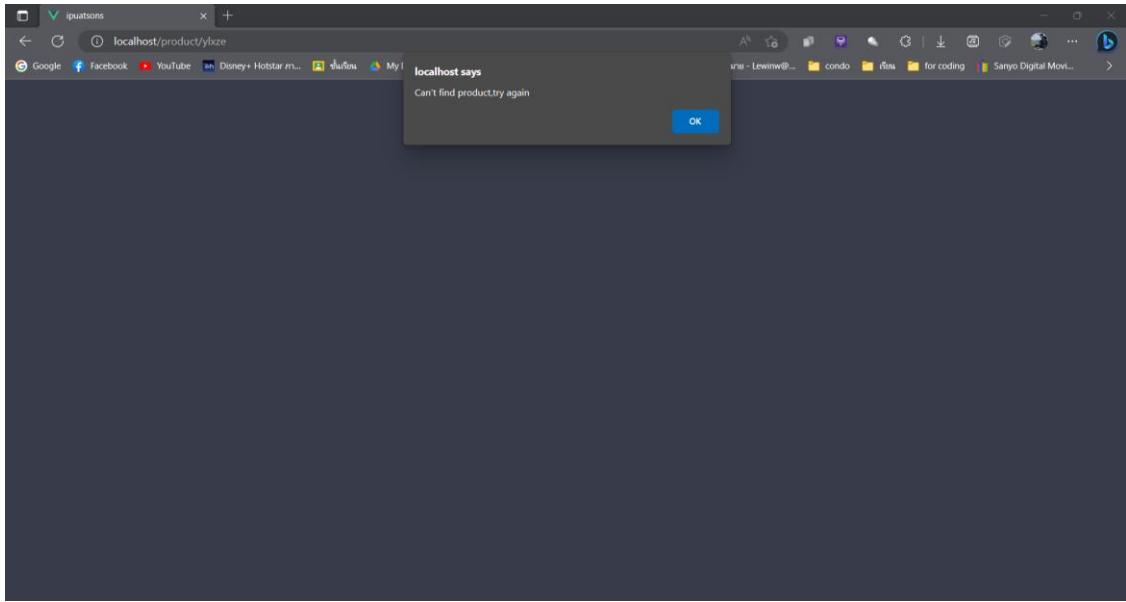
ทดลองโดยการเพิ่มสินค้าที่เป็นข้อมูลที่เตรียมเอาไว้ หลากหลายรูปแบบที่สามารถถูกเพิ่มเข้ามาใน Web applications ได้และ ทดลองทำการเข้าหาสินค้าที่ไม่มีอยู่จริงซึ่งถ้าหากหาสินค้าในฐานข้อมูลไม่พบ ก็จะแสดงข้อความแจ้งเตือนข้อผิดพลาด

#### 4.10.1.3 ผลการทดลอง

จากการทดลองระบบแสดงผลข้อมูลสินค้า ผลคือแสดงผลรายละเอียดสินค้าได้อย่างถูกต้องและครบถ้วน และ สามารถแจ้งเตือนผู้ใช้ได้หากไม่มีสินค้านั้นๆอยู่หรือเกิดข้อผิดพลาด ได้รูปที่ 4.34 , 4.35



รูป 4.34 แสดงรายละเอียดของสินค้าที่ถูกบันทึกเอาไว้ในฐานข้อมูล



รูป 4.35 แสดงข้อความแจ้งเตือนหากทำหาสินค้าที่ไม่มีอยู่ในฐานข้อมูล

## 4.11 ส่วนของการแบ่งภาระงาน

### 4.11.1 การทดลองระบบการแบ่งภาระงานของ Web Service

#### 4.11.1.1 จุดประสงค์การทดลอง

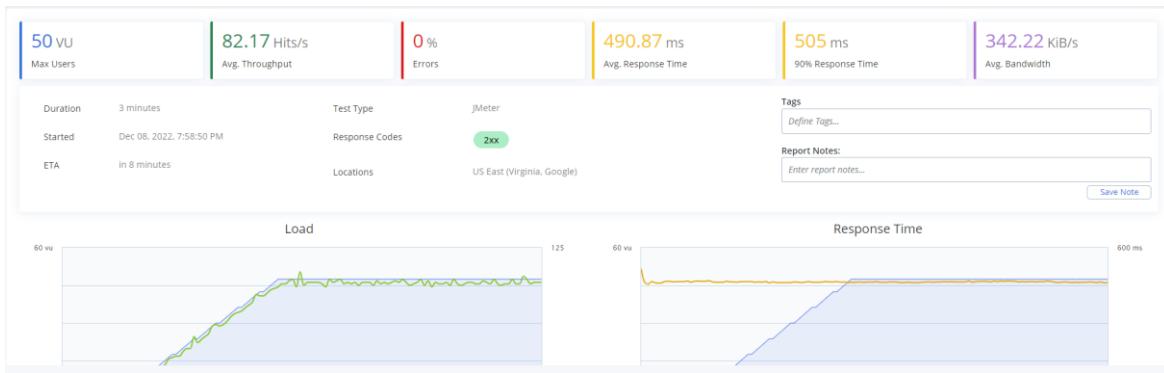
เพื่อทดสอบการแบ่งภาระงานของการเข้าใช้งาน Web Service โดยที่มีการกระจายผู้ใช้งานเข้าไปยังเครื่องอื่น ๆ บนระบบ Cluster เพื่อกระจายทรัพยากรในการเข้าใช้งานไปใหม่ประสิทธิภาพ

#### 4.11.1.2 วิธีการทดลอง

ทดสอบผ่าน Demo Web Application ที่พัฒนาขึ้นมาสำหรับการทดสอบโดยเฉพาะ โดยการนำเครื่องมือในการตั้ง Request เข้ามาจำนวน 50 คนพร้อม ๆ กันในเวลา 10 นาทีเพื่อตรวจสอบว่าสามารถเข้าใช้งานได้ สามารถแบ่งเป็นจัดสรรทรัพยากรได้คุ้มค่าเพียงพอ กับแต่ละผู้ใช้ในการใช้งานในส่วนของ Web Application ทั้ง Front-End และ Back-End ของระบบ Web Application ในการเข้ามาใช้งานในหน้าเว็บไซต์

#### 4.11.1.3 ผลการทดลอง

จากการทดลองกระจายภาระงานของ Web Service สามารถรองรับการเข้ามาใช้งานในตัวของเว็บไซต์พร้อมกันได้เฉลี่ย 8 คนต่อ 1 เครื่องใน Cluster โดยที่หลังจาก 8 คนไปประสิทธิภาพจะลดลงเรื่อยๆ และ มีการตอบสนองของ Server ต่อการเข้ามาใช้งานบริการในหน้าเว็บไซต์ที่ช้าลงจากรูป 4.36



รูป 4.36 ผลการทดสอบส่ง Request เข้ามาใช้งาน

#### 4.11.2 การทดสอบระบบการแบ่งภาระงานของ Application

##### 4.11.2.1 จุดประสงค์การทดสอบ

เพื่อทดสอบการแบ่งภาระงานของการสั่งงานประมวลผล Application โดยที่มีการกระจายงานไปยังเครื่องอื่น ๆ บนระบบ Cluster เพื่อกระจายทรัพยากรในการเข้าใช้งานไปให้มีประสิทธิภาพมากขึ้น ในการประมวลผล และเปรียบเทียบการแบ่งภาระงานระหว่างการประมวลผลด้วยเครื่องเดียวและการประมวลผลด้วยระบบคลัสเตอร์

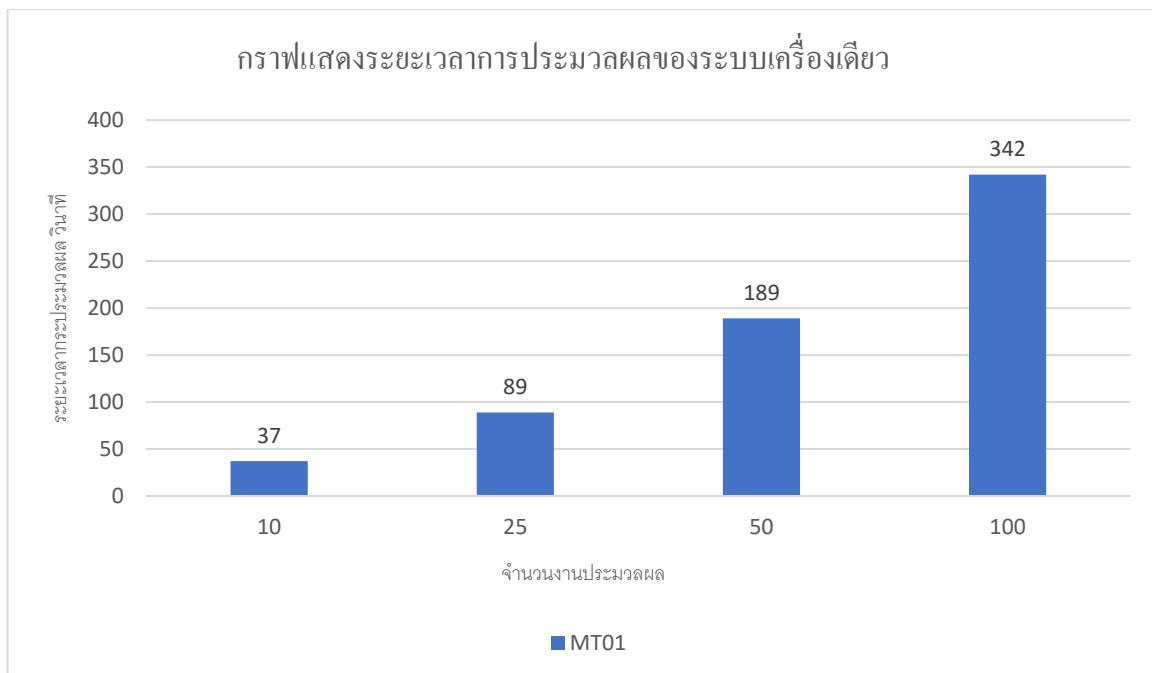
##### 4.11.2.2 วิธีการทดสอบ

ทดสอบผ่าน Demo Application ที่พัฒนาขึ้นมาสำหรับการทดสอบโดยเฉพาะ โดยให้ตัว Demo Application นั้นสั่งงานประมวลผลภาพโดยกำหนดให้ใช้รูปภาพชุดเดียวกันในการประมวลผลและประมวลผลด้วย Image processing application เดียวกัน เป็นจำนวนตั้งแต่ 10, 25, 50, และ 100 งาน ซึ่งจะทำการจับเวลาในการประมวลผลแต่ละจำนวนในแต่ละรูปแบบการประมวลผล โดยจะเปรียบเทียบระหว่างการประมวลผลด้วยเครื่องเดียว ที่ใช้เครื่องคอมพิวเตอร์ชื่อ MT01 โดยมีสเปคคือ CPU i9-9900K, RAM 32 GB, GPU RTX 2060 6 GB และการประมวลผลด้วยระบบคลัสเตอร์ ที่ใช้เครื่องคอมพิวเตอร์ชื่อ MT01 โดยมีสเปคคือ CPU i9-9900K, RAM 32 GB, GPU RTX 2060 6 GB เป็น Node หลัก และ เครื่องคอมพิวเตอร์ชื่อ WK01 โดยมีสเปคคือ CPU i5-8300H, RAM 16 GB, GPU GTX 1050 4 GB เป็น Node ย่อย

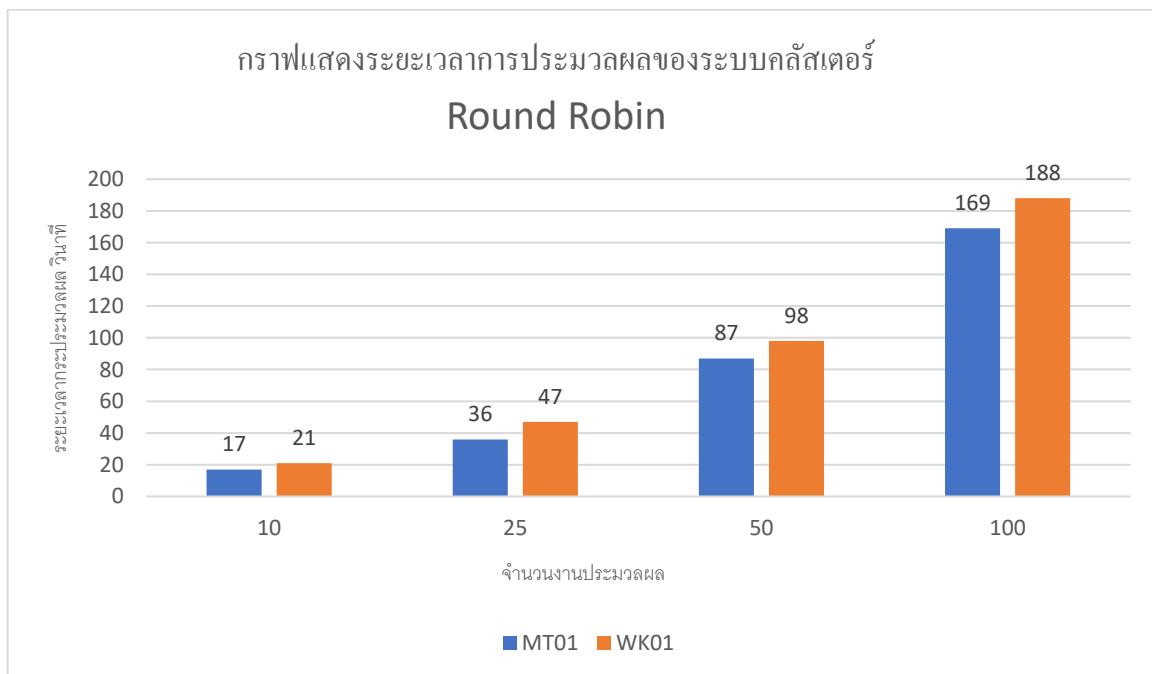
##### 4.11.2.3 ผลการทดสอบ

จากการทดสอบเมื่อมีการสั่งงานประมวลผลแบบเครื่องเดียวนั้นยังมีปริมาณงานมากขึ้น ระยะเวลาที่ใช้งานก็จะสูงขึ้นตาม ซึ่งถ้าเทียบกับการประมวลผลด้วยระบบคลัสเตอร์ ระยะเวลาที่ใช้งานการประมวลผลนั้นใช้ระยะเวลาที่ลดลงเหลือต่ำกว่า 40 เมื่อเปรียบเทียบกับการประมวลผลด้วยเครื่องเดียว แต่ทั้งนี้ระยะเวลาที่ได้มาต้องไม่แน่นอน เพราะเมื่อมีงานประมวลผลเข้ามายังมีการจัดการกระจายงานประมวลผลไปที่ Node แต่ละเครื่องโดยที่แต่ละเครื่องนั้นมีทรัพยากรไม่เท่ากัน และอาจ

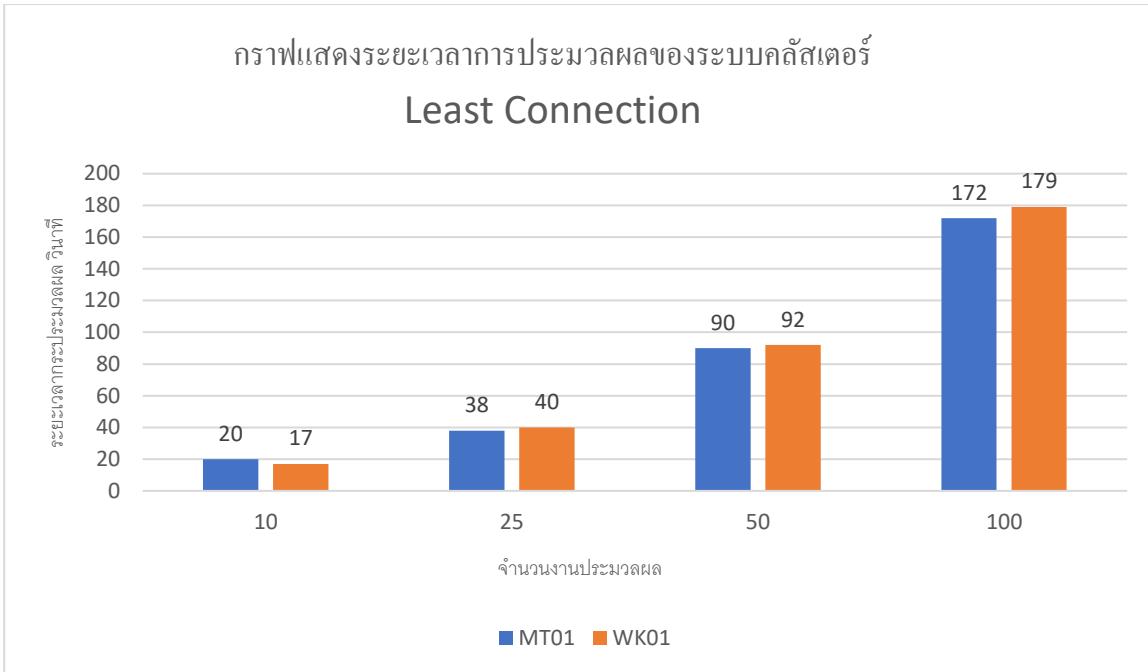
ใช้เวลาในการประมวลผลไม่เท่ากันแต่ยังคงความสามารถในการกระจายงานประมวลผล และ เนื่องจาก ยังมีระยะเวลาของการจัดการงาน และ ระยะเวลาในการตรวจสอบทรัพยากร จึงทำให้มีระยะเวลาที่ทำให้มี คุ้มค่ากับการประมวลผลงานในจำนวนที่น้อย แต่เมื่อมีงานจำนวนมากทำให้คุ้มค่ากับระยะเวลา ซึ่งผล การทดลองแบบ Round Robin และ Least Connection สามารถได้จากรูปภาพที่ 4.37, 4.38 และ 4.39



รูป 4.37 กราฟของผลการทดลองเมื่อสั่งงานประมวลผลด้วยการประมวลผลด้วยเครื่องเดียว



รูป 4.38 กราฟของผลการทดลองเมื่อสั่งงานประมวลผลด้วยการประมวลผลด้วยระบบคลัสเตอร์ด้วย Round Robin



รูป 4.39 กราฟของผลการทดลองเมื่อสั่งงานประมวลผลด้วยการประมวลผลด้วยระบบคลัสเตอร์ด้วย Least Connection

#### 4.11.2.4 เป้าหมายการทดลอง

จากการทดลองเพื่อทดสอบว่าระบบในการประมวลผลแบบ Cluter นั้นมีการประมวลผลที่ใช้ระยะเวลาโดยรวมน้อยกว่ารูปแบบ Single ในกรณีที่งานปริมาณเท่า ๆ กันเพื่อเป็นข้อสรุปของเป้าหมาย ตัวโครงสร้างนี้ในการช่วยลดระยะเวลาในงานประมวลผล ซึ่งจากการเปรียบเทียบการระยะเวลาการประมวลผลของระบบคลัสเตอร์แบบ Round Robin และ แบบ Least Connection ทำให้ทราบได้ว่าการประมวลผลของระบบคลัสเตอร์แบบ Least Connection นั้นมีระยะเวลาที่ใช้ในการประมวลผลที่ได้ดีกว่า ซึ่งสามารถวิเคราะห์ได้ว่าในกรณีเครื่องคอมพิวเตอร์ที่อยู่ในระบบคลัสเตอร์นั้นมีทรัพยากรที่แตกต่างกัน ใช้การประมวลผลของระบบคลัสเตอร์แบบ Least Connection นั้นน่าจะทำได้ดีกว่า เพราะด้วยการที่ทรัพยากรในการประมวลผลของแต่ละเครื่องนั้นมีไม่เท่ากันจึงทำให้การประมวลผลแต่ละงานในแต่ละเครื่องนั้นสำเร็จช้า-เร็วต่างกันออกไป ซึ่งการที่การประมวลผลของระบบคลัสเตอร์แบบ Least Connection จะพิจารณาการแจกจ่ายงานประมวลผลด้วยจำนวนของ Connection และจำนวน Queue ใน การเข้มต่อเข้าใช้งาน ซึ่งทำให้งานประมวลผลล่วงหน้าญี่ปุ่นโดยแยกจ่ายไปที่เครื่องที่มีทรัพยากรสูง ซึ่งสามารถประมวลผลได้ไวและมีจำนวนของ Connection และจำนวน Queue ในการเข้มต่อเข้าใช้งาน น้อยกว่า ทำการประมวลผลงานนั้นๆ จัดแจงการใช้ทรัพยากรได้ดีกว่า เทียบกับการประมวลผลของระบบคลัสเตอร์แบบ Round Robin ที่พิจารณาการแจกจ่ายงานประมวลผลด้วยลำดับของเครื่องโดยไม่สนใจจำนวนของ Connection และจำนวน Queue ในการเข้มต่อเข้าใช้งาน ซึ่งทำให้งานประมวลผลนั้นๆ แบ่งไปยังเครื่องที่มีทรัพยากรสูงและต่ำในปริมาณที่เท่ากัน จึงทำให้ระยะเวลาที่ใช้ในการประมวลผลนั้น เพิ่มขึ้นมากจากจำนวนงานประมวลผลที่ถูกแบ่งไปยังเครื่องที่มีทรัพยากรต่ำและประมวลผลได้ช้า

## 4.12 ส่วนของ Application Image Processing

### 4.12.1 การทดลองระบบการทำงานของ Application Image Processing

#### 4.12.1.1 จุดประสงค์การทดลอง

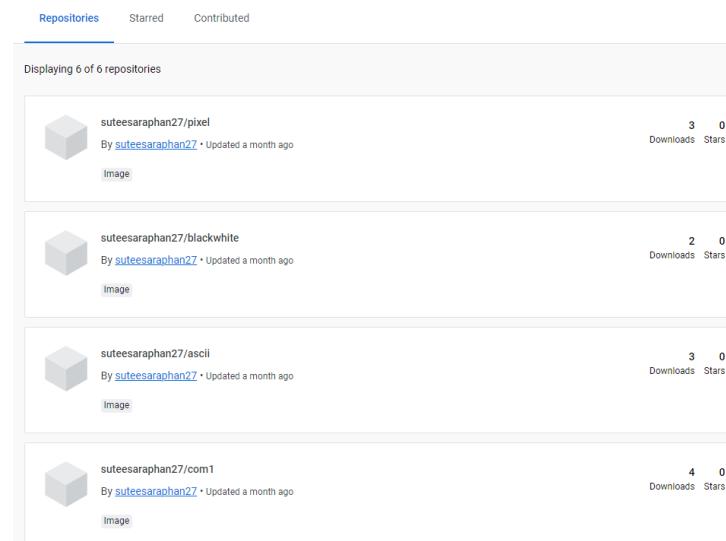
เพื่อทดสอบว่าระบบในการสั่งงานประมวลผลบนรูปแบบ Container ที่อยู่บน DockerHub ที่เราได้ทำการ Build และ Push ขึ้นไว้สำหรับเรียกใช้งานเพื่อนำมาประมวลผลงานตามคำสั่งเรียกใช้งานที่เราได้เขียนไว้บน Kubernetes Cluster และสามารถประมวลผลได้อย่างถูกต้อง

#### 4.12.1.2 วิธีการทดลอง

ทดสอบการสั่งงานด้วย Kubernetes ในการจัดเก็บ Application ไว้บน DockerHub และทดสอบด้วยรูปภาพจำนวนหนึ่งที่เก็บไว้ภายในเครื่อง Cluster สามารถถูกประมวลผลได้อย่างถูกต้อง และครบถ้วน โดยทำการเปรียบเทียบรูปภาพที่ผ่านการประมวลผลโดย Application ที่จัดเก็บไว้บน DockerHub และ Application ที่ทำการพัฒนาบนเครื่อง Local ซึ่งรูปผลลัพธ์ที่ได้รับจากทั้ง 2 แห่งต้องมีความเหมือนกัน

#### 4.12.1.3 ผลการทดลอง

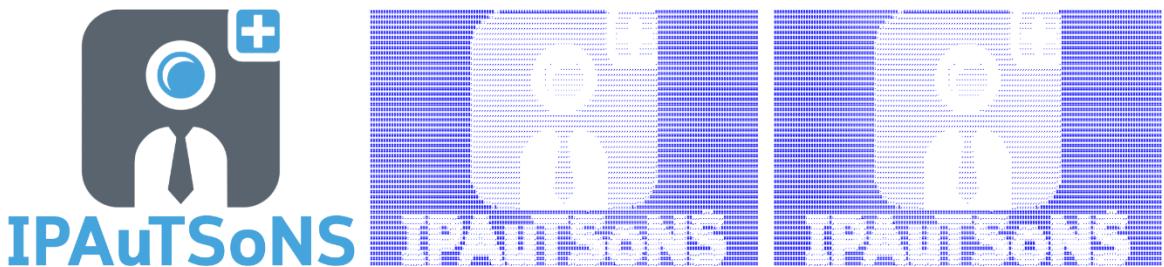
จากการทดลองสามารถประมวลผลภาพได้ถูกต้องบนระบบ Cluster ด้วย Kubernetes ซึ่งผลลัพธ์รูปภาพที่ได้ผ่านการประมวลผลด้วย Application ที่เก็บไว้บน DockerHub นั้น เมื่อกับรูปภาพที่ได้รับจากการประมวลผลด้วย Application ที่พัฒนาบนเครื่อง Local โดยไม่มีการทำงานผิดพลาดใดๆ แต่มีเวลาในการโหลดครึ่งแรกของการสั่งงานที่ค่อนข้างนานเนื่องจากต้องไปทำการโหลด Image ของ Application มาจาก DockerHub ก่อนที่จะเริ่มทำงานประมวลผลได้ โดยรูปภาพของ Repositories ของ DockerHub ที่ทำการเก็บ Application จะอยู่ในรูปภาพที่ 4.40



รูป 4.40 Repositories ของ DockerHub ที่ทำการเก็บ Application ไว้

#### 4.12.1.3.1 ผลการทดลอง Application ASCII

จากการทดลองด้วย Application ASCII โดยการทำงานอยู่บน Docker Image ด้วยการประมวลผลภาพเป็นรูปแบบ ASCII สามารถทำงานได้ปกติกับการทำงานอยู่บนระบบ Local โดยจะมีการประมวลผลสำเร็จแล้วดังรูปภาพที่ 4.41



รูป 4.41 (ซ้าย)รูปภาพต้นฉบับ (กลาง)รูปภาพที่ผ่านการประมวลผลด้วย Application ASCII บน Local (ขวา)รูปภาพที่ผ่านการประมวลผลด้วย Application ASCII ไว้บน DockerHub

#### 4.12.1.3.2 ผลการทดลอง Application BlackWhite

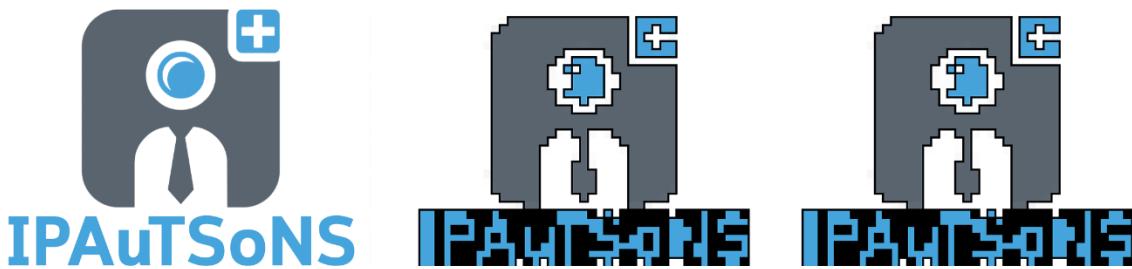
จากการทดลองด้วย Application BlackWhite โดยการทำงานอยู่บน Docker Image ด้วยการประมวลผลภาพเป็นรูปแบบ BlackWhite สามารถทำงานได้ปกติกับการทำงานอยู่บนระบบ Local โดยจะมีการประมวลผลสำเร็จแล้วดังรูปภาพที่ 4.42



รูป 4.42 (ซ้าย)รูปภาพต้นฉบับ (กลาง)รูปภาพที่ผ่านการประมวลผลด้วย Application BlackWhite บน Local (ขวา)รูปภาพที่ผ่านการประมวลผลด้วย Application BlackWhite ไว้บน DockerHub

#### 4.12.1.3.3 ผลการทดลอง Application PixelArt

จากการทดลองด้วย Application PixelArt โดยการทำงานอยู่บน Docker Image ด้วยการประมวลผลภาพเป็นรูปแบบ PixelArt สามารถทำงานได้ปกติกับการทำงานอยู่บนระบบ Local โดยจะมีการประมวลผลสำเร็จแล้วดังรูปภาพที่ 4.43



รูป 4.43 (ซ้าย)รูปภาพต้นฉบับ (กลาง)รูปภาพที่ผ่านการประมวลผลด้วย Application PixelArt บน Local (ขวา)รูปภาพที่ผ่านการประมวลผลด้วย Application PixelArt ไว้บน DockerHub

#### 4.12.1.3.4 ผลการทดลอง Application Mosaic

จากการทดลองด้วย Application Mosaic โดยการทำงานอยู่บน Docker Image ด้วยการประมวลผลภาพเป็นรูปแบบ Mosaic สามารถทำงานได้ปกติกับการทำงานอยู่บนระบบ Local โดยจะมีการประมวลผลสำเร็จแล้วดังรูปภาพที่ 4.44



รูป 4.44 (ซ้าย)รูปภาพต้นฉบับ (กลาง)รูปภาพที่ผ่านการประมวลผลด้วย Application Mosaic บน Local (ขวา)รูปภาพที่ผ่านการประมวลผลด้วย Application Mosaic ไว้บน DockerHub

## บทที่ 5

# สรุปผลและข้อเสนอแนะ

### 5.1 สรุปผลที่จากโครงการ

โครงการร้านค้าสำหรับแอพพลิเคชั่นการประมวลผลภาพซึ่งจัดการงานบนระบบประมวลผลแบบกลุ่ม มีส่วนการทำงานหลักๆอยู่ 3 ส่วนคือ ส่วนเว็บแอพพลิเคชั่น (Web Service) ส่วนของแอพพลิเคชั่น (Application Image Processing) และ ส่วนของเซิฟเวอร์คลัสเตอร์คอมพิวเตอร์ (Server Cluster) มีฟังก์ชันที่พัฒนาแล้วเสร็จตามแผนการพัฒนา ซึ่งได้มีการพัฒนาและ Deploy ไปรูปแบบ Production ที่สามารถใช้ทดลองใช้งานได้แล้วดังนี้

#### 5.1.1 ส่วนของเว็บแอพพลิเคชั่น (Web Service)

##### 5.1.1.1 ส่วนของหน้าลงทะเบียนสมัครเข้าใช้งาน

เป็นส่วนต่อประสานกับผู้ใช้ระดับ Client สำหรับทำการลงทะเบียนสมัครเข้าใช้งาน Web application โดยต้องกรอก Email, รหัสผ่าน, ชื่อ และ นามสกุล เพื่อเป็นข้อมูลในการสมัครใช้งาน

##### 5.1.1.2 ส่วนของหน้าลงทะเบียนเข้าสู่ระบบใช้งาน

เป็นส่วนต่อประสานกับผู้ใช้ระดับ Client สำหรับทำการลงทะเบียนเข้าสู่ระบบเพื่อใช้งาน Web application โดยต้องกรอก Email, รหัสผ่าน เพื่อเป็นข้อมูลในการเข้าสู่ระบบเพื่อใช้งาน

##### 5.1.1.3 ส่วนของหน้าแก้ไขข้อมูลผู้ใช้งาน

เป็นส่วนต่อประสานกับผู้ใช้สำหรับการแก้ไขข้อมูลของผู้ใช้อง รวมไปถึงการแก้ไขรหัสผ่านด้วย

##### 5.1.1.4 ส่วนของหน้าจัดการไฟล์

เป็นส่วนต่อประสานกับผู้ใช้สำหรับทำการจัดการ Folder สำหรับเก็บไฟล์รูป และ ตัวไฟล์รูปด้วย เพื่อให้สามารถนำรูปภาพไปใช้งานต่อในการสั่งงานประมวลผลภาพ

##### 5.1.1.5 ส่วนของแอพพลิเคชั่นประมวลผลภาพ

เป็นส่วนต่อประสานกับผู้ใช้สำหรับทำการสั่งงานประมวลผลภาพ โดยจะต้องกรอกข้อมูลรายละเอียดของงานเข้าไปเพื่อทำการสร้างไฟล์ YAML ไปสั่งงานประมวลผลภาพ

##### 5.1.1.6 ส่วนของหน้าข่าวสาร

เป็นส่วนต่อประสานกับผู้ใช้สำหรับแสดงผลสินค้าที่มีการเปลี่ยนแปลงข้อมูลล่าสุดจำนวนหนึ่งเพื่อให้ผู้ใช้ได้ทราบถึงข่าวสารล่าสุดของสินค้า

### 5.1.1.7 ส่วนของหน้าจอบันทึกการใช้งานของผู้ใช้

เป็นส่วนต่อประสานกับผู้ใช้สำหรับแสดงบันทึกรายการการใช้หรือได้รับเครดิตของผู้ใช้ว่าถูกใช้ไปกับสินค้าใดหรือบริการใด

### 5.1.1.8 ส่วนของหน้าจอบันทึกการขายของสินค้า

เป็นส่วนต่อประสานกับผู้ใช้สำหรับแสดงบันทึกรายการการนำสินค้าไปใช้งานสั่งงานประมวลผลเมื่อใดและจำนวนเท่าไหร่ เป็นเครดิตให้กับเจ้าของสินค้าเท่าไหร่

### 5.1.1.9 ส่วนของหน้าร้านค้าแสดงสินค้า

เป็นส่วนต่อประสานกับผู้ใช้สำหรับจัดแสดงสินค้าให้ผู้ใช้ได้เข้ามาเลือกซื้อและรับชมรวมไปถึงจัดแสดงสินค้าของตนเองได้

### 5.1.1.10 ส่วนของหน้าแสดงข้อมูลเฉพาะสินค้า

เป็นส่วนต่อประสานกับผู้ใช้สำหรับทำแสดงข้อมูลเฉพาะสินค้านั้นๆ โดยในแสดงข้อมูลจะมีรูปภาพตัวอย่างของสินค้า รายละเอียดการใช้งาน ความสามารถ ประเภทของสินค้า และ ราคา ระบุอยู่ภายใน

## 5.1.2 ส่วนแอพพลิเคชั่น (Application Image Processing)

### 5.1.2.1 แอพพลิเคชั่นประมวลผลภาพเป็นขาวดำ (Black White Application)

การแปลงภาพสี RGB ที่มีอยู่ 3 Channel สีให้ออกมาเป็น 1 Channel สีจึงเกิดเป็นภาพสีขาว ดำ ขึ้นมาโดยใช้ตัวของ Python PIL ในการทำงาน

<https://hub.docker.com/repository/docker/suteesaraphan27/blackwhite>

### 5.1.2.2 แอพพลิเคชั่นประมวลผลภาพเป็นอักษรและสกุล (ASCII Application)

ในส่วนของ ASCII Image คือ การแปลงรูปภาพให้เป็น Pixel อยู่ในรูปของ Array และ นำตัวอักษร ASCII เช่น “,:irs?@9B#” มารวมแทนที่แต่ละ Pixel นั้นตามค่าตัวอักษรที่เราได้เรียนมาเป็นในรูปแบบของ Array List โดยสีของรูปภาพ ASCII นั้นจะขึ้นอยู่กับความเข้มของสีในภาพเดิมก่อนนำมาทำเป็น ASCII Image ยิ่งภาพมีสีเข้มตัวของ ASCII นั้นจะมีสีเข้มตามໄลร์ดับจาก สีขาว สีเทา สีน้ำเงิน และ สีดำ

<https://hub.docker.com/repository/docker/suteesaraphan27/ascii>

### 5.1.2.3 แอพพลิเคชั่นประมวลผลภาพเป็นโมเสก (Mosaic Application)

การนำรูปภาพ 1 ภาพมาตัดแบ่งเป็น Tiles โดยกำหนดขนาดของ Tiles ออกมาก็ได้ และ ยิ่งขนาดเล็กเท่าไหร่ ความละเอียดของ Tiles ก็จะยิ่งสูงขึ้น โดยเมื่อแบ่ง Tiles เสร็จเราวางรูปภาพอื่น ๆ จำนวนมากที่เราเตรียมไว้มาทำการ Feature Map กับตัว Tiles เพื่อทำให้ Tiles นั้น ๆ เกิดออกมารูปภาพอื่น ๆ ที่

เราได้เตรียมไว้และออกแบบให้มีองค์ประกอบแบบ Mosaic

<https://hub.docker.com/repository/docker/suteesaraphan27/mosaic>

#### 5.1.2.4 แอพพลิเคชันประมวลผลภาพเป็นพิกเซลศิลปะ (PixelArt Application)

การแปลงภาพปกติให้ออกมาเป็น Pixel ให้สวยงาม โดยใช้การกำหนดค่า Kernel\_size, Pixel\_size และ Edge\_thresh โดยเมื่อกำหนดค่าทั้ง 3 แล้วจึงนำภาพมาประมวลผลให้อยู่ในรูปแบบ tensor ก่อนแล้วจึงนำมาคำนวณด้วยการเติมค่าทั้ง 3 ลงไปใน tensor ที่ได้แปลงจากภาพมาแล้วจึงนำมาแปลงค่า tensor กลับมาเป็นภาพ

<https://hub.docker.com/repository/docker/suteesaraphan27/pixel>

### 5.1.3 ส่วนของเซิฟเวอร์คลัสเตอร์คอมพิวเตอร์ (Server Cluster)

#### 5.1.3.1 ในส่วนของ K3S

ได้จัดการติดตั้ง K3S เพื่อทำการจำลองระบบ Cluster ขึ้นมาในรูปแบบของ Container เพื่อทดลองใช้งาน และ พัฒนาตัวโครงงานนี้ขึ้น

#### 5.1.3.2 ในส่วนของ K3D

ได้ทำการแก้ไขตัวของ K3D เพื่อให้รองรับการทำงานของ GPU และ ได้ปรับปรุงใหม่ให้ตัว Kubernetes มีการรองรับการทำงานของ CUDA ขึ้นมาด้วย

#### 5.1.3.3 ในส่วนของ Monitor

ได้ทำการติดตั้ง LENS Kubernetes สำหรับการ Monitoring ตัว Cluster เพื่อตรวจสอบการทำงาน และ การแจ้งเตือนต่าง ๆ ได้ง่ายยิ่งขึ้น และ ได้ติดตั้ง Prometheus Grafana เพื่อทำการตรวจเช็คการทำงานของ CPU RAM GPU ในรูปแบบ Metric เพื่อเก็บสถิติ และ นำไปใช้งานต่อไปได้

#### 5.1.3.4 ในส่วนของ Load Balancer

ได้ทำการตั้งค่า และ ติดตั้ง Nginx ด้วย Ingress สำหรับการ Exposing Services ออกไปสู่ภายนอก และ มีการ Load Balancing ในการเข้าใช้งาน Web Service ในส่วนของการ Load Balancer งานประมวลผลได้ใช้อัลกอริทึมเป็น Round Robin

#### 5.1.3.5 ในส่วนของฐานข้อมูล (Database)

ได้ทำการติดตั้ง Database เป็น MongoDB และ ได้จัดการตั้งค่าต่าง ๆ ของ Server ให้ใช้งานจัดเก็บข้อมูล ไว้นอก Cluster

### 5.1.3.6 ในส่วนของ Persistent Volume

ได้จัดการในส่วนของ Persistent Volume ขึ้นมาเพื่อให้ Cluster System มอง Volume จากภายนอก และ เข้าใช้งานได้ทุกเครื่องบน Cluster เพื่อกันการที่ข้อมูลเสียหายภายใน Cluster จึงได้ทำการเก็บข้อมูลไว้ในส่วนนอกของ Cluster System

## 5.2 ปัญหาและอุปสรรค

- 1) การทำระบบ Cluster นั้นจะต้องมีเครื่องคอมพิวเตอร์อย่างน้อย 2 เครื่องขึ้นไปจึงมีปัญหาในการหาเครื่องคอมพิวเตอร์มาใช้ทำโครงงานนี้
- 2) เนื่องจากการทดลองทดสอบระบบส่วนใหญ่ยังใช้การจำลองเพียงเครื่องเดียวจึงทำให้ได้ผลสรุปมาค่อนข้างไม่ตรงกับความต้องการที่จะทดลอง
- 3) การพัฒนา Web application ที่ต้องใช้ Library เสริมเข้ามาช่วยในการพัฒนานี้ เกิดปัญหาการ Conflict กันเองระหว่าง Library ที่เมื่อมี Library ใหม่ถูกติดตั้งเพิ่มเข้ามาก็อาจจะทำให้เกิดปัญหากับ Library เดิมที่มีอยู่
- 4) ระบบ Network ที่ใช้ในการพัฒนานี้เป็น Network ของทางสถาบันที่มีการตั้งข้อจำกัดด้านความปลอดภัยเอาไว้ ทำให้ไม่สามารถใช้งานแบบ Public ได้ จึงจำเป็นต้องใช้การจำลอง Private Network ในการทดสอบแทนการใช้ Public Network

## 5.3 แนวทางการพัฒนาต่อไปในอนาคต

- 1) พัฒนาส่วนของการแบ่งภาระงานประมวลผลให้มีประสิทธิภาพมากยิ่งขึ้น
- 2) พัฒนา Web Application ให้ใช้งานได้ปลอดภัยในการส่งข้อมูลไปยัง Database และ Cluster โดยที่ยังคงหรือเพิ่มความรวดเร็ว และ ลดจำนวนครั้งที่ส่ง-รับข้อมูลลงไป
- 3) พัฒนาแก้ไขในส่วนของ User interface ให้มีความสวยงาม และ เหมาะสมกับการใช้งานมากขึ้น
- 4) พัฒนาตัวจัดการ Weight Model ในการใช้งานบน Application
- 5) พัฒนาระบบเติมเงินซื้อเครดิตเข้ามาใช้จริง
- 6) แก้ไข-ปรับปรุงในส่วนของระบบเครดิต เพื่อให้การใช้งานสั่งงานประมวลผลนั้นคุ้มค่ากับ Client และผู้ให้บริการ

## បរចនានៃក្រម

Salem Saleh Al-amri. 2011. **Contrast Stretching Enhancement in Remote Sensing Image.**

Department of Computer Science

North Maharashtra University, Jalgaon, India.

Najia Naz, Abdul Haseeb Malik, Abu Bakar Khurshid, Furqan Aziz, Bader Alouffī, M. Irfan Uddin,

Ahmed AlGhamdi. 2020. **Efficient Processing of Image Processing Applications on CPU/GPU.** [Online].

Available : <https://www.hindawi.com/journals/mpe/2020/4839876> .

Yuqi Fu, Shaolun Zhang, Jose Terrero, Ying Mao, Guangya Liu, Sheng Li, Dingwen Tao. 2019.

**Progress-based Container Scheduling for Short-lived Applications in a Kubernetes Cluster.** [Online].

Available : <https://ieeexplore.ieee.org/abstract/document/9006427> .

Jonas Wagner. 2015. **Javascript Film Emulation.** [Online].

Available : <https://29a.ch/2015/06/07/javascript-film-emulation> .

Wongsathorn Insawasd. 2020. **Vue.js គឺខ្លះនៅវិញ មិនខ្លះនៅវិញ តាមតើអ្វីដែលត្រូវបានបង្កើតឡើងទេ.** [Online].

Available : <https://insawasd.com/what-is-vuejs/> .

ឯម៌ព្រាកភូមិថ្មីរ៉ាប់រ៉ាប់. 2021. **Docker (software).** [Online].

Available : [https://en.wikipedia.org/wiki/Docker\\_\(software\)](https://en.wikipedia.org/wiki/Docker_(software)) .

## ภาคผนวก ก

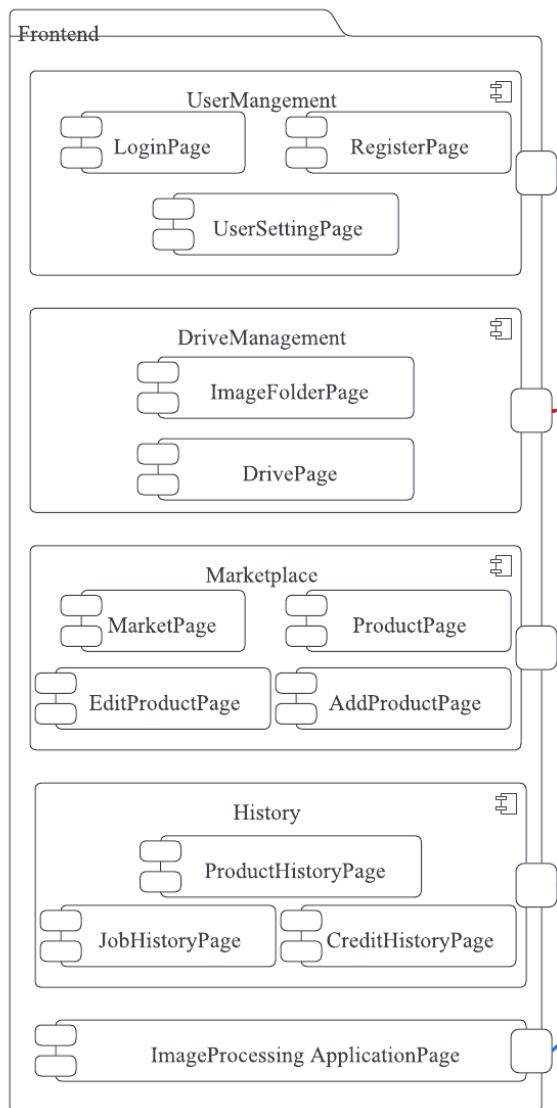
### รายละเอียดของ diagram

#### ก.1 Component diagram(Detail)

แผนภาพองค์ประกอบหรือ Component diagram จะถูกแบ่งออกเป็น 4 ส่วนด้วยกันคือ

##### ก.1.1 ส่วน Frontend

รูปที่ ก.1 จะแสดงแผนภาพองค์ประกอบส่วน Frontend ซึ่งมีหน้าที่หลักในการต่อประสานกับผู้ใช้ด้วยหน้าจอ GUI ที่ได้ออกแบบเอาไว้ โดยให้ผู้ใช้สามารถป้อนค่าแล้วส่งผ่านไปให้ Backend เพื่อประมวลผลและส่งกลับมาเพื่อแสดงผล



รูป ก.1 แผนภาพองค์ประกอบส่วน Frontend

โดยสามารถแบ่งย่อยแต่ละส่วนได้ลงไปอีกตามตาราง ก.1 - ก.5

#### ตาราง ก.1 UserManagement component

<b>Component name</b>	UserManagement
<b>Sub-component</b>	LoginPage, RegisterPage, UserSettingPage
<b>Role</b>	แสดงผลในส่วนของหน้าการจัดการข้อมูลผู้ใช้เป็นหลัก
<b>Required Interface</b>	Backend(user_authentication), Backend(user_management)

#### ตาราง ก.2 DriveManagement component

<b>Component name</b>	DriveManagement
<b>Sub-component</b>	ImageFolderPage, DrivePage
<b>Role</b>	แสดงผลในส่วนของหน้าการจัดการรูปภาพที่ผู้ใช้ได้ทำการ Upload ขึ้นมาบน Web application
<b>Required Interface</b>	Backend(user_authentication), Backend(drive_management)

#### ตาราง ก.3 Marketplace component

<b>Component name</b>	Marketplace
<b>Sub-component</b>	MarketPage, ProductPage, EditProductPage, AddProductpage
<b>Role</b>	แสดงผลในส่วนของหน้าติดต่อการซื้อขายและเปิดใช้งานสินค้า
<b>Required Interface</b>	Backend(user_authentication), Backend(product_management), Backend(payment_management)

**ตาราง ก.4 History component**

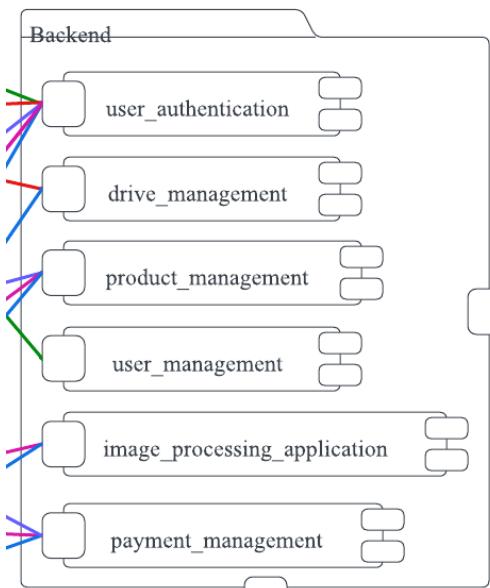
<b>Component name</b>	History
<b>Sub-component</b>	ProductHistoryPage, JobHistoryPage, CreditHistoryPage
<b>Role</b>	แสดงผลในส่วนของหน้าการดูประวัติการทำรายการ ต่างๆ
<b>Required Interface</b>	Backend(user_authentication), Backend(product_management), Backend(image_processing_application), Backend(payment_management)

**ตาราง ก.5 ImageProcessingApplicationPage component**

<b>Component name</b>	ImageProcessingApplicationPage
<b>Sub-component</b>	-
<b>Role</b>	แสดงผลในส่วนของหน้าการใช้งาน Image processing application เช่น การสั่งงานประมวลผล เป็นต้น
<b>Required Interface</b>	Backend(user_authentication), Backend(drive_management), Backend(product_management), Backend(payment_management),

### ก.1.2 ส่วน Backend

รูปที่ ก.2 จะแสดงแผนภาพองค์ประกอบส่วน Frontend ซึ่งมีหน้าที่หลักในการต่อประสานกับผู้ใช้ด้วยหน้าจอ GUI ที่ได้ออกแบบเอาไว้ โดยให้ผู้ใช้สามารถป้อนค่าแล้วส่งผ่านไปให้ Backend เพื่อประมวลผลและส่งกลับมาเพื่อแสดงผล



รูป ก.2 แผนภาพองค์ประกอบส่วน Backend

โดยสามารถแบ่งย่อยแต่ละส่วนได้ลงไปอีกด้านตาราง ก.6 - ก.11

ตาราง ก.6 user\_authentication component

<b>Component name</b>	user_authentication
<b>Sub-component</b>	-
<b>Role</b>	ทำงานในส่วนของการยืนยันตัวผู้ใช้
<b>Required Interface</b>	Storage(Database)

ตาราง ก.7 drive\_management component

<b>Component name</b>	drive_management
<b>Sub-component</b>	-
<b>Role</b>	ทำงานในส่วนของการทำงานจัดการ Drive เก็บรูปภาพ เช่น เพิ่ม, ลบ, ดาวน์โหลด เป็นต้น
<b>Required Interface</b>	Storage(Database), Storage(File storage),

**ตาราง ก.8 product\_management component**

<b>Component name</b>	product_management
<b>Sub-component</b>	-
<b>Role</b>	ทำงานในส่วนของการทำงานจัดการติดตามซื้อขายและสินค้า เช่น เพิ่ม, ลบ, แก้ไข เป็นต้น
<b>Required Interface</b>	Storage(Database), Storage(File storage),

**ตาราง ก.9 user\_management component**

<b>Component name</b>	user_management
<b>Sub-component</b>	-
<b>Role</b>	ทำงานในส่วนของการทำงานจัดการผู้ใช้ เช่น เพิ่มผู้ใช้, แก้ไขข้อมูล เป็นต้น
<b>Required Interface</b>	Storage(Database)

**ตาราง ก.10 image\_processing\_application component**

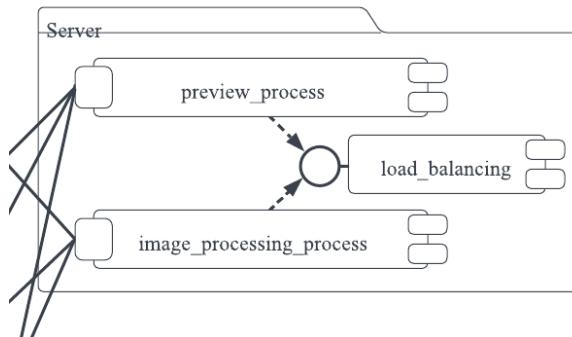
<b>Component name</b>	image_processing_application
<b>Sub-component</b>	-
<b>Role</b>	ทำงานในส่วนของการสั่งงานประมวลผลภาพและ preview รูปภาพ ก่อนประมวลผลจริง
<b>Required Interface</b>	Storage(Database), Storage(File storage), Server(preview_process), Server(image_processing_process), Server(load_balancing),

ตาราง ก.11 payment\_management component

<b>Component name</b>	payment_management
<b>Sub-component</b>	-
<b>Role</b>	ทำงานในส่วนของการทำงานจัดการเครดิตของผู้ใช้ เช่น การบันทึกรายการการเปลี่ยนแปลงเครดิต เป็นต้น
<b>Required Interface</b>	Storage(Database)

### ก.1.3 ส่วน Server

รูปที่ ก.3 จะแสดงแผนภาพองค์ประกอบส่วน Frontend ซึ่งมีหน้าที่หลักในการต่อประสานกับผู้ใช้ด้วยหน้าจอ GUI ที่ได้ออกแบบเอาไว้ โดยให้ผู้ใช้สามารถป้อนค่าแล้วส่งผ่านไปให้ Backend เพื่อประมวลผลและส่งกลับมาเพื่อแสดงผล



รูป ก.3 แผนภาพองค์ประกอบส่วน Server

โดยสามารถแบ่งย่อยแต่ละส่วนได้ลงไปอีกตามตาราง ก.12 - ก.14

ตาราง ก.12 preview\_process component

<b>Component name</b>	preview_process
<b>Sub-component</b>	-
<b>Role</b>	ทำงานในส่วนของการ Preview รูปภาพของผู้ใช้ที่สั่งงานประมวลผลกับสินค้าต่างๆ เพื่อให้ได้รูปผลลัพธ์ที่เร็วที่สุด
<b>Required Interface</b>	Storage(Database), Storage(File storage)

ตาราง ก.13 image\_processing\_process component

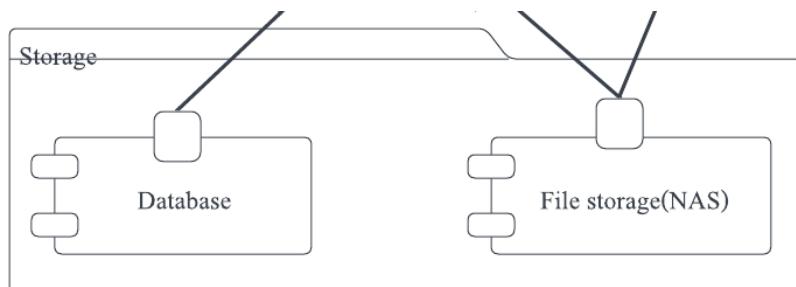
<b>Component name</b>	image_processing_process
<b>Sub-component</b>	-
<b>Role</b>	ทำงานในส่วนของการประมวลผลรูปจากการที่ผู้ใช้ได้สั่งประมวลผลเข้ามา โดยจะรองรับงานประมวลผลจำนวนมากๆ
<b>Required Interface</b>	Storage(Database), Storage(File storage)

ตาราง ก.14 load balancing component

<b>Component name</b>	load balancing
<b>Sub-component</b>	-
<b>Role</b>	ทำงานในส่วนของการแบ่งปันจัดสรรทรัพยากรในการประมวลผลให้กับส่วนอื่นๆ
<b>Required Interface</b>	-
<b>Provided interfaces</b>	Server(image_processing_process), Server(preview_process)

#### ก.1.4 ส่วน Storage

รูปที่ ก.4 จะแสดงแผนภาพองค์ประกอบส่วน Frontend ซึ่งมีหน้าที่หลักในการต่อประสานกับผู้ใช้ด้วยหน้าจอ GUI ที่ได้ออกแบบเอาไว้ โดยให้ผู้ใช้สามารถป้อนค่าแล้วส่งผ่านไปให้ Backend เพื่อประมวลผลและส่งกลับมาเพื่อแสดงผล



รูป ก.4 แผนภาพองค์ประกอบส่วน Storage

โดยสามารถแบ่งย่อยแต่ละส่วนได้ลงไปอีกตามตาราง ก.15 และ ก.16

ตาราง ก.15 Database component

<b>Component name</b>	Database
<b>Sub-component</b>	-
<b>Role</b>	เป็นฐานข้อมูลในการเก็บข้อมูลต่างๆ เพื่อนำไปใช้แสดงผลและประมวลผล
<b>Required Interface</b>	-

ตาราง ก.16 File storage(NAS)component

<b>Component name</b>	File storage(NAS)
<b>Sub-component</b>	-
<b>Role</b>	เป็นส่วนของพื้นที่การเก็บไฟล์รูปภาพและไฟล์ Weight ของผู้ใช้เพื่อสามารถนำไปประมวลผลได้
<b>Required Interface</b>	-

# ภาคผนวก ข

## Test case

### ๑.1 Frontend test

Frontend test เป็นการทดสอบหน้า Web application ส่วนแสดงผลหลักว่าสามารถแสดงผลได้ถูกต้องตรงตามที่ได้ออกแบบเอาไว้และสามารถเชื่อมต่อและ ส่ง-รับค่าต่างๆ จากทาง API Backend ได้โดยไม่เกิดข้อผิดพลาดขึ้น โดยมีทั้งหมด 103 case ที่ใช้ในการทดสอบดังตารางที่ ๑.๑

### ๑.1 Frontend testcase

Test ID	Test description
F-001	Register form cannot be null
F-002	Register form password and password confirm check
F-003	Register form call API backend correctly when press regis button
F-004	Register form clear all input values when press clear button
F-005	Register form deny special character in First name and Last name input
F-006	Login form cannot be null
F-007	Login form call API backend correctly when press Login button
F-008	Login page can enter home page when login process is success
F-009	Login page add user token to local storage in web browser when login process is success
F-010	Login page can show pop up window to alert when login process is unsuccess
F-011	Go to Register page when press register button
F-012	Sidebar can show pop up window to alert when user token is not existed and redirection to Login page
F-013	Sidebar display user data correctly
F-014	Sidebar can link to JobHistory page
F-015	Sidebar can link to History page
F-016	Sidebar can link to Home page
F-017	Sidebar can link to Drive page
F-018	Sidebar can link to Setting page

<b>Test ID</b>	<b>Test description</b>
F-019	Sidebar can link to Image processing applications page
F-020	Sidebar can link to Marketplace page
F-021	Sidebar delete user token from local storage in web browser when press logout button
F-022	Home page call API backend and get display data correctly
F-023	Home page show 9 latest product
F-024	Home page can enter Product page when click on product card
F-025	Home page can link to Marketplace page with see more button
F-026	JobHistory page call API backend and get display data correctly
F-027	JobHistory page can display user job order history correctly
F-028	JobHistory page can change display user job order history list to latest record first
F-029	JobHistory page can change display user job order history list to oldest record first
F-030	JobHistory page can change display user job order history list to specific date only
F-031	History page call API backend and get display data correctly
F-032	History page can display user credit earn and pay history correctly
F-033	History page can change display user credit earn and pay history list to latest record first
F-034	History page can change display user credit earn and pay history list to oldest record first
F-035	History page can change display user credit earn and pay history list to specific date only
F-036	Drive page call API backend and get display data correctly
F-037	Drive page can display all image folder that user owned
F-038	Drive page can send new image folder data to API backend
F-039	Drive page can sent delete image folder data to API backend
F-040	Drive page can download chosen image folder in zip file format
F-041	Drive page can enter chosen image folder
F-042	Image folder page can authenticate image folder owner user
F-043	Image folder page can show pop up window to alert when user that authenticate is not image folder owner user
F-044	Image folder page can send upload images data to backend by press upload button
F-045	Image folder page can sent delete image to API backend by press delete button on image

<b>Test ID</b>	<b>Test description</b>
F-046	Image folder page can make web browser download image data from API backend by press download button on image
F-047	Image folder page can select range of image to display by select options image folder page
F-048	Image folder page can show full quality image by press on image card
F-049	Marketplace page can call API backend and get display data correctly
F-050	Marketplace page can display products data correctly
F-051	Marketplace page can go to Add product page by press add product button
F-052	Marketplace page can change display products order to latest product update first
F-053	Marketplace page can change display products order to oldest product update first
F-054	Marketplace page can change display products order to specific date only
F-055	Marketplace page can change display products type filter
F-056	Marketplace page can change display products model filter
F-057	Marketplace page can go to Product page by press on product card
F-058	Product page can call API backend and get display data correctly
F-059	Product page can display product data correctly
F-060	Product page can show pop up window to alert when product is not exited and redirection to Marketplace page
F-061	Product page can show edit this product button and view product use history button when user is authenticating as product owner
F-062	Product page can add product to open product list by press try this product button
F-063	Add product page can call API backend
F-064	Add product page user can filled all form
F-065	Add product page user can upload product image when press choose product image upload button
F-066	Add product page user can upload product weight file when press choose product weight file upload button
F-067	Add product page can check all form are filled when press add product button
F-068	Add product page can show pop up window to alert when add product process is success
F-069	Add product page can show pop up window to alert when add product process is unsuccess

<b>Test ID</b>	<b>Test description</b>
F-070	Edit product data page can call API backend and display product data correctly
F-071	Edit product data page user can filled all form
F-072	Edit product data page user can upload product image when press choose product image upload button
F-073	Edit product data page user can upload product weight file when press choose product weight file upload button
F-074	Edit product data page can show pop up window to alert when edit product data process is success
F-075	Edit product data page can show pop up window to alert when edit product data process is unsuccess
F-076	User setting page can call API backend and get display data correctly
F-077	User setting page can display user data correctly
F-078	User setting page all form must be filled when press submit button
F-079	User setting page user can change only first name and last name of user data
F-080	User setting page can send changed user data to API backend when press submit button
F-081	User setting page can show pop up window to alert when change is not complete
F-082	User setting page can show pop up window to alert when change is complete
F-083	User setting page can go to Change password page when press change password button
F-084	User setting page can clear value in all input form by press clear button
F-085	Change password page display user data correctly
F-086	Change password page can check password and password confirm is match
F-087	Change password page all form must be filled when press submit button
F-088	Change password page can send user old password and new password to API backend when press
F-089	Change password page can show pop up window to alert when change is not complete
F-090	Change password page can show pop up window to alert when change is complete
F-091	Change password page can clear value in all input form by press clear button
F-092	Image processing application page can call API backend and get display data correctly
F-093	Image processing application page can display all image folders of user correctly

Test ID	Test description
F-094	Image processing application page can display list of all images from user chosen image folder when select image folder in image folder list
F-095	Image processing application page can display user chosen image from list of all images from chosen image folder when press image thumbnail
F-096	Image processing application page can display all product that user opened correctly
F-097	Image processing application page user can select product to preview product processing result with chosen image
F-098	Image processing application page user can change parameter of preview product processing result with chosen image
F-099	Image processing application page can display result image of preview product processing result
F-100	Image processing application page user can send order job to API backend when press export button
F-101	Image processing application page can pop up window to alert when credit payment is required
F-102	Image processing application page can pop up window to alert when job is on processing
F-103	Image processing application page can pop up window to alert when job is failed to process

## ๔.2 Backend test

Backend test เป็นการทดสอบการทำงานเบื้องหลังในส่วนของ Backend ส่วนประมวลผลหลักว่า สามารถประมวลผลและจัดเก็บข้อมูลที่ได้รับจากส่วน Frontend ได้ถูกต้องตรงตามที่ได้ออกแบบเอาไว้ โดยไม่เกิดข้อผิดพลาดขึ้น โดยมีทั้งหมด 51 case ที่ใช้ในการทดสอบดังตารางที่ ๔.2

## ๔.2 Backend testcase

Test ID	Test description
B-001	Backend can start and using at configuration IP address
B-002	Backend can display backend develop version at URL /API/ for checking version
B-003	Backend can add new user data to database
B-004	Backend can send response message back to frontend if able to regis new user
B-005	Backend can send response error message back to frontend if unable to regis new user

<b>Test ID</b>	<b>Test description</b>
B-006	Backend can make and send response with user authenticate token and user data back to frontend if able to login
B-007	Backend can add user login log to database
B-008	Backend can send response error message back to frontend if unable to login
B-009	Backend must authenticate user token in any method when have call from frontend
B-010	Backend can edit user data in database
B-011	Backend can send response message back to frontend if able to edit user data
B-012	Backend can send response error message back to frontend if unable to edit user data
B-013	Backend can edit user password in database
B-014	Backend can send response message back to frontend if able to edit user password
B-015	Backend can send response error message back to frontend if unable to edit user password
B-016	Backend can send response massage with user job order history query to frontend
B-017	Backend can send response massage with user credit earn and pay history query to frontend
B-018	Backend can send response massage with user image folders quarry to frontend
B-019	Backend can add new image folder data to database and create folder in NAS
B-020	Backend can send response massage to frontend if able to add new image folder
B-021	Backend can send response error massage to frontend if unable to add new image folder
B-022	Backend can add delete image folder data to database and delete folder in NAS
B-023	Backend can send response massage to frontend if able to delete image folder
B-024	Backend can send response error massage to frontend if unable to delete image folder
B-025	Backend can receive image data and image files to add to database and store in NAS
B-026	Backend can send response massage to frontend if able to add new image
B-027	Backend can send response error massage to frontend if unable to add new image
B-028	Backend can send response massage with compress image data in selected image folder to frontend
B-029	Backend can send response massage with full size image data of selected image to frontend
B-030	Backend can delete image data from database and delete image from NAS
B-031	Backend can send response massage to frontend if able to delete new image
B-032	Backend can send response error massage to frontend if unable to delete new image

<b>Test ID</b>	<b>Test description</b>
B-033	Backend can send response massage to frontend if able to download new image
B-034	Backend can send response error massage to frontend if unable to download new image
B-035	Backend can send response massage with product list query to frontend
B-036	Backend can send response massage with product data to frontend
B-037	Backend can send response error massage to frontend if unable to query product list
B-038	Backend can send response error massage to frontend if unable to find product
B-039	Backend can receive product image file ,weight file and product data to add to database and store in NAS
B-040	Backend can send response massage to frontend if able to add new product data
B-041	Backend can send response error massage to frontend if unable to add new product data
B-042	Backend can receive product image file ,weight file and product data to edit product data database and store in NAS
B-043	Backend can send response massage to frontend if able to edit product data
B-044	Backend can send response error massage to frontend if unable to edit product data
B-045	Backend can receive open product data to add to database
B-046	Backend can receive image data and product data for send to API preview and get result image
B-047	Backend can send preview result image to frontend
B-048	Backend can send response massage with credit payment require to frontend
B-049	Backend can receive product data, parameter and image folder for order image processing job
B-050	Backend can send response massage to frontend if able to order image processing job
B-051	Backend can send response error massage to frontend if unable to order image processing job