

ระบบซื้อขายสินค้าออนไลน์เกี่ยวกับสวนและแนวการจัดสวน (Leafy)

นางสาวปณาลี ปาลศรี 63130500075

นายพีรพัฒน์ ค้าเกิด 63130500088

นายสหทัศน์ ยิ่งสกุลเกียรติ 63130500113

รายงานนี้เป็นส่วนหนึ่งของรายวิชาโครงการรายบุคคลด้านเทคโนโลยีสารสนเทศ
หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีสารสนเทศ
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
ปีการศึกษา 2566

ลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

หัวข้อโครงการรวมยอด	ระบบซื้อขายสินค้าออนไลน์เกี่ยวกับสวนและแนวทางจัดสวน
หน่วยกิต	6
รายชื่อสมาชิก	นางสาวปณาลี ปาลศรี นายพิรพัฒน์ ค้าเกิด นายสหทัศน์ ยิ่งสกุลเกียรติ
อาจารย์ที่ปรึกษา	อาจารย์พิเชฐฐ์ ลิมวชิรานันต์
หลักสูตร	วิทยาศาสตรบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
คณะ	เทคโนโลยีสารสนเทศ
ปีการศึกษา	2566

บทคัดย่อ

Leafy เป็นเว็บแอปพลิเคชันสำหรับการซื้อขายสินค้าออนไลน์เกี่ยวกับสวนและแนวทางจัดสวน ซึ่งมี Features หลักอย่างเช่น การแนะนำสินค้าด้วย Machine Learning และระบบจัดการสินค้าที่ยืดหยุ่น เป็นต้น เว็บแอปพลิเคชันนี้ถูกพัฒนาด้วย Vue JS และ Express JS Frameworks ที่เชื่อมต่อด้วยฐานข้อมูลแบบ Relational Database Management System (RDBMS) โดยใช้เป็น MySQL และ Minio ใช้ในการเก็บไฟล์ข้อมูล เช่นรูปภาพ แบบ Object Storage ด้วยการ Deploy สภาพแวดล้อมแบบ Docker container บน Server ของคณะ โดยมีการแบ่ง Container การทำงานที่ชัดเจนในส่วนของ Proxy, Frontend, Backend, File location และ Database

คำสำคัญ: การจัดสวน/ เครื่องมือและอุปกรณ์การจัดสวน/ ต้นไม้/ แพลตฟอร์มการซื้อขาย/
ระบบแนะนำสินค้า

Project Title	Online Gardening Product Trading Platform and Landscaping (Leafy)
Project Credits	6
Members	Miss Panalee Palasri Mr. Phiraphat Kakerd Mr. Sahathat Yingsakulkiet
Project Advisor	Lect. Pichet Limvachiranana
Program	Bachelor of Science
Field of Study	Information Technology
Faculty	School of Information Technology
Academic Year	2023

Abstract

Leafy is a web application platform for buying and selling online products related to gardens and landscaping. Its main features include product recommendations using machine learning and a flexible inventory management system. It was developed using the Vue.js and Express.js frameworks, connected to a Relational Database Management System (RDBMS) utilizing MySQL. Minio was also used for storing data files such as images in an object storage format. The application is deployed in a Docker container environment on the faculty's server, with clear container separation for the proxy, frontend, backend, file location, and database.

Keywords: Gardening/ Gardening Tools and Equipment/ Plants/ Product Recommendation System/ Trading Platform

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
สารบัญ	ค
บทที่	
1. บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	2
1.3 ภาพรวมระบบ	2
1.4 จุดเด่นของระบบ	6
1.5 แผนการส่งมอบงาน (Roadmap)	6
2. การศึกษาความเป็นไปได้	8
2.1 การศึกษาความเป็นไปได้	8
2.2 งานวิจัยหรือโครงการที่เกี่ยวข้อง	9
3. การออกแบบเชิงเทคนิค	11
3.1 Product Backlog	11
3.2 System Architecture	39
3.3 การออกแบบซอฟต์แวร์ (Software Design)	41
3.3.1 UX/UI	41
3.3.2 Frontend	43
3.3.3 Backend	59
3.3.4 Machine Learning	86
3.3.5 Database	90
3.4 การออกแบบกรณีทดสอบ (Test Cases)	104
3.4.1 Frontend Testing	104
3.4.2 Backend Testing	138

สารบัญ (ต่อ)

	หน้า
4. ผลการดำเนินการ	139
เอกสารอ้างอิง	140

บทที่ 1 บทนำ

ในยุคปัจจุบันที่เทคโนโลยีและอินเทอร์เน็ตมีบทบาทสำคัญในการให้บริการผ่านช่องทางออนไลน์ที่กำลังเติบโตอย่างรวดเร็ว การซื้อขายสินค้าผ่านแพลตฟอร์มออนไลน์เป็นสิ่งสำคัญอย่างยิ่ง อย่างไรก็ตาม การค้นหาร้านค้าที่มีสินค้าตามความต้องการสำหรับผู้ที่ชื่นชอบการจัดสวนยังคงเป็นเรื่องที่มีความยากลำบาก ในบทนี้จะพูดถึงความเป็นมาและความสำคัญของปัญหา ซึ่งจะเข้าใจถึงปัญหาที่เกิดขึ้นในการซื้อขายสินค้าเกี่ยวกับสวน รวมถึงวัตถุประสงค์ของระบบนี้ โดยมุ่งเน้นที่การบริการลูกค้าและการเพิ่มโอกาสในการขยายธุรกิจ นอกจากนี้ยังมีการกล่าวถึงจุดเด่นของระบบและแผนการส่งมอบงาน (Roadmap) ที่จะนำเสนอในบทต่อไป

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันเป็นยุคที่เทคโนโลยีและอินเทอร์เน็ตมีบทบาทสำคัญในการให้บริการผ่านช่องทางออนไลน์ที่กำลังเติบโตอย่างรวดเร็ว การซื้อขายสินค้าหรือบริการผ่านแพลตฟอร์มออนไลน์เป็นเรื่องที่สำคัญมาก รวมถึงกลุ่มคนที่ชื่นชอบการปลูกต้นไม้และการจัดสวน แต่ยังมีความไม่สะดวกในการซื้อขายสินค้าที่ตรงตามความต้องการ ทำให้การซื้อขายต้นไม้และอุปกรณ์จัดสวนต่างๆ เป็นไปได้ยาก สำหรับบุคคลที่ไม่ใช่ผู้เชี่ยวชาญด้านความหลากหลายและรวดเร็ว

เว็บแอปพลิเคชันนี้จะเพิ่มโอกาสทางธุรกิจให้กับผู้ขาย ทำให้ผู้ใช้งานสามารถเข้าถึงสินค้าที่ต้องการได้อย่างสะดวกและรวดเร็ว ดังนั้น คณะผู้จัดทำจึงเล็งเห็นปัญหาที่เกิดขึ้น และมีแนวคิดที่จะพัฒนาเว็บแอปพลิเคชัน ระบบซื้อขายสินค้าออนไลน์เกี่ยวกับสวนและแนวการจัดสวน (Leafy) ขึ้นมา เพื่ออำนวยความสะดวกให้กับผู้ที่ต้องการซื้อขายสินค้า

การพัฒนาเว็บแอปพลิเคชัน ระบบซื้อขายสินค้าออนไลน์เกี่ยวกับสวนและแนวการจัดสวน (Leafy) จึงเป็นการตอบสนองต่อปัญหาดังกล่าว โดยมีการใช้ทฤษฎีและหลักการทำงานเทคโนโลยีในการสร้างระบบแนะนำสินค้า การจัดการตระกร้าสินค้าที่รองรับการสั่งซื้อจากหลายร้านค้า และการสร้างระบบการจัดการสินค้าที่ผู้ใช้งานสามารถเลือกซื้อสินค้าจากหลากหลายหมวดหมู่ได้อย่างสะดวก นอกจากนี้ ระบบยังมีการออกแบบ UX/UI ที่ใช้งานง่ายและมีประสิทธิภาพ รวมถึงการจัดการด้านความปลอดภัยที่เหมาะสมสำหรับผู้ใช้ทุกคน

1.2 วัตถุประสงค์

1.2.1 ระบบที่มีคุณภาพ

- 1.2.1.1 เพื่อร่วบรวมข้อมูลการสินค้าและการออกแบบเกี่ยวกับสวนในที่เดียว
- 1.2.1.2 เพื่อเพิ่มแพลตฟอร์มออนไลน์ในรูปแบบใหม่ๆ ที่เข้าถึงกลุ่มเป้าหมายและ การบริการให้แก่ผู้ซื้อและผู้ขาย
- 1.2.1.3 เพื่อเป็นแรงบันดาลใจในการออกแบบจัดสวน

1.2.2 การบริการลูกค้า

- 1.2.2.1 เพื่ออำนวยความสะดวกให้แก่ลูกค้าและได้รับการบริการที่ดี
- 1.2.2.2 เพื่อสร้างความสัมพันธ์ระยะยาวระหว่างผู้ขายและลูกค้า

1.2.3 ประสิทธิภาพในการซื้อขาย

- 1.2.3.1 เพื่อเพิ่มยอดขายสินค้าของผู้ขาย
- 1.2.3.2 เพื่ออำนวยความสะดวกให้แก่ผู้ขายและผู้ซื้อในการซื้อขายสินค้า
- 1.2.3.3 เพื่อลดค่าใช้จ่ายในกระบวนการต่างๆ

1.2.4 การเพิ่มโอกาสในการขยายธุรกิจ

- 1.2.4.1 เพื่อเพิ่มโอกาสในการสร้างรายได้ให้กับผู้ขาย
- 1.2.4.2 เพื่อสร้างโอกาสในการขยายธุรกิจ

1.3 ภาพรวมระบบ

ระบบซื้อขายสินค้าออนไลน์เกี่ยวกับสวนและแนวทางการจัดสวน (Leafy) เป็นระบบเว็บแอป พลิกาชันที่รวบรวมทุกอย่างเกี่ยวกับสวนไว้ในที่เดียว โดยสามารถซื้อขายสินค้าและดูแนวทางการจัดสวน พร้อมทั้งแก้ปัญหาสำหรับกลุ่มผู้ขายต้นไม้ในเว็บต่างๆ ให้มีคุณสมบัติมากยิ่งขึ้น ระบบเน้นถูกออกแบบมาเพื่อตอบสนองความต้องการของผู้ใช้และเพิ่มโอกาสทางธุรกิจให้กับผู้ขาย ด้วยฟีเจอร์ การแนะนำสินค้า ระบบตัคร้าสินค้าที่ยืดหยุ่น และระบบจัดการสินค้าที่หลากหลาย ทำให้ Leafy เป็นเครื่องมือที่มีประสิทธิภาพและสามารถตอบสนองความต้องการของผู้ใช้และผู้ขายในระดับสูงสุด

1.3.1 กลุ่มเป้าหมาย

- 1.3.1.1 ผู้ซื้อที่มีความสนใจเกี่ยวกับการจัดสวนหรือต้นไม้ต่างๆ และชื่นชอบในการปลูกต้นไม้
- 1.3.1.2 บุคคลที่มีความต้องการขายสินค้า เพื่อให้ตอบสนองความต้องการสินค้า ในปัจจุบัน และเพิ่มช่องทางในการหารายได้

1.3.2 ฟีเจอร์หรือฟังก์ชันการทำงานที่เป็นความต้องการของระบบ

- 1.3.2.1 ระบบมีการ Login และการ Register User

- 1.3.2.2 สามารถเลือกซื้อสินค้าทางออนไลน์ โดยระบบมี Machine Learning ใช้ในการทำ Recommend สินค้าที่ลูกค้าสนใจได้
- 1.3.2.3 สามารถค้นหาสินค้าที่ต้องการ ผ่าน Search bar ปุ่มคัดกรอง และเรียงลำดับจากสิ่งที่ลูกค้าต้องการได้
- 1.3.2.4 สามารถรองรับสินค้าจากหลายร้านลงทะกراและสร้าง Order ในแต่ละร้านได้
- 1.3.2.5 สามารถชำระเงินผ่านระบบจำลองและดูสถานะการจัดส่งสินค้าของแต่ละร้านได้
- 1.3.2.6 มีระบบจัดการสินค้าสำหรับผู้ขายที่มีความยืดหยุ่น
- 1.3.2.7 สามารถสร้างเนื้อหาเกี่ยวกับการจัดสวน (Garden content) เป็นแนวทางในการจัดสวนสำหรับคนที่สนใจ

ระบบซื้อขายสินค้าออนไลน์เกี่ยวกับสวนและแนวการจัดสวน ประกอบด้วยผู้ใช้งาน 4 บทบาท ได้แก่ ผู้เยี่ยมชม, ผู้ใช้งาน, ผู้ขายสินค้า, และผู้ดูแลระบบ โดยแต่ละบทบาทสามารถเข้าถึงฟีเจอร์ได้แตกต่างกัน ดังแสดงในตารางที่ 1.1

ตารางที่ 1.1 ตารางฟีเจอร์ในระบบและสิทธิ์ในการเข้าใช้งานของผู้ใช้ในแต่ละบทบาท

Feature / Role	ผู้เยี่ยมชม	ผู้ขายสินค้า	ผู้ใช้งาน	ผู้ดูแลระบบ
Landing Page	/	/	/	/
Sign In		/	/	/
Register	/			/
View Account	/	/	/	/
Edit Personal Account		/	/	/
Manage Users				/
View Product List	/	/	/	/
View Product Detail	/	/	/	/
Search / Filter Product	/	/	/	/
Create Products / SKU		/		/
Edit Product / SKU		/		/
Delete Products / SKU		/		/
Add Product to Cart		/	/	/
Increment/Decrement Cart		/	/	/

ตารางที่ 1.1 ตารางฟีเจอร์ในระบบและสิทธิ์ในการเข้าใช้งานของผู้ใช้ในแต่ละบทบาท (ต่อ)

Feature / Role	ผู้เยี่ยมชม	ผู้ขายสินค้า	ผู้ใช้งาน	ผู้ดูแลระบบ
Delete Product from Cart	/	/	/	/
Buy Product/Project	/	/	/	/
Place order	/	/	/	/
Order Checker	/	/	/	/
Change order status (payment, received, canceled)	/	/	/	/
Change order status (in progress)	/	/		/
View Content List	/	/	/	/
View Content Detail	/	/	/	/
Create Content		/	/	/
Edit Content		/	/	/
Delete Content		/	/	/
Review Product		/	/	/
Read Review	/	/	/	/
Like Review		/	/	/
Update Review		/	/	/
Delete Review		/	/	/
Comment content		/	/	/
Read comment		/	/	/
Delete comment		/	/	/

1.3.3 ประเภทของระบบ

Leafy เป็นเว็บแอปพลิเคชันที่ประกอบด้วย 3 ส่วน ได้แก่ ส่วนFrontend ส่วน Backend และส่วนของฐานข้อมูล ซึ่งถูกออกแบบให้ใช้งานบนอุปกรณ์ทั่วไปได้ เช่น คอมพิวเตอร์ โน้ตบุ๊ค แท็บเล็ต และโทรศัพท์มือถือ เป็นต้น

1.3.4 Technology Stack

1.3.4.1 Frontend

1.3.4.1.1 Markup Language: HTML, CSS

1.3.4.1.2 Script Language: JavaScript

1.3.4.1.3 Framework: Vue.JS

1.3.4.1.4 Promise Based HTTP Client: AXIOS

1.3.4.2 Backend

1.3.4.2.1 Language: Node.JS

1.3.4.2.2 Framework: Express.JS

1.3.4.2.3 Authentication: jsonwebtoken

1.3.4.2.4 File upload: s3-multer

1.3.4.2.5 Object Relational Mapping: Prisma

1.3.4.3 Database and Files storage

1.3.4.3.1 Language: MySQL

1.3.4.3.2 Objects storage: Minio

1.3.4.4 Infrastructure

1.3.4.4.1 Containerization: Docker, Docker-compose

1.3.4.4.2 Server Engine: Nginx

1.3.4.4.3 OS: Ubuntu

1.3.4.4.4 Server: SIT KMUTT Virtual Machine

1.3.4.5 Machine Learning

1.3.4.5.1 Recommender System: Node.JS

1.3.4.6 Development Tools

1.3.4.6.1 Code Editor: Visual Studio Code

1.3.4.6.2 API Testing: Postman

1.3.4.6.3 Design: Figma, Canva

1.3.4.6.4 Git Version Control: GitHub, Source tree

1.3.4.7 Project Management Tools

1.3.4.7.1 Meeting: Microsoft Teams

1.3.4.7.2 Communication: Line

1.3.4.7.3 Project planning: ClickUp, Excel

1.3.4.7.4 Research planning and brainstorming tool: FigJam

1.3.5 ขอบเขต/ข้อจำกัดของโครงการ

1.3.5.1 Frontend ใช้ Vue.js framework เนื่องจากทีมงานมีประสบการณ์ พัฒนาที่มากกว่าและใช้การเขียน CSS โดยไม่ใช้ Framework เพื่อเป็น การท้าทายทักษะ

1.3.5.2 Backend ในกรณีที่ระบบขยายใหญ่ขึ้น อาจทำให้มีค่าใช้จ่ายเพิ่มขึ้นและ ต้องใช้เซิร์ฟเวอร์มากขึ้นในการจัดการ express.js ซึ่งหมายความว่ากับ

backend ที่ต้องการความยืดหยุ่นสูง ซึ่งระบบจ่ายเงินต้องเชื่อมต่อกับเซิร์ฟเวอร์กลางในการดำเนินการชำระเงิน และระบบจัดการซองทางการเงินมีความซับซ้อนในการจ่ายเงินผ่านหลายร้านค้าในครั้งเดียว ทำให้ระบบจ่ายเงินยังไม่ได้ถูกนำมาใช้จริงในโครงการนี้

- 1.3.5.3 Recommender system สามารถแนะนำสินค้าได้ แต่ยังไม่มีความเป็นส่วนตัวมากนักสำหรับผู้เยี่ยมชม (Guest) และยังไม่มีการวัดประสิทธิภาพ เนื่องจากข้อมูลใน dataset ต้องมีขนาดใหญ่เพียงพอ

1.4 จุดเด่นของระบบ

- 1.4.1 ระบบแนะนำสินค้า (Recommender System) แนะนำสินค้าตามความสนใจของลูกค้าแนะนำสินค้าตามความสนใจของลูกค้า เพื่อเสนอตัวเลือกในการซื้อสินค้าอย่างมีประสิทธิภาพและตรงกับความต้องการของลูกค้า
- 1.4.2 ระบบติดกร้าสินค้าแบบหลายร้านค้า รองรับการสั่งซื้อจากหลายร้านค้าพร้อมทั้งการชำระเงินที่สามารถปรับตัวเข้ากับกระบวนการที่ซับซ้อนได้ เพื่อความสะดวกในการสั่งซื้อสินค้า
- 1.4.3 ระบบจัดการสินค้าที่หลากหลาย จัดการสินค้าที่ผู้ใช้สามารถเลือกซื้อสินค้าจากหลากหลายหมวดหมู่ได้อย่างง่ายดายและรวดเร็ว
- 1.4.4 ระบบ Content การจัดสวน แสดงแนวการจัดสวนต่างๆ ตามความสนใจของผู้ใช้ เพื่อเป็นแรงบันดาลใจและแนวทางในการจัดสวน
- 1.4.5 การออกแบบ UX/UI ที่ใช้งานง่ายและมีประสิทธิภาพ เพื่อเพิ่มประสบการณ์การใช้งานที่ดีให้กับผู้ใช้
- 1.4.6 การใช้ Framework ที่มีประสิทธิภาพ ในการเพิ่มประสิทธิภาพและความสามารถในการทำงานของระบบ ทำให้ระบบมีความเสถียรและสามารถรองรับการใช้งานในปริมาณมากได้
- 1.4.7 ระบบความปลอดภัย (Security) การจัดการสิทธิ์ต่างๆ อย่างเชี่ยวชาญสำหรับบทบาทต่างๆ เช่น Admin, Supplier, User และ Guest เพื่อปกป้องข้อมูลและความเป็นส่วนตัวของผู้ใช้

1.5 แผนการส่งมอบงาน (Roadmap)

Capstone Project I-II Milestone [Working SW]

- | | |
|--|---------------------|
| 1. วันที่ 26 ตุลาคม – 22 พฤศจิกายน 2566 | => Core Feature 30% |
| 2. วันที่ 23 พฤศจิกายน 2566 – 10 มกราคม 2567 | => Core Feature 40% |

3. วันที่ 10 มกราคม - 19 กุมภาพันธ์ 2567 => Core Feature 60%
4. วันที่ 20 กุมภาพันธ์ - 23 เมษายน 2567 => Core Feature 100%

Release	1	2
Date	26 October – 22 November 2023	23 November 2023 – 10 January 2024
Goal	Basic web e-com & Basic API	View Accounting, Role based system
Features	Home page Product shop Product details of User Frontend can connect API from Backend Backend API with connected database and file storage Use HTTP connection	Frontend can manage API from Backend Improve security (authentication with cookies and authorization) Improve product list and product detail View other supplier profile

Release	3	4
Date	11 January - 19 February 2024	20 February - 23 April 2024
Goal	Accounting & Recommender system	Product Review, Garden design content & Chat system
Features	User account setting Recommender system Improve product list and product detail HTTPS protocol	Supplier owner shop Product Payment, Tracking and Review Product admin system Garden design content Order & Cart system Basic chat app (cannot communicate between user and user) Improve UI

บทที่ 2 การศึกษาความเป็นไปได้

การศึกษาความเป็นไปได้เป็นขั้นตอนสำคัญในการพัฒนาระบบการขายสินค้าออนไลน์ การวิเคราะห์ความเป็นไปได้ช่วยให้สามารถมองเห็นภาพรวมของโครงการ รวมถึงการประเมินศักยภาพในการนำระบบไปใช้จริง นอกจากนี้ การศึกษาดังครอบคลุมถึงการวิเคราะห์ตลาดและกลุ่มเป้าหมาย การตรวจสอบความต้องการและความคาดหวังของผู้ใช้ การพิจารณาเทคโนโลยีที่ใช้ในการพัฒนา และการประเมินความเสี่ยงต่างๆ ที่อาจเกิดขึ้น การค้นคว้าและวิเคราะห์เหล่านี้จะช่วยให้สามารถวางแผนการพัฒนาและการดำเนินงานได้อย่างมีประสิทธิภาพ

2.1 การศึกษาความเป็นไปได้ (Feasibility Studies)

2.1.1 การศึกษาความเป็นไปได้ทางเทคนิค

2.1.1.1 UX/UI: การออกแบบ (UX/UI) ได้ศึกษาทฤษฎีที่เกี่ยวข้องและมีการทดลองปฏิบัติมาก่อน

2.1.1.2 Frontend: ได้มีการทดสอบสร้าง prototype เป็นต้น ทำให้มั่นใจว่าสามารถพัฒนาได้จริง

2.1.1.3 Backend: ได้ทดลองทำ Production และปรับปรุงประสิทธิภาพ รวมถึงได้ศึกษาทฤษฎีที่เกี่ยวข้อง เช่น Role-based system, Product, Cart, Order และ Content designer ทำให้มั่นใจว่าสามารถพัฒนาได้จริง

2.1.1.4 Infrastructure: ได้ทดลอง Deploy server บน VM ของที่ฝึกงานและแก้ไขปัญหาต่างๆ เพิ่มเติม เช่น Database ที่ไม่สามารถเชื่อมต่อได้ โดยได้รับคำแนะนำจากผู้เชี่ยวชาญ ทำให้มั่นใจว่าสามารถพัฒนาได้จริง

2.1.1.5 Machine Learning: ได้สอบถามผู้คนที่สนใจในจัดสวนและจัดทำข้อมูลจำลองเพื่อสร้าง Model ทดสอบและผลที่ได้เป็นที่น่าพอใจ ทำให้มั่นใจว่าสามารถพัฒนาได้จริง

2.1.1.6 Database: สามารถเก็บข้อมูลเบื้องต้นได้ เนื่องจากโครงสร้างของ Data มีความซับซ้อน จึงต้องศึกษาเพิ่มเติม เพื่อเพิ่มประสิทธิภาพในการเก็บข้อมูลให้ดีมากขึ้น

2.1.1.7 File Storage: ได้ทำการศึกษาและทดสอบวิธีการเก็บข้อมูลรูปภาพ โดยพิจารณา 2 ตัวเลือกหลัก ได้แก่

2.1.1.7.1 Firebase Storage: เป็น Storage service จาก Google ที่ใช้งานง่าย และมีประสิทธิภาพ แต่มีข้อจำกัดเรื่องการเรียกใช้ของจำนวน Request และพื้นที่สำหรับเก็บข้อมูล

2.1.1.7.2 Minio: เป็น Object Storage Service ที่สามารถทดแทน AWS S3 ได้ และสามารถใช้งานได้ฟรี

2.1.2 การศึกษาความเป็นไปได้ทางธุรกิจ

การศึกษาความเป็นไปได้ทางธุรกิจ โดยเน้นไปที่ความต้องการของผู้ใช้งาน กระบวนการที่จะสามารถตอบโจทย์และสร้างความพึงพอใจให้กับผู้ใช้ได้ การวิเคราะห์คู่แข่ง แนวโน้มความต้องการของตลาด และการเลือกใช้กลยุทธ์เพื่อรักษาผู้ใช้งาน นอกจากนี้ยังต้องคำนึงถึงต้นทุนในการพัฒนาเว็บแอปพลิเคชัน เพื่อให้โครงการประสบความสำเร็จ

2.2 โครงการที่เกี่ยวข้อง

ในการพัฒนาระบบซื้อขายสินค้าออนไลน์เกี่ยวกับสวนและแนวทางจัดสวน ได้ทำการศึกษาโครงการและระบบที่ใกล้เคียงเพื่อเป็นแนวทางในการออกแบบและพัฒนาระบบจัดการสินค้า โครงการที่ศึกษา ได้แก่ เว็บแอปพลิเคชัน Shopee และเว็บไซต์ Nike ซึ่งแต่ละโครงการมีคุณสมบัติและฟีเจอร์ที่น่าสนใจและสามารถนำมาประยุกต์ใช้ในการพัฒนาระบบจัดการสินค้าได้

2.2.1 การศึกษาระบบ Shopee [34]

Shopee เป็นแพลตฟอร์มอีคอมเมิร์ซที่มีฟีเจอร์ครบครันสำหรับการซื้อขายสินค้าออนไลน์ ฟีเจอร์สำคัญของ Shopee ได้แก่ ระบบการค้นหาสินค้าที่มีประสิทธิภาพ การจัดการตระกร้าสินค้า การชำระเงินผ่านระบบออนไลน์ การรีวิวและให้คะแนนสินค้า การจัดโปรโมชั่นและส่วนลด ระบบแชทระหว่างผู้ซื้อและผู้ขาย และการติดตามสถานะการจัดส่งสินค้า

Shopee มีการออกแบบ UX/UI ที่ใช้งานง่ายและเป็นมิตรกับผู้ใช้ ทำให้ผู้ใช้งานสามารถทำรายการซื้อขายได้สะดวกและรวดเร็ว ระบบการค้นหาสินค้าของ Shopee มีการฟิลเตอร์ข้อมูลที่หลากหลาย เช่น ราคา ประเภทสินค้า และยี่ห้อ ทำให้ผู้ใช้งานสามารถค้นหาสินค้าที่ต้องการได้อย่างง่ายดาย

2.2.2 การศึกษาระบบ Nike [35]

Nike เป็นเว็บไซต์อีคอมเมิร์ซที่เน้นการขายสินค้าเกี่ยวกับกีฬา เช่น รองเท้ากีฬา เสื้อผ้า และอุปกรณ์กีฬา ฟีเจอร์สำคัญของ Nike ได้แก่ ระบบการแสดงสินค้าที่มีภาพและรายละเอียดชัดเจน การจัดการตระกร้าสินค้า การชำระเงินผ่านระบบออนไลน์ ระบบการจัดการบัญชีผู้ใช้ การรีวิว และให้คะแนนสินค้า และการติดตามสถานะการจัดส่งสินค้า

Nike มีการออกแบบ UX/UI ที่เน้นความเรียบง่ายและสวยงาม ทำให้ผู้ใช้งานรู้สึกเพลิดเพลินในการเลือกซื้อสินค้า ระบบการแสดงสินค้าของ Nike มีการแสดงภาพสินค้าที่คมชัดและรายละเอียดสินค้าที่ครบถ้วน ทำให้ผู้ใช้งานสามารถตัดสินใจซื้อสินค้าได้ง่ายขึ้น

จากการศึกษาระบบ Shopee และ Nike ทำให้ได้รับแนวคิดและฟีเจอร์ที่สามารถนำมาประยุกต์ใช้ในการพัฒนาระบบ Leafy โดยเฉพาะการออกแบบ UX/UI ที่เน้นความเรียบง่ายและเป็นมิตรกับผู้ใช้ การค้นหาสินค้าที่มีฟิลเตอร์หลากหลาย และระบบการจัดการตะกร้าสินค้าที่มีประสิทธิภาพ นอกจากนี้ยังได้ศึกษาแนวทางในการจัดการรีวิวสินค้า ซึ่งเป็นฟีเจอร์สำคัญที่ช่วยส่งเสริมการขายและสร้างความพึงพอใจให้กับผู้ใช้งาน โดยสามารถสรุปข้อมูลโดยย่อ ดังแสดงในตารางที่ 2.1

ตารางที่ 2.1 เปรียบเทียบระบบ Shopee และ Nike กับระบบ Leafy

Feature	Shopee	Nike	Leafy
การออกแบบ UX/UI	เรียบง่าย ใช้งานง่าย	เรียบง่าย สวยงาม	เน้นความเรียบง่ายและเป็นมิตรกับผู้ใช้
การค้นหาสินค้า	มีการฟิลเตอร์ข้อมูลหลากหลาย	ค้นหาสินค้าได้สะดวก	ฟิลเตอร์ตามประเภทสินค้า ราคา และรีวิว
การจัดการตะกร้าสินค้า	มีประสิทธิภาพและใช้งานง่าย	มีประสิทธิภาพและใช้งานง่าย	สามารถจัดการตะกร้าสินค้าได้อย่างรวดเร็ว
การทำเงิน	ทำเงินผ่านระบบออนไลน์	ทำเงินผ่านระบบออนไลน์	ทำเงินผ่านระบบออนไลน์ (Mockup)
รีวิวและให้คะแนนสินค้า	มีระบบบริวิวและให้คะแนนสินค้า	มีระบบบริวิวและให้คะแนนสินค้า	มีระบบบริวิวและให้คะแนนสินค้า
การติดตามสถานะจัดส่ง	มี	มี	มี
ระบบแซท	มีระบบแซทระหว่างผู้ซื้อและผู้ขาย	ไม่มี	ไม่มี
การจัดการบัญชีผู้ใช้	มี	มี	มี

บทที่ 3 การออกแบบเชิงเทคนิค

กระบวนการและขั้นตอนในการออกแบบสำหรับ Leafy โดยมีการออกแบบขั้นตอนการพัฒนาเว็บแอปพลิเคชัน การออกแบบสถาปัตยกรรมระบบ (System Architecture) และส่วนอื่นๆ เช่น UX/UI, Frontend, Backend, Machine Learning, และ Database รวมถึงการออกแบบกรณีทดสอบ (Test Cases) เพื่อให้ระบบที่มีประสิทธิภาพและสามารถทำงานได้จริง และรวมถึงการออกแบบการเข้าถึง Feature ตามบทบาทต่างๆ ดังนี้ ผู้เยี่ยมชม (Guest) ผู้จัดหาสินค้าหรือผู้ขายสินค้า (Supplier) ผู้ใช้งาน (User) และผู้ดูแลระบบ (Admin)

3.1 Product Backlog

การออกแบบและการวางแผนของ Release และ Product Backlog Item เป็นภาพรวมของการพัฒนาเว็บแอปพลิเคชัน ดังแสดงในรูปที่ 1.3.1

Release 1 ⇒ October - November 2023	Release 3 ⇒ January - February 2024
Home Page	
Shop Page	Profile User Page
Error Handle	Secure Connection Feature
Feature: Search in the menu component	Product Detail Page
Release 2 ⇒ November 2023 - January 2024	Release 4 ⇒ February - April 2024
Sign Up Page	Account Setting Page
Sign In Page	Payment Page
Sign Out Page	Gallery Page
Refresh Token Feature	Cart Page
	Improve UI

รูปที่ 1.3.1 แบบและแผนของ Release และ Product Backlog Item ใน Release ที่ 1 – 4

รายละเอียดทั้งหมดของ Product Backlog Item ในแต่ละ Sprint ดังแสดงในตารางที่ 3.1.1 – 3.1.4

ตารางที่ 3.1.1 ข้อมูล Product Backlog Item ของ Sprint ที่ 1 และ 2 ใน Release ที่ 1

PBI Name	Description	Release	Type	Priority	Size	Sprint
VIEW-HOME-MENU	1.[FRONTEND] จะต้องแสดงเนื้อหาตามที่ออกแบบไว้แล้ว 2.[FRONTEND] สามารถกดปุ่ม Link ได้ทั้งหมด 3.[FRONTEND] หากยังไม่ได้ Sign-in กดปุ่มลงทะเบียนนำไปสู่หน้า Sign-in 4.[FRONTEND] หาก Sign-in เรียบร้อยแล้วกดปุ่มลงทะเบียนแล้วนำไปสู่หน้า Card list 5.[FRONTEND] หาก Sign-in เรียบร้อยแล้วกดปุ่มรูปภาพ Profile และไปยังหน้า Profile ของตัวเอง 6.[FRONTEND] Responsive Mobile, Tablet, Laptop	1	Must	1	3	1
VIEW-HOME-MENU-EVENT-SEARCH	1.[FRONTEND] เมื่อกดเข้าແທບค้นหาและทำการใส่ข้อความและกดค้นหาจะต้องนำไปสู่หน้า SHOP 2.[BACKEND/FRONTEND] เมื่อส่งคำค้นหาไปแล้วจะต้องกรองข้อมูลที่มีอยู่และส่งข้อมูลที่มีการคัดกรองแล้วกลับมาด้วยค่าที่เหมาะสม	1	Must	2	2	1
VIEW-HOME-INFORMATION	1.[FRONTEND] จะต้องแสดงเนื้อหาตามที่ออกแบบไว้แล้ว 2.[FRONTEND] Responsive Mobile, Tablet, Laptop	1	Must	3	1	1

ตารางที่ 3.1.1 ข้อมูล Product Backlog Item ของ Sprint ที่ 1 และ 2 ใน Release ที่ 1 (ต่อ)

PBI Name	Description	Release	Type	Priority	Size	Sprint
VIEW-HOME-SOLUTION	1.[FRONTEND] จะต้องแสดงเนื้อหาตามที่ออกแบบไว้แล้ว 2.[FRONTEND] Responsive Mobile, Tablet, Laptop	1	Must	4	1	1
VIEW-HOME-RANDOM-DESIGNER	1.[FRONTEND] จะต้องแสดงเนื้อหาตามที่ออกแบบไว้แล้ว 2.[FRONTEND] จะต้องสามารถกดปุ่มเลื่อนตัวไปหรือย้อนกลับเพื่อดูข้อมูลของ DESIGNER ได้ 3.[FRONTEND] จะต้องสามารถกดที่ Profile ของ Designer และจะนำไปยังหน้า Gallery detail ของ User คนนั้น 4.[FRONTEND] Responsive Mobile, Tablet, Laptop	1	Must	5	3	1
VIEW-HOME-RANDOM-DESIGNER-EVENT-USER-LIST	1.[BACKEND/FRONTEND] จะต้องสามารถเรียกใช้งาน API และได้ข้อมูลของ Role Designer สำหรับแสดงข้อมูลพื้นฐานไม่เกิน 10 ชุด 2.[BACKEND] ไม่จำเป็นจะต้อง Authentication เพื่อเข้าถึง	1	Must	6	1	1
VIEW-HOME-HISTORY	1.[FRONTEND] จะต้องแสดงเนื้อหาตามที่ออกแบบไว้แล้ว 2.[FRONTEND] Responsive Mobile, Tablet, Laptop	1	Must	7	1	1
VIEW-HOME-LANDSCAPE	1.[FRONTEND] จะต้องแสดงเนื้อหาตามที่ออกแบบไว้แล้ว 2.[FRONTEND] เมื่อกดปุ่มที่รูปแล้วจะต้องนำไปยังหน้า Gallery (Garden design list) และกรองข้อมูลตามที่กด 3.[FRONTEND] Responsive Mobile, Tablet, Laptop	1	Must	8	2	1

ตารางที่ 3.1.1 ข้อมูล Product Backlog Item ของ Sprint ที่ 1 และ 2 ใน Release ที่ 1 (ต่อ)

PBI Name	Description	Release	Type	Priority	Size	Sprint
VIEW-HOME-PLANT-INSPO	1.[FRONTEND] จะต้องแสดงเนื้อหาตามที่ออกแบบไว้แล้ว 2.[FRONTEND] เมื่อกดปุ่มที่รูปแล้วจะต้องนำไปยังหน้า Shop และกรองข้อมูลตามที่กด 3.[FRONTEND] Responsive Mobile, Tablet, Laptop	1	Must	9	2	1
VIEW-HOME-REVIEW	1.[FRONTEND] จะต้องแสดงเนื้อหาตามที่ออกแบบไว้แล้ว 2.[FRONTEND] จะต้องสามารถกดปุ่มเลื่อนตัวไปหน้ารีวิวต่อไปเพื่อดูข้อมูลของคนที่รีวิวล่าสุดได้ 3.[FRONTEND] Responsive Mobile, Tablet, Laptop	1	Must	10	3	1
VIEW-HOME-REVIEW-EVENT-REVIEW-LIST	1.[BACKEND] จะต้องสามารถเรียกใช้งาน API และได้ข้อมูล Review ล่าสุด ไม่เกิน 10 ชิ้น 2.[BACKEND] ไม่จำเป็นจะต้อง Authorization เพื่อเข้าถึง	1	Must	11	2	1
VIEW-HOME-FOOTER	1.[FRONTEND] จะต้องแสดงเนื้อหาตามที่ออกแบบไว้แล้ว 2.[FRONTEND] สามารถกดปุ่ม Link ได้ทั้งหมด 3.[FRONTEND] สามารถกรอก Email และกดปุ่ม Sign-up จะต้องนำไปยังหน้า Sign-up และกรอกอีเมลให้ด้วย 4.[FRONTEND] Responsive Mobile, Tablet, Laptop	1	Must	12	3	1

ตารางที่ 3.1.1 ข้อมูล Product Backlog Item ของ Sprint ที่ 1 และ 2 ใน Release ที่ 1 (ต่อ)

PBI Name	Description	Release	Type	Priority	Size	Sprint
VIEW-SHOP-LIST-PRODUCT-EVENT-PRODUCT-LIST	<p>1.[BACKEND/FRONTEND] จะต้องสามารถเรียกใช้งาน API และได้ข้อมูล Product ไม่เกิน 18 ชุด</p> <p>2.[BACKEND] ไม่จำเป็นจะต้อง Authentication ก็สามารถเข้าถึง สินค้าที่ลูกค้าสนใจได้</p>	1	Must	13	3	2
VIEW-SHOP-FILTER-ITEM	<p>1.[FRONTEND] จะต้องแสดง เนื้อหาตามที่ออกแบบไว้แล้ว</p> <p>2.[FRONTEND] สามารถกดปุ่ม หรือเลือกข้อมูลในส่วนของ Filter ได้</p> <p>3.[BACKEND/FRONTEND] สามารถเรียงลำดับข้อมูล Product list ได้</p> <p>4.[FRONTEND] Responsive Mobile, Tablet, Laptop</p>	1	Must	14	2	2
VIEW-SHOP-FILTER-ITEM-EVENT-TYPE	<p>1.[BACKEND/FRONTEND] จะต้องสามารถคัดกรองข้อมูลที่มี อยู่ให้เกี่ยวกับ Category ที่เลือก และส่งข้อมูล Product ที่คัดกรองแล้ว กลับมาด้วย</p> <p>2.[BACKEND/FRONTEND] จะต้องสามารถคัดกรองข้อมูลที่มี อยู่ให้เกี่ยวกับ Price ที่เลือกและ ส่งข้อมูล Product ที่คัดกรองแล้ว กลับมาด้วย</p> <p>3.[BACKEND/FRONTEND] จะต้องสามารถคัดกรองข้อมูลที่มี อยู่ให้เกี่ยวกับ Star point ที่เลือก และส่งข้อมูล Product ที่คัดกรอง แล้วกลับมาด้วย</p> <p>4.[BACKEND/FRONTEND] จะต้องสามารถคัดกรองข้อมูลที่มี อยู่ให้เกี่ยวกับ Popular tag ที่</p>	1	Must	15	4	2

	<p>เลือกและส่งข้อมูล Product ที่คัดกรองแล้วกลับมาด้วย</p> <p>5.[BACKEND/FRONTEND] สามารถเรียงลำดับข้อมูล Product list ตามที่ออกแบบไว้ได้</p>					
--	--	--	--	--	--	--

ตารางที่ 3.1.1 ข้อมูล Product Backlog Item ของ Sprint ที่ 1 และ 2 ใน Release ที่ 1 (ต่อ)

PBI Name	Description	Release	Type	Priority	Size	Sprint
VIEW-SHOP-FILTER-ITEM-EVENT-SORT	<p>1.[BACKEND/FRONTEND] จะต้องสามารถเรียกใช้งาน API และได้ข้อมูล Product ไม่เกิน 18 ชุด</p> <p>2.[BACKEND] ไม่จำเป็นจะต้อง Authentication สามารถเข้าถึง สินค้า</p> <p>3.[BACKEND/FRONTEND] สามารถเรียงลำดับข้อมูล Product list ตาม</p> <ul style="list-style-type: none"> - ""ราคามากสุดน้อยที่สุด"" - ""เพิ่มเข้ามาใหม่สุดเก่าที่สุด"" - ""ขายดีสุดน้อยที่สุด"" ได้ 	1	Must	16	2	2
VIEW-SHOP-LIST-PRODUCT	<p>1.[FRONTEND] จะต้องแสดง เนื้อหาตามที่ออกแบบไว้แล้ว</p> <p>2.[FRONTEND] จะต้องสามารถ กดปุ่มรูป Product และนำไปยัง หน้า Product detail ตามข้อมูล นั้นๆ</p> <p>3.[FRONTEND] Responsive Mobile, Tablet, Laptop</p>	1	Must	17	3	2
VIEW-EVENT-ERROR-HANDLE	<p>1.[FRONTEND] จัดการกับปัญหา เกี่ยวกับ PAGE 404, 500</p> <p>2.[FRONTEND] หน้าโหลด สำหรับกำลังโหลดข้อมูลอยู่</p> <p>3.[BACKEND] ถ้า Response Page ส่งกลับมาเกิด Error ความ ส่งกลับมาให้ Frontend ด้วย</p>	1	Could	18	1	2

	4.[FRONTEND] ศึกษาการทำ Lazy load เพื่อให้ลดการทำงานหนัก					
--	--	--	--	--	--	--

ตารางที่ 3.1.2 ข้อมูล Product Backlog Item ของ Sprint ที่ 1 และ 2 ใน Release ที่ 2

PBI Name	Description	Release	Type	Priority	Size	Sprint
VIEW-SIGN-IN-INPUT	1.[FRONTEND] จะต้องแสดงเนื้อหาตามที่ออกแบบไว้แล้ว 2.[FRONTEND] สามารถกรอกข้อมูล E-mail, Mobile number, Password ได้ 3.[FRONTEND] Responsive Mobile, Tablet, Laptop	2	Must	19	3	1
VIEW-SIGN-IN-INPUT-EVENT-LINK	1.[BACKEND] ส่งข้อมูลของ User คนนั้นกลับมาพร้อมเข้ารหัส โดยที่จะส่ง token ออกมา 2 ตัวคือ access token และ refresh token เก็บเป็น Http-only cookie ใน Server-side cookies และสามารถเรียกใช้งานผ่าน credential ซึ่งรวมถึง proxy container ด้วย 2.[FRONTEND] จะต้องสามารถเรียกใช้งาน API เพื่อเข้าสู่ระบบได้โดยที่เก็บข้อมูลของ user ผ่าน Client-side cookies	22	Must	20	3	1
VIEW-SIGN-IN-INPUT-EVENT-VALIDATION	1.[BACKEND/FRONTEND] เมื่อใส่ข้อมูล E-mail หรือเบอร์ ถูกต้อง จะสามารถเข้าไปใช้งานได้ให้ส่ง 200 กลับมา 2.[BACKEND/FRONTEND] เมื่อใส่ข้อมูล E-mail หรือ เบอร์ ไม่ถูกต้อง แล้วหาไม่เจอให้ส่ง 404 กลับมา	2	Must	21	2	1

	<p>3.[BACKEND/FRONTEND] เมื่อใส่ข้อมูล E-mail หรือ เบอร์โทร แล้วผิดเงื่อนไขให้ส่ง 400 กลับมา</p> <p>4.[BACKEND/FRONTEND] เมื่อใส่ข้อมูล E-mail หรือ เบอร์โทร แล้วไม่สามารถใช้งานได้ให้ส่ง 403 กลับมา</p> <p>5.[BACKEND/FRONTEND] เมื่อใส่ข้อมูลรหัสผ่าน และไม่ถูกต้องได้ให้ส่ง 401 กลับมา</p> <p>6.[BACKEND/FRONTEND] เมื่อใส่ข้อมูลรหัสผ่าน และความยาวรหัสผ่านไม่ได้อย่างในระหว่าง 8-20 ให้ส่ง 400 กลับมา</p>				
--	--	--	--	--	--

ตารางที่ 3.1.2 ข้อมูล Product Backlog Item ของ Sprint ที่ 1 และ 2 ใน Release ที่ 1 (ต่อ)

PBI Name	Description	Release	Type	Priority	Size	Sprint
VIEW-SIGN-UP-INPUT	<p>1.[FRONTEND] จะต้องแสดงเนื้อหาตามที่ออกแบบไว้แล้ว</p> <p>2.[FRONTEND] สามารถกรอกข้อมูลส่วนตัวตามที่ออกแบบการเก็บข้อมูลไว้ได้</p> <p>4.[FRONTEND] สามารถจัดการปัญหา Error ได้</p> <p>5.[FRONTEND] Responsive Mobile, Table</p>	2	Must	22	3	2
VIEW-SIGN-UP-INPUT-EVENT-REDIRECT	<p>1.[BACKEND/FRONTEND] จะต้องสามารถเรียกใช้งาน API เกี่ยวกับสร้าง User ได้</p> <p>2.[FRONTEND] เมื่อสมัครสมาชิกผ่านจะต้องแจ้งเตือนให้รู้และเมื่อสำเร็จแล้วจะต้องนำไปยังหน้า Sign-in ได้</p>	2	Must	23	1	2
VIEW-SIGN-UP-INPUT-	1.[BACKEND/FRONTEND] เมื่อใส่ข้อมูลส่วนบุคคลแล้วจะต้อง	2	Must	24	3	2

EVENT-VALIDATION	<p>สามารถสร้าง Account ของเข้าได้แล้วให้ส่ง 201 กลับมา แล้วไปยังหน้า Verify email</p> <p>2.[BACKEND/FRONTEND] เมื่อใส่ข้อมูลอะไรต่างๆ เข้าไปแล้ว ที่ไม่สามารถให้ค่าอะไรมา ถ้าหากหาเจอก็แล้วให้ส่ง 400 กลับมา</p> <p>3.[BACKEND/FRONTEND] เมื่อใส่ข้อมูลอะไรต่างๆ เข้าไปแล้ว ซึ่งจริง นามสกุล และอีเมล ที่ความยาวตัวอักษรที่เกินกว่าที่กำหนดถ้าหากหาเจอก็แล้วให้ส่ง 400 กลับมา</p> <p>4.[BACKEND/FRONTEND] เมื่อใส่ข้อมูลอะไรต่างๆ เข้าไปแล้ว รหัสผ่าน ความมีความยาวตัวอักษร 8-20 ตัวอักษร แต่ถ้าไม่ใช่ให้ส่ง 400 กลับมา</p> <p>5.[BACKEND/FRONTEND] เมื่อ Sign-up เรียบร้อยแล้ว Role ที่เพิ่งสมัครจะต้องเป็น User เท่านั้น</p> <p>6.[BACKEND/FRONTEND] เมื่อใส่ข้อมูลอะไรต่างๆ เข้าไปแล้ว และพบว่าข้อมูล ซึ่ง อีเมลและเบอร์มือถือซ้ำกับข้อมูลในระบบไปแล้วจะต้องส่ง 400 กลับมา</p> <p>7.[BACKEND/FRONTEND] เมื่อใส่ข้อมูลอะไรต่างๆ เข้าไปแล้ว อีเมลและเบอร์โทรศัพท์ของเขามิ่งเป็น Format มาตราฐานที่รับรองไว้จะส่ง 400 กลับมา</p> <p>8.[FRONTEND] เมื่อใส่ข้อมูลรหัสผ่านกับรหัสผ่านยืนยัน จะต้องตรงกันถึงจะสามารถสมัครสมาชิกได้</p>				
------------------	--	--	--	--	--

ตารางที่ 3.1.2 ข้อมูล Product Backlog Item ของ Sprint ที่ 1 และ 2 ใน Release ที่ 1 (ต่อ)

PBI Name	Description	Release	Type	Priority	Size	Sprint
VIEW-SIGN-UP-EVENT-EMAIL-SENDING	<p>1.[BACKEND] เมื่อ User Sign-up แล้วจะต้องส่งอีเมลกลับมาเพื่อทำการยืนยัน แล้วถ้าหากยืนยันที่อีเมลภายในระยะเวลาที่กำหนด ก็สามารถเข้าไปใช้งานได้เลย แต่ถ้าเข้าไม่ได้ก็ต้องยืนยันระบบจะไม่สามารถให้เข้า Login ได้ นอกจากจะต้องติดต่อไปทาง Admin</p> <p>2. เมื่อ User เข้าไปยังหน้า Verify email เขาจะต้องกดยืนยันผ่านอีเมลและเมื่อกดเสร็จจะเข้าใน Sign-in และนำ Email กับ Password ที่กูปกรปิดใส่เข้าไปได้เลย</p>	2	Could	25	6	2
EVENT-SIGN-OUT	<p>1.[FRONTEND] เมื่อ User ทำการ Log-out เข้าไปหน้า Sign-in</p> <p>2.[BACKEND] เมื่อ User ทำการ Log out Http-only cookie จะต้องโดนลบ</p>	2	Must	26	1	2
EVENT-REFRESH-TOKEN-FOR-AUTHENTICATION	<p>1.[BACKEND] เมื่อ User ใช้ Access token และหมดอายุ สามารถต่ออายุ Access token และ Refresh token เมื่อ Refresh token นั้นยังไม่หมดอายุ และเก็บ Token ใน Http-only cookie ได้</p> <p>2.[FRONTEND] เมื่อ User ใช้ Refresh token จาก Backend ที่สามารถให้ User สามารถใช้งานได้ตามปกติ</p>	2	Must	27	2	2

ตารางที่ 3.1.3 ข้อมูล Product Backlog Item ของ Sprint ที่ 1 ถึง 4 ใน Release ที่ 3

PBI Name	Description	Release	Type	Priority	Size	Sprint
VIEW-PROFILE-USER-INFORMATION-DETAIL	1.[FRONTEND] จะต้องแสดงเนื้อหาตามที่ออกแบบไว้แล้ว 2.[FRONTEND] Responsive Mobile, Table	3	Must	28	3	1
VIEW-PROFILE-RECOMMEND-PRODUCT	1.[FRONTEND] จะต้องแสดงเนื้อหาตามที่ออกแบบไว้แล้ว 2.[FRONTEND] Responsive Mobile, Table	3	Must	29	1	1
VIEW-PROFILE-PRODUCT-LIST	1.[FRONTEND] จะต้องแสดงเนื้อหาตามที่ออกแบบไว้แล้ว 2.[FRONTEND] Responsive Mobile, Table	3	Must	30	1	1
VIEW-PROFILE-PRODUCT-LIST-FILTER	1.[FRONTEND] จะต้องแสดงเนื้อหาตามที่ออกแบบไว้แล้ว 2.[FRONTEND] Responsive Mobile, Table	3	Must	31	1	1
VIEW-PROFILE-PRODUCT-OUTSTOCK	1.[FRONTEND] จะต้องแสดงเนื้อหาตามที่ออกแบบไว้แล้ว 2.[FRONTEND] Responsive Mobile, Table	3	Must	32	1	1
EVENT-PROFILE-USER-INFORMATION-DETAIL-EVENT	1.[FRONTEND] จะต้องสามารถดึงข้อมูลที่เกี่ยวข้องเช่น Rating โดยรวมของร้านค้าได้ 2.[FRONTEND] จะต้องสามารถดึงข้อมูลที่เกี่ยวข้องเช่น Products ทั้งหมดที่เป็นเจ้าของได้	3	Must	33	2	1

ตารางที่ 3.1.3 ข้อมูล Product Backlog Item ของ Sprint ที่ 1 ถึง 4 ใน Release ที่ 3 (ต่อ)

PBI Name	Description	Release	Type	Priority	Size	Sprint
EVENT-PROFILE-RECOMMEND-PRODUCT-EVENT	1.[FRONTEND/BACKEND] จะต้องสามารถแสดงข้อมูลที่สามารถแนะนำสินค้าได้ 2.[BACKEND] เมื่อ Train model แล้วจะต้องสามารถแสดงข้อมูลสินค้าแนะนำได้ 3.[FRONT/BACKEND] เมื่อกดปุ่ม จะต้องนำไปยังหน้า Product Detail ของสินค้านั้นๆ	3	Must	34	3	1
EVENT-PROFILE-PRODUCT-LIST-EVENT	1.[FRONTEND/BACKEND] จะต้องสามารถแสดงข้อมูลจากฐานข้อมูลเกี่ยวกับสินค้าของเจ้าของ Profile คนนั้นเท่านั้น 2.[FRONTEND/BACKEND] จะต้องสามารถเปลี่ยนหน้าของสินค้าได้	3	Must	35	2	1
EVENT-PROFILE-PRODUCT-LIST-FILTER-EVENT	1.[FRONTEND/BACKEND] จะต้องสามารถคัดกรองสินค้าของเจ้าของ Profile คนนั้นได้	3	Must	36	1	1
EVENT-PROFILE-PRODUCT-OUTSTOCK	1.[FRONTEND/BACKEND] จะต้องสามารถแสดงข้อมูลเกี่ยวกับสินค้าที่หมดของเจ้าของ Profile นั้นได้	3	Must	37	1	1
VIEW-PRODUCT-DETAIL-STORE	1.[FRONTEND] จะต้องแสดงเนื้อหาตามที่ออกแบบไว้แล้ว 2.[FRONTEND] จะต้องสามารถกดปุ่มรูปเพื่อดูรายละเอียดร้านค้าได้ 3.[FRONTEND] Responsive Mobile, Tablet	3	Must	38	3	2

ตารางที่ 3.1.3 ข้อมูล Product Backlog Item ของ Sprint ที่ 1 ถึง 4 ใน Release ที่ 3 (ต่อ)

PBI Name	Description	Release	Type	Priority	Size	Sprint
VIEW- PRODUCT- DETAIL- DISCRIPTION	1.[FRONTEND] จะต้องแสดงเนื้อหาตามที่ออกแบบไว้แล้ว 2.[FRONTEND] จะต้องสามารถกดปุ่มเพื่อขยายข้อมูล 3.[FRONTEND] Responsive Mobile, Tablet, Laptop	3	Must	39	2	2
VIEW- PRODUCT- DETAIL- PRODUCT- EVENT-DETAIL	1.[BACKEND/FRONTEND] สามารถดึง API ของข้อมูล Product detail ที่กดเข้ามาได้ 2.[BACKEND] ไม่จำเป็นจะต้อง Authentication เพื่อเข้าถึง	3	Must	40	1	2
VIEW- PRODUCT- DETAIL- REVIEW	1.[FRONTEND] จะต้องแสดงเนื้อหาตามที่ออกแบบไว้แล้ว 2.[FRONTEND] สามารถกดปุ่มเพื่อไปยังหน้า Review ทั้งหมด ของ Product อันนั้นได้ 3.[FRONTEND] Responsive Mobile, Tablet	3	Must	41	6	3
VIEW- PRODUCT- DETAIL- PRODUCT- EVENT- REVIEW	1.[BACKEND/FRONTEND] สามารถดึง API ของข้อมูล Review ที่กดเข้ามาได้ 2.[BACKEND] ไม่จำเป็นจะต้อง Authentication เพื่อเข้าถึง	3	Must	42	2	3
VIEW- PRODUCT- DETAIL- PRODUCT	1.[FRONTEND] จะต้องแสดงเนื้อหาตามที่ออกแบบไว้แล้ว 2.[FRONTEND] จะต้องสามารถกดปุ่มรูปเพื่อดูรายละเอียดรูปภาพได้ 3.[FRONTEND] Responsive Mobile, Tablet	3	Must	43	6	3

ตารางที่ 3.1.3 ข้อมูล Product Backlog Item ของ Sprint ที่ 1 ถึง 4 ใน Release ที่ 3 (ต่อ)

PBI Name	Description	Release	Type	Priority	Size	Sprint
VIEW- PRODUCT- DETAIL- PRODUCT- EVENT- IMAGES	1.[BACKEND/FRONTEND] สามารถดึง API ของข้อมูล รูปภาพของ Product อันนั้นได้ ทั้งหมด 2.[BACKEND] ไม่จำเป็นจะต้อง Authentication เพื่อเข้าถึง	3	Must	44	4	3
VIEW- PRODUCT- DETAIL- RECOMMEND- PRODUCT	1.[FRONTEND] ศึกษาการทำ Lazy load เพื่อให้ลดการทำงานหนัก 2.[FRONTEND] จะต้องแสดงเนื้อหาตามที่ออกแบบไว้แล้ว 3.[FRONTEND] สามารถกดปุ่มของ Product เพื่อไปยัง Product detail อันนั้นได้ 4.[FRONTEND] สามารถเลื่อนดู Product หน้าก้าดไปหรือย้อนกลับหน้าได้ 5.[FRONTEND] Responsive Mobile, Tablet, Laptop	3	Must	45	1	3
VIEW- PRODUCT- DETAIL- PRODUCT- EVENT- RECOMMED	1.[BACKEND/FRONTEND] สามารถดึง API ของข้อมูล Product (Recommend) ที่กดเข้ามาดูได้ 2.[BACKEND] ไม่จำเป็นจะต้อง Authentication เพื่อเข้าถึง	3	Must	46	2	3
SECURE- CONNECTION	1.[INFRA] สามารถทำให้ Public URL secure ตอนเวลาเขื่อมต่อ 2.[FRONTEND/BACKEND] เมื่อเว็บไซต์ขึ้นบน Public URL แล้ว จะต้องสามารถเขื่อมต่อแบบ HTTPS เท่านั้น 3.[FRONTEND/BACKEND] เมื่อเข้าสู่เว็บไซต์บน Public URL	3	Must	47	2	4

	แล้วเมื่อมีการ Redirect ที่ Port ไหนแล้วจะทำให้ไปที่ Port 443 เท่านั้น					
--	--	--	--	--	--	--

ตารางที่ 3.1.3 ข้อมูล Product Backlog Item ของ Sprint ที่ 1 ถึง 4 ใน Release ที่ 3 (ต่อ)

PBI Name	Description	Release	Type	Priority	Size	Sprint
USER-ACCOUNT-ENCRYPTION	1.[BACKEND/FRONTEND] เมื่อ User ได้ทำการ Encrypt แล้วเข้า จะไม่สามารถเห็นข้อมูลส่วนตัว เป็นรูปแบบปกติแต่จะเป็น Cipher text แทน 2.[BACKEND/FRONTEND] เมื่อ User ได้ Redirect ไปยังหน้า Profile จะเห็น URL path เป็น Cipher text 3.[BACKEND/FRONTEND] เมื่อ User เข้าใช้งานเข้าสามารถ Decrypt cipher text และ สามารถเห็นข้อมูลส่วนตัวของเข้าได้	3	Could	48	3	4

ตารางที่ 3.1.4 ข้อมูล Product Backlog Item ของ Sprint ที่ 1 ถึง 4 ใน Release ที่ 4

PBI Name	Description	Release	Type	Priority	Size	Sprint
VIEW-ACCOUNT-SETTING	1.[FRONTEND] จะต้องแสดงเนื้อหาตามที่ออกแบบไว้แล้ว 2.[FRONTEND] จะต้องสามารถกดปุ่มเพื่อไปยังหน้าต่างๆ ได้	4	Must	49	4	1
VIEW-ACCOUNT-SETTING-EVENT-LINK	1.[FRONTEND] เมื่อกดปุ่ม Link จะต้องนำไปยังหน้าเว็บไซต์ได้จริงๆ 2.[FRONTEND] จะต้องเข้าสู่ระบบก่อนเท่านั้นถึงจะมาหน้านี้ได้	4	Must	50	2	1

ตารางที่ 3.1.4 ข้อมูล Product Backlog Item ของ Sprint ที่ 1 ถึง 4 ใน Release ที่ 4 (ต่อ)

PBI Name	Description	Release	Type	Priority	Size	Sprint
VIEW-ACCOUNT-SETTING-PROFILE	1.[FRONTEND] จะต้องแสดงเนื้อหาตามที่ออกแบบไว้แล้ว 2.[FRONTEND] จะต้องมีที่สำหรับกรอกข้อมูลต่างๆ	4	Must	51	6	1
VIEW-ACCOUNT-SETTING-PROFILE-EVENT	1.[FRONTEND] จะต้องสามารถใช้ API สำหรับการ Update ข้อมูลได้ เพื่อเอาข้อมูลที่ได้มาลงฐานข้อมูลที่ตรงกับ ID ที่เข้าสู่ระบบได้ 2.[FRONTEND/BACKEND] จะต้องสามารถใช้ API สำหรับการ Upload รูปภาพของ User ตาม ID ที่เข้าสู่ระบบไว้ได้ 3.[FRONTEND] สามารถตรวจสอบ และคัดกรองข้อมูลก่อนส่งไปยัง BACKEND ได้	4	Must	52	3	1
VIEW-ACCOUNT-SETTING-ADDRESS-LIST	1.[FRONTEND] จะต้องแสดงเนื้อหาตามที่ออกแบบไว้แล้ว 2.[FRONTEND] จะต้องสามารถกดปุ่มเพิ่มที่อยู่ใหม่ได้ 3.[FRONTEND] จะต้องสามารถกดปุ่มเพื่อลดหรือแก้ไขได้	4	Must	53	4	1
VIEW-ACCOUNT-SETTING-ADDRESS-LIST-EVENT	1.[FRONTEND] เมื่อกดปุ่มเพิ่มที่อยู่จะต้องนำไปยังหน้าสำหรับการเพิ่มที่อยู่ได้ 2.[FRONTEND/BACKEND] เมื่อกดปุ่มลบข้อมูลจะต้องสามารถลบข้อมูลออกໄປได้จริงด้วย API ของ Backend 3.[FRONTEND/BACKEND] เมื่อกดปุ่มแก้ไขจะต้องสามารถนำไปยังหน้าสำหรับแก้ไขข้อมูลได้จากข้อมูลเดิมได้	4	Must	54	3	1

ตารางที่ 3.1.4 ข้อมูล Product Backlog Item ของ Sprint ที่ 1 ถึง 4 ใน Release ที่ 4 (ต่อ)

PBI Name	Description	Release	Type	Priority	Size	Sprint
VIEW-ACCOUNT-SETTING-ADDRESS-CREATE	1.[FRONTEND] จะต้องแสดงเนื้อหาตามที่ออกแบบไว้แล้ว 2.[FRONTEND] จะต้องสามารถใส่ข้อมูลลงในช่องกรอกข้อมูลได้	4	Must	55	4	1
VIEW-ACCOUNT-SETTING-ADDRESS-CREATE-EVENT-CREATE	1.[FRONTEND/BACKEND] จะต้องสามารถเพิ่มข้อมูลใหม่ได้จาก API 2.[FRONTEND] สามารถตรวจสอบและคัดกรองข้อมูลก่อนส่งไปยัง BACKEND ได้	4	Must	56	4	1
VIEW-ACCOUNT-SETTING-ADDRESS-CREATE-EVENT-UPDATE	1.[FRONTEND/BACKEND] จะต้องสามารถดึงข้อมูลมาจากการ API ได้เพื่อนำมาแก้ไขจากข้อมูลเดิม 2.[FRONTEND] สามารถตรวจสอบและคัดกรองข้อมูลก่อนส่งไปยัง BACKEND ได้	4	Must	57	41	
VIEW-ACCOUNT-SETTING-PASSWORD-CHANGE	1.[FRONTEND] จะต้องแสดงเนื้อหาตามที่ออกแบบไว้แล้ว 2.[FRONTEND] จะต้องสามารถใส่ข้อมูลลงในช่องกรอกข้อมูลได้ 3.[FRONTEND] จะต้องแจ้งเตือนเมื่อกด Submit	4	Must	58	3	1
VIEW-ACCOUNT-SETTING-PASSWORD-CHANGE-EVENT	1.[FRONTEND/BACKEND] จะต้องสามารถตรวจสอบรหัสผ่านก่อนไปยังหน้าสำหรับ Change password ได้ 2.[FRONTEND/BACKEND] จะต้องสามารถส่ง Password ใหม่ที่เปลี่ยนแล้วผ่าน API ได้	4	Must	59	2	1

	3.[FRONTEND] สามารถตรวจสอบ และคัดกรองข้อมูลก่อนส่งไปยัง BACKEND ได้					
--	---	--	--	--	--	--

ตารางที่ 3.1.4 ข้อมูล Product Backlog Item ของ Sprint ที่ 1 ถึง 4 ใน Release ที่ 4 (ต่อ)

PBI Name	Description	Release	Type	Priority	Size	Sprint
VIEW-ACCOUNT-SETTING-BANK-LIST	1.[FRONTEND] จะต้องแสดงเนื้อหาตามที่ออกแบบไว้แล้ว 2.[FRONTEND] จะต้องสามารถกดปุ่มเพิ่มธนาคารได้ 3.[FRONTEND] จะต้องสามารถกดปุ่มเพื่อลดหรือแก้ไขได้	4	Must	60	4	1
VIEW-ACCOUNT-SETTING-BANK-LIST-EVENT	1.[FRONTEND] เมื่อกดปุ่มเพิ่มที่อยู่จะต้องนำไปยังหน้าสำหรับการเพิ่มธนาคารได้ 2.[FRONTEND/BACKEND] เมื่อกดปุ่มลบข้อมูลจะต้องสามารถลบข้อมูลออกໄປได้จริงด้วย API ของ Backend 3.[FRONTEND/BACKEND] เมื่อกดปุ่มแก้ไขจะต้องสามารถนำไปยังหน้าสำหรับแก้ไขข้อมูลได้จากข้อมูลเดิมได้	4	Must	61	3	1
VIEW-ACCOUNT-SETTING-BANK-CREATE	1.[FRONTEND] จะต้องแสดงเนื้อหาตามที่ออกแบบไว้แล้ว 2.[FRONTEND] จะต้องสามารถใส่ข้อมูลลงในช่องกรอกข้อมูลได้	4	Must	62	4	1
VIEW-ACCOUNT-SETTING-BANK-CREATE-EVENT-CREATE	1.[FRONTEND/BACKEND] จะต้องสามารถเพิ่มข้อมูลใหม่ได้จาก API 2.[FRONTEND] สามารถตรวจสอบ และคัดกรองข้อมูลก่อนส่งไปยัง BACKEND ได้	4	Must	63	4	1

ตารางที่ 3.1.4 ข้อมูล Product Backlog Item ของ Sprint ที่ 1 ถึง 4 ใน Release ที่ 4 (ต่อ)

PBI Name	Description	Release	Type	Priority	Size	Sprint
VIEW-ACCOUNT-SETTING-BANK-CREATE-EVENT-UPDATE	1.[FRONTEND/BACKEND] จะต้องสามารถดึงข้อมูลมากจาก API ได้เพื่อนำมาแก้ไขจากข้อมูลเดิม 2.[FRONTEND] สามารถตรวจสอบและคัดกรองข้อมูลก่อนส่งไปยัง BACKEND ได้	4	Must	64	4	1
VIEW-ACCOUNT-SETTING-MYSHOP-PRODUCT-LIST	1.[FRONTEND] จะต้องแสดงเนื้อหาตามที่ออกแบบไว้แล้ว 2.[FRONTEND] จะต้องสามารถกดปุ่มเพื่อเพิ่มสินค้าได้ 3.[FRONTEND] จะต้องสามารถกดปุ่ม เพื่อจัดการเลียงลำดับสินค้าใหม่ได้ 4.[FRONTEND] จะต้องสามารถกดปุ่มเพื่อลบสินค้าได้ 5.[FRONTEND] จะต้องสามารถกดปุ่มเพื่อแก้ไขสินค้าได้	4	Must	65	2	2
VIEW-ACCOUNT-SETTING-MYSHOP-PRODUCT-LIST-EVENT-FETCH	1.[FRONTEND/BACKEND] จะต้องสามารถดึงข้อมูลเกี่ยวกับสินค้าที่เป็นเจ้าของเดียวกับที่เข้าสู่ระบบได้ 2.[FRONTEND/BACKEND] จะต้องสามารถลบข้อมูลจากฐานข้อมูลออกได้จริงจาก API	4	Must	66	3	2
VIEW-ACCOUNT-SETTING-MYSHOP-PRODUCT-LIST-EVENT-FILTER	1.[FRONTEND/BACKEND] จะต้องสามารถคัดกรองตามรายการต่างๆ ที่ออกแบบไว้ได้	4	Must	67	2	2

ตารางที่ 3.1.4 ข้อมูล Product Backlog Item ของ Sprint ที่ 1 ถึง 4 ใน Release ที่ 4 (ต่อ)

PBI Name	Description	Release	Type	Priority	Size	Sprint
VIEW-ACCOUNT-SETTING-MYSHOP-PRODUCT-CREATE	1.[FRONTEND] จะต้องแสดงเนื้อหาตามที่ออกแบบไว้แล้ว 2.[FRONTEND] จะต้องสามารถกรอกข้อมูลในช่องต่างๆ ได้จริง	4	Must	68	6	2
VIEW-ACCOUNT-SETTING-MYSHOP-PRODUCT-LIST-EVENT-CREATE	1.[FRONTEND/BACKEND] จะต้องสามารถเพิ่มข้อมูลสินค้าอันใหม่ได้ 2.[FRONTEND/BACKEND] จะต้องสามารถ Upload รูปได้ 3.[FRONTEND] สามารถตรวจสอบ และคัดกรองข้อมูลก่อนส่งไปยัง BACKEND ได้	4	Must	69	6	2
VIEW-ACCOUNT-SETTING-MYSHOP-PRODUCT-LIST-EVENT-UPDATE	1.[FRONTEND/BACKEND] จะต้องสามารถดึงข้อมูลเพื่อนำมาแก้ไขจากข้อมูลเดิมได้ 2.[FRONTEND/BACKEND] จะต้องสามารถแก้ไขรูปภาพได้ 3.[FRONTEND] สามารถตรวจสอบ และคัดกรองข้อมูลก่อนส่งไปยัง BACKEND ได้	4	Must	70	6	2
VIEW-ACCOUNT-SETTING-MYSHOP-ORDER-LIST	1.[FRONTEND] จะต้องแสดงเนื้อหาตามที่ออกแบบไว้แล้ว 2.[FRONTEND] จะต้องสามารถกรอกข้อมูลในช่องต่างๆ ได้จริง	4	Must	71	6	2
VIEW-ACCOUNT-SETTING-MYSHOP-ORDER-LIST-EVENT	1.[FRONTEND/BACKEND] จะต้องสามารถดึงข้อมูลเกี่ยวกับสินค้าที่ถูกสั่งจาก User คนอื่นได้ 2.[FRONTEND/BACKEND] จะต้องสามารถคัดกรองตามรายการต่างๆ ที่ออกแบบไว้ได้	4	Must	72	6	2

	<p>3.[FRONTEND/BACKEND] จะต้องสามารถลบข้อมูลจาก ฐานข้อมูลออกได้จริงจาก API</p> <p>4.[FRONTEND/BACKEND] สามารถกดดูรายละเอียดเพิ่มเติม สำหรับสินค้าได้</p>					
--	--	--	--	--	--	--

ตารางที่ 3.1.4 ข้อมูล Product Backlog Item ของ Sprint ที่ 1 ถึง 4 ใน Release ที่ 4 (ต่อ)

PBI Name	Description	Release	Type	Priority	Size	Sprint
VIEW- ACCOUNT- SETTING- PURCHASE	<p>1.[FRONTEND] จะต้องแสดง ข้อมูลตามที่ออกแบบไว้ได้</p>	4	Must	73	6	3
VIEW- ACCOUNT- SETTING- PURCHASE- EVENT	<p>1.[FRONTEND/BACKEND] จะต้องสามารถแสดง List ข้อมูล สินค้าที่สั่งซื้อได้</p> <p>2.[FRONTEND/BACKEND] จะต้องสามารถกดปุ่มเพื่อไปยัง หน้าร้านค้าได้</p> <p>3.[FRONTEND/BACKEND] จะต้องสามารถกดปุ่มเพื่อไปยัง หน้า Confirm payment ได้หาก สินค้ายังไม่ได้จ่าย</p> <p>4.[FRONTEND/BACKEND] จะต้องสามารถกดปุ่มเพื่อยกเลิก รายการสินค้าได้หากสินค้ายังไม่ได้ จ่าย</p> <p>5.[FRONTEND/BACKEND] จะต้องสามารถกดปุ่มเพื่อรับ รายการสินค้าได้</p> <p>6.[FRONTEND/BACKEND] จะต้องสามารถกดปุ่มไปยังหน้า สำหรับแสดงรายละเอียดข้อมูล ของรายการสินค้าได้</p>	4	Must	74	6	3

ตารางที่ 3.1.4 ข้อมูล Product Backlog Item ของ Sprint ที่ 1 ถึง 4 ใน Release ที่ 4 (ต่อ)

PBI Name	Description	Release	Type	Priority	Size	Sprint
VIEW-ACCOUNT-SETTING-PURCHASE-EVENT-FILETER	1.[FRONTEND/BACKEND] จะต้องสามารถดึงข้อมูลที่กรองตามที่ออกแบบไว้ได้	4	Must	75	3	3
VIEW-ACCOUNT-SETTING-PURCHASE-DETAIL	1.[FRONTEND] จะต้องแสดงข้อมูลตามที่ออกแบบไว้ได้	4	Must	76	3	3
VIEW-ACCOUNT-SETTING-PURCHASE-DETAIL-EVENT	1.[FRONTEND/BACKEND] จะต้องดึงข้อมูลเกี่ยวกับรายละเอียดของรายละเอียดสินค้าได้ 2.[FRONTEND/BACKEND] จะต้องสามารถกดปุ่มเพื่อไปยังหน้าร้านค้าได้ 3.[FRONTEND/BACKEND] จะต้องสามารถกดปุ่มเพื่อรับรายการสินค้าได้	4	Must	77	3	3
VIEW-ACCOUNT-SETTING-PURCHASE-REVIEW	1.[FRONTEND] จะต้องแสดงข้อมูลตามที่ออกแบบไว้ได้	4	Must	78	3	3
VIEW-ACCOUNT-SETTING-PURCHASE-REVIEW-EVENT-ADD	1.[FRONTEND/BACKEND] จะต้องสามารถสร้าง Review ของสินค้าทุกชิ้นหรือบางชิ้นในรายการสินค้าที่รับแล้วได้ 2.[FRONTEND/BACKEND] จะต้องมีการตรวจสอบความ	4	Must	79	6	3

	ถูกต้องของข้อมูลที่รับมาก่อนส่ง Backend 3.[FRONTEND/BACKEND] จะต้องบังคับให้กรอกข้อมูลทุกช่องเท่านั้นถึงจะส่ง Form ได้					
--	--	--	--	--	--	--

ตารางที่ 3.1.4 ข้อมูล Product Backlog Item ของ Sprint ที่ 1 ถึง 4 ใน Release ที่ 4 (ต่อ)

PBI Name	Description	Release	Type	Priority	Size	Sprint
VIEW-ACCOUNT-SETTING-PURCHASE-REVIEW-EVENT-EDIT	1.[FRONTEND/BACKEND] หากมีการรีวิวจะต้องดึงข้อมูลเกี่ยวกับรายละเอียดของการรีวิวได้ 2.[FRONTEND/BACKEND] จะต้องมีการตรวจสอบความถูกต้องของข้อมูลที่รับมาก่อนส่งไป Backend 3.[FRONTEND/BACKEND] จะต้องบังคับให้กรอกข้อมูลทุกช่องเท่านั้นถึงจะส่ง Form ได้ 4.[FRONTEND/BACKEND] จะต้องสามารถแก้ไขข้อมูล Review ได้	4	Must	80	6	3
VIEW-CART	1.[FRONTEND] จะต้องแสดงข้อมูลตามที่ออกแบบไว้ได้	4	Must	81	2	3
VIEW-CART-EVENT	1.[FRONTEND/BACKEND] จะต้องสามารถแสดงข้อมูลสินค้าในตะกร้าของผู้ใช้งานได้ 2.[FRONTEND] จะต้องสามารถเลือกสินค้าเพื่อพาไปยังหน้า Payment ได้	4	Must	82	3	3
VIEW-PRODUCT-DETAIL-PRODUCT-EVENT-ADD	1.[FRONTEND/BACKEND] จะต้องสามารถเพิ่มสินค้าลงตะกร้าได้ 2.[FRONTEND] จะต้องสามารถกดซื้อสินค้าทันทีแล้วพาไปยังหน้า Place order ได้	4	Must	83	3	3

	3.[FRONTEND] จะต้องสามารถตรวจสอบจำนวนสินค้าก่อนมีการสั่งซื้อและแจ้งให้ผู้ใช้งานรู้ได้					
--	---	--	--	--	--	--

ตารางที่ 3.1.4 ข้อมูล Product Backlog Item ของ Sprint ที่ 1 ถึง 4 ใน Release ที่ 4 (ต่อ)

PBI Name	Description	Release	Type	Priority	Size	Sprint
VIEW-PAYMENT	1.[FRONTEND] จะต้องแสดงข้อมูลตามที่ออกแบบไว้ได้	4	Must	84	3	3
VIEW-PAYMENT-EVENT	1.[FRONTEND/BACKEND] จะต้องสามารถยืนยันรายการสินค้าได้ 2.[FRONTEND] จะต้องสามารถนำข้อมูลสินค้าที่เลือกมาแสดงให้กับผู้ใช้งานรู้ได้ 3.[FRONTEND/BACKEND] หากกดยืนยันรายการสินค้าแล้ว ไม่มีที่อยู่จัดส่งจะต้องแจ้งให้ผู้ใช้งานรู้ได้ 4.[FRONTEND/BACKEND] เมื่อยืนยันแล้วจะต้องพาไปยังหน้า Confirm payment ได้	4	Must	85	3	3
VIEW-PAYMENT-ADDRESS	1.[FRONTEND] จะต้องแสดงข้อมูลตามที่ออกแบบไว้ได้	4	Must	86	2	3
VIEW-PAYMENT-ADDRESS-EVENT	1.[FRONTEND/BACKEND] จะต้องสามารถดึงข้อมูลเกี่ยวกับที่อยู่จัดส่งของผู้ใช้ได้ 2.[FRONTEND/BACKEND] จะต้องสามารถเพิ่มที่อยู่ใหม่ได้ 3.[FRONTEND] จะต้องสามารถเลือกที่อยู่ที่ต้องการจัดส่งได้	4	Must	87	6	3
VIEW-PAYMENT-CONFIRM	1.[FRONTEND] จะต้องแสดงข้อมูลตามที่ออกแบบไว้ได้	4	Must	88	1	3

ตารางที่ 3.1.4 ข้อมูล Product Backlog Item ของ Sprint ที่ 1 ถึง 4 ใน Release ที่ 4 (ต่อ)

PBI Name	Description	Release	Type	Priority	Size	Sprint
VIEW-PAYMENT-CONFIRM-EVENT	1.[FRONTEND/BACKEND] จะต้องสามารถกดปุ่มเพื่อจำลองการจ่ายเงินได้ 2.[FRONTEND] สามารถกดย้อนกลับได้เมื่อยังไม่ต้องการจ่ายเงิน 3.[FRONTEND/BACKEND] จะต้องสามารถดึงข้อมูลของสินค้าที่ต้องการจ่ายได้	4	Must	89	1	3
VIEW-GALLERY	1.[FRONTEND] จะต้องแสดงข้อมูลที่ออกแบบไว้ได้	4	Must	90	3	4
VIEW-GALLERY-EVENT	1.[FRONTEND/BACKEND] จะต้องสามารถแสดง List ข้อมูลเนื้อหาได้ 2.[FRONTEND/BACKEND] สามารถคัดกรองข้อมูลตามที่ออกแบบไว้ได้ 3.[FRONTEND/BACKEND] สามารถกดเพื่อดูรายละเอียดของเนื้อหาได้	4	Must	91	1	4
VIEW-GALLERY-DETAIL	1.[FRONTEND] จะต้องแสดงข้อมูลตามที่ออกแบบไว้ได้	4	Must	92	3	4
VIEW-GALLERY-DETAIL-EVENT	1.[FRONTEND/BACKEND] จะต้องสามารถดึงข้อมูลรายละเอียดของเนื้อหาได้ 2.[FRONTEND/BACKEND] จะต้องสามารถกดเพื่อดูໂປຣໄຟລ์ของผู้เขียนได้	4	Must	93	1	4
VIEW-GALLERY-PROFILE	1.[FRONTEND] จะต้องแสดงข้อมูลตามที่ออกแบบไว้ได้	4	Must	94	3	4

ตารางที่ 3.1.4 ข้อมูล Product Backlog Item ของ Sprint ที่ 1 ถึง 4 ใน Release ที่ 4 (ต่อ)

PBI Name	Description	Release	Type	Priority	Size	Sprint
VIEW-GALLERY-PROFILE-EVENT	<p>1.[FRONTEND/BACKEND] จะต้องสามารถดึง List ข้อมูลเนื้อหาของเจ้าของ Profile ได้</p> <p>2.[FRONTEND/BACKEND] จะต้องสามารถกดเพื่อดูรายละเอียดของเนื้อหาได้</p> <p>3.[FRONTEND/BACKEND] สามารถคัดกรองข้อมูลตามที่ออกแบบไว้ได้</p> <p>4.[FRONTEND/BACKEND] หากผู้เขียนเป็น Role supplier จะต้องสามารถกดปุ่มเพื่อไปยังหน้า Profile ที่มีสินค้าแสดงได้</p>	4	Must	95	2	4
VIEW-ACCOUNT-SETTING-MYGALLERY	1.[FRONTEND] จะต้องแสดงข้อมูลตามที่ออกแบบไว้ได้	4	Must	96	6	4
VIEW-ACCOUNT-SETTING-MYGALLERY-EVENT-LIST	<p>1.[FRONTEND/BACKEND] จะต้องสามารถดึงข้อมูล List เนื้อหาของเจ้าของ Profile ได้</p> <p>2.[FRONTEND/BACKEND] จะต้องสามารถลบข้อมูลเกี่ยวกับเนื้อหาของตัวเองได้</p> <p>3.[FRONTEND/BACKEND] จะต้องสามารถกดปุ่มเพื่อไปยังหน้าสำหรับแก้ไขได้</p> <p>4.[FRONTEND/BACKEND] จะต้องสามารถกดปุ่มเพื่อพาไปยังหน้าเพิ่มเนื้อหาได้</p>	4	Must	97	2	4
VIEW-ACCOUNT-SETTING-	1.[FRONTEND/BACKEND] จะต้องสามารถตรวจสอบความถูกต้องของข้อมูลที่กรอกมาได้	4	Must	98	6	4

MYGALLERY-EVENT-ADD	2.[FRONTEND/BACKEND] จะต้องสามารถเพิ่มรูปได้ 3.[FRONTEND/BACKEND] จะต้องสามารถกดส่งเนื้อหาที่สร้างได้				
---------------------	--	--	--	--	--

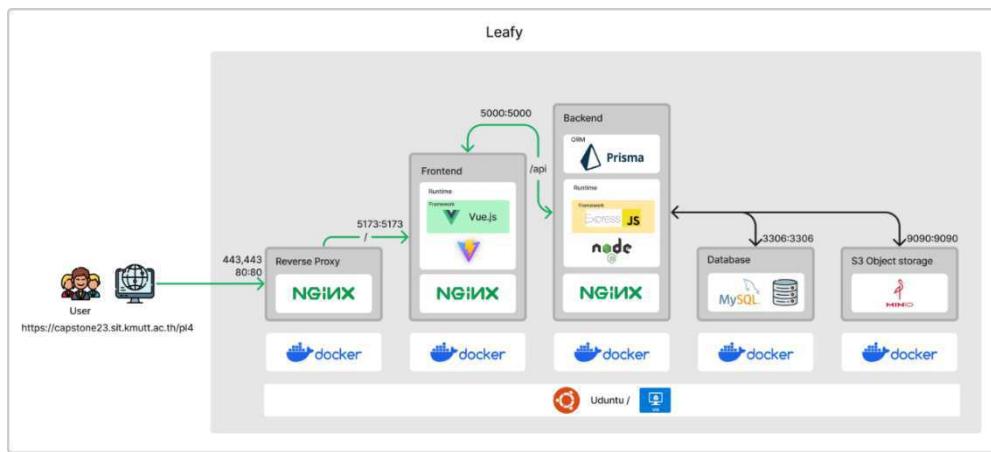
ตารางที่ 3.1.4 ข้อมูล Product Backlog Item ของ Sprint ที่ 1 ถึง 4 ใน Release ที่ 4 (ต่อ)

PBI Name	Description	Release	Type	Priority	Size	Sprint
VIEW-ACCOUNT-SETTING-MYGALLERY-EVENT-EDIT	1.[FRONTEND/BACKEND] หากมีเนื้อหาที่ได้สร้างไว้แล้วจะต้องแสดงข้อมูลเนื้อหาเก่าได้ 2.[FRONTEND/BACKEND] จะต้องสามารถตรวจสอบความถูกต้องของข้อมูลที่กรอกมาได้ 3.[FRONTEND/BACKEND] จะต้องสามารถเพิ่มรูปได้ 4.[FRONTEND/BACKEND] จะต้องสามารถกดส่งเนื้อหาที่แก้ไขได้	4	Must	99	6	4
VIEW-IMPROVE-UI	1.[FRONTEND] ปรับปรุงการแสดงผล 2.[FRONTEND] รองรับ Responsive ตามที่ได้ออกแบบไว้ได้ 3.[FRONTEND] เพิ่ม Loading animation 4.[FRONTEND] หากไม่สามารถส่งข้อมูลหรือดึงข้อมูลได้จาก Backend จะต้องแจ้งเตือนให้ผู้ใช้รับรู้ 5.[FRONTEND] เพิ่ม Alert เพื่อแจ้งเตือนผู้ใช้งานในกรณีต่างๆ ที่ความเมื่อนาน แจ้งเตือนรูปจะต้องมีขนาดไม่เกิน 2 MB เท่านั้นเป็นต้น	4	Must	100	6	4

	6.[FRONTEND] หากไม่มีข้อมูล จะต้องแสดงข้อความเพื่อให้ ผู้ใช้งานได้รับรู้				
--	--	--	--	--	--

3.2 System Architecture

สถาปัตยกรรมของระบบและโครงสร้างพื้นฐาน (Infrastructure) ของ Leafy โดยจะแสดงถึงการทำงานร่วมกันในแต่ละโดเมน (Frontend -> Backend -> Database -> Infrastructure) ในระดับ Container/Port เพื่อให้เห็นภาพรวมของการทำงาน การออกแบบสถาปัตยกรรม มีจุดประสงค์เพื่อให้ระบบมีความยืดหยุ่น มีประสิทธิภาพ และสามารถขยายขนาดได้ง่าย แผนภาพแสดงโครงสร้างการทำงานของ Leafy ดังแสดงในรูปที่ 3.2.1



รูปที่ 3.2.1 แผนภาพโครงสร้างเว็บแอปพลิเคชัน

เว็บแอปพลิเคชันที่พัฒนาขึ้นมีโครงสร้างที่แบ่งออกเป็น 5 ส่วนหลักๆ ได้แก่ Frontend, Backend, Machine Learning, Database, File Storage และ Infrastructure และส่วนจะทำงานร่วมกันผ่านการเชื่อมต่อของ API และบริการต่างๆ เพื่อให้ระบบทำงานได้อย่างมีประสิทธิภาพ โดยสามารถอธิบายรายละเอียดของแต่ละส่วนได้ ดังนี้

3.2.1 Frontend

- Technology: VUE JS Framework
- Functionality: ใช้ในการแสดงผลข้อมูลเชิงวัตถุของเว็บแอปพลิเคชัน ให้ผู้ใช้งานสามารถโต้ตอบกับระบบได้
- Container/Port: Frontend Container, Port 8080

3.2.2 Backend & Machine learning

- Technology: Node JS Express
- Functionality:

- Backend: จัดการในส่วนของ Service ต่างๆ เพื่อการทำงานที่เป็นส่วนชัดเจน รวมถึงการจัดการ API ต่างๆ สำหรับการสื่อสารระหว่าง Frontend และ Database
- Machine Learning: ใช้ JavaScript สำหรับการฝึก (Train) โมเดลโดยใช้ Dataset และ Weight Scoring เพื่อการคาดการณ์ (Predict) พฤติกรรมของผู้ใช้ โดยจะจัดกลุ่มและเรียงข้อมูลตาม Percent ที่ได้สูงสุด [13]
- Container/Port: Backend Container, Port 5000

3.2.3 Database

- Technology: MySQL
- Functionality: ใช้เป็นฐานข้อมูลหลักสำหรับการเก็บข้อมูลทุกประเภทที่จำเป็นในระบบ รวมถึง Log ต่างๆ
- Container/Port: Database Container, Port 3306

3.2.4 File Storage

- Technology: Minio
- Functionality: เป็น Object Storage ที่สามารถเก็บไฟล์รูปภาพได้เทียบเท่ากับ Firebase Storage โดยเก็บข้อมูลในรูปแบบ Buckets ซึ่งสามารถดาวน์โหลดจาก Service Containers ได้
- Container/Port: Minio Container, Port 9090

3.2.5 Infrastructure

- Technology: Docker, Nginx
- Functionality:
 - Docker: ใช้เป็น Platform ในการสร้างและจัดการ Containers ของ Service ต่างๆ เพื่อให้ระบบมีความยืดหยุ่นและขยายขนาดได้ง่าย
 - Nginx: เป็นตัวกลางในการสื่อสารระหว่างระบบซึ่งใช้ Reverse Proxy เพื่อจัดการการรับส่งข้อมูลให้แก่หน้าเว็บ
- Container/Port: Nginx Container, Port 80, 443 และเชื่อมต่อผ่าน Frontend และ Backend ผ่าน port 443 ด้วยการเชื่อมต่อที่ปลอดภัยเท่านั้น

ในแผนภาพโครงสร้างของเว็บแอปพลิเคชัน (รูปที่ 3.2.1) จะเห็นได้ว่าแต่ละส่วนมีการเชื่อมต่อกันผ่าน Container และ Port ที่กำหนดไว้ชัดเจน ทำให้การสื่อสารระหว่างส่วนต่างๆ ของระบบมีความราบรื่นและมีประสิทธิภาพ ด้วยการออกแบบสถาปัตยกรรมในลักษณะนี้ จะช่วยให้

ระบบสามารถรองรับการขยายตัวและปรับปรุงได้จ่ายในอนาคต รวมถึงสามารถตรวจสอบและแก้ไขปัญหาได้อย่างรวดเร็ว ทำให้เว็บแอปพลิเคชันมีความเสถียรและพร้อมใช้งาน

3.3 การออกแบบซอฟต์แวร์ (Software Design)

การออกแบบเว็บแอปพลิเคชัน Leafy ประกอบด้วย 5 ส่วนหลัก ได้แก่ UX/UI, Frontend, Backend, Machine Learning และ Database โดยแต่ละส่วนจะมีการออกแบบและพัฒนาให้ทำงานร่วมกันและมีการเชื่อมต่อที่ดี เพื่อให้ผู้ใช้งานสามารถใช้งานระบบได้สะดวกและมีประสิทธิภาพ

3.3.1 UX/UI

การออกแบบ User Experience (UX) และ User Interface (UI) ของ Leafy เน้นไปที่การสร้างประสบการณ์ที่ดี โดยคำนึงถึงความง่ายในการใช้งานและความสะดวกสบาย การออกแบบในส่วนนี้จึงมีแนวคิดและขั้นตอนที่สำคัญ ดังนี้

แนวคิดการออกแบบ UX/UI

1. User-Centered Design: การออกแบบโดยคำนึงถึงความต้องการและพฤติกรรมของผู้ใช้งานเป็นหลัก โดยทำการศึกษาผู้ใช้งานกลุ่มเป้าหมายเพื่อให้เข้าใจถึงสิ่งที่ผู้ใช้งานต้องการจากระบบ
2. Simplicity: การออกแบบที่เน้นความเรียบง่ายและใช้งานง่าย ทำให้ผู้ใช้งานสามารถทำความเข้าใจและใช้ระบบได้โดยไม่ต้องมีการสอนหรือคำแนะนำ
3. Consistency: การออกแบบที่มีความสม่ำเสมอในทุกๆ ส่วนของระบบ เช่น สี, Font, ปุ่ม, และการนำทาง เพื่อให้ผู้ใช้งานรู้สึกคุ้นเคยและสามารถใช้งานได้อย่างราบรื่น
4. Feedback: แจ้งเตือนผู้ใช้งานทันทีเมื่อผู้ใช้งานทำการกระทำใดๆ เช่น การกดปุ่ม การกรอกข้อมูล เพื่อให้ผู้ใช้งานทราบว่าสิ่งที่ทำไปนั้น ได้รับการยอมรับหรือมีข้อผิดพลาดอะไรเกิดขึ้น

การออกแบบ Theme/Style/Color

1. Theme: Leafy จะมีธีมที่เกี่ยวข้องกับธรรมชาติและการจัดสวน ใช้สีที่ให้ความรู้สึกสดชื่นและสวยงาม เช่น สีเขียว, สีเทา, และสีขาว
2. Style: ใช้การออกแบบที่เรียบง่ายและทันสมัย เน้นการใช้พื้นที่ว่าง (White space) เพื่อให้การแสดงผลดูไม่อึดอัดและมีระเบียบ
3. Color: ใช้สีหลักเป็นสีเขียวเพื่อสื่อถึงความเป็นธรรมชาติ สีรองคือสีขาวและสีเทา เพื่อสร้างความสมดุลและความสวยงาม

การออกแบบ Wireframe

Wireframe เป็นขั้นตอนสำคัญในการออกแบบ UX/UI เพื่อวางแผนร่างและการจัดวางองค์ประกอบต่างๆ บนหน้าจอ โดยไม่ต้องสนใจเรื่องสีหรือสไตล์ เน้นไปที่การจัดวางปุ่ม การนำทาง และการจัดการข้อมูล โดย Wireframe จะช่วยให้เห็นภาพรวมของการจัดวางองค์ประกอบและการเชื่อมโยงระหว่างหน้าเพจต่างๆ ในระบบ เช่น

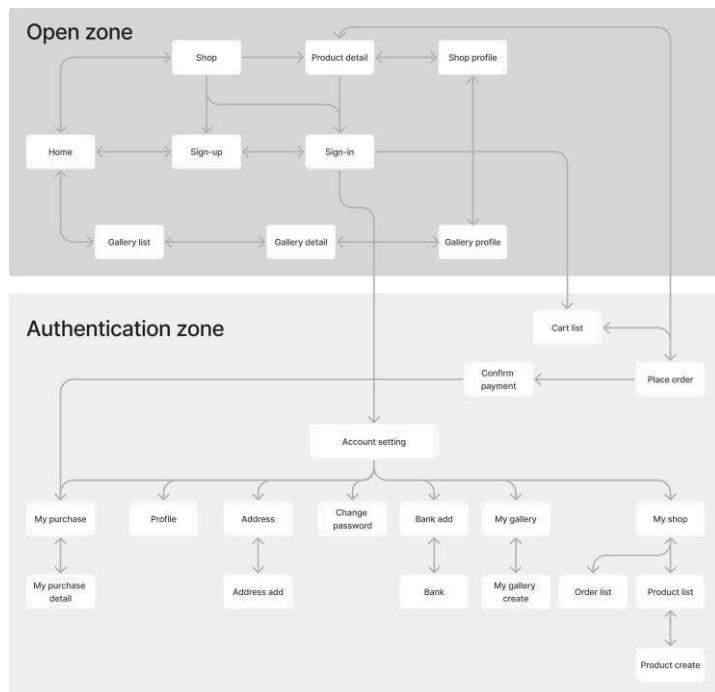
- Home Page: มีการออกแบบให้มีแถบเมนูหลักด้านบนเพื่อการนำทางไปยังหน้าต่างๆ เช่น หน้าแรก หน้าสินค้า หน้าบัญชีผู้ใช้ และหน้าอื่นๆ หน้าหลักจะมีแบบเนอร์ (Banner) ใหญ่แสดงข้อมูลสำคัญของระบบหรือสินค้าที่น่าสนใจ
- Product Page: จัดวางสินค้าที่แสดงในรูปแบบตาราง (Table) หรือกริด (Grid) โดยมีการแสดงภาพสินค้า ชื่อสินค้า ราคา และปุ่มเพิ่มสินค้าลงตะกร้า
- Cart Page: แสดงรายการสินค้าที่ผู้ใช้เลือกซื้อ โดยมีรายละเอียดของสินค้า เช่น ชื่อ สินค้า ราคา จำนวน และปุ่มสำหรับลบสินค้าหรือแก้ไขจำนวนสินค้า
- Checkout Page: มีการออกแบบให้ผู้ใช้สามารถกรอกข้อมูลการจัดส่งและข้อมูลการชำระเงินได้อย่างง่ายดายและเป็นระเบียบ

การออกแบบ Wireframe ช่วยให้ทีมพัฒนาสามารถเห็นภาพรวมของการจัดวางองค์ประกอบต่างๆ และสามารถปรับปรุงได้ตามความเหมาะสมก่อนที่จะลงมือออกแบบในระดับที่ละเอียดมากขึ้น

3.3.2 Frontend

1. Overview

หลักการที่ใช้ในการออกแบบ Component โดยจะเป็นการสร้าง Component ที่มีโอกาสใช้ซ้ำมากกว่า 1 ครั้ง เพื่อลดการเขียนโค้ดให้สั้นและประหยัดเวลามากขึ้น ดังแสดงในรูปที่ 3.3.2.1



รูปที่ 3.3.2.1 แผนภาพการออกแบบการเข้าถึง Page ต่างๆ ของ Leafy

- Folder JS: เก็บฟังก์ชันที่มีการเรียกใช้ซ้ำไว้ในไฟล์ JS เพื่อเรียกใช้ซ้ำได้โดยไม่จำเป็นต้องสร้างใหม่ทุกรอบ โดยเก็บข้อมูลดังนี้
 - Enum: เก็บคำที่มีการเรียกใช้ซ้ำบ่อยๆ เช่น คำที่ใช้และค่าของ Filter ประเภทต่างๆ
 - API: เก็บฟังก์ชันสำหรับเรียก API ที่มีโอกาสใช้งานบ่อยต่างๆ เช่น การดึงข้อมูลสินค้า
 - Cookie: เก็บฟังก์ชันที่เกี่ยวกับการใช้งาน Cookie โดยสามารถเรียกใช้งานได้ซ้ำตามประเภทการใช้งาน
 - Validation: รวบรวมฟังก์ชันสำหรับใช้งานทั่วไปที่มีโอกาสใช้งานมากกว่า 1 ครั้ง เช่น การตรวจสอบข้อมูลเฉพาะตัวเลข

3. Home page เป็นหน้าสำหรับแสดงข้อมูลเพื่อแนะนำเว็บไซต์ โดยมีการใช้ Component ดังแสดงตารางที่ 3.3.2.1

ตารางที่ 3.3.2.1 ข้อมูล Component ทั้งหมดที่ใช้ภายในหน้าเว็บ

Component	Description
BaseMenu	แสดงข้อมูลเกี่ยวกับปุ่มเพื่อนำพาไปยังหน้าต่างๆ และการค้นหาข้อมูลสินค้า
BaseInformation	แสดงข้อมูลที่เกี่ยวข้องกับเว็บไซต์
BaseSolution	แสดงข้อมูลที่บ่งบอกความต่างจากเว็บไซต์อื่นอย่างไร
BaseRandDesigner	แสดงข้อมูลเกี่ยวกับบทความที่ผู้ใช้สร้างขึ้นมาแบบสุ่ม
BaseHistory	แสดงข้อมูลความเป็นมาของ Leafy
BaseLandScapeDesign	แสดงข้อมูลตัวอย่างของการจัดสวนแต่ละแบบ
BasePlantInspo	แสดงข้อมูลตัวอย่างของต้นไม้แต่ละประเภท
BaseReview	แสดงข้อมูลที่มีการรีวิวสินค้าไว้ล่าสุด
BaseFooter	แสดงข้อมูลเกี่ยวกับ Service ต่างๆ ที่รองรับเพิ่มเติม

4. Shop page เป็นหน้าสำหรับแสดงข้อมูลเกี่ยวกับสินค้าทั้งหมดที่มีขายอยู่ โดยมีการใช้ Component ดังแสดงตารางที่ 3.3.2.2

ตารางที่ 3.3.2.2 ข้อมูล Component ทั้งหมดที่ใช้ภายในหน้าเว็บ

Component	Description
BaseMenu	แสดงข้อมูลเกี่ยวกับปุ่มเพื่อนำพาไปยังหน้าต่างๆ และการค้นหาข้อมูลสินค้า
BaseFilterItem	แสดงที่กรอกข้อมูลต่างๆ ที่สามารถคัดกรองข้อมูลสินค้าตามที่ต้องการได้

ตารางที่ 3.3.2.2 ข้อมูล Component ทั้งหมดที่ใช้ภายในหน้าเว็บ (ต่อ)

Component	Description
BaseSortItem	แสดงปุ่มที่สามารถเรียงลำดับข้อมูลสินค้าตามหมวดหมู่ต่างๆ ได้
BaseProductList	แสดงข้อมูลรายการสินค้าทั้งหมดโดยมีการกำหนดจำนวนข้อมูลสินค้าสูงสุดอยู่ที่ 18 รายการสินค้า
BaseEmptyList	ใช้สำหรับแสดงข้อมูลที่ เมื่อดึงมาไม่มีข้อมูลสินค้าที่ต้องการ อาจจะเกิดจากการกรองสินค้าหรือ Backend ที่มีปัญหา
BaseSelectPage	สำหรับเลือกหน้ารายการสินค้าที่ต้องการเปลี่ยน เนื่องจากข้อมูลสินค้ามีจำนวนมากจึงจำเป็นต้องแบ่งเป็นหน้าต่างย่อย
BaseAlert	แจ้งเตือน เมื่อมีการส่งคำร้องไปยัง Backend แต่ไม่มีการตอบกลับมา
BaseFooter	แสดงข้อมูลเกี่ยวกับ Service ต่างๆ ที่รองรับเพิ่มเติม

5. Product detail page เป็นหน้าสำหรับแสดงข้อมูลรายละเอียดของสินค้าทั้งหมด เพื่อประกอบการตัดสินใจในการซื้อ โดยมีการใช้ Component ดังแสดงตารางที่ 3.3.2.3

ตารางที่ 3.3.2.3 ข้อมูล Component ทั้งหมดที่ใช้ภายในหน้าเว็บ

Component	Description
BaseMenu	แสดงข้อมูลเกี่ยวกับปุ่มเพื่อนำพาไปยังหน้าต่างๆ และการค้นหาข้อมูลสินค้า

ตารางที่ 3.3.2.3 ข้อมูล Component ทั้งหมดที่ใช้ภายในหน้าเว็บ (ต่อ)

Component	Description
BaseProductType	แสดงข้อมูลเกี่ยวกับรายละเอียดของตัวสินค้า ทั้งหมดอย่างเช่น ขนาด สี เป็นต้น โดยมีการใช้ Component อื่นๆ เพิ่มเติมคือ <ul style="list-style-type: none"> • BaseStar เพื่อแสดงข้อมูลที่ผู้ซื้อคนอื่น ได้รีวิวไว้ • BaseAlert เพื่อแจ้งเตือนให้กับผู้ใช้ เกี่ยวกับ Stock ของสินค้า
BaseStore	แสดงข้อมูลของร้านค้าที่ขายสินค้าตัวนี้ ซึ่ง สามารถพาไปยังหน้า Profile ของร้านค้าเพื่อดู สินค้าเพิ่มเติมได้
BaseDescription	แสดงคำอธิบายรายละเอียดสินค้าเพิ่มเติมโดย ผู้ขาย
BaseReview	แสดงข้อมูลของผู้ซื้อที่มีความคิดเห็นต่อสินค้า เพื่อประกอบการตัดสินใจในการซื้อของผู้ซื้อคน อื่น โดยมีการใช้ Component อื่นๆ เพิ่มเติมคือ <ul style="list-style-type: none"> - BaseStar เพื่อแสดงข้อมูลที่ผู้ซื้อคนอื่น ได้รีวิวไว้
BaseSelectPage	ปุ่มสำหรับเลือกดูหน้าอื่นๆ ของ Review ใน สินค้าขึ้นนั้น
BaseRecommendation	สำหรับแสดงสินค้าที่มีการแนะนำ ซึ่งเป็นการ เรียกใช้งานร่วมกับ BaseProductList แต่ถูก ปรับแต่งให้แสดงผลในรูปแบบตามที่ออกแบบไว้ โดยมีการใช้ Component อื่นๆ เพิ่มเติมคือ <ul style="list-style-type: none"> - BaseProductList เพื่อแนะนำสินค้า อื่นๆ ให้กับผู้ใช้งาน
BaseAlert	แจ้งเตือน เมื่อมีการส่งคำร้องไปยัง Backend และ ไม่มีการตอบกลับมา
BaseFooter	แสดงข้อมูลเกี่ยวกับ Service ต่างๆ ที่รองรับ เพิ่มเติม

6. Gallery list page เป็นหน้าสำหรับแสดงข้อมูลของบุทความการอุกเบบสวนทั้งหมด ที่ผู้ใช้ทุกคนสามารถเขียนเป็นของตัวเองได้ โดยมีการใช้ Component ดังแสดงตารางที่ 3.3.2.4

ตารางที่ 3.3.2.4 ข้อมูล Component ทั้งหมดที่ใช้ภายในหน้าเว็บ

Component	Description
BaseMenu	แสดงข้อมูลเกี่ยวกับปุ่มเพื่อนำพาไปยังหน้าต่างๆ และการค้นหาข้อมูลสินค้า
BaseGallerySort	แสดงปุ่มสำหรับกรองข้อมูลตามลำดับในหมวดหมู่ต่างๆ
BaseGalleryCard	แสดงข้อมูลของบุทความสั้นๆ เพื่อให้ผู้ใช้กดเข้าไปดูรายละเอียดเพิ่มเติมได้
BaseEmptyList	แจ้งผู้ใช้ เมื่อกรองข้อมูลแล้วไม่พบหรือเกิดข้อผิดพลาดของ Backend
BaseSelectPage	แสดงปุ่มสำหรับเปลี่ยนหน้าของบุทความเพื่อดูข้อมูลเพิ่มเติมได้
BaseAlert	แจ้งเตือน เมื่อมีการส่งคำร้องไปยัง Backend แต่ไม่มีการตอบกลับมา
BaseFooter	แสดงข้อมูลเกี่ยวกับ Service ต่างๆ ที่รองรับเพิ่มเติม

7. Gallery detail page เป็นหน้าสำหรับแสดงรายละเอียดของบุทความทั้งหมด โดยมีการใช้ Component ดังแสดงตารางที่ 3.3.2.5

ตารางที่ 3.3.2.5 ข้อมูล Component ทั้งหมดที่ใช้ภายในหน้าเว็บ

Component	Description
BaseMenu	แสดงข้อมูลเกี่ยวกับปุ่มเพื่อนำพาไปยังหน้าต่างๆ และการค้นหาข้อมูลสินค้า

BaseGallerySort	แสดงปุ่มสำหรับกรองข้อมูลตามลำดับในหมวดหมู่ต่างๆ
-----------------	---

ตารางที่ 3.3.2.5 ข้อมูล Component ทั้งหมดที่ใช้ภายในหน้าเว็บ (ต่อ)

Component	Description
BaseGalleryCard	แสดงข้อมูลของบทความสั้นๆ เพื่อให้ผู้ใช้กดเข้าไปดูรายละเอียดเพิ่มเติมได้
BaseEmptyList	แจ้งผู้ใช้ เมื่อกรองข้อมูลแล้วไม่พบหรือเกิดข้อผิดพลาดของ Backend
BaseSelectPage	แสดงปุ่มสำหรับเปลี่ยนหน้าของบทความเพื่อดูข้อมูลเพิ่มเติมได้
BaseAlert	แจ้งเตือน เมื่อมีการส่งคำร้องไปยัง Backend แต่ไม่มีการตอบกลับมา
BaseFooter	แสดงข้อมูลเกี่ยวกับ Service ต่างๆ ที่รองรับเพิ่มเติม

8. Gallery detail page เป็นหน้าสำหรับแสดงรายละเอียดของข้อมูลบทความ ออกรูปแบบจัดสวน โดยมีการใช้ Component ดังแสดงตารางที่ 3.3.2.6

ตารางที่ 3.3.2.6 ข้อมูล Component ทั้งหมดที่ใช้ภายในหน้าเว็บ

Component	Description
BaseMenu	แสดงข้อมูลเกี่ยวกับปุ่มเพื่อนำพาไปยังหน้าต่างๆ และการค้นหาข้อมูลสินค้า
BaseAlert	แจ้งเตือน เมื่อมีการส่งคำร้องไปยัง Backend แต่ไม่มีการตอบกลับมา
BaseFooter	แสดงข้อมูลเกี่ยวกับ Service ต่างๆ ที่รองรับเพิ่มเติม

9. Profile page เป็นหน้าสำหรับแสดงข้อมูล Profile ของผู้ใช้ต่างๆ รวมถึงข้อมูลสินค้าที่เจ้าของ Profile มีขายทั้งหมด โดยมีการใช้ Component ดังแสดงตารางที่ 3.3.2.7

ตารางที่ 3.3.2.7 ข้อมูล Component ทั้งหมดที่ใช้ภายในหน้าเว็บ

Component	Description
BaseMenu	แสดงข้อมูลเกี่ยวกับปุ่มเพื่อนำพาไปยังหน้าต่างๆ และการค้นหาข้อมูลสินค้า
BaseProductList	แสดงข้อมูลเกี่ยวกับสินค้าของเจ้าของ Profile ทั้งหมด และรวมถึงการใช้เพื่อแนะนำสินค้าอื่นๆ เพิ่มเติม
BaseEmptyList	แจ้งผู้ใช้ เมื่อไม่พบข้อมูลสินค้าหรือเกิดข้อผิดพลาดของ Backend
BaseFilterItem	ใช้สำหรับกรองข้อมูลสินค้าที่มีทั้งหมดของเจ้าของ Profile
Basesortitem	ใช้สำหรับกรองข้อมูลสินค้าของเจ้าของ Profile ตามลำดับตามหมวดหมู่ต่างๆ
BaseSelectPage	แสดงปุ่มสำหรับเลือกดูข้อมูลสินค้าอื่นๆ
BaseConfirm	ใชเมื่อผู้ใช้กดเข้าไปดู Profile ของตัวเองจะแสดงเพื่อยืนยันหากผู้ใช้ต้องการลบข้อมูลสินค้าของตัวเอง
BaseStar	ใช้สำหรับแสดงข้อมูลจำนวนดาวของสินค้า
BaseAlert	แจ้งเตือน เมื่อมีการส่งคำร้องไปยัง Backend แต่ไม่มีการตอบกลับมา
BaseFooter	แสดงข้อมูลเกี่ยวกับ Service ต่างๆ ที่รองรับเพิ่มเติม

10. Gallery Profile page เป็นหน้าสำหรับแสดงโปรไฟล์ของผู้เขียนบทความการอุกเบบ จัดสวน โดยมีการใช้ Component ดังแสดงตารางที่ 3.3.2.8

ตารางที่ 3.3.2.8 ข้อมูล Component ทั้งหมดที่ใช้ภายในหน้าเว็บ

Component	Description
BaseMenu	แสดงข้อมูลเกี่ยวกับปุ่มเพื่อนำพาไปยังหน้าต่างๆ และการค้นหาข้อมูลสินค้า
BaseGallerySort	แสดงปุ่ม เพื่อใช้ในการกรองลำดับตามหมวดหมู่ต่างๆ ของบทความ
BaseGalleryCard	แสดงข้อมูลของบทความสั้นๆ ที่เจ้าของ Profile เป็นคนเขียนทั้งหมด เพื่อให้ผู้ใช้กดเข้าไปดูรายละเอียดเพิ่มเติมได้
BaseSelectPage	สามารถเปลี่ยนหน้าเนื้อหาเพื่อดูข้อมูลบทความเพิ่มเติมได้
BaseConfirm	ใช้เมื่อผู้ใช้กดเข้าไปดู Profile ของตัวเองจะแสดงเพื่อยืนยันหากผู้ใช้ต้องการลบบทความของตัวเอง
BaseEmptyList	ใช้สำหรับแสดงข้อความ เมื่อเจ้าของ Profile ไม่มีบทความที่เกี่ยวข้อง
BaseAlert	แจ้งเตือน เมื่อมีการส่งคำร้องไปยัง Backend แต่ไม่มีการตอบกลับมา
BaseFooter	แสดงข้อมูลเกี่ยวกับ Service ต่างๆ ที่รองรับเพิ่มเติม

11. Sign-in page เป็นหน้าสำหรับเข้าสู่ระบบของผู้ใช้ที่มีอยู่ในฐานข้อมูล โดยมีการใช้ Component ดังแสดงตารางที่ 3.3.2.9

ตารางที่ 3.3.2.9 ข้อมูล Component ทั้งหมดที่ใช้ภายในหน้าเว็บ

Component	Description
BaseMenu	แสดงข้อมูลเกี่ยวกับปุ่มเพื่อนำพาไปยังหน้าต่างๆ และการค้นหาข้อมูลสินค้า

ตารางที่ 3.3.2.9 ข้อมูล Component ทั้งหมดที่ใช้ภายในหน้าเว็บ (ต่อ)

Component	Description
BaseAlert	แจ้งเตือน เมื่อมีการส่งคำร้องไปยัง Backend แต่ไม่มีการตอบกลับมา
BaseFooter	แสดงข้อมูลเกี่ยวกับ Service ต่างๆ ที่รองรับเพิ่มเติม

12. Sign-up page เป็นหน้าสำหรับการสมัครสมาชิกเพื่อเข้าใช้บริการเว็บไซต์ โดยมีการใช้ Component ดังแสดงตารางที่ 3.3.2.10

ตารางที่ 3.3.2.10 ข้อมูล Component ทั้งหมดที่ใช้ภายในหน้าเว็บ

Component	Description
BaseMenu	แสดงข้อมูลเกี่ยวกับปุ่มเพื่อนำพาไปยังหน้าต่างๆ และการค้นหาข้อมูลสินค้า
BaseAlert	แจ้งเตือน เมื่อมีการส่งคำร้องไปยัง Backend แต่ไม่มีการตอบกลับมา
BaseFooter	แสดงข้อมูลเกี่ยวกับ Service ต่างๆ ที่รองรับเพิ่มเติม

13. Cart list page เป็นหน้าสำหรับแสดงข้อมูลรายการสินค้าทั้งหมดของแต่ละผู้ใช้งาน โดยมีการใช้ Component ดังแสดงตารางที่ 3.3.2.11

ตารางที่ 3.3.2.11 ข้อมูล Component ทั้งหมดที่ใช้ภายในหน้าเว็บ

Component	Description
BaseMenu	แสดงข้อมูลเกี่ยวกับปุ่มเพื่อนำพาไปยังหน้าต่างๆ และการค้นหาข้อมูลสินค้า
BaseProductList	ใช้แสดงสินค้าแนะนำสำหรับผู้ใช้เพิ่มเติม
BaseSummary	แสดงข้อมูลเกี่ยวกับราคารวมทั้งหมดของสินค้าในตะกร้าของผู้ใช้

ตารางที่ 3.3.2.11 ข้อมูล Component ทั้งหมดที่ใช้ภายในหน้าเว็บ (ต่อ)

Component	Description
BaseConfirm	ใช้สำหรับยืนยันเมื่อผู้ใช้ต้องการเอาสินค้าออกจากตะกร้า
BaseEmptyList	ใช้สำหรับแสดงข้อความ เมื่อผู้ใช้ไม่มีข้อมูลสินค้าในตะกร้า
BaseAlert	แจ้งเตือน เมื่อมีการส่งคำร้องไปยัง Backend แต่ไม่มีการตอบกลับมา
BaseFooter	แสดงข้อมูลเกี่ยวกับ Service ต่างๆ ที่รองรับเพิ่มเติม

14. Place order page เป็นหน้าสำหรับการตรวจสอบ Order ของผู้ใช้งาน เพื่อแจ้งเตือนไปยังร้านค้าเกี่ยวกับรายการสินค้าใหม่ที่เพิ่มเข้ามา โดยมีการใช้ Component ดังแสดง ตารางที่ 3.3.2.12

ตารางที่ 3.3.2.12 ข้อมูล Component ทั้งหมดที่ใช้ภายในหน้าเว็บ

Component	Description
BaseMenu	แสดงข้อมูลเกี่ยวกับปุ่มเพื่อนำพาไปยังหน้าต่างๆ และการค้นหาข้อมูลสินค้า
BaseShowErrorInput	ใช้สำหรับแสดงข้อความแจ้งเตือนผู้ใช้ เมื่อผู้ใช้งานกรอกข้อมูลไม่ถูกต้อง
BaseBankItem	เป็น Template สำหรับแสดงข้อมูล Address ของผู้ใช้งานทั้งหมด
BaseSummary	แสดงข้อมูลเกี่ยวกับราคารวมสินค้าที่ผู้ใช้เลือกมาจากตะกร้าหรือจากหน้ารายละเอียดสินค้า
BaseOrderItem	ใช้สำหรับแสดงข้อมูลสินค้า โดยแยกตามร้านค้า เพื่อให้ผู้ใช้งานดูได้ง่ายขึ้น
BaseAlert	แจ้งเตือน เมื่อมีการส่งคำร้องไปยัง Backend แต่ไม่มีการตอบกลับมา

ตารางที่ 3.3.2.12 ข้อมูล Component ทั้งหมดที่ใช้ภายในหน้าเว็บ (ต่อ)

Component	Description
BaseFooter	แสดงข้อมูลเกี่ยวกับ Service ต่างๆ ที่รองรับเพิ่มเติม

15. Confirm order page เป็นหน้าสำหรับแสดงรายละเอียดของสินค้า เพื่อจำลองการชำระเงิน โดยมีการใช้ Component ดังแสดงตารางที่ 3.3.2.13

ตารางที่ 3.3.2.13 ข้อมูล Component ทั้งหมดที่ใช้ภายในหน้าเว็บ

Component	Description
BaseMenu	แสดงข้อมูลเกี่ยวกับปุ่มเพื่อนำพาไปยังหน้าต่างๆ และการค้นหาข้อมูลสินค้า
BaseAlert	แจ้งเตือน เมื่อมีการส่งคำร้องไปยัง Backend แต่ไม่มีการตอบกลับมา
BaseFooter	แสดงข้อมูลเกี่ยวกับ Service ต่างๆ ที่รองรับเพิ่มเติม

16. Setting page เป็นหน้าหลักสำหรับแสดงข้อมูลเกี่ยวกับการตั้งค่า Account ต่างๆ โดยมีการใช้ Component ดังแสดงตารางที่ 3.3.2.14

ตารางที่ 3.3.2.14 ข้อมูล Component ทั้งหมดที่ใช้ภายในหน้าเว็บ

Component	Description
BaseMenu	แสดงข้อมูลเกี่ยวกับปุ่มเพื่อนำพาไปยังหน้าต่างๆ และการค้นหาข้อมูลสินค้า
BaseFooter	แสดงข้อมูลเกี่ยวกับ Service ต่างๆ ที่รองรับเพิ่มเติม

และมีการเรียกใช้งานหน้าต่างย่อยภายใต้ใน ดังนี้

- Profile เป็นหน้าสำหรับแสดงข้อมูลของผู้ใช้งาน ซึ่งสามารถแก้ไขได้ตามที่ผู้ใช้งานต้องการ โดยมีการใช้งาน Component ดังแสดงตารางที่ 3.3.2.15
ตารางที่ 3.3.2.15 ข้อมูล Component ทั้งหมดที่ใช้ภายในหน้าเว็บ

Component	Description
BaseShowErrorInput	ใช้สำหรับแจ้งเตือนผู้ใช้งาน เมื่อกรอกข้อมูลไม่ถูกต้อง
BaseSubmit	เป็น Template ปุ่มสำหรับการยืนยัน เมื่อแก้ไขข้อมูลสำเร็จ
BaseAlert	แจ้งเตือน เมื่อมีการส่งคำร้องไปยัง Backend แต่ไม่มีการตอบกลับมาหรือแจ้งเตือนต่างๆ เกี่ยวกับการกรอกข้อมูลรูปภาพ

- Address และ Address_add เป็นหน้าสำหรับเพิ่มและแก้ไขที่อยู่ของผู้ใช้งาน เพื่อข้อมูลใช้ในการจัดส่งสินค้า โดยมีการใช้งาน Component ดังแสดงตารางที่ 3.3.2.16

ตารางที่ 3.3.2.16 ข้อมูล Component ทั้งหมดที่ใช้ภายในหน้าเว็บ

Component	Description
BaseShowErrorInput	ใช้สำหรับแจ้งเตือนผู้ใช้งาน เมื่อกรอกข้อมูลไม่ถูกต้อง
BaseSubmit	เป็น Template ปุ่มสำหรับการยืนยัน เมื่อแก้ไขข้อมูลสำเร็จ
BaseAlert	แจ้งเตือน เมื่อมีการส่งคำร้องไปยัง Backend แต่ไม่มีการตอบกลับมา
BaseConfirm	ใช้สำหรับยืนยันเมื่อผู้ใช้งานต้องการที่จะลบข้อมูลที่อยู่
BaseEmptyList	แสดงเมื่อไม่พบข้อมูลที่อยู่ของผู้ใช้
BaseBankItem	เป็น Template แสดงข้อมูลที่อยู่ของผู้ใช้งาน

3. Change password เป็นหน้าสำหรับเปลี่ยน Password ของผู้ใช้งาน โดยมีการใช้งาน Component ดังแสดงตารางที่ 3.4.2.17

ตารางที่ 3.3.2.17 ข้อมูล Component ทั้งหมดที่ใช้ภายในหน้าเว็บ

Component	Description
BaseShowErrorInput	ใช้สำหรับแจ้งเตือนผู้ใช้งาน เมื่อกรอกข้อมูลไม่ถูกต้อง
BaseSubmit	เป็น Template ปุ่มสำหรับการยืนยัน เมื่อแก้ไขข้อมูลสำเร็จ
BaseAlert	แจ้งเตือน เมื่อมีการส่งคำร้องไปยัง Backend แต่ไม่มีการตอบกลับมา

4. Banks และ Bank_add เป็นหน้าสำหรับเพิ่มและแก้ไขข้อมูลธนาคารของผู้ใช้งาน เพื่อข้อมูลใช้ใน Feature การชำระเงินในอนาคต โดยมีการใช้งาน Component ดังแสดงตารางที่ 3.3.2.18

ตารางที่ 3.3.2.18 ข้อมูล Component ทั้งหมดที่ใช้ภายในหน้าเว็บ

Component	Description
BaseShowErrorInput	ใช้สำหรับแจ้งเตือนผู้ใช้งาน เมื่อกรอกข้อมูลไม่ถูกต้อง
BaseSubmit	เป็น Template ปุ่มสำหรับการยืนยัน เมื่อแก้ไขข้อมูลสำเร็จ
BaseAlert	แจ้งเตือน เมื่อมีการส่งคำร้องไปยัง Backend แต่ไม่มีการตอบกลับมา
BaseConfirm	ใช้สำหรับยืนยันเมื่อผู้ใช้งานต้องการที่จะลบข้อมูลธนาคาร
BaseEmptyList	แสดงเมื่อไม่พบข้อมูลธนาคารของผู้ใช้
BaseBankItem	เป็น Template แสดงข้อมูลธนาคารของผู้ใช้งาน

5. My purchase และ My purchase details เป็นหน้าสำหรับแสดงข้อมูลสินค้าที่มีการ Place order ไปทั้งหมด และมีการแสดงรายละเอียดข้อมูลต่างๆ ของ Order นั้นๆ เพิ่มเติม โดยมีการใช้งาน Component ดังแสดงตารางที่ 3.3.2.19

ตารางที่ 3.3.2.19 ข้อมูล Component ทั้งหมดที่ใช้ภายในหน้าเว็บ

Component	Description
BaseOrderGroup	เป็น Template สำหรับแสดงกลุ่มข้อมูล Order สินค้าที่อยู่ในสถานะยังไม่ชำระเงิน โดยมีการใช้ Component อื่นๆ เพิ่มเติมคือ <ul style="list-style-type: none"> - BaseAlert เพื่อแจ้งเตือนให้กับผู้ใช้งาน เมื่อผู้ใช้งานได้รีวิวสินค้า - BaseOrderItem ใช้สำหรับแสดงข้อมูล Order สินค้าโดยแยกเป็นแต่ละร้านค้า
BaseSelectPage	สำหรับเปลี่ยนหน้าเพื่อดูรายการ Order สินค้าของผู้ใช้เพิ่มเติม
BaseAlert	แจ้งเตือน เมื่อมีการส่งคำร้องไปยัง Backend แต่ไม่มีการตอบกลับมา
BaseOrderItem	เป็น Template สำหรับแสดงข้อมูล Order สินค้าที่อยู่ในสถานะต่างๆ
BaseEmptyList	แสดงเมื่อไม่พบข้อมูล Order สินค้าของผู้ใช้

6. My gallery และ My gallery Create เป็นหน้าสำหรับเพิ่มบทความเกี่ยวกับการออกแบบจัดสวนของผู้ใช้ทั่วไป โดยมีการใช้งาน Component ดังแสดงตารางที่ 3.3.2.20

ตารางที่ 3.3.2.20 ข้อมูล Component ทั้งหมดที่ใช้ภายในหน้าเว็บ

Component	Description
BaseMovePage	ปุ่มสำหรับเปลี่ยนหน้าเพื่อถูรยละเอียดของบทความเพิ่มเติม
BaseConfirm	แจ้งเตือนผู้ใช้ เมื่อผู้ใช้ต้องการที่จะลบบทความของผู้ใช้ออกจากระบบ
BaseAlert	แจ้งเตือน เมื่อมีการส่งคำร้องไปยัง Backend แต่ไม่มีการตอบกลับมา
BaseShowErrorInput	แจ้งเตือนผู้ขาย หากผู้ขายกรอกข้อมูลบทความไม่ถูกต้อง
BaseSubmit	เป็น Template ปุ่มสำหรับการยืนยัน เมื่อแก้ไขหรือเพิ่มข้อมูลสำเร็จ
BaseGallerySort	แสดงปุ่ม เพื่อใช้ในการกรองลำดับตามหมวดหมู่ต่างๆ ของบทความ
BaseEmptyList	แสดงเมื่อไม่พบข้อมูล Order สินค้าของผู้ใช้

7. My shop Product และ My shop Product creates เป็นหน้าสำหรับผู้ใช้งานที่มีบทบาทเป็นผู้ขาย ซึ่งสามารถเพิ่มสินค้าได้ตามที่ผู้ขายต้องการ โดยมีการใช้งาน Component ดังแสดงตารางที่ 3.3.2.21

ตารางที่ 3.3.2.21 ข้อมูล Component ทั้งหมดที่ใช้ภายในหน้าเว็บ

Component	Description
BaseMovePage	ปุ่มสำหรับเปลี่ยนหน้าเพื่อถูรยละเอียดของข้อมูลสินค้าเพิ่มเติม
BaseConfirm	แจ้งเตือนผู้ขาย เมื่อผู้ขายต้องการที่จะลบข้อมูลสินค้าของผู้ขายออกจากระบบ

ตารางที่ 3.3.2.21 ข้อมูล Component ทั้งหมดที่ใช้ภายในหน้าเว็บ (ต่อ)

Component	Description
BaseAlert	แจ้งเตือน เมื่อมีการส่งคำร้องไปยัง Backend แต่ไม่มีการตอบกลับมา
BaseShowErrorInput	แจ้งเตือนผู้ขาย หากผู้ขายกรอกข้อมูลสินค้าไม่ถูกต้อง
BaseSubmit	เป็น Template ปุ่มสำหรับการยืนยัน เมื่อแก้ไขหรือเพิ่มข้อมูลสำเร็จ
BaseStar	แสดงข้อมูลแสดงดาวของสินค้าที่ผู้ใช้คนอื่นๆ ได้รีวิวไว้
BaseEmptyList	แสดงเมื่อไม่พบข้อมูล Order สินค้าของผู้ใช้

8. My Shop Order list เป็นหน้าสำหรับแสดงข้อมูลของผู้ใช้งานที่ต้องการซื้อสินค้าจากผู้ขาย ซึ่งผู้ขายจะสามารถเตรียมสินค้าเพื่อจัดส่งให้กับผู้ใช้งานได้ โดยมีการใช้งาน Component ดังแสดงตารางที่ 3.3.2.22

ตารางที่ 3.3.2.22 ข้อมูล Component ทั้งหมดที่ใช้ภายในหน้าเว็บ

Component	Description
BaseMovePage	ปุ่มสำหรับเปลี่ยนหน้าเพื่อดูรายละเอียดของ Order สินค้าเพิ่มเติม
BaseConfirm	ใช้สำหรับยืนยันเมื่อผู้ขายต้องการที่จะเปลี่ยนสถานะของการสินค้าเพื่อจัดส่งให้กับผู้ลูกค้า
BaseAlert	แจ้งเตือน เมื่อมีการส่งคำร้องไปยัง Backend แต่ไม่มีการตอบกลับมา
BaseEmptyList	แสดงเมื่อไม่พบข้อมูล Order สินค้าของผู้ใช้

3.3.3 Backend

1. Overview

หลักการที่ใช้ในการออกแบบ API (Application Programming Interface) Services ข้างต้นข้อมูลตามเอกสาร API ซึ่งทำให้สามารถนำ API ดังกล่าวไปใช้ได้หลายครั้ง โดยเบื้องหลังการพัฒนา API ได้สร้างขึ้นเพื่อใช้ในการเรียกข้อมูลที่สำคัญ เช่น ข้อมูลผู้ใช้งาน รวมถึงผลิตภัณฑ์ที่สามารถปรับแต่ง (Customize) ได้เองตาม SKU (Stock Keeping Unit คือ ฉลากสินค้าที่ระบุจำนวนสินค้าและราคาที่แตกต่างกัน) [30] และความหลากหลายของ SKU นอกจากนี้ยังมีบทความเกี่ยวกับการออกแบบจัดสวน ซึ่งเปิดโอกาสให้ผู้คนที่สนใจได้เข้ามาเรียนรู้และเป็นแนวทางในการออกแบบจัดสวน

2. Role-based system

ระบบจัดการสินค้าสร้างขึ้นมาเพื่อรับรับจำนวนผู้ใช้งานได้หลายบทบาทคือ ในฐานะผู้ใช้ (User) จะต้องสามารถดูเฉพาะผลิตภัณฑ์ที่น่าสนใจ ซึ่งรองรับการค้นหาและการคัดกรองคุณสมบัติต่างๆ ตามที่ลูกค้าต้องการ นอกจากนี้ยังสนับสนุนระบบ Recommender system [16] เพื่อแนะนำสินค้าให้ลูกค้า โดยใช้หลักการการเรียนรู้ของ Machine learning ซึ่งจะอธิบายเพิ่มเติมในส่วนของการออกแบบซอฟต์แวร์ผู้ใช้ Machine learning โดยผู้ใช้สามารถดูรายละเอียดสินค้าที่ถูกกดไลค์บันรีวิวได้ ดูร้านค้าของผู้ขายผลิตภัณฑ์ สามารถเลื่อนดูรูปภาพและเลือกสินค้าที่ต้องการได้ ซึ่งนำไปสู่ตัวกรองสินค้าหรือสามารถซื้อได้ทันทีผ่านระบบ Order โดยรองรับการสร้างใบสั่งซื้อ (Order) โดยแนบด้วยที่อยู่ (Address) และวิธีการชำระเงิน (Payment) การสั่งซื้อสามารถดำเนินการจากร้านค้าหนึ่งร้านหรือหลายร้านก็ได้ ขึ้นอยู่กับการเลือกของผู้ใช้งาน แต่ละร้านสามารถดำเนินการก่อนที่จะส่งสินค้าถึงลูกค้า เมื่อสินค้าถึงลูกค้า ลูกค้าจะมีสิทธิรีวิวสินค้าหรือไม่ก็ได้ และสามารถดูบทความน่าสนใจของการออกแบบจัดสวน รวมถึงปรับเปลี่ยนข้อมูลส่วนตัวของเขารองจากนักออกแบบนี้ ระบบยังรองรับการแก้ไขข้อมูลส่วนตัว และข้อมูลการสั่งซื้อจาก Order ของผู้ใช้งานจากร้านค้าต่างๆ

ในฐานะของผู้จัดหาสินค้าหรือผู้ขาย (Supplier) สามารถจัดการข้อมูลได้เหมือนกับผู้ใช้ทั่วไป (User) แต่ในหน้าการตั้งค่า (Settings) และไฟล์ของร้าน พวกร้านสามารถระบุรายละเอียดของสินค้าที่จะขายได้ ซึ่งสินค้าที่เข้ายานั้นจะไม่ถูกแสดงในรายการสินค้าทั่วไป เนื่องจากเมื่อเข้าเพิ่มสินค้าลงใน Order จะไม่สามารถทำเช่นนั้นได้

ผู้จัดหาสินค้ายังสามารถปรับแต่งสินค้าได้หลากหลายจาก SKU นอกจากนี้ยังรองรับการอัปโหลดรูปภาพได้หลายรูปต่อ SKU เพื่อแสดงผลสินค้าอย่างชัดเจนบนหน้าผลิตภัณฑ์ของผู้ใช้งาน ผู้จัดหาสินค้าจะต้องจัดการ Order ที่ลูกค้าซื้อสินค้าของพวกร้าน โดยเมื่อสินค้าถูกซื้อไปแล้ว ทางร้านจะต้องเตรียมสินค้าเพื่อนำไปจัดส่งให้แก่ลูกค้าทันที

ในฐานะของ Admin สามารถจัดการข้อมูลผู้ใช้งานในระบบจัดการสินค้า

3. API documentation เป็นเอกสารที่ใช้ในการตรวจสอบ API (Application Programming Interface) ด้วยปัจจัยสำคัญคือ Endpoint ที่ทำงานในแต่ละหมวดหมู่ เช่น ผู้ใช้งาน (User), สินค้า (Product), ตะกร้า (Cart), ใบเสร็จจ่ายเงิน (Order) การวิจารณ์สินค้า (Product review) และแนวการอภิแบบจัดสวน (Garden content)

API Method บ่งบอกถึงมาตรฐานในการเรียกใช้ API โดย GET เป็น Method ในการเรียกข้อมูลขึ้นมาโดยไม่ใส่ข้อมูลบน Body, POST เป็น method การเรียกข้อมูลขึ้นมาโดยใส่ข้อมูลบน Body ก่อนที่สร้าง, PUT เป็น Method การแก้ไขข้อมูลขึ้นมาจากใส่ข้อมูลให้ครบใน Field ที่กำหนด , PATCH เป็น Method เหมือนกับ PUT แต่ว่าสามารถแก้ไขข้อมูลใน Field ไหนก็ได้ และ DELETE เป็น Method ที่ใช้ในการลบข้อมูล

เมื่อส่งข้อมูลกลับมาจะต้องส่ง HTTP Status กลับไป เช่น 2xx เป็นการส่งผลลัพธ์กลับมาได้อย่างสำเร็จ, 4xx และ 5xx เมื่อส่งผลลัพธ์จากข้อผิดพลาดของมา

เมื่อมีระบบ Authentication และ Authorization มาแล้วต้องสามารถตรวจสอบได้ว่า ข้อมูลดังกล่าวต้องใช้ JWT (JSON Web Token) หรือไม่ และสามารถตรวจสอบว่าถ้า user ใช้ JWT ต้องระบุชื่อผู้ใช้งาน (User information) บทบาท (Role) ของพวกรเข้า

โดยเอกสาร API นำข้อมูล Endpoint ทุกส่วนที่ความเข้าใจกับผู้พัฒนาเว็บแอพพลิเคชันในการสื่อสารและการเรียกใช้ข้อมูลดังกล่าวได้อย่างถูกต้องในส่วนต่างๆ ของหน้าเว็บไซต์ ดังแสดงในตารางที่ 3.3.3.1-3.3.3.9

ตารางที่ 3.3.3.1 Authentication endpoint

Endpoint	Method	HTTP Status	Use JWT	Description
/api/authentication/	POST	200 401,403	No	กรอกข้อมูลของ User และรหัสผ่านเพื่อเข้าใช้งาน โดยสร้าง Access Token และ Refresh Token
/api/authentication/refresh	POST	200 401	Yes	ใช้ Refresh Token สร้าง Access Token และ Refresh Token ใหม่ เพื่อให้ User สามารถใช้งานใหม่ได้
/api/authentication/signout	GET	200 404	Yes	ลบ Access Token และ Refresh Token และบังคับให้ User Login ใหม่

ตารางที่ 3.3.3.3.2 User endpoint

Endpoint	Method	Status	Use JWT	Admin	User	Supplier	Description
/api/users(?(name=:name email=:email role=:role) & page=:page & limit=:limit)	GET	200 403	Yes	Yes	No	No	เรียกข้อมูลของ User ทั้งหมด โดยสามารถผ่าน Filter Query และเป็น Page Custom ได้
/api/users/garden_designer	GET	200	No	Yes	Yes	Yes	เรียกข้อมูลที่เรียงจากบุคคลที่มีผล Popular มากที่สุดจากยอด Like ใน การจัดสวน
/api/users/:id	GET	200 404	Yes	Yes	Username	Username	เรียกรายละเอียดข้อมูลของ User ผ่าน Id หรือ Username ของ Admin ได้
/api/users/views/:id	GET	200 404	No	Yes	Yes	Yes	เรียกรายละเอียดข้อมูลของ User ผ่าน Id หรือ Username ได้
/api/users	POST	201 400, 403	Yes	Yes	No	No	สร้าง User และ Role
/api/users/register	POST	201 400	no	Yes	Yes	Yes	สร้าง User ภายใต้ Role Guest
/api/users/:id	PATCH	200 400, 403,404	Yes	Yes	No	No	แก้ไขข้อมูลของ User โดยผ่าน Id หรือ Username
/api/users/views/edit	PATCH	200 400	Yes	Username	Username	Username	แก้ไขข้อมูลส่วนตัวของ User
/api/users/:id	DELETE	200 403, 404	Yes	Yes	No	No	ลบข้อมูลของ User โดย Id

ตารางที่ 3.3.3.3 Address endpoint

Endpoint	Method	Status	Use JWT	Admin	User	Supplier	Description
/api/addresses/:username	GET	200 404	Yes	Yes	Username	Username	เรียกที่อยู่ทั้งหมดของ User คนนั้น
/api/addresses/:username/:id	GET	200 404	Yes	Yes	Username	Username	เรียกที่อยู่โดย Address Id ของ User คนนั้น
/api/addresses/:username	POST	201 400, 403	Yes	Yes	Username	Username	สร้างที่อยู่ของ User คนนั้น
/api/addresses/:username/:id	PATCH	200 400, 403,404	Yes	Yes	Username	Username	แก้ไขที่อยู่ของ User คนนั้น
/api/addresses/:username/:id	DELETE	200 403, 404	Yes	Yes	Username	Username	ลบที่อยู่ของ User คนนั้น

ตารางที่ 3.3.3.4 Payments endpoint

Endpoint	Method	Status	Use JWT	Admin	User	Supplier	Description
/api/payments/:username	GET	200 404	Yes	Yes	Username	Username	เรียกบัญชีทั้งหมดของ User คนนั้น
/api/payments/:username/:id	GET	200 404	Yes	Yes	Username	Username	เรียกบัญชีโดย Payment id ของ User คนนั้น
/api/payments/:username	POST	201 400, 403	Yes	Yes	Username	Username	สร้างบัญชีของ User คนนั้น
/api/payments/:username/:id	PATCH	200 400, 403,404	Yes	Yes	Username	Username	แก้ไขบัญชีของ User คนนั้น
/api/payments/:username/:id	DELETE	200 403, 404	Yes	Yes	Username	Username	ลบบัญชีของ User คนนั้น

ตารางที่ 3.3.3.5 Products endpoint

Endpoint	Method	Status	Use JWT	Admin	User	Supplier	Description
/api/products(? (product=:name & type=:type & tag=:tag & min_price=:price & max_price=:price & rating=:rating, owner=:username, isRecommend=:isRecommen d) & page=:page & limit=:limit, sort_name=:sortName, sort=:sort)	GET	200 404	No	Yes	Yes	Yes	เรียกสินค้าที่มีทั้งหมด ออกมา ซึ่งสามารถ Filter, Sort, Recommend และ Paging ผ่าน Query กรณีของร้านค้าจะไม่ เท็งสินค้าของตัวเอง และเรียกสินค้าที่หมด Stock ออกไปด้วย
/api/products/supplier/:item Owner(? (type=:type, sort_name=:sortName, sort=:sort) & page=:page & limit=:limit)	GET	200 404	Yes	Yes	No	Yes	เรียกสินค้าจาก เจ้าของร้านเท่านั้น เพื่อทำการแก้ไข
/api/products/:id	GET	200 404	No	Yes	Yes	Yes	เรียกรายละเอียด สินค้าผ่านโดย Item Id
/api/products	POST	201 400, 403	Yes	Yes	No	Yes	เพิ่มสินค้าที่ Customize SKU และ Variation
/api/products/:id	POST	201 400, 403,404	Yes	Yes	No	Item Owner	เพิ่ม SKU ของสินค้า ตาม Variation โดย Item Id
/api/products/isFav/:id	PUT	200 400, 403,404	Yes	Not item Owner	Not item Owner	Not item Owner	เพิ่ม Favorite Item ของ User ผ่าน Item Id

ตารางที่ 3.3.3.3.5 Products endpoint (ต่อ)

Endpoint	Method	Status	Use JWT	Admin	User	Supplier	Description
/api/products/:id	PATCH	200 400, 403,404	Yes	Yes	No	Item Owner	แก้ไขข้อมูลของ Product เป็นต้น โดย Item Id
/api/products/:id/:style	PUT	200 400, 403,404	Yes	Yes	No	Item Owner	แก้ไขรายละเอียด ข้อมูลของ Product บน SKU ที่ทุก Variation โดย Item Id และ SKU Style Code
/api/products/:id/:style	DELETE	200 400, 403,404	Yes	Yes	No	Item Owner	ลบ Item SKU โดย Item id และ SKU Style Code
/api/products/:id	DELETE	200 403, 404	Yes	Yes	No	Item Owner	ลบสินค้า โดย Item Id
/api/products/all/reviews	GET	200	No	Yes	Yes	Yes	ดูวิจารณ์สินค้าทั้งหมด
/api/products/:prodid/reviews	GET	200 404	No	Yes	Yes	Yes	ดูวิจารณ์สินค้าทั้งหมด ผ่าน Item Id
/api/products/:prodid/reviews/:reviewId	GET	200 400, 403,404	No	Review Owner	Review Owner	Review Owner	ดูวิจารณ์สินค้าผ่าน Item Id และ Review Id
/api/products/:prodid/reviews/_orders/:orderId/:itemId/:itemStyle/:itemSize	GET	200 400, 403,404	No	Review Owner	Review Owner	Review Owner	ดูวิจารณ์สินค้าผ่าน Item Id และ SKU Style Code, Variation Name และ Order Id
/api/products/:prodid/reviews	POST	201 400, 403	Yes	Yes	Order Detail Item	Order Detail Item	สร้างวิจารณ์จาก รายละเอียดสินค้าบน Order Detail

ตารางที่ 3.3.3.5 Products endpoint (ต่อ)

Endpoint	Method	Status	Use JWT	Admin	User	Supplier	Description
/api/products/:prodid/reviews/:reviewid	PATCH	200 403, 404	Yes	Review Owner	Review Owner	Review Owner	แก้ไขข้อมูลวิจารณ์จาก Item Id และ Review id
/api/products/:prodid/reviews/:reviewid/like	PUT	200 400, 404	Yes	Yes	Yes	Yes	เพิ่มยอด Like บน Review ได้
/api/products/:prodid/reviews/:reviewid	DELETE	200 403, 404	Yes	Review Owner	Review Owner	Review Owner	ลบคำวิจารณ์บน Review โดย Item Id และ Review Id

ตารางที่ 3.3.3.6 Images endpoint

Endpoint	Method	Status	Use JWT	Admin	User	Supplier	Description
/api/image/product/:id	GET	200 404	No	Yes	Yes	Yes	Download ไฟล์ภาพที่ Stream จาก Minio ผ่าน Item Id
/api/image/product/:id	POST	201 400, 413	Yes	Yes	No	Item Owner	Upload รูปภาพ Stream ขึ้นบน Minio ผ่าน Item Id
/api/image/product/:id	DELETE	200 404	Yes	Yes	No	Item Owner	ลบรูปภาพบน Minio ผ่าน Item Id
/api/image/product/:id/:style/:filename	GET	200 404	No	Yes	Yes	Yes	Download ไฟล์ภาพจาก Minio ผ่าน Item Id และ SKU Style Code

ตารางที่ 3.3.3.6 Images endpoint (ต่อ)

Endpoint	Method	Status	Use JWT	Admin	User	Supplier	Description
/api/images/products/:id/:style	POST	201 400, 413,	Yes	Yes	No	Item Owner	Upload รูปภาพบน Minio ผ่าน Item Id และ SKU Style Code ด้วยรูปภาพหลายรูป
/api/images/products/:id/:style	DELETE	200 404, 500	Yes	Yes	No	Item Owner	ลบรูปภาพจาก Minio ผ่าน Item Id และ SKU Style Code
/api/images/products/:id/:style/all	DELETE	200 404, 500	Yes	Yes	No	Item Owner	ลบรูปภาพทั้งหมด จาก Minio ผ่าน Item Id และ SKU Style Code ผ่าน Item Id
/api/image/users/:id	GET	200 404	No	Yes	Yes	Yes	Download ไฟล์ภาพ Main จาก Minio ผ่าน User Id
/api/image/users/:id/coverphoto	GET	200 404	No	Yes	Yes	Yes	Download ไฟล์ภาพ Cover Photo จาก Minio ผ่าน User Id
/api/image/users/:id	POST	201 400, 413	Yes	Yes	Username	Username	Upload รูปภาพ Main บน Minio ผ่าน User Id
/api/image/users/:id/coverphoto	POST	201 400, 413	Yes	Yes	Username	Username	Upload รูปภาพ Cover Photo บน Minio ผ่าน User Id

ตารางที่ 3.3.3.6 Images endpoint (ต่อ)

Endpoint	Method	Status	Use JWT	Admin	User	Supplier	Description
/api/image/users/:id	DELETE	200 404	Yes	Yes	Username	Username	ลบรูปภาพ Main จาก Minio ผ่าน user Id
/api/image/users/:id/coverphoto	DELETE	200 404	Yes	Yes	Username	Username	ลบรูปภาพ Cover Photo จาก Minio ผ่าน User Id
/api/images/reviews/:id/:filename	GET	200 404	No	Yes	Yes	Yes	Download ไฟล์ภาพ จาก Minio ผ่าน Review Id และ Filename
/api/images/reviews/:id	POST	201 400, 413,	Yes	Yes	Review Owner	Review Owner	Upload รูปภาพบน Minio ผ่าน Review Id และ Filename ที่เลือก
/api/images/reviews/:id	DELETE	200 404, 500	Yes	Yes	Review Owner	Review Owner	ลบรูปภาพบน Minio ผ่าน Review Id และ Filename ที่เลือก
/api/images/reviews/:id/all	DELETE	200 404, 500	Yes	Yes	Review Owner	Review Owner	ลบรูปภาพทั้งหมดบน Minio ผ่าน Review Id
/api/image/gallery/:id	GET	200 404	No	Yes	Yes	Yes	Download ไฟล์ภาพบน Minio ผ่าน Item Id
/api/image/gallery/:id	POST	201 400, 413	Yes	Yes	Content Owner	Content Owner	Upload รูปภาพขึ้นบน Minio ผ่าน Item Id

ตารางที่ 3.3.3.6 Images endpoint (ต่อ)

Endpoint	Method	Status	Use JWT	Admin	User	Supplier	Description
/api/image/gallery/:id	DELETE	200 404	Yes	Yes	Content Owner	Content Owner	ลบรูปภาพบน Minio ผ่าน Item Id
/api/image/gallery/:id/:filename	GET	200 404	no	Yes	Yes	Yes	Download ไฟล์ภาพบน Minio ผ่าน Content Id และ Filename
/api/images/gallery/:id	POST	201 400, 413,	Yes	Yes	Content Owner	Content Owner	Upload รูปภาพขึ้นบน Minio ผ่าน Content Id และ Filename ที่เลือก
/api/images/gallery/:id	DELETE	200 404, 500	Yes	Yes	Content Owner	Content Owner	ลบรูปภาพบน Minio ผ่าน Content Id และ Filename ที่เลือก
/api/images/gallery/:id/all	DELETE	200 404, 500	Yes	Yes	Content Owner	Content Owner	ลบรูปภาพทั้งหมดบน Minio ผ่าน Content Id

ตารางที่ 3.3.3.7 Carts point

Endpoint	Method	Status	Use JWT	Admin	User	Supplier	Description
/api/carts	GET	200 404	Yes	Username	Username	Username	เรียกข้อมูลสินค้าทั้งหมดในตะกร้า
/api/carts/count	GET	200 404	Yes	Username	Username	Username	จำนวนสินค้าบนตะกร้าที่นับได้บนตะกร้าของลูกค้า
/api/carts/products	POST	201 400	Yes	Username	Username	Username	เพิ่มสินค้าบนตะกร้าของลูกค้า
/api/carts/:id	PUT	200 400, 403	Yes	Yes	Username	Username	แก้ไขจำนวนสินค้าที่ลูกค้าซื้อบนตะกร้าผ่าน Item Id
/api/carts/:cartId	DELETE	200 400	Yes	Yes	Username	Username	ลบสินค้าที่อยู่ในตะกร้าของลูกค้าผ่าน Item Id
/api/carts/:sessionId/session	DELETE	200 403, 404	Yes	Yes	Username	Username	ลบ Session ตะกร้าที่เก็บข้อมูลสินค้าของลูกค้า

ตารางที่ 3.3.3.8 Orders endpoint

Endpoint	Method	Status	Use JWT	Admin	User	Supplier	Description
/api/orders (? [owner_item_or_product=:owner_item_or_product & status=:status & sort=:sort & page=:page & limit=:limit])	GET	200 404	Yes	Yes	Username	Username	เรียก Order ทั้งหมดของ User คนนั้น ซึ่งสามารถดูข้อมูลสินค้าที่เข้าซื้อและสถานะของ Order, Sort และ Paging custom ได้

ตารางที่ 3.3.3.8 Orders endpoint (ต่อ)

Endpoint	Method	Status	Use JWT	Admin	User	Supplier	Description
/api/orders/count/:status	GET	200 404	Yes	Yes	Username	Username	จำนวน Order ที่ส่งไป
/api/orders/supplier (?[status=:status & dateStart=:dateStart & dateEnd=:dateEnd & sort=:sort & page=:page & limit=:limit])	GET	200 403, 404	Yes	Yes	No	Item Owner	เรียก Order ที่ร้านค้าได้รับจากการขาย สินค้า ซึ่งสามารถดูเวลาสินค้าที่ถูกซื้อ สามารถ Sort และ Paging Custom ได้
/api/orders/supplier/count/:status	GET	200 404	Yes	Yes	No	Item Owner	จำนวน Order ที่ร้านได้รับจากลูกค้า
/api/orders/:orderId	GET	200 403, 404	Yes	Yes	Username	Username	รายละเอียด Order ที่ได้รับจากลูกค้า รวมทั้งการวิจารณ์สินค้าของเจ้าด้วย
/api/orders/groups/:orderId	GET	200 403, 404	Yes	Yes	Username	Username	รายละเอียดกลุ่ม Order และภาพรวมการวิจารณ์สินค้าของเจ้าด้วย
/api/orders	POST	201 400, 403,404	Yes	Yes	Username	Username	สร้าง Order ขึ้นมาจากการเลือกสินค้า บนตัชกร้าและที่อยู่ ด้วยสถานะ Required
/api/orders/no_cart	POST	201 400, 403,404	Yes	Yes	Username	Username	สร้าง Order ขึ้นมาจากการเลือกสินค้า บนรายละเอียด Product และที่อยู่ ด้วยสถานะ Required

ตารางที่ 3.3.3.8 Orders endpoint (ต่อ)

Endpoint	Method	Status	Use JWT	Admin	User	Supplier	Description
/api/orders/check_order/:orderId	PUT	200 400, 403,404	Yes	Yes	Username	Username	ตรวจสอบ Order ผึ้งลูกค้าโดย Order Id
/api/orders/prepare_order/:orderId	PUT	200 400, 403,404	Yes	Yes	No	Item Owner	ตรวจสอบ Order ผึ้งร้านค้าโดย Order Id
/api/orders/pay_order/:orderGroupId	PUT	200 400, 403,404	Yes	Yes	Order Owner	Order Owner	ตรวจสอบ Order ผึ้งลูกค้าโดย Order Group Id

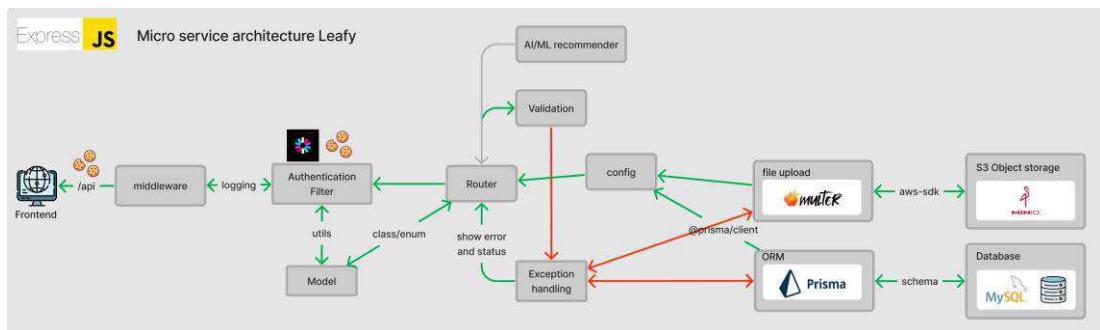
ตารางที่ 3.3.3.9 Galleries endpoint

Endpoint	Method	Status	Use JWT	Admin	User	Supplier	Description
/api/contents (?(page=:page & limit=:limit, sort_name=:sortName, sort=:sort, style=:style, content_owner=:contentOwner))	GET	200 404	No	Yes	Yes	Yes	เรียกบทความทั้งหมดที่สามารถ Filter ผ่าน Style, Content Owner, Sort และ Paging ได้
/api/contents/owner	GET	200 404	Yes	Content Owner	Content Owner	Content Owner	เรียกบทความของเจ้าของทั้งหมด
/api/contents/:id	GET	200 404	No	Yes	Yes	Yes	เรียกรายละเอียดของบทความที่มียอด Like และจำนวน Comment
/api/contents	POST	201 400, 403	Yes	Content Owner	Content Owner	Content Owner	เพิ่ม Content สำหรับ Garden Designer
/api/contents/:id	PATCH	200 400, 403,404	Yes	Content Owner	Content Owner	Content Owner	แก้ไข Content โดย Content Id

ตารางที่ 3.3.3.9 Galleries endpoint (ต่อ)

Endpoint	Method	Status	Use JWT	Admin	User	Supplier	Description
/api/contents/:id	DELETE	200 403, 404	Yes	Content Owner	Content Owner	Content Owner	ลบ Content โดย Content Id
/api/contents/:contentId/like	PUT	200 400, 404	Yes	Not Content Owner	Not Content Owner	Not Content Owner	กด Like ให้แก่ Content Designer
/api/contents/:contentId/comments	GET	200 404	No	Yes	Yes	Yes	เรียก Comment ทั้งหมดจาก Content Id
/api/contents/:contentId/comments	POST	201 400, 403	Yes	Yes	Yes	Yes	สร้าง Comment จาก Content Id
/api/contents/:contentId/comments/:commentId	DELETE	200 403, 404	Yes	Comment Owner	Comment Owner	Comment Owner	ลบ Comment ผ่าน Content Id และ Comment Id

4. Backend architecture layers ซึ่งได้อธิบายหลักการตามรูปที่ 3.3.4.1



รูปที่ 3.3.3.4.1 Overview Leafy Microservice Architecture Layers

ภาพรวมของ Microservice Architecture ผัง Server-side พบร่วมกับผู้ใช้ (User) เรียกข้อมูลจาก API ของ Backend ภายใต้ API นั้นใช้แนวคิดของ MVC (Model, View, Controller) ในการจัดการโครงสร้างและการทำงาน

- Model: เป็นส่วนที่จัดการโครงสร้างของข้อมูลและตรรกะที่เกี่ยวข้อง
- View: เป็นส่วนที่แสดงข้อมูลให้กับผู้ใช้
- Controller: เป็นส่วนที่ทำหน้าที่เชื่อมต่อระหว่าง Model และ View โดยรับคำสั่งจากผู้ใช้แล้วเรียกใช้ Model เพื่อนำข้อมูลมาแสดงผลใน View

แนวคิด MVC ใน Microservice Architecture นี้ช่วยให้การพัฒนาระบบท่มีความยืดหยุ่นและง่ายต่อการบำรุงรักษา และยังทำให้การจัดการ API มีประสิทธิภาพมากยิ่งขึ้น ซึ่งในระบบจัดการข้อมูลสินค้าใช้ Express.js framework ในการพัฒนาระบบจัดการข้อมูล ทำให้สามารถจัดการกระบวนการทำงานของข้อมูลสินค้าในรูปแบบต่างๆ ได้ยืดหยุ่น ดังนั้นโครงสร้าง MVC ของ Express.js มีดังนี้

Express Router เป็น Controller:

- Express router ทำหน้าที่เป็น Controller ในการรับ Endpoints แต่ละ Endpoint request จะถูกส่งไปยัง Router ที่มี Endpoint
- มีการปรับรูปแบบการส่งและรับ Message ให้สามารถส่งและรับข้อมูลได้
- เมื่อ Request สำเร็จ ส่ง Message และ HTTP status กลับไปบน Response ไม่ว่าจะเป็น JSON หรือไฟล์ต่างๆ

Response Template เป็น View:

- Template ที่ส่งกลับไปคือ View ซึ่งรองรับการส่งกลับทั้งในกรณีที่สำเร็จและการจัดการข้อผิดพลาด (Exception handling)
- เมื่อ Request ถูกส่งไปยัง Router endpoint จะมี Middleware เก็บ Log ของข้อมูลที่ส่งเข้าไป และต้องผ่าน Authentication filter โดยใช้ JWT (JSON Web Token) สำหรับการทำ Authentication filtering
- ถ้าการ Authentication ไม่ผ่าน จะส่ง Response ให้ผู้ใช้งานไป Login ก่อนใช้งาน และถ้าหากไปยุ่งกับบทบาทอื่นๆ หรือบุคคลอื่นๆ จะส่ง Response เพื่อบล็อกไม่ให้ผู้ใช้งานได้เข้าไปรับภาระการทำงานของบทบาทอื่นๆ หรือบุคคลอื่นๆ
- ระบบ Authentication ใช้ Session cookie โดยมี Cookie parser เป็นตัวสร้าง Http cookie ซึ่งสามารถปรับ Config เพื่อกำหนดวันหมดอายุและรับ Cookie จาก Request ได้ รวมถึง Authorization ที่เก็บข้อมูลของ User และ Role เพื่อรับ Role-based access

Exception Handling:

- ในกรณีที่เกิด Exception handling เช่นจากการ Validation ข้อมูลหรือการ Upload file จะมีการสร้าง Error และส่ง Response เป็น JSON message ที่มี Error message และ Status code กลับมา
- ข้อมูล Trace จะถูกเก็บใน Log เพื่อให้สามารถแก้ไขปัญหาได้ย้อนหลัง

Model:

- ข้อมูลจะถูกเก็บและเรียกใช้งานผ่าน ORM (Object Relational Mapping) ของ Prisma โดยมี Prisma schema เป็นตัว Mapping กับ MySQL
- รูปภาพจะถูกเก็บในรูปแบบ Object Storage จาก Minio ซึ่งต้อง Config เพื่อเชื่อมต่อไปยังจุดปลายทางได้โดยตรง

Recommender System:

- Node.js สามารถทำ Recommender system ได้ แต่อาจมีการเปลี่ยนแปลงในอนาคตขึ้นอยู่กับทรัพยากรที่มี
- หากมี User เข้ามาใช้งานจำนวนมาก จะต้องจัดการ Resource ภายในการมากขึ้นเพื่อป้องกันการล่มของระบบ

องค์ประกอบเพิ่มเติม:

- Class: รองรับ Method ในการเรียกใช้งาน เช่น การจัดการ Cookie เพื่อ Refresh token, การจัดการ Time format ของ Prisma, การกรองและเรียงข้อมูล, การ Query ข้อมูลแบบ Pagination และจัดส่วนของ Recommender system เป็นต้น
- Enum: เก็บข้อมูล Label ซึ่งมีประโยชน์ในการทำ Validation ข้อมูล และไม่สามารถแก้ไขได้
- Error: เป็น Model ที่ใช้ในการส่ง Response ในรูปแบบต่างๆ พร้อม Status code และ Trace เพื่อให้ผู้ใช้ได้ติดตามกิจกรรมและผู้พัฒนาสามารถแก้ไขข้อผิดพลาดในระบบได้รวดเร็ว

5. เทคนิคที่ทำให้ Project นี้ทำงานได้มีประสิทธิภาพซึ่งมีดังนี้

5.1 Prisma and Model

Prisma เป็นเครื่องมือที่ใช้เทคนิคของ ORM (Object-Relation Mapping) เพื่อดึงข้อมูลแบบตารางออกจาก Schema ที่เป็นเหมือน Model ก่อนที่จะนำโครงสร้างของฐานข้อมูลไปใช้งานได้จริง การใช้ Prisma จะทำให้การจัดการฐานข้อมูลเป็นไปอย่างสะดวกและรวดเร็วยิ่งขึ้น โดยสามารถใช้คำสั่ง

การทำ Generate Prisma model และ Migrate model เพื่อปรับเปลี่ยนโครงสร้างของฐานข้อมูลตามที่ต้องการ ก่อนจัดการข้อมูลใช้หลักการ ORM ไปเชื่อมต่อกับ MySQL โดยใช้ Schema บน /prisma/schema.prisma ตามภาพที่ 3.3.3.5.1.1 เพื่อดึงข้อมูลตามโครงสร้างของการสร้างตารางจาก Prisma model

```

spike> poc> expressjs> prisma> schema.prisma
1 generator client {
2   provider = "prisma-client-js"
3 }
4
5 datasource db {
6   provider = "mysql"
7   url      = env("DATABASE_URL")
8 }
9
10 model carts {
11   cartId    Int      @id @default(autoincrement())
12   userEmail String  @db.VarChar(100)
13   itemId    Int
14   qty       Int      @default(0)
15   isPaid    Boolean  @default(false)
16   items     Items    @relation(fields: [itemId], references: [itemId], map: "Carts_itemId_fkey")
17   accounts  Accounts @relation(fields: [userEmail], references: [email], map: "Carts_userEmail_fkey")
18
19   @@index([itemId], map: "fk_carts_items_idx")
20   @@index([userEmail], map: "fk_carts_users_idx")
21 }
22
23 model favprd {
24   userEmail String  @db.VarChar(100)
25   itemId    Int
26   items     Items    @relation(fields: [itemId], references: [itemId], map: "FavPrd_itemId_fkey")
27   accounts  Accounts @relation(fields: [userEmail], references: [email], map: "FavPrd_userEmail_fkey")
28
29   @@id([itemId, userEmail])
30   @@index([itemId], map: "fk_FavPRD_items1_idx")
31   @@index([userEmail], map: "fk_FavPRD_users1_idx")
32 }

```

รูปที่ 3.3.3.5.1.1 โครงสร้างฐานข้อมูลที่ใช้ใน Prisma Schema บน /prisma/schema.prisma

จะเห็นได้ว่า Prisma schema ได้เขียนต่อไปฐานข้อมูล MySQL เพื่อใช้ในการสร้างข้อมูล ซึ่งคล้ายกับคำสั่ง DDL (Data Definition Language) บน MySQL ในการสร้างตารางซึ่งคำสั่งดังกล่าวเนี้ยในการ Generate model ไปยังฐานข้อมูล โดยที่เก็บข้อมูล URL ไว้ที่ Environment file เพื่อรักษาความลับของข้อมูล และใช้ Prisma client เรียก (Query) ข้อมูลที่มากจากฐานข้อมูล

การเรียกใช้ข้อมูลผ่าน Prisma client ผ่าน Express router ผ่าน API มีต่อไปนี้ดังนี้ เช่น ถ้าเรียกใช้ Verify id ซึ่งสามารถตรวจสอบว่า Endpoint นี้มีข้อมูลหรือไม่ ก่อนที่นำ Product detail ไป Customize item ตาม SKU เพื่อสามารถเรียกความหลากหลายของสินค้า ตามราคาและจำนวนสินค้าในคลังได้ โครงสร้างของภาษาโปรแกรมในการเรียกข้อมูลสินค้าจากใช้ Prisma client ก็จะมีดังรูปที่ 3.3.3.5.1.2

```

// -- method zone --
const verifyId = async (id) => {
  // find product by id
  let filter_pd = await prisma.items.findFirst({
    where: {
      itemId: Number(id)
    },
    include: {
      item_preview: {
        orderBy: {
          createdAt: "desc"
        }
      }
    }
  })

  // check that product is found
  if (filter_pd == null) notFoundError("item id " + id + " does not exist")

  // return converter of product
  return productConverter(filter_pd)
}

```

รูปที่ 3.3.3.5.1.2 การเรียก Product โดย Product Id ผ่าน Prisma client

เมื่อข้อมูลเฉพาะส่วนที่ผู้ใช้งานต้องการนั้นในคำสั่งของ Prisma client มีการ Query คำสั่ง เพื่อกระทำการต่างๆ กับข้อมูล โดยจะมีคำสั่งคล้ายกับ Data Manipulation Language (DML) และ Data Query Language (DQL) ของ Structured Query Language (SQL) คือมีการเลือก จากทั้งหมดหรือเฉพาะ Column (Column selection), ใช้เงื่อนไขในการกรอง (Condition statement), มีการจำแนกเป็นหมวดหมู่ (Aggregate query), เรียกข้อมูลที่อยู่ตามตารางที่เกี่ยวข้อง ผ่าน Foreign key (Table relation), เรียกข้อมูลจาก Column, บอกลำดับข้อมูลและจำนวนข้อมูล ที่เรียกจำกัด

ในการจัดรูปแบบให้เข้ากับ Prisma นั้น จำเป็นต้องใช้ความยืดหยุ่นในการเรียกใช้โมเดล เช่น การใช้คำสั่ง Select ตามด้วย Field ที่ต้องการเรียกข้อมูลออกมา โดยต้องสามารถทำการจับคู่ Model เพื่อเลือก Model ในรูปแบบ JavaScript Object ออกมาให้ใช้งานได้อย่างเหมาะสม ทั้งนี้ ควรลบ Attribute ที่มีค่าเป็น Null ออกไปก่อนที่จะส่งกลับไปยัง Response ผ่าน View template ตามรูปที่ 3.3.3.5.1.3

```

103 let orderView = {
104   orderId: true,
105   orderGroupId: true,
106   customerName: true,
107   status: true,
108   createdAt: true,
109   address: true,
110   order_details: {
111     select: {
112       itemStyle: true,
113       itemSize: true,
114       itemId: true,
115       qtyOrder: true,
116       priceEach: true
117     }
118   }
119 }
120
121 let orderDetailView = {
122   orderId: true,
123   customerName: true,
124   status: true,
125   createdAt: true,
126   paidOrderDate: true,
127   shippedOrderDate: true,
128   receivedOrderDate: true,
129   rateOrderDate: true,
130   address: true,
131   order_details: {
132     select: {
133       itemStyle: true,
134       itemSize: true,
135       itemId: true,
136       qtyOrder: true,
137       priceEach: true
138     }
139   }
140 }

```

รูปที่ 3.3.3.5.1.3 Prisma Selector Model เพื่อใช้การควบคุมการเรียก Field จาก Prisma Client

5.2. Pagination List

การเรียกข้อมูลเป็น Page ทำให้ Frontend สามารถเรียกข้อมูลไปแสดงผลได้อย่างรวดเร็ว โดยใช้คำสั่ง Prisma ในการสร้างคำสั่งเพื่อบอกว่าหน้าที่เท่าไรและจำกัดไว้กี่ข้อมูลต่อหน้า ก่อน เรียกใช้ API จาก Frontend โดยมี Response ส่งรูปของ Attribute มาดังนี้ ดังรูปที่ 3.3.3.5.2.1

```

1 {
2   "page": 1,
3   "pageSize": 18,
4   "allPage": 2,
5   "allItems": 20,
6   "list": [
7     {
8       "itemId": 300073,
9       "name": "Web plant",
10      "description": "*ใบพืชญะนีดีคุณในบ้าน*"
11    }
12  ]
13}

```

รูปที่ 3.3.3.5.2.1 การส่งข้อมูลการเรียกสินค้ากลับในรูปแบบ Pagination List จาก /api/products

- Page คือลำดับหน้าที่ไป
- Page size คือจำนวนข้อมูลต่อหน้าที่รองรับข้อมูลทั้งหมด
- All Page คือจำนวนหน้าทั้งหมดที่มีข้อมูลแสดง
- All Item คือจำนวนข้อมูลทั้งหมดที่แสดง
- List และ Out stock คือข้อมูลที่แสดงผลออกตามลำดับ Page และจำนวน Page size ที่กำหนดไว้ ถ้าข้อมูลเป็นผลิตภัณฑ์แสดงผลออกแล้วหมดในคลังสินค้านั้นจะถูกคัดกรองบน Out stock แทน

5.3. 3A (Authentication, Authorization และ Accounting) [20]

5.3.1. Authentication มี 2 กรณีคือ

- เมื่อ User login เข้าสู่ระบบ จะสร้าง HTTP only cookie เพื่อเก็บเข้าใช้งานต่อบน Session cookie ได้ทันทีซึ่งทำให้ JavaScript ไม่สามารถใช้งาน HTTP cookie นอกจาก API service เท่านั้น ซึ่งข้อมูล Log ถูก Encrypt โดย JWT (Json web token) เป็น Third party ในการ Login ซึ่งสร้างทั้ง Access token และ Refresh token และสามารถนำออกได้ผ่าน Sign out ซึ่งสามารถรับ Proxy server ได้โดยทันที ผ่านการ Public URL
- เมื่อ Access token หมดอายุภายใน 1 ชั่วโมง Access token จะถูกลบโดยระบบ ระบบจะนำข้อมูลของ User ไป Refresh โดยใช้ Refresh token ภายใน 1 วันก่อน หมดอายุเพื่อสร้าง Access token อันใหม่ เมื่อ Refresh token หมดอายุก็จะบังคับให้ User เข้าไป Login ใหม่อีกรอบ พร้อมลบ Refresh token ด้วย

กระบวนการนี้เริ่มต้นจาก Middleware ที่มีตัวรองสำหรับการตรวจสอบสิทธิ์ โดยใช้ตัวกรองในการตรวจสอบ Endpoint เพื่อให้แน่ใจว่าผู้ใช้มีการยืนยันสิทธิ์และการใช้สิทธิ์ที่เหมาะสมเพื่อเข้าใช้งาน Router ได้ ตัวอย่างเช่น ในกรณีของ JwtAuth ที่มี Method ที่ถูกกำหนดค่าเพื่อรองรับผู้ใช้ที่ใช้ JWT ซึ่งทั้งสองกรณีคือ Access token และ Refresh token โดยทุก Request จะต้องใช้ Bearer token ผ่าน Access token ในการยืนยันตามที่แสดงในรูปที่ 3.3.3.5.3.1 เมื่อ Access Token หาข้อมูลผู้ใช้งานเจอแล้วจะกำหนดค่าใส่ในตัวแปร User ผ่าน Request เพื่อใช้ในการจัดการการใช้สิทธิ์ของบัญชีผู้ใช้งาน และเมื่อตรวจสอบ Token พบร่วมความผิดปกติ เช่น ผิดปกติหมดอายุ หรือ ลายเซ็น (Signature) ไม่ตรงกับ Token ดังกล่าว จะดำเนินการลบ Token นั้นทันที และบังคับให้เข้าหน้า Login ใหม่

```

16 exports.jwtAuth = (req, res, next) => {
17   // เอา token จาก headers or cookies
18   const jwtToken = "Bearer " + req.cookies.token;
19   // const jwtRefreshToken = req.headers.refresh || "Bearer " + req.cookies.refreshToken ;
20   // ตรวจสอบว่ามี token และต้องมีไม่ได้
21   if ([null, undefined].includes(req.cookies.token)) return res.status(401).json(errorRes("need login first", req.originalUrl))
22   try {
23     // ตรวจสอบ user ใน access token
24     let user = jwt.verify(jwtToken.substring(7), process.env.TOKEN_SECRET);
25     console.log(user)
26     // ตรวจสอบนี่ token มีการทํา format ของ user ถูกต้องไหม
27     if (user.email === undefined || user.role === undefined) {
28       | return res.status(401).json(errorRes("invalid token", req.originalUrl))
29     }
30     // ตัวเองจะ request ไปยัง user
31     req.user = user;
32     next();
33   } catch (err) {
34     // ถ้า access token ไม่ถูกต้องหรือหมดอายุ
35     return res.status(401).json(errorRes("Access token: " + err.message, req.originalUrl))
36   }
37 };

```

รูปที่ 3.3.3.5.3.1 Method JwtAuth() Authentication Filter

ดังนั้น ระบบการตรวจสอบสิทธิ์จึงมีวิธีการสำหรับการสร้าง Token และการเรียกใช้งานได้ผ่าน API ที่ใช้ในการตรวจสอบสิทธิ์ ซึ่งคือ /api/auth.js ดังที่แสดงในรูปที่ 3.3.3.5.3.2

```

8   // get token
9   const getToken = (user,date) => {
10   |   // สร้าง jwt token จาก user
11   |   return jwt.sign(user, process.env.TOKEN_SECRET, { expiresIn: date });
12 }

```

รูปที่ 3.3.3.5.3.2 Method ที่ใช้สร้าง token ด้วยข้อมูล (Body) ลายเซ็น (signature) และวัน
หมดอายุ

โดยตั้งค่าให้ตรวจสอบว่าถ้าผู้ใช้งานได้ Login และ ตรวจสอบ Password ได้อย่างถูกต้อง จะสร้าง Access token หมดอายุภายใน 1 ชั่วโมง และ Refresh token หมดอายุภายใน 24 ชั่วโมง อย่างไรก็ตาม เนื่องจากได้ติดตั้งตัวประมวลผลคุกกี้ (Cookie parser) และ เมื่อการยืนยันสิทธิ์ เสร็จแล้ว สามารถเก็บ Token ได้ผ่านคุกกี้ทันที และตั้งค่าให้เป็น Http-only เพื่อให้ JavaScript ไม่สามารถใช้ข้อมูลบน Cookie ดังกล่าวได้ และสามารถกันการโจมตีในรูปแบบ Cross site script attack (XSS attack) โดยต้องดำเนินการตั้งค่าในไฟล์ index.js และใน /api/auth.js ตามที่แสดงในรูป 3.3.3.5.3.3

```

75
76 // เก็บเมื่อ cookie ให้สู่พัฒนา backend สามารถใช้งานได้
77 const cookieConfig = {
78   maxAge: 24 * 60 * 60 * 1000,
79   httpOnly: true,
80   sameSite: 'Strict'
81   // secure: true
82 }
83 res.cookie("token", token, cookieConfig);
84 res.cookie("refreshToken", refreshtoken, cookieConfig);
85
86 res.status(200).json({
87   "id": getUser(token).id,
88   "firstname": getUser(token).firstname,
89   "email": getUser(token).email,
90   "role": getUser(token).role,
91   "token": token,
92   "refreshToken": refreshtoken
93 })
94 })
95

```

รูปที่ 3.3.3.5.3.3 การทำการเก็บ Cookie Parser และ Customize ด้วย HTTP-only Cookies บน /api/auth.js

ส่วน Password ใช้ Argon2 ซึ่งเป็น Slow hashing ที่ใช้ตามคำแนะนำของ OWASP Cheat sheet Recommendation [19] ดังรูป 3.3.3.5.3.4

```

const hashingConfig = { // based on OWASP cheat sheet recommendations (as of March, 2022)
  parallelism: 1,
  memoryCost: 64000, // 64 mb
  timeCost: 3 // number of iterations
}

// ตรวจสอบ password ที่ได้จาก mysql2 ว่าเป็น hash match กับ password ที่กรอกมาหรือไม่
if (!(await argon2.verify(user.password, password, hashingConfig))) {
  return res.status(401).json(errorRes("user email or password is invalid please login again", req.originalUrl))
}

```

รูปที่ 3.3.3.5.3.4 การ Match Password ด้วย Slow Hashing ของ Argon2 Algorithm

รหัสผ่านของ argon2 จะมีลักษณะดังนี้ ตามตัวอย่าง

\$argon2id\$v=19\$m=64000,t=3,p=1\$YFz2Pt/jwBN3xVI+au9ybw\$mu2wooC7N8HenZA+I4Qu97vhHeDOTIRIEMEuQ66hDng

กระบวนการของ Refresh token จะทำงานเมื่อ Access token หมดอายุ แต่ Refresh token ยังไม่หมดอายุ และสามารถตรวจสอบได้ ดังนั้น สามารถสร้าง Access token และ Refresh token ใหม่ได้ หาก Refresh token ยังไม่หมดอายุ เมื่อ Refresh token หมดอายุ User จะต้องเข้าสู่หน้า Login อีกครั้งเพื่อสร้าง Token ใหม่ที่ใช้ในการเข้าสู่ระบบตามที่แสดงในรูปที่ 3.3.3.5.5

โดยการใช้งาน Refresh token เป็นองค์ประกอบสำคัญในการบริหารจัดการการเข้าสู่ระบบและความปลอดภัยของระบบ

```

68 exports.refreshTokenAuth = (req, res, next) => {
69   // เช็ค refresh token เพื่อเช็คต่อ refresh ถ้าหาก access token จะทำการลบข้อมูลของ user ที่ให้ลง token ดัง
70   const jwtRefreshToken = "Bearer " + req.cookies.refreshToken;
71
72   try {
73     if (req.cookies.refreshToken === undefined) {
74       // ตรวจสอบ user ใน access token
75       // console.log(jwtToken.substring(7))
76       let user = jwt.verify(jwtRefreshToken.substring(7), process.env.TOKEN_SECRET);
77       // console.log(user)
78       // ตรวจสอบใน token มีการฟิต format ของ user ถูกต้องไหม
79       if (user.email === undefined || user.role === undefined) {
80         return res.status(401).json(errorRes("invalid token", req.originalUrl))
81       }
82       // ถ้าจะ request ไปยัง user
83       req.user = user;
84     }
85   next();
86 } catch (err) {
87   // ถ้า refresh token ไม่ถูกต้องหรือหมดอายุ
88   // clear session cookie
89   res.clearCookie("token")
90   res.clearCookie("refreshToken")
91   return res.status(403).json(errorRes("Refresh token: " + err.message, req.originalUrl))
92 }
93 };

```

รูปที่ 3.3.3.5.3.5 Method refreshTokenAuth() Authentication Filter

5.3.2. Authorization ที่เกิดการการตักเพื่อนไขมี 2 กรณีคือ

- ตรวจจับจาก Role-based เมื่อ User ไม่มี Role ที่กำหนดไว้ เช่น User ไม่มีสิทธิที่จะสร้าง Product โดยที่จะต้องให้เจ้าของร้านสร้างเท่านั้น
- ตรวจจับจาก User ที่ไปเรียก API เพื่อกระทำการบางอย่างจาก Other user เช่น User ไม่มีสิทธิที่จะแก้ไขข้อมูลของคนอื่นๆ

กระบวนการการให้สิทธิ (Authorization) มาจากผู้ใช้ที่ยืนยันสิทธิ (Authentication) เข้าใช้งานไปแล้วพูกเข้าจะมีบทบาทที่แตกต่างกัน เช่น เป็นผู้จัดหาสินค้าหรือผู้ใช้งานตามปกติ ซึ่ง Authorization เป็นส่วนสำคัญที่มีตัวอย่าง เช่น Method ที่ใช้ในการตรวจสอบบทบาท (Verify role) ดังภาพที่ 3.3.3.5.3.6

```

exports.verifyRole = (...roles) => {
  return (req, res, next) => {
    // เช็ค role จาก header หรือ cookie
    const jwtToken = "Bearer " + req.cookies.token;
    let reqRole = getUser(jwtToken.substring(7)).role;

    // // ถ้าไม่มี role จะสามารถเข้าระบบได้
    let result
    if (reqRole) {
      result = [...roles].includes(reqRole);
    } else {
      result = true
    }

    // ถ้า role ไม่ใช่ user, admin_it และ admin_pr จะไม่มีสิทธิ์สามารถเข้าระบบในส่วนนี้
    if (!result) return res.status(403).json(errorRes("the role is not allowed to use", req.originalUrl));
    next();
  }
}

```

รูปที่ 3.3.3.5.3.6 Method verifyRole Authentication Filter

ทำการนำมาผูกกันใน Router เช่นที่ไฟล์ routes/api/user.js ใน Method get จะต้องกำหนดว่าผู้ใช้งาน (User) สามารถเป็นบทบาท (Role) ได้เพียง Admin เท่านั้น เมื่อทำการเข้าสู่ระบบ (JWT Auth) และจะต้องมีบทบาทเป็น Admin เท่านั้นเพื่อที่จะทำการเข้าถึง (Verify Role) ตามภาพขั้นตอนที่ 3.3.3.5.3.7

```
router.get('/', JwtAuth, verifyRole(ROLE.Admin), async (req, res, next) => {
  // let sorting = req.query.sort == 'desc' ? 'desc' : 'asc'
  // console.log(req.query.name)
  let page = Number(req.query.page)
  let limit = Number(req.query.limit)
  // let varPage = page > 0 ? (page - 1) * limit : 0
  // let varLimit = (limit <= 0 || isNaN(limit)) ? 0 : limit >= 10 ? 10 : limit
  // let count_user = await prisma.accounts.count()
  let filter_u = await prisma.accounts.findMany({
```

รูปที่ 3.3.3.5.3.7 การเรียกใช้ Method Verify Role ของ User list ด้วยการตรวจสอบ Role เป็น Admin เท่านั้น

5.4. Data compression and Limit per request

การบีบอัดข้อมูล (Data compression) เป็นกระบวนการที่ใช้ในการลดขนาดของข้อมูลโดยไม่สูญเสียข้อมูลเดิม โดยมีวัตถุประสงค์เพื่อเพิ่มความเร็วในการส่งข้อมูลผ่านเครือข่ายหรือเก็บข้อมูลในพื้นที่จัดเก็บที่มีขนาดจำกัด คำว่า "บีบอัด" หมายความว่าการลดขนาด ซึ่งสามารถทำได้โดยใช้วิธีการต่างๆ อาทิเช่นการลบข้อมูลที่ไม่จำเป็น การใช้วิธีการเข้ารหัสที่มีประสิทธิภาพ

การลดขนาดนี้สามารถช่วยลดเวลาในการส่งข้อมูลผ่านเครือข่ายให้เกิดความรวดเร็วขึ้นได้เนื่องจากจำนวนข้อมูลที่ต้องส่งไปในแต่ละครั้งจะลดลง เมื่อข้อมูลมีขนาดเล็กลง ดังนั้นการบีบอัดข้อมูลเป็นกระบวนการที่สำคัญในการจัดการข้อมูลในระบบคอมพิวเตอร์และเครือข่ายสารสนเทศ

Limit request handle [22] คือการจำกัด Request ไปยัง server ใน 1 หน่วยเวลา เช่นระบบ Browser request ไปยัง Server ด้วยจำนวน 20000 ครั้งต่อนาที แสดงว่า Server ควรรองรับได้สูงสุด 20000 ครั้งต่อนาที ถ้าหากเกินกว่า 20000 ครั้งต่อนาที จะส่ง HTTP status 429 Too many request คือระบบบริการข้อมูลจะรองรับการ Request เกินกว่าที่ Server request ไว้ไม่ได้ จากรูปที่ 3.3.3.5.4.1

```
// Set up rate limiter: maximum of twenty requests per minute
const RateLimit = require("express-rate-limit");
// const limiter = RateLimit({
//   windowMs: 1 * 30 * 1000, // 30 second
//   max: 20000,
// });

// Apply rate limiter to all requests
app.use(limiter);

// Init Middleware
app.use(logger)
app.use(cors(corsOptions))
app.use(compression())
```

รูปที่ 3.3.3.5.4.1 Config Limit Request Per Time บน /index.js

เพราการ Request ที่มีจำนวนมากเกินไปจะทำให้ Server ทำงานหนักและเกิดการ Down ของระบบในที่สุด เมื่อใช้ Apache benchmark [23] ทดสอบยิ่ง Backend request บน VM คณะ ผลลัพธ์ที่ได้คือ

เมื่อ Request 20000 request ไปที่
cp23pl4.sit.kmutt.ac.th:5000/api/products ได้ผลลัพธ์ตามนี้

- กรณีที่ 1 บน cp23pl4.sit.kmutt.ac.th:5000/api/products ไม่ได้ต่อ Authentication ออกแบบรูปที่ 3.3.3.5.4.2

```

Finished 20000 requests

Server Software: cp23pl4.sit.kmutt.ac.th
Server Hostname: cp23pl4.sit.kmutt.ac.th
Server Port: 5000

Document Path: /api/products
Document Length: 95 bytes

Concurrency Level: 10
Time taken for tests: 13.918 seconds
Complete requests: 20000
Failed requests: 0
Non-2xx responses: 20000
Total transferred: 10669892 bytes
HTML transferred: 1900000 bytes
Requests per second: 1436.99 [#/sec] (mean)
Time per request: 6.959 [ms] (mean)
Time per request: 0.696 [ms] (mean, across all concurrent requests)
Transfer rate: 748.66 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:      0    0   0.1     0     6
Processing:   1    7   3.1     6    47
Waiting:     1    7   3.1     6    46
Total:        1    7   3.1     6    47

Percentage of the requests served within a certain time (ms)
 50%       6
 66%       6
 75%       7
 80%       7
 90%       9
 95%      12
 98%      16
 99%      21
100%      47 (longest request)

```

รูปที่ 3.3.3.5.4.2 ผลลัพธ์ของกรณีที่ไม่ได้ต่อ Authentication ด้วยการยิง Apache Benchmark Request จำนวน 20000 Request

- กรณีที่ 2 บน cp23pl4.sit.kmutt.ac.th:5000/api/products ทดสอบ Authentication ออกมายังสามารถ List product โดยไม่อาศัย Filtering and paging ได้จำนวน 6 rows ตามรูปที่ 3.3.3.5.4.3

```

Server Software:                   cp23pl4.sit.kmutt.ac.th
Server Hostname:                 cp23pl4.sit.kmutt.ac.th
Server Port:                     5000

Document Path:                  /api/products
Document Length:                1377 bytes

Concurrency Level:              10
Time taken for tests:          322.252 seconds
Complete requests:              20000
Failed requests:                0
Total transferred:              36180000 bytes
HTML transferred:               27540000 bytes
Requests per second:            62.06 [#/sec] (mean)
Time per request:               161.126 [ms] (mean)
Time per request:               16.113 [ms] (mean, across all concurrent requests)
Transfer rate:                  109.64 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    0    0.1     0      1
Processing:    10   161   33.8   155    335
Waiting:       8   158   33.6   153    325
Total:         10   161   33.8   156    335

Percentage of the requests served within a certain time (ms)
  50%    156
  66%    167
  75%    175
  80%    181
  90%    202
  95%    226
  98%    258
  99%    273
100%    335 (longest request)

```

รูปที่ 3.3.3.5.4.3 ผลลัพธ์ของกรณีที่ทดสอบ Authentication และ List product โดยไม่อาศัย Filtering and paging ได้จำนวน 6 rows ด้วยการยิง Request ด้วย Apache Benchmark

- พบว่าใน กรณีที่ 1 ถ้าใส่ Authentication จะไป Block การเรียกข้อมูลสินค้า ทำให้ใช้ระยะเวลา Request ลดลง และอัตราความเร็วในการ Request สูงขึ้น
- พบว่าใน กรณีที่ 2 ถ้าเรียกข้อมูลสินค้า แบบ ไม่ใส่ Authentication ด้วยจำนวน 6 ชิ้น ทำให้ใช้ระยะเวลา Request เพิ่มขึ้น และอัตราความเร็วในการ Request ลดลง อาจจะส่งผลต่อการรองรับจำนวนผู้ใช้งานในจำนวนมากได้ที่ทำให้ต้องอาศัยต้นทุนการรองรับ Server ขนาดที่มีขนาดใหญ่ขึ้นในอนาคต

3.3.4 Machine Learning

ในโครงการได้ใช้ Recommender system [16] เพื่อสร้างระบบเชิงแนะนำ โดยใช้เทคโนโลยี AI โดยเฉพาะ Machine Learning [13] เป็นส่วนหนึ่ง ซึ่งช่วยให้ระบบสามารถแนะนำสินค้าตามความต้องการของลูกค้าได้

ในโครงการได้ใช้เทคนิค Content-based filtering [32] เพื่อสร้าง Model ในการแนะนำสินค้า โดยวิธีการนี้คือการคัดเลือกสินค้าโดยอิงตามลักษณะของสินค้านั้นๆ โดยการใช้ค่า Weight ที่ได้จากพารามิเตอร์ต่างๆ ซึ่งมีอิทธิพลต่อการเลือกสินค้า กระบวนการนี้ถูกอ้างอิงจาก [27] และประกอบด้วยขั้นตอนต่างๆ ดังนี้

- สร้าง Weight จาก Dataset ที่กำหนดคือ item_events จาก Database ดังรูป

3.3.4.1

```
183      let pos = await prisma.itemEvents.findMany()
184      if (isRecommend === "true") {
185          let event = await prisma.item_events.findMany({ include: { items: true } })
186          event = event.map(event => {
187              event.itemPrice = event.items.minPrice
188              event.itemRating = event.items.totalRating
189              event.itemType = event.items.type
190              delete event["items"]
191              return event
192          })
193
194          let category = []
195          let weight = {}
196          let mockUser = []
197          // user case
198          if (req.user !== undefined) {
199              // list latest user activities on event
200              mockUser = await prisma.item_events.findMany({
201                  where: { userId: req.user.id },
202                  include: { items: true },
203                  orderBy: { timestamp: "desc" },
204                  take: 6,
205                  skip: 1
206              })
207
208              // guest case
209          } else {
210              // list latest user activities on event
211              mockUser = await prisma.item_events.findMany({
212                  include: { items: true },
213                  orderBy: { timestamp: "desc" },
214                  take: 6, // note this value can be change when user capacity is increase.
215                  skip: 1
216              })
217          }
218      }
```

รูปที่ 3.3.4.1 การนำ Item Event เอามา Mockup เพื่อทำ Dataset

กรณีของผู้ใช้งานเข้าสู่ระบบแล้วจะต้องเรียกรายการสินค้าตามพฤติกรรมที่ผู้ใช้ทำอยู่ เช่น การดูสินค้า การเพิ่มสินค้าบันทะกร้า และการชำระเงิน ตามสินค้าที่พวกรเข้าซื้อ มากมาที่สุด 5 ข้อมูล ย้อนหลัง ยกเว้นสินค้าที่พวกรเข้าดูก่อนหน้านั้นแล้ว เพื่อป้องกันการแนะนำสินค้าซ้ำและกรณีของผู้มาเยือนจะเรียกสินค้าที่พวกรเขานำไป ตอนที่พวกรเขานำสินค้าโดยภาพรวมของระบบ ซึ่งสามารถนำไปทำเป็นผู้ใช้จำลอง (Mock user) เพื่อที่จะนำไปฝึกฝน (Train) ให้สามารถแปลงเป็นค่าน้ำหนัก (Weight) ได้

Note: Train dataset ใน Recommender system นี้ยังไม่มี เพราะโครงงานขั้นนี้เน้นไปที่การนำ Recommender system การทดลองบน Web ecommerce ในการแนะนำสินค้าที่ลูกค้าสนใจจริงๆ ซึ่งได้ทดสอบผ่าน Dataset จำนวน 200,000 Records ไว้แล้วและทดลองคำนวณค่า Weight เพื่อหาค่า Scoring ที่เหมาะสมต่อการแนะนำสินค้า

2. สร้าง Weight จาก Dataset ที่กำหนดคือ item_events จาก Database ดังรูป

3.3.4.2

```
// join mockUser to item data
mockUser = mockUser.map(event => {
  // console.log(event)
  event.itemType = event.items.type
  event.itemPrice = event.items.minPrice
  event.itemRating = event.items.totalRating
  event.itemEvent = (event.itemEvent === ITEMEVENT.View ? 0.5 : event.itemEvent === ITEMEVENT.ATC ? 0.75 : 1)
  delete event["items"]
  return event
})

// map latest category when user see
category = mockUser.map(event => event.itemType)
// console.log(category)

// average weight before content based filtering
avg_event = mockUser.reduce((pre, cur) => pre + Number(cur.itemEvent), 0) / (mockUser.length > 5 ? 5 : mockUser.length)
avg_rating = mockUser.reduce((pre, cur) => pre + Number(cur.itemRating), 0) / (mockUser.length > 5 ? 5 : mockUser.length)
avg_price = mockUser.reduce((pre, cur) => pre + Number(cur.itemPrice) / 100, 0) / (mockUser.length > 5 ? 5 : mockUser.length)
// console.log(avg_event)
// console.log(avg_rating)
// console.log(avg_price)

weight = {
  totalRating: avg_rating,
  itemEvent: avg_event,
  itemPrice: avg_price,
}
// console.log(weight)
// console.log(mockUser)
topItem = getTopItems(event, category, weight, pds.length)

topItem.forEach(item => {
  filter_pd.push(pds.filter(pd => item.itemId === pd.itemId)[0])
})
// console.log(filter_pd)
```

รูปที่ 3.3.4.2 การปรับ MockUp เพื่อสร้าง Initial Weight ที่ใช้ในการคำนวณ

จากรูปได้นำข้อมูลเกี่ยวกับผู้ใช้งาน, รหัสสินค้า, ประเภทของสินค้า, ราคาสินค้าต่อสุด, คะแนนรวมของสินค้า, และราคาเฉลี่ยของสินค้า เพื่อสร้าง MockUser โดยปรับเปลี่ยนค่าต่างๆ ตามพฤติกรรมของลูกค้าที่สนใจ ซึ่งข้อมูลเหล่านี้จะถูกนำไปใช้ในฟีเจอร์ที่เอามาหาค่าเฉลี่ย โดยการหาราคาของสินค้าด้วย 100 เพื่อให้ได้ค่าเฉลี่ย จากนั้นใช้ข้อมูลที่ได้นี้ในการสร้างกราฟแจกแจงปกติ เพื่อหาค่ามาตรฐานเพื่อใช้เป็นน้ำหนักที่เหมาะสม ก่อนจะใช้ Method เพื่อแนะนำผลิตภัณฑ์ที่ดีที่สุด โดยใช้เหตุการณ์และหมวดหมู่ที่ผู้ใช้งานใจ, น้ำหนัก และจำนวนผลิตภัณฑ์ทั้งหมดที่มีอยู่ในรายการที่สร้างขึ้นใหม่

3. นำ Weight คำนวณจากการหาค่า Parameter จากสูตรคำนวณ

$$\text{weight value} = \frac{\text{sum of weight parameter}}{\text{max weight}}$$

นำค่า Weight ได้จากสัดส่วนเอามาอิงตาม Weight parameter ที่ใช้ค่าน้ำหนักใส่ไป ซึ่งรวมถึงการจัดการ Category rating และ Price จากสินค้าที่ลูกค้ามีความสนใจมากที่สุด ลงไปในสูตรคำนวณนี้

$$\text{weight ratio} = \left(\frac{\text{weight value}}{\text{weight parameter total}} \right) * 100$$

เพื่อคำนวณใน Method หาค่า Weight โดยอาศัยค่าสถิติเข้ามาช่วยปรับค่า Weight ให้ตรงกับกลุ่มเป้าหมายที่นำเสนอ จากรูปที่ 3.3.4.3

```

66 const maxWeight = (weights) => Math.max(...Object.values(weights));
67 const meanP = (arr) => arr.reduce((pre, cur) => pre + Number(cur), 0) / arr.length // arithmetic mean
68 const sd = (arr, mean) => Math.sqrt(arr.reduce((pre, cur) => pre + Math.pow(Number(cur) - mean, 2)) / arr.length, 0) // standard deviation
69 const normalProp = (x, mean, sd) => (1 / (Math.sqrt(2 * Math.PI) * sd)) * Math.exp(-Math.pow(x - mean, 2) / (2 * sd)) // area of under normal distribution
70
71 const calculateWeight = (userData, mean, sdp, weights, category = []) => {
72   let maxWeightValue = maxWeight(weights)
73   const { itemType, itemRating, itemPrice, itemEvent } = userData;
74   // console.log(category)
75   // console.log(weights.totalRating)
76   // console.log(itemRating)
77   // console.log(itemRating > weights.totalRating)
78   // console.log(normalProp(itemPrice / 100, maxWeightValue, sdp))
79   const weight = itemRating / maxWeightValue +
80     ((category.includes(itemType)) ? 1 : -1) / maxWeightValue +
81     (normalProp(itemPrice / 100, mean, sdp)) / maxWeightValue +
82     (itemEvent == "view" ? 0.5 : itemEvent == "adtc" ? 0.75 : 1) / maxWeightValue
83   const weightType = category.filter(cate => cate === itemType).length;
84   // console.log(category)
85   return (weight / (weightType + weights.totalRating + weights.itemPrice + weights.itemEvent)) * 100;
86 };

```

รูปที่ 3.3.4.3 Method ที่ใช้ในการคำนวณหาค่า Single Weight

ส่วนสุดท้ายนำค่า Weight ratio จัดกลุ่มอิงตามสินค้า ทำให้เป็น Content based filtering หลังจากนั้น หาค่าผลรวมของน้ำหนักแต่ละกลุ่มโดยเฉลี่ย (Weight) โดยใช้ค่า Weight ที่ได้จัดกลุ่ม อ้างอิงตามสินค้าเพื่อหาค่าผลรวม และหารด้วยจำนวนสินค้าที่จัดกลุ่มไว้แล้วทำให้ได้ค่าคะแนน เปอร์เซ็นต์ที่ Similarity หากที่สุดต่อ User คนนั้น ก่อนที่จะเรียงลำดับคะแนนดังกล่าวจากมากไปยังน้อย จากรูปที่ 3.3.4.4

```

32 // Get 10 results
33
34 const getTopItems = (data, category, weights, top_n = 10) => {
35   let itemPriceList = []
36   data.forEach(item => {
37     itemPriceList.push(item.itemPrice)
38   });
39   const mean = meanP(itemPriceList)
40   // console.log(mean)
41   const sdp = sd(itemPriceList, mean)
42   // console.log(sdp)
43   return _.chain(data)
44     .map((item) => [item.itemId, calculateWeight(item, mean, sdp, weights, category)])
45     .groupBy((item) => item[0])
46     .map((items) => {
47       const totalWeights = _.sumBy(items, (item) => item[1]);
48       const avgWeight = totalWeights / items.length;
49       return { itemId: items[0][0], score: avgWeight }
50     })
51     .sortBy('score')
52     .reverse()
53     .take(top_n)
54     .value()
55 }

```

รูปที่ 3.3.4.4 Method เรียก Top N สินค้าที่แนะนำสินค้าที่น่าสนใจ

4. Limitation:

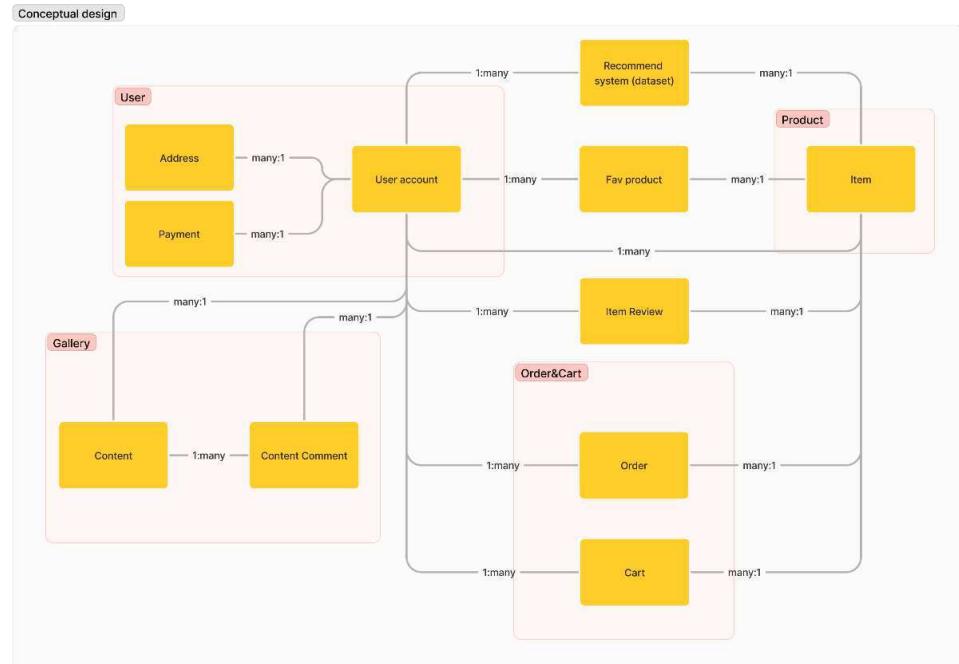
รูปแบบในการทำ Train และ Test dataset นั้นไม่มีความเป็นส่วนตัว (Personalize) เพราะว่าการบันทึกเหตุการณ์บนข้อมูล (Log event) ที่ใช้งาน เช่น บทบาทผู้มาเยือนจะอิงตามสินค้าของผู้ใช้งานทั้งหมดที่เข้ามาเพื่อแนะนำสินค้าที่ลูกค้าสนใจ ในอนาคตทำให้สามารถมีบทบาทผู้ที่มาเยือนมีความเป็นส่วนตัวมากขึ้นในการแนะนำสินค้า

กรณีที่เป็นบัญชีผู้ใช้งานและระบบตอบนี้ยังไม่สนับสนุนการคัดกรองโดยใช้ความคล้ายคลึงกันของผู้ใช้งาน (Collaborative filtering) คือการเรียกสินค้าแบบอิงตามผู้ใช้งานที่พวกราชานใจ สินค้าประเภทเดียวกัน ซึ่งก็หมายความว่า บางสินค้าอาจจะไม่ได้ตอบโจทย์กับการใช้งานของลูกค้า บางคนได้ ซึ่งถ้าหากทำให้เป็น Hybrid ก็สามารถจัดการกระบวนการที่ซับซ้อนในการแนะนำสินค้าที่ตรงจุดกับลูกค้าต้องการจากความน่าสนใจของพวกราชานใจมากขึ้น ก็จะทำให้ลูกค้าเข้าไปเลือกสินค้าที่พวกราชานใจได้

อีกรายนีคือ ระบบดังกล่าวยังไม่มีเกณฑ์การประเมินผลที่แน่นอน เช่น ไม่มีค่า Accuracy แต่มี Precision และ Recall จาก Mean Average Precision at K (MAP@K) และ Mean Average Recall at K (MAR@K) และผลต่างจากค่า Error เช่น MSE และ R-square ที่สามารถวัดค่าความเป็น Similarity ได้เท่านั้น [31]

3.3.5 Database

1. ภาพรวม Conceptual Design ของ Leafy ในรูปที่ 3.3.5.1



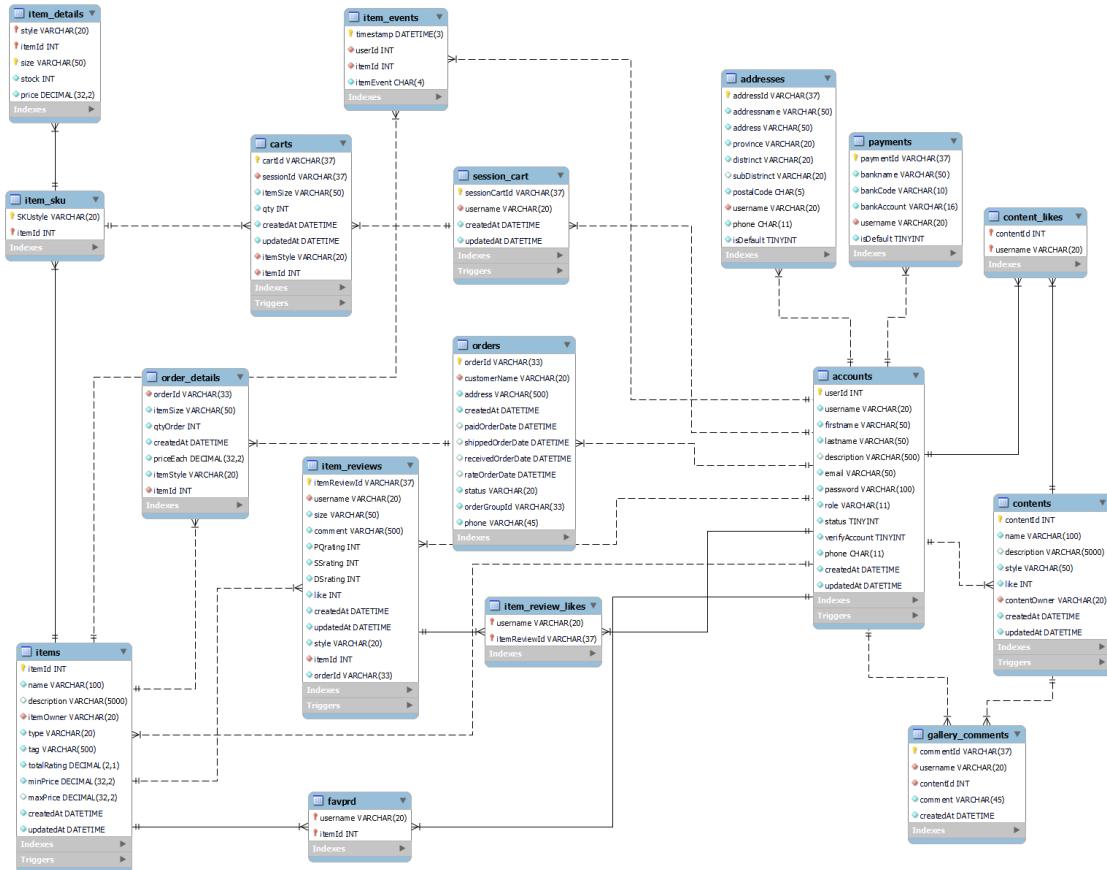
รูปที่ 3.3.5.1 ภาพรวม Conceptual Design ของ Leafy

โดยระบบ Leafy นั้นใช้ข้อมูลของผู้ใช้เป็นจุดศูนย์กลางเพื่อที่จะไปเชื่อมต่อกับข้อมูลอื่นได้ หลายส่วน เช่น สำหรับ Role user จะไม่สามารถสร้าง Product ได้ นอกจากจะเป็น Role supplier ซึ่งจะสามารถสร้างหลาย Product ได้ ซึ่ง User สามารถมีที่อยู่สำหรับจัดส่งได้ หากเลยที่และสามารถมี หลายบัญชีที่ใช้ในการจ่ายเงินได้ User สามารถกด Product ที่เข้าถูกใจได้แล้วข้อมูลจะจัดเก็บแยก สำหรับแต่ละ Account ดังนั้นจึงมีหลาย User ต่อที่ไป Map กับหลาย Product ได้ เมื่อคุณติดตั้ง ของ User กับ Product แต่ละชิ้น เราสามารถนำ Dataset ที่เป็น Test set เอาไปสร้าง Recommend product ที่เขานำมาใช้ จึงต้องมีการ Map ระหว่าง User กับ Product เช่นกัน กับ การ Review ของ User ตอนที่ User ซื้อสินค้าแล้ว

เมื่อพวกราคาอย่างซื้อสินค้า เขามาสามารถเพิ่มสินค้าใน Cart และจ่ายบน Order ของเข้า ซึ่ง มีหลาย Order ต้อง Map กับหลาย Cart และ Order ซึ่งในบางกรณี ลูกค้าอาจจะซื้อสินค้าทันที โดยไม่ผ่านตະกร้าทำให้ต้องสร้าง Order ทันทีโดยไม่ผ่านตະกร้า และระบบรองรับการออกแบบ แนวคิดในการออกแบบจัดสวน ให้ผู้ใช้ทุกคนสามารถเขียนแนวคิดในการออกแบบจัดสวนรวมถึง Comment ของการออกแบบจัดสวนได้ จึงทำให้ User สามารถสร้างเนื้อหา (Gallery) สะดวกได้

หlays เนื้อหาร่วมถึง Comment ในแต่ละความคิดได้หลาย Comment และ User 1 คนสามารถ Comment ได้หลายเนื้อหาให้กับเนื้อหาของตัวเองหรือคนอื่นๆได้

2. ภาพรวม Logical design ของระบบ Leafy - DBMS: MySQL version 8.3.0-1.el8 ในรูปที่ 3.3.5.2 [28] และ [29]



รูปที่ 3.3.5.2 Entity Relational Diagram ของ Leafy บน MySQL

3. Data dictionaries ดังแสดงในตารางที่ 3.3.5.3.1-3.3.5.3.17

ตารางที่ 3.3.5.3.1 accounts attributes

No	Column Name	Column Description	Data Type	Default Value	Data Example	Constraint	Referenced Table
1	userId	รหัสของผู้ใช้งาน	INT	AUTO INCREMENT VALUE		Primary Key	
2	username	ชื่อแสดงผลของผู้ใช้งาน	VARCHAR(20)			UNIQUE, NOT NULL	
3	firstname	ชื่อจริงของผู้ใช้งาน	VARCHAR(50)			UNIQUE(firstname, lastname), NOT NULL	
4	lastname	นามสกุลของผู้ใช้งาน	VARCHAR(50)			UNIQUE(firstname, lastname), NOT NULL	
5	description	ลายละเอียดที่ผู้ใช้งานเขียนไว้	VARCHAR(500)	NULL		NULL	
6	email	อีเมลของผู้ใช้งาน	VARCHAR(50)		leafy@example.com	UNIQUE, NOT NULL	
7	password	รหัสผ่านของผู้ใช้งานที่ถูก Hash เอาไว้โดย Argon2 algorithm	VARCHAR(100)		\$argon2id\$v=19\$m=64000,t=3,p=1\$YFz2Pt/jwBN3...	NOT NULL	
8	role	บทบาทที่ User เป็น คือ User, Supplier และ Admin	VARCHAR(11)	user	ENUM(user, supplier, admin)	NOT NULL	
9	status	ระบุว่า User คนนี้ใช้งานบนระบบนี้ได้หรือไม่	BOOLEAN	TRUE		NOT NULL	
10	verifyAccount	ระบุว่า User คนนี้ยืนยันตัวตนหรือยัง	BOOLEAN	FALSE		NOT NULL	

ตารางที่ 3.3.5.3.1 accounts attributes (ต่อ)

No	Column Name	Column Description	Data Type	Default Value	Data Example	Constraint	Referenced Table
11	phone	เบอร์โทรศัพท์ของ User ที่ใช้ในการติดต่อ	CHAR(11)		0901001000	UNIQUE, NOT NULL	
12	createdAt	ระบุว่า User คนนี้ สร้างในเวลานี้เท่าไร	DATETIME	CURRENT_TIME STAMP		NOT NULL	
13	updatedAt	ระบุว่า User คนนี้ แก้ไขข้อมูลของเขา ในเวลานี้เท่าไร	DATETIME	CURRENT_TIME STAMP ON UPDATE CURRENT_TIME STAMP		NOT NULL	

ตารางที่ 3.3.5.3.2 addresses attributes

No	Column Name	Column Description	Data Type	Default Value	Data Example	Constraint	Referenced Table
1	addressId	รหัสของสถานที่ของ ผู้ใช้งาน	VARCHAR(37)		[username]- [รหัส 16 หลัก]	Primary Key	
2	username	ชื่อแสดงผลของ ผู้ใช้งาน	VARCHAR(20)			Foreign Key, NOT NULL	accounts
3	addressname	ชื่อแสดง Address ของผู้ใช้งาน	VARCHAR(50)			NOT NULL	
4	address	ที่อยู่จริงของผู้ใช้งาน	VARCHAR(50)			NOT NULL	
5	province	จังหวัดของผู้ใช้งาน	VARCHAR(20)			NOT NULL	
6	district	อำเภอของผู้ใช้งาน	VARCHAR(20)			NOT NULL	
7	subDistrict	ตำบลของผู้ใช้งาน	VARCHAR(20)	NULL		NULL	
8	postalCode	รหัสที่อยู่ของ ไปรษณีย์ของ ผู้ใช้งาน	CHAR(5)		10140	NOT NULL	
9	phone	เบอร์โทรศัพท์ของ สถานที่ User ใช้ในการติดต่อ	CHAR(11)		0901001000	NOT NULL	

ตารางที่ 3.3.5.3.2 addresses attributes (ต่อ)

No	Column Name	Column Description	Data Type	Default Value	Data Example	Constraint	Referenced Table
10	isDefault	ระบุว่าสถานที่เป็นสถานที่หลักที่ User ติดต่อได้	BOOLEAN	FALSE (เป็น True เมื่อ user address ถูกสร้างครั้งแรกบน account คนนั้น ซึ่งลบไม่ได้)		NOT NULL	

ตารางที่ 3.3.5.3.3 payments attributes

No	Column Name	Column Description	Data Type	Default Value	Data Example	Constraint	Referenced Table
1	paymentId	รหัสของบัญชีจ่ายเงินของผู้ใช้งาน	VARCHAR(37)		[username]-[รหัส 16 หลัก]	Primary Key	
2	username	ชื่อแสดงผลของผู้ใช้งาน	VARCHAR(20)			Foreign Key, NOT NULL	accounts
3	bankname	ชื่อแสดงบัญชีจ่ายเงินของผู้ใช้งาน	VARCHAR(50)			NOT NULL	
4	bankCode	ชื่อแสดงที่อยู่ของบัญชีของผู้ใช้งาน	VARCHAR(10)		BBL	NOT NULL	
5	bankAccount	เลขที่บัญชีจ่ายเงินของผู้ใช้งาน	VARCHAR(16)		0001000203	NOT NULL	
6	isDefault	ระบุว่าบัญชีจ่ายเงินนี้เป็นบัญชีหลักที่ User ติดต่อได้	BOOLEAN	FALSE (เป็น True เมื่อ User address ถูกสร้างครั้งแรกบน Account คนนั้น ซึ่งลบไม่ได้)		NOT NULL	

ตารางที่ 3.3.5.3.4 items attributes

No	Column Name	Column Description	Data Type	Default Value	Data Example	Constraint	Referenced Table
1	itemId	รหัสของสินค้า	INT	AUTO INCREMENT VALUE		Primary Key	
2	name	ชื่อสินค้า	VARCHAR(100)			NOT NULL	
3	description	รายละเอียดของสินค้า	VARCHAR(5000)	NULL		NULL	
4	itemOwner	ชื่อเจ้าของสินค้า	VARCHAR(20)			Foreign Key, NOT NULL	accounts
5	type	ชนิดสินค้า	VARCHAR(20)		ENUM(plant, flower, cactus, seed, equirement, meterial)	NULL	
6	tag	คำที่เป็นจุดเด่นของสินค้า	VARCHAR(500)		TagA, TagB, TagC	UNIQUE, NOT NULL	
7	totalRating	คะแนนโดยเฉลี่ยรวมของสินค้านั้น	DECIMAL(2,1)	0.00		NOT NULL	
8	minPrice	ราคาต่ำสุดในทุกSKU Item	DECIMAL(32,2)	0.00	15.00	NOT NULL	
9	maxPrice	ราคาสูงสุดในทุกSKU Item (ถ้า SKU Item นั้นมีรายละเอียดแค่อันเดียว ทำให้ค่าเป็น NULL)	DECIMAL(32,2)	NULL	100.00 (ห้ามต่ำกว่า 15 บาท)	NOT NULL	
10	createdAt	ระบุว่า Item ขึ้นนี้สร้างในเวลา哪ที่	DATETIME	CURRENT_TIME STAMP		NOT NULL	
11	updatedAt	ระบุว่า Item ขึ้นนี้แก้ไขข้อมูลของเขานะในเวลา哪ที่	DATETIME	CURRENT_TIME STAMP ON UPDATE CURRENT_TIME STAMP		NOT NULL	

ตารางที่ 3.3.5.3.5 item_sku attributes

No	Column Name	Column Description	Data Type	Default Value	Data Example	Constraint	Referenced Table
1	itemId	รหัสของสินค้า	INT			Composite Key, Foreign Key	items
2	SKUstyle	รหัสของ SKU สินค้า	VARCHAR(20)			Composite key	

ตารางที่ 3.3.5.3.6 item_details attributes

No	Column Name	Column Description	Data Type	Default Value	Data Example	Constraint	Referenced Table
1	itemId	รหัสของสินค้า	INT			Composite Key, Foreign Key	item_sku
2	style	รหัสของ SKU สินค้า	VARCHAR(20)			Composite Key, Foreign Key	item_sku
3	size	ชื่อความหลากหลายของสินค้า	VARCHAR(50)	No		Composite Key	
4	stock	สินค้าที่มีอยู่	INT	0.00		NOT NULL	
5	price	ราคาสินค้าแต่ละชิ้น	DECIMAL(32,2)	0.00		NOT NULL	

ตารางที่ 3.3.5.3.7 favprd attributes

No	Column Name	Column Description	Data Type	Default Value	Data Example	Constraint	Referenced Table
1	itemId	รหัสของสินค้า	INT			Composite Key, Foreign Key	items
2	username	ชื่อแสดงผลของผู้ใช้งาน	VARCHAR(20)			Composite Key, Foreign Key	accounts

ตารางที่ 3.3.5.3.8 item_events attributes

No	Column Name	Column Description	Data Type	Default Value	Data Example	Constraint	Referenced Table
1	timestamp	เวลา Event ที่ User เกิดขึ้น	DATETIME	CURRENT_TIMESTAMP(3)		Primary Key	
2	itemId	รหัสของสินค้า	INT			Foreign Key	items
3	username	ชื่อแสดงผลของผู้ใช้งาน	VARCHAR(20)			Foreign Key	accounts
4	itemEvent	พฤติกรรมของลูกค้า	CHAR(4)		ENUM(view, adtc, paid)	NOT NULL	

ตารางที่ 3.3.5.3.9 session_cart attributes

No	Column Name	Column Description	Data Type	Default Value	Data Example	Constraint	Referenced Table
1	session CartId	รหัส Session ที่ลูกค้าสร้างขึ้นเพื่อเก็บสินค้าในตะกร้า	VARCHAR(37)		[itemOwner]-[รหัส 16 หลัก]	Primary Key	
2	username	ชื่อแสดงผลของผู้ใช้งาน	VARCHAR(20)			Foreign Key	accounts
3	createdAt	ระบุว่า Session Cart นี้สร้างในเวลา นี้เท่าไร	DATETIME	CURRENT_TIMESTAMP		NOT NULL	
4	updatedAt	ระบุว่า Session Cart นี้แก้ไขข้อมูลของเขางานนี้เท่าไร	DATETIME	CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP		NOT NULL	

ตารางที่ 3.3.5.3.10 carts attributes

No	Column Name	Column Description	Data Type	Default Value	Data Example	Constraint	Referenced Table
1	cartId	รหัส Session ที่ลูกค้าสร้างขึ้นเพื่อเก็บสินค้าในตะกร้า	VARCHAR(37)		[username]-[รหัส 16 หลัก]	Primary Key	
2	session CartId	รหัส Session ที่ลูกค้าสร้างขึ้น เพื่อเก็บสินค้าในตะกร้า เพื่อเก็บสินค้าในตะกร้า	VARCHAR(37)		[itemOwner]-[รหัส 16 หลัก]	Foreign Key	session_cart
3	itemId	รหัสของสินค้า	INT			Composite Foreign Key	item_sku
4	itemSize	รหัสของ SKU สินค้า	VARCHAR(20)			Composite Foreign Key	item_sku
5	itemStyle	ชื่อความหลากหลายของสินค้า	VARCHAR(50)			NOT NULL	
6	qty	จำนวนสินค้าในตะกร้า	INT	1		NOT NULL	
7	createdAt	ระบุว่า Cart ขึ้นนี้สร้างในเวลา哪ที่	DATETIME	CURRENT_TIMESTAMP	NOT NULL	7	createdAt
8	updatedAt	ระบุว่า Cart ขึ้นนี้แก้ไขข้อมูลของเขานะในเวลา哪ที่	DATETIME	CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP	NOT NULL	8	updatedAt

ตารางที่ 3.3.5.3.11 orders attributes

No	Column Name	Column Description	Data Type	Default Value	Data Example	Constraint	Referenced Table
1	orderId	รหัส Order ของลูกค้าจากร้านค้า	VARCHAR(33)		[itemOwner]-[รหัส 12 หลัก]	Primary Key	
2	customer Name	ชื่อแสดงผลของผู้ใช้งาน	VARCHAR(20)			Foreign Key, NOT NULL	accounts

ตารางที่ 3.3.5.3.11 orders attributes (ต่อ)

No	Column Name	Column Description	Data Type	Default Value	Data Example	Constraint	Referenced Table
3	address	ที่อยู่ของลูกค้า	VARCHAR(500)			Composite Foreign Key	
4	OrderGroupId	รหัสกลุ่ม Order ที่ใช้ในการจ่ายเงิน	VARCHAR(33)		[username]-[รหัส 12 หลัก]	Composite Foreign Key	
5	phone	เบอร์โทรศัพท์ของลูกค้า	VARCHAR(11)		0901001000	NOT NULL	
6	status	สถานะการดำเนินการสินค้าถึงลูกค้า	VARCHAR(20)	required	required, pending, in progress, completed, canceled	NOT NULL	
7	createdAt	ระบุว่า Order ในนี้สร้างในเวลาไหน	DATETIME	CURRENT_TIMESTAMP		NOT NULL	
8	paidOrder Date	ระบุว่า Order ในนี้จ่ายเงินตอนไหน	DATETIME			NULL	NULL
9	shippedOrder Date	ระบุว่า Order ในนี้จัดส่งตอนไหน	DATETIME			NULL	NULL
10	receivedOrder Date	ระบุว่า Order ในนี้รับสินค้าตอนไหน	DATETIME			NULL	NULL
11	rateOrderDate	ระบุว่า Order ในนี้ Review สินค้าตอนไหน	DATETIME			NULL	NULL

ตารางที่ 3.3.5.3.12 order_details attributes

No	Column Name	Column Description	Data Type	Default Value	Data Example	Constraint	Referenced Table
1	orderId	รหัส order ของลูกค้าจากร้านค้า	VARCHAR(33)		[itemOwner]-[รหัส 12 หลัก]	Primary Key	orders
2	itemId	รหัสของสินค้า	INT			Composite Foreign Key	item_details
3	itemStyle	รหัสของ SKU สินค้า	VARCHAR(20)			Composite Foreign Key	item_details

ตารางที่ 3.3.5.3.12 order_details attributes (ต่อ)

No	Column Name	Column Description	Data Type	Default Value	Data Example	Constraint	Referenced Table
4	itemSize	ชื่อความหลากหลายของสินค้า	VARCHAR(50)			Composite Foreign Key	item_details
5	qtyOrder	จำนวนสินค้าที่อยู่ใน Order	INT	1		NOT NULL	
6	priceEach	ราคาสินค้าที่ใช้ในการจ่ายเงินในแต่ละชิ้น	DECIMAL(32,2)	0		NOT NULL	
7	createdAt	ระบุว่า Order ในนี้สร้างในเวลาใดเท่าไร	DATETIME	CURRENT_TIMESTAMP		NOT NULL	

ตารางที่ 3.3.5.3.13 item_reviews attributes

No	Column Name	Column Description	Data Type	Default Value	Data Example	Constraint	Referenced Table
1	itemReviewId	รหัสการ Review สินค้า	VARCHAR(37)		[username]-[รหัส 16 หลัก]	Primary Key	
2	orderId	รหัส Order ของลูกค้าจากร้านค้า	VARCHAR(33)		[itemOwner]-[รหัส 12 หลัก]	NOT NULL	order_details
3	itemId	รหัสของสินค้า	INT			Composite Foreign Key	order_details
4	itemStyle	รหัสของ SKU สินค้า	VARCHAR(20)			Composite Foreign Key	order_details
5	itemSize	ชื่อความหลากหลายของสินค้า	VARCHAR(50)			Composite Foreign Key	order_details
6	username	ชื่อแสดงผลของผู้ใช้งาน	VARCHAR(20)			Foreign Key	accounts
7	comment	ข้อความแสดงการ Review สินค้า	VARCHAR(500)			NOT NULL	
8	PQrating	คะแนนด้าน Product Quality จาก 1-5	INT	0		NOT NULL	

ตารางที่ 3.3.5.3.13 item_reviews attributes (ต่อ)

No	Column Name	Column Description	Data Type	Default Value	Data Example	Constraint	Referenced Table
9	SSrating	คะแนนด้าน Selling Service จาก 1-5	INT		0	NOT NULL	
10	DSrating	คะแนนด้าน Delivery Service จาก 1-5	INT		0	NOT NULL	
11	like	จำนวน User ที่ชอบ การ Review สินค้า	INT		0	NOT NULL	
12	createdAt	ระบุว่า Review นี้ สร้างในเวลา哪ที่	DATETIME		CURRENT_TIMESTAMP	NOT NULL	
13	updatedAt	ระบุว่า Review นี้ แก้ไขข้อมูลของเข้า ในเวลา哪ที่	DATETIME		CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP	NOT NULL	

ตารางที่ 3.3.5.3.14 item_review_likes attributes

No	Column Name	Column Description	Data Type	Default Value	Data Example	Constraint	Referenced Table
1	ItemReviewId	รหัสการ Review สินค้า	VARCHAR(37)			Composite Key, Foreign Key	item_reviews
2	username	ชื่อแสดงผลของผู้ใช้งาน	VARCHAR(20)			Composite Key, Foreign Key	accounts

ตารางที่ 3.3.5.3.15 contents attributes

No	Column Name	Column Description	Data Type	Default Value	Data Example	Constraint	Referenced Table
1	contentId	รหัสของ Content Gallery	INT	AUTO INCREMENT VALUE		Primary Key	
2	name	ชื่อ Content Gallery	VARCHAR(100)			NOT NULL	
3	description	รายละเอียดของ Content Gallery	VARCHAR(5000)	NULL		NULL	
4	content Owner	ชื่อเจ้าของ Content Gallery	VARCHAR(20)			Foreign Key, NOT NULL	accounts
5	style	Style การออกแบบ Content Gallery	VARCHAR(50)			NULL	
6	like	จำนวนคนที่ชอบใน Content นั้น	INT	0		UNIQUE, NOT NULL	
7	createdAt	ระบุว่า Content ขึ้นมาสร้างในเวลา哪ที่	DATETIME	CURRENT_TIMESTAMP		NOT NULL	
8	updatedAt	ระบุว่า Content ขึ้นมาแก้ไขข้อมูลของ เขาในเวลา哪ที่	DATETIME	CURRENT_TIMESTAMP ON UPDATE CURRENT_TIME STAMP		NOT NULL	

ตารางที่ 3.3.5.3.16 gallery_comment attributes

No	Column Name	Column Description	Data Type	Default Value	Data Example	Constraint	Referenced Table
1	commentId	รหัสของ Comment	VARCHAR(37)			Primary Key	
2	contentId	รหัสของ Content Gallery	INT	AUTO INCREMENT VALUE		Foreign Key, NOT NULL	contents
3	username	ชื่อเจ้าของ Content Gallery	VARCHAR(20)			Foreign Key, NOT NULL	accounts
4	comment	ข้อความใน Content Gallery	VARCHAR(500)	NULL		NULL	

ตารางที่ 3.3.5.3.16 gallery_comment attributes (ต่อ)

No	Column Name	Column Description	Data Type	Default Value	Data Example	Constraint	Referenced Table
5	createdAt	ระบุว่า Comment นี้สร้างในเวลาใด เท่าไร	DATETIME	CURRENT_TIMESTAMP		NOT NULL	

ตารางที่ 3.3.5.3.17 content_likes attributes

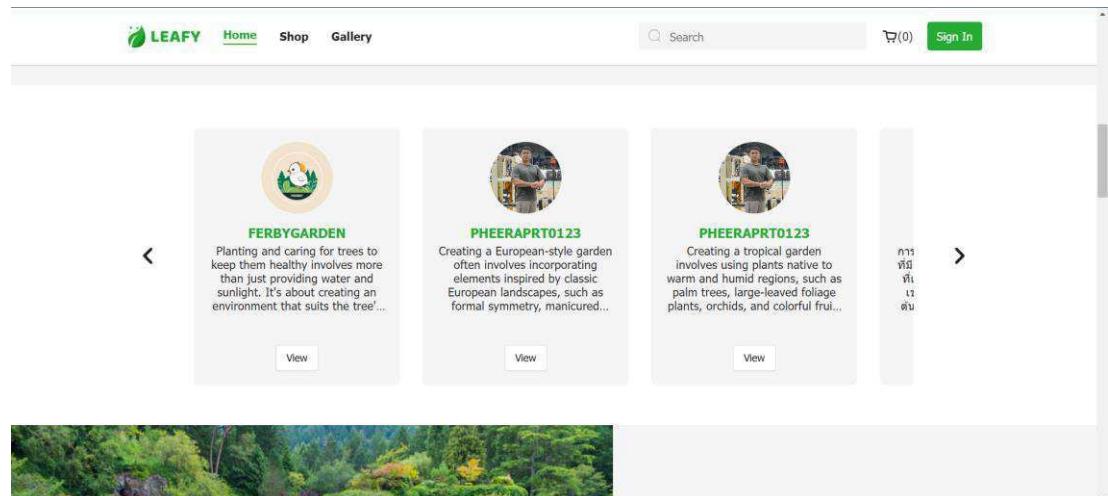
No	Column Name	Column Description	Data Type	Default Value	Data Example	Constraint	Referenced Table
1	commentId	รหัสการ Review สินค้า	VARCHAR(37)			Composite Key, Foreign Key	gallery_comments
2	username	ชื่อแสดงผลของผู้ใช้งาน	VARCHAR(20)			Composite Key, Foreign Key	accounts

3.4 การออกแบบกรณีทดสอบ (Test Cases)

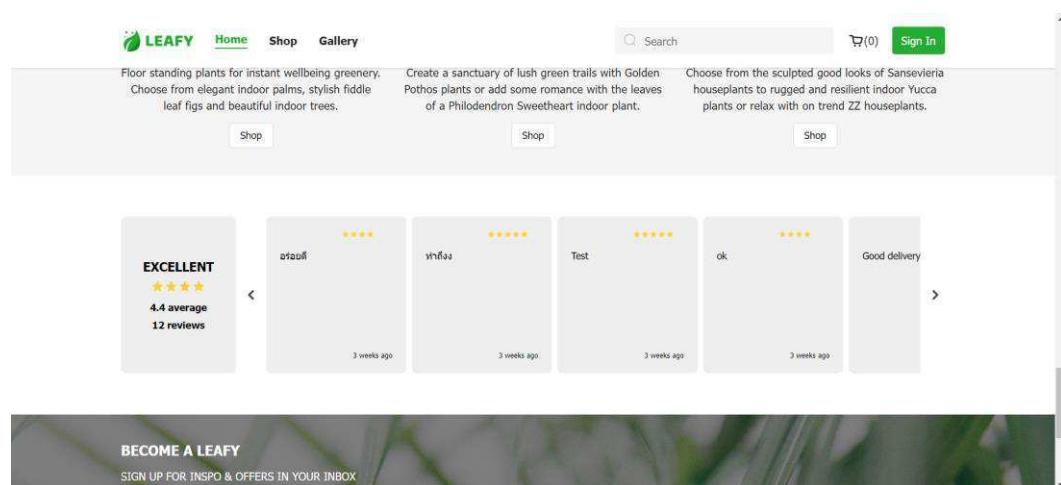
การออกแบบกรณีทดสอบของ Leafy มีกระบวนการแบ่งการทดสอบเป็น 2 กลุ่ม คือการทดสอบการออกแบบและการฟังก์ชันการทำงานของข้อมูลบนฝั่ง Browser ของ Frontend และการทดสอบข้อมูลบน API ของ Backend โดย 2 การทดสอบนี้จะต้องทำร่วมกันในระดับ Production level ดังนี้

3.4.1. ฝั่ง Frontend ได้ทำการทดสอบโดยการ Demo การทำงาน ดังต่อไปนี้

1. Home page เป็นหน้าสำหรับแสดงข้อมูลของเว็บไซต์เบื้องต้น โดยจะต้องสามารถนำข้อมูลที่ได้มาแสดงในส่วนของบทความการออกแบบจัดสวนและข้อมูลการรีวิวของผู้ใช้งานล่าสุดได้ ดังแสดงรูปที่ 3.4.1.1 – 3.4.1.2



รูปที่ 3.4.1.1 ข้อมูลบทความการออกแบบจัดสวนโดยการสุม



รูปที่ 3.4.1.2 ข้อมูลรีวิวของผู้ใช้งานล่าสุด

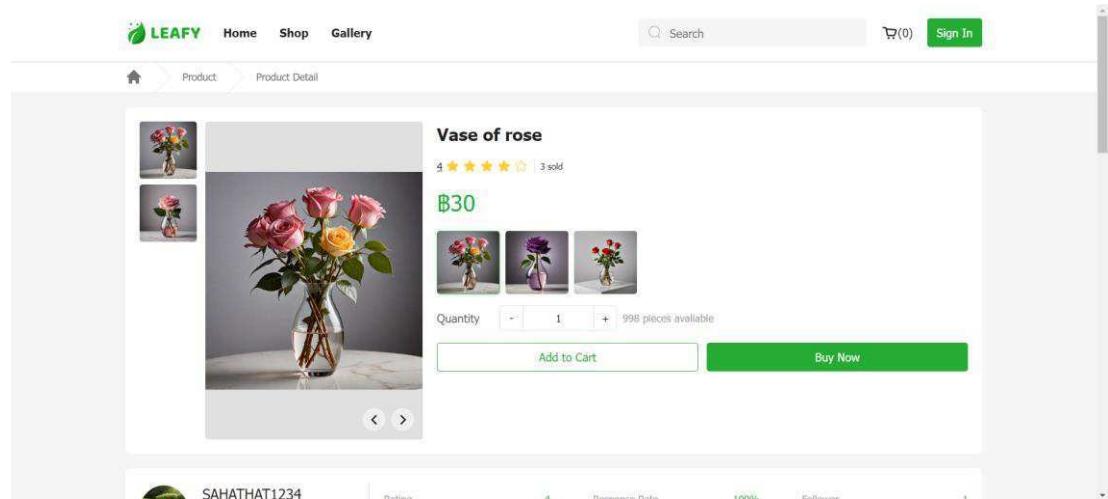
2. Shop page เป็นหน้าสำหรับแสดงข้อมูลสินค้า โดยจะต้องสามารถแสดงข้อมูลของสินค้าทั้งหมดได้ สามารถคัดกรองเพื่อให้ได้สินค้าที่ต้องการได้ ดังแสดงรูปที่

3.4.1.3 – 3.4.1.4

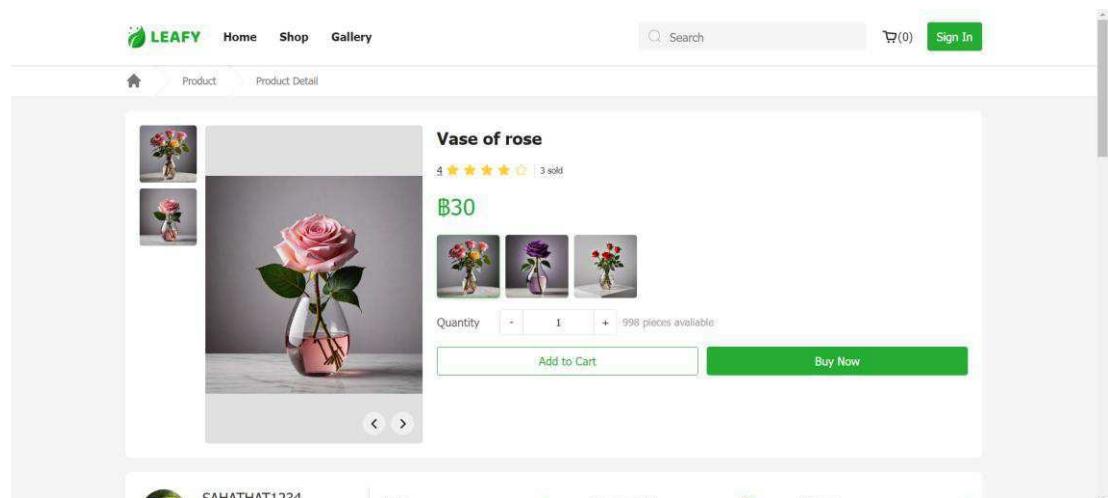
รูปที่ 3.4.1.3 ข้อมูลสินค้าทั้งหมดที่มีขายในเว็บไซต์

รูปที่ 3.4.1.4 ข้อมูลสินค้าที่ถูกคัดกรองเพื่อให้ได้สินค้าที่ผู้ใช้งานต้องการ

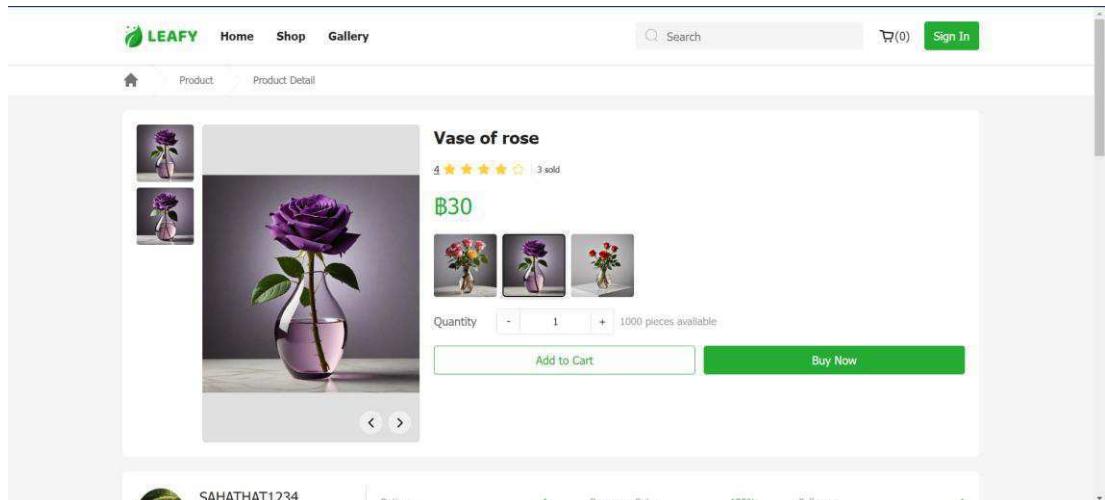
3. Product detail page เป็นหน้าสำหรับแสดงข้อมูลรายละเอียดสินค้าหลัก โดยจะต้องสามารถแสดงรายละเอียดของรูปแบบสินค้าอื่นๆ และหากมีรูปของรูปแบบสินค้ามากกว่า 1 จะต้องสามารถเปลี่ยนรูปได้ ดังรูปที่ 3.4.1.5 – 3.4.1.8



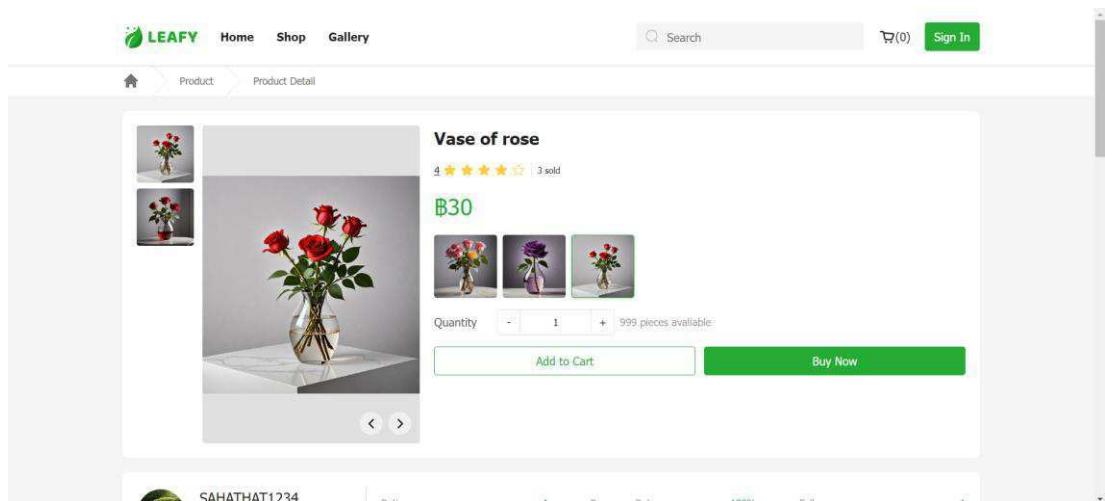
รูปที่ 3.4.1.5 ตัวอย่างข้อมูลรายละเอียดรูปแบบสินค้าที่ 1



รูปที่ 3.4.1.6 ตัวอย่างข้อมูลรายละเอียดรูปแบบสินค้าที่ 1

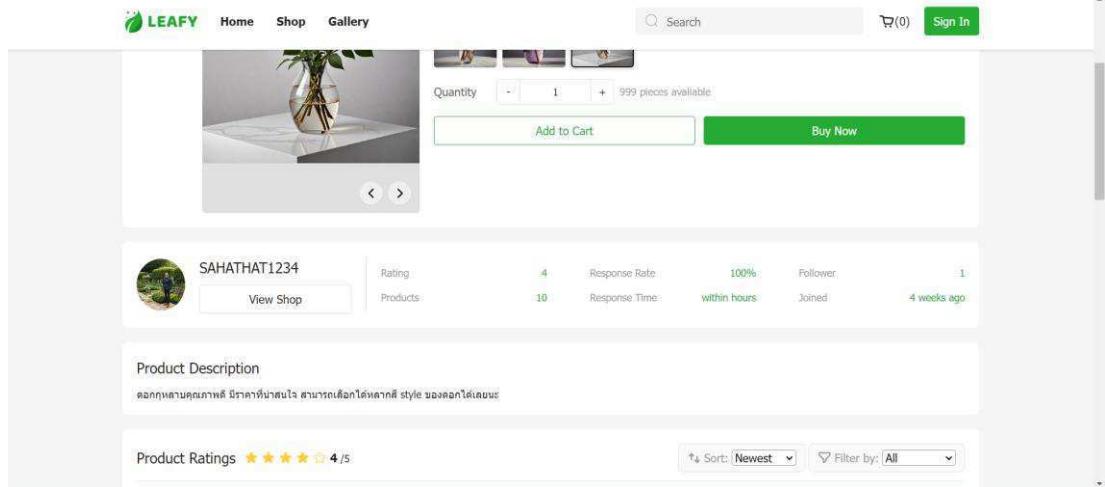


รูปที่ 3.4.1.7 ตัวอย่างข้อมูลรายละเอียดรูปแบบสินค้าที่ 2



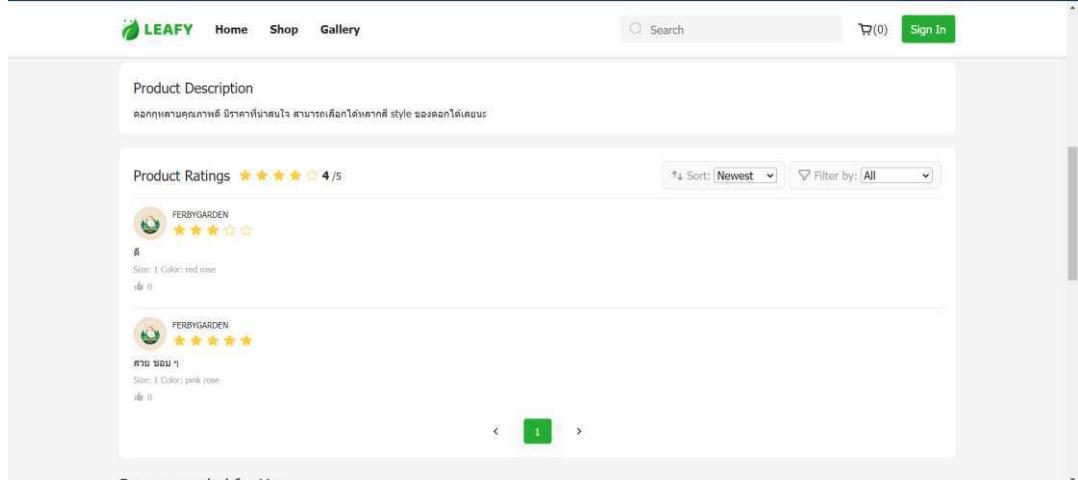
รูปที่ 3.4.1.8 ตัวอย่างข้อมูลรายละเอียดรูปแบบสินค้าที่ 3

จะต้องสามารถแสดงคำอธิบายรายละเอียดเพิ่มเติม(หากมี)และข้อมูลร้านค้าที่เกี่ยวข้องกับสินค้าได้ ดังแสดงรูปที่ 3.4.1.9

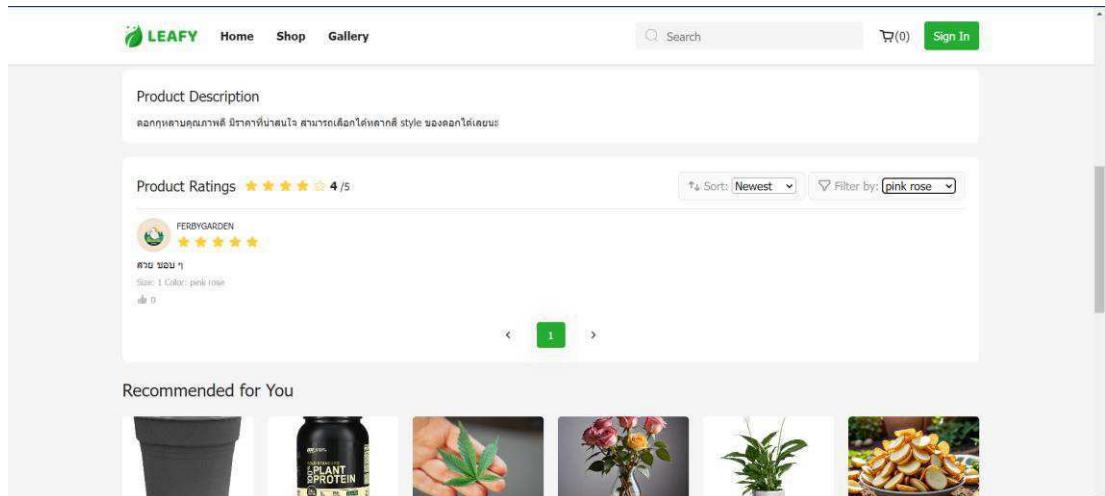


รูปที่ 3.4.1.9 ข้อมูลร้านค้าและคำอธิบายที่เกี่ยวกับสินค้า

ข้อมูลที่จำเป็นสำหรับการตัดสินใจเลือกซื้อของผู้ใช้งานที่ขาดไม่ได้คือข้อมูลรีวิวสินค้า ดังนั้น จะต้องสามารถแสดงข้อมูลรีวิวของสินค้าของผู้ใช้คนอื่นๆ ได้ ดังแสดงรูปที่ 3.4.1.10 – 3.4.1.11

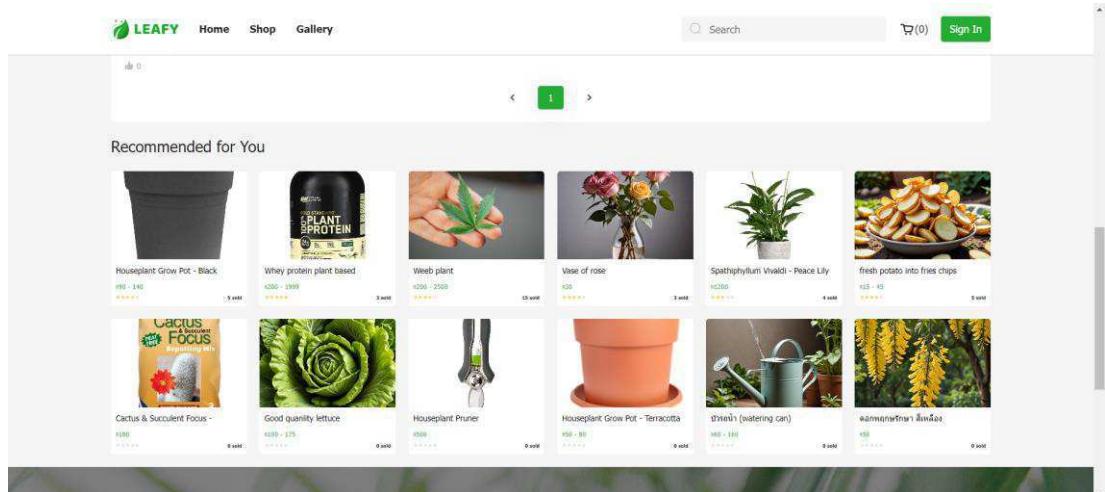


รูปที่ 3.4.1.10 ข้อมูลรีวิวสินค้าจากผู้ใช้งานที่ซื้อสินค้า



รูปที่ 3.4.1.11 ตัวอย่างข้อมูลรีวิวที่ถูกคัดกรองตามรูปแบบสินค้า

และในส่วนสุดท้ายจะต้องสามารถแสดงข้อมูลแนะนำสินค้าที่น่าสนใจให้กับผู้ใช้งานเพิ่มเติมได้ ดังแสดงรูปที่ 3.4.1.12



รูปที่ 3.4.1.12 ตัวอย่างข้อมูลสินค้าแนะนำที่มีรายละเอียดในเว็บไซต์

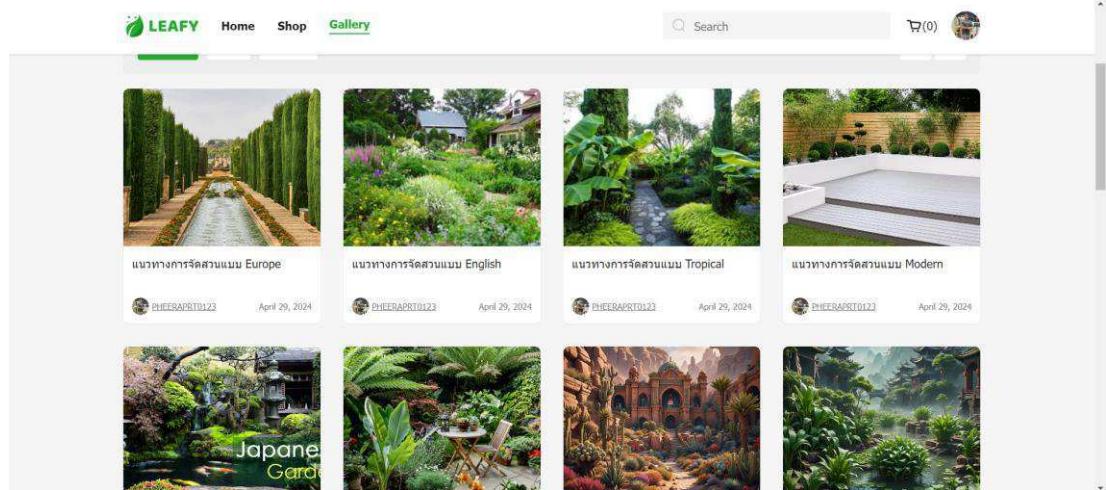
4. Sign-in และ Sign-up เป็นหน้าสำหรับสมัครสมาชิกและสำหรับเข้าสู่ระบบ โดยจะต้องสามารถสร้าง Account และคัดกรองข้อมูลที่ผิดพลาดแล้วแจ้งเตือนให้กับผู้ใช้งานทราบได้ ดังแสดงรูปที่ 3.4.1.13

รูปที่ 3.4.1.13 แสดงการแจ้งเตือนให้กับผู้ใช้ เมื่อกรอกข้อมูลผิดพลาด

เมื่อผู้ใช้งานกรอกข้อมูลสำหรับสมัครสมาชิกสำเร็จ แล้วต้องการเข้าสู่ระบบแต่ไม่มีการกรอกข้อมูลผิด ซึ่งจะต้องแสดงการแจ้งเตือนให้กับผู้ใช้งานได้รู้ ดังแสดงรูปที่ 3.4.1.14

รูปที่ 3.4.1.14 การแจ้งเตือนให้กับผู้ใช้งานได้รู้ เมื่อไม่สามารถเข้าสู่ระบบได้

5. Gallery list page และ Gallery detail page เป็นหน้าสำหรับแสดงข้อมูลเกี่ยวกับบทความการออกแบบจัดสวนที่ถูกเขียนจากผู้ใช้งานในระบบทั้งหมด โดยจะต้องสามารถแสดงข้อมูลบทความได้ ดังแสดงรูปที่ 3.4.1.15



รูปที่ 3.4.1.15 ตัวอย่างข้อมูลบทความที่ถูกเขียนโดยผู้ใช้งานในระบบ

และจะต้องสามารถแสดงข้อมูลรายละเอียดทั้งหมดของบทความได้ ดังแสดงรูปที่ 3.4.1.16

แนวทางการจัดสวนแบบ Europe

April 29, 2024

 **PHEERAPRT0123**



Creating a European-style garden often involves incorporating elements inspired by classic European landscapes, such as formal symmetry, manicured lawns, geometrically shaped flower beds, and clipped hedges. Consider using traditional European plants like roses, lavender, boxwood, and cypress trees to evoke the charm of European gardens. Adding architectural features like stone pathways, wrought iron gates, and statuary can enhance the garden's elegance and sophistication. Water features such as fountains or ornamental ponds add a tranquil atmosphere. To maintain aesthetics, pay attention to detail in design and layout, aiming for a harmonious balance between

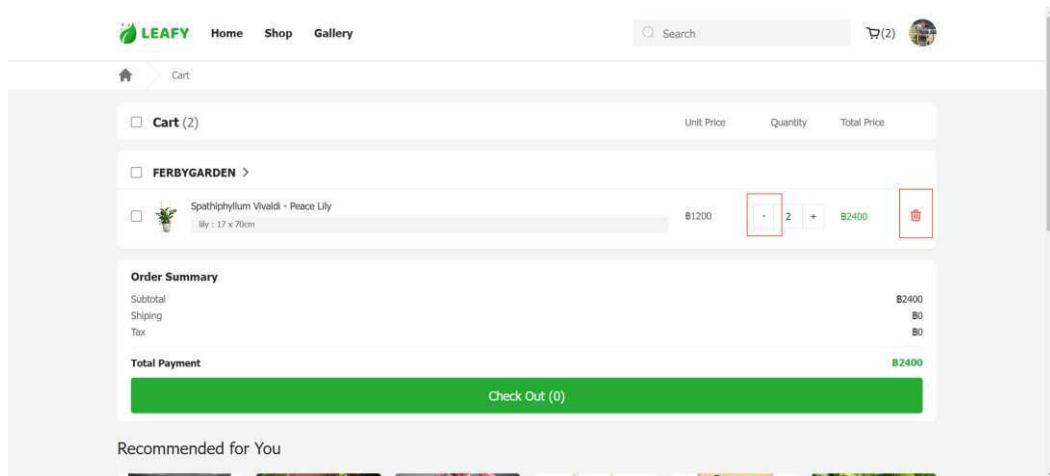
รูปที่ 3.4.1.16 ตัวอย่างข้อมูลรายละเอียดของบทความ

6. Cart list เป็นหน้าสำหรับรวมสินค้าที่ผู้ใช้สนใจสั่งซื้อและ Place order (Check out) เป็นหน้าสำหรับส่งคำร้องไปยังผู้ขายเพื่อให้ผู้ขายได้ทราบและเตรียมของให้พร้อม โดย Cart list จะต้องสามารถแสดงสินค้าที่ผู้ใช้งานเลือกได้ ดังแสดงรูปที่ 3.4.1.17 – 3.4.1.18

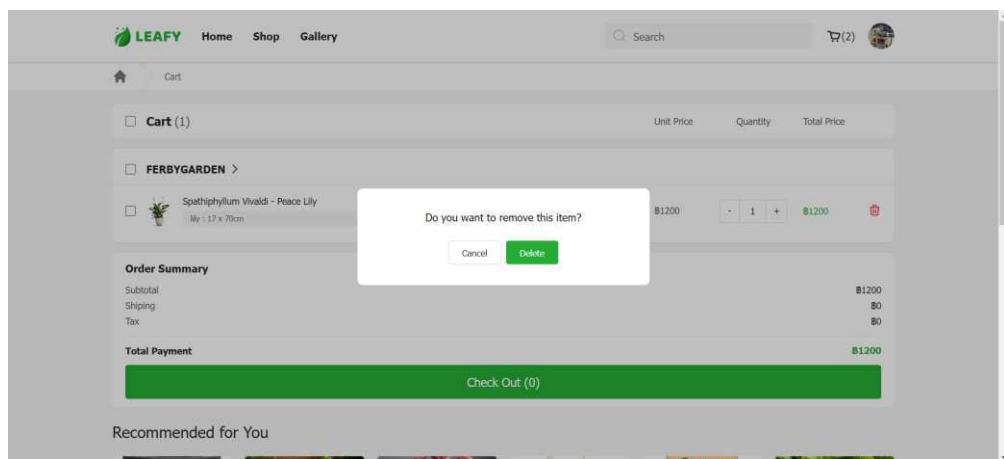
รูปที่ 3.4.1.17 ตัวอย่างการเลือกซื้อสินค้าของผู้ใช้ลงตะกร้า

รูปที่ 3.4.1.18 ตัวอย่างสินค้าที่ผู้ใช้ได้เพิ่มลงตะกร้า

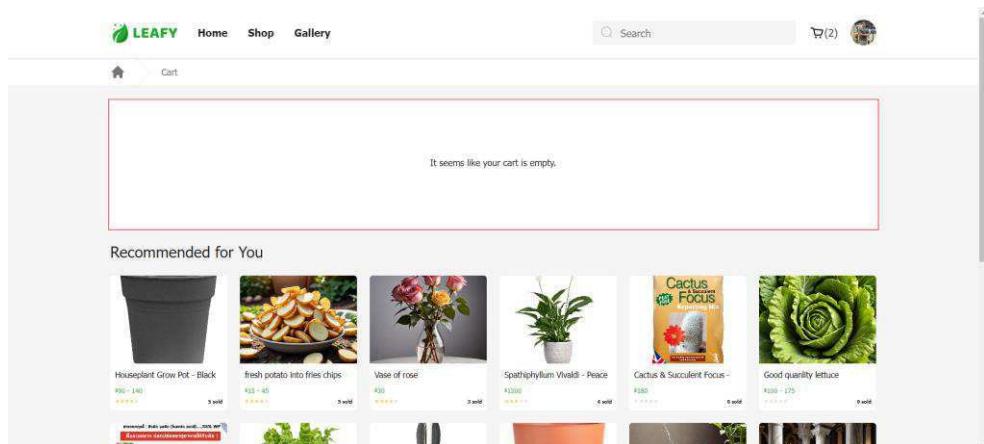
หากผู้ใช้ต้องไม่ต้องการสินค้าแล้วจะต้องสามารถนำออกจากระบบทันทีได้ ดังแสดงรูปที่ 3.4.1.19 – 3.4.1.21



รูปที่ 3.4.1.19 วิธีการนำสินค้าออกจากตะกร้า



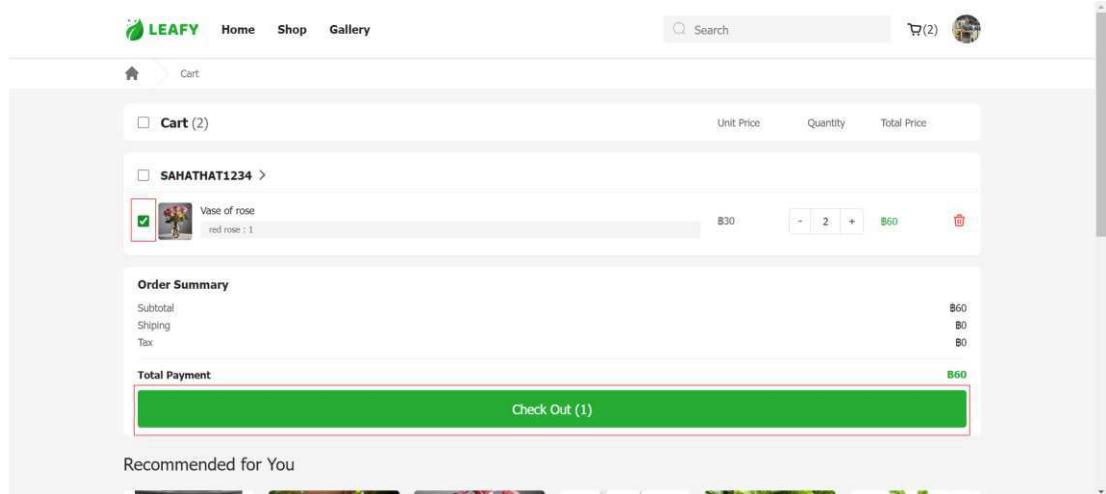
รูปที่ 3.4.1.20 แจ้งเตือนให้กับผู้ใช้ เมื่อต้องการนำสินค้าออกจากตะกร้า



รูปที่ 3.4.1.21 เมื่อผู้ใช้งานได้เอาสินค้าออกจากตะกร้าสำเร็จแล้วไม่มีสินค้าเหลืออยู่

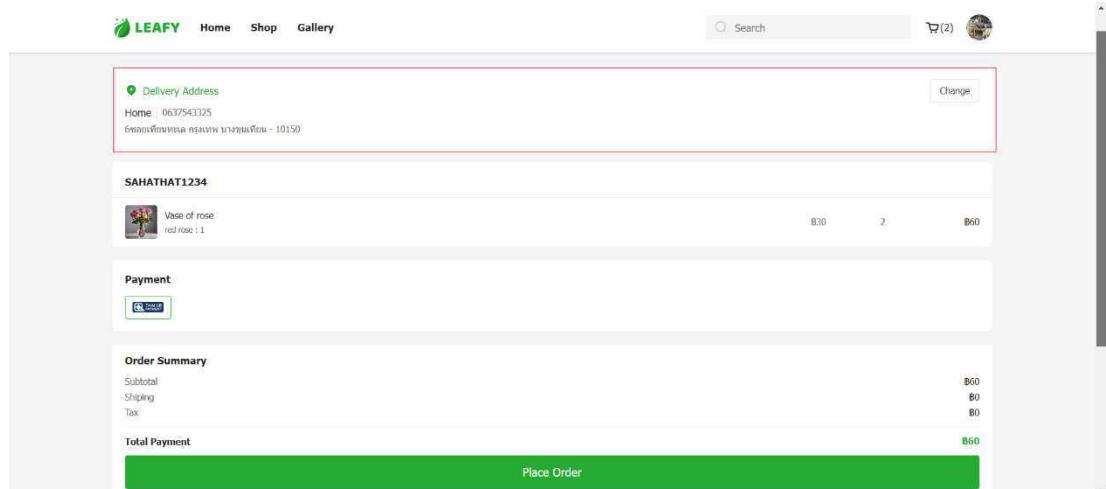
หากผู้ใช้งานต้องการที่จะสั่งซื้อสินค้าขึ้นนั้นจริงๆ จะต้องผ่านกระบวนการดังนี้ ดังแสดงรูปที่

3.4.1.22

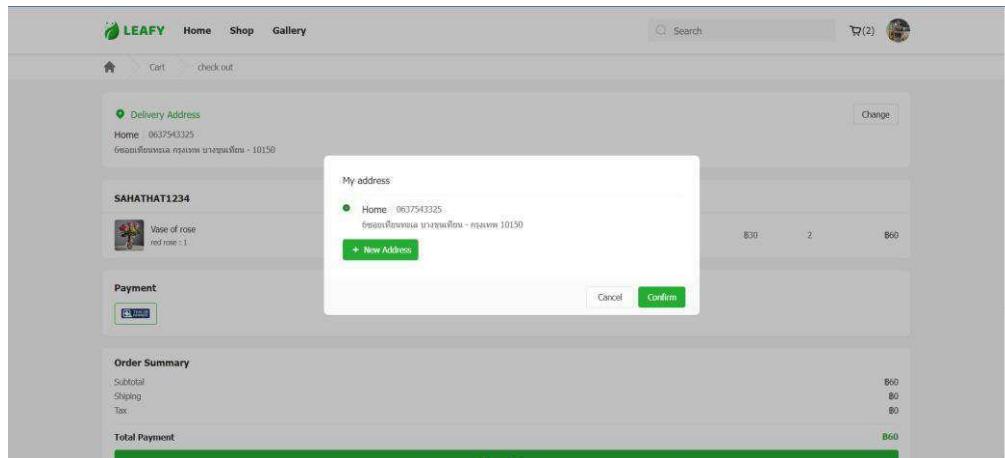


รูปที่ 3.4.1.22 ตัวอย่างการเลือกสินค้าก่อนส่งคำร้องไปยังผู้ขาย

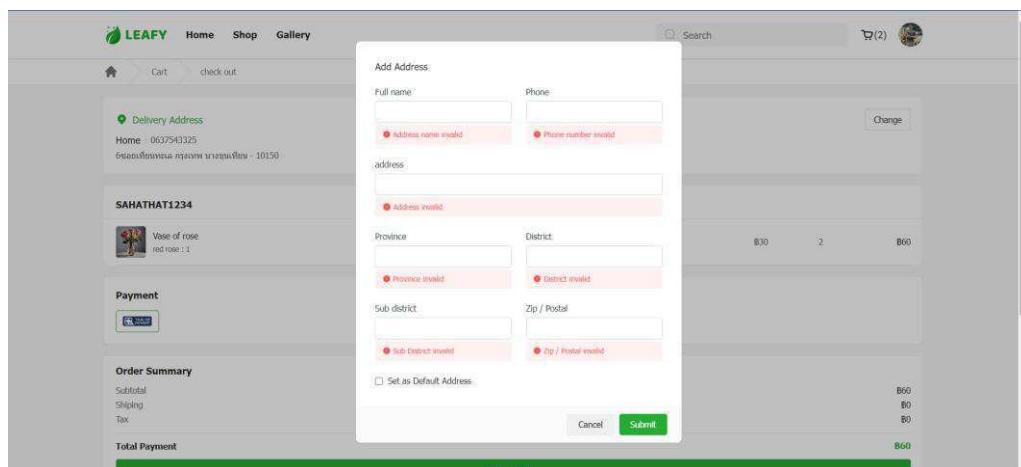
ในการส่งคำร้องผู้ใช้งานจำเป็นจะต้องเลือกที่อยู่จัดส่งหรือสร้างที่อยู่ใหม่เพื่อให้ผู้ขายสามารถจัดส่งสินค้าถึงผู้ใช้งานได้ ดังแสดงรูปที่ 3.4.1.23 – 3.4.1.26



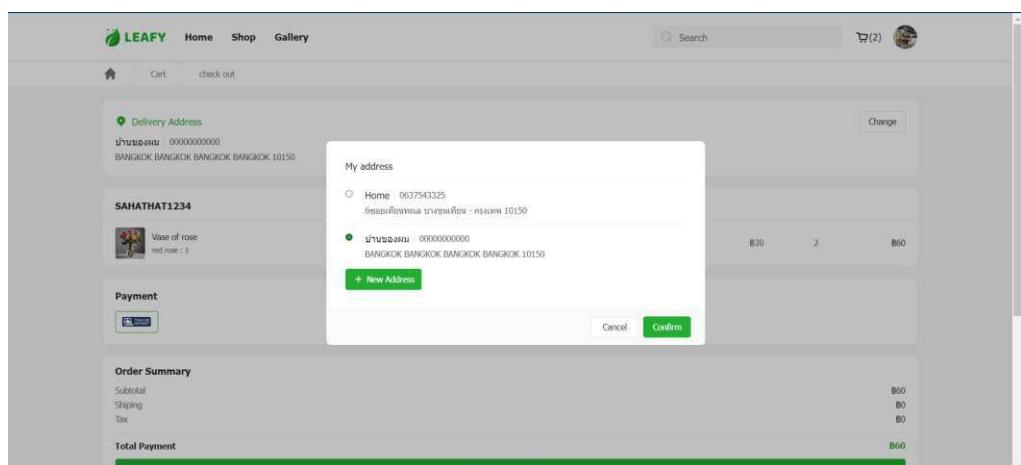
รูปที่ 3.4.1.23 ตัวอย่างที่อยู่สำหรับจัดส่งของผู้ใช้งาน



รูปที่ 3.4.1.24 ตัวอย่างข้อมูลที่อยู่ทั้งหมดของผู้ใช้งาน

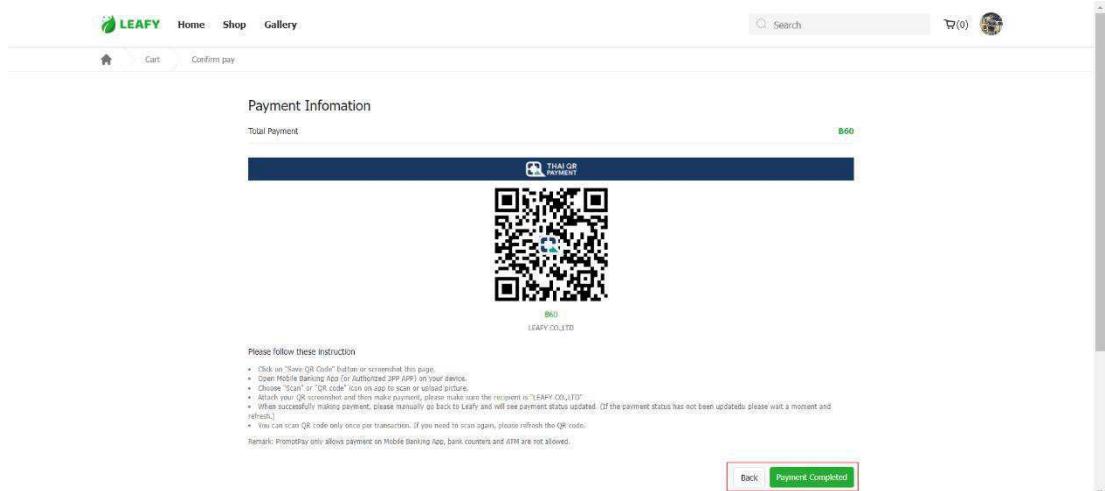


รูปที่ 3.4.1.25 ตัวอย่างการแจ้งเตือนให้กับผู้ใช้ เมื่อกรอกข้อมูลที่อยู่ไม่ถูกต้อง



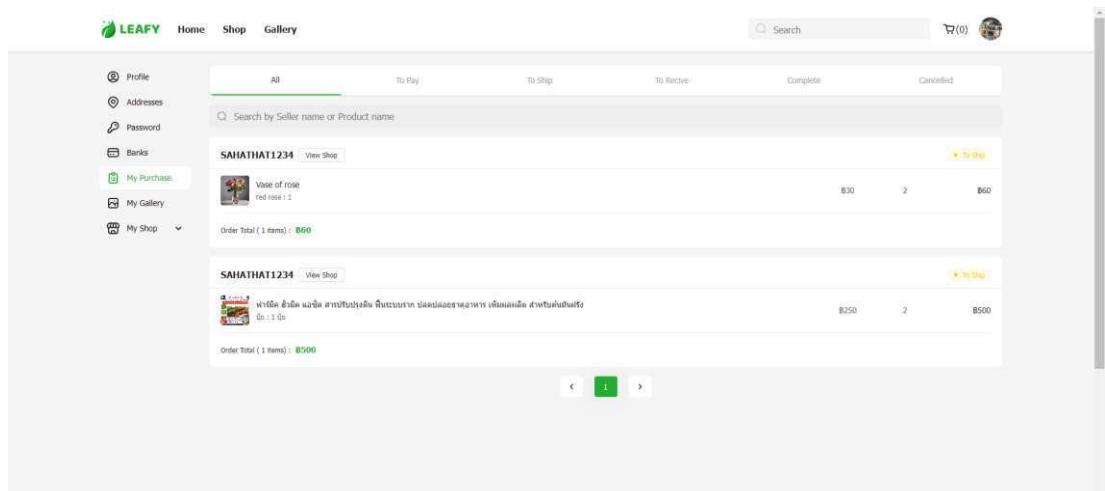
รูปที่ 3.4.1.26 ตัวอย่างข้อมูลที่อยู่ใหม่ที่ถูกสร้างโดยผู้ใช้งาน

หากผู้ใช้งานต้องการส่งคำร้องแล้วจะต้องสามารถกด “Place order” เพื่อไปสู่กระบวนการจำลองชำระเงินได้ ดังแสดงรูปที่ 3.4.1.27



รูปที่ 3.4.1.27 กระบวนการจำลองการชำระเงินหลังผู้ใช้ส่งคำร้อง

ผู้ใช้งานสามารถเลือกได้ว่าต้องการชำระเงินตอนนี้หรือชำระเงินทีหลังได้ โดยเมื่อเลือกอย่างใดอย่างหนึ่งแล้วจะพาผู้ใช้ไปยังหน้า My purchase ได้ ดังแสดงรูปที่ 3.4.1.28



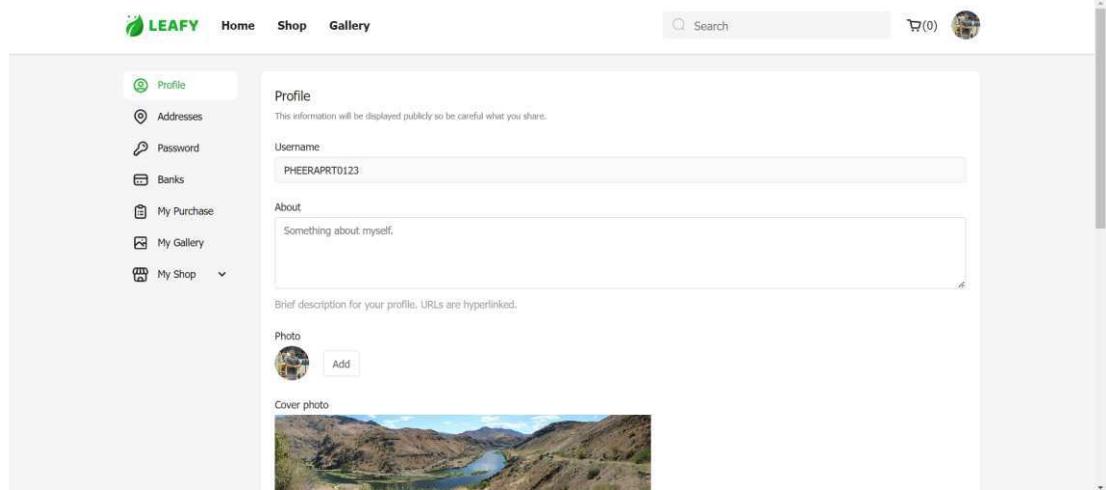
รูปที่ 3.4.1.28 ตัวอย่างสินค้าที่ถูกส่งคำร้องและชำระเงินสำเร็จ

7. Profile page เป็นหน้าสำหรับแสดงข้อมูลของผู้ใช้งานทุกคน ซึ่งสามารถแบ่งได้เป็น 2 แบบคือ Product และ Gallery โดยจะต้องสามารถแสดงเฉพาะเนื้อหาที่เป็นของผู้ใช้งานนั้นเท่านั้น ดังแสดงรูปที่ 3.4.1.29 – 3.4.1.30

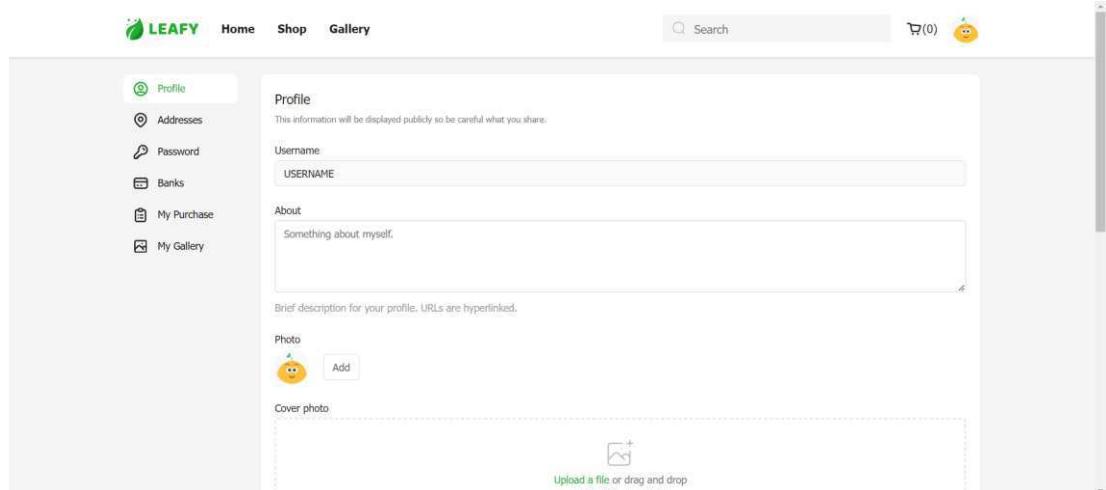
รูปที่ 3.4.1.29 ตัวอย่างข้อมูล Profile ของผู้ใช้งานแบบ Product

รูปที่ 3.4.1.30 ตัวอย่างข้อมูล Profile ของผู้ใช้งานแบบ Gallery

8. Setting page เป็นหน้ารวม Feature สำหรับผู้ใช้งาน โดยจะต้องสามารถแสดง Feature ที่สามารถใช้งานได้ที่แบ่งตาม Role ดังแสดงรูปที่ 3.4.1.31 – 3.4.1.32



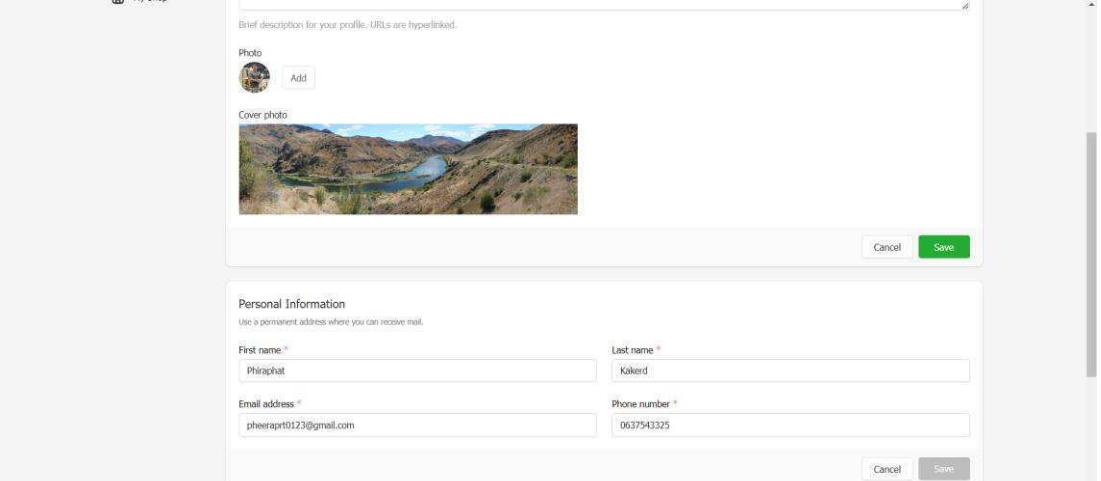
รูปที่ 3.4.1.31 Feature ทั้งหมดของผู้ใช้ Role Supplier



รูปที่ 3.4.1.32 Feature ทั้งหมดของผู้ใช้ Role User

โดย Feature ภายในทั้งหมดของ Setting มีดังนี้

1. Profile setting เป็น Feature สำหรับปรับเปลี่ยนข้อมูลส่วนตัวของผู้ใช้งานด้วยตัวเอง โดยจะต้องสามารถเปลี่ยนข้อมูลที่จำเป็นและสามารถเปลี่ยนรูปได้ ดังแสดงรูปที่ 3.4.1.33 – 3.4.1.34



Brief description for your profile. URLs are hyperlinked.

Photo  Add

Cover photo 

Personal Information

Use a permanent address where you can receive mail.

First name * Phiraphat

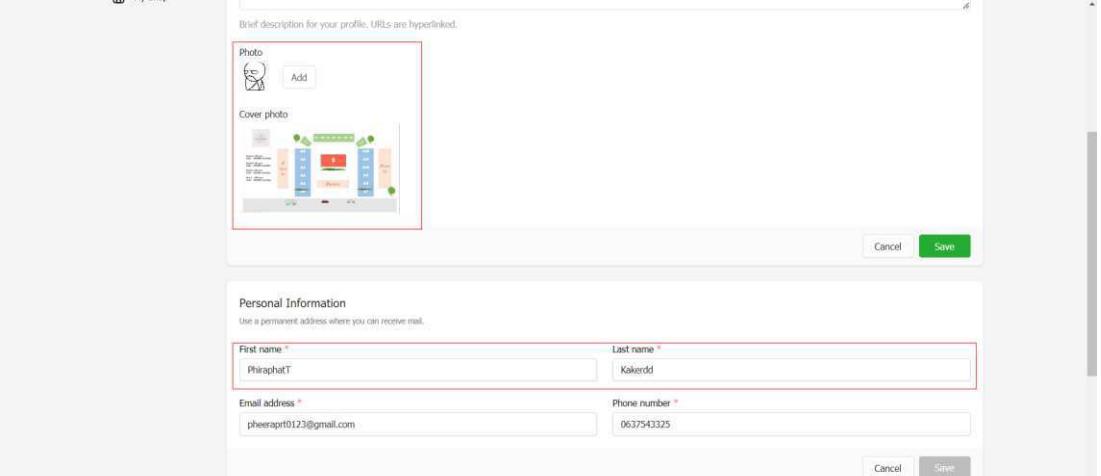
Last name * Kakerdd

Email address * pheeraprt0123@gmail.com

Phone number * 0637543325

Cancel Save

รูปที่ 3.4.1.33 ตัวอย่างข้อมูลของผู้ใช้งานก่อนเปลี่ยน



Brief description for your profile. URLs are hyperlinked.

Photo  Add

Cover photo 

Personal Information

Use a permanent address where you can receive mail.

First name * PhiraphatT

Last name * Kakerdd

Email address * pheeraprt0123@gmail.com

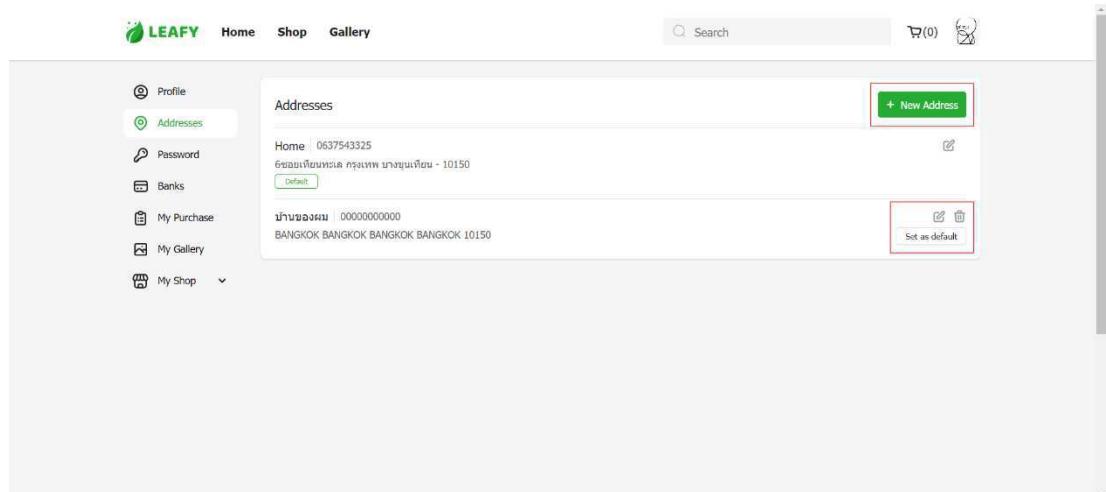
Phone number * 0637543325

Cancel Save

รูปที่ 3.4.1.34 ตัวอย่างข้อมูลของผู้ใช้งานหลังเปลี่ยน

2. Address list และ Address create เป็น Feature สำหรับจัดการกับที่อยู่ของผู้ใช้งานเพื่อจัดส่งสินค้า โดยจะต้องสามารถเพิ่ม แก้ไข หรือลบได้ ดังแสดงรูปที่

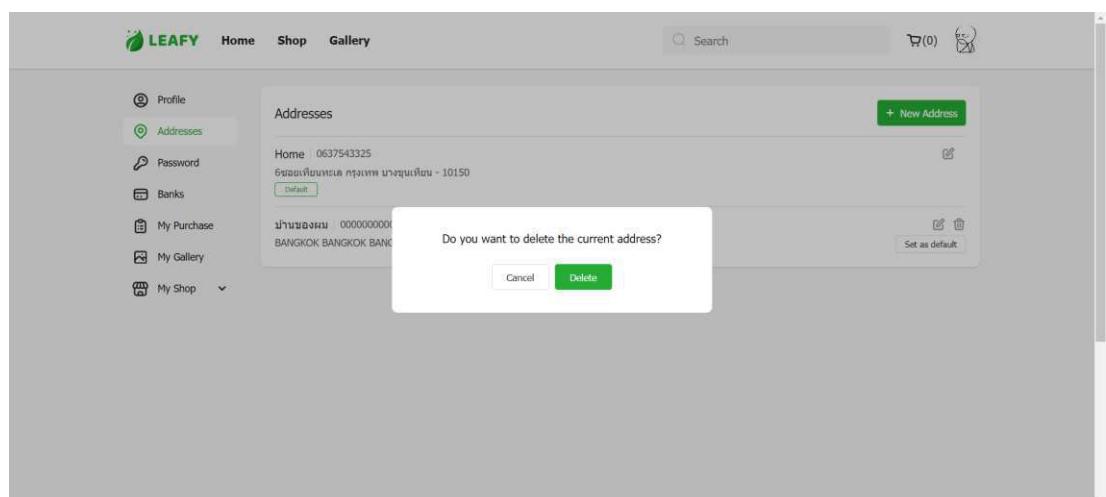
3.4.1.35



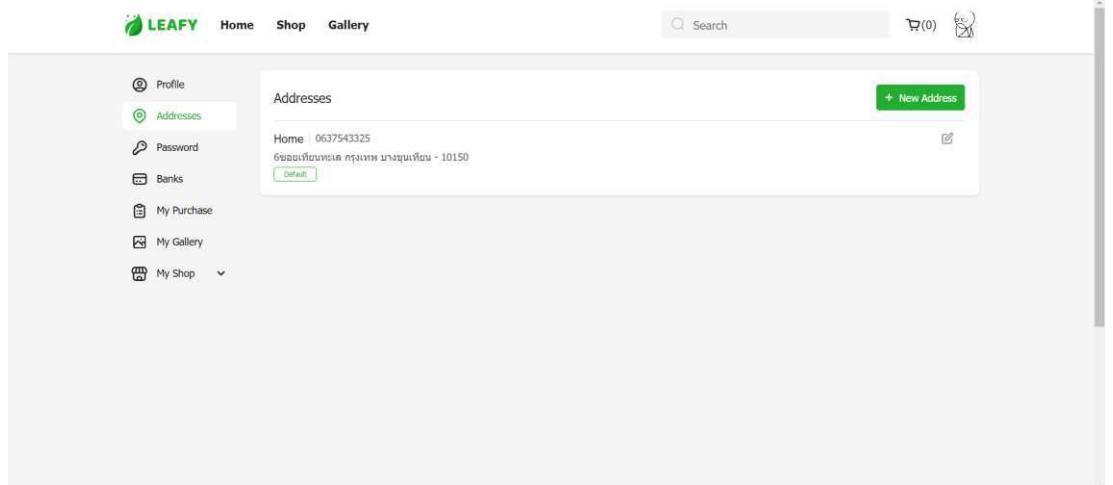
รูปที่ 3.4.1.35 วิธีการจัดการที่อยู่ด้วยการลบ แก้ไข เพิ่มหรือตั้งให้เป็น Default

หากผู้ใช้ต้องการที่จะลบข้อมูลที่อยู่ จะต้องมีการแจ้งเตือนให้กับผู้ใช้ทราบ ดังแสดงรูปที่

3.4.1.36 – 3.4.1.37

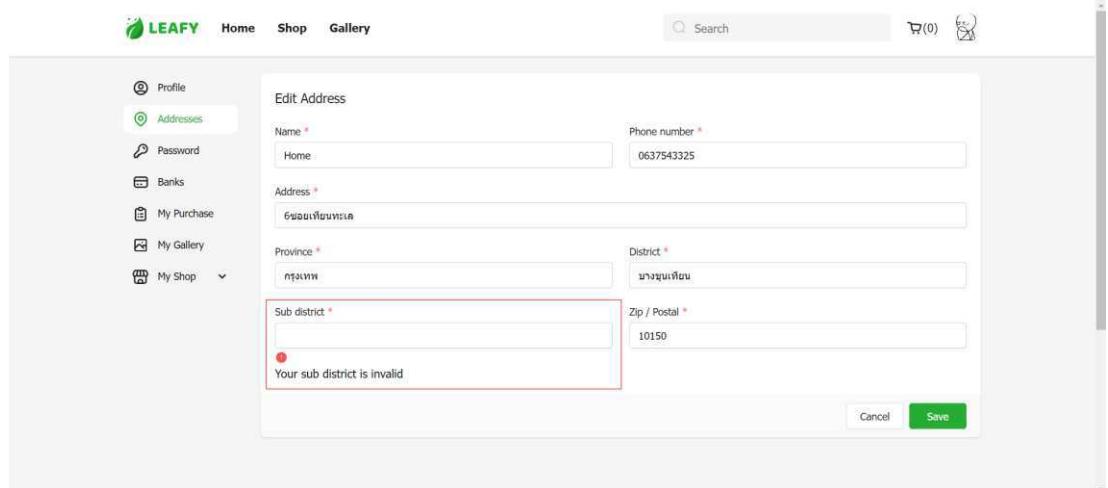


รูปที่ 3.4.1.36 การแจ้งเตือนให้กับผู้ใช้งานทราบ เพื่อยืนยันการลบข้อมูล



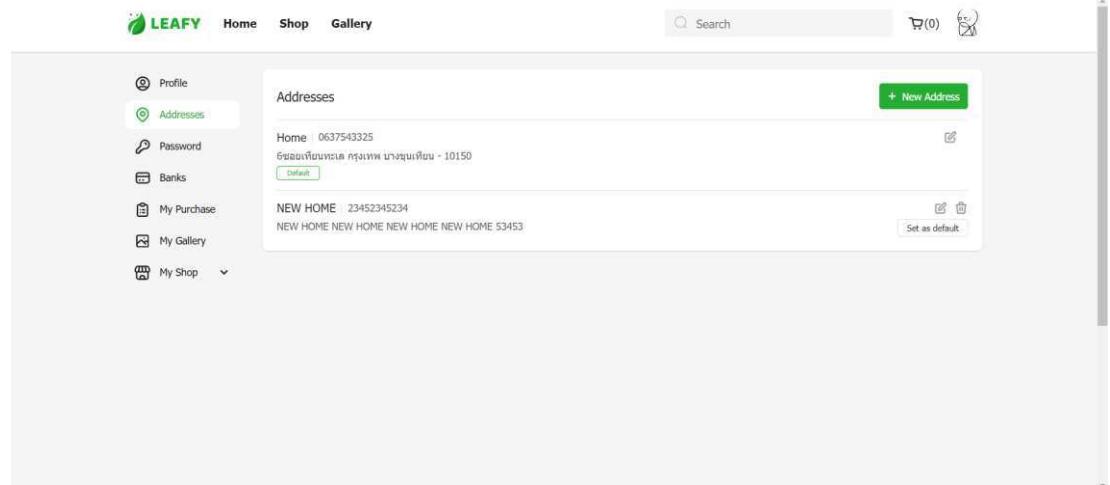
รูปที่ 3.4.1.37 เมื่อข้อมูลที่อยู่ถูกลบจะต้องหายไปจากระบบ

เมื่อผู้ใช้ต้องการสร้างที่อยู่ใหม่หรือแก้ไขจะต้องมีการตรวจสอบความถูกต้องของข้อมูลแล้ว
แจ้งให้กับผู้ใช้งานทราบ โดยจะต้องไม่สามารถสร้างหรือแก้ไขได้ ดังแสดงรูปที่ 3.4.1.38

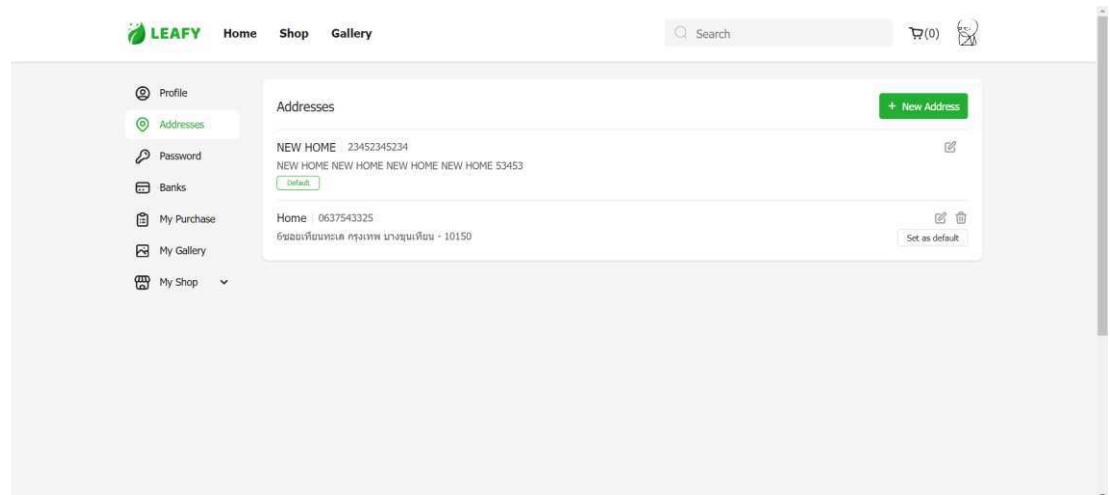


รูปที่ 3.4.1.38 ตัวอย่างการแจ้งเตือนผู้ใช้จากการกรอกข้อมูลผิดพลาด

หากผู้ใช้ต้องการตั้งค่าที่อยู่อื่นๆ ให้เป็น Default จะต้องสามารถทำได้ ดังแสดงรูปที่ 3.4.1.39 – 3.4.1.40

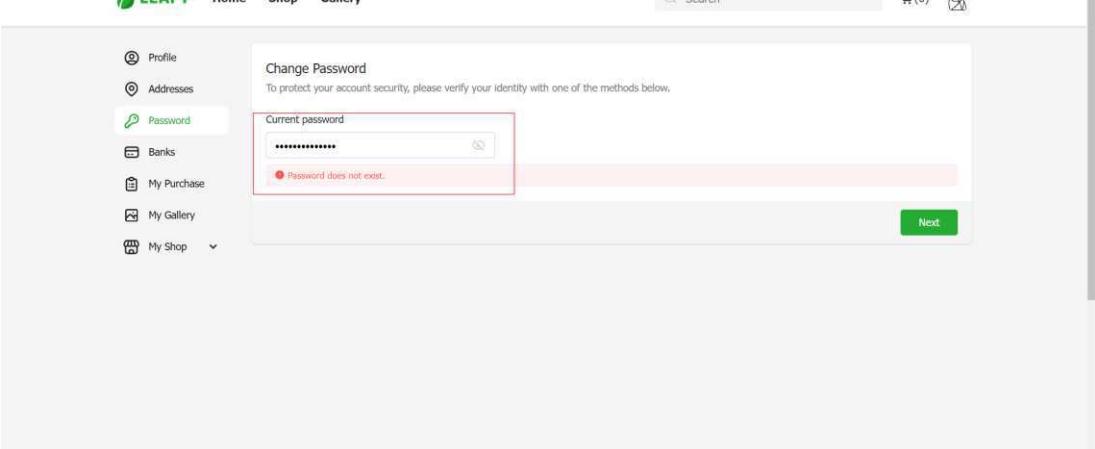


รูปที่ 3.4.1.39 ตัวอย่างข้อมูลที่อยู่ก่อนเปลี่ยนเป็น Default



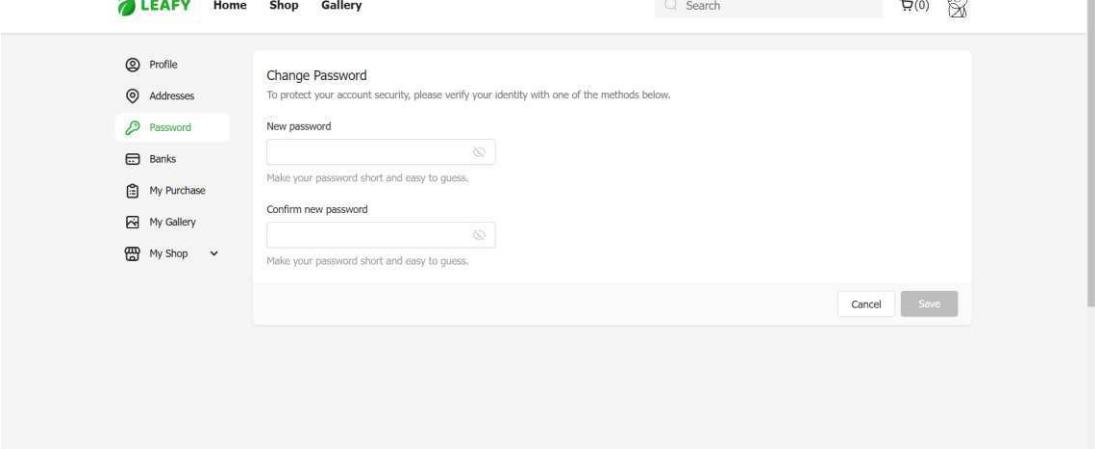
รูปที่ 3.4.1.40 ตัวอย่างข้อมูลที่อยู่หลังเปลี่ยนเป็น Default

3. Change password เป็น Feature สำหรับการเปลี่ยน Password เก่าของผู้ใช้เป็น Password ใหม่ โดยจะต้องมีการยืนยันที่อยู่เก่าก่อนถึงจะสามารถเปลี่ยนเป็นข้อมูลใหม่ได้ ดังแสดงรูปที่ 3.4.1.41 - 3.4.1.42



The screenshot shows the LEAFY application's user interface. At the top, there is a navigation bar with the logo 'LEAFY', 'Home', 'Shop', and 'Gallery'. On the right side of the header, there is a search bar, a shopping cart icon with '(0)', and a user profile icon. The main content area has a sidebar on the left with links: 'Profile', 'Addresses', 'Password' (which is highlighted in green), 'Banks', 'My Purchase', 'My Gallery', and 'My Shop'. The main panel is titled 'Change Password' and contains the following text: 'To protect your account security, please verify your identity with one of the methods below.' Below this is a 'Current password' input field. A red box highlights this field, and a red error message below it says 'Password does not exist.' At the bottom right of the panel is a green 'Next' button.

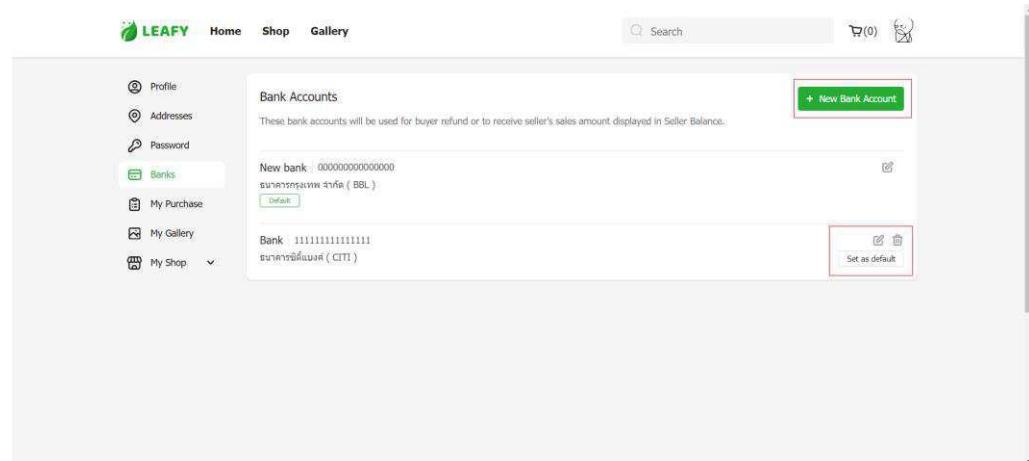
รูปที่ 3.4.1.41 เมื่อผู้ใช้กรอกยืนยันรหัสผ่านเก่าไม่ถูกต้อง



The screenshot shows the 'Change Password' page again. The sidebar on the left is identical to the previous screenshot. The main panel is titled 'Change Password' and contains the following text: 'To protect your account security, please verify your identity with one of the methods below.' Below this are two input fields: 'New password' and 'Confirm new password'. Both fields have placeholder text 'Make your password short and easy to guess.' At the bottom right of the panel are 'Cancel' and 'Save' buttons.

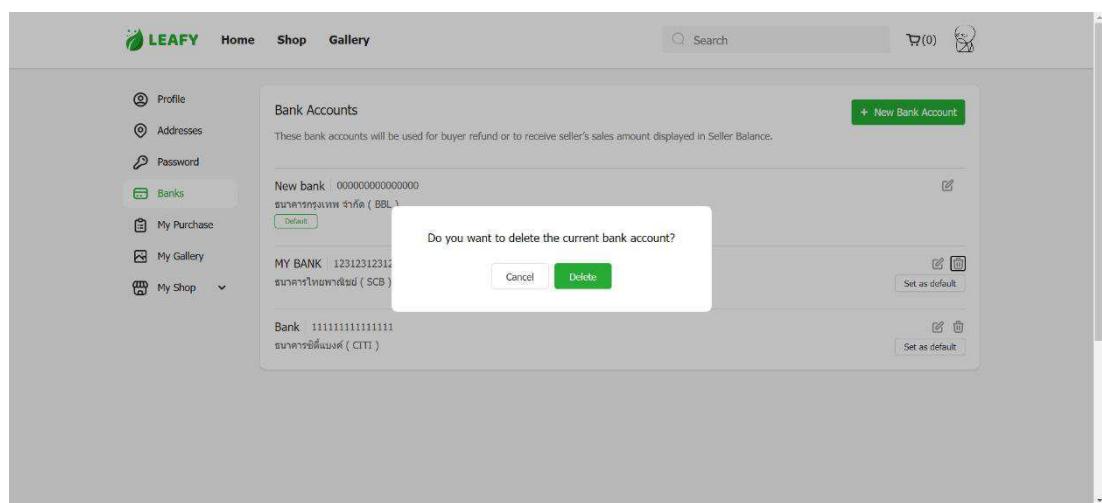
รูปที่ 3.4.1.42 เมื่อผู้ใช้กรอกยืนยันรหัสผ่านเก่าถูกต้อง

4. Bank list และ Bank create เป็น Feature สำหรับจัดการกับบัญชีธนาคารของผู้ใช้งานเพื่อใช้จำลองการชำระเงิน (Coming soon) โดยจะต้องสามารถเพิ่ม แก้ไข หรือลบได้ ดังแสดงรูปที่ 3.4.1.43

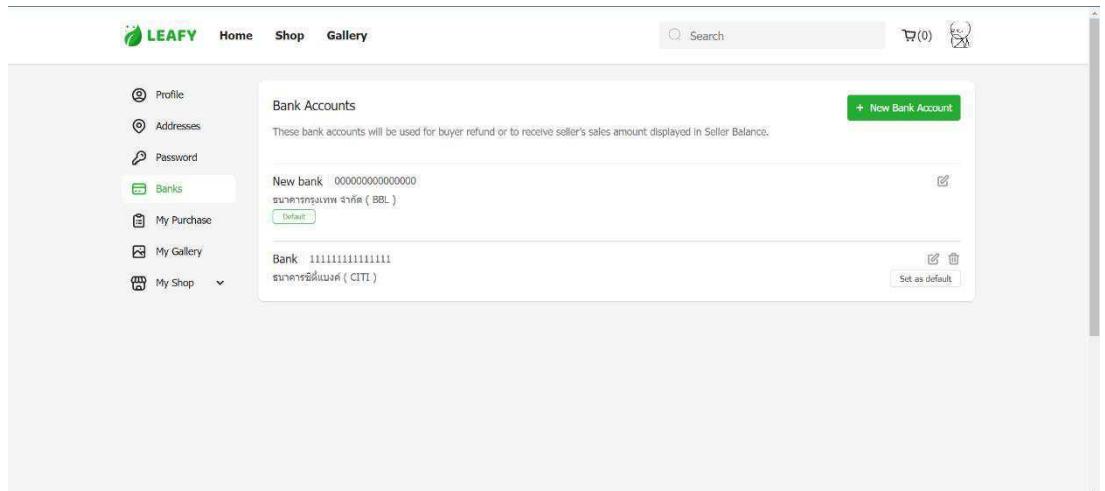


รูปที่ 3.4.1.43 วิธีการจัดการบัญชีธนาคารด้วยการลบ แก้ไข เพิ่ม
หรือตั้งให้เป็น Default

หากผู้ใช้ต้องการที่จะลบข้อมูลบัญชีธนาคารจะต้องมีการแจ้งเตือนให้กับผู้ใช้ทราบ ดังแสดง
รูปที่ 3.4.1.44 - 3.4.1.45

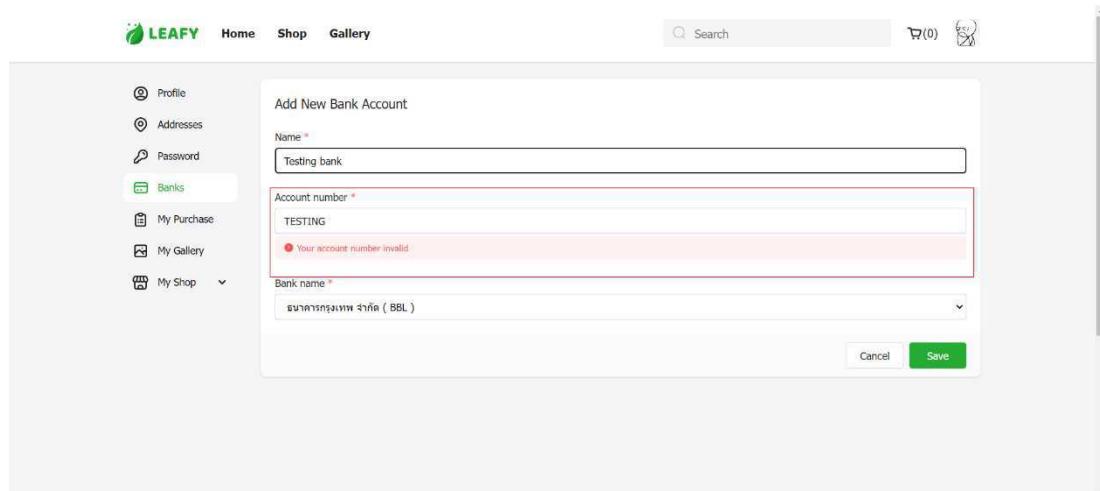


รูปที่ 3.4.1.44 การแจ้งเตือนให้กับผู้ใช้งานทราบ เพื่อยืนยันการลบข้อมูล



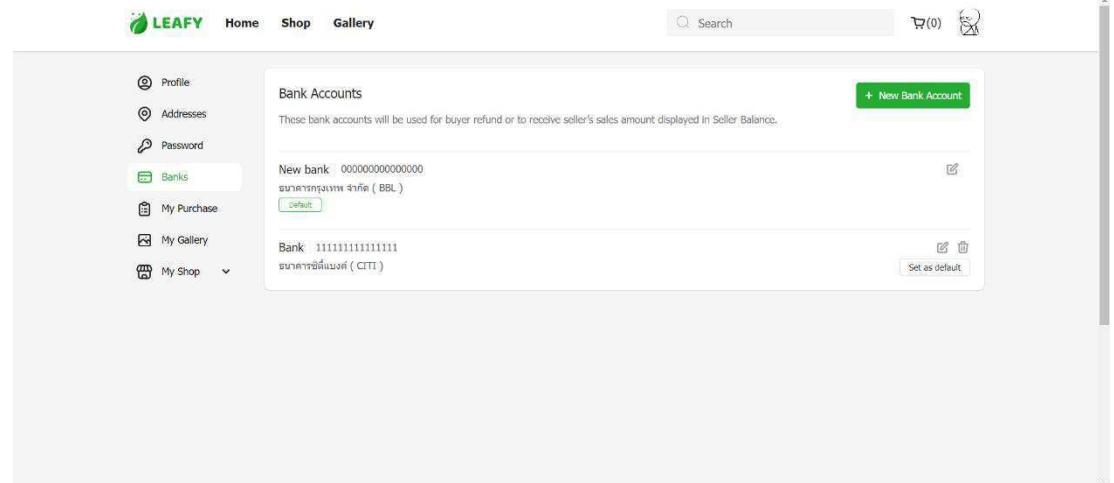
รูปที่ 3.4.1.45 เมื่อข้อมูลบัญชีธนาคารถูกลบจะต้องหายไปจากระบบ

เมื่อผู้ใช้ต้องการสร้างบัญชีธนาคารใหม่หรือแก้ไขจะต้องมีการตรวจสอบความถูกต้องของข้อมูลแล้วเจ้าหน้าที่กับผู้ใช้งานทราบ โดยจะต้องไม่สามารถสร้างหรือแก้ไขได้หากข้อมูลไม่ถูกต้อง ดังแสดงรูปที่ 3.4.1.46

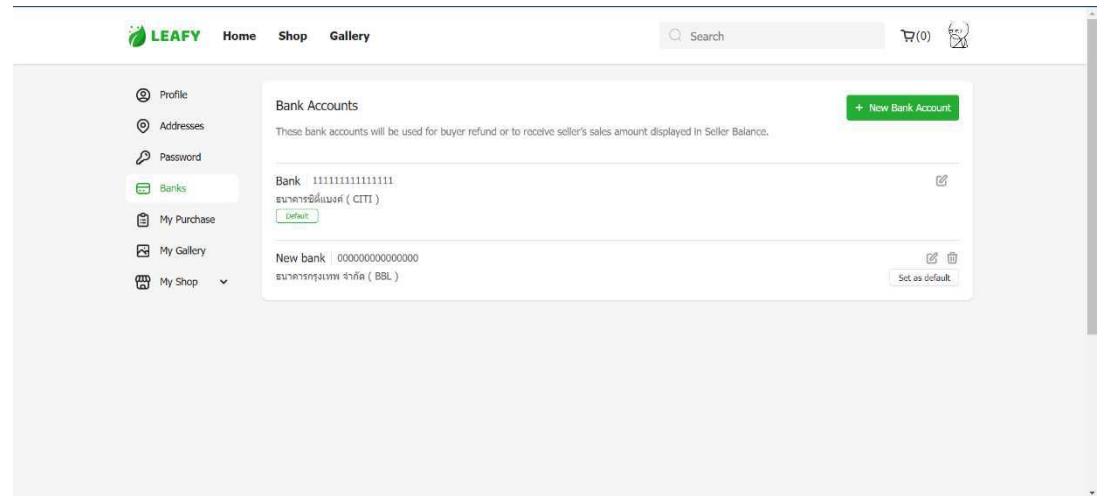


รูปที่ 3.4.1.46 ตัวอย่างการแจ้งเตือนผู้ใช้จากการกรอกข้อมูลผิดพลาด

หากผู้ใช้ต้องการตั้งค่าบัญชีธนาคารอื่นๆ ให้เป็น Default จะต้องสามารถทำได้ ดังแสดงรูปที่ 3.4.1.47 – 3.4.1.48



รูปที่ 3.4.1.47 ตัวอย่างข้อมูลบัญชีธนาคารก่อนเปลี่ยนเป็น Default



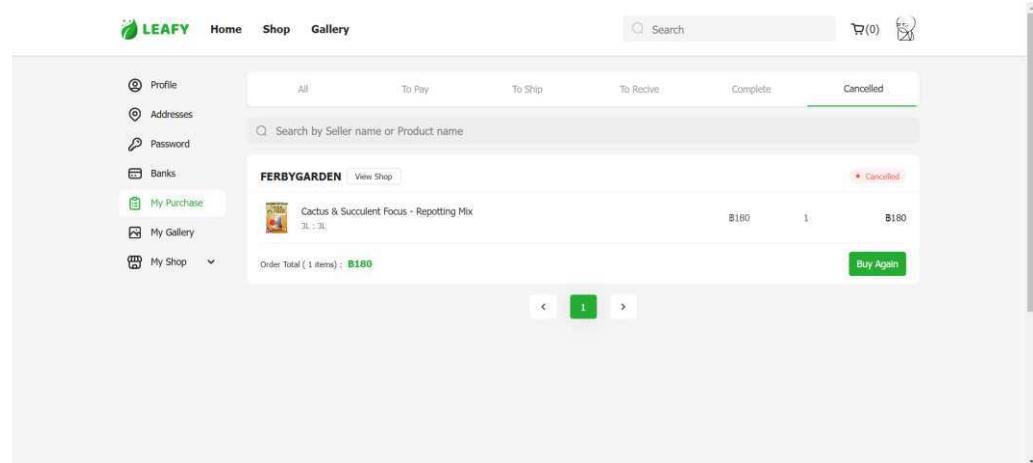
รูปที่ 3.4.1.48 ตัวอย่างข้อมูลบัญชีธนาคารหลังเปลี่ยนเป็น Default

5. My purchase และ My purchase detail เป็น Feature สำหรับรวบรวมข้อมูลเกี่ยวกับสินค้าที่มีการส่งคำร้องไปยังร้านค้าทั้งหมดของผู้ใช้งาน ซึ่งมีการแบ่งสถานะของรายการสินค้า โดยจะต้องสามารถกรอกข้อมูลตามสถานะต่างได้ดังแสดงรูปที่ 3.4.1.49 - 3.4.1.50

รูปที่ 3.4.1.49 ตัวอย่างข้อมูลรายการสินค้าทั้งหมดที่ส่งคำร้องไปแล้ว

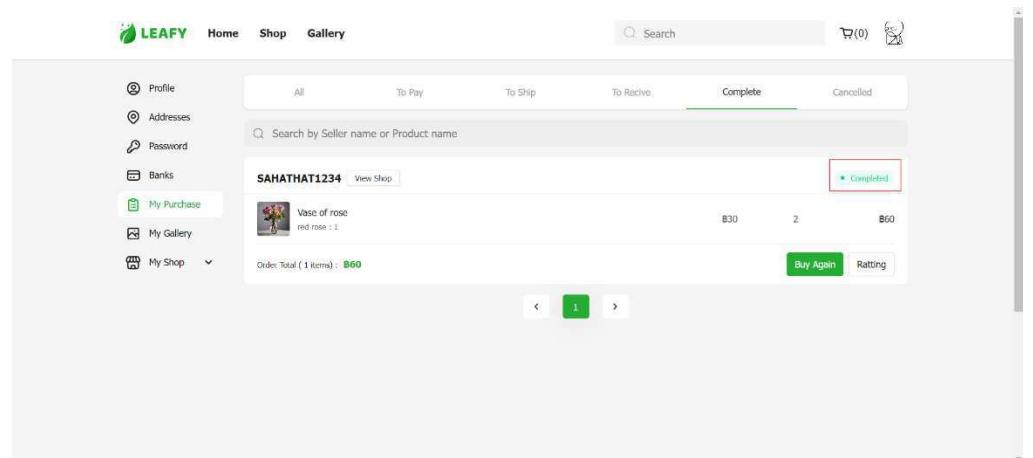
รูปที่ 3.4.1.50 ตัวอย่างข้อมูลรายการสินค้าที่มีสถานะยังไม่ชำระเงิน

หากผู้ใช้ต้องการต้องการยกเลิกรายการสินค้าจะต้องสามารถทำได้ ดังแสดงรูปที่ 3.4.1.51

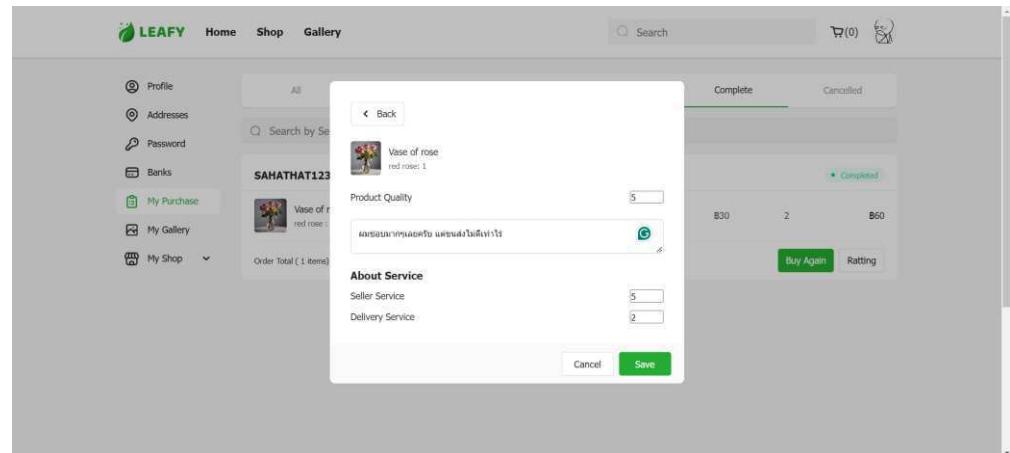


รูปที่ 3.4.1.51 ตัวอย่างข้อมูลที่ผู้ใช้งานยกเลิกคำขอการสินค้า

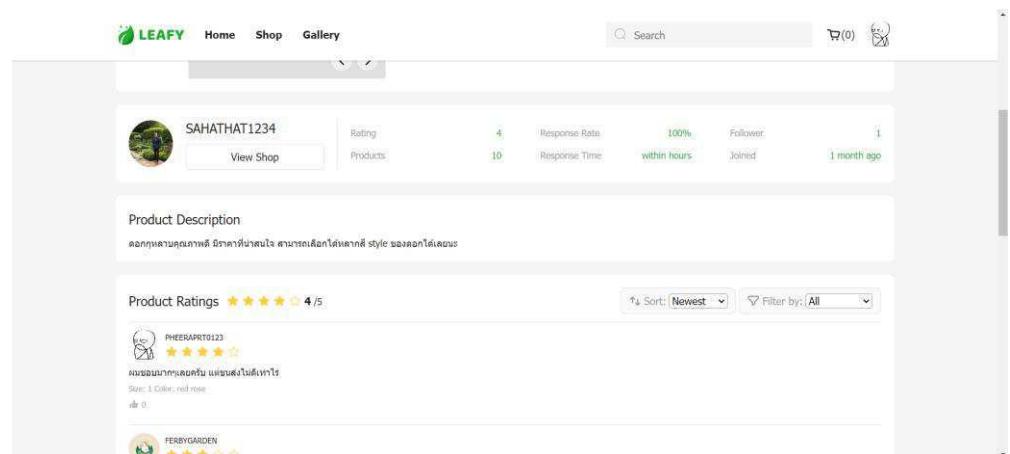
และในกรณีที่ผู้ขายส่งสินค้าถึงผู้ใช้งานแล้วจะต้องสามารถกดปุ่มเพื่อยืนยันการรับสินค้าและสามารถรีวิวสินค้านั้นๆ ได้ ดังแสดงรูปที่ 3.4.1.52 – 3.4.1.54



รูปที่ 3.4.1.52 ตัวอย่างข้อมูลสินค้าที่ถูกรับโดยผู้ใช้สำเร็จ

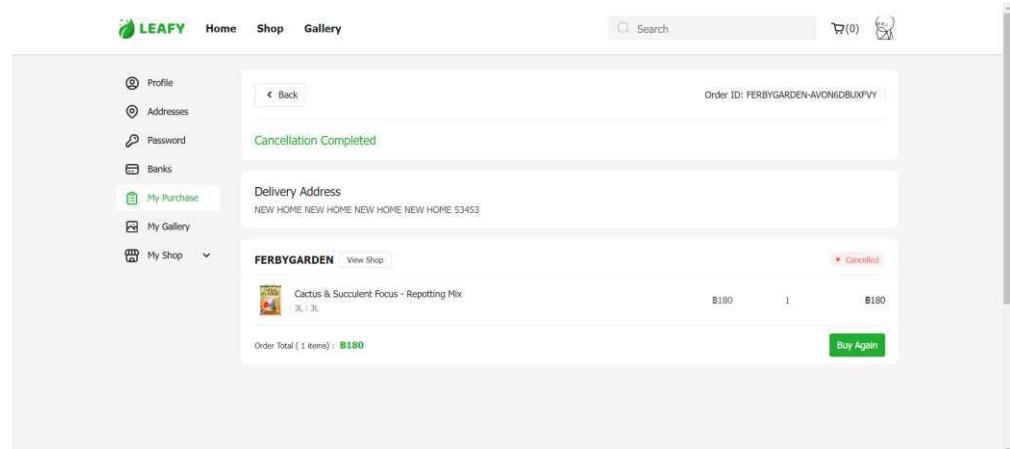


รูปที่ 3.4.1.53 ตัวอย่างการรีวิวสินค้าที่ผู้ใช้งานได้สั่งซื้อสำเร็จ

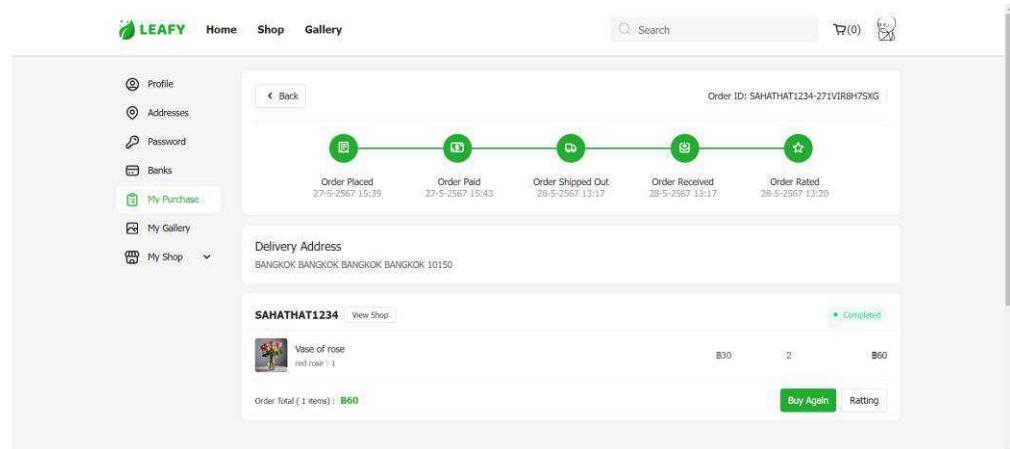


รูปที่ 3.4.1.54 ตัวอย่างการรีวิวสินค้าแสดงบนข้อมูลรายละเอียดสินค้า

หากผู้ใช้ต้องการดูรายละเอียดของคำร้องรายการสินค้า จะต้องสามารถกดเพื่อดูรายละเอียด ดังแสดงรูปที่ 3.4.1.55 – 3.4.1.56



รูปที่ 3.4.1.55 ตัวอย่างคำร้องรายการสินค้าที่ผู้ใช้ได้ทำการยกเลิก



รูปที่ 3.4.1.56 ตัวอย่างคำร้องรายการสินค้าที่ผู้ใช้งานยืนยันรับสินค้าเรียบร้อย

6. My gallery list และ My gallery creates เป็น Feature สำหรับจัดการกับบทความการออกแบบจัดสวนของผู้ใช้งานเพื่อเผยแพร่ความรู้ให้กับชุมชน โดยจะต้องสามารถเพิ่ม แก้ไข หรือลบได้ ดังแสดงรูปที่ 3.4.1.57

Name	Style	Date	Operation
欧洲花园设计	europe	April 29, 2024	
英式花园设计	english	April 29, 2024	
热带花园设计	tropical	April 29, 2024	

รูปที่ 3.4.1.57 วิธีการจัดการบทความการออกแบบจัดสวนด้วยการลบ แก้ไข เพิ่ม

หากผู้ใช้ต้องการที่จะลบข้อมูลบทความจะต้องมีการแจ้งเตือนให้กับผู้ใช้ทราบ
ดังแสดงรูปที่ 3.4.1.58 – 3.4.1.59

Name	Style	Date	Operation
欧洲花园设计	europe	April 29, 2024	

รูปที่ 3.4.1.58 การแจ้งเตือนให้กับผู้ใช้งานทราบ เพื่อยืนยันการลบข้อมูล

#	Name	Style	Date	Operation
1	แบบทั่วไป	europe	April 29, 2024	
2	แบบอังกฤษ	english	April 29, 2024	
3	แบบเขตร้อน	tropical	April 29, 2024	
4	แบบโมเดิร์น	modern		

รูปที่ 3.4.1.59 เมื่อข้อมูลบทความถูกใจต้องหายไปจากระบบ

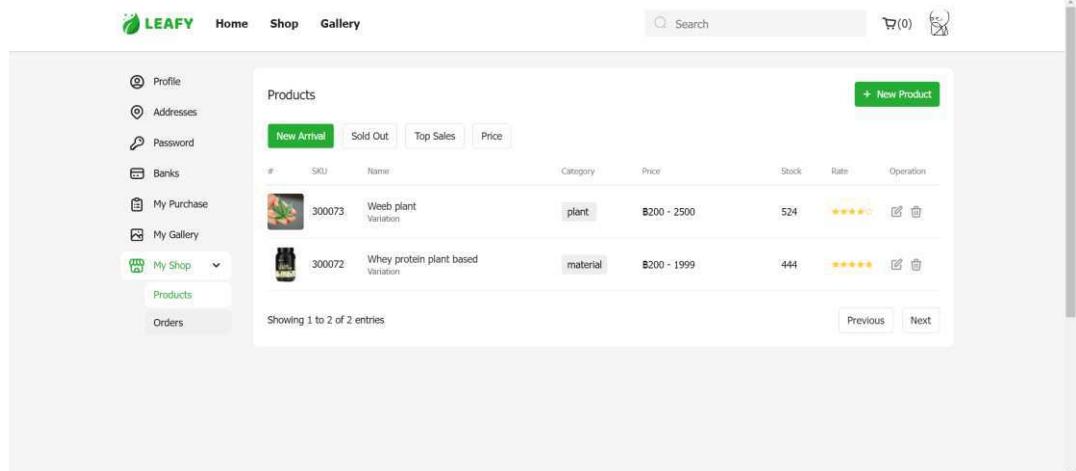
เมื่อผู้ใช้ต้องการสร้างบทความใหม่หรือแก้ไขจะต้องมีการตรวจสอบความถูกต้องของข้อมูลแล้วแจ้งให้กับผู้ใช้งานทราบ โดยจะต้องไม่สามารถสร้างหรือแก้ไขได้หากข้อมูลไม่ถูกต้อง ดังแสดงรูปที่ 3.4.1.60

รูปที่ 3.4.1.60 ตัวอย่างการแจ้งเตือนผู้ใช้จากการกรอกข้อมูลผิดพลาด
หรือขาดข้อมูลที่จำเป็น

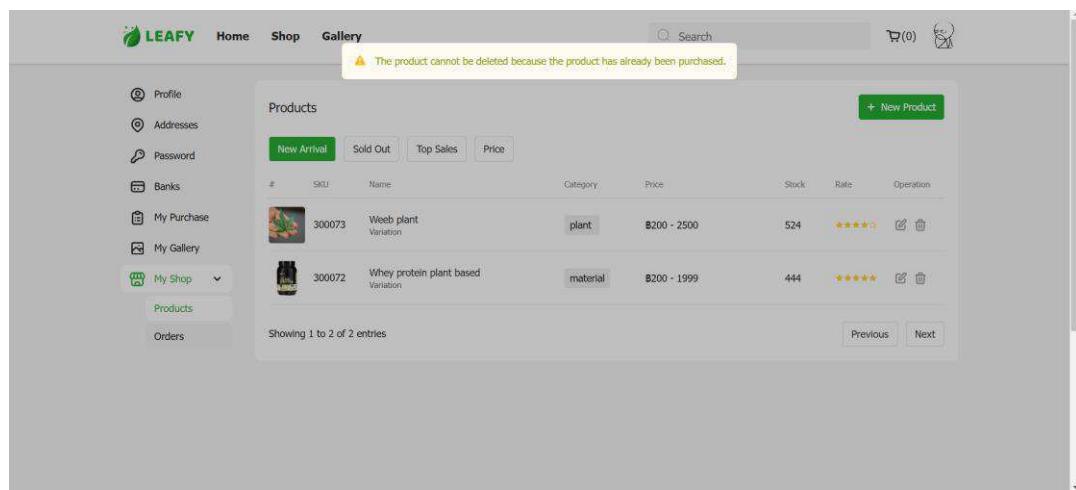
7. Product list และ Product create เป็น Feature สำหรับผู้ใช้ Role Supplier ใช้ในการจัดการเกี่ยวกับสินค้าของผู้ใช้งาน โดยจะต้องสามารถเพิ่ม แก้ไข หรือลบข้อมูลได้ ดังแสดงรูปที่ 3.4.1.61

รูปที่ 3.4.1.61 วิธีการจัดการข้อมูลสินค้าด้วยการเพิ่ม แก้ไข หรือลบหากผู้ใช้งานต้องการลบข้อมูลสินค้าจะต้องสามารถลบได้ โดยข้อมูลสินค้าจะต้องไม่มีคำร้องสั่งซื้อถึงจะลบได้ ดังแสดงรูปที่ 3.4.1.62 - 3.4.1.64

รูปที่ 3.4.1.62 ตัวอย่างการแจ้งเตือนผู้ใช้งาน เมื่อต้องการลบข้อมูลสินค้า



รูปที่ 3.4.1.63 ตัวอย่างการลบข้อมูลสินค้าออกจากระบบโดยผู้ใช้งาน



รูปที่ 3.4.1.64 ตัวอย่างการลบข้อมูลที่มีคำร้องการสั่งซื้อยู่

หากผู้ใช้ต้องการที่จะสร้างข้อมูลสินค้าใหม่หรือแก้ไข จะต้องสามารถแจ้งเตือนให้กับผู้ใช้งานทราบ เมื่อผู้ใช้กรอกข้อมูลผิด ดังแสดงรูปที่ 3.4.1.65 – 3.4.1.67

Name*
Name: *
Description
8. **ສອກຄາມເຄີຍເຄືອຂໍ້ຕົວຢ່າງນີ້ແລ້ວສະໜອງໃຫຍ່ເພື່ອການທີ່ຈີ່ຕົວຢ່າງນີ້ແລ້ວສະໜອງໃຫຍ່ເພື່ອການທີ່ມີມາການຊັບ
ການໃຫ້ເຖິງຄູ່ອາວາຈວ່າໄດ້ໂຄດນແລ້ວຮອດຕ່າງໆ ເພື່ອວັນນີ້ຂອງຕົວຢ່າງນີ້ແລ້ວສະໜອງໃຫຍ່ເພື່ອການທີ່ມີມາການຊັບ
ການຮັບຮອດຕ່າງໆ ແລ້ວໃຫ້ເວັນນີ້ຂອງຕົວຢ່າງນີ້ແລ້ວສະໜອງໃຫຍ່ເພື່ອການທີ່ມີມາການຊັບ
ການຮັບຮອດຕ່າງໆ ແລ້ວໃຫ້ເວັນນີ້ຂອງຕົວຢ່າງນີ້ແລ້ວສະໜອງໃຫຍ່ເພື່ອການທີ່ມີມາການຊັບ
ການຮັບຮອດຕ່າງໆ

Cover Photo*

Category*
Plant

Tag
plant

● please input your product name

Cancel Save

ຮູບທີ 3.4.1.65 ຕ້ວຍ່າງການແຈ້ງເຕືອນ ໡ື້ອື່ນໃຊ້ກຣອກຂໍ້ມູນຜິດພາດ
ທີ່ອາດຂໍ້ມູນທີ່ຈໍາເປັນ

LEAFY Home Shop Gallery

Add New Product

Name*
Name: New product

Description
New product

Cover Photo*

Category*
Material

Tag
tag1, tag2...

ຮູບທີ 3.4.1.66 ຕ້ວຍ່າງການກຣອກຂໍ້ມູນເພື່ອເພີ່ມຂໍ້ມູນສິນຄ້າໃໝ່

#	SKU	Name	Category	Price	Stock	Rate	Operation
1	300076	New product Variation	material	\$10 - 100	119		
2	300073	Webb plant Variation	plant	\$200 - 2500	524		
3	300072	Whey protein plant based Variation	material	\$200 - 1999	444		

Showing 1 to 3 of 3 entries

ຮູບທີ 3.4.1.67 ຕ້ວຍ່າງຂໍ້ມູນສິນຄ້າທີ່ເພີ່ມໃໝ່

8. Order list เป็น Feature สำหรับ Role Supplier จัดการเกี่ยวกับคำร้องการสั่งซื้อสินค้าของผู้ใช้งานอื่นๆ เพื่อให้ผู้ขายสามารถเตรียมสินค้าและจัดส่งให้กับผู้สนใจได้ โดยจะต้องสามารถเปลี่ยนสถานะของคำร้องสำหรับการจัดส่งเท่านั้น ดังแสดงรูปที่ 3.4.1.68 - 3.4.1.69

#	Order ID	Customer	Address	Date	Price	Status
1	PHEERAP...	TEST123	If a b ok 10150	13-5-2567 1...	\$200	Pending In Progress
#	SKU	Name		Price	Qty	Total
	Tu	Weed plant Tu : 1 Tu		\$200	1	\$200
				Subtotal		\$200
				Shipping		\$ 0
				Tax		\$ 0
				Total Payment		B \$200
				Name: TEST123		
				Phone: 23232323232		
				Address: If a b ok 10150		
2	PHEERAP...	DDAY	KMUTT (มหาวิทยาลัย ท่าข่าย กรุงเทพฯ)	29-4-2567 1...	\$2500	Completed
3	PHEERAP...	DDAY	KMUTT (มหาวิทยาลัย ท่าข่าย กรุงเทพฯ)	29-4-2567 1...	\$700	Completed

รูปที่ 3.4.1.68 ตัวอย่างคำร้องสั่งซื้อสินค้าที่ผู้ขายพร้อมจัดส่งสินค้า

#	Order ID	Customer	Address	Date	Price	Status
1	PHEERAP...	TEST123	If a b ok 10150	13-5-2567 1...	\$200	In Progress
#	SKU	Name		Price	Qty	Total
	Tu	Weed plant Tu : 1 Tu		\$200	1	\$200
				Subtotal		\$200
				Shipping		\$ 0
				Tax		\$ 0
				Total Payment		B \$200
				Name: TEST123		
				Phone: 23232323232		
				Address: If a b ok 10150		
2	PHEERAP...	DDAY	KMUTT (มหาวิทยาลัย ท่าข่าย กรุงเทพฯ)	29-4-2567 1...	\$2500	Completed
3	PHEERAP...	DDAY	KMUTT (มหาวิทยาลัย ท่าข่าย กรุงเทพฯ)	29-4-2567 1...	\$700	Completed

รูปที่ 3.4.1.69 ตัวอย่างคำร้องสั่งซื้อสินค้าที่กำลังไปถึงผู้ซื้อสินค้า

จะต้องสามารถแบ่งประเภทของสถานะต่างๆ เพื่อให้ผู้ขายสามารถตรวจสอบได้ร่ายชื่นและสามารถเลือกดูช่วงเวลาของคำร้องได้ ดังแสดงรูปที่ 3.4.1.70 – 3.4.1.71

#	Order ID	Customer	Address	Date	Price	Status
1	PHEERAP...	DDAY	KMUTT เอกมานาจฉัพท์ยิ่ง ทำร่อง กรุงเทพฯ...	29-4-2567 1...	\$2500	Completed ✓
2	PHEERAP...	DDAY	KMUTT เอกมานาจฉัพท์ยิ่ง ทำร่อง กรุงเทพฯ...	29-4-2567 1...	\$700	Completed ✓
3	PHEERAP...	SAHATHAT1234	99 ถ. 15 หมู่ 9 บ้านชุมชนอิน หมู่ที่ 10140	29-4-2567 1...	\$3998	Completed ✓
4	PHEERAP...	DDAY	KMUTT เอกมานาจฉัพท์ยิ่ง ทำร่อง กรุงเทพฯ...	29-4-2567 1...	\$700	Completed ✓
5	PHEERAP...	DDAY	KMUTT เอกมานาจฉัพท์ยิ่ง ทำร่อง กรุงเทพฯ...	29-4-2567 1...	\$4200	Completed ✓
6	PHEERAP...	SAHATHAT1234	99 ถ. 15 หมู่ 9 บ้านชุมชนอิน หมู่ที่ 10140	29-4-2567 1...	\$1400	Completed ✓
7	PHEERAP...	DDAY	KMUTT เอกมานาจฉัพท์ยิ่ง ทำร่อง กรุงเทพฯ...	29-4-2567 1...	\$200	Completed ✓
8	PHEERAP...	DDAY	KMUTT เอกมานาจฉัพท์ยิ่ง ทำร่อง กรุงเทพฯ...	29-4-2567 1...	\$200	Completed ✓

รูปที่ 3.4.1.70 ตัวอย่างการกรองตามประเภทของสถานะคำร้องสั่งซื้อ

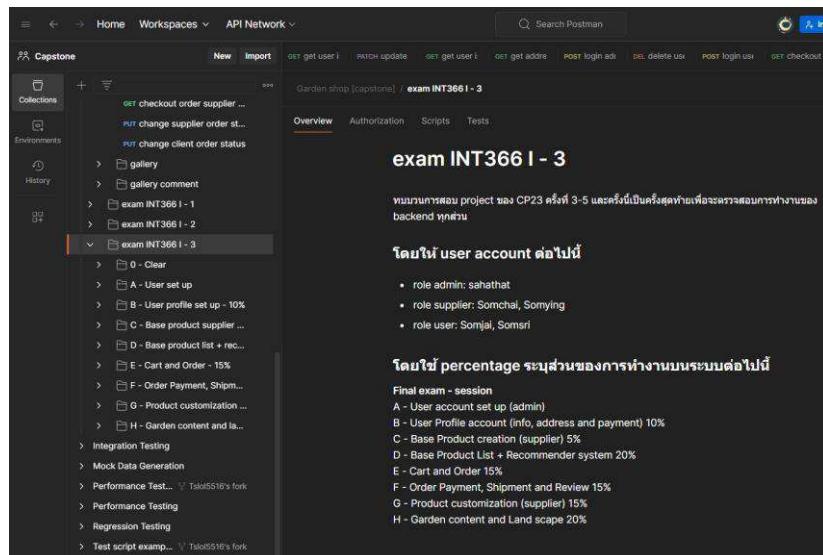
#	Order ID	Customer	Address	Date	Price	Status
1	PHEERAP...	TEST123	if a b ak 10150	13-5-2567 1...	\$200	In Progress

รูปที่ 3.4.1.71 ตัวอย่างการกรองตามช่วงเวลาคำร้องสั่งซื้อ

3.4.2. การทดสอบส่วน Backend จะใช้ Postman [24] เพื่อทำการทดสอบโดยการส่งคำขอและรับคำตอบ จะมีการทดสอบทั้งหมด 4 รอบ โดยเน้นการตรวจสอบความถูกต้องของข้อมูลที่ส่งคำขอและการตอบกลับ โดยในการทดสอบข้อมูลที่มีโครงสร้างที่ซับซ้อน เช่นระบบ Recommendation ที่มีข้อมูลขนาดใหญ่ ซึ่งจัดการกับการออกแบบกรณีทดสอบไว้ดังนี้ รอบแรกจะเป็นการทดสอบอัตโนมัติ 2 ครั้ง และการทดสอบด้วยวิดีโอ 1 ครั้ง โดยทำการทดสอบเมื่อกำลังดำเนินการความคืบหน้าของโครงการ และรอบสุดท้ายจะใช้ในการทดสอบฟีเจอร์ทุกส่วนของแอป Leafy โดยใช้การทดสอบ End-to-End แบบ Manual โดยอัดวิดีโอ [33] ทำให้มีผลต่อการทำงานในฝั่ง Frontend ในการจัดการข้อมูล Request และ Response ได้อย่างถูกต้อง ซึ่งผลลัพธ์ที่ได้ตาม [33] โดยแบ่งเป็น 8 ส่วน ดังต่อไปนี้:

1. User Account Set Up (Admin)
2. User Profile Account (Info, Address and Payment)
3. Base Product Creation (Supplier)
4. Base Product List + Recommender System
5. Cart and Order
6. Order Payment, Shipment and Review
7. Product Customization (Supplier)

และ 8. Garden Content and Landscape ดังรูปที่ 3.4.2.1



รูปที่ 3.4.2.1 แบบกรณีทดสอบรอบสุดท้ายของ Leafy E-commerce

บทที่ 4 ผลการดำเนินการ

จากการพัฒนาเว็บแอปพลิเคชัน Leafy ซึ่งในระหว่างการพัฒนามีปัญหาและอุปสรรคเช่น การสื่อสาร การทดลอง Spike/POC เพื่อทดสอบองค์ความรู้และความเป็นไปได้ของเครื่องมือที่เลือกใช้งาน และเวลาในการพัฒนาของสมาชิกในทีมที่ไม่ตรงกัน ทำให้เวลาสำหรับการพัฒนาลดลง ส่งผลให้การทำ Project มีความล่าช้า แต่สามารถแก้ไขปัญหาดังกล่าวได้ ทำให้การพัฒนาเป็นไปได้ด้วยดีและสามารถนำเสนอในวัน D-Day ได้ ซึ่งเป็นไปตามแผนที่กำหนดไว้ในการพัฒนา

4.1 ผลการดำเนินการ

1. Team Retrospective

1. รอบที่ 1 และรอบที่ 2

ปัญหา: ทีมมีการทำงานโดยยึดตามแผนมากเกินไป จึงทำให้การดำเนินงานไม่เป็นไปตามเป้าหมายที่วางไว้และทำให้พัฒนาได้ช้ากว่าเวลากำหนด

วิธีแก้ไขปัญหา: ปรับรูปแบบการทำงานและเพิ่มการสื่อสารระหว่างสมาชิกภายในทีมให้มากขึ้น เพื่อช่วยกันแก้ไขปัญหาของสมาชิกในทีม

2. รอบที่ 3 และรอบที่ 4

ปัญหา: เกิดจากคนในทีมมีช่วงเวลาการทำงานที่ต่างกัน ทำให้งานที่ได้ไม่สามารถสำเร็จตามเป้าหมายที่วางไว้และทำให้เกิดงานที่ค้างจากเป้าหมายครั้งก่อน

วิธีแก้ไขปัญหา: กำหนดเวลาการประชุมให้ชัดเจนและสอดคล้องสำหรับทุกคน เพื่อทำให้สมาชิกในทีมไม่รู้สึกกดดันจากการประชุมมากเกินไป และให้สมาชิกในทีมสามารถเลือกเวลาพัฒนาด้วยตัวเองได้ โดยจะต้องส่งงานภายในวันที่ประชุมตามที่ตกลงกันไว้ เพื่อให้ผลงานที่ได้มีประสิทธิภาพมากขึ้น

2. ผลการทดสอบ

พบว่าทีมของเราไม่สามารถดำเนินการตามแผนที่เราคาดหวังไว้ได้ ดังนั้นเราจึงต้องตัดคุณสมบัติหรือ Feature บางอย่างออกไปอย่างเช่น ระบบสำหรับติดตามสินค้า เมื่อลูกค้าได้สั่งซื้อสินค้าจากผู้ขายสินค้า ระบบส่งข้อความระหว่างผู้ขายกับผู้ซื้อ เป็นต้น ถึงแม้ว่าจะไม่สามารถพัฒนาได้ตามที่คาดหวัง แต่ทีมของเราสามารถพัฒนา Feature ที่จำเป็นของระบบขายสินค้าออนไลน์และสามารถนำเสนอผลงานในวัน D-Day ได้ และในอนาคตเราสามารถนำ Leafy มาพัฒนาต่อยอดและนำไปใช้กับธุรกิจอื่นๆ ได้

ເອກສາຣ້ອງອີງ

- [1] Threeroots Group Co., Ltd., 2022, “Landscape Architect and Consultants”, Threeroots [Online], Available: <https://three-roots.com> [2023, July 29].
- [2] Gingganbai Group Co., Ltd., 2021, “Landscape and Garden”, Gingganbai [Online], Available: <https://www.ginggaanbai.com> [2023, July 29].
- [3] Add Green in Garden Co., Ltd., “Landscape and Garden Design”, Addgreeningarden [Online], Available: <http://addgreeningarden.com> [2023, July 29].
- [4] ສວນລຸ່ມບຸ້ມູນ, 2017, “Landscape Architecture”, Uncleboongarden [Online], Available: <https://uncleboongarden.com> [2023, July 29].
- [5] Shopee (Thailand) Co., Ltd., 2015, “Online Store”, Shopee [Online], Available: <https://shopee.co.th> [2023, August 9].
- [6] Lazada, Ltd., 2012, “Online Store”, Lazada [Online], Available: <https://lazada.co.th/> [2023, August 9].
- [7] Fastwork Technologies Co., Ltd., 2016, “Freelances Platform”, Fastwork [Online], Available: <https://fastwork.co/> [2023, August 17].
- [8] Grafolio Pte., Ltd., 2019 “Creative Contents Community”, Grafolio [Online], Available: <https://grafolio.naver.com/> [2023, August 17].
- [9] Pinterest Trading Co., Ltd., 2014, “Style Inspiration”, Pinterest [Online], Available: <https://www.pinterest.com> [2023, August 17].
- [10] Emil Lamprecht, 2023, June, “The Difference Between UX and UI Design: A Beginner’s Guide”, Available: <https://careerfoundry.com/en/blog/ux-design/the-difference-between-ux-and-ui-design-a-laymans-guide/> [2023, August 20].
- [11] Ravneet Singh, 2022, November 11, “What Makes a Good UX/UI Design?”, Available: <https://www.netsolutions.com/insights/good-ux-ui-design/> [2023, August 20].
- [12] MIT License, 2014-2023, “Vuejs”, Vuejs [Online], Available: <https://vuejs.org/> [2023, August 17].
- [13] Yizhen Zhao, 2021, March 17, “Machine Learning in Production: A Literature Review”, <https://staff.fnwi.uva.nl/a.s.z.belloum/LiteratureStudies/Reports/2021-LiteratureStudy-report-Yizhen.pdf> [2023, September 6].

- [14] W3.CSS, Copyright 1999-2023, “w3schools”, w3schools [Online], Available: <https://www.w3schools.com/> [2023, September 7].
- [15] MinIO, Copyright 2016-2023, “MinIO Object Storage for Container”, MinIO [Online], Available: <https://min.io/docs/minio/container/index.html> [2023, September 6].
- [16] Sirinart Tangruamsub, 2019, Dec 1, “Recommendation System (ແບບສຽນມີຄ່ອຍສັ້ນ)”, Medium [Online], Available: [https://medium.com/@sinart.t/recommendation-system-%E0%B9%81%E0%B8%9A%E0%B8%9A%E0%B8%AA%E0%B8%A3%E0%B8%B8%E0%B8%9B%E0%B9%80%E0%B8%AD%E0%B8%87-ce6246f49754](https://medium.com/@sinart.t/recommendation-system-%E0%B9%81%E0%B8%9A%E0%B8%9A%E0%B8%AA%E0%B8%A3%E0%B8%B8%E0%B8%9B%E0%B9%80%E0%B8%AD%E0%B8%B2%E0%B9%80%E0%B8%AD%E0%B8%87-ce6246f49754) [2023, September 8].
- [17] Skooldio Co., Ltd., 2021, February 2, “Microservices vs Monolithic ບຣິ່ຫຼັກມະນາກົມໂລກໃຈໆໄວ້?”, Skooldio [Online], Available: <https://blog.skooldio.com/microservices-vs-monolithic/> [2023, September 7].
- [18] Frank Joseph, 2022, October 19, “Building microservices with Node.js”, LogRocket [Online], Available: <https://blog.logrocket.com/building-microservices-node-js/> [2023, September 7].
- [19] MIT License, Copyright 2015-2023, “npm-argon2 package”, Npmjs [Online], Available: <https://www.npmjs.com/package/argon2> [2023, August 25].
- [20] ພັກ ພະພລສື່ງ (ໄດ້ຮັບຕຳມາດ), ມកຣາຄມ 1, 2023, “AAA ກັບຄວາມສຳຄັນໃນ Network Security”, Cyberdrive [Online], Available: <https://www.cyberdrive.in.th/aaa-%E0%B8%81%E0%B8%B1%E0%B8%9A%E0%B8%84%E0%B8%A7%E0%B8%B2%E0%B8%A1%E0%B8%AA%E0%B8%B3%E0%B8%84%E0%B8%B1%E0%B8%8D%E0%B9%83%E0%B8%99-network-security/> [2023, September 8].
- [21] International Business Machines Corporation (IBM), Copyright 1911-2023, Published by 2023, May 31, “Digital signatures in SSL/TLS”, IBM [Online], <https://www.ibm.com/docs/en/ibm-mq/9.1?topic=tls-digital-signatures-in-ssltls> [2023, September 8].
- [22] Davit Vardanyan, 2021, July 25, “How Many Requests Can a Real-World Node.js Server-Side App Handle?”, plainenglish.io [Online], <https://plainenglish.io/blog/how-many-requests-can-handle-a-real-world-nodejs-server-side-application> [2023 November 1].

- [23] Apache benchmark, Copyright 2024, “ab - Apache HTTP server benchmarking tool”, Apache [Online], <https://httpd.apache.org/docs/2.4/programs/ab.html> [2023 November 1].
- [24] Postman automates testing, Copyright 2024, “Automated testing with Postman”, Postman [Online], <https://www.postman.com/automated-testing/> [2023 December 1].
- [25] Prisma, © 2024 Prisma Data, Inc., “Why Prisma ORM?”, Prisma [Online], <https://www.prisma.io/docs/concepts/overview/why-prisma> [2023 August 8].
- [26] Abbas Aslanbay, 2023, Feb 21, “Uploading Files to AWS S3 with Multer and the Node.js AWS SDK”, Medium [Online], <https://abbaslanbay.medium.com/uploading-files-to-aws-s3-with-multer-and-the-node-js-aws-sdk-7cad8dc87fc2> [2023 October 23].
- [27] Coding Nexus, 2023, January 15, “Master YouTube Algorithm Cloning - Unlock Secrets with NodeJS JavaScript!”, Youtube [Online], <https://www.youtube.com/watch?v=7luxGqFFEzM> [2024 February 5].
- [28] Classic models Database, Copyright © 2008 - Present by www.mysqltutorial.org. All Rights Reserved, “MySQL Sample Database”, mysqltutorial [Online], <https://www.mysqltutorial.org/getting-started-with-mysql/mysql-sample-database/> [2024 February 20].
- [29] Shanika Wickramasinghe, 2020, Nov 12, “How Do You Design a Shopping Cart Database for E-Commerce?”, Fabric blog [Online], <https://fabric.inc/blog/commerce/shopping-cart-database-design> [2024 February 20].
- [30] Kim Karun, © Copyright CLICKNEXT Company limited, 2019, “SKU គឹន មាន តិច? តើ តុក្ខុណីទីផ្សេងៗ e-Commerce តាំងបានតួងអី!”, MakeWebEasy [online], <https://www.makewebeeasy.com/th/blog/what-is-sku/> [2024 March 15].
- [31] Mayukh Bhattacharyya, 2022, Dec 7, “Metrics of Recommender Systems: An Overview”, Medium [online], <https://towardsdatascience.com/metrics-of-recommender-systems-cde64042127a> [2023 September 6].
- [32] Google developer, © Copyright 2024 Google Inc., “Content-based Filtering”, Google developer [online], <https://developers.google.com/machine-learning/recommendation/content-based/basics?hl=th> [2023 September 6].

- [33] Sahathat Yingsakulkiet, 2024, April 28, “Final Manual Testing on Postman Part 1 - 3”, Microsoft stream [Online],
https://mailkmuttach.sharepoint.com/:v/s/Project450/EfxVwcLNJjBIrXeAxGBe5tEBhZLCJQMGxorC_RDwPhS0Qw?e=S0p8XI,
<https://mailkmuttach.sharepoint.com/:v/s/Project450/Eby--p00REZCmXgeLKZXulsBJV61uiyCM7u9WdCceEI7IA?e=jNdlew>,
https://mailkmuttach.sharepoint.com/:v/s/Project450/EasFBoZuPr9DpOITTMyfPxwBi_d6wTnXUgc9AZ_NfcJeXQ?e=ilWnVs [2024 April 28].
- [34] "Shopee", แพลตฟอร์มอีคอมเมิร์ซ [Online], Available: <https://www.shopee.com> [2023, September 1].
- [35] "Nike", เว็บไซต์อีคอมเมิร์ซเกี่ยวกับสินค้าและอุปกรณ์กีฬา [Online], Available: <https://www.nike.com> [2023, September 1]



มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
ข้อตกลงว่าด้วยการโอนลิขสิทธิ์การศึกษาโครงการแพทย์เรื่อง

วันที่...1.....เดือน...กรกฎาคม...พ.ศ...2567..

ข้าพเจ้า (นางสาว)...ปนาลี ปาลศรี...รหัสประจำตัว...63130500075....

อยู่บ้านเลขที่.....11/39.....ตรอก/ซอย.....พิกัด2.....ถนน.....แขวงวัฒนา 14.....

ตำบล/แขวง.....ทุ่งสองห้อง.....อำเภอ/เขต.....หลักสี่.....จังหวัด.....กรุงเทพมหานคร.....

รหัสไปรษณีย์...10210...โทรศัพท์0992309111..... E-mail:..... fernsat45@gmail.com..... และ

ข้าพเจ้า (นาย).....พีรพัฒน์ ค้าเกิด.....รหัสประจำตัว...63130500088...

อยู่บ้านเลขที่.....6.....ตรอก/ซอย...เทียนทะเล 23 แยก 2.....ถนน...บางบุนเทียน ขายทะเล.....

ตำบล/แขวง.....ท่าข้าม.....อำเภอ/เขต.....บางบุนเทียน.....จังหวัด...กรุงเทพมหานคร.....

รหัสไปรษณีย์...10150...โทรศัพท์ ...0637543325.... E-mail:..... pheeraprt0123@gmail.com..... และ

ข้าพเจ้า (นาย)....สหทัศน์ ยิ่งสกุลเกียรติ...รหัสประจำตัว...63130500113....

อยู่บ้านเลขที่.....19.....ตรอก/ซอย.....ท่าข้าม 28.....ถนน.....พระราม 2.....

ตำบล/แขวง.....แสมดำ.....อำเภอ/เขต.....บางบุนเทียน.....จังหวัด.....กรุงเทพมหานคร.....

รหัสไปรษณีย์...10150...โทรศัพท์ ...0927716621... E-mail:..... sahathat.44@mail.kmutt.ac.th.....

เป็นนักศึกษาของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชา

เทคโนโลยีสารสนเทศ คณะ เทคโนโลยีสารสนเทศ ขอโอนลิขสิทธิ์การศึกษาโครงการแพทย์เรื่องให้

ไว้กับมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี โดยมี

ผศ.ดร.ณรงค์ฤทธิ์ วรารถน์ ตำแหน่ง คณบดี เป็นผู้รับโอนลิขสิทธิ์และมีข้อตกลงดังนี้

1. ข้าพเจ้าได้จัดทำการศึกษาโครงการแพทย์เรื่อง เรื่อง ระบบชี้อักษรินค์ออนไลน์เกี่ยวกับสวน และแนวการจัดสวน (Leafy) ซึ่งอยู่ในความควบคุมของ อาจารย์พิเชฐฐ์ ลิ่มวิรานันต์ ตามมาตรา 14 แห่ง พ.ร.บ. ลิขสิทธิ์ พ.ศ. 2537 และถือว่าเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรของมหาวิทยาลัย เทคโนโลยีพระจอมเกล้าธนบุรี

2. ข้าพเจ้าตกลงโอนลิขสิทธิ์จากผลงานทั้งหมดที่เกิดขึ้นจากการสร้างสรรค์ของข้าพเจ้าใน การศึกษาโครงการแพทย์เรื่องให้กับมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ตลอดอายุแห่งการ คุ้มครองลิขสิทธิ์ตามมาตรา 23 แห่งพระราชบัญญัติลิขสิทธิ์ พ.ศ. 2537 ตั้งแต่วันที่ได้รับอนุญาตโครงการ การศึกษาโครงการแพทย์เรื่องจากมหาวิทยาลัย

3. ในกรณีที่ข้าพเจ้าประสงค์จะนำการศึกษา โครงการแพทย์เรื่องไปใช้ในการเผยแพร่ในสื่อใด ๆ ก็ตาม ข้าพเจ้าจะต้องระบุว่าการศึกษาโครงการแพทย์เรื่องเป็นผลงานของมหาวิทยาลัยเทคโนโลยีพระ จอมเกล้าธนบุรีทุก ๆ ครั้งที่มีการเผยแพร่

4. ในกรณีที่ข้าพเจ้าประสงค์จะนำการศึกษาโครงการแพทย์เรื่องไปเผยแพร่หรืออนุญาต

ให้ผู้อื่นทำซ้ำหรือดัดแปลงหรือเผยแพร่ต่อสาธารณะนหรือกระทำการอื่นใด ตามมาตรา 27 มาตรา 28 มาตรา 29 และมาตรา 30 แห่งพระราชบัญญัติลิขสิทธิ์ พ.ศ. 2537 โดยมีค่าตอบแทนในเชิงธุรกิจ ข้าพเจ้าจะกระทำได้เมื่อได้รับความยินยอมเป็นลายลักษณ์อักษรจากมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

5. ในกรณีที่ข้าพเจ้าประสงค์จะนำข้อมูลจากวิทยานิพนธ์ไปประดิษฐ์หรือพัฒนาต่อยอดเป็นสิ่งประดิษฐ์หรืองานทรัพย์สินทางปัญญาประเภทอื่น ภายในระยะเวลาสิบ (10) ปีนับจากลงนามในข้อตกลงฉบับนี้ ข้าพเจ้าจะกระทำได้เมื่อได้รับความยินยอมเป็นลายลักษณ์อักษรจากมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี และมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรีมีสิทธิ์ในทรัพย์สินทางปัญญานี้ พร้อมกับได้ชำระค่าตอบแทนการอนุญาตให้ใช้สิทธิดังกล่าว รวมถึงการจัดสรรผลประโยชน์อันพึงเกิดขึ้นจากส่วนหนึ่งหรือทั้งหมดของวิทยานิพนธ์ในอนาคต โดยให้เป็นไปตามระเบียบสถาบันเทคโนโลยีพระจอมเกล้าธนบุรี ว่าด้วย การบริหารผลประโยชน์อันเกิดจากทรัพย์สินทางปัญญา พ.ศ. 2538

6. ในกรณีที่มีผลประโยชน์เกิดขึ้นจากวิทยานิพนธ์หรืองานทรัพย์สินทางปัญญาอื่นที่ข้าพเจ้าทำขึ้น โดยมีมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรีเป็นเจ้าของ ข้าพเจ้าจะมีสิทธิ์ได้รับการจัดสรรผลประโยชน์อันเกิดจากทรัพย์สินทางปัญญาดังกล่าวตามอัตรากำหนดไว้ในระเบียบสถาบันเทคโนโลยีพระจอมเกล้าธนบุรี ว่าด้วย การบริหารผลประโยชน์อันเกิดจากทรัพย์สินทางปัญญา พ.ศ. 2538

ลงชื่อ.....ปานะ ใจครรชี.....ผู้อ่อนลิขสิทธิ์
(ปานะ ใจครรชี)

ลงชื่อ.....ผู้รับโอนลิขสิทธิ์.....ผู้รับโอนลิขสิทธิ์
(พศ.ดร.ณรงค์ฤทธิ์ วรารณ์)

ลงชื่อ.....พีรพัฒน์.....ผู้อ่อนลิขสิทธิ์
(พีรพัฒน์ คำเกิด)

ลงชื่อ.....น้ำตาล สุรากุล.....พยาน
(นางพรทิพย์ สิริจุติกุล)

ลงชื่อ.....สุจิตา.....ผู้อ่อนลิขสิทธิ์
(สุจิตา ยิ่งสกุลเกียรติ)

ลงชื่อ.....สาวนุร วงศ์กุลวงศ์.....พยาน
(นางสาวอากรนุร เชี่ยวชาญเกณทร)